2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Cultural Learning for Multi-Agent System and its Application to Fault Management

Juan Terán, José Aguilar, Mariela Cerrada

***Abstract*—It is usually agreed that a system capable of learning deserves to be called intelligent; and conversely, a system being considered as intelligent is, among other things, usually expected to be able to learn. Learning always has to do with the self-improvement of future behavior based on past experience. In this paper we present a learning model for Multi-Agent System, which aims to the optimization of coordination schemes through a collective learning process based on Cultural Algorithms.**

I. INTRODUCTION

A multi-agent system (MAS), a branch of distributed artificial intelligence, consists of agents community which interacts between them by using high level communication protocols and languages, to solve problems beyond his capabilities or knowledge [1]. The individuals (agents) of the MAS can learn trough their interactions. In the field of artificial intelligence, learning is usually a process through which a solitary agent acquires information about regularities in its environment, and then it uses that information to guide its behaviour. In a MAS, it is certainly possible to equip many agents with mechanisms allowing each one to learn individually. However, it has recently been argued that employing learning in the context of MASs may actually change the nature of the learning task, and make possible novel forms of learning [2]. Learning is, informally, the acquisition and incorporation of knowledge and skills by an agent, leading to an improvement in the agent's performance. Learning is necessary in MAS because many times the environment of the MAS is large, complex, open and time-varying [3]. Large and complex imply designing a good agent behavior that takes into consideration all the possible circumstances the agents might encounter, this is a very difficult, if not impossible, undertaking. The second two properties, openness and variation over time, imply that even if such a behavior were somehow designed, it would quickly become obsolete as the environment changes. There is a common agreement that there are two important reasons for studying learning in MASs: to be able to endow artificial MAS (e.g., systems of interacting autonomous robots, software agents) with the ability to automatically improve their behavior; and to get a better understanding of the learning processes in natural MAS (e.g., human groups or insect societies). In MASs two forms of learning can be distinguished [4]. First, centralized or isolated learning, i.e. learning that is done by a single agent on its own (e.g. motor activities). And second, distributed or collective learning, i.e. learning that is carried out by the agents as a group (e.g. by exchanging knowledge or by observing other agents).

J. Terán is with the Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID), Facultad de Ing., Universidad de Los Andes, Mérida-Venezuela, 5101 (carlostp@ula.ve).
J. Aguilar is with the Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID), Facultad de Ing., Universidad de Los Andes, Mérida-Venezuela, 5101 (aguilar@ula.ve).
M. Cerrada is with the Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID), Dpto. de Sistemas de Control, Universidad de Los Andes, Mérida-Venezuela, 5101 (cerradam@ula.ve).
This work was supported for the CDCHT project I-1237-10-02-AA of the Universidad de Los Andes.

An important issue in the MASs field is the learning processes for coordination. There are some works in the problem of learning in MAS´s coordination; in [5] the *ad hoc* coordination problem is studied; that is to design an autonomous agent which is able to achieve optimal flexibility and efficiency in a multi-agent system with no mechanisms for prior coordination. They conceptualise this problem formally using a game-theoretic model, called the Stochastic Bayesian Game, in which the behaviour of a player is determined by its private information. Based on this model, they derive a solution, called *Harsanyi-Bellman Ad hoc coordination* (HBA), which utilises the concept of Bayesian Nash equilibrium in a planning procedure to find optimal actions in the sense of Bellman optimal control. Other works that address the learning approach to coordinate MASs are [4, 6]. These works use one of the technics more used in multi-agent learning, i. e., the Reinforcement Learning (RL). In [6] they propose a Bayesian model for optimal exploration in multi-agent RL problems that allow the exploration costs to be weighed against their expected benefits using the notion of value of information. Unlike standard RL models, this model requires reasoning about how one's actions will influence the behavior of other agents. The estimated value of an action given current model estimates requires predicting how the actions will influence the future action choices of other agents. The value of information associated with an action includes the information it provides about other agents' strategies. For other side, [4] deals with learning in reactive MAS. The central problem addressed is how several agents can collectively learn to coordinate their actions such that they solve a given environmental task together. In approaching this problem, two important constraints have to be taken into consideration: the incompatibility constraint, that is, the fact that different actions may be mutually exclusive; and the local information constraint, that is a fraction of its environment. Here, two algorithms called ACE and AGE (standing for "ACtion Estimation" and "Action Group Estimation", respectively) for the reinforcement learning appropriate sequences of action sets in MAS are described.

Now, the previous works has the same north: 'the coordination in MAS' based on a planning procedures-oriented approach, to ensure the achievement of the objectives. This means addressing the internal behavior of the MAS to achieve learning. That is, they attack the problem of coordination based on internal actions and behaviors of MAS, in order to obtain models of agents, and achieve in this way collective learning.

Our model aims to coordination in MAS, searching the optimization of coordination schemes, but from another point of view, namely, an external approach of the MAS, by searching the communication protocols to be used by a conversation, through a collective learning processes based on Cultural Algorithms (CA) [7]. That communication protocols are obtained from the common knowledge space provided by the CA. In particular, our model uses the following coordination mechanisms (CM) standardized by the Foundation for Intelligent Physical Agents (FIPA): English Auction (SI), Dutch Auction (SH), the contract net (tender, L), and planning (PL). So, the MAS collectively achieve to discern which CM is the most suited for a conversation, by using that space of knowledge exchange. The model optimizes the part of coordination of MAS related to the costs of processing and communication generated by the use of a particularly CM. In this paper the interactions between agents are viewed as conversations, which in turn may have sub-conversations. To characterize these conversations and sub- conversations we have defined four "types of conversations" (TCs), TC1: Consult, TC2: Assign, TC3: Inform and TC4: Request, they are based on the FIPA communicative acts and defines the interactions patterns [8]. These patterns allow generalizing interactions or conversations between agents of any community. Here, a learning model is proposed to optimize coordination schemes in MAS, based on CA as an important tool for the learning process. The CA can provide knowledge, since one of its main components is a common space of experiences, thus providing the capacity for collective learning based on knowledge sharing. The paper is organized as follows; section II discusses the theoretical framework in which the model are based. Section III presents the approach: the proposed formal model of learning coordination schemes for MAS. Section IV presents the experiments with CLEMAS (a simulation tool, which means Cultural LEarning for Multi-Agent Systems) and the application of the model to a case of study and the results; finally, section V presents the conclusions. The case study is in the field of industrial automation, which consists in a MAS-based Fault Management System (MAS-FMS).

II. Theoretical Framework

A. Coordination on MAS

The problem of coordination arises in MASs due to the distributed nature of the control exercised by the agents. Coordination is defined by [3] as the process by which the individual decisions of the agents result in good overall decisions for the group. The problem is more stringent in cooperative multi-agent systems, but also appears when the agents are self-interested. We can describe coordination in MASs as the set of complementary activities necessary to be performed in a community of agents to act collectively [1]. In MASs, coordination can be seen as a process in which agents involved engage in order to ensure their community acts in a coherent manner. Coherence refers to how well a system agent behaves as a unit. There are several reasons why agents need to be coordinated [9]:

--Preventing anarchy or chaos: coordination is necessary or desirable because, with the decentralization in agent-based systems, anarchy can set in easily. No longer does any agent possess a global view of the entire agency to which it belongs. This is simply not feasible in any community of reasonable complexity.

--Meeting global constraints: there usually exist global constraints which a group of agents must satisfy if they are to be deemed successful. Agents need to co-ordinate their behavior if they are to meet such global constraints.

--Distributed expertise, resources or information: agents may have different capabilities and specialized knowledge. Alternatively, they may have different sources of information, resources (e.g. processing power, memory), reliability levels, responsibilities, limitations, charges for services, etc. In such scenarios, agents have to be coordinated in just the same way.

Let's now present some of the most representative coordination mechanisms finding on the literature [10].

1) Market Protocols: The most well known market structures take the form of auction houses [10] in which the type of auction indicates a prescribed guide of how the bids are treated. For example, the bids could be open and known to all the participants or they could be sealed and none of the bidders know about the others' proposals. Alternative types of auction are when the bidders must follow a defined pattern; i.e., bids should raise the current price until one bid is the winner (when dealing with ascending price auctions) or when bidders are only allowed to decrease because the process starts with a high price (descending price auction). These protocols specify the rules the agents have to follow in order to propose, to deliver and to take decisions about the proposals. However, although the protocol specifies the rules of how agents can bid, it is clear that agents still have to take their preferences into account to decide how to actually bid. Moreover, if an agent is aware of the offers of the other agents it might use a different strategy than if it does not know about their bids.

2) The contract Net protocol (Tender): The Contract net, designed by Randall Davis and Reid Smith [11], are mechanisms used for collaborative problems solving. In this case, the goal is that several agents communicate and coordinate with each other, so they can carry out a task whose complexity makes it difficult to be performed by a single agent. Each agent in the network takes one or two roles related to the execution of an individual task: manager or contractor. When an agent has the manager role must carry out certain activities, which are: *(i)* break a complex task into less complex subtasks, and so it will be easier to solve, *(ii)* announce to other agents that there is a sub-task expecting to be executed. These ads are distributed through a broadcast message to all agents or are directed to a specific group of agents, *(iii)* when it receives the response to their requests, the manager selects the most appropriate offer and assign this sub-task to this agent, for which a contract is created, *(iv)* monitor the progress of the contract, possibly asking for information, reports, etc. It is free to reassign the subtask if the contractor fails to complete, finally, *(v)* integrate the partial results produced by the contractors in a complete solution. Meanwhile, the contractors agents perform the following activities: *(i)* receive announcements of tasks and assess their skills and availability to perform them, *(ii)* if they are able to satisfactorily perform the task, make an offer, and *(iii)* if this offer is accepted they reserve the resources required for its execution. When the contractors have won bids, they must perform the tasks assigned to them, and generate reports about the progress of these tasks and their final results. The contractor agents can become managers if the subtask is too complex for their abilities, then they subdivide the sub-task and the process of assignation of task is repeated. In summary, the protocol uses a number of basic messages:

--Notification of the task, sent by the manager agent to announce the availability of a task and the need for contractors

--Offer, sent by the contractor agents when they are available to perform a task

--Winner, sent by the manager agent to the contractor, stating that he is now responsible for the task

--Report, submitted by the contractor agent to the administrator/manager agent stating whether or not successfully it executes the task and the results (also, it sends partial results).

These messages provide the functionality needed to make the contracts between agents [11].

3) Multi-Agent Planning: The intuitive concept of multi-agent planning involves the agents agreeing about the order in which their actions are executed in order to obtain a coordinated global plan (joint plan) [10]. Such planning involves two phases: building the plan (design phase) and executing it (execution phase). The design phase's objective consists of trying to obtain a joint plan where the actions of all the agents are scheduled and the conflicts that might cause harmful interactions have been removed. The challenge of this stage is to reconcile the various choices raised to find the best sequence of actions about which all the participants agree. This agreement not only covers the order and time at which the actions take place, but also the resources assigned to each action. The aim of the latter phase is to execute the actions of the joint plan.

In the multi-agent planning literature there are two main approaches to constructing and/or executing a plan, the centralised and the distributed solution. The differences between them relate to whether the responsibility for constructing and managing the execution of the plan lies with a single agent or with multiple agents. This responsibility involves things like maintaining the coherency of the plan, solving problems as they arise, and giving priorities to the various actions. In the centralised case, one agent has the global vision or knowledge of the problem and is capable of solving any discrepancies that might appear when constructing or executing the plan. In the distributed case, however, several agents participate in coordinating and deciding upon their actions, about avoiding conflicts when executing the plan, and helping each other to achieve the plan.

B. Learning to Select Mechanisms of Coordination

To date there has been comparatively little work concerned with learning which CM to select in a given context. However, there are two systems in which such learning is exploited; namely, COLLAGE and LODES [10]. The objective in both systems is to improve coordination by learning to select a coordination strategy in appropriate situations. However the aspects each system addresses are different and their findings are complementary. LODES is more interested in having agents that are capable of learning the key information that is necessary to improve coordination in specific situations. In COLLAGE agents learn how to choose the most appropriate coordination strategy given a particular situation. Thus, LODES focuses on "what information to learn" and COLLAGE on "learning the situation where to use a coordination strategy". It is important to notice that both systems are concerned with the detailed activities of coordination as part of the learning process. For agents to solve a particular coordination problem, they have to solve all the interrelations and dependencies between their actions. Thus agents first plan the actions to perform and then execute them. To solve this, both systems have to handle explicit knowledge about the domain in the case of LODES and about coordination strategies in the case of COLLAGE.

III. Learning Model Of Coordination Schemes In MAS

The learning model consist in to define and specify each component of a CA on the basis of our proposed objective, which is learn coordination protocols for possible conversations of a MAS. These components are: a population space, a belief space, and a protocol (acceptance and influence functions) that describes how knowledge is exchanged between the two spaces. In addition to CA, a mathematical formalization of CM carried out in previous work is used [8, 12, 17]. The basic pseudocode of our learning model is shown (Fig. 1).

```
Begin
  t = 0;
  Initialize population P(t);
  Initialize Coordinating Mechanisms MC(t);
  Initialize Belief Space BS(t);
   Repeat
      Evaluate population P(t) with the objective function;
      Adjust BS(t) with the objective function of acceptance P(t)
      Variation of P(t) from P(t-1) using the influence function BS(t) and
the genetics operators
   Until number of generations reached;
End
```

Fig. 1. Basic Pseudocode for Learning Model Based on CA

A. The population

The population, as in any method of Evolutionary Computation (EC) is formed by individuals. In our model, each individual is a MAS composed of *n* different conversations involved in the community of agents (that is an instantiation of the MAS using in each case different CM). Remember that every conversation, in turn, has sub-conversations eventually. All of them are characterized by the TCs previously defined [8].

The individuals in our populations has the structure of Fig. 2, where C_i denotes the conversation i existing in the MAS, FO is the value of the objective function of individual, $C_{i,k}$ denotes the sub-conversation k of the conversation i, being m_i the number of associated sub-conversations to the conversation, TCs is the type of conversation, $MC_{i,k}$ is the CM used, and $P_{i,k}^{u}$ are the u parameters of those CM. When a conversation have not sub-conversations, k = 1.

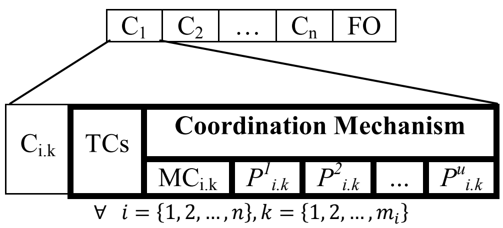

Fig. 2. Internal Structure of an Individual

The highlighted part of the individual represents its knowledge or experience. In order to describing a little more the Fig. 2, the following example assumes: a MAS with three conversations (C_1, C_2, C_3), where C_1 has two sub-conversations ($C_{1.1}$, $C_{1.2}$), C_2 has two sub-conversations ($C_{2.1}$, $C_{2.2}$) and C_3 has three sub-conversations ($C_{3.1}$, $C_{3.2}$, $C_{3.3}$). Besides it is assumed that the type of conversation of the sub-conversation $C_{2.1}$= is TC2: assign and; for that TC the individual uses the CM English Auction (SI). In previous works have already been parameterized this CM, [8, 12, 17]; so, for the case of SI its parameters are C_0, is the initial price of the auction, ε_I, which represents the maximum amount that each bidder agent i can bid, and $C_P(j)$ (it is the condition for stopping the auction, where j is the number of rounds).

In the Fig. 3 is shown this explanatory example doing a specific zoom for $C_{2.1}$. This figure also represents the gene of the individual.

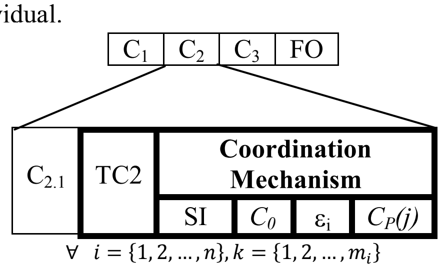

Fig. 3. Example of a Gene on Individual, Zoom in C_2

Well now, for the reproduction of the population are used two genetic operators, mutation and crossover. An example of crossover between these individuals is shown below. For example assume the individual of Fig. 3. We further assume that each conversation (C1, C2, C3) has only one sub-conversation (that is the conversation itself), namely, that individual shall be composed of three TCs. Now, assume that each of these TCs deserves to be treated by a CM. For mutation and crossover, the CMs represent individual genes, that is, each CM has implicitly a sub-conversation, a TC, and a set of parameters. The following figure illustrates this example.

In Fig. 4, part (a) are the parents, and part (b) the offspring. In the individual parent to the left, the first CM used is 'SI' for the TC of C1, 'L' for the TC of C2, and 'SH' for the TC of C3, the analyze is similar in the case of the another individual parent. The one-point crossover is used (indicated by the arrows), which is applied at the level of the CM only. We see as two new children are generated.

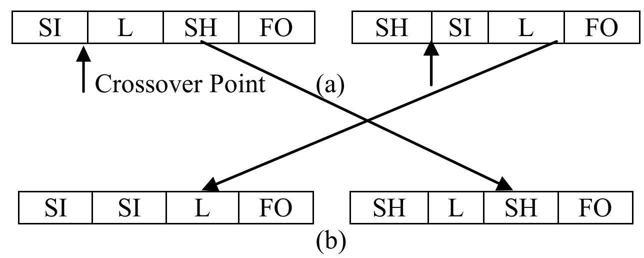

Fig. 4. Crossover Operador

The mutation is simpler; it takes a CM or more from the conversations of a parent, and changes it randomly by other CM.

B. Objective Function

The objective function evaluates the performance of each individual. This function is based on the processing cost (CP) and communication cost (CC) of each CM used by the individual, see equation (1). There, 'a' and 'b' are constants defined by the user to weigh the importance of the communication part with respect to the processing part. n is the number of conversations, m_i is the number of sub-conversations in a conversation i, $CP_{i,k}$, is the cost of processing of the CM used in the sub-conversation k and conversation i. $CC_{i,k}$ is the cost of the communication. For our case, the best individual will be the one that minimizes objective function [12].

$$FO = \sum_{i=1}^{n}\sum_{k=1}^{m_i}\left(a * CP_{i,k} + b * CC_{i,k}\right) \quad (1)$$

The cost of processing $CP_{i,k}$ is given by equation (2), and its units are based on the average execution time:

$$CP_{i,k} = PI_k + PE_k + \sum_{l=1}^{j}\sum_{q=1}^{n_j} A_{l,q} \quad (2)$$

This cost depends of the actors involved, and processing algorithms. Up for auction (english or dutch), PI_k is the initial price setting and start of auction of the sub-conversation k, PE_K is the process of selecting the winning agent, j the number of rounds, n_j the number of bidders for round, and $A_{l,q}$ is the time to prepare proposal for auction of participating agents. For tender PI_k is the specification of conditions (initial) in which a service is required, PE_K is the process of selecting the servicing agent, j is equal to a one round (1), n_j number of bidders, and $A_{l,q}$ is the time to prepare proposal for bidding agents. For both CM, PI, PE and A according to Table I, are parameters measured qualitatively (e. g., low, medium and high).

TABLE I
QUALITATIVELY VALUES OF THE PARAMETERS *PI, PE* AND *A*

MC	*PI*	*PE*	*A*
Tender (L)	Medium	Medium	Medium
English and Dutch Auction (SI, SH)	Low	Medium	Low

To measure these parameters with numbers (so equations require) consider as a reference the Likert scale [13], which allows to assign numerical values to such parameters. Thus, we assign the value 0.2 to low 0.6 to medium and 1 to high. The other parameters are quantifiable. The cost of communication is based on the estimated time for messages exchange (communicative acts) used in the CM in each conversation, its equation is as follow:

$$CC_{i,k} = \sum_{l=1}^{j}\left(\sum_{r=1}^{N-1} CEP_{l,r} + \sum_{s=1}^{n_j} CEO_{l,s}\right) + \sum_{r=1}^{N-1} CS_r \quad (3)$$

Where *j* is the number of rounds, *N-1* the number of agents least the sender of message and n_j is the number of participants in each round. For auction and tender, *CEP* is the sending cost of the initial proposal, *CEO* is the sending cost of bids, and *CS* is the cost of informing the winner. Table II shows the qualitative values for the parameters *CEP*, *CEO*, *CS*.

TABLE II
PARAMETERS *CEP*, *CEO*, *CS*

MC	*CEP*	*CEO*	*CS*
Tender (L)	Low	Medium	Low
English and Dutch Auction (SI, SH)	Low	Medium	Low

C. Belief Space

There are two categories of knowledge in belief space: situational and normative.

1) Situational Knowledge: In situational knowledge is kept information over good or bad individuals. In our model, it is based on each TC, for which is included each CM used in this TC, their rate of occurrence (IO), and finally, the total occurrences (TO) of the TC (Fig. 5).

TCs	MCi	$IO_{(TCs,\ MCi,\ t-1)}$	$TO_{(TCs)}$

$\forall \quad s = \{1, 2, 3{,}4\}, i = \{1, 2, \ldots, n\}$

Fig. 5. Situational Knowledge

2) Normative Knowledge: The normative knowledge defines the ranges suitable for each of the variables of the CMs. In Fig. 6, LI and LS are the lower and upper limits of each variable (parameter P^i) forming each CM.

MCi	P^1		P^2		…	P^u	
	LI	LS	LI	LS		LI	LS

$\forall \ i = \{1, 2, \ldots, n\}$

Fig. 6. Normative Knowledge

D. Communication Protocol

Functions of acceptance and influence are those that allow the interaction between the population space and the belief space. These functions in this proposal are:

1) Acceptance Function for the Situational Knowledge: This function takes a percentage of the population (20% of individuals is sufficient according to Reynolds [7]), in order to nurture the belief space with their experiences. The acceptance function updates the situational knowledge as follows: for a specific type of conversation each of the mechanisms involved in this type of conversation is updated by (4).

$$IO_{(TCs,MCi,t)} = IO_{(TCs,MCi,t-1)} + \left(NO_{(TCs,MCi,t)}/TO_{(TCs,t)}\right) \quad (4)$$

$$\forall \quad s = \{1, 2, 3{,}4\}, i = \{1, 2, \ldots, n\}$$

Where $IO_{(TCs,\ MCi,\ t)}$ is the rate of occurrence in the iteration (*t*) for TCs and MC_i. $IO_{(TCs,\ MCi,\ t-1)}$ is the rate of occurrence in the iteration (*t-1*) which is currently in the belief space, $TO_{(TCs)}$ is the total occurrences of MC_i for TCs, and $NO_{(TCs,\ MCi,\ t)}$ is the number of occurrences in the current instantiation of the MAS for each MCi for this TC. It is also necessary to update the total occurrences TO, for that the following equation is used, where *k* is the number of CM used in this TCs:

$$TO_{(TCs,t)} = TO_{(TCs,t-1)} + \sum_{i=1}^{k} NO_{(TCs,MCi,t)_i} \quad (5)$$

$$\forall \quad s = \{1, 2, 3{,}4\}, i = \{1, 2, \ldots, n\}$$

2) Acceptance Function for the Normative Knowledge: The acceptance function updates the normative knowledge by the following equation:

$$Lac(P^u) = \ [(lv * \bar{m} + \bar{P} * m)/2] \quad (6)$$

Where, $Lac(P^u)$ is the current limit (either LI or LS), Lv is the previous limit, $\bar{m}$ is the complement of the moment, namely, (1 - m), $\bar{P}$ is the average value of the limit of all individuals accepted within 20% from the population. Finally, *m* is the moment that is given by the equation:

$$m = \mu / t \quad (7)$$

Where μ is a time constant between 0 and 1, and *t* is the iteration number, (*t = 1, 2, 3 …)*. Thus, each time it reaches a new experience of the people, the limits of each parameter of the mechanism are updated.

3) Influence Function: The influence function determines how the knowledge of the system influences over the individuals in the population. In the case of situational knowledge, is based on the use of the mutation operator, which switches the current CM of a given conversation, according to a probabilistic rule (stochastic universal sampling or roulette wheel [14]) based on the IO parameter of each TC (we call that a targeted mutation).

In the case of normative knowledge is also based on the mutation operator, only that here the complete structure is not varied, but only specific values of the ranges are altered for each variable of a specific CM.

IV. Experiments

A. Study Platform

This section present a tool (CLEMAS) to implement the learning model of coordination schemes for a given MAS. Besides it is presented a case of study oriented to the integration in automation agent-based.

1) CLEMAS (Cultural Learning for Multi-Agent Systems): This tool has four main components: the execution engine, a system that emulates the CA, a graphical interface for configuring the system initially and visualizes the learning process with their results, and a database that stores the existing prior knowledge in the belief space. The execution engine component runs the learning process through a class called 'simulation', using for this the initial system configuration. The CA component represents individuals and the belief space, the knowledge base component stores the situational and normative knowledge in the belief space (it is saved with the extension .ccg, for others futures executions).

2) Case of Study: Fault Management System MAS-based: The case study of this paper corresponds to a MAS for handling faults in industrial processes, whose specification is described in detail in [15] using the MASINA specification methodology proposed in [16]. The Fault Management Systems (FMS) is composed of two modules, the first performs the monitoring and failure analysis, and the second performs the tasks of the maintenance management system. The FMS interacts with the Maintenance Engineering and the Fault Tolerant process. The Monitoring and Failure Analysis module includes the fault detection and diagnosis; the Maintenance Tasks module includes the following tasks: Prediction of the occurrence of a functional failure, Planning of preventive maintenance, and execution of maintenance. The FMS is a subsystem of level of supervision of an automated system. Thus, the FMS can be seen as a system composed of intelligent agents that cooperate to solve problems related to the handling of system failures. Furthermore, some activities of the FMS follow a distributed computing model, such as those performed for the fault detection in equipment or processes, the performance index estimation, among others. To illustrate the application of the proposed CA-based learning for the coordination of the MAS, two specific conversations of MAS for handling faults described above are taken. Before that, the coordination model is presented in a general manner for such MAS. Then it specify in detail the coordination schemes chosen under the formalism proposed in this paper, and an execution of CLEMAS is shown.

Coordination Model: The MAS has six conversations that are [15]: On-condition maintenance (C1), maintenance tasks (C2), urgent tasks (C3), replanning of tasks (C4), state of maintenance (C5), and identify functional failure (C6). Of these six conversations only two will detail, the rest can be found at [15].

--Conversation 4 (C4): Replanning of Tasks, this conversation is made up of three sub-conversations: C4.1 of type TC1, C4.2 of type TC3 and C4.3 of type TC4.

Description: Through this conversation, the coordinator agent seeks information from the database agent to reschedule outstanding maintenance tasks on the system, and make a new maintenance plan. If the task is urgent and cannot reschedule an alarm is given.

--Conversation 5 (C5): State of Maintenance, this conversation is made up of four sub-conversations: the C5.1of type TC1, C5.2 of type TC1, C5.3 of type TC3 and C5.4 of type TC4.

Description: Through this conversation, the observer agent seeks information from the database and the actuator agent agent to store outstanding maintenance tasks on the system.

To show the characterization of TCs in the sub-conversations in every conversation, we present as an example the interaction diagram of conversation state of maintenance (C5), identifying the TCs in boxes (Fig. 7). In that conversation, the observer agent (AO) consults (TC1) the database agent (ABD) twice (process information and maintenance information), reports (inform, TC3) to the actuator agent (AA) maintenance tasks, and those do not made, and requests (TC4) to ABD to incorporate information.

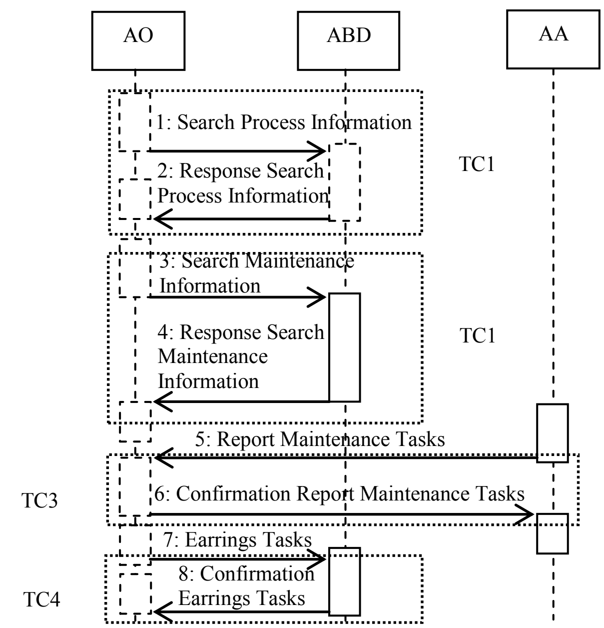

Fig. 7. Conversation with its TCs

B. Design of the Experiments

For the case study we characterized the previous two conversations in two different scenarios.

Scenario One: For this scenario, we assume that for C4 must be optimized only the sub-conversation (C4.1) which has 5 agents (4 database agents and 1 coordinator agent), and for C5 must be optimized the sub-conversation (C5.3) with 4 agents (1 observer agent and 3 actuators agents). The objective of the simulation is to show how it influences the number of generations (iterations) for the learning process. To achieve this, CLEMAS is configured initially with a low number of generations. The maximum number of auctions rounds is 5 and for tender 1. The initial values of the parameters of the auctions mechanisms (English and dutch) for TC1 are: C_0 = [5 ... 15] ε_i = [5 ... 20], $C_P(j)$ = [1 ... 5]. For TC3 the initial values of tender are $M(f)$ = [1 ... 3] that is the expiration date for offering [17], and *f(T)* = [5 ... 20] that is a function that allows potential contractors to assess their capacities to respond to the notice of request for the performance of the task [17]. For this simulation the population size is 20 individuals, 35 generations (35 iterations), crossover probability of 0.7 and mutation of 0.5.

Scenario Two: In this scenario the same sub-conversations are assumed to optimize, and the same CM are used. Additionally, the number of agents is the same. Here, we increase the number of generations to 50 to see if individuals actually improve their behavior. The values of the initial parameters remain the same in this scenario, population and genetic probabilities.

(1) Results of Simulations: In the following, the results provided by CLEMAS are presented.

For Scenario One:

TABLE III
RESULTS (PERCENTAGE OF USE OF EACH CM IN EACH TC, BASED ON 20% OF THE POPULATION)

TC	Tender	English Auction	Dutch Auction
Consult (TC1)	96.42%	1.42%	2.14%
Assign (TC2)	0.0%	0.0%	0.0%
Inform (TC3)	2.85%	35.0%	62.14%
Request (TC4)	0.0%	0.0%	0.0%
Total of Occurrences	49.64%	18.21%	32.14%

The next figure, shows the evolution of the objective function.

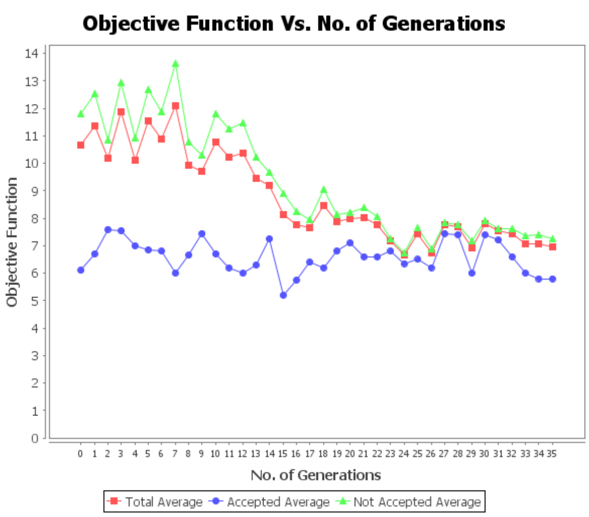

Fig. 8. Evolution of the objective function for the Scenario One.

For Scenario Two:

TABLE IV
RESULTS (PERCENTAGE OF USE OF EACH CM IN EACH TC BASED ON 20% OF THE POPULATION)

TC	Tender	English Auction	Dutch Auction
Consult (TC1)	55.20%	2.60%	42.18%
Assign (TC2)	0.0%	0.0%	0.0%
Inform (TC3)	1.04%	60.93%	38.02%
Request (TC4)	0.0%	0.0%	0.0%
Total of Occurrences	28.12%	31.77%	40.10%

2) Analysis of results: In both tables (III and IV) the sum of the values of each mechanism are the 100%. Now, in the table III, we see that for scenario one, the TCs being optimized were consult (TC1) and inform (TC3), which are precisely the TCs using C4.1 and C5.3 respectively. The table also shows for this scenario that tender has prevailed with 49.64% with respect to other mechanisms (total occurrences). This is because the TC1 uses 96.42% this mechanism, whereas TC3 use 62.14% Dutch auction. The English auction only is used 35.0% for the TC3. Fig. 8 shows the evolution of the objective function through the generations. In Fig. 8 the red curve (squares) represents the average of the objective function of the entire population. The blue curve (circles) represents the objective function of 20% of the population (selected for the individual acceptance function, it represents the *desired behavior*).

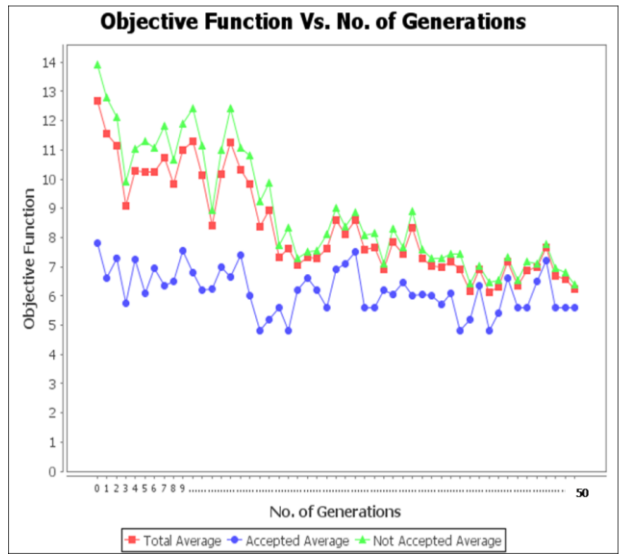

Fig. 9. Evolution of the objective function for the Scenario two.

Finally the green curve (triangles) represents the average of the remaining 80% of the population. This graphic could be seen as the learning of individuals because its reflects how the individual reduce their costs (FO) through the use of appropriate CM. Thus we see how the curves tend to follow that of the best individuals (circles) even though the number of generations is quite small.

For scenario two, Table IV shows that the dominant mechanism was the tender with 55.20% for TC1, because it is less expensive for this TC. Table shows that for this scenario all mechanisms were used. The English auction is most used by TC3 in a 60.93% and a 2.60% by TC1. Finally Dutch auction is used for TC1 in a 42.18% and in a 38.02% for TC3. As is observed, the tender mechanism for TC1, and the auction mechanism for TC3, are the bests. Fig. 9 shows how when is increased the number of generations, the curves tend to drop, reducing its objective function (FO) to the range of 5.0 and 8.0. Additionally, the curves tend to converge to similar values. That is due to that when is increased the number of generations individuals acquire more knowledge.

As a final result analysis, for both scenarios, despite TC1 start with auction and TC3 with tender (section IV, part B), it appears that individuals choose tender for TC1 and auction for TC3, possibly due to the number of agents in each sub-conversation. The value of the objective function is significantly improved in scenario two by increasing the number of generations, that means, they have more experiences to learn.

3) Qualitative comparison on coordination learning strategies: In this section we will make a qualitative comparison of related-learning works in MAS, in order to highlight the contribution that our job offers. Previous works related to learning in multi-agent systems are limited, much of these works relies on techniques derived from RL [4, 6], and more few in learning which CM to select in a given context [10]. While these approaches address the coordination problem based-learning, this is accomplished without any interaction between agents. Theirs algorithms and approach are based on actions, states and rewards (own of the RL). These approaches require that each agent has a model of the strategies of others (states, actions and rewards). On the other hand, the approach of [10] is very close to ours, but their difference is that the MAS of this work consists of complex cooperative agents (each agent is a sophisticated problem solver), and each local agent interacts with the other agents' in intricate ways. Our approach is simpler, since we do not explore the inside of the agents (like in RL). Our approach does not require sophisticated agents and do not depend on the kind of application, or what specifically makes the MAS. The determining factor in our approach is its Structural Characterization (how agents interact with each other, what types of conversations uses, which is the cost of processing and communication in achieve interaction protocols, etc.), i. e., it is transparent to the goal of the MAS. This makes it is more universal, because can be applied to any MAS.

V. Conclusion

CA are presented as a powerful learning tool for individuals in different societies. In this work, it has been used to learning how to coordinate a MAS. A learning model of coordination schemes for communities of agents using CA is proposed. The cultural model is systematized in the CLEMAS platform, which allows interactively present different scenarios in the case study, and graphically displays the results of these scenarios.

One of the main advantages of the system is its simplicity and flexibility to adapt to any scenario, allowing the test of issues like scalability in the agent community. In addition, all the accumulated knowledge in the belief space can be reused by the system to optimize the CM of other MAS, that is, that knowledge can be reused continuously for design CM of MAS. In summary, we have presented a Cultural Learning System for coordination schemes for MAS, and the same has been applied to a case study, a fault handler system based on MAS. CLEMAS is presented as a useful tool for collective learning in communities of agents and can handle different types of knowledge (in our case, situational and normative).

Upcoming work will carry out a more thorough study of the different parameters that can be considered in a learning process of coordination mechanisms of MAS (number of agents, number of communications, etc.), as well as the suitable values of CLEMAS (number of generations, probabilities, etc.)

References

[1] J. Aguilar, A. Rios Bolivar, F. Hidroboro, and M. Cerrada, "Sistemas multiagentes y sus aplicaciones en automatización industrial," First edition 2012, ISBN: 978-980-12-5942-8, Universidad de Los Andes, Facultad de Ingeniería, Mérida-Venezuela.

[2] J. Noble and D. W. Franks, "Social Learning in a Multi-Agent Systems," Computing and Informatics, Vol. 22, pp. 1001–1015, Apr. 2004.

[3] L. Busoniu, B. De Schutter, and R. Babuˇska "Learning and Coordination in Dynamic Multiagent Systems," Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, Tech. Rep. 05-019, October 2005.

[4] G. Weiß. Lernen und Aktionskoordinierung in Mehragentensystemen. In J. Müller (Ed.), Verteilte Künstliche Intelligenz – Methoden und Anwendungen (pp. 122–132). BI Verlag. 1993.

[5] S.V. Albrecht, S. Ramamoorthy. A Game-theoretic Model and Best-response Learning Method for Ad Hoc Coordination in Multiagent Systems. *Technical Report, School of Informatics, The University of Edinburgh*, United Kingdom, February 2013.

[6] G. Chalkiadakis, C. Boutilier, "Coordination in Multiagent Reinforcement Learning: A Bayesian Approach," *AAMAS'03*, Melbourne, Australia, July 14-18, 2003.

[7] R. Reynolds, "Cultural Algorithms: Theory and Applications", New Ideas in Optimization. David Corne, Marco Dorigo and Fred Glover, Editors. Chapter Twenty-Four; pp.367–377, 1999.

[8] J. C. Terán, J. L. Aguilar, and M. Cerrada, "Modelo cultural para la coordinación de sistemas multi-agente" Tech. Rep. Universidad de Los Andes, Venezuela. 2013.

[9] H. Nwana, L. Lee and N. Jennings, "Co-ordination in software agent systems". *BT Technol J*, Vol. 14, no. 4, October 1996.

[10] C. B. Excelente Toledo, "The Dynamic Selection of Coordination Mechanisms in Multi-Agent Systems (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. and Comp. Sci., Univ. of Southampton, February 2003.

[11] R. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactionson on Computers*, Vol. C-29, no. 12, December 1980.

[12] J. Terán, J. L. Aguilar and M. Cerrada, "*Cultural Algorithms-Based Learning Model for Multi-Agent Systems* (Presented Conference Paper style)," presented at the XXXIX Latin American Computing Conference (CLEI 2013), Vargas, Venezuela, October 7–13, 2013.

[13] D. Bertram, "Likert Scales", CPSC 681 – Topic Report. http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf

[14] M. Mitchell, "An introduction to genetic algorithms", Eighth printing, 2002, first MIT press paperback edition, Massachusetts Institute of Technology, 1998.

[15] M. Cerrada, J. Aguilar, J. Cardillo, R. Faneite, *Agents-based design for fault management systems in industrial processes*, *Journal Computers in Industry*, Vol. 58, no. 4, pp. 313-328, 2007.

[16] J. L. Aguilar, M. Cerrada, G. Mousalli, F. Rivas, F. Hidrobo, "A Multiagent Model for Intelligent Distributed Control Systems", Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 3681, pp. 191-197, 2005.

[17] J. Terán, J. L. Aguilar, and M. Cerrada, "Mathematical Models of Coordination Mechanisms in Multi-Agent Systems," CLEI Elec. Journal, Vol. 16, Nro. 2, August 2013.