New Approach Timer for the Control of the Distributed Systems with the Multi-Agents Systems

Saqalli Meryem
Computer Science
System Architecture Team,
Computing, Systems and
Renewable Energy
Laboratory

Medromi Hicham
Computer Science
System Architecture Team,
Computing, Systems and
Renewable Energy
Laboratory

ABSTRACT

The real-time computing is widely used and gets more and more importance in the everyday life. Several fields of application are concerned, in particular, the automobile, the avionics, the multimedia, the energy, etc. The real-time systems concern IT applications having a role of follow-up or control of process in an environment which evolves dynamically. In this article we propose a new approach called "Timer" for the distributed multi-agents architectures. This approach is validated and implemented on the platform of security and detection of intrusion. The objective of this Timer approach is to verify the capacity of every agent to perceive its environment and to elaborate an answer in the required time.

General Terms

Internet and Distributed Computer Systems.

Keywords

Platform, Distributed System, Multi-Agents Systems, Real Time System. Control.

1. 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 (part) of the specification of the implanted system) [1].

2. MULTI-AGENT APPROACH

2.1 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].

2.2 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].

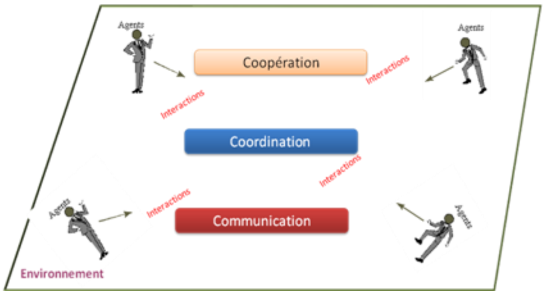

Fig1: Principle of a Multi agents System

3. 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, 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].

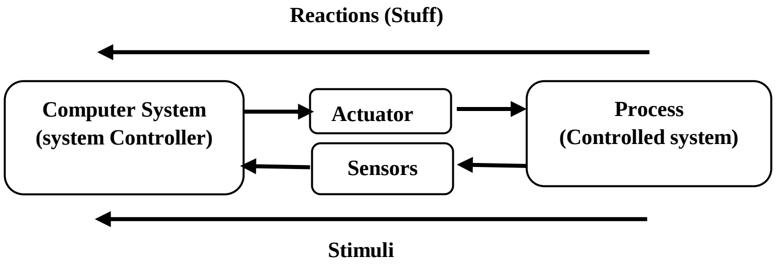

Fig2: Real Time System

3.1 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.

3.2 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 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, 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).

3.3 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.

3.4 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.

3.5 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.

4. TIMER APPROACH

4.1 Generalities

This need in validation of the data of temporal point of view brings to turn to solutions which allows the continuous control of the time of execution following the requirements of the various systems.

Our article 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].

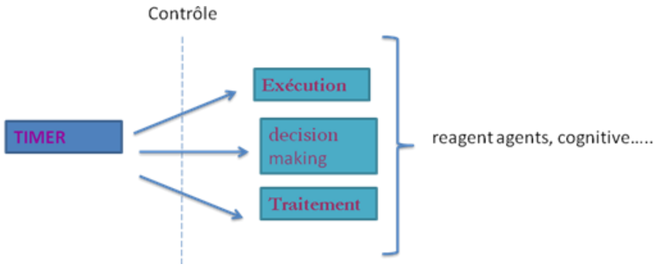

Fig 3: Principle of a Timer Approach

4.2 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.

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

4.3 Architecture to TIMER

4.3.1 Generalities

Some types of Multi- Agents System are concerned by our approach TIMER, among others the System of intrusion's detection. Where from the interest to finalize 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.

4.3.2 Functional Requirements

The role of the TIMER is to control the time, its existence within the Multi- Agents 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
- The decision-making from information feedback.
- Etc.

4.3.3 proposed real-time distributed Architecture

We propose architecture consisting of four modules:

- Multi- agents System: this module is a software or hardware solution which answers a given need.

- System operational, constitutes the environment of execution of the SMA (eg. Robot, android, 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 binds (connects) all these entities (SMA, operational System and real world) and which has for objective the control of the processing time of the SMA and the operational system with regard to the constraints of the real world.

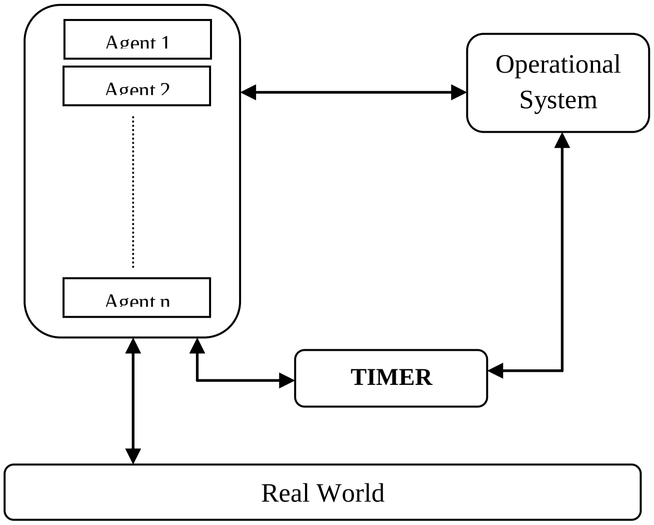

Fig 4: Real-time Distributed architecture

4.3.4 Structure of the 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 figure 5 following one contains in his turn several modules.
The proposed architecture consists of one:

- Modulate temporal database of the 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 time of executions. 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 measure is conditioned by the execution of an agent. For that purpose, the role of this module is to be listening the Multi- Agent System to detect every agent coming to run.

- Module of time measure of execution

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

- 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 database.
At the end of the check, the information on the processing time of every executed agent is registered in a 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.

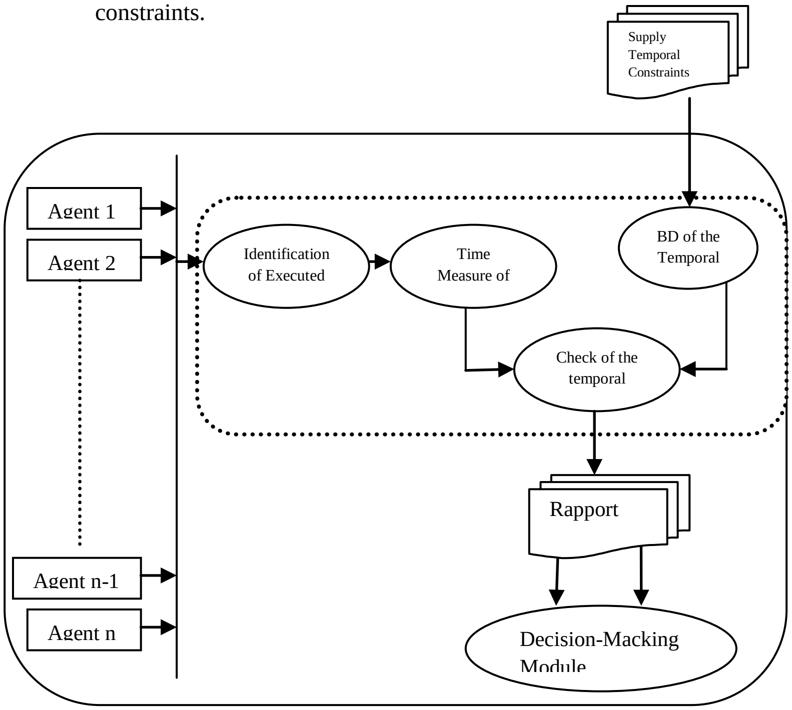

Fig 5: Structure of proposed Timer

4.3.5 Implementation of TIMER-SMA

The approach " Timer " is the solution proposed to control the time of execution, treatment or decision-making of the "reactive and cognitive" agents forming the application taken as a case study " Detection of intrusion " [3].
The reactive 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 reactive agent in the following way (Fig.6 [3]).

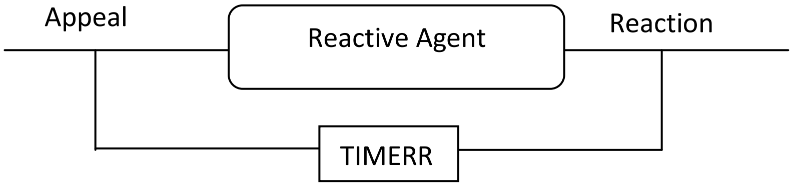

Fig 6: Timer for the reactive agent

On contrary to the reactive 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.7 [10].

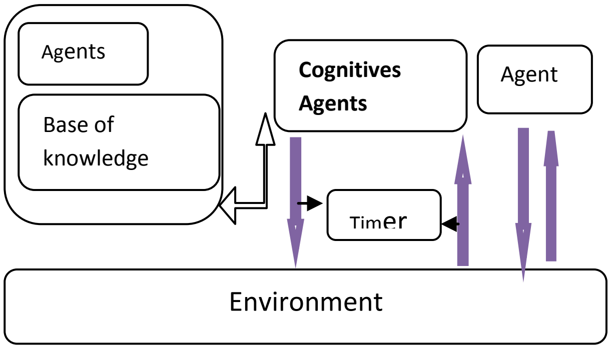

Fig 7: Timer for the cognitive agent

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.

5. APPLICATION: DETECTION OF INTRUSION

The platform of security and detection of intrusions distributed, real-time based on the multi-agents system (SMA), is represented in the figure 8.

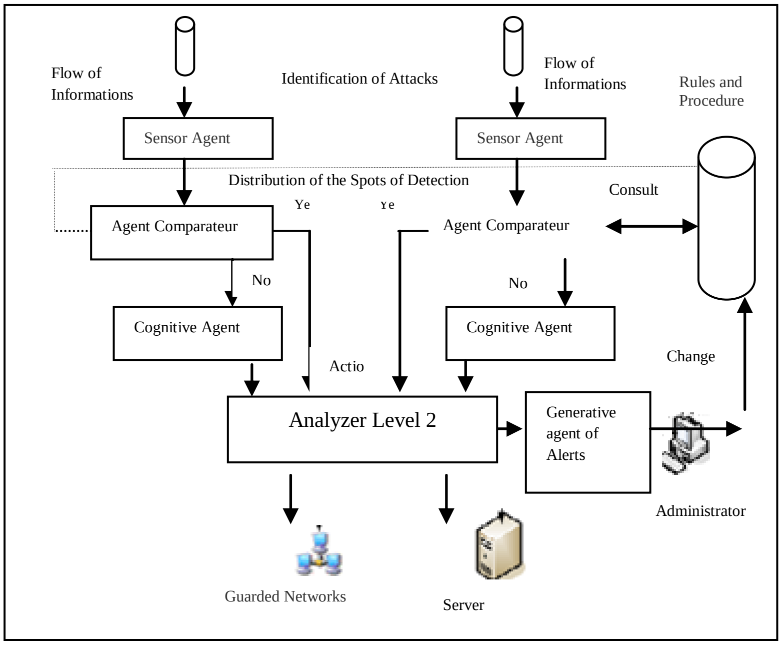

Fig 8: Plan of the platform of detection of intrusions

This model of security and detection of intrusion based on a distributed approach uses the multi-agents system to benefit from the intelligence of every agent. This model is formed on one hand by agents having the reactive capacities to react quickly against the known attacks and on the other hand of agents having the cognitive capacities to detect also, the unknown attacks [11].

The reactive agents (comparator agent and decision-maker agent), have for role to analyze the flow of information collected by the sensor agent. The agent "comparator" is in charge of confronting the traffic captured with the base of signatures. According to the degree of threat that the intervention can represent, the agent "decision-maker" is going to block or to let the traffic continue his road towards the cognitive agent.

The cognitive agents are in charge of the study of the behavior of the watched target. They are going to make a more detailed analysis of the data and to decide afterward to block or not the traffic and to generate a relevant alert.

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

- The reactive agents and the cognitive agents constitute the SMA
- The watched target presents the operational part.
- The viruses and Spam constitute the constraints of the real world
- Timer is the solution proposed for the temporal validation of the system of detection of intrusion.

6. CONCLUSIONS

In this article, we propose a new distributed real-time approach "TIMER" who can become integrated into any platforms with the multi- agent system.

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

The solution "TIMER" was tested on the platform "Security and Detection Of Intrusion" consisted by reactive agents and cognitive ones. 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 multi-agents systems formed by cognitive agent and reactive one, we intend to spread the field of applications of this approach to include the distributed multi-agents systems with agents " hybrid, deliberate, ", as well as to develop and to detail of advantage the link between the block "Timer" and the block real World.

7. REFERENCES

[1] A.Rahni, « Contributions à la Validation D’ordonnancement 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 Hérmès 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. Ordonnancement temps réel hors-ligne optimal à l’aide de réseaux de Petri en environnement monoprocesseur et multiprocesseur. Thèse, ENSMA-Université Poitiers, 1999.

[10] M.Saqalli, R.Elbejjet, H.Medromi « Nouvelle Solution Timer Temps Réel Intégré aux Systèmes Distribués Multi Agents »

[11] 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