



HAL
open science

Adaptive Collaborative Agent-Based System for Crisis Management

Sarah Ben Othman, Nesrine Zoghlami, Slim Hammadi, Hayfa Zgaya

► **To cite this version:**

Sarah Ben Othman, Nesrine Zoghlami, Slim Hammadi, Hayfa Zgaya. Adaptive Collaborative Agent-Based System for Crisis Management. 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Aug 2014, Warsaw, Poland. 10.1109/WI-IAT.2014.162 . hal-01716102

HAL Id: hal-01716102

<https://hal.science/hal-01716102>

Submitted on 5 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Collaborative Agent-based System for Crisis Management

Sarah Ben Othman, Nesrine Zoghalmi, Slim Hammadi
Ecole Centrale de Lille
BP, Villeneuve d'Ascq, France
{sara.ben.othman; nesrine.zoghalmi; slim.hammadi}@ec-lille.fr

Hayfa Zgaya
Institut Lillois d'Ingénierie de la Santé 42
Rue Ambroise Paré – LOOS, France
hayfa.zgaya@univ-lille2.fr

Abstract—Agent-oriented approaches highlight logistic concepts which give structure to large distributed systems. We present an innovative cooperative agent-based model for Crisis Management Supply Chain in order to solve a highly-distributed delivery scheduling problem. The Supply Chain studied in this work is complex and composed of several elements such as transport means, suppliers, areas in need, resources, etc. The objective is to find an effective and quick solution for resources (food, water, clothes, etc.) distribution to the areas affected by the crisis to minimize damages and especially human losses. We propose an adaptive collaborative decision support system based on agents' cooperation and interaction. Our solution is an innovative architecture both dynamic and generic founded on the alliance between multi-agent systems and optimization tools. Depending on the studied crisis situation, agents can use the appropriate optimization tool. In the proposed solution, we involved several agents having various properties and varied roles. Inspired by the fact that agents are autonomous and intelligent entities, we consider in this work the ability of agents to have more than one role which changes according to the nature of the crisis and the needs of the studied situation.

Keywords—distributed Systems; crisis management supply chain; scheduling; adaptive collaborative decision support system; multi-agent systems; optimization; roles.

I. INTRODUCTION

Natural disasters (earthquakes, floods, tsunamis etc.) or man-made hazards such as wars can threaten population's life and lead to harmful damages. It is often learned the hard way that catastrophes can come suddenly, so it is not effective to wait until calamity strikes to think about what should be done. Instead, detailed plans should be developed to cope with hazards and to save people at risk. The dynamic nature of crisis highlights the need for crisis management systems [1]. So, we must think about an operational coordinated and effective action plan, and its implementation in case of threat. The presented work is related to the definition of an agent-based architecture to simulate a Crisis Management Supply Chain (CMSC), which may at any time encounter delivery delays, poor consumption estimation, spontaneous consumption peaks and other unexpected events. All these emergencies are

likely to lead to stock-outs at any point in the Supply Chain (SC), which can have dramatic consequences that may engender human losses. To resolve these issues, a delivery scheduling problem is highlighted. The objective is to deliver the right quantity of needed resources to the areas affected by the crisis at the right time and with the minimum costs.

Crisis management systems where it is difficult to anticipate the environment's variations are starting to become quite attractive. However, working in an environment where changes are diverse, rapid and can suddenly occur incites to be equipped with different optimization tools able to simulate real and/or logistical situations likely to observe the behavior of different areas affected by crisis and identify the best strategies according to the situation of crises.

In cooperation with the logistics department of the European Aeronautic Defense and Space Company (EADS), in order to respond to these challenges, we propose to model and develop OBAC (Optimization Based on Agents Communication), a SC engineering tool developing and demonstrating technologies to improve logistics planning [2]. OBAC proposes a refined model for logistical needs under emergency conditions, taking into account existing intervention models, both military and non-military actions and previous results.

This work focuses on the processes involved in emergency and highlights the relationships between the different actors of the CMSC. Management's activities include forecasting crises and planning their treatment to have the time and resources to prepare a plan for crisis management in order to minimize the damage when they appear.

SC management can be improved through simulations. However, analytical models of Operational Research (OR), usually used in this type of problems, do not take into account interactions between entities. Simulation of Multi-Agent Systems (MASs) is a powerful tool to understand the dynamics of complex systems and to study possible change. It allows explicitly representing the phenomena of interaction and collaboration between the entities involved in the system. When these entities must cooperate in a decentralized manner to achieve a common task, distributed systems seem perfectly matched.

In this paper, we propose a distributed agent-based architecture for crisis management situation to optimize traffic flow in a crisis management supply chain. The presented solution consists on the use of a three-level agent-based framework including a set of optimization tools and a negotiation scheme for resources delivery to the areas affected by the crisis. Our study is based on two approaches. The first one is a generic scheduling approach involving a set of scheduling algorithms and thanks to agents' intelligence and autonomy; they can pick up the more suited algorithm to the studied crisis. The second tool is the ability of agents to change roles according to the complexity of the studied situation. A role is an abstract description of the responsibilities and functions of an agent [3].

This paper is organized as follows; the proposed solution is presented in Section 2. Section 3 describes the different used optimizing tools. Our system's convergence is studied in Section 4 followed by experimental results in Section 5.

II. PROPOSED SOLUTION

A. Proposed multi-agent system

Data of Crisis Management System are spread across a geographic network of a fairly wide range. The distribution of these data is highly random and inconsistent. It therefore requires a model that deals with their variability to represent effectively and in real time, the scalability of the environment incorporating them. MASs have shown their relevance to complex distributed applications design [4]. The concept of agent is not only an efficient technology, it is also a new paradigm for software development in which the agent is an autonomous entity operating in a dynamic environment [5] and interacting with other agents using languages and protocols.

A MAS is a set of homogeneous or heterogeneous interacting agents located in the same environment. These agents communicate and collaborate to complete individual or collective objectives or to solve problems beyond the capabilities or knowledge of each single agent. Agents can dynamically discuss how to partition a problem, how to distribute subtasks and exchange information to resolve possible dependencies between partial solutions and how to combine the partial results in the initial problem's solution.

To solve the multi-site scheduling problem we propose a dynamic Multi-Agent Organization (MAO) (figure 1). It considers each actor in the CMSC as an autonomous entity, able to exchange information with other actors. In our SC, actors are numerous and varied and multiple models are possible. We model the different areas of the SC through one or more agents called "zone managers", responsible for logistical areas and local delivery tasks scheduling. There are three types of zone managers:

- Metropolis_Agent (MA),
- Intermediate_Zone_Agents (IZAs),
- Terminal_Zone_Agents (TZAs).

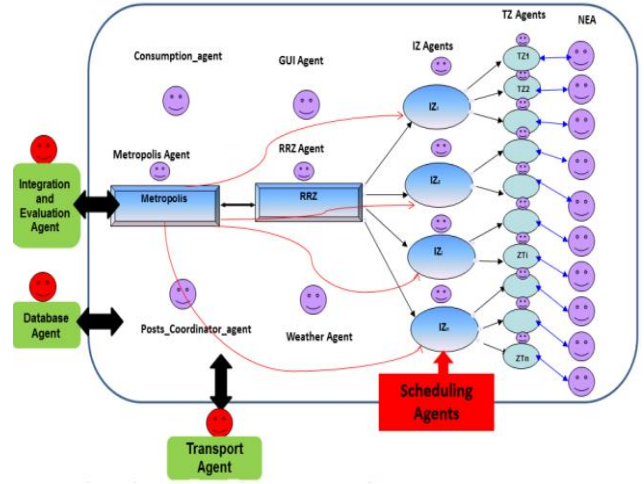


Figure 1. Architecture of the proposed CMSC

Flows exchange between areas is the communication result between the different managers of logistical areas. Other agents are also created for crisis management.

Once the crisis starts, the first agent created is MA which is responsible for crisis analysis and depending on the situation in the affected area, creates as many IZAs as necessary which are responsible for delivery tasks execution. Then, the created IZAs create TZAs which are in direct contact with Need_Estimating_Agents (NEAs). The role of NEA is to provide the needed estimated consumption in the affected areas. This agent is directly related to the Weather_Agent in order to get information about the environment in the relative area. It is also connected to the Posts_Coordinator_Agent and Consumption_Agent whose role is to ensure the smooth functioning of the SC. From the information provided by the Weather_Agent, NEA determines a handicap ratio by fuzzy logic, which represents the severity of climatic conditions, and corresponds to a need for additional resources. Taking into account the number of people in the theater and the daily handicap coefficient, the estimator gets an optimized value of resources to deliver to the areas in need.

Other agents are generated:

- GUI_Agent designed to interact with the system users [2].
- Database_Agent: communicates with all other agents and is responsible of providing information about resources, transport means and benchmark values of performance indicators.
- Integration_Evaluation_Agent who is charged to calculate performance indicators after receiving information from IZAs about the distribution of resources to the terminal areas, compares the performance indicators with given reference indicators and decides if the scheduling has succeeded. If the performance indicators exceed the benchmark values, IZAs execute the rescheduling after sending a request to the Transport_Agent to change the allocated means of transport for scheduling.

- **Transport_Agent**: is responsible of providing the means of transport requested by the IZA.

In a MAS, the change of status of certain objects in the agents' environment can affect their behavior and decisions. Thus, to optimize their choices and to guide their decisions, agents can be equipped with optimization approaches suited to their skills and knowledge in order to make combinatorial multi-objective decision. Hence the alliance between MASs and optimization methods; these two approaches are different but perfectly complementary.

B. Alliance between MASs and optimization methods through a 3-Level agent-based architecture

There are several optimization algorithms that can be involved in scheduling applications. But we can't find an efficient scheduling strategy that is common to all the crisis management tasks. The choice of tools depends on the crisis nature and the situation's complexity. Thus, we propose a three-level agent based architecture. We have chosen CMSC modeling based on communicating agents. In this model, agents representing the various actors in the SC are connected to the theater; they pump continuous information of the field level and compare the actual situation with logistics baseline. Based on these data as well as mathematical models, these agents will need to adapt their behavior in order to better respond to the ground disturbance, in order to go back to the previous standard situation.

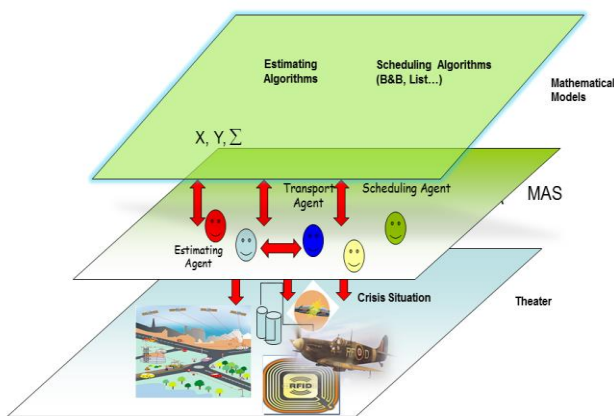


Figure 2. The proposed 3-Level agent-based architecture

Figure 2 shows the three levels representing the suggested architecture. The main level contains the MAS modeling the CMSC. In this level agents are collaborating and negotiating in order to make decisions on scheduling strategies. The decisions made depend on data received from the bottom level (the crisis situation). The higher level contains scheduling optimization tools including different mathematical models. As agents are autonomous entities characterized by decision-making capabilities, we propose to implement in this level a set of scheduling algorithms, and according to the complexity of the crisis situation, agents evaluate the global preference of a proposal to find out which scheduling algorithm should be

used in order to better respond to the needs of lower level. The global preference is based on the performance indicators calculated.

III. OPTIMIZING TOOLS

C. Scheduling Strategy

In SC management, we always try to minimize delivery costs and penalties due to delays. These objectives are within the framework of optimization problems. Several methods have been developed to help to find out optimal solutions using appropriate optimizing methods such as exact methods, heuristics, meta-heuristics, etc.

The advantage of using multi-agent paradigm for optimization approach is firstly to describe operation optimization which facilitates the design methods. So at first, MASs provide a powerful and flexible support for modeling and implementation of optimization solutions. Then, in terms of resolution, the distribution of calculation allows more efficient exploration of the search space while allowing a parallel implementation of the optimizing algorithms. MASs seem therefore suitable for solving a variety of combinatorial problems. This alliance allows simultaneous application of several local search algorithms, and offers us the opportunity to explore a large number of solutions. Indeed, the social features that characterize MASs, such as communication and cooperation, guaranteeing visiting a large number of solutions, something that turns out not possible if each optimization tool is used independently.

The combination of theoretical work in the field of optimization with the decision support systems based on communicating agents facilitates the efficient preparation and automation of large and complex human decisions. Agents collaborate to achieve a common goal shared by all agents, a goal set in advance by the designer of MAS. Agents can then be associated with one or more goals. For example, depending on the crisis degree and the complexity of the situation to solve, zone agents may decide to go, for the appropriate scheduling algorithm to execute delivery tasks. The choice is based on the observation of the real SC behavior and information received from other agents. Indeed, metropolis agent which is responsible for crisis analysis provides all information about the problem to treat such as the crisis degree. Actors can coordinate their activities via a specific strategy, a way of indirect communication through the modification of the environment.

Depending on the size of the crisis management problem and on:

- SC actors' behavior, tasks selection, performance indicators, environment modification;
- the relation input/output ;
- Amount of work done (complexity);
- Simplicity, clarity;
- Optimality;

agents can pick up one of these scheduling algorithms to execute delivery tasks:

- **Exact algorithms**

In case of simple crisis management problem, exact algorithms, such as tree exploration or linear programming coupled with filtering mechanisms can be very powerful. These methods provide accurate solutions, but the computation time is not polynomially bounded. That is why they are often reserved for instances of moderate size.

For example, it is interesting to use Branch and Bound algorithm to simulate problems whose structure such as tasks number and their execution time is known beforehand.

```

B&B algorithm
begin
  activeset :={0};
  bestval:=NULL;
  currentbest:=NULL;
  while activeset is not empty do
    choose a branching node, node  $k \in$  activeset;
    remove node k from activeset;
    generate the children of node k, child i,
     $i=1, \dots, n_k$ , and corresponding optimistic bounds
    obi;
    for i=1 to  $n_k$  do
      if obi worse than bestval then kill child i;
    else if child is a complete solution then
      bestval:=obi, currentbest:=child i;
    else add child i to activeset
    end for
  end while
end
end
```

In this algorithm, currentbest designs best complete solution found so far known as the incumbent. Its value represents the bestval which is used to find whether the nodes worth expending (creating children). In the other hand, the bound b represents an optimistic estimation for a partial solution (or node) once completed. However, there is no need to evaluate the node's children in the case where currentbest is greater than the bound b.

- **Heuristic algorithms**

They are able to find out quickly "good solutions".

A heuristic algorithm can solve a given optimization problem, but cannot guarantee an optimal solution in a minimum execution time. They can find a solution close to the best one and they find it fast and easily.

In case of emergency, the critical crisis can be treated with List algorithm in order to find a quick solution.

```

List Algorithm
Début
U:={}; t:=0;
while U<>I do
  begin
    if the subset of the tasks available in t is not
    empty then
      begin
        determine which of these tasks i that is of
        higher priority;
        ti:=t; U:=U+{i};
      end
    else
      determine the smallest time t when a task
      becomes available
    end
  end
end
```

This algorithm is particularly suited to the studied SC due to its dynamic priority rules. It is characterized by its flexibility and its easy implementation in real time. The problem is solved by static or dynamic priority rules. The principle of this algorithm is to maintain a list to hold all ready tasks appeared after a crisis.

- **Metaheuristic algorithms**

A metaheuristic refers to a general algorithmic scheme that can be applied to various combinatorial optimization problems. More specifically, it uses strategies that guide the search in the space of solutions; these strategies are independent of the problem. Metaheuristic algorithms are useful to solve combinatorial optimization problems (COP), a kind of the discrete optimization problems whose most are NP-hard. The basis of these algorithms is to guess the right directions in order to find the nearest optimal solution of complex problems so that the space searched, and thus the time required, can be significantly reduced.

Metaheuristics can be classified into 2 groups [6]:

- The population-based algorithms work on a population of solutions:
 - Genetic Algorithm,
 - Ant Colony Optimization,
 - Particle Swarm Optimization.
- The single-solution-based algorithms work on a single solution:
 - Hill Climbing,
 - Tabu Search,
 - Simulated Annealing.

Below is highlighted the behavior of zone agents (zone managers) which are responsible for tasks scheduling. According to the treated crisis degree, they choose the suitable scheduling algorithm. Here, depending on logistics choices and previous crisis management situations, we define 3 degrees of crises. If the crisis is of

“DEGREE1”, the scheduler goes for B&B algorithm, for genetic algorithm in case of more complicated crisis and for List algorithm if the degree is “DEGREE3” and urgent solution is requested.

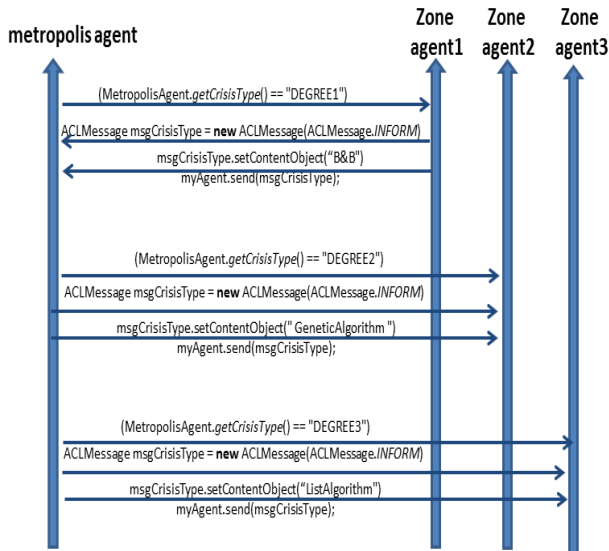


Figure 3. Interaction between zone agents and Metropolis_Agent

D. Role changing in the proposed MAS

MASs offer two perspectives for modeling such complex systems:

- **Agent:** autonomous software entity able to reason and interact by exchanging knowledge with other agents in the system to meet its own goals or the system’s goals. It is a computer system capable of acting independently and flexibly in its environment [7]. Each agent has a role that specifies the authorized behavior of agent in the organization through all the activities he can exercise.
- **Society:** all representations and mechanisms for managing interactions and organization or structuring agents in the system. Agents interact while respecting rules which constrain their behavior [8].

Although the concept of role is in the majority of work on MAS organization, it is not formalized. The role is often defined as a class of agent behaviors or services in the system [9 and 10], details are relegated to implementation. Conversely when there are formal definitions, they are often specific to the application [11 and 12].

A role R is defined by a set of tasks T. A mission expresses a possible behavior in the system in terms of authorizations or prohibitions on the goals and plans to follow combinations of actions and usable resources.

A role is an abstract description of the responsibilities and functions of an agent [3]. In [13] it can represent a form of identifying an agent and is based on a list of tasks to be performed sequentially or in parallel way. For Ferber, it describes the constraints (in terms of obligations, abilities,

and skills) and the consequences (in terms of capacity, authorizations) that the agent will receive in this role [14]. Hoogendoorn considers that assigning a role to an agent can be dynamic [15].

In this work, we consider that an agent can have more than one role, and depending on the situation’s complexity and the needs of system management, the role can change.

R is defined by a set of goals, plans, actions and resources authorized under the mission. And assuming a given role in an agent-based organization must have a behavior that satisfies the authorities specified in the missions of the role. A role is defined as follows:

$R = \langle G, P, A, R \rangle$ where G, P, Q, R respectively represent goals together, all plans, all actions and all resources authorized under the mission.

For example, in the organization structure of our application for CMSC, one role involves resources delivery to the areas affected by the crisis which is the mission of zone agent. This requires providing the needed transport means provided by Transport_Agent to the Scheduling_Agent. The objective afterwards, for Transport_Agent is to be able to change the means of transport allocated for rescheduling in case of predicted delays.

Therefore, transport means allocation is the role of Transport_Agent. This action is prohibited for other agent. It must comply with the demand of the scheduler. This plan is defined by the execution of the action A (providing a transport means) that requires resources such as transport means, fuel, driver, etc.

In case of rescheduling, the action B is executed (change the transport means). This action will be executed in the context of rescheduling.

While delivery, some incident can occur (accidents, break-downs, loss of control, etc.) leading us to think about anticipation. Thus, at any time Transport_Agent should be available to spot-on perturbations. The role here concerns just anticipation. This includes:

- Action C: Identification of dangerous events and vulnerabilities by analyzing the system and its dangers [16];
- Action D: Implementation of preventive measures and monitoring their effectiveness.

Transport_Agent executes its missions by alternating between these two roles.

The tasks performed in the system will depend not only on individual goals of agents, but also on the global goals of the CMSC.

IV. CONVERGENCE OF THE PROPOSED SYSTEM

The main difficulty induced by the use of MASs is that it is difficult to control the behavior and evolution of the overall system consisting of autonomous agents: for a given problem, it is not assured that all agents converge to a single common solution.

Referring to the definitions presented in the work of Jiang [17], we identified the leading agents and ordinary agents of our CMSC in our system:

- Ordinary agents:
 - Integration_Evaluation_Agent's strategy is to assess the validity of local solutions based on predefined performance indicators, and to send notifications to schedulers.
 - The strategy of the Transport_Agent is to allocate the means of transport and roads area for each Scheduler_Agent for routing resources.
 - The strategy of the Metropolis_Agent is the creation and deployment of CMSC.

Sufficient conditions are provided to guarantee the convergence of these agents.

- Leading agents
 - The strategy of each scheduler is to provide a local scheduling optimized for delivering resources to the affected areas, using optimization algorithms known in advance. Each scheduler can search into the mathematical level of our architecture to choose the algorithm best suited to their situation. Thanks to the distributed aspect of our approach, the input and output data in each scheduling problem are in limited quantities and planning tasks are also limited. Each zone agent is generally faced with a small scheduling problem. This should significantly save computing time and to ensure the convergence of each of our agents.

Starting from the assumption that the strategies of ordinary agents tend to converge towards the dominant strategies of leading agents, we obtain the global convergence of our system.

V. EXPERIMENTAL RESULTS

On March 11th in 2011, Eastern Japan was damaged by a powerful earthquake that struck the region and left serious problems. This earthquake was followed by nuclear plant crisis. This shocking tragedy invoked the transportation of foods, medicines, water, etc. in order to support the survivors. Thus, we will consider this crisis situation in order to evaluate the proposed solution. This scenario lasts three weeks:

- *Week 1:* Location of crisis due to floods;
- *Week 2:* Back to a stationary state;
- *Week 3:* New emergency situation due to nuclear risks.

We consider a total of 9 affected areas, metropolis zone (France) and three intermediate zones. The onset of the crisis detected by metropolis zone led to the creation of three intermediate zones: IZ1, IZ2 and IZ3 (Figure 4), and for each intermediate zone we consider three terminal zones to supply in Japan.

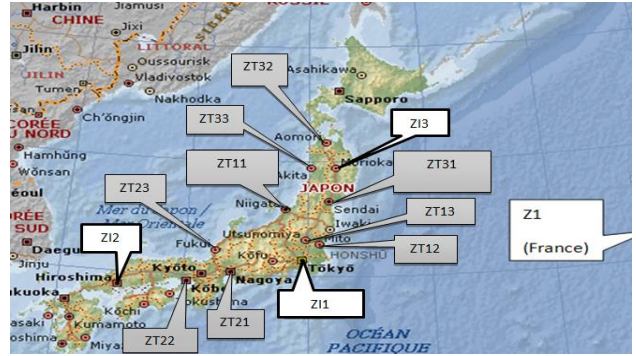


Figure 4. Implementation of the chain

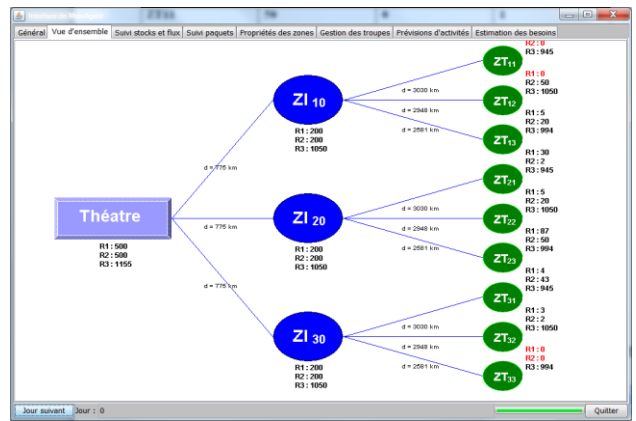


Figure 5. The structure of the simulated SC

For the means of transport, we propose the use of:

- Aircraft (Airbus A400M, being developed by EADS)
- Trucks logistics (based on characteristics WPK 440 4x4 logistics French Army)

For delivery, we assume that the transit time between intermediate zones and terminal areas is 3 days by land and one day by air. The transit time between metropolitan areas and intermediate areas is 2 days by land and one day by air. In our case the transport between the metropolis zone (France) and intermediate zones (located in Japan) can only be done by air.

A. Simulation 1: Location of emergency due to floods

At $t = 1$ day, the three IZ1, IZ2 and IZ3 agents receive all of the following queries:

- V1: {sender TZ11, receiver: IZ1, need: R2, quantity: 5}
- V2: { sender TZ12, receiver: IZ1, need: R3, quantity: 20}
- V3: { sender TZ12, receiver: IZ1, need: R1, quantity: 5}
- V4: { sender TZ21, receiver: IZ2, need: R3, quantity: 5}
- V5: { sender TZ22, receiver: IZ2, need: R3, quantity: 20}

- V6: { sender TZ23, receiver: IZ2, need: R3, quantity: 5}
- V7: { sender TZ33, receiver: IZ3, need R1, quantity: 5}
- V8: { sender TZ33, receiver: IZ3, need: R2, quantity: 20}
- V9: { sender TZ33, receiver: IZ3, need: R3, quantity: 5}

These queries are based on information provided by Need_Estimating_Agent. Each scheduler retrieves relevant information about the received request and performs scheduling by the arrival of tasks and within the requested delivery date. To execute delivery tasks we have chosen B&B algorithm due its simplicity searches. The goal is to find a feasible schedule that can minimize tardiness.

Integration_Evaluation_Agent assesses the costs of schedules generated. Performance indicators obtained are compared with reference values. As it is difficult to give an exact value for the costs, reference values used in these simulations are approximate values (depending on the location, the situation, the nature of the goods etc.). Their values are as follows;

Local Schedule	Indicamr1		Indicamr2		Indicamr3		Indicamr4	
	C _{max}	C _{avg}	C _{max}	C _{avg}	C _{max}	C _{avg}	C _{max}	C _{avg}
Z1	1,1	2,5	11,3	30	2,7	2	55	15
Z2	1,97	2,5	16,4	30	4,2	2	30	15
Z3	2,7	2,5	9,4	30	1,6	2	25	15

Figure 6. Obtained costs (K€)

In practice, we allow a percentage of cost overruns by reporting schedules to fixed costs. So a plan will not directly be rejected if its cost is higher than the reference cost but if that percentage is greater than a threshold. Integration_Evaluation_Agent detects the overflow at the IZ1 agent. A re-scheduling order is generated. The scheduler IZ1 must re-schedule and sends a request to the Transport_Agent to change the means of transport.

B. Simulation 2: New emergency situation due to nuclear risks

Suddenly, a nuclear plant crisis appears which exacerbates the situation. This has led to more resources to be routed to the areas affected by the crisis and then to more delivery tasks to be executed. Thus, an urgent solution is needed to save people and to avoid damages. What matters here is the rapidity of the solution and not the costs. So, after receiving a notification from MA, zone agents will go for List Algorithm in order to provide a quick and effective solution and minimize human losses. Scheduling tasks are performed one by one by the zone agent responsible for scheduling. Priority is given to the task whose requested delivery date is the smallest

considering a delay in delivery can cause dramatic consequences during the crisis.

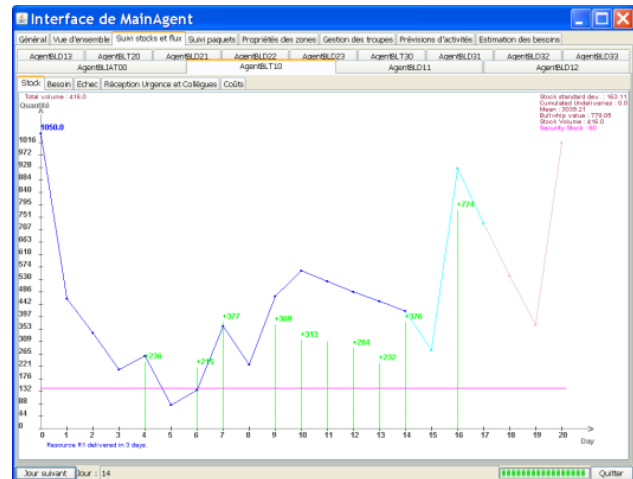


Figure 7. Stocks evolution in IZ1

Figure 7 shows stocks evolution of resources to be delivered to the areas in need. We notice a consumption peak in the day 16. Indeed, NEA related to IZ1 notifies this latter about this peak and asks him to send big quantities of resources to the affected areas to save people. IZ1 whose role is resources delivery is unable to afford all the needed quantities; thus, he sends a request to NEA to look for another for supply in order to reduce losses. In this case, so NEA changes his role and goes from estimating to scheduling. It uses safety stocks and sends request for replenishment. This request can be sent to Gui_Agent whose role is to communicate with users.

VI. CONCLUSION

In this paper, we proposed a distributed scheduling system for crisis management. We presented two optimization approaches for decision support in crisis management logistics. These approaches are integrated into a MAS whose behavior is to satisfy the needs in the affected zones in order to meet the needs of the supply chain's actors. The reactive proposed optimization approaches are based on the complementarity between MASs and optimization tools.

Approaches used in this research including supply chain management and MASs are flexible and scalable. For the future, improvements are possible for synchronization of flows between areas. We can also integrate the technology RFID (Radio Frequency Identification) for the SC traceability.

REFERENCES

- [1] Z. Yongsong, L. Siuming, Y. Kwokkit, Research of Crisis Management System based on CG and GIS. IEEE International Conference on Computer Science and Automation Engineering - CSAE, 596-599, 2011.
- [2] A. Kaddouci, H. Zgaya, S. Hammadi, F. Bretaudeau, PAAN: Partial Agreement Negotiation Network based on Intelligent Agents in Crisis Situation, International Journal of

Mathematics and Computers in Simulation, Issue 4, Volume 3, December 2009.

- [3] M. Wooldridge and N. Jennings, *Intelligent Agent: Theory and practice*, *Knowledge Engineering Review*, vol. 10, n° 2, pp. 115-142, 1995.
- [4] Ni. Wang, Y. Chen, L. Zhang, *Design of Multi-agent based Distributed Scheduling System for Bus Rapid Transit*. Third International Conference on Intelligent Human-Machine Systems and Cybernetics, 111-114, 2011.
- [5] N. Zoghalmi 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] C. Blum, A. Roli., AND M. Dorigo, *HC-ACO: The hypercube framework for ant colony optimization*. In Proceedings of MIC'2001—Meta-heuristics International Conference. Vol. 2.Porto, Portugal, 399-403, 2001.
- [7] M. Wooldridge, and Y. Jennings, *Intelligent Agent: Theory and practice*, *Knowledge Engineering Review*, vol. 10, n° 2, pp. 115-142, 1995.
- [8] N. Frishberg, S. Corazza, L. Day, S. Wilcox, R. Schulmeister, Sign Language Interfaces. In: Wiklund M. E. (ed): Proceedings of ACM INTERCHI'93 Conf. on Human Factors in Computing Systems, Panel, pp. 194-197, 1993.
- [9] H. Pattison, E., D. Corkill and V.R. Lesser., *Instantiating Descriptions of Organizational Structures* , in *Distributed Artificial Intelligence*, M. N. Huhns editors, pp. 59-96, 1987.
- [10] O. Gutknecht and J. Ferber, *An organizational meta-model for the analysis, design and implementation of multi-agent systems*. Act of JFIADSMA, 1998.
- [11] M. Prasad, V. Lesser and S.E. Lander, *Learning Organizational Roles in a Heterogeneous Multi-agent System*, in *ICMAS'96*, Kyoto, Japan, December 1996.
- [12] P. Stone, M. Veloso, *Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork*, in *ATAL'98*, Paris, France, 1998.
- [13] H. Savall, V. Zaretd, *The contribution of economics to human resources management*, *Staff review*, December, pp.54-61, 2000.
- [14] J. Ferber F. Michel, O. Gutknecht, *Agent/Group/Roles: Simulating with organizations*. Agent Based Simulation Agent Based Simulation 4, Montpellier, 28 - 30 Avril 2003.
- [15] M. Hoogendoorn, CR. Wetering, AM. Schols, MP. Rutten-Van Molken, *Inter Disciplinary Community-based COPD management (INTERCOM) cost effective*, *European Respiratory Journal* 35 (1): 79-87, 2010.
- [16] J. S. Rosenschein and M.R. Genesereth, *Deals Among Rational Agents*, 84-44 HPP Report, Stanford University, 1984.
- [17] J. Jiang and Y. Jiang, *Convergence at Prominent Agents: A Non-Flat Synchronization Model of Situated Multi-Agents* (Short Paper). *Proc. of 7th Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS 2008)*, Padgham, Parkes, Mueller and Parsons (eds.), Estoril, Portugal, pp.1493-1496, 2008.