How to build Real-Time Multi-Agent Systems using Anytime Techniques

Claude Duvallet and Bruno Sadeg
LIH, Faculté des Sciences et Techniques
25, Rue Philippe Lebon
76058 Le Havre Cedex, France
{duvallet, sadeg}@univ-lehavre.fr

Alain Cardon
LIH, IUT du Havre
Place Robert Schuman
76610 Le Havre Cedex, France
cardon@iut.univ-lehavre.fr

Abstract

Anytime techniques are techniques which allow some algorithms to provide useful results at anytime. These techniques seem to be a very interesting approach to use in a lot of applications where partial or inaccurate results, obtained early, may be more useful than complete or accurate results obtained late. This is the case of systems which manage emergency situations or systems used to manage currently technological applications like electronic market place and other application based on internet. Moreover, Multi-Agent Systems (MAS) are a promised approach to manage complex and distributed systems. Adding anytime techniques to MAS could lead to obtain power systems in order to manage complex and real-time aspects of whatever application. In this paper, we present the ANYMAS (ANYtime Multi-Agent System) model, a new approach for real-time multi-agent systems based on anytime algorithms.

1 Introduction

In the area of aid based decision systems and systems used to manage efficiently applications where technological or economical risks may occur, we must be able to respond timely to abnormal situations (that is to avoid risks or to limit their consequences as earlier as possible). Moreover, in a lot of situations managed by these applications, partial or inaccurate results obtained early are more useful than complete or accurate results obtained late. Such systems could be Multi-Agent Systems (MAS) [11] where are added real-time aspects [7] in order to obtain Real-Time Multi-Agent Systems(RT-MAS). The applications where RT-MAS may be an appropriated approach are market place management, multimedia systems, technological risks management and so on. For example, in an application of market place management, a multi-agent system can be used to manage different information systems in order to extract informations in a constrained time.

In order to integrate real-time aspects in MAS, anytime approach [3] seems to be the best approach. Indeed, anytime techniques deal with partial and/or progressive responses. In this paper, we describe in section 2 classical real-time systems and anytime techniques. We describe in section 3, ANYMAS model (ANYtime Multi-Agent System). In section 4, we give an example of ANYMAS implementation and we conclude by describing ANYMAS future extensions.

2 Related work

Real-time multi-agent systems is a new domain of research which merge multi-agent systems and real-time systems. These two domains are usually studied separatly whereas, we neede to study them together in order deal with the complex and time-constrained current applications. In this section, we begin to present real-time systems and anytime techniques, before presenting related work on RT-MAS.

2.1 Real-Time Systems and Anytime techniques

Real-time systems are systems which are submitted to temporal constraints (deadlines, periods,...). They must be able to react timely facing the environment which they control. Indeed, in many computer systems tasks must be executed before certain deadlines. These tasks are called real-time tasks. However, classical real-time systems [10] are not suited to manage some real-time applications where approximate responses obtained timely may be more useful than complete responses obtained late. So, a new approach, called anytime [12], has appeared which allow to build progressive solutions. Indeed, anytime techniques allow the system to provide useful partial results at any time.

2.2 Real-Time Multi-Agent System

Anytime approach is used to take into account real-time in MAS [6][9] because in a lot of applications managed by MAS the timeliness of a result is privileged, and the quality of the result may be altered. Indeed, in applications like market management or process control, we have not always enough time to wait to get a complete result in order to take a decision. This kind of applications are hard real-time applications. In classical hard real-time applications, wost case execution times of tasks are needed. However, in applications based on multi-agent systems, it is difficult to know execution times; then anytime techniques are often used to design real-time multi-agent systems. In the next section, we present ANYMAS, a MAS model where anytime approach is used to implement real-time aspects in MAS.

3 ANYMAS model

ANYMAS architecture is composed of anytime agents which belong partially to a component-based architectures [4] where some improvements are done. We add an introspection component which allows agents to predict the time needed to execute tasks. This component may modify dynamically previous predictions if necessary (if there is not enough time to terminate execution). A task being composed of subtasks, an agent's celerity compute subtasks necessary time during a MAS training mode. To this purpose, agents can create temporal coordination agents, called reification of a temporal coordination agent, used to reduce communications between agents. This agent manage a group of agents. According to the available time, it might be interesting to influence the behavior of agents so that the system can provide useful partial results. Therefore, in ANYMAS, we encourage some groups of agents to get this result as priority and to discard other groups of agents. In summary, to control duration of system's execution, it is necessary to control interactions between groups of agents. This control is done by temporal coordination agents.

3.1 Anytime agent

Anytime agent permits to obtain partial result at agent level. In an anytime agent, we call agent's celerity its capacity and its speed to build a result. In ANYMAS model, an Augmented Transition Network (ATN) is used to allow the system to stabilize agents in different states [4]. We consider two modes of execution: training mode and working mode. During training mode, agent's speed is constantly computed in order to get an average value of successive speeds, whereas in working mode it only uses these values to compute the necessary-time to execute the next action. These average time values are discretized in a unit of time at the end of this step. This construction is made in the following way: given an ATN with n states (see figure 1, where n is set to 5), $d_{i,i+1}$ $(1 \leq i < n-1)$ is the transition duration between state i and state i+1; Let a variable k be a subdivision of the time and $n_{i,i+1}$ the number of k subdivisions between states i and i+1.

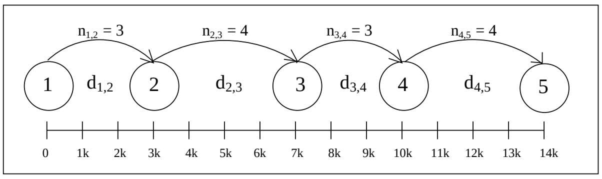

Figure 1: Example of discretization of time in ATN

Therefore, we obtain the following equation:

$$d_{i,i+1} = k * n_{i,i+1}, \text{ where i=1,2,3,...} \Rightarrow \text{k} = \frac{d_{i,i+1}}{n_{i,i+1}}$$

The number of subdivisions for the first interval being set, an algorithm (see figure 2) allows to determine (1) the value of the k variable and (2) the time subdivisions between states. ε is a variable providing a degree of time precision allowed. k value represents the computed unit value which measures the transition-time between two successive states of the ATN. This algorithm is implemented in JAVA language. Simulations we have done [2] have shown that its behavior is compliant to our expectations, that is it determines the optimal value of the K variable. In summary, based on slack time's execution, the algorithm evaluate the agent's celerity in order to determine whether or not the execution will be terminated beforea deadline. This outlines the ANYMAS predictive aspect.

3.2 Temporal coordination agent

Reification agents are produced by agents that have a component used to reify temporal coordination agents. A temporal coordination agent component acts like a threshold function: it is triggered when the

```
n_{1,2} ← 0
n ← number of state
k ← 0
i ← 1
WHILE (|d_{i,i+1} - k * n_{i,i+1}| > ε)
        AND i < n-1)
        i ← 1
        n_{1,2} ← n_{1,2} + 1
        k = round (d_{i,i+1} / n_{i,i+1})
        WHILE (|d_{i,i+1} - k * n_{i,i+1}| < ε
                AND i < n-1)
                i ← i + 1
                d_{i,i+1} = d_{i,i+1} + d_{i-1,i} - k * n_{i-1,i}
                n_{i,i+1} = round (d_{i,i+1} / k)
        END WHILE
END WHILE
```

Figure 2: Algorithm of discretization of time

number of agents or the number of communications between them exceed a certain threshold. When the threshold is exceeded, it is necessary to synchronize agents concerned by the communication exchanges. The objective is to verify if it is possible or not to connect these agents to an existing temporal coordination agent, that has yet controled other communications. If such a temporal coordination agent do not exist then a new temporal coordination agent is reified and its acquaintances are initialized with the two agent's identity, which have triggered this reification [1]. The temporal coordination agent scrutinizes the state of a small group of agents and controls permanently their communications. It requests them about their celerity and the number of exchanged communications. They have to take into account the global deadline and must cooperate with other temporal coordination agents in order to negotiate the agent group importance. That is done according to priority of the tasks the agent is currently doing. When it becomes necessary, a temporal coordination agent may decide to reduce the communicative activity of his group of agents. Reducing the activity of certain groups of agents allows other groups to reach an exploiting result more quickly. Then a progressive solution can be constructed.

4 An example of ANYMAS application

In order to validate ANYMAS model, an application of Market Place management is used (cf figure 3). In an electronic marketplace, actors use systems to buy and sell products. When a seller wants to sell a product, he proposes his product to the other actors of the market place. In order to decide if he must buy a product, a buyer need to have some informations, i.e., the current product price on the market, its quality, etc. To help the buyer in taking his decision, we have designed a market place manager system whose goal is to extract the information needed by the buyer to take the good decision. The concurrency of the other actors require the buyer to take the decision as earlier possible even if he does not have the whole information needed to take the best decision. So, the RT-MAS must be able to give him a maximum amount of information in a minimum amount of time.

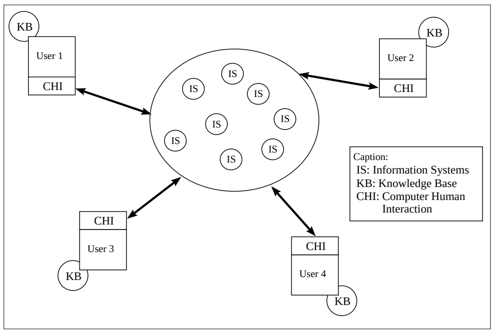

Figure 3: Multi-users marketplace management

In the application we implemented, we use several Information Systems where are stored informations about products, their quality, their price, informations about actors, etc. The system who manage this application is based on agents which interact together in order to extract, analyze and filter needed informations which are located in the Information Systems. This application was implemented using the MadKit platform [5]. In order to implement ANYMAS model, we have had to increase functionnalities of the agents of the MadKit platform: an ATN, a clock and a function of discretization (see figure 4), are added to the initial structure of the agent. The instropection component was implemented with both an ATN and a disretisation function.

Our application need to be distributed on several machines. Therefore, we use CORBA norm [8] to distribute it thanks to the ORB ORBacus. The implementation langage we have used is the same than that it is used in the MadKit platform: java.

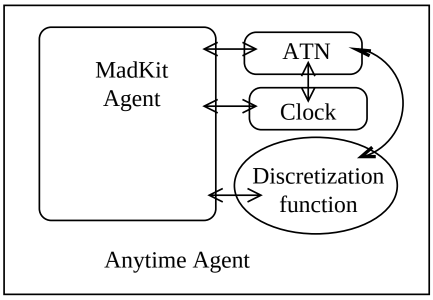

Figure 4: Construction of an anytime agent with MadKit

5 Conclusion and future work

This paper deals with the problem of introducing anytime techniques in multi-agent sytems which allow to obtain partial or incuurate results at anytime. Anytime techniques allow to take into account real-time aspects in multi-agent systems. Therefore, we used this approach to design a model called ANYMAS (ANYtime Multi-Agent System). We have chosen to apply our model on an application of Electronic Market Place where we use a MAS to extract and filter information from complex Information Systems. This application has allowed to validate a part of ANYMAS model. An extended version of the Marketplace is being currently implemented in the context of a system used to help users to make their transactions (sell and buy products) in the best conditions. We are currently working on distributed aspects of this application and of ANYMAS model.

Our perspective is to implement ANYMAS model in hard real-time application, that is in sirens'manager system of a town. Indeed, we are extending ANYMAS in order to take into account emergence aspects in anytime component of multiagent systems and to make intelligent and friend interface to the users. Other aspects need to be detailled or completed (anytime query manager, RT-MAS design methodologies, time constraints specification, ...).

References

[1] A. Cardon and F. Lesage. Toward adaptative information systems: considering concern and intentionality. In *Proceedings of KAW'98*, May 1998.

[2] C. Duvallet, B. Sadeg, and A. Cardon. Real-time in multi-agents systems. In *Proceeding of CAINE'99, Atlanta*, pages 212–215, 1999.

[3] J. Grass and S. Zilberstein. Programming with anytime algorithms. In *Proceedings of the 14th Internationnal Joint Conference on Artificial Intelligence*, pages 22–27, Montreal, 1995.

[4] Z. Guessoum and J.-P. Briot. From concurrent objects to autonomous agents. In *8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Ronneby, 1997.

[5] O. Gutknecht and J. Ferber. Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems. Technical report, LIRMM, UMR 9928, Université de Montpellier II, 1997.

[6] A.I. Mouaddib, F. Charpillet, Y. Gong, and J.P. Haton. Real-time specialist society. In *Proceedings IEEE, Singapore International Conference on Intelligent Control and Instrumentation*, pages 751–754, 1992.

[7] M. Occello, Y. Demazeau, and C. Baeijs. Designing organized agents for cooperation with real time constraints. In *1st International Workshop on Collective Robotics*, pages 25–37, Paris, 1998.

[8] OMG. The common object request broker: Architecture and specification. In *OMG TC Document 95-7-X*, July 1995.

[9] T. Salvant and S. Brunessaux. Multi-agent based time-critical decision support for c31 systems. In *Practical Applications of Intelligent Agents and Multi-Agents, PAAM*, London, 1997, 1997.

[10] J.A. Stankovic and K. Ramamritham. *Hard real-time systems: a tutorial.* IEEE Computer Society Press, 1988.

[11] M. Wooldridge and N.R. Jennings. Agent theories, architectures and language : A survey. In M. Wooldridge and N. Jennings, editors, *Intelligent Agents, ECAI 1994*, volume LNAI 890, pages 1–32. Springer Verlag, 1994.

[12] S. Zilberstein. Resource-bounded sensing and planning in autonomous systems. *Autonomous Robots*, 3:31–48, 1996.