

Multi-Agent System for Real Time Planning Using Collaborative Agents

Ana Lilia Laureano-Cruces^{1*}, Tzitziki Ramírez-González², Lourdes Sánchez-Guerrero¹,
Javier Ramírez-Rodríguez¹

¹Departamento de Sistemas, Universidad Autónoma Metropolitana—Azcapotzalco, México D.F., México

²Maestría en Ciencias de la Computación, Universidad Autónoma Metropolitana—Azcapotzalco, México D.F., México

Email: *clc@azc.uam.mx, tzitzikig@hotmail.com, lsg@azc.uam.mx, jararo@azc.uam.mx

Received 15 August 2014; revised 17 September 2014; accepted 1 October 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Autonomous agents are an important area of research in the sense that they are proactive, and include: goal-directed and communication capabilities. Furthermore each goals of the agent are constantly changing in a dynamic environment. Part of the challenge is to automate the process corresponding to each agent in order that they find their own objectives. Agents do not have to work individually, but can work with others and develop a coordinated group of actions. These agents are highly appreciated, when real time problems are involved, meaning that an agent must be able to react within a specific time interval, considering external events. Our work focuses on the design of a multi-agent architecture consisting of autonomous agents capable of acting through a goal-directed with: a) constraints, b) real-time, and c) with incomplete knowledge of the environment. This paper shows a model of collaborative agents architecture that share a common knowledge source, allowing knowledge of the environment; where we analyze it and its changes, choosing the most promising way for achieving the goals of the agent, in order to keep the whole system working, even if a fault occurs.

Keywords

Multi-Agent Systems, Blackboard Architecture, Planning, Schedule, Collaborative Systems, Cognitive Systems

*Corresponding author.

1. Introduction

Each time we generate an *artificial intelligent model*, we are inspired by cognitive sciences. These allow us to understand how humans emulate the behavior of cognitive systems [1] [2]. Cognitive systems include both natural and artificial information processing systems responsible for perception, learning, reasoning, communication, action and more. One of the goals is to understand how natural and artificial processing devices could solve the most complex systems [1] [2]. In the case of the brain the robustness is completed by an ultra massive parallel processing in neural systems that can be compared to continue developing of very large scale integration. In order to get such complex computer systems it is necessary to think in the integration of these issues: (1) dynamics: how simple elements provide complex behavior over time; (2) self-organization: how an ultramassive parallel processing system organize themselves in a particular task without external intervention; (3) synchronization: how many processors operate without a signal (clock); (4) processing speed: the brain is fast even with slow processors; (5) power use: the power used by the brain to many processors; (6) timing: related to the ability of the brain to ensure everything happen on time; (7) robustness: the brain can tolerate fails in their components and arrive to his goal; (8) information representation in transmission: the brain communicates information between components in different ways; (9) construction and micro circuitry: the brain is organized in a complex hierarchy. This work is an approximation to the robust problem of simulating a human behavior, and is related with the (1), (2), (3), (6), and (7) issues that try to resolve a complex communication between processes. To achieve this, it is proposed to use reactive agents and their ability to build multi-agent systems. The agents cover a variety of topics including: a) *distributed problem solving*: the agent concept can be used to simplify the solution of large problems by distributing them to a number of collaborating problem-solving units; b) *multi-agent systems (MAS)*: this research deals with appropriate ways of organizing agents. These include general organizational concepts, the distribution of management tasks, dynamic organizational changes like team formation and underlying communication mechanisms; c) *autonomous agents*: research on autonomous agents is primarily concerned with the realization of a single agent. This includes topics like sensing, models of emotion, motivation, personality, action selection, and planning; d) *a reactive behavior*: an agent that follows another or reacts to his movements is perceived as an object that is driven by internal intentions and objectives. Based on the above idea, we define an agent as a computer system situated in some environment and capable of flexible autonomous action in this environment in order to achieve the objectives for which it was designed [3]. There are three questions related to the design of agents: 1) *How can an agent determine the direction of your goals?* The answer is our cognitive model that allows a view of the work as a whole, which enables us to view a model of collaborative agents. Each one independently carries out its work [1]. The abstraction in analysis and design is an important tool. Last make possible an appropriate level of description of the behavior, 2) *How can an agent plan a series of actions to achieve these objectives and address?* The interaction among agents are not pre-defined, there are not a global system goal. The agent has his own goals and capabilities. So in MAS the agents need to coordinate their activities and cooperate with each other. In the study case the planning is articulated in the design of behavior. Last is possible taking into account alternative behaviors that will be immersed inside the design of mental model. This is achieved by providing dynamic behaviors based in the stimuli presented in the dynamic environment. In this case using a shared memory, denominated *blackboard*, 3) *How can an agent interact with other human and software agents to achieve their objectives?* In the theory of MAS research there is a sub-line that refers to the different forms of communication that can be used in order to carry out a particular task. One way is through the changes that occur in the environment [4]. Each agent has its motivations and goals; each of them has different levels of competition. When presented in the dynamic environment stimuli that make a match with their responsibilities, the first make that trigger actions associated with different competences. Applications can be many: military, commerce, education, industrial processes, health, and expert systems in different areas. With this in mind this work proposes a *fault tolerant MAS*. This paper shows a model of collaborative agents architecture that share a common *blackboard* allowing a common knowledge source of the environment and the communication between agents through it, with soft kind restrictions. In this case, the system has available a planning, based on the earliest deadline first algorithm. If external factors (such as local faults) modify the behavior of the system during communication stage, it is important that the system keeps working successfully. In this proposal, the *network control system* is able to reconfigure itself to assure that the blackboard, which reflex the environment and has an own knowledge source, will be updated by messages between agents, and the task will be transmitted through some assigned communication channel using multi-agent approach. Each module in

charge of this reassignment is represented by a multi-agent system that has the necessary knowledge to react to those eventualities, reassigning its transmission frequency in order to fulfill the service.

2. Multi Agent System

In artificial intelligence, one of the hardest problems is to consider the real-world events, as input to the entity in charge of solving a particular problem. The latter is one of the reasons for the introduction of the agents in the field, where the real world with its dynamic changes, should be considered by the entity responsible for problem solving. This work introduces an algorithm based on artificial intelligence techniques like *blackboard architecture*, and *collaborative and reactive agents*. This permits distribute the decision making process into reactive autonomous agents all of them are incorporated in a MAS. This translates into time and the more time is available for the agent's computation, the more intelligent the behavior will become. Furthermore, the iterative improvement enables the planning process to easily adapt the plan to change or unexpected situation. In the case of the reactive agents, they do not have internal models of the environment and based its action in a cycle of stimulus-reaction, according to the events presented in the current state of the environment in which are embedded; hence, it could not have objectives, nor explicit planning mechanisms and still be able to solve problems classified as complex. However, this situation can be partially resolved by the different ways in which you can achieve the interaction between them and between them and the environment. In this work an agent is a real or abstract entity that is: 1) able to choose about himself and about the environment; 2) has a partial representation of the environment and 3) within a universe multi-agent is able to communicate with other agents. The behavior of this agent is the product of their comments, their objectives and their interactions with other agents. This form of approximation is based on the interaction and the purpose of making many specialists agents cooperate, instead of using a central program and general to solve a problem as it does the traditional artificial intelligence. System components emerge with a semantic skill obtained from a decomposition of the problem. This decomposition can meet the agents, their responsibilities and interactions. The skills acquired can be viewed conceptually as the points where decisions, such as levels of abstraction of a plan that is the solution of a problem or as partial solutions to solve the problem are taken. It follows that planning is in the analysis and design, hence it is so important. The internal system modeling is done based on the behavior you want to display the system. That is, we must have a clear and complete picture of what we want to do with the system. Otherwise it would be impossible to set sub-goals according to their tasks to each of the agents. This means that system designers with reactive agents must invest time in design to let precompiled or hard-wire fixed in the action-selection mechanism. And so it is concluded that while planning mechanisms (traditional artificial intelligence) leave much work to deliberative agent. Reactive deliberative approach leaves much work to the designer. According to [3], in the *analysis and design* of a MAS we must take into consideration next aspects: 1) *agent modelling*: an agent need a model of its world; 2) *multi-agent planning*: in some cases agents will share plans in order to coordinate their behavior or to achieve a goal using others; 3) *social relationships*: agents may have social relationships, 4) *interaction*: agents may interact. In a multi-agent world in which interaction is not pre-defined, agents may need models of each other to decide how interact; 5) *communication*: agents may communicate to exploit interaction and ensure coordination. Taking into consideration last aspects next section are related with the analysis and design of the agent.

2.1. Agents

According to Smart framework proposed by [3] and resumed in Figures. First of all, we need to define a goal, which is a group of objectives to be fulfilled by an entity [3]. The agents must select their actions depending on the environment and their own and co-share goals with other agents. A goal describes a set of states, plus the attitude that the agent should have toward them. It is represented by a state made up of a set of affairs that can be carried out within the environment. Now we will define the agents within the aforementioned hierarchy (Figure 1). Here an *agent is an instantiation object, which is associated with one or several goals*. We have to consider that an agent goal can generate a *passive* or *active* reaction. *Passive goals* are goals which are imposed or assigned and *active goal* is how handles the actions to satisfy its goals interacting with the environment. An agent immersed in an environment has a set of perceptions producing the only attributes that the agents can get from the outside world; these perceptions depend on the *agent capabilities* and also on the *world state*. It is important to

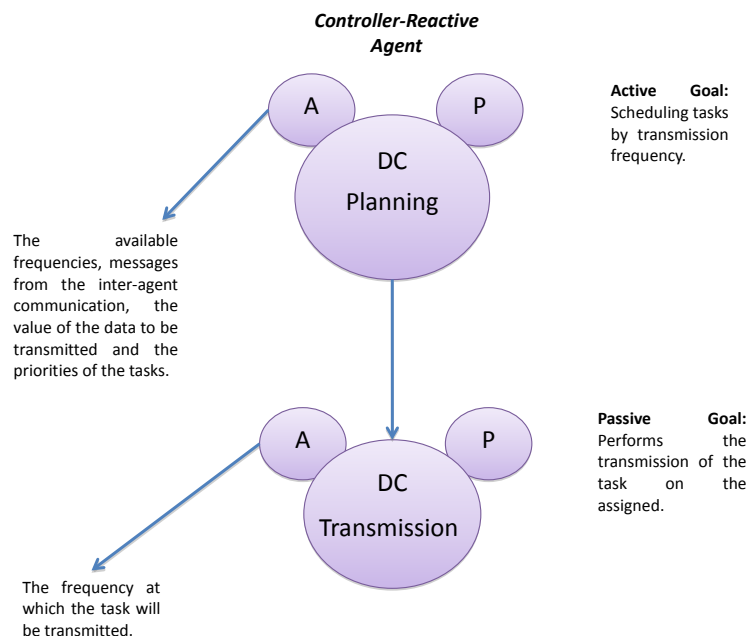


Figure 1. Controller Reactive Multi-Agent System.

mention that *an agent cannot perceive all the environment attributes*; therefore, its behavior is based only on a subset of those attributes. This subject is related to the partial world representation proposed by Brooks [5].

2.2. Case Study

According to the number of nodes in the system, it means that one MAS, which will call *controller-reactive agent* from now on; is assigned to each node in the network. The case study have two agents one *active* and the other *passive* as shown in **Figure 1**, it do not have presentation part, because all the proceeding is inside the software system.

Another important characteristic is the collaboration between the agents to select the most convenient option supported by the blackboard, due to this collaboration and the information of the blackboard architecture, the *controller-reactive agent* has a more real perception of the environment and is capable of a *self-reconfiguration* choosing its transmitting channel. In this way it ensures the transmission, getting a new frequency every time it needs to transmit. A real time distributed control system should execute its tasks at a specific time; it uses an algorithm to organize them named the *Planning* [6]-[8], it could be, for instance, earliest deadline first (EDF) to guarantee a good operation of the system. However there is no guarantee of this behavior; there are external factors that modify the operation at the communication stage [5] [9] [10], so one way to guarantee that the information arrives to its destination is being aware of this variation. The goal of this work is to provide a way to have the information transmitted at different communication channels (according to several frequency transmissions); where, there is not a static frequency (it refers to quantity of information transmitted in a unit of time) to transmit, instead of an option is selected from a blackboard, taking into account the restrictions and the agreement amongst agents. The following strategy is considered to overcome undesirable effects when collisions are present during communication between processes in a distributed environment. If a process needs to execute a set of tasks, which must be planned, one has to take into account that the sum of the times that each of them consume must be less than the specified time to run the process. If the tasks comply with this condition are said to be feasible to be planned. In this case it's a bandwidth planning, a task is likely to be planned if there is a suitable frequency within its bandwidth to ensure the time period in which it should be transmitted (to guarantee the real-time). The variables considered in each planner are: 1) the number of tasks; 2) the number of frequencies; 3) allowable range of frequencies for each task; 4) priority tasks (all tasks have different priority); 5) duration.

In a test schema where you have two agents with three tasks for planning each one, you can distinguish be-

tween two scenarios, in the first one all the tasks of two agents can be scheduled, in the second one there can be tasks that can be scheduled in this moment therefore the algorithm identifies them and does not allow that the information to be transmitted, as there is no available frequency, therefore they continue by selecting the best option for the other tasks. On having continued with the review of tasks that must be planned, it checks again if it can be scheduled in the current environment and continues with the procedure. Dynamic bandwidth scheduling allows the transmission of a task to a different speed each time, that is to say, the transmission frequency is changed using a parameter that indicates the period of time in which it is used. To notify to other agents the frequency that has been selected a message is sent and there are updated the blackboards of each of them. Our work is focused on performing the ordering of the communication between processes in a dynamic way, so the information can be transmitted on different communication channels to size the bandwidth and improve the performance of the system. That is, we use a static or predefined scheduling of communication, and allow each process to communicate through different channels, taking into account several restrictions, like: bandwidth, availability of frequencies and priority. In this paper distributed artificial intelligence is pursued based on collaborating multi-agent systems having a shared memory, by simplicity we call it, blackboard (black board architecture) [11] [12]. To achieve this goal, collaborating reactive agents are used at the scheduling stage in a distributed control system in order to implement an agreement with respect to the use of several communication channels (see Figure 2). Each agent performs a local scheduling algorithm supported by the blackboard information, which is modified periodically reflecting the state of the environment in which they are acting. If a *controller-reactive agent* needs to modify the messages sent through a specific communication channel, this is agreed with the rest. If this is a valid option the related agent is capable of schedule its transmission. This is implemented by the use of a blackboard among reactive agents in order to get a consensus with respect to the use of the available channels on the bandwidth.

Each node is composed by a *controller-reactive agent*, which performs the scheduling of tasks, assigning an adequate frequency for each process in order to transmit. To determine if the related communication channel is the adequate, the *controller-reactive agent* produces a selection based on the information at the blackboard, [6] [11] [13] [14]. The selection mechanism consists of a consultation among *controller-reactive agents* to know which communication channel is available, according to the interval assigned per process and the related global scenario. For the case of collision there is a mechanism based on the priorities of the tasks and the identifier of each *controller-reactive agent*. On the other hand we must remember that we have more channels than agents. The latter ensures that the message will always be transmitted. It is important to mention that competition focuses on the fastest channel. The blackboard is a table that reflects the state of the environment. In order to assign a communication channel, each *controller-reactive agent* reviews the general information contained in the blackboard and reach an agreement based upon two key issues: the flexibility of local processes to perform communication and the pursued scenario with these modifications. This blackboard contains the frequencies assigned to every *controller-reactive agent* per local process. So every certain period the *controller-reactive agents* are looking at and modifying this table if needed. In case of collisions, if the chosen channel is busy, the

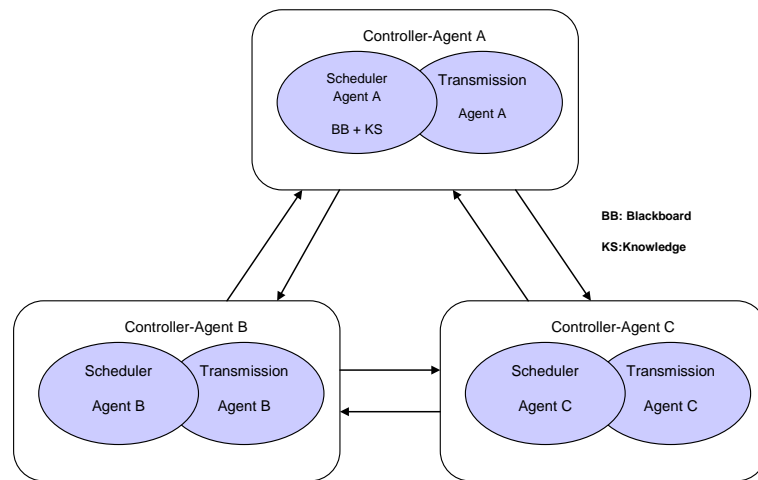


Figure 2. Scheduler inside of a Controller-Reactive Agent Anatomy.

process is re-scheduled to choose another communication channel with respect to the available range assigned per process. The adaptation capacity represents a robust way of adaptability and predictability. The last idea drives the system to maintain or improve its operation. The final goal is to get a balance between the dynamic environment and a decentralized process in the system core [14] [15].

2.3. Collaborative Systems

This paradigm refers to have independent modules that cooperate in order to solve a problem together. The details and complexity associated with each module is encapsulated within them. Problems are solved in a distributed way; this is accomplished by dividing the knowledge of the solution among the different modules [9] [16] [17]. Depending on the different ways of communication modules, we have: 1) strongly coupled systems, where communication takes place through shared memory; and 2) weakly coupled, where communication is through message passing. The case of strongly coupled systems is based on blackboard architectures. This kind of processing is driving by data and by the target, having at the end the problem performed. In the other side the case of weakly coupled systems, the communication among modules is achieved through message passing. There are different ways of message passing: 1) coherence based on cooperation, negotiation-based agreement; 2) hierarchy and organization; 3) computing and communication [4] [13]-[17]. In the case study we conceive a hybrid module that combines the two aspects of weakly coupled systems and strongly coupled systems. Corkill [16] proposes a very flexible way to combine the blackboard architecture with the multi-agent architecture. Taking into account the characteristics of collaborating processes. The case study is based on a Full-Fledged Blackboard Agents; where there are different reactive agents with its own knowledge source, and its own blackboard. For further information review the Corkill document. With this approach it is designed a model that allows autonomy based on the ability to share data. According to Corkill [16], a collaborative system must have the following 6 features: 1) representation, having a sufficient number of modules that allow the understanding of the system as a whole; 2) incorporation, the modules are able to recognize the stimuli according to their skills; 3) emergency modules provide the ability to find the information they need to react in the environment; 4) interaction, create modules that are able to use the concurrent work of others while they are working in a shared task; 5) integration, providing them with a mechanism that allows them to combine and interpret the results obtained by other modules; 6) rationale, design modules capable of doing the right things at the right moment. This work has a model that allows take advantage of essential elements from next two approaches: 1) share data and; 2) have autonomy. These approaches make possible have modules with an emergent behavior. Taking into account the actions of other agents [11] [14]-[16] [18]. In order to design behaviors and achieve the six characteristics of a collaborative system, the *controller-reactive agent* is designed considering the detailed of the SMART Framework proposed by d’Inverno and Luck [3]. It provides us a classification based on objects, agents and autonomous agents [14] [16].

3. Basic Implementation

We propose a collaborative system based on a shared memory and multi-agent architecture taking into account that communication need share a set of data; that allows triggering different actions in the different *controller-reactive agents*. These mechanisms generate the autonomy. According to the theory developed by d’Inverno and Luck [1] [3] [16], a multi-agent system is one that contains a collection of two or more agents. Since goals cannot exist without having been generated by autonomous agents, it is impossible for agents to exist without autonomy somewhere in the system. So the definition of a multi-agent system is any system that contains: 1) two or more agents; 2) at least one autonomous agent and 3) at least one relationship between two agents where one satisfies the goal of the other. For this case the multi-agent system has an autonomous one, the *planning agent*, and a passive one, the *transmission agent* (Figure 3). The blackboard is used by the autonomous agent in order to handle its autonomy, through the knowledge of the changes in the environment, which in the case under study refers to knowing which *controller-reactive agent* is using a given communication channel per process. For further clarification, see Figure 2 and Figure 3 [11] [12] [14] [16] [18]. A variant of the models proposed by Corkill [16] is used, where each autonomous agent (planning) has just one source of knowledge and a blackboard (see Figure 2). This MAS, called *controller-reactive agent*, allows: 1) communication, through the data written in each of the blackboards; 2) coordination, through the updating of such data; and 3) an autonomous behavior, due to the ability to perceive the dynamic environment through the data on the blackboard. In what

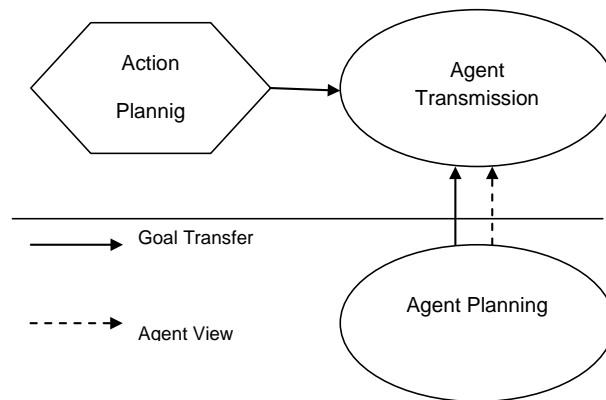


Figure 3. Dependence of agents in a Multi-Agent System.

follows we describe the distributed control problem and the proposed solution. In the proposed MAS, at each *controller-reactive agent* there is an autonomous agent that represents the knowledge source and a passive one that has the necessary capabilities to achieve the objectives of the autonomy [14] [15] [17]-[19]. **Figure 3**, shows the dependences between active and passive agents. Their functions are described using the notation of d'Inverno and Luck [3].

Planning: Autonomous Agent

Type of Goal: Active

Objective: Scheduling tasks taking into account the transmission frequency.

Perceptions

Action it perceives: a) selects an adequate frequency for the transmission of a task, based on the record of available frequencies (blackboard).

b) Updates its record (blackboard) in accordance with the messages coming from other agents.

c) Modifies the frequency to transmit the task.

What can it perceive? The available frequencies, messages from the inter-agent communication, the value of the data to be transmitted and the priorities of the tasks.

What will it perceive? The adequate available frequency, in accordance to the priorities of the task at hand.

Action

Performs the planning of tasks to be transmitted in a dynamic way for a specific frequency, chosen from the revision of a shared memory (blackboard) that shows the possible options to transmit according to the allowed range, and updates itself checking the mailbox, were there is information about whether other agent is using or releasing a frequency range. In a collision case, the *controller-reactive agent* communicates with the other *controller-reactive agent* involved and defines the use of the *frequency based on the priority of the task*, updating the blackboard later on.

Transmission: Passive Agent

Type of Goal: Passive

Objective: Performs the transmission of the task on the assigned frequency.

Perceptions

Action it perceives: Transmits the task.

What can it perceive? The frequency at which the task will be transmitted.

What will it perceive? The frequency in which the task will be transmitted.

Action

Transmits the tasks in the frequency decided by the *planning autonomous agent*.

In this case, since we are dealing with frequency planning, we may say it will be able to be planned if there is an adequate frequency to warrant the time in which it must be transmitted (in real time) [6] [7] [8] [13]. The *Planning agent* may fulfill several actions depending on the variables that it is managing at an instant of time (t); 1) it verifies if there is a message in the mailbox. If there is a collision, *i.e.* if a chosen frequency is being used, a message is sent to the respective *controller-reactive agent*. Each *controller-reactive agent* knows the priority of the processes and the identifier of every *controller-reactive agent*. Taking into account the former data and the

criterion based on the priorities of the processes, a channel is assigned and thus the assignment of frequencies is restructured from the *Planning agent* for this current scenario. In the agent's blackboard (**Table 1** and **Table 2**) there is a list of available channels that are related to the process willing to communicate and to the corresponding *controller-reactive agent*. The attributes of the Frequencies table (**Table 1**) are: *frequency*, *task* and *agent*. If the value of *process* is less than zero, it means that no communication channel has been assigned to the process; otherwise, there is a frequency assigned to the process (task).

Inside the same agent's blackboard there is a table of attributes per process, which shows the minimum and maximum frequency. The blackboard is updated as a result of the communication between agents [15] [16] [18]; in this way each *controller-reactive agent* finds out the state of the others as well as its own. The attributes of the priorities table are: *Process*, *Port*, *MinFrq* and *MaxFrq*, *Duration*. For practical purposes it is assumed that the process is equal to the task priority, to avoid redundancy information. The *Port* column of the table shows if the process is a proper one (if the number is greater than zero) or the process of another *controller-reactive agent* (if it is less than zero) or is an available frequency (if it is equal to zero). The minimum and maximum frequencies, together with the priority number, allow us to select the frequency more adequate for each process, *i.e.*, they will restrict and bind the range of frequencies in which the process can be transmitted. *Duration* shows the period of time on which the task will be transmitted in an assigned frequency.

4. The Mental Model

In some areas it is possible to formulate cognitive theories of competition, specifying: what has to be calculated, when, and why, then on the basis of these theories an algorithm is developed to represent the mental model. This area of study is known as the theory of competition and is based on performing mental models. The essential part of mental models is its functional characteristic. They are defined as functional abstractions, with respect to the job or jobs, and provide a deductive framework for problem solving. The basic aspect of this doctrine lies in assuming that most of the inferences made daily depend on: spontaneous conclusions and a set of arguments that do not always lead to a valid conclusion. Once we have the design, the next step is to get the mental models; it is a pseudo-code that shows all the inputs, outputs and methods. The dynamic planning is based on a MAS supported by the *Blackboard in a Full-Fledged* scheme in order to share the information about environment. This configuration is due to it is required that agents must be independent in their self-performance, that means, they keep sharing information through the blackboard even if one of them stops working. Each agent, has its own *Knowledge source* (independent modules that contain the expertise needed to solve a problem) based on their competences and local views, that allows interpret the information according to their goals, executing the best action to achieve its mission. To execute a process we must take into account the sum of every period of time that are used to realize a task by each agent, which must be lower than the time specified for the process. If the every task fulfills these conditions, it means are feasible to be planned. In this case, as it is a bandwidth plan-

Table 1. Frequencies (blackboard).

Frequency	Task	Agent
1	-1	-1
2	1	1
3	2	2
5	3	3
6	-1	-1
8	-1	-1
10	-1	-1

Table 2. Priorities (blackboard).

Task	Min Frq	Max Frq	Duration
1	2	10	10
2	2	10	20
3	2	10	30

ning, a task is feasible to be planned if there is an appropriate frequency available to guarantee the task must be transmitted in time (Guaranteed Real Time). The *Planning Agent* must perform many actions: 1) if it is a task, verify if it has a message in the mailbox; if it is true, 2) selects a feasible frequency to transmit the task; 3) sends a message to the other agents notifying the selected frequency. If a frequency is schedulable, the frequency is changed (just before this, the mailbox is checked again to ensure that the frequency selected is available) and, at that time, that interaction between an autonomous and non-autonomous agents begins, the *Transmission agent* receives the task and the frequency, and executes the transmission returning the control to the *Planning agent* in the next instant of time.

<p>The planning agent mental model is: Begin Initialize the Blackboard Get the task from the queue tasks agent Verify the system state If (the agent must act due to the events generated in the environment) Select the frequency from the Blackboard (Contemplates the Blackboard update) If is a process feasible to be planned then Begin Verify the mailbox to guarantee the frequency is available Notify to the other agents that the frequency is being used. Modify the transmission frequency End Else Get next task in queue (second step in the algorithm) End</p>
--

4.1. Planning Agent Attributes

The Blackboard has a list of available frequencies that are related to the task that is being transmitted.

The Frequency table attributes are: Frequency, Task and Agent. Task, is the task that the agent must transmit in a specific moment, here the duration is known and is the parameter that indicates when must be changed the transmission frequency. In the Blackboard (**Table 1**), if Task is less than zero indicates that is available, if not, indicates that the task is being transmitted. The Blackboard also contains a table for Frequency priorities (**Table 2**), which indicates the minimum and maximum frequency on the bandwidth, defining the priority for each Frequency of every task. Both, *Task and Frequency are updated as a result of communication between agents*, so, in this way the agents knows the status of others. A multi-agent architecture is based on the premise that there should be at least two autonomous agents and one other non-autonomous to help them to achieve their goals [5]. This proposal includes two agents, the first, Planning is autonomous and performs the bandwidth planning for each task and it assigns a specific frequency for transmission and manages the blackboard. It is the responsible of the communication with other agents by sending a message to the mailbox, updates the Blackboard if a message is received in the mailbox. Finally changes the frequency to transmit the task. The second agent is passive, and called Transmission, is who executes decisions from *Planning agent* and produces changes directly to the environment.

4.2. Planning Agent Methods

Planning selects and change the frequency transmission, sends messages to achieve communication between agents. All this is done through of five methods:

a) Select Frequency (SelectFrc). Analyze the Blackboard to get a feasible frequency to transmit a task. It depends on the priority and the minimum and maximum frequency on the allowed bandwidth. If a task is waiting for a frequency assignment, the mailbox is checked and the blackboard is updated, if it is a schedulable frequency, it sends a message to the other agents and executes ModifyFrc.

Inputs: Priority Task

Outputs: return the Frequency assigned or -1 if the task is not schedulable

b) Change Frequency (ModifyFrc); modifies the frequency where the task is to be transmitted.

Inputs: Task, frequency

Outputs: None

The mental model to change the frequency is:
 If there are a task to be transmitted in a preselected frequency,
 It is transmitted
 It checks for another task in the queue

c) Send Message to the Mailbox (SendMail)

Sends messages to the mailbox in order to change information with other agents.

Inputs: Message

Outputs: None

The mental model to send a message is:

Sends the message to the mailbox

d) Check Mailbox (CheckMail)

Checks the mailbox in order to perceive the environment, interchanging messages with other agents.

Inputs: None

Outputs: Message, if there's no message return 0

The mental model to check mailbox is: gets message
 If there are a message in the mailbox
 returns message
 if there aren't message in the mailbox
 returns empty mailbox

Transmission agent methods: transmission has only one method to transmit a task,

a) It Transmits a Task (Transmit)

Inputs: Task

Outputs: None

The mental model to transmit a task is:
 It is enable output port
 It transmits a task

5. Experiments

The dynamic planning in a bandwidth, allows the task transmission in a different speed, see **Figure 4**. The frequency of transmission is changed using a duration parameter, which indicate the period of time, used in (see **Table 2**). Taking this into account, the parameters are defined and the **Table 1** and **Table 2** are built, while planning is carried out see **Figure 4**, it is noted as the transmission speed decreases and increases in run time, considering duration parameter. While the frequency is lower there are more pulses in the graph, it means the transmission speed is higher, in the other hand, while the frequency is higher, there are less pulses in the graph, it means the transmission speed is lower.

Figure 5, shows changes in frequency for each task, it is important to note that frequencies are not repeated at the same instant, it is possible due to the messages interchange between agents. In the first graph Task 1 is shown, where the frequency change is each second, the second graph shows Task 2 frequency changing every two seconds and the third graph shows Task 3 by the frequency change every three seconds. Whenever contemplate a new frequency selected frequencies being used for other tasks.

It should be noted, that Task 1 corresponds to agent 1, Task 2 to agent 2 and Task 3 to agent 3, and it is possible to plan n Tasks per agent.

6. Conclusion

The root of the problem comes from the distributed system in fault situations and its capability to be robust to these faults. In this case (distributed control system), special attention is placed on the communication aspect, and a dynamic planning is devised allowing it to organize the task communication, choosing a frequency at execution time. Due to the fact that the conditions of the system are dynamic, *the assignment must be coupled to*

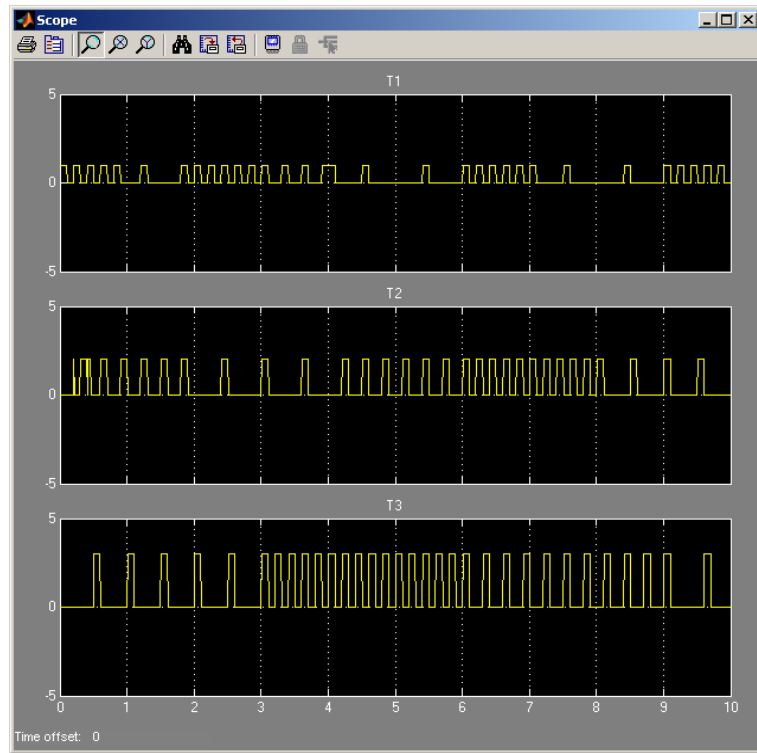


Figure 4. Frequency assignment.

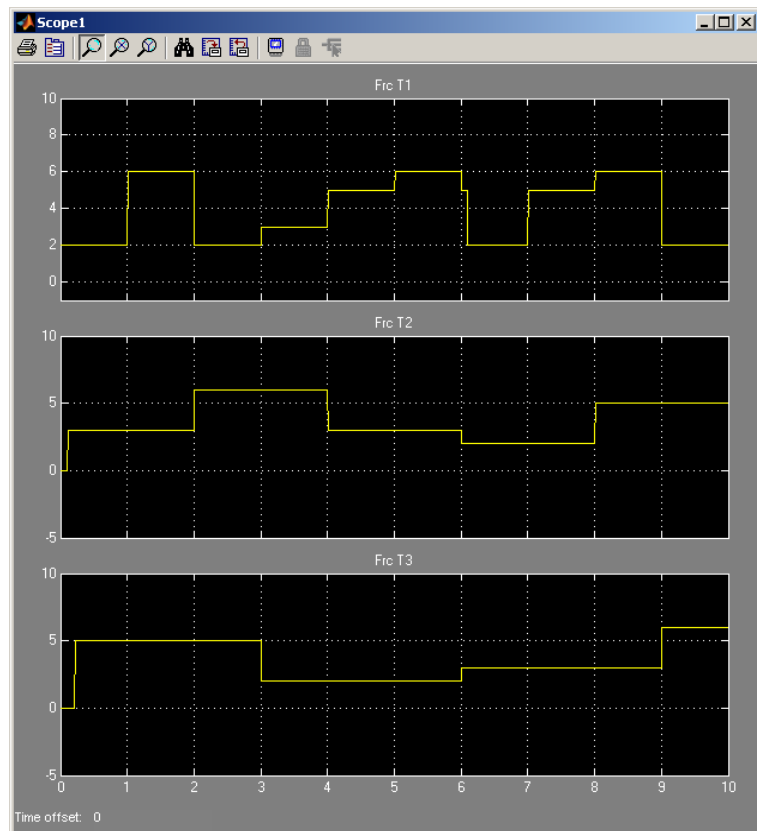


Figure 5. Time in the assignment frequency.

the environment. This proposal presents a strategy in which the information is transmitted in a flexible manner, since the value of the transmission frequency is no longer constant. There is a range of frequencies in which the process is allowed to transmit, making the management of frequency range options possible. It is worth mentioning that within the distributed system several local schedulers are required, which together perform the total planning of the system. The blackboard architecture allows displaying the partial solutions along the way to the global solution of the system; these solutions are those contributed by each one of the *controller-reactive agents* when making a choice for each process in which each one is taking part. At design time agents with blackboard included (Full-Fledged Blackboard Agents) was chosen [11], since it was required that *collaborative multi-agent system* be independent through the data they handle, allowing them to develop their *controller-reactive agents*, *i.e.*, if one of them ceases to operate, this doesn't mean that others stop working, because in the case in point the system is distributed physically. With this in mind we model an hybrid module that combines the two main aspects of the two different multi-agent architectures. On the one hand a weakly coupled systems (messages passing) and on the other a strong-coupled system (blackboard) [11] [12] [16]. On the other hand, the multi-agent system warrants sending the information, tasks and messages for the communication among the *controller-reactive agents*. However, the one who warrants that the information gets to its destination is the communication protocol, which is being used [20] [21]. It is important to remark that this algorithm was thought for a real time distributed control reconfiguration systems, in order to manage all the faults that could appear in it, and then take one action to *stabilize the system*. In this case, if a *controller-reactive agent* stops its work, the others search for a free frequency and transmit its information in the selected frequency. That's the reason why this algorithm has a self-reconfiguration core. No matter if a fault occurs, the system get on working because its self-reconfiguration. Like we are using a multi-agent architecture we can add more parameters to be perceived from the environment (thought the autonomous *controller-reactive agent*), and more agents to execute specific actions [15] [17] [18] [22]. One use could be in the cognitive radio (CR) which is a wireless communication in which one element, the transmitter can detect which channels are busy and which are available to send its information, through intelligent methods, all of this independently of the used technology [10]. According with Tachwali, Basma, Refai [23], other concepts important to take into account in the design of an effective cognitive radio are configurability and adaptability. Configurability is related with the ability to change the communication mode, as is inferred from the definition of cognitive radio. And adaptability enables an automatic communication mode selection based on wireless channel conditions. The demand for ubiquitous wireless services has increased during the last decade [3]; spectrum resources are limited and are allocated according to a fixed spectrum assignment policy. So another aspect to consider is the operation opportunistically in the frequency band allocated to a licensed system, on the primary user. In the case of cognitive systems [2] we make a contribution to a complex behavior basically because the design of the *controller-reactive agents* is based on a local view of different elements, last take into consideration the environment. The *controller-reactive agents* can have an organization without the external intervention. We propose a synchronize behavior based on hierarchy of the different processes. There is an assurance to achieve the final goal. Finally this algorithm is robust because it is tolerant to fails. The authors consider that the algorithm proposed is independent of the case study and could be implemented to: 1) improve the communication among process; or 2) to simulate human reasoning of an expert; given an additional positive aspect with reactive characteristics that implies autonomy.

Acknowledgments

This paper is part of the research undertaken by Tzitziki Ramírez González to obtain the MSc. Degree in Computer Sciences at Universidad Autónoma Metropolitana-Azcapotzalco. Also as part of the Divisional project Soft Computing and Applications in the sub line Multi Agent Systems: Modeling and Simulation funded by the same University.

References

- [1] Laureano-Cruces, A.L., De Arriaga, F. and García-Alegre, M.C. (2001) Cognitive Task Analysis: A Proposal to Model Reactive Behaviors. *Journal of Experimental & Theoretical Artificial Intelligence*, **13**, 227-239. <http://dx.doi.org/10.1080/09528130110063092>
- [2] Quiñones-Reyes, P., Benitez-Pérez, H., Mendez-Monroy, E., Cardenas-Flores, F. and García-Nocetti, F. (2006) Recon-

- figurable Network Control using Fuzzy Logic for Magnetic Levitation Case Study. IFAC SAFEPROCESS.
- [3] d'Inverno, M. and Luck, M. (2006) Understanding Agent Systems. 2nd Edition, Springer Verlag, Berlin.
 - [4] Laureano-Cruces, A. and Barceló-Aspeitia, A. (2003) Formal Verification of Multi-Agent Systems Behavior Emerging from Cognitive Task Analysis. *Journal of Experimental & Theoretical Artificial Intelligence*, **15**,407-431. <http://dx.doi.org/10.1080/0952813031000119719>
 - [5] Brooks, R. (1991) Intelligence without Representation. *Artificial Intelligence*, **47**, 139-159. [http://dx.doi.org/10.1016/0004-3702\(91\)90053-M](http://dx.doi.org/10.1016/0004-3702(91)90053-M)
 - [6] Benítez-Perez, H. and García-Nocetti, F. (2003) Reconfigurable Distributed Control Using Smart Peripheral Elements. *Control Engineering Practice*, **11**, 975-988. [http://dx.doi.org/10.1016/S0967-0661\(02\)00209-5](http://dx.doi.org/10.1016/S0967-0661(02)00209-5)
 - [7] Frey, D., Nimis, J., Wörn, H. and Lockermann, P. (2003) Benchmarking and Robust Multi-Agent-Based Production Planning and Control. *Engineering Applications of artificial Intelligence*, **16**, 307-320. [http://dx.doi.org/10.1016/S0952-1976\(03\)00075-7](http://dx.doi.org/10.1016/S0952-1976(03)00075-7)
 - [8] Liu, J.W.S. (2000) Real Time Systems. Prentice Hall, Upper Saddle River.
 - [9] Abdelzaher, T.F., Stankovic, J.A., Lu, C., Zhang, R. and Lu, Y. (2003) Feedback Performance Control in Software Services. *Control Systems Magazine IEEE*, **23**, 74-90. <http://dx.doi.org/10.1109/MCS.2003.1200252>
 - [10] Bellahsene-Hatem, N.R., Mostefai, M. and El Kheir, A.O. (2011) Contribution au Diagnostic des événements défalliants dans un système Dynamique Hybride. *Proceedings ICGST Conference on Computer Science and Engineering*, Istanbul, 19-21 December 2011, 137-142. <http://www.icgst.com/paper.aspx?pid=P1111134764>
 - [11] Cervin, A., Henriksson, D., Lincoln, B., Eker, J. and Arzen, K. (2003) How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and True Time. *IEEE Control Systems*, **23**, 16-30. <http://dx.doi.org/10.1109/MCS.2003.1200240>
 - [12] Ramírez-González, T., Quiñones-Reyes, P., Benítez-Pérez, H. and Laureano-Cruces, A. (2006) Reconfigurable Fuzzy Takagi-Sugeno Networked Control Using Cooperative Agents and xPC Target. *Proceedings of International Symposium on Robotics and Automation (ISRA 2006)*, San Miguel Regla, Hgo, Mexico, 25-28August 2006.
 - [13] Dong-Chan, O. and Yong-Hwan, L. (2010) Cooperative Spectrum Sensing with Imperfect Feedback Channel in the Cognitive Radio Systems. *International Journal of Communication Systems*, **23**, 763-779.
 - [14] Laureano-Cruces, A. and De Arriaga-Gómez, F. (1998) Multi-Agent Architecture for Intelligent Tutoring Systems. *Journal of Interactive Learning Environments*, Swets & Zeitlinger, **6**, 225-250.
 - [15] Laureano-Cruces, A., De Arriaga, F., Ramírez-Rodríguez, J. and Escarela-Pérez, R. (2006) Control of Intelligent Learning Systems with Reactive Characteristics. *Interactive Learning Environments*, Swets & Zeitlinger, **14**.
 - [16] Corkill, D. (2003) Collaborating Software: Blackboard and Multi-Agent Systems & the Future. *Proceedings of the International Lisp Conference*, New York.
 - [17] Jennings, N.R. and Bussmann, S. (2003) Agent Based Control Systems: Why Are They Suited to Engineering Complex Systems? *IEEE Control Systems*, **23**, 61-73. <http://dx.doi.org/10.1109/MCS.2003.1200249>
 - [18] Laureano-Cruces, A. and Espinosa-Paredes, G. (2005) Behavioral Design to Model a Reactive Decision of an Expert in Geothermal Wells. *International Journal of Approximate Reasoning*, **39**, 1-28. <http://dx.doi.org/10.1016/j.ijar.2004.08.002>
 - [19] García-Zavala, A. (2003) Propuesta de un método de planificación para la reconfiguración en línea de un sistema de tiempo real. Master Degree Thesis, Posgrado en Ciencia e Ingeniería de la Computación, UNAM.
 - [20] Almeida, L., Pedreiras, P. and Fonseca, J.A.G. (2002) The FTT-CAN Protocol: Why and How. *IEEE Transactions on Industrial Electronics*, **49**, 1189-1201. <http://dx.doi.org/10.1109/TIE.2002.804967>
 - [21] Low, S.H., Paganini, F. and Doyle, J.C. (2002) Internet Congestion Control. *IEEE Control Systems*, **22**, 28-43. <http://dx.doi.org/10.1109/37.980245>
 - [22] Quiñones-Reyes, P., Benítez-Pérez, H., Mendez-Monroy, E., Cardenas-Flores, F. and García-Nocetti, F. (2006) Reconfigurable Network Control Using Fuzzy Logic for Magnetic Levitation Case Study. IFAC SAFEPROCESS.
 - [23] Tachwali, Y., Basma, F. and Refai, H. (2012) Adaptability and Configurability in Cognitive Radio Design on Small Form Factor Software Radio Platform. *Wireless Personal Communications*, **62**, 1-29. <http://dx.doi.org/10.1007/s11277-010-0035-3>