# Dynamic Fuzzy Cognitive Maps for the Supervision of Multiagent Systems

Aguilar Jose

**Abstract.** In this work we propose a Dynamical Fuzzy Cognitive Map (DFCM) where its casual relationships are based on fuzzy rules, in a way that the structure of the map changes during the phase of execution (runtime). We propose the modification of the values of the relationships between the concepts through of fuzzy rules derived from the concept states that represent the system modeled by the map. Our DFCM is ideal to build supervision systems for multiagent systems (MAS), in order to study the behavior of the agents community when they fail, use a lot of resource, etc. In this paper, the DFCM is used to build a supervision system for a faults management system based on multiagent systems. Very good results were obtained, demonstrating that the use of these maps as supervisor of multiagent systems is good and reliable.

## 1 Introduction

In [2, 4] Kosko introduces the Fuzzy Cognitive Maps (FCM) based on the Cognitive Maps of Axelrod [1]. The Fuzzy Cognitive Maps are a tool of causal representation, that are composed by concepts and causal relationships between the concepts, which use the theory of fuzzy logic to describe their structure and to infer answers of the map from input data. In this way, they represent the causal relationship between concepts and analyze inference patterns from a given input to the map. For the design of the structure of these maps, we can use the knowledge of the experts or the historical data of the phenomenon to model, being able to create models of complex systems for which an exact mathematical model is not possible.

These maps have been used for the strategic planning and the analysis of the behavior of the automobile industrial market [4, 7]. Other authors have proposed

---

Aguilar Jose
CEMISID, Departamento de Computación,
Facultad de Ingeniería, Universidad de Los Andes
Núcleo la Hechicera, Mérida 5101-Venezuela
e-mail: aguilar@ula.ve

M. Glykas (Ed.): Fuzzy Cognitive Maps, STUDFUZZ 247, pp. 307–324, 2010.
springerlink.com 

the use of the FCM to model supervision systems of complex control systems [8, 11] or to control plants [8, 17]. The FCM have been used also to model the behavior of systems in several areas like in political sciences [3, 7, 9], in medical diagnostics [21, 28], failure modes effects analysis and detection [10], software development modeling project [14], for the analysis and decision making [13, 15, 18], for the coordination of cooperative distributed agents [16], and many others. Other works have proposed learning approaches for FCM [19, 22], some of them based on the Hebb rule [20, 21, 26], Genetic Algorithms [23, 25], fuzzy rules [12], among others. Also, new FCM based in neural models have been proposed [5, 7].

An interesting domain of application of the FCM is in the area of supervision of MAS, that is, to study the behavior of a community of agents from initial hypothesis. In this way, we can determine and predict the individual behaviors of the agents according to this hypothesis. That can be very important to improve the design of the MAS, feedback the MAS for its self-organization, etc.

A first version of a dynamic approach of FCM, based on the random neural model, was introduced by Jose Aguilar in [6] (called DRFCM), to allow that the maps change their relationship during their runtime, of such form to be able to model systems of greater complexity than those than can model traditional FCMs. In that approach, random functions were used to provide the dynamic capacity within the maps, based on the random neural networks. In this work, we propose that the dynamics in the DFCM are giving by the modification of the values of the relationship between the different concepts that compose the map, through fuzzy sets derived from concept states of the system modeled by the map. This approach is ideal to build supervision systems for multiagent systems, in order to predict their behaviors when their agents fail, etc. We tested our approach in tasks of Supervision of Multiagent Systems for Faults Management.

This work is organized as follows. In section 2 is presented our proposition of FCM (the DFCM). Section 3 presents the case of study, where a DFCM is used like supervisor of a Faults Management System based on agents. Finally, the conclusions a further works are presented.

## 2 Dynamical Fuzzy Cognitive Maps

DRFCM are FCM based on the Random Neural Model, where the causal relationships are dynamics [6]. That is, the values of the arcs could be modified during the runtime of the FCM to adapt them to the new environment conditions. The quantitative concepts allowed us develop a feedback mechanism that was included in the causal model to update the arcs. In this way, with the DRFCM we could consider on-line adaptive procedures of the model like real situations. For example, our DRFCM could structure virtual worlds that change with time. The DRFCM does not write down differential equations to change the virtual world. It maps input states to limit-cycle equilibrium. A limit cycle repeats a sequence of events or a chain of actions and responses. Additionally, our DRFCM changed their fuzzy causal web during the runtime using neural learning laws. In this way, our model could learn new patterns and reinforce old ones.

In the DFCM we are going to use fuzzy sets adapted to the system modeled to obtain the dynamics of the causal relationship, unlike the proposition in [6], where a generic form of dynamics of the relationships was defined based on neural learning laws. With this approach we obtain a better adaptation of the FCM to the real system that it models.

## 2.1 The Causal Relationships Defined as Fuzzy Rules

The relationships are established using a series of fuzzy rules of the type "if the concept t is in $C_t$ then the casual relationship with concept i is $w_{ti}$", where $C_t$ is one of the possible states of the concept t and $w_{ti}$ will be the value of the casual relationships for this state. In this way, a set of rules defining the value of the relationship is used to determine the relationship value between two concepts.

To define the set of rules, we define a general procedure. For instance, we assume that the state of the concepts in the modeled multiagent system can be located in three zones, according to the following figure:

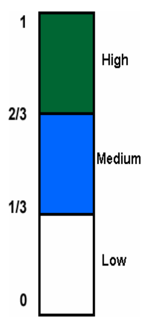


**Fig. 1** Identification zones of each concept's state

Using this figure, we can define the state of a concept like:

- A concept has a *high state* (between 2/3 and 1) when it works correctly and contributes substantially with the functioning of the modeled system.
- A concept has a *medium state* (between 1/3 and 2/3) when its functioning must be validated and its contributions to the systems' functioning is not so substantial.
- A concept has a *low state* (between 0 and 1/3) when it does not work and it does not contribute to the functioning of the system.

The state can be defined as a fuzzy variable composed by three fuzzy sets: high, medium and low. Additionally, the possible types of relationships between concepts can be defined like:

**Table 1** Possible types of relationships between concepts

| Value | Linguistic Variable |
|---|---|
| 1.00 | $Complete^+$ |
| 0.75 | $High^+$ |
| 0.50 | $Medium^+$ |
| 0.25 | $Low^+$ |
| 0.00 | Null |
| -0.25 | $Low^-$ |
| -0.50 | $Medium^-$ |
| -0.75 | $High^-$ |
| -1.00 | $Complete^-$ |

Also, the type of relationships can be defined as a fuzzy variable composed by nine fuzzy sets: $Complete^+$, $High^+$, etc. Now, we can define the following set of generic fuzzy rules using the concept states and the possible types of relationships defined previously, to define the casual relationships between concepts:

- If the preceding concept is **High** and the consequent one is also **High** then the relationship is **$Complete^+$**(1.0).
- If the preceding concept is **High** and the consequent one is **Medium** then the relationship is **$High^+$** (0.75).
- If the preceding concept is **High** and the consequent one is **Low** then the relationship is **$Low^+$** (0.25).
- If the preceding concept is **Medium** and the consequent one is **High** then the relationship is **$High^+$** (0.75).
- If the preceding concept is **Medium** and the consequent one is **Medium** then the relationship is **$Medium^-$** (-0.5).
- If the preceding concept is **Medium** and the consequent one is **Low** then the relationship is **$High^-$** (-0.75).
- If the preceding concept is **Low** and the consequent one is **High** then the relationship is **$High^-$** (-0.75).
- If the preceding concept is **Low** and the consequent one is **Medium** then the relationship is **$Medium^-$** (-0.5).
- If the preceding concept is **Low** and the consequent one is **Low** then the relationship is **$Complete^-$** (-1.0).

The set of generic fuzzy rules follows an adaptation mechanism similar to the hebb learning rule. These rules would be used to determine all the relationships between the different concepts. Thus, every relationship would be determined under the same set of rules, but each one would have a weight defined by the experts that could vary from relationship to relationship. For example, if we take the relationship between Concept 1 and Concept 2, and we assume that Concept 2 has a High state and that Concept 1 has a Medium state, then the relationship resulting from the rules would yield a high$^+$ value (that is, 0.75, if we suppose that we use crisp variables). This value is multiplied by the weight of the relationship defined by the expert (in this example we assume that the weight of this relationship is 0.5, then the final value of the relationship would be 0.375 (see figure 2)).

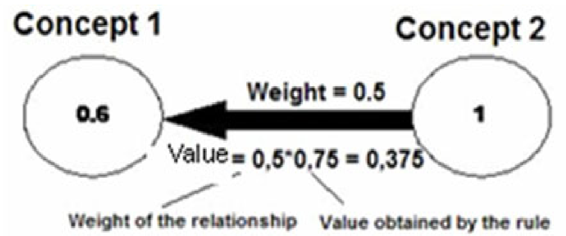


**Fig. 2** Example to establish the relationships value between concepts

We can establish equivalence among a concept state and an agent state. In our approach, each agent of a multiagent system is a concept, and the weight of the relationship is defined by the expert according to the relationship that it defines between the agents on the supervised multiagent system.

## *2.2 General Algorithm of a DFCM*

The design of a DFCM is very similar to the design of a DRFCM, which means we need to define the concepts (in or case, the agents) and the initial causal relationships among them. In our case, the initial causal relationships are defined according to the relationships of the agents on the agents' community. Once the DFCM has been designed, the execution algorithm is:

1. Define the Initial agent states $C^0 = \left[C_0^0, C_1^0, \ldots, C_n^0\right]$
2. While the system does not converge

   a. Calculate the values of the causal relationship using the procedure defined in the previous section.

   b. Calculate the current states using $C_j^t = \sum_{i=0}^{n} \left(w_{ij} \cdot C_i^{t-1}\right)$, where n is the number of agents.

## *2.3 Relationships Initial Weight Definition*

The initial weights can be defined according to the next procedures:

- Each expert defines its FCM. For the allocation of the weights (opinion of the experts) is used a scale from the 0 to the 1, where 0 represents that the antecedent concept (agent) has not influences over the consequent concept (agent), and a weight of 1 indicates that the consequent concept is sensible to the changes of the antecedent concept.

- We determine a global FCM. For the final calculation of the weight of the concept (global FCM) we can use two formulas:

$$E_{ji}^{G} = \max_{e}\{E_{ji}^{e}\}, \ \forall\ e=1, NE \text{ (number of experts)} \quad (1)$$

or

$$E_{ji}^{G} = \sum_{e=1}^{NE} b_e E_{ji}^{e} / NE \quad (2)$$

Where $E_{ji}^{e}$ is the opinion of the expert e about the causal relationship among agents $C_j$ and $C_i$, and $b_e$ is the expert's opinion credibility weight.

# 3 DFCM Like Supervisor of Multiagent Systems

In this section we test the utilization of the DFCM to supervise a reference framework based on MAS to model the Industrial Automation.

## *3.1 The IDCSA Reference Framework*

In this work we study the reference framework "Intelligent Distributed Control system based on Agents" (IDCSA) proposed in [24] to model industrial automation systems. The IDCSA is a multiagent platform designed specifically for control systems (see figure 3).

The IDCSA proposes the next set of agents, which represent the present elements in a loop of control process, with the intention to establish a generic mechanism for the handling of the activities related to the control process:

- *Observer Agent:* this agent will have the mission of gathering the data coming from the repositories of data that can give information about the state of the processes. Also, it can pre-process and/or validate the data, calculate averages and estimations, etc.
- *Controller Agent:* this agent receives the state information emitted by the Observer agent and compares the current conditions of the process with the conditions wished for the same on. In the case where the current conditions are moved away from a certain band of tolerance, it executes control structures, what can be orders of activation of alarms, orders for execution of applications of diagnostic, etc.
- *Actuator agent:* depending on the decisions taken by the controller agent, active alarms and makes them visible for each actor involved with the resolution of the problem (SCADA operators, engineers of optimization, maintenance engineers, etc.), produces changes in the SCADA (for example, it changes the set point), etc.
- *Coordinator agent:* it supervises the operation of the control system and it modifies it if is necessary, changes established values for conditions of normal operation (for example, value nominal for process variables), executes

tests that allow to identify and to locate faults, emits the requirements of services to the specialized agents, etc.

- *Specialized agents:* inside the detection process and diagnostic of faults, maybe is necessary to carry out activities of data mining, mathematical and statistical calculations, predictions, etc. These activities are carried out by specialized agents, each one of them with a specific task to carry out.


**Fig. 3** IDCSA Model

### 3.1.1 IDCSA and the DFCM

In this work we study the behavior of the agents within IDCSA; for that, we develop a supervisor system based on DFCM. In this way, we can study the relationship between the agents that compose IDCSA by means of the DFCM. In this case, each agent of IDCSA will be a concept (see figure 4).

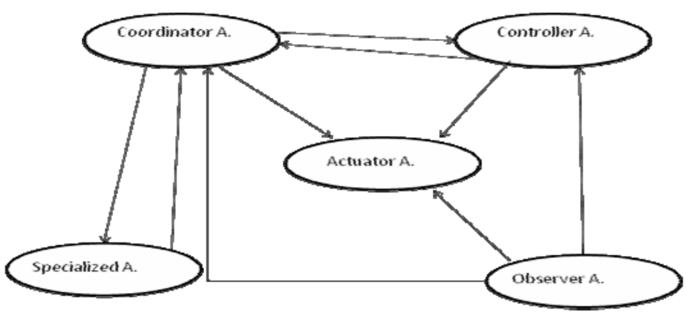


**Fig. 4** Example of the DFCM for IDCSA

To test the performance of the DFCM, in the next part we are going to use the modeled of a faults management system developed in [28] using IDCSA.

### 3.2 Fault Management System (FMS)

The FMS proposed in [27] is composed by two subsystems: the first subsystem accomplishes Monitoring and Fault Analysis Tasks (MFAT); the second subsystem accomplishes the Maintenance Management Tasks (MMT). The FMS and the Engineering Management define together the productivity indexes, the human and financial resources, the components stock, among other things. On the other hand, the FMS also interacts with the Fault Tolerant Controlled Process, it being the final receiver of fault detection-diagnosis-decision tasks (see figure 5).

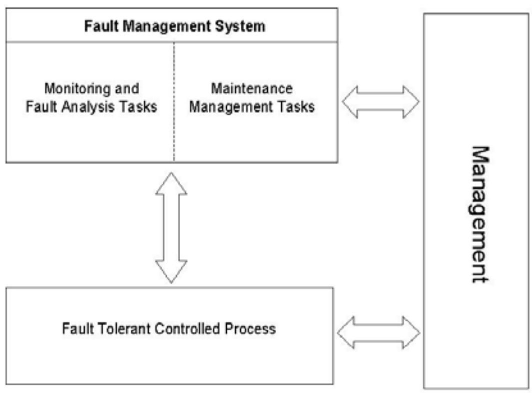


**Fig. 5** The FMS

The MFAT subsystem is concerned with the following tasks:

- *Detection:* this task permits the identification of an invalid state in the process. A fault presence is stated from the behavior of significant variables; as a consequence, based on this behavior, the systems may be fault-free, in abrupt failure or incipient fault. The detection task needs detection models and the information from the process.
- *Isolation:* this task estimates the place where is occurring the faulty item.
- *Diagnosis:* This task determines the failure mode related with the fault detection, their causes and consequences. The diagnostic task needs diagnosis models and the information from detection and isolation tasks.

The MMT subsystem is related with the following tasks:

- *Prediction*: this task tries to estimate when an incipient fault goes to a functional failure. This task needs predictive models, and the information from fault detection and diagnosis tasks may be used in order to build these models.
- *Planning*: this task provides a preventive maintenance plan proposing the maintenance tasks that avoid the occurrence of a functional failure. It must also propose a contingency plan in case of abrupt failure, in order to avoid fatal consequences on the process.

- *Maintenance Action:* is concerned to set up and running the maintenance tasks according to the maintenance plan. These tasks are on-condition (detection-isolation-diagnosis) and on-time tasks. Maintenance actions are also related to set up and running contingency plans in case of abrupt or unexpected failures.

### 3.2.1 MAS-Based Reference Model for FMS

The FMS as MAS should consider the following items:

- Information exchange between the levels of the industrial processes.
- Monitoring and analysis of variables from the lower levels of the industrial processes (Local Control).
- Fault detection, isolation and diagnosis mechanisms and reasoning mechanisms supporting the decision-making process in the preventive maintenance.
- Distributed processing: the FMS activities as fault detection/isolation and estimation of working index should be embedded into a distributed computational model.

Consequently, the FMS works like a system where their modules interact of cooperative form, of way to reach the objective of the process. The FMS provides the following functionalities: Monitoring, faults management system, Detection, Isolation and Analysis of failures, Predictions of failure occurrence, Scheduling of preventive maintenance tasks, Set up and Running of preventive plans and corrective actions.

These functionalities can be embedded into the agents defined in the generic conceptual framework IDCSA [24], which has been adapted to the fault management problem. In this sense, this problem is defined like a generic problem of feedback control system. In this way, using the framework IDCSA eight agents are defined: Detector Agent (specialized agent), Finder Agent (specialized agent), Diagnostician Agent (specialized agent), Predictor Agent (specialized agent), Coordinator Agent, Controller Agent, Actuator Agent, and Observer Agent.

The *Detector* agent identifies if a component is under the presence of an incipient fault. The *Finder* agent looks up for the exact site where the fault is happens in the system, if it is not determined by the Detector agent. The *Diagnostician* agent determines the way of fault, its causes and their consequences. The *Predictor* agent prevents that an incipient fault becomes in a total functional fault in the system.

The *Coordinator* agent gathers the information about the process' items from the specialized agents and, based on this information, it schedules the maintenance tasks. The timeline should be defined according to the item's reliability and the failure effects. These aspects can suggest a decision-making process: take a corrective action or redefine the preventive maintenance plan; the last is also done if a maintenance task is not performed. This timeline can include a long time horizon (Long Term (LT) Plan) and not take into account the human resources and inventory on hand. In this sense, the *Controller* agent takes the preliminary plan provided by the *Coordinator* agent and it proposes a short time horizon plan (Short Term (ST) Plan) satisfying the human resources, inventory on hand and critical claims. That permits to give a daily or weekly plan for easy tracking. On-condition maintenance tasks are notified to the *Coordinator* agent; On-time and corrective maintenance tasks are notified to the *Actuator* agent. The *Observer* agent gathers the information about the process control in order to determine if a functional failure occurs; it also notices about the maintenance state and correlates it with the process state. Maintenance state is related with the foreseen maintenance tasks performance and process state is related with the operational function.

In figure 6 we observe the FMS based on the IDCSA framework (the specialized and the Coordinator agent in the Supervisor level; and the Controller, Observer and Actuator agents in the Process level). In addition, the FMS interacts with the IDCSA of the control process through the Middleware [27].

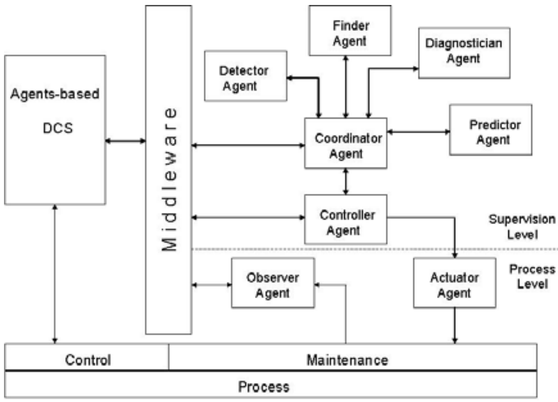


**Fig. 6** MAS for the FMS

### 3.2.2 DFCM for the FMS

We assume that each concept of the map will represent an agent in particular, for that reason we need to define the relationship between the agents of the FMS:

- *Detector Agent:* it work with the coordinator agent informing if the system is in an incipient fault, this task is carried out monitoring variables of the system. A bad detection can cause false alarms and to cause that false plans of maintenance are realized, this entails to a bad operation of the coordinator agent. A good detection can even specify the location of the fault saving therefore the work of the finder agent. The causal relationships that affect to the Detector agent are from the observer agent, which indicates the state of the different variables, and from the process and the middleware.
- *Finder Agent:* Once a fault is detected and its detection does not indicate its location, this agent must locate the place where the fault is happens. If it indicates a false location can produce the accomplishment of unnecessary tasks, and an excellent detection can facilitate the work of the coordinator agent. This agent is affected by the Coordinator and Detector agents and the Middleware.
- *Diagnostician Agent:* this agent determines the type of fault and communicates it to the coordinator agent. A bad diagnostic can entail to plans of maintenance of low quality, the coordinator can generate maintenance plans erroneous if this agent does not determine the type of fault well. This agent is affected by the controller and coordinator agents and the Middleware.

- *Predictor Agent:* this agent avoids that an incipient fault becomes in a total fault of the system. It has a relationship with the Coordinator agent to indicate it if the fault possibly entails to a total collapse of the system. This agent is affected by the Coordinator, the Observer and the diagnostician agents, and the Middleware and the process.
- *Coordinator Agent:* it collects the information of the specialized agents to carry out the plans of maintenance. A bad coordination can waste resources and a total collapse of the system. On the other hand, a good coordinator can try to detect the bad operation of some of the specialized agents and to try to resolve that situation. This agent is affected by the specialized and the Observer Agents, and the Middleware.
- *Controller Agent:* it defines the plan of preventive maintenance to be applied. This agent has relationship with the actuator agent, which executes the maintenance plan, and it is affected by the Coordinator and the observer agents, and the Middleware.
- *Actuator Agent:* this agent executes the maintenance plan and it has relationship with the process. This agent is affected by the Controller and the observer agents, and the process.
- *Observer Agent:* it observes the system and monitors the execution of the maintenance plans, for that reason is affected by the actuator agent and the process.
- *Middleware:* the data of this concept in the map is obtained by the Observer agent, which is the unique one that affects it.
- *Process:* it establishes that so good is the communication of the agents with the process and it is affected by the actuator agent.

The resulting cognitive map is shown in figure 7.

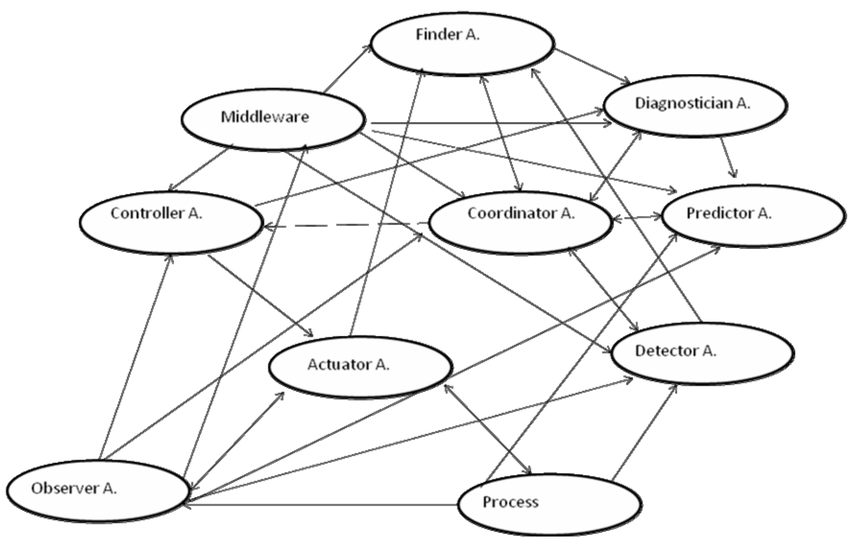


**Fig. 7** The DFCM for the FMS

In our case of study, an agent is in *high state* when it fulfills all its functions correctly, it can use new mechanisms of inference successfully, and it does not occupy the resources in an excessive form; in addition, the conclusions and decisions taken by the agent are considered correct in all cases. One agent is in *medium state* when it fulfills its functions but it is not able to use new mechanisms of inference; in addition, the conclusions and decisions taken by the agent must be validated. Finally, an agent is in *low state* when it is not able to use new mechanisms of inference, it excessively occupies the resources of the system, and the conclusions obtained by the agent are considered erroneous and false.

The set of generic fuzzy rules are used to calculate all the relationships between the different concepts (agents) of the DFCM for the FMS. Previously, we have defined the relationship weights according to the experts. We have consulted several experts to define these weights and used the equation 2 to calculate the global FCM, for $b_e$=1. The global FCM is shown in table 2.

**Table 2** Weights of the casual relationships for the DFCM of the FMS

| | Coordinator A | Diagnost A | Predictor A | Finder A | Detector A | Controller A | Actuator A | Observer A | Midleware | Process |
|---|---|---|---|---|---|---|---|---|---|---|
| Coordin. A. | - | 0.31 | 0.27 | 0.38 | 0.2 | 0.28 | - | - | - | - |
| Diagn. A. | 0.13 | - | 0.17 | - | - | - | - | - | - | - |
| Predictor A | 0.13 | - | - | - | - | - | - | - | - | - |
| Finder A. | 0.13 | 0.19 | - | - | - | - | - | - | - | - |
| Detector A. | 0.13 | - | - | 0.14 | - | - | - | - | - | - |
| Controller | - | 0.12 | - | - | - | - | 0.60 | - | - | - |
| Actuator A. | - | - | - | - | - | - | - | 0.23 | - | 1.0 |
| Observer A | 0.10 | - | 0.10 | - | 0.3 | 0.35 | 0.20 | - | 1.0 | - |
| Middleware | 0.34 | 0.38 | 0.34 | 0.47 | 0.3 | 0.35 | - | - | - | - |
| Process | - | - | 0.10 | - | 0.2 | - | 0.20 | 0.77 | - | - |

## 3.3 Results Analysis

We have tested the map for different inputs. Particularly, we have observed that there is a large sensitivity of the complete system to the changes in the Middleware, which is the most influential concept of the system. If this concept has a low value the concepts of the rest of the system will be 0 (zero) (see figure 8).

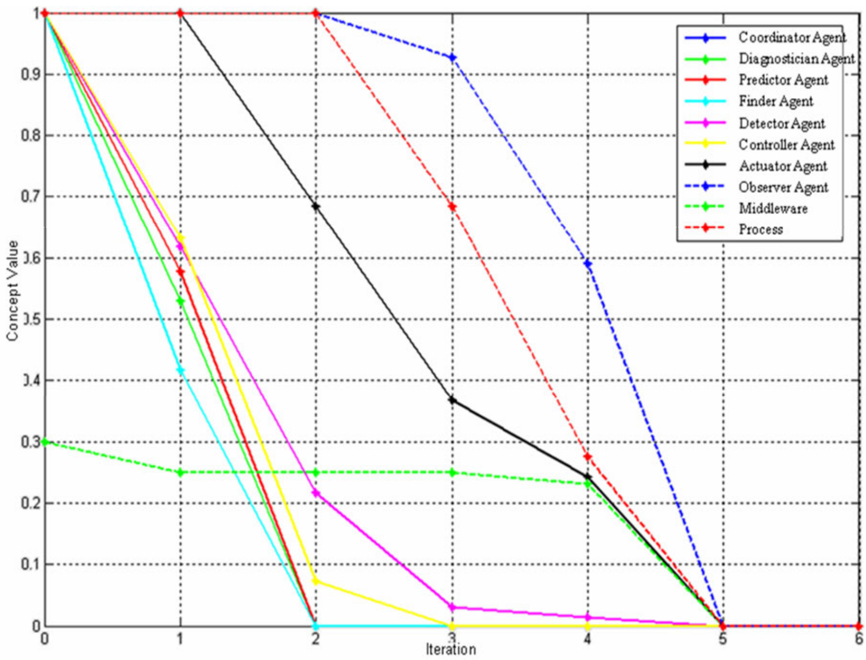


**Fig. 8** Evolution of the agents with a Middleware with value of 0.3

We can see that the first concepts that begin to decay are the specialized agents, the coordinator and the controller agents, to a very fast rate. In iteration 1 is observed that the actuator agent begins to very quickly decay due to the fall of the controller agent, and finally, from iteration 2 the observer agent begins to fail, which entails to a total fall of the system. That is due because the agents use the Middleware to collect data to carry out their tasks. In this way, this concept (the middleware) must be in a good state to avoid situations like the predicted in figure 8. When the middleware works badly, the rest of the community of agents begins to handle erroneous information, entailing to a total fault of the system.

Another very important agent in the system is the coordinator agent. In a state of a coordinator agent where it does not work well, the system tends to fall in a bad operation (see the figure 9).

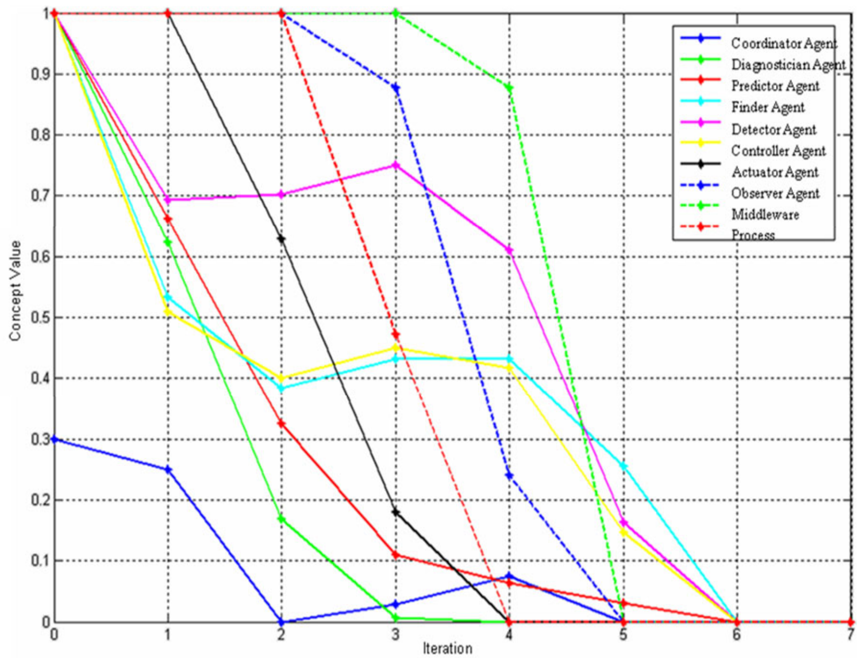


**Fig. 9** Evolution of the DFCM for the FMS with a bad coordinator agent

As it is possible to observe in this figure, the specialized agents and the controller agent are themselves affected by the bad operation of the coordinator agent. That is due because these agents use the information of the coordinator agent to carry out their tasks, if the coordinator does not obtain good results and excessively occupies the resources of the system, the specialized agents are affected.

Another interesting test is when the specialized agents were near the threshold between a high and medium state. In this case, we can observe that these agents significantly diminish the operation of all the system (see figure 10).

In the figure 10 we can see that the specialized agents very quickly diminish the behavior of the coordinator agent. Once the coordinator agent does not work absolutely well, the controller agent diminishes the quality of its work since the maintenance plans are generated by the coordinator agent. This type of fault allows that the system continues in operation, but the adjustment must be made in some future time.

Another case of test is when we have a bad actuator agent (see figure 11). A bad actuator agent makes fall the process, falling therefore the quality of the behavior of the observer agent. Once the observer stops working well, the Middleware begins to obtain erroneous data, and as we already saw previously, once this concept begins to fall, the rest of the system fall quickly.

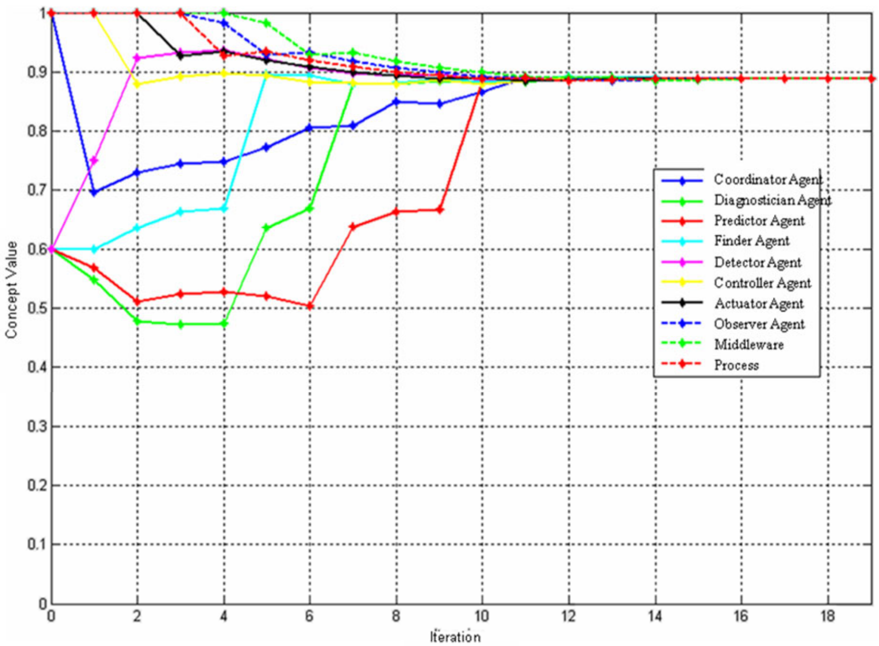


**Fig. 10** Evolution of the concepts of the DFCM with specialized agents of medium quality

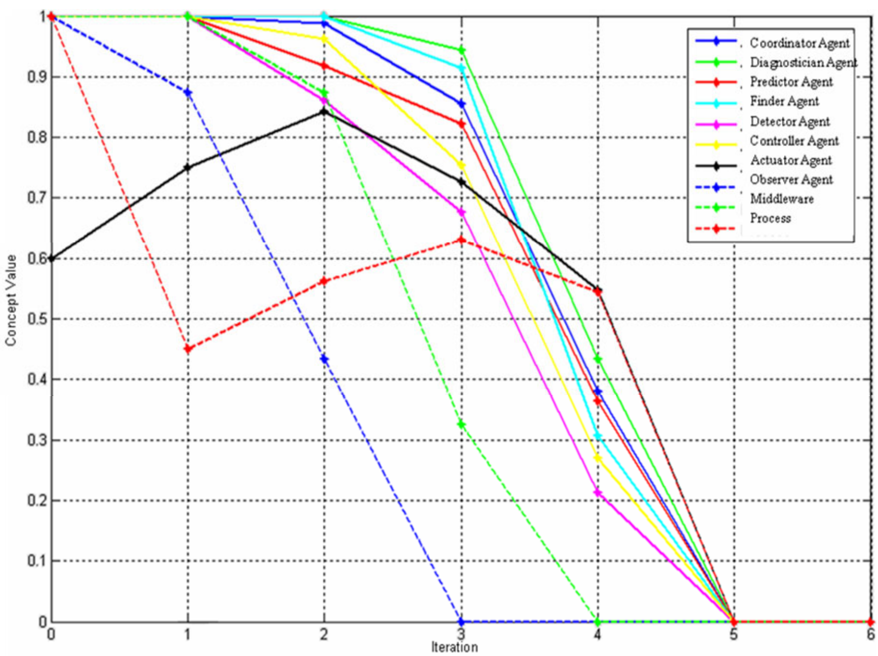


**Fig. 11** Evolution of the DFCM with a bad Actuator agent

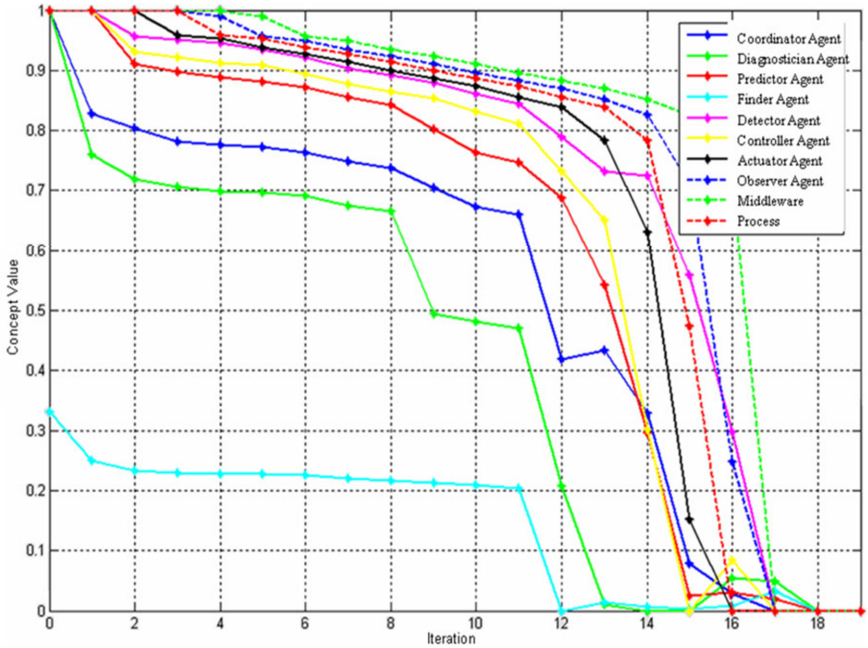


**Fig. 12** Evolution of the agents with a bad finder agent

The last test is shown in figure 12, where we can observe that from a state where the finder agent does not work absolutely well, its fault is propagated slowly to the community of agents. Particularly, the diagnoses generated by the diagnostician agent and the plans of maintenance of the Coordinator Agent try to adjust to this situation, before to start to work bad. This type of problem does not entail to a total fault of the system immediately, but in a very near future time.

## 4 Conclusions

The cognitive maps have demonstrated to be a modeled tool of effective, even more their extensions. The utilization of the DFCM like a supervisor system of MAS demonstrates to be a good tool due to that the behavior of a community of agents can be studied. Specifically, we can study as the bad or good operation of an agent in particular can affect the rest of the community of agents, allowing so make decisions based on the predictions from the map.

Only a previous work has been developed to study MAS using FCM [29], specifically the casual relationship among the agents. In our approach we can model that, additionally we can analyze the behavior of the MAS, predict the behavior of the agents, etc. New extensions to our approach to define different forms of casual relationships can be done. For example, we can define casual relationships like mathematical equations. Other possibility is to define each concept like fuzzy variables, and then the casual relationships are defined like fuzzy relationships among fuzzy variables. Next works will study these approaches.

## References

[1] Axelrod, R.: The Structure of Decision: Cognitive Maps of Political Elites. Princeton University Press, Princeton (1976)
[2] Kosko, B.: Fuzzy Cognitive Maps. International Journal of Man-Machine Studies 24, 65–75 (1986)
[3] Eden, C.: On the Nature of Cognitive Maps. Journal of Management Studies 29, 261–265 (1992)
[4] Kosko, B.: Fuzzy Engineering. Prentice-Hall, New Jersey (1999)
[5] Aguilar, J.: A Fuzzy Cognitive Map Based on the Random Neural Model. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 333–338. Springer, Heidelberg (2001)
[6] Aguilar, J.: A Dynamic Fuzzy-Cognitive-Map Approach Based on Random Neural Networks. International Journal of Computational Cognition 1, 91–107 (2003)
[7] Aguilar, J.: A Survey about Fuzzy Cognitive Maps Papers (invited paper). International Journal of Computational Cognition 3, 27–33 (2005)
[8] Goto, K., Yamaguchi, T.: Fuzzy Associative Memory Application to a Plant Modeling. In: Proc. of the International Conference on Artificial Neural Networks, pp. 1245–1248 (1991)
[9] Dickerson, J., Kosko, B.: Fuzzy Virtual worlds as Fuzzy Cognitive Maps. Presence 3, 173–189 (1994)
[10] Pelaez, C., Bowles, J.: Using fuzzy Cognitive Maps as a System Model for Failure Models and Effects Analysis. Information Sciences 88, 177–199 (1996)
[11] Stylios, C., Groumpos, P.: The challenge of modelling supervisory systems using Fuzzy Cognitive Maps. Journal of Intelligent Manufacturing 9, 339–345 (1998)
[12] Carvalho, J., Tomé, J.: Rule Based Fuzzy Cognitive Maps – Fuzzy Causal Relations. In: Mohammadian, M. (ed.) Computational Intelligence for Modelling, Control and Automation, pp. 276–281. IOS Press, Amsterdam (1999)
[13] Khan, M., Quaddus, M.: Group Decision Support using Fuzzy Cognitive Maps for Causal Reasoning. Group Decision and Negotiation Journal 13, 463–480 (2004)
[14] Sharif, A., Irani, Z.: Exploring Fuzzy Cognitive Mapping for IS Evaluation. European Journal of Operational Research 173, 1175–1187 (2006)
[15] Zhang, W., Chen, S., Besdek, J.: Pool2: A generic system for cognitive map development and decision analysis. IEEE Trans, Systems Man Cybernet 19, 31–39 (1989)
[16] Zhang, W., Chen, S., Wang, W., King, R.: A Cognitive Map Based Approach to the Coordination of distributed cooperative agents. IEEE Trans. Systems Man Cybernet 22, 103–114 (1992)
[17] Stylios, C., Geogrgopoulos, V., Groumpos, P.: The use of Fuzzy Cognitive Maps in Modeling Systems. In: Proceeding of 5th IEE Mediterranean Conference on Control an System, pp. 518–527 (1997)
[18] Papageorgiou, E., Stylios, C., Groumpos, P.: An integrated two-level hierarchical decision making system based on fuzzy cognitive maps. IEEE Trans. Biomed. Eng. 50, 1326–1339 (2003)

[19] Papageorgiou, E., Parsopoulos, K., Groumpos, P., Vrahatis, M.: Fuzzy Cognitive Maps Learning through Swarm Intelligence. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 344–349. Springer, Heidelberg (2004)

[20] Papageorgiou, E., Stylios, C., Groumpos, P.: Active Hebbian Learning algorithm to train fuzzy cognitive maps. International Journal of Approximate Reasoning 37, 219–247 (2004)

[21] Papageorgiou, E., Spyridonos, P., Ravazoula, P., Stylios, C., Groumpos, P., Nikiforidis, G.: Grading urinary bladder tumors using unsupervised Hebbian algorithm for fuzzy cognitive maps. Biomed. Soft Comput. Hum. Sci. 9, 33–39 (2004)

[22] Stach, W., Kurgan, L., Pedrycz, W.: A Survey of Fuzzy Cognitive Map Learning Methods. In: Grzegorzewski, P., Krawczak, M., Zadrozny, S. (eds.) Issues in Soft Computing: Theory and Applications, Exit, pp. 71–84 (2005)

[23] Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Learning Fuzzy Cognitive Maps with Required Precision Using Genetic Algorithm Approach. Electronics Letters 40, 1519–1520 (2004)

[24] Aguilar, J., Cerrada, M., Mousalli, G., Rivas, F., Hidrobo, F.: A Multiagent Model for Intelligent Distributed Control Systems. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3681, pp. 191–197. Springer, Heidelberg (2005)

[25] Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Genetic Learning of Fuzzy Cognitive Maps. Fuzzy Sets and Systems 153, 371–401 (2005)

[26] Elpiniki, I., Papageorgioua, E., Stylios, C., Groumpos, P.: Unsupervised learning techniques for fine-tuning fuzzy cognitive. Int. J. Human-Computer Studies 64, 727–743 (2006)

[27] Cerrada, C., Aguilar, J., Cardillo, J., Faneite, R.: Agents-Based design for fault management systems in industrial processes. Computer in Industry 58, 313–328 (2007)

[28] Papageorgiou, E., Spyridonos, P., Glotsos, D., Stylios, C., Ravazoula, P., Nikiforidis, G., Groumpos, P.: Brain tumor characterization using the soft computing technique of fuzzy cognitive maps. Applied Soft Computing 8, 820–828 (2008)

[29] `http://www.damas.ift.ulaval.ca/~fabiola/recherche/`