



**INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH
TECHNOLOGY**

**NEW PLATFORM TIMER FOR THE TEMPORAL CONTROL OF THE SYSTEMS
DISTRIBUTED WITH THE MULTI-AGENT SYSTEMS - APPLICATION IN A
PLATFORM MACHINE TO MACHINE OF REMOTE CONTROL**

M.Saqalli*, S.Benaissa, H.Medromi³, S.Tallal

Department of Computer Science, System Architecture Team, Computing, Systems and Renewable
Energy Laboratory, ENSEM School, Hassan II University, Casablanca.

ABSTRACT

In the current era where the parameter time constitutes a critical component in several domains, most of the applications are of real-time type. They can serve the domains of the aeronautics, the automobile, the multimedia, etc. The specificity of these real-time systems is in the obligation to answer in time very precise, otherwise, a defect of precise answer for the deadline can provoke even catastrophic or disastrous consequences in certain cases.

The new platform Timer arrives to solve the problem of the control of time for the systems distributed with the systems multi-agent -SMA- , sound objective it is to connect to the systems SMA to control and to elaborate a report to detail various periods taken for every agent of the SMA.

In this paper we propose a new approach multi-platform called "Timer" for the distributed architectures multi-agents. This approach is validated and implemented on platforms Machine to Machine of computer-aided for remote control of a mobile robot Khepera III via various modes of communications. The works described below validate the functioning of our architecture of control of the temporal constraints of this platform.. The objective of this Timer is to verify the capacity of every agent of the platform to perceive its environment and to elaborate an answer in required time.

This new approach can control several SMA (develop it c, c or Java) situated in different operating systems (Windows or Linux),rejoinder on several machines, And elaborate a report detail various periods taken for every agent of the SMA.

KEYWORDS: Platform M2M, Distributed System, Multi-Agents Systems, Real Time System.

INTRODUCTION

The real-time systems are more and more present in the everyday life, we find them in the aeronautics, the railroad transport, the automobile, the household electrical appliances or the multimedia. We indicate by real time, any application implementing a computer system the behavior (functioning) is conditioned by the dynamic evolution of the state of an environment (called proceeded). The role of the computer system is to follow or to monitor this process by respecting temporal constraints predefined in the specifications of the application. Also, we qualify in the real time any computer system which has the correction of the functioning which depends not only on the logical exactness of the results which it supplies, but especially depends on the date in which these results are produced. So, the system does not treat any more only values, but valuable couples values and time. This definition implies that the only

average speed of execution of the software does not accept the validity of the system, but temporal constraints must be respected (by examples the terms of treatments are relative to a measurable physical time and are a member of the specification of the implanted system).[1]

MULTI-AGENT APPROACH

Agent

An agent is an autonomous and intelligent, real or abstract entity, who is capable of acting on itself and on her environment in a multi-universe agent also, she can communicate with the other agents and whose behavior is the consequence of her observations, her knowledge and the interactions with the other agents [2].

Multi-Agent System

A multi-system agent is a compound distributed system of a set of agents, who feign to a certain extent the capacities of the human reasoning, the SMA is ideally conceived and implemented as a set of agents interacting between each other (Fig.1 [3]), mostly, according to modes of cooperation, competition or coexistence.

A system multi-agent can be:

- Opened: the agents enter it and go out of it freely. (An example of that is in e-commerce).
- Closed: group agents stay the same. (A football match is an example).
- Homogeneous: all the agents are built on the same model. (A working meeting, a colony of ants).
- Heterogeneous: agents of different models, granularities different. (An ecosystem).
- Mixed (or not): the agents "human beings" are integral part of the system. It is opened and heterogeneous. (A Medie workgroup by agents assistants) [4].

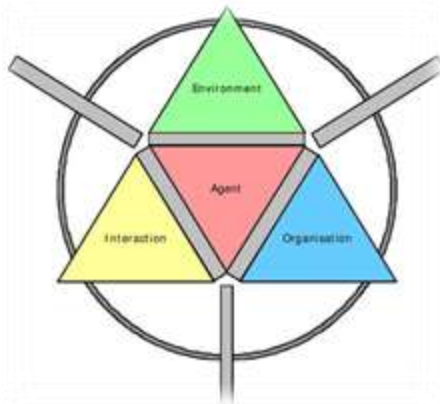


Fig1: Principle of a Multi agents System

REAL TIME SYSTEM

The systems agent times realities are defined by their capacity to react constantly to the requests of their environment by conforming to a certain number of temporal constraints. At limited time, the system has to acquire and treats the data and the events characterizing the temporal evolution of this environment, to make the appropriate decisions and to transform them into actions. The feature of the system results from its capacity to present the good exits (logical correction) at the right time (temporal correction). Because of the often critical character of this type of applications, the corresponding software and material architectures are specified, developed,

[http:// www.ijesrt.com](http://www.ijesrt.com)

© International Journal of Engineering Sciences & Research Technology

validated with the biggest care and are then congealed so as to make sure that the system that will have determinist and predictable behavior. The contribution of the multi-agents systems in this frame can then seem limited, in particular due to the autonomy and the pro-activity which we attribute generally to the agents. Actually, numerous examples of use of the multi-agents systems in a context time - reality advance more the distributed aspect than the aspects of decentralization and autonomy. It is a question of putting on together distributed entities but with motionless global architecture [5].

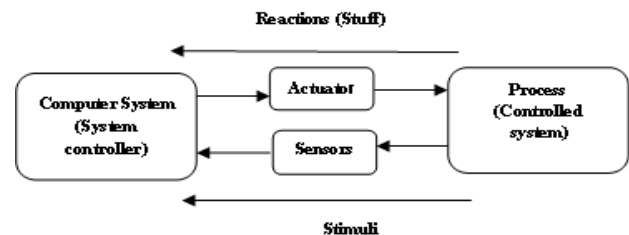


Fig2: Real Time System

Characteristics of the real-time systems

In the context of the real-time systems, the data have a validity defined at once by the domain of accepted values and by the duration of validity of these, which depends naturally on the term. The data have duration of limited existence. Otherwise, the systems of control have to respect two functional and temporal requirements in parallel [6].

According to the temporal constraints, we distinguish two big categories of the real-time systems:

- Real-time systems with strict constraints (hard real time): when all the temporal constraints must be necessarily respected [7]. The non compliance with the constraints can provoke catastrophic consequences (The systems of control of flight, systems of control of nuclear station).
- Real-time systems with relative constraints (supple real time): unlike the hard systems, the non compliance with the temporal constraints is tolerated (acceptable) by the system and without that it has catastrophic consequences [8], for example multimedia applications.
- Real-time systems with mixed constraints: tasks with strict constraints and tasks with relative constraints [9] consist.

Real-time Operating systems owners

The purpose of this part is to make a fast survey of the main real-time operating systems:

- VxWorks: the real-time executive the most used in the industry. He is developed by the company Wind River [10] which also acquired recently the rights of

the real-time pSOS, a little bit former but also widely used kernel.

- QNX: developed by the Canadian company QNX software [11], QNX is type UNIX's real-time system. He is in accordance with POSIX, allows to develop directly on the target and honorable platform the graphic Photon environment, close to X Window System.

- ECOS: acronym standing for the Embeddable and Configurable Operating System, the eCOS was initially developed by the peer society Cygnus, today connected with the Software of Red Hat society. This system is adapted to the solutions of very low imprint memory and profoundly buried. Its environment of development is based on LINUX and chain of compilation GNU with conformity in standard (standard POSIX).

Forced by the systems owners

The majority of the systems owners described this above suffer however some binding strong defects. The systems are often realized by medium-sized societies which have difficulty in following the technological evolution: the material evolves very fast, the standards software also makes more and more equipments require the integration of the constituents that we owe importer of the world of the classic computer systems or the multimedia.

Therefore, the costs of license and the rights of redistribution of the systems (or royalties) are sometimes very high. The development cost of applications around systems owners is often more raised because the tools of development are badly known of the majority of developers available on the labor market. It is thus necessary to recruit of the very specialized staff. The formations around these tools are also expensive because very specialized what obliges the editor to practice costs raised to compensate for the lack of effect of mass.

All this imply a set of binding specificities for the global management of the computing tools of the company. For it we have to migrate to the open source systems.

Advantages of the open source

The following three points of the definition of the open source software are fundamental in the case of the embarked and real-time software:

The redistribution without royalties. This point solves the economic problem of the rights of redistribution or royalties, very binding in the case of a system distributed on a large scale.

The availability of the code source. This point is even more fundamental because it is on the base of the conception of a software quality and especially maintainable in the time.

The possibility of realizing a development derived of this code source. The open source allows realizing substantial savings by integrating existing components into an industrial solution.

LINUX as real-time system

Strong by the arguments concerning the open source, it is normal to be tried by the use of LINUX as real-time system. Besides the advantages inherent to the open source, the legendary reliability of LINUX in fact an ideal candidate. Regrettably, LINUX is not natively a real-time system. The kernel LINUX was indeed conceived with the aim of making a non-specialized system based on the notion of shared time and not real time.

The LINUX community is being very active, several technical solutions are however available with the aim of improving the behavior of the kernel so that it is compatible with the constraints of a real-time system as described at the beginning of the article. Concretely, the available technical solutions are divided into two families:

- The "preemptive" said patches allowing to improve the behavior of the kernel LINUX by reducing the latent periods of this last one. These modifications do not transform LINUX into "hard" real-time kernel but allow to get satisfactory results in the case of "supple" real-time constraints. This technology is available with various projects open source and it is commercially also born by diverse specialized editors whose American MontaVista the most known is. The notion of preemptive kernel is integrated into the kernel of development 2.5.

- The auxiliary real-time kernel. The promoters of this technology consider that the kernel LINUX will never be really real-time and adds to this kernel a real-time scheduler with fixed priorities. This auxiliary kernel treats directly the real-time tasks and delegates the other tasks to the kernel LINUX, considered as the background task of more low priority. This technique allows to set up "hard" real-time systems. This technology used by RTLinux and his European cousin RTAI. RTLinux is commercially born by FSMLabs who has chaotic relations with the community open source because of the software certificate which covers the use of this auxiliary kernel. For that reason, the RTAI project - which is not associated with a commercial entity - tends to use a similar technique but not falling under the blow of the certificate.

APPROACH TIMER

Generalities

This need in validation of the data of temporal point of view brings to turn to the solution which allows

the continuous control of the time of execution following the requirements of the various systems.

Our paper concerns the implementation of a solution of temporal control for the Multi- Agents Systems so that they become in real - time.

This approach conscript TIMER, comes to meet the multiple requirements of the Real-time Multi-Agents application, such as the System of the intrusion detection [3].

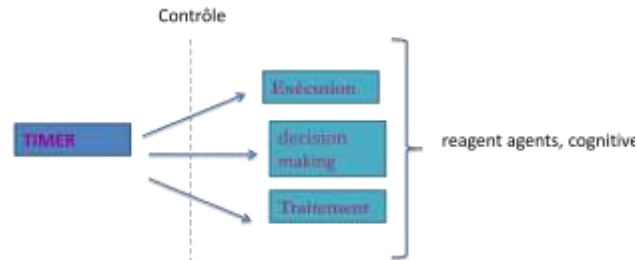


Fig. 3. Principe of a Timer Approach

Notion of TIMER

The role of the TIMER in the approached applications is the check of the time of execution of every agent. Indeed, every agent of program has to run in the temporal slice predefined or imposed by his environment with With a certain term.

This measure of time of executions has for consequence the decision-making such as the validation or not for the program.

Architecture TIMER

Généralités

Some types of Multi- Agents Systems are concerned by our approach TIMER, among others the System of Robot Kepera III. Where from the interest to work out a universal solution which takes into account the variety of existing architectures.

This capacity of adaptation must be conceived so as to answer the functional requirements of the environment to facilitate the interfacing with the system.

General Architecture:

The general architecture of the timer is decomposed into three levels:

- The level 1 (physics): send and reception the data between the timer and the SMA systems.
- The level 2 (Treatment): the processing of moderate data and the data théoriques.
- The level 3 (Report): the elaboration of the report conclusion (hard Real-time SMA, supple Real-time SMA either SMA is not real time) who considers the term.

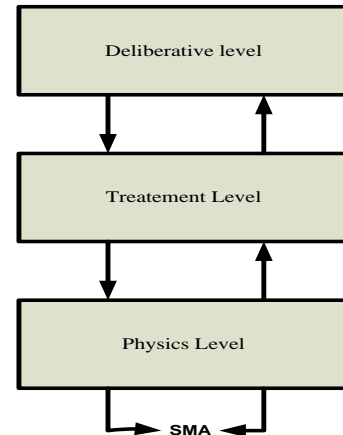


Fig 4 : General architecture of Timer

Every level supplies information at the upper level, the SMA sends to the level treatment - according to the mode of communication- the time of the beginning and of the end of the execution of his agent x by means of an application Timer Client, the level treatment receives the information of the SMA and begins to make the calculations and the comparisons; the result of its calculations is sent to the level 3 deliberative who make the decision: SMA Hard, SMA Supple either SMA not real time.

Functional Requirements

The role of the TIMER being to control the time, its existence within the Multi- Agent System does not have to force the execution of the program. This indispensable property depends on several parameters such as:

- The size of TIMER
- The interfacing with the various agents of the various SMA.
- The decision-making from information feedback.
- Etc.

Proposed real-time distributed Architecture

We propose architecture consisted of four modules:

- Multi-agents Systems: this module is a software or hardware solution which answers a given need.
- System operational, constitutes the environment of execution of the SMA (ex. Robot, anderwead, etc.)
- Real World: it is the universe either the space that groups together all the real constraints which conditions the functioning of the system operational by contribution to the environment.
- Timer: it is the module which connects all these entities (operational SMA, System and real world) and which has for objective the control of the processing time of the SMA and the system

operational with regard to the constraints of the real world.

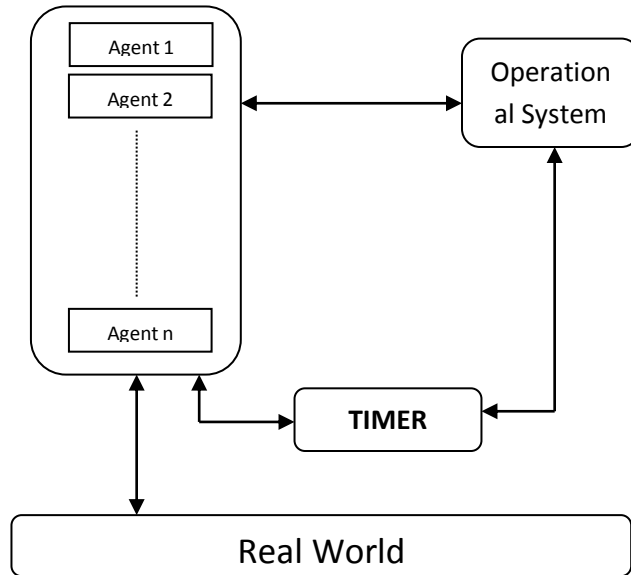


Fig. 5. Real-time Distributed architecture

Structure proposed Timer:

The best solution to answer all the temporal requirements is to set up the TIMER in the form of block, which could implement on every type of Multi-Agents System. This part illustrated by the face5 following one contains in his turn several modules.

The proposed architecture consists of one:

- Modulate database of the temporal constraints

All the temporal constraints dictated by the environment of execution or defined by the user are codified in a database to serve as reference during the check of the run time. This database is fed in every implementation of the TIMER on a Multi- Agents System to adapt the functioning of the TIMER to the required requirements.

- Module of identification of executed agent

The release of the TIMER for the time measurement is conditioned by the execution of a system SMA. For that purpose, the role of this module is to listening the Multi- Agent System to detect every agent coming to run.

The communication between Timer and SMA will be locally on the same machine either via socket TCP / IP.

- Module of time measure of execution

He establishes the means of time measure. This module is activated every time an executed agent is detected by the module of identification of executed agents.

During the reception of Tstart, we calculate a field TendPrevu for every agent in execution (Aims from which one we send a message of announcement):

$$TendPlanned = Tstart + (Tplanned(1 + Term/100))$$

If an agent of the same SMA runs several times, TSold it is the average of time sold by every execution of the same agent within the same sma.

- Module of check of the temporal constraints

In connection with the module temporal database of the constraints and the module of time measure of execution, this module estimates the measures resulting from the agent by leaning on the temporal constraints stored in the base of the data.

At the end of the check, the information on the processing time of every executed agent is registered in a dynamic report establishing the exit of the TIMER.

- Decision-making Module

The information sent by the Module of check of the temporal constraints being in the rough, they are put at the disposal of the decision-making modules to treat them and interpret them with the aim of having a vision on the respect for the temporal constraints and in case of overtaking of time prevue by taking into account the term, this module is going to send message of announcement.

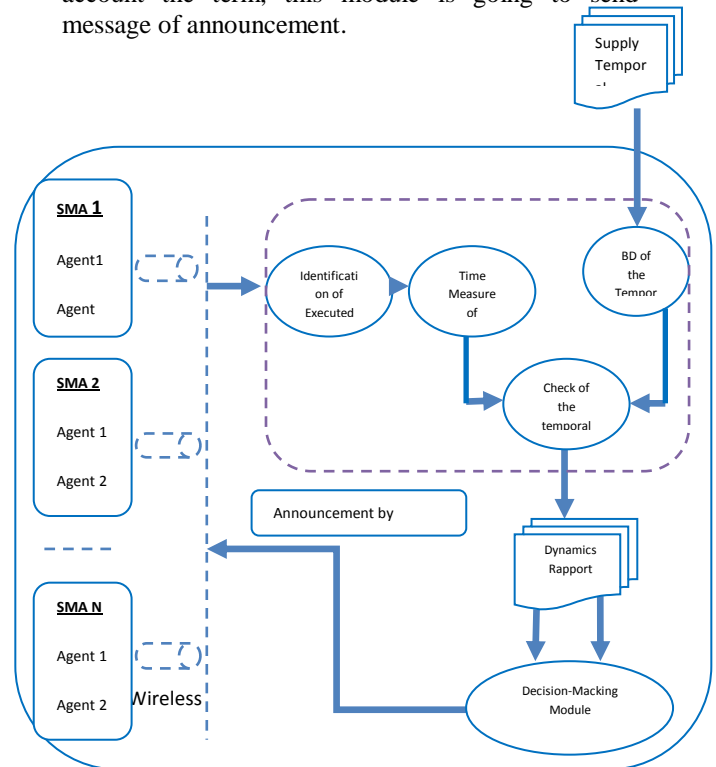


Fig. 6. Structure of proposed Timer

implementation TIMER-SMA

The approach " Timer " is the solution proposed to control the time of execution, treatment or decision-making of the "reagent and cognitive" agents forming the application of study " M2M Architecture for the Robot Kepera III " [3].

The reagent agent is an agent who has to answer in a convenient way the appeals of his environment; thus the proposed approach "Timer" is integrated into the reagent agent in the following way (Fig.6 [3]).

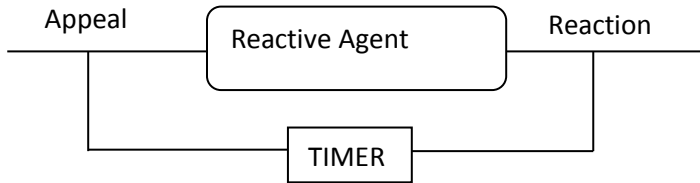


Fig. 7. Timer pour l'agent réactif

Contrary to the reagent agent, the cognitive agent receives the appeal of his environment but to elaborate the answer to this appeal he has to consult the other agents either the knowledge base; for it the cognitive agent will be controlled according to the plan of Fig.8[12].

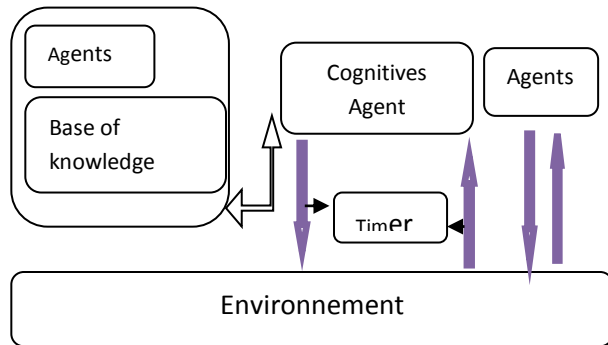


Fig. 8. Timer pour l'agent cognitif

According to time calculated by the solution " Timer " we can re-configure our system multi-agent so that every real-time agent executes his spot without falling in term.

PLATFORM OF TEST: M2M WITH THE MULTIAGENTS SYSTEM APPLIED TO THE MOBILE ROBOTS KHEPERA.

Architectue

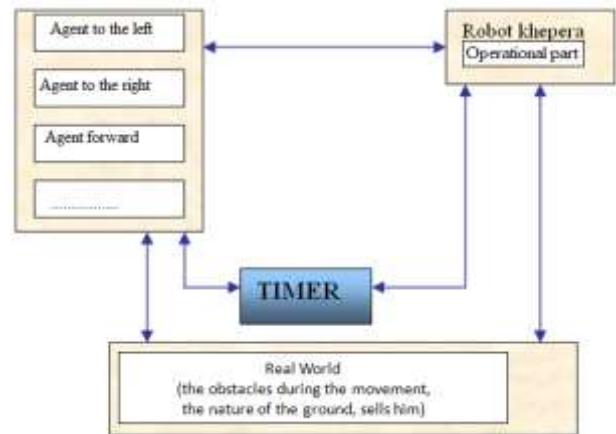


Fig. 9. Architecture Timer Appliquée à l'Architecture M2M

It is distributed architecture multi-agents, every agent represents a very particular function with the characteristic which these agents cannot discuss directly between them but have to pass by a kind of scheduler which manages their communications, their states.

Our real-time distributed architecture is applied to this model knowing that:

- The reactive agents and the cognitive agents constitute the SMA
- The robot presents the operational part.
- The obstacles during the movement, the nature of the ground, sells it constitute the constraints of the real world
- Timer is the solution proposed for the temporal validation of this system.

Realization of the platform and Experimental Test

To operate and validate our platform Timer, we based ourselves on a robotics platform. In this test the mobile robot Kepera has to follow a route by basing itself on her agents (agent rotation, left agent, right agent).

The platform Timer connects in the robot by a wireless connection. After identification of the agents executed under the robot, the timer stays there listening of every launch begins/End at agent's, and in connection with the file which contains the theoretical duration of every agent, the timer elaborates a detailed report for the various agents of the robot Kepera, even if the agent runs several times (we take the average) and on the same report we find below the various durations taken for every agent of the SMA Kepera.

Scenario of validation:

The test case consists in giving a destination in affect by the robot khépera III (Fig. 10), this destination is

formed by three states of the agent piloting:

- Agent1: gone up Agent
- Agent2: turned right Agent
- Agent3: gone down Agent

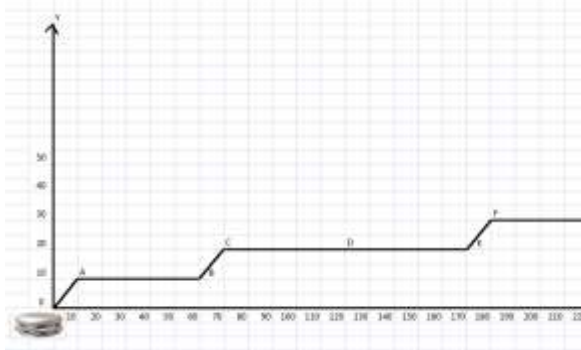


Fig 10 : Trajectory of Khépera III

Board of measures

Table 1: Board of measure of the agents of the Khépera III

Position	Coordinates	Name-Agent	Moderate time
A	(1,1)	Agent 1	5s
B	(6,1)	Agent 2	6s
C	(7,2)	Agent 1	4s
D	(12,2)	Agent 2	7s
E	(17,2)	Agent 2	6s
F	(18,3)	Agent 1	4s
G	(23,3)	Agent 2	7s
H	(23,1)	Agent 3	5s
I	(23,-2)	Agent 3	6s

Graphs :

The agent 1 is three times launched as shown in the following figure (Fig 11):

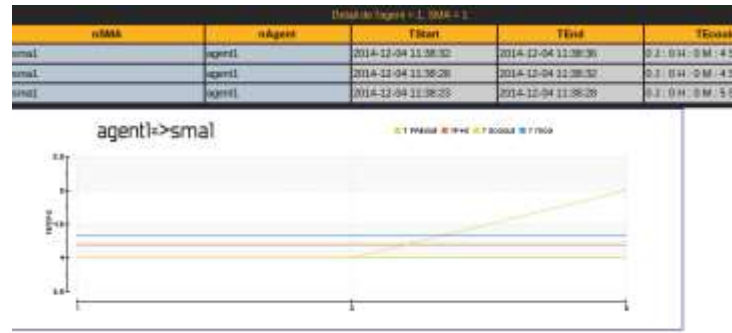


Fig 11 : measures of the agent 1

The agent 2 is four times launched as shown in the following figure (Fig 12):

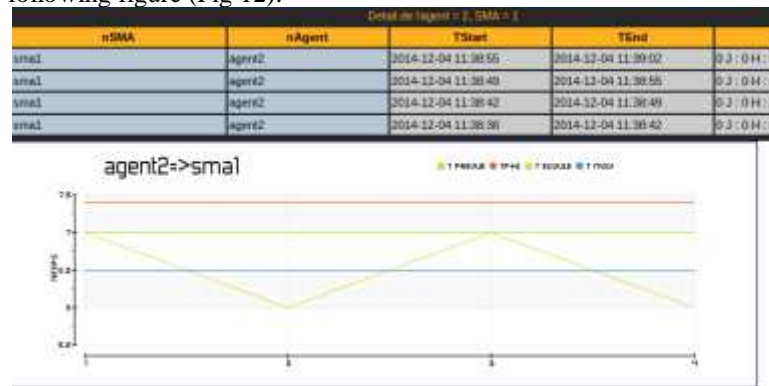


Fig 12 : measures of the agent 2

The agent 3 is two times launched as shown in the following figure (Fig 13):



Fig 13 : Measures of the agent 3

Rapport et Décision :

According to the decision-making report, we notice that the agent 1 (Agent go up) is not real-time because the average time of its executions is not situated between time planned and time planned more the term. On the contrary the agent 2 (turned right Agent) and the agent 3 (Agent go down) are real time as illustrated by figures: Fig 12 and Fig 13.

ID	Name	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7	Value 8
1
2
3
4
5
6
7
8
9
10

Fig. 14 : Report

CONCLUSION

In this paper, we propose a new distributed real-time approach "TIMER" who can communicate with any platforms with the multi- agent system to turn within different operating system.

This new approach can control several SMA (develop it c, c++ or Java) situated in different operating systems (Windows or Linux) and rejoiner on several machines.

This solution multi-platform has for role the check of the run time of every agent, to be able to make the decision such as the validation or not of a program.

The works described in this paper validate the functioning of our architecture of control of the time of the system SMA. The part makes out a will experimental treat an application realized within our laboratory with the robot Kepera, which consists in following a route by basing itself on her agents (agent rotation, left agent, right agent).





The solution "TIMER" was tested on the platform " M2M of the Robot Kepera III " consisted by reactive agents and cognitifs agents. this agents tests allowed us the check of the temporal aspects of the agents and by consequence the validation of program of this platform.

The first phase of this work being dedicated to the integration of the approach distributed "Timer" to the systems multi-agents formed by cognitive and reagent agents, we intend to spread the field of application of this approach to include the distributed systems multi-agents with agents says " hybrid, deliberate, " and to integrate the platform Timer as the module within Linux.

REFERENCES

- [1] A.Rahni, « Contributions à la Validation D'ordonnement Temps Réel en Présence de Transactions Sous Priorités Fixes et EDF » Rapport de thèse 2008-2009 LISI/ENSEM, Universités de Poitiers.
- [2] M.Saqalli, F.Qrichi Aniba, H.Medromi. « Développement d'une Plateforme Temps Réel Pour les Systèmes Distribués à Base des Systemes Multi-agents : Application d'une Equipe de Football » Vèm Conférence Internationale en Recherche Opérationnelle Marrakech, 24-27 Mai CIRO '10
- [3] M.Saqalli, D.Raoui, H.Medromi. « Plateforme Temps Réel de Sécurité Resaux Pour Les Systèmes Distribués » WC2M ENSAM Workshop Codes, Cryptologie et leurs Mathématiques Les 23 , 24 Avril 2010 à l'ENSAM-Meknès
- [4] A. Sayouti, F. Moutaouakkil, H. Medromi. « The Interaction-Oriented Approach for Modeling and Implementing Multi-Agent Systems». International Review on Computers and Software (I.RE.CO.S), Vol. 5, N. 2, Mars 2010.
- [5] Hutzlerg, Klaudlh, Wangd, « Automates Temporisés et Systèmes Multi-Agents Temps Réels » Journées francophones sur les systèmes Multi-Agents Hermès Science, Paris 2004
- [6] L. Zaffalon. Programmation concurrente et temps réel. Presses Polytechniques et Universitaires Romandes(PPUR), 2007.
- [7] J. Xu and D. Parnas. On satifying timing constraints in hard-real-time systems. Software Engineering Notes,16(4):132-46, December 1991.
- [8] P. Chevochot and I. Puaut. An approach for fault-tolerance in hard-time distributed systems. Proc. 18th IEEE Symposium on Reliable Distributed Systems (SRDS'99)(short paper), Lausanne, Switzerland, pages 292–293, October 1999.
- [9] E. Grolleau. Ordonnement temps réel hors-ligne optimal à l'aide de réseaux de Petri en environnement monoprocasseur et multiprocasseur. Thèse, ENSMA-Université Poitiers, 1999.
- [10] <http://www.windriver.com>
- [11] <http://www.qnx.com>
- [12] M.Saqalli,R.Elbejjat, H.Medromi « Nouvelle Solution Timer Temps Réel Intégré aux Systèmes Distribués Multi Agents »
- [13] Driss Raoui, Siham Benhadou, Hicham Medromi. « New Distributed Methodology for Intrusion Detection Based on Multi-agents System at Two Levels». International Review on Computers and Software (I.RE.CO.S), Vol. 5. n. 3, pp. 298-302

Author Bibliography

	<p>Athor Meryem SAQALLI Received her engineering degree in Networks and Telecommunications in 2006, from the National School of Applied sciences, Tanger. In 2008s he joined the System Architecture Team of the ENSEM School, Casablanca, Morocco. His current main research interests embedded systems Based on Multi-agents Systems</p>
	<p>Athor Said Benaissa received his Degree in High Education Deepened in Telecommunication Networks in 2008, from the National Institute of Science and Technology, Marrakech. In 2009 he joined the System Architecture Team of the ENSEM School, Casablanca, Morocco. His current main research interests embedded Systems Based on Multi-agents Systems.</p>
	<p>Athor Hicham Medromi received his PhD in engineering science from the Sophia Antipolis University in 1996, Nice, France. He is responsible of the system architecture team of the ENSEM Hassan II University, Casablanca, Morocco. His current main research interests concern Control Architecture of Mobile Systems Based on Multi Agents Systems. Since 2003 he is a full professor for automatic productic and computer sciences at the ENSEM School, Hassan II University, Casablanca.</p>
	<p>Athor Saida Tallal Received her PhD from the University of Poitiers in France in 1990, and she is currently Research Professor of Computer Science at ENSEM, Hassan II University in Casablanca, Morocco. She is PhD and her current research focuses on Multi Agent Systems in the Systems Architecture Team related to LISER Laboratory.</p>