Generic Need Estimating Agents for Resources Forecasting

Ayda KADDOUSSI, Slim HAMMADI, Emmanuel DUFLOS, Philippe VANHEEGHE
Ecole Centrale de Lille, Cité Scientifique – BP,
Villeneuve d'Ascq, France
E-mail :{ayda.kaddoussi;slim.hammadi;emmanuel.duflos;philippe.vanheeghe}@ec-lille.fr

and

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

ABSTRACT

The management and optimization of the crisis management supply chain is very complex and involves multiple concepts: supply resources, probability and statistics, means of transportation, disturbances on the delivery dates, and the importance of decisions involving human lives. To meet these requirements, multi-agent systems are a well-suited solution for modeling the supply chain through interactive autonomous entities. Our model represents a distributed logistic system in which flows of resources are hierarchically forwarded from one zone to another, taking into account the randomness of resources' consumption. The issue is then to optimize the procurement policy to avoid shortages that could cripple the whole system. In this paper we propose the multi-agent technology for modeling the different actors of the logistic chain. Then we propose an innovative method to estimate the future needs in resources for every zone. Our forecasting method is based on fuzzy calculations combined to ARMAX time series modelling. The objective of our work is to avoid, in a crisis situation, stock outs by optimizing the estimation of the future needs in resources for each zone and balancing the flows throughout the system.

Keywords: Multi-Agent System, Distributed Supply Chain, Estimating Agents, Fuzzy Logic.

1. INTRODUCTION

Accurate need forecast will lead to appropriate scheduling and planning for the operations in crisis management systems. Classical modeling tools are valid when used to model a stationary developing system, but when applied to dynamic and fast changing systems they have unsatisfactory performance. This is due to the fact that most classical forecasting techniques rely on static state assumptions and intrinsic data, without taking into account extrinsic inputs. The ARMAX (Auto-Regressive Moving Average with eXogenous inputs) model is a modeling technique which is capable of incorporating external inputs and is used as a forecasting method for fast developing systems.
In our work, we focus on a special kind of Supply Chain (SC): a distributed Crisis Management Supply Chain (CMSC) completely decentralized, thanks to a model based on communicating agents, and characterized by a hierarchical structure and a high expected disturbance impact. The SC represents all the links from the final customer up to the first level supplier. The main objective of such structure is the satisfaction of the final client. It is thus necessary to have an accurate need forecast of resources along the CMSC to optimize the circulating flows.

Within EADS's Logistic Information Systems department, we developed a logistic flows demonstrator for crisis management. We propose an approach, based on interactive autonomous entities, to represent the whole logistic chain. The multi -agents systems offer a modeling of the logistic system by proposing to represent its elements, their behaviors and their interactions directly under the shape of computer entities having their own autonomy.

In this paper, we describe, in a first time, the distributed logistic system. Then, a multi-agent architecture is proposed in the third paragraph, modeling the distributed CMSC characterized by the communication and the cooperation between its various entities. In paragraph 4, we focus on the optimizing agents for needs estimation, which can determine the future consumption of a given resource in a given area, based on fuzzy logic estimators and statistic optimization in order to have a better resource management. Finally, experimentations in paragraph 5 show the contributions of the proposed model.

2. THE DISTRIBUTED MILITARY LOGISTIC SYSTEM AND ITS CHALLENGES

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. Each area or zone must cooperate in order to satisfy the needs of the whole system. When casualties occur, the optimal routing of flows (of information, goods and persons) is one of the keys to success in facing crisis. The management of the flows of a Distributed Logistic System (DLS) spreads on several zones, starting from the resource's supplier ending to the customer. The general idea is then to route the flows leaving from a regrouping zone via intermediate zones to reach the terminal zones (zones of distribution to the customers). An optimal routing requires a communication between these different zones. The independent treatment of the zones can generate redundancies of information or erroneous data since every zone has an incomplete information and a limited capacity to solve the problem. These limits will be able to influence therefore on the global behaviour of the system. For this reason, the coordination of the zones proves to be a key element for the reliability of our system. To meet these multi participant requirements, a multi-agent cooperation model has been realised. In fact, multi-agent solution has the undeniable advantage of being a model that closely fits the reality and through which we will automate all aspects of the DLS. Intelligent agents co-operate and compete to reach the desired objectives designed by theirs 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.
In the next section we will discuss the relevance of using the multi agents' approach in the logistics' domain.

3. THE MULTI-AGENT SUPPLY CHAIN ARCHITECTURE

3.1 Multi-Agent System

The concept of Multi-Agent System (MAS) is intimately linked to that of Distributed Artificial Intelligence (DAI). The paradigm of Artificial Intelligence, to concentrate the intelligent capabilities into a single entity, proved to be insufficient to solve certain types of complex problems. To correct this limitation, a sub domain has emerged advocating the passage from an "individual intelligence" to a "collective intelligence" [2]. The MAS is so far an important branch of the DAI. The MAS permit to "model the behavior of a set of expert entities, organized more or less according to social type laws. These entities or agents have some autonomy and are immersed in an environment in which and with whom they interact" [1]. Ferber [3] defines an agent as a physical or virtual entity that is capable of acting in an environment, communicating directly with other agents and which is driven by a set of tendencies. A MAS is then a network of agents (solvers) weakly coupled that cooperate to solve the problems that pass the capacities or every agent's individual knowledge. These agents are autonomous and can be of heterogeneous natures [12] [13]. So, the multi agent solution plans 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. In the next section, we propose an original MAS architecture. It is hybrid architecture because it involves three different tools: a mathematical formalism, a MAS model and real environment data.

3.2 The proposed architecture

Our proposition is presented by three-level architecture in figure 1. The main level is the middle one, which corresponds to modeling the CMSC by a multi-agent system. In this model, agents working within the multi-agent system continuously receive information from the theater of operations. Based on this information and the various mathematical models available, these agents adapt their behavior to respond the best to the different disturbances that occur in the bottom level. It is interesting to see that the behavior of agents may suggest different actions and decisions, among them the correction and adjustment of the mathematical models. So, the originality of this approach consists in the fact that the agents may not follow the mathematical models blindly but they permanently try to correct and adjust it according to the real environment.

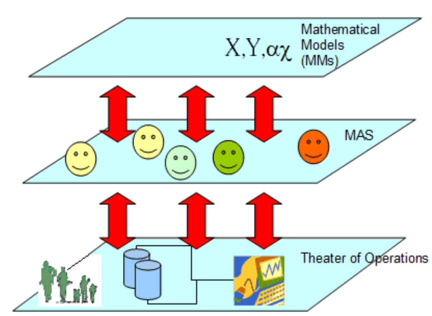

Figure 1. The three-level architecture

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. The hierarchical relationship between the different zone-agents of the CMSC is verified thanks to a social capacity function. Other agents can then intervene, and will help to smooth exchange of information and resources.

In our work, we focus on the estimation of the future needs in resources (e.g. water, food, medicine). To do that, we propose optimizing agents for multi-zones estimation needs, which is developed in the next section.

4. THE NEED ESTIMATING AGENT

The system is organized around these principal agents:

- An *Estimator Agent* responsible of the calculation of needs' estimations. The Estimator agent uses an ARMAX model to make the calculation. The architecture of the system is generic and open and allows us to add more estimators in the future to create a library of models.

- *TransactionDBAgent*: It is a data storage structure. We call *transaction* a set consisting of an estimation demand, the necessary parameters to calculate this estimation, its final result, and a post made feedback that tells about the quality of this estimation.

- *TestAgent:* It is a test interface that allows us to manipulate the NEA and verify its behavior. This agent gathers the needed information for calculating estimations.

The Need Estimating Agent (NEA) is a tool for decision support that is used to indicate to a zone-agent what it will need; using all the data provided. 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 data acquired on the field (how many persons to feed, etc.) and some estimation in order to create the membership functions and the rule matrix. Then, a statistical optimization process will be able to correct those estimations to make the more precise answer the NEA could calculate. [7][17]

4.1 Use of fuzzy logic

The advantage of this approach is that it allows us to propose a law according to our common sense, which can rather be realistic. This rule could then be improved to reflect in best the behaviour of the expert, without questioning for all that the rest of the system. The model starts with the data acquisition from the network of agents of the architecture. These data are then modeled using membership functions through fuzzification. The decisions are elaborated by fuzzy inference using fuzzy control rules. After the deffuzzification, the model offers a regulation of the average consumption of one resource as decision classes. These new suggested decisions will be added to the expertise used by the model [8].

We arbitrarily decided to treat the case of an order of clothes. The common sense allows us to determine certain number of parameters which comes into play in the consumption of clothes on a logistic theatre. Those parameters are temperature and humidity.

The NEA works as following:

- The need estimating agent is called by the zone to which it is attached to provide a need assessment of a given resource of the area.
- The need estimating agent contacts the weather agent, who shall provide all relevant information on humidity and temperature in the next 7 days.
- From this information, it determines a fuzzy logic coefficient, which represents the influence of the climatic conditions on the average consumption of the resource.
- Taking into account the present population and the fuzzy logic coefficient for each day, the estimator calculate a first non optimised value.
- Then the agent makes a correction of this value, taking into account the history of the consumptions, and straightens the value obtained by this first estimation.

4.1.1 The fuzzification

The fuzzification is the Numerical/Linguistic conversion of the input variables. For our application, we propose to take into account the values of Temperature and the Humidity in the considered zone to adjust the average consumption of clothes and to provide an accurate estimation of the future need in this resource.

** Membership functions*

We consider X to be a space of points, with a generic element of X denoted by x. A fuzzy class A in X is characterized by a membership function A μ (x) which associates with each point in X a real number in the interval [0,1], with the value of A μ (x) representing the grade of membership of x in A [4]. In other words, it is the degree to which x belongs to A.

Inputs: We classify the *Temperature* into the three sorts: *Cold*, *Temperate* and *Hot*. We classify the *Humidity* into the three sorts: *Dry*, *Temperate* and *Humid*.

So for a given Temperature t, we can determine its degree of membership in the 3 sets (*Cold, Temperate and Hot*).

The membership functions characterizing the fuzzy subsets of the variable *Temperature* are shown in figure 2:

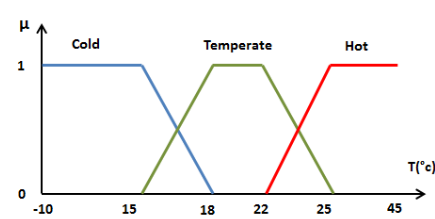

Figure 2. Membership functions for the Temperature

We proceed by the same way for the humidity.

4.1.2 Fuzzy inference

The fuzzy inference allows us to develop a decision by using the decision rules. Those rules are based on the experiences on the weather influence on the consumption of clothes. The decision rules are described by linguistic terms. The relationship between Temperature, Humidity and the variation of the average amount of clothes to be consumed is summarized in Table 1. This table is to be read: "If *Temperature* is Cold, and *Humidity* is Dry, then *Consumption* is Strong."

The inference rules describe the influence of the climatic conditions on the average consumption of clothes. (There will be an under consumption, an overconsumption or we will maintain the average consumption?).

	Cold	Temperate	Hot
Dry	Strong	Weak	Weak
Temperate	Strong	Average	Weak
Humid	Strong	Average	Average

Table 1. The inference rules

4.1.3 The deffuzification

The deffuzzification is the linguistic/Numerical conversion of the different variables. The method used in our application is the method of the center of gravity.

Output: We classify the variation of the average consumption of clothes into three sorts: *weak*, *average* and *Strong*.

- Set1: *Weak*: under consumption with regard to the average;
- Set 2: *Average*: about average consumption;
- Set 3: *Strong*: overconsumption with regard to the average.

The matrix of inference is then going to define the degrees of membership in each of these classes, according to the real values of the Temperature and the Humidity. The result refines the estimation of the consumption with regard to the average consumption.

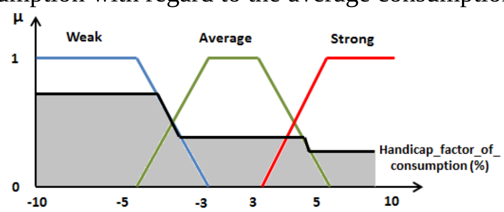

Figure 3. Membership functions for the consumption

The fuzzy logic offers a behavior 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.

4.2 ARMAX Optimization

The interest of the second step of estimation is to take into account the history of the consumptions to straighten the value obtained by this first estimation.

The NEA is a calculation engine that operates by learning, so the more it will calculate estimations, the more efficient it will be. For this, the NEA uses results of previous estimations and an assessment of these results to improve future calculations. Our system must react according to measures that require a mathematical model's representation of the phenomenon measured. Our objective in this phase is to approximate the function of consumption's variation by an ARMAX model for temporal series. In fact, the calculated measures will not be exactly aligned with a known mathematical model or law. Besides, the measures are most of the time mixed with a noise. The ARMAX model is capable of incorporating external inputs and model feedbacks.

- *The ARMAX model:*

The ARMAX process is one of the static time series models. It is suitable for the time series with no trend and seasonality that exhibit time homogeneity. The ARMAX model is a generalization of ARMA model which is capable of incorporating an external, (X), input variable. The form of the ARMAX model is:

$$\boxed{y^*(n+1) = \boldsymbol{A}^T\,\boldsymbol{y}(n) + \boldsymbol{B}^T\,\boldsymbol{x}(n) + \boldsymbol{C}^T\,\boldsymbol{e}(n)}$$

where $\boldsymbol{y}$ is an n dimensional vector of observed output variables (this corresponds to the real amounts of resources consumed on the field at the end of the day of calculation), $\boldsymbol{x}$ is an n dimensional vector of input variables (in our case, $\boldsymbol{x}$ is the input variables vector obtained after fuzzy logic calculation), $\boldsymbol{e}$ is an n dimensional unobserved disturbance vector process (white noise) and $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are polynomial matrices of the appropriate dimension.

The steps of analysis to identify the model that best suits the dynamics of resource consumption are presented as it follows:

a) Identification of the model: to build the ARMAX model, it is necessary to identify the different parameters. To do that, we use a Recursive Least-Squares (RLS) identification algorithm. Nelles [15] briefly summarizes the RLS algorithm with exponential forgetting. The RLS algorithm consists in calculating for each iteration:

$$\boldsymbol{e}(\boldsymbol{n}) = y(n) - \boldsymbol{\varphi}^{\boldsymbol{T}}(n)\boldsymbol{\theta}(n-1)$$
$$\boldsymbol{k}(n) = \boldsymbol{Q}(n)\,\boldsymbol{\varphi}(n)$$
$$\boldsymbol{\theta}(n) = \boldsymbol{\theta}(n-1) + \boldsymbol{k}(n)e(n)$$

Where **k** is the Kalman gain, $\boldsymbol{\theta}(n)$ contains the model parameters to identify, $\boldsymbol{\varphi}(n)$ is the state vector of the system and $\boldsymbol{\lambda}$ is a forgetting factor which reduces the influence of older terms in newly calculated values. If $\lambda=1$**,** all the data are weighted equally and the algorithm has an infinite memory length. On the other hand, with a smaller value of λ (and so a shorter memory length), the algorithm is better adapted to the system dynamics. The matrix $\boldsymbol{Q}$ is defined as:

$$\boldsymbol{Q}(n) = \lambda^{-1}\boldsymbol{Q}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\,\boldsymbol{x}^{\boldsymbol{T}}(n)\boldsymbol{Q}(n-1)$$

(Riccati equation).

b) Model checking: To evaluate the efficiency of the implemented model we use the Akaike information criterion (AIC)**.** Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. In the general case, the AIC is:

$$AIC = -2\log(\varepsilon_l^2) + \frac{2(l+1)}{M}$$

where l is the number of parameters in the statistical model, ε_l^2 is the value of the likelihood function for the estimated model and M the number of measures. The criterion takes into account both the statistical relevance of the fit and the number of parameters that have to be estimated to achieve this fit's level, by imposing a penalty for increasing the number of parameters [16]

c) Forecasting: this is the step of the data extrapolation through the found model.

5. SIMULATION RESULTS

We will consider three parameters of optimization for this logistic situation: the temperature, the humidity degree and the number of present people on the site. For the multi-agent system design, we use the platform Jade which integrates the agent's functionalities.

5.1 Study Case

- We are going to vary the number of people on the zone.
- The weather conditions are stable: 20 °, 90% of humidity.
- The middle debit of the resource is of 100 units per week for 100 people.
- The site has a life span of eight weeks. The first two weeks represent the establishment of the site, and the two last its evacuation.

5.2 Communication protocol

The communication steps are the following: The *TestAgent* retrieves the data filled in on the interface to build a *transaction object* and stores the transaction in the *TransactionDBAgent*. Then, the NEA receives a request for estimation. To do that, the NEA must before any operation go get the *transaction* from *TransactionDBAgent.* The *NEA* calculates a first value made up of membership functions and rule matrices. Then, it will optimize the fuzzy logic result according to ARMAX model and the historic of consumptions. The NEA returns the final result to both *TestAgent* and *TransactionDBAgent.*

We propose to compare the ARMAX model forecasting to three existent statistical estimators in the system:

- *RegLinAgent*: this agent's estimation algorithm is based on a linear regression model.
- *PTPAgent*: interpolation is based on a piecewise linear regression.
- *RegNonLinAgent*: this agent's estimation algorithm is based on a nonlinear regression model.

5.3 First Test

As the variation in the number of people on the site is linear, we compare in this series of tests the ARMAX results to the linear estimators (RegLinAgent and PTPAgent) results. Our application offers the possibility to use among those estimators the one(s) that seems the most appropriate; it means the one(s) that describes best the consumption variation.

week	Number of persons	Fuzzy logic estimation	Corrected value	ARMAX estimation	Statistical estim. error	ARMAX error
1	500	486,26	450	486,26	8,05%	-2,75%
2	1000	972,52	900	1033,37	8,05%	3,34%
3	1500	1458,79	1350	1520,54	3,92%	1,37%
4	2000	1945,05	1800	1995,171	2,59%	-0,24%
5	2500	2431,31	2250	2492,511	1,96%	-0,30%
6	3000	2917,5	2700	2992,701	1,57%	-0,24%
7	3500	3403,85	3200	3494,842	0,27%	-0,15%
8	4000	3890,11	3650	3995,67	0,38%	-0,11%

Table 1: First test

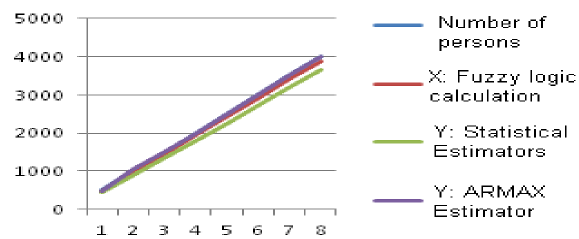

Figure 4. Need estimation for the first test.

Estimators need at least 2 points for statistical calculation. They therefore fall into account starting from the third week. The *statistical estim. error* corresponds to the error obtained using the linear statistical estimators.

The *Corrected value* is the actual amount that was consumed on the site. The graph shows that the estimation is refined over time to practically stick to reality and give an accurate estimation of the need expressed on the zone. The ARMAX model is more accurate in term of numerical precision.

5.4 Second Test

In this case, it is assumed that the variations of consumption follow a hyperbolic appearance. For this reason, we will compare the ARMAX model to all of the 3 statistical estimators (non-linear estimator included).

week	Number of persons	Fuzzy logic estimation	Corrected value	ARMAX estimation	Statistical estim. error	ARMAX error
1	500	486,26	450	486,26	8,05%	-2,75%
2	1500	1458,79	1350	1550,0985	8,05%	3,34%
3	2000	1945,05	1800	1995,86	6,10%	-0,21%
4	3000	2917,59	2700	3006,0967	1,98%	0,20%
5	2500	2431,31	2250	2442,912	1,60%	-2,28%
6	2500	2431,31	2250	2498,8213	1,34%	-0,05%
7	1300	1264,29	1170	1295,9613	0,02%	-0,31%
8	750	729,4	675	749,3133	0,02%	-0,09%

Table 2: Second test

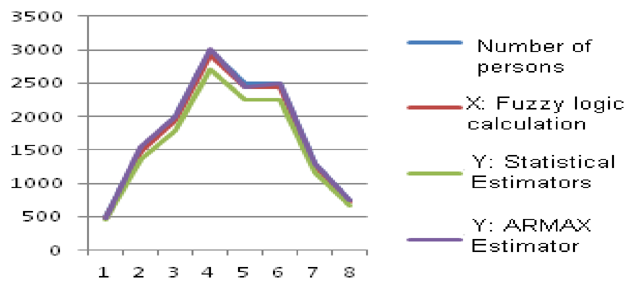

Figure 5. Need estimation for the second test.

The ARMAX model gives quite satisfying results. We obtain an error that is less than 2% from the 2^{nd} week.

The forgetting factor makes the algorithm closer to the system dynamics and more reactive than the statistical estimators.

In the future, we propose to develop a large library of estimators that cover as much mathematical models as possible. The operator will then choose to privilege the most adequate ones

6. CONCLUSIONS

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 condition throughout the CMSC. We propose a military logistic support tool that models its logistics functions, processes, expertise, and interactions between entities. Simulation results showed that this method seems to be effective because it allows an efficient estimation of the future needs in a zone stricken by a disaster. The NEA offers the possibility to combine the fuzzy logic technology with statistical tools to have the best estimation of future needs. It allows integrating as many estimators as the operator needs for his application.

In next works, we focus on the use of probabilistic models and scalable estimators to take into account the statistical characteristic of the phenomenon studied. We will also focus on the disturbed mode of the supply chain so we adopt an advanced interaction between the autonomous entities. Therefore, we propose a new form of anticipation to avoid, in a crisis situation, undesired states.

7. REFERENCES

[1] J. Erceau et J. Ferber, “ L’Intelligence Artificielle Distribuée”, La recherche, vol. 22, pp. 750-758, 1991.

[2] M.Larbarth, “ Modélisation et simulation orientées agents de chaînes logistiques, dans un contexte de personnalisation de masse : modèles et cadres méthodologiques ”. Thèse de doctorat, Université Laval, 2006.

[3] J.Ferber. “Les Systèmes Multi-Agents vers une intelligence collective ”. InterEditions, 1995.

[4] L.A. Zadeh. “Fuzzy Sets”. Information and Control, 8:338{353, 1965.

[5] C. da Costa Pereira and G. B. Tettamanzi, “Goal Generation with Relevant and Trusted Beliefs”. In Proceedings of AAMAS'08, pages 397{404, 2008.

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

[7] 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.

[8] L.A. Zadeh. “Fuzzy sets as a basis for a theory of possibility”. Fuzzy Sets and Systems, 1:3{28, 1978.

[9] N.Zoghlami, S.Hammadi. “Multi-agent model for a distributed logistics system”, In Proceedings of the 2006 IMACS Multiconference on Computational Engineering in Systems Applications (CESA’2006).

[10] V.Giard et G.Mendy, “ Production à flux tirés dans une chaîne logistique ”, Revue Française de Génie Industriel 26(1), pages 87-110, 2007.

[11] V.Giard et G.Mendy, “Exploitation of the knowledge of the final demand in the piloting of a logistics chain”, In Proceedings ILS 2006 (International Conference on Information Systems, Logistics And Supply Chain), 2006.

[12] K.S. Barber, C.E. Martin and A.Goel, “Dynamic adaptative autonomy in multiagent systems”. Journal of Experimental and Theoretical Artificial Intelligence, Vol.12, No.2, pages 129-147, 2000.

[13] B. Fayech, “Régulation des réseaux de transport multimodal: systèmes multi-agents et algorithmes évolutionnistes”, Thesis from « université des sciences et technologies de Lille et École centrale de Lille », 2003.

[14] P. Hajek. “Metamathematics of Fuzzy Logic”. 297 pp, Kluwer, 1998.

[15] O. Nelles, “Nonlinear system identification”. Pages 60-61, Springer 2001.

[16] B.S. Everitt, “The Cambridge Dictionary of Statistics”. Cambridge University Press, 1998.

[17] E. Duflos and P. Vanheeghe, “Estimation Prédiction”. Collection Sciences et Technologies dirigée par P. BORNE. Edition Technip