Distributed Electronic Commerce Transactions thanks to Anytime Agents

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

Abstract

Electronic commerce are inherently distributed. They must allow actors (sellers and buyers) to trade products thanks to computer systems. These systems must be designed so as to manage distributed information systems and to allow timely negotiation between the actors. That is why decisions must be taken timely in spite of the lack of information. Futhermore, these sytems use a lot of information which may be complex and difficult to analyze. So, a promizing approach is to use a multiagent system in which anytime techniques are added. In this paper, we present ANYMAS, a model of real-time multiagent systems applied to the judicious extraction of information from distributed information systems.

Keywords: MultiAgents Systems, Real-Time Systems, Anytime Algorithms, Electronic Commerce.

1 Introduction

In a variety of applications, computer systems are used to help the decision-maker. These systems are often based on information systems which are used to extract some information about a current situation. For example, in an electronic marketplace, if we want to buy or to sell some products, we need to gather some information about the state of the market, the product stock and so on. We need also some information about the actor who proposes the transaction. All this information must be available in a simple manner in order to permit to the decision-maker to understand the situation quickly. The management of marketplaces (electronic commerce) is complex. A promized approach to manage them is to use multiagent systems [13] where are added real-time aspects in order to deal with the needs of the market, that is to take decisions timely about selling or buying something. These real-time aspects are taken into account by techniques called *anytime techniques*. Indeed, anytime techniques allow to provide query results at any time. For example, the analysis of the criteria to determine the price of a product can take a day, a week, etc. According to each analysis duration, the result is differrent: more we have time to analyze the information, more the result is accurate.

The design of anytime MultiAgent System (MAS) is not so easy because we must take into account real-time apsect at both the agent level and the multiagent level. Some works has been done about anytime techniques [1, 7] [3] but few works have been done about anytime multiagent systems [11] [8]. In this paper, after a brief description in section 2 of the related works, we describe in section 3, ANYMAS model (ANYtime Multi-Agent System), a model we design to take into account real-time aspects in MAS. In section 4, we illustrate ANYMAS implementation in the area of electronic commerce and we conclude by describing ANYMAS future extensions.

2 Anytime MultiAgent Systems

Real-time multiagent systems is a new domain of research which merge multiagent systems and real-time systems. These two domains are usually studied separatly whereas it is needed to study them together in order to build an efficient decision aid-based system, notably in the domain of electronic marketplace management. In this section, we begin to present some features of real-time systems and anytime techniques. Then, we give state of the art of real-time multiagent systems.

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 [12]. Indeed, in many computer systems tasks must be executed before certain deadlines. These tasks are called real-time tasks. One can classify these tasks according to the type of deadlines that they must respect:

- soft deadlines: if a deadline is not met, the task can still continue to execute but the quality of service (QoS) offered by the system is deteriored,
- firm deadlines: if a deadline is missed, the task associated becomes useless for the system, and it is aborted,
- hard or critical deadlines: missing a deadline leads to catastrophic consequences.

We note that transactions in electronic commerce belongs to the two last categories according to the economical consequences led by the transaction deadline missing. Classical real-time systems[12] are not suited to manage some real-time applications where approximate responses obtained timely may be more useful than complete responses obtained late [2]. For example, in a market management application, a buyer or a seller must take a decision as earlier as possible even if he/she has not the complete information. So, a new approach, called anytime [14] [6], has appeared which allow to build progressive solutions according to the time allowed. Indeed, anytime techniques allow the system to provide useful partial results at any time.

2.2 Real-time in MultiAgent Systems

Anytime approach is widely used to take into account real-time in MAS [11] [2] because in a lot of applications managed by MAS the timeliness of a result is privileged, whereas 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 for a complete result in order to take a decision. This kind of applications can be qualified as hard real-time applications. In classical hard real-time applications, we need to know the worst case execution time if we want to design it. In applications based on multiagent systems, it is difficult to know the execution's time of a task. That is why we have chosen anytime techniques to design our system.

Some works exist about real-time multiagent systems using anytime techniques. Salvant and al. [11] designed an anytime agent architecture composed of four components: tasks monitor, domain tasks, real-time clock and computing resources. However, the real-time aspect is not taken into account at the system level.

Occello and al. [9] use an architecture based on a blackboard system in order to build real-time agent. This aproach seems to be too restrictive in comparison with possibilities of agent architectures [13].

Most of current works are concentrated on real-time implementation at agent level. The model we present, ANYMAS, deals with implementation both at the agent level and at the interaction one and at the organization one.

In the next section, we present ANYMAS, a MAS model where anytime approach is used to implement real-time aspects in order to manage applications where partial results obtained timely may be more useful that complete results obtained late, i.e., decisions taking in electronic commerce [2].

3 ANYMAS model

ANYMAS architecture is composed of anytime agents which belong partially to a component-based architecture [4] where some improvements are done. We add an introspection component to the initial agent architecture in order to allow 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. To this purpose, agents can create temporal coordination agents, called *reification of a temporal coordination agent*, used to reduce communications between agents since the MAS execution is tightly dependent on the communications between agents. Futhermore, 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 discard some groups of agents and we encourage other groups that are well-suited to get an exploitable result. In summary, to control the duration of system's execution, it is necessary to control the interactions between groups of agents. This control is done by temporal coordination agents.

3.1 Anytime technics

The term "anytime" was coined by Dean and al. in the late 1980s [1] in the context of his work on time-dependant planning. Anytime algorithms are algorithms whose quality of results improve gradually as computational time increases, hence they offer a tradeoff between resource consumption and output quality. This technique are particuliarly appropriated in the context of artificial intelligence in order to take into account the real-time aspect of computer-aid based systems. The quality of a result is represented by performance profils as it is shown in figure 1 where x axis represent the time and y axis represent the quality of the result.

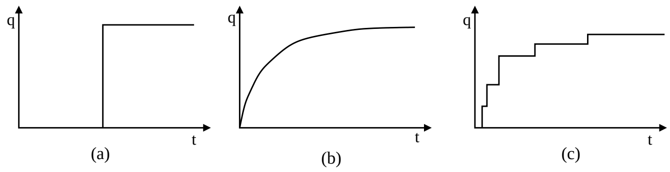

Figure 1: Typical performance profile : (a) classical algorithm (b) Idealized anytime algorithm (c) Actual anytime algorithm

We apply this technique to the design of agents and multiagents systems in order to design an Anytime MultiAgent System.

3.2 The proposed model

ANYMAS model is composed of anytime agents based on an Augmented Transition Network (ATN) which is used to allow the system to stabilize agents in different states. Each ATN's state corresponds to a step of the anytime algorithm (executed by the anytime agent) where an exploitable partial result may be provided. An anytime agent execute a transaction with initial global deadline, which is then subdivided into steps where partial results may be provided. In an anytime agent, we define agent's celerity as its capacity and its speed to build a result. These notions allow to measure agent performances (see section 4).

In the multiagent system where ANYMAS is implemented, 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 transaction. These average time values computed during training mode are discretized in a unit of time at the end of this mode. This construction is made in the following way:

Given an ATN with n states (see figure 2, 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.

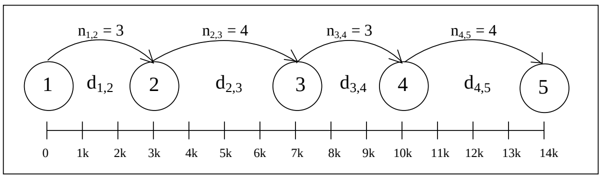

Figure 2: Example of discretization of time in ANYMAS

Therefore, we obtain the following equation:

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

For example, in figure 2, we obtain:

$$k = \frac{d_{1,2}}{n_{1,2}} = \frac{d_{2,3}}{n_{2,3}} = \frac{d_{3,4}}{n_{3,4}} = \frac{d_{4,5}}{n_{4,5}}$$

The number of subdivisions for the first interval being set, an algorithm (see figure 3) 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 measure the transition time between two successive states of the ATN.

```
n_{1,2} ← 1
Last State ← false
WHILE ((|d_{i,i+1} - k * n_{i,i+1}| > ε)
        AND ((i+1) Not Last State))
        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+1) Not Last State))
                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 3: Algorithm of discretization of time

The algorithm was implemented in Java. Simulation results (see figure 4) illustrates how the algorithm determine the K value (during the training mode) which is used as a unit of time during execution mode in order to go from step i (where a partial result is available) to step i+1 (where a better partial result is available) (see figure 5). These experimentations show that the algorithm behavior is compliant to our expectations. In summary, based on slack time's execution, the goal of the algorithm is to evaluate agent's celerity in order to determine whether or not the execution will be terminated before deadline. This outlines the ANYMAS predictive aspect.

State i	State i+1	Time in ms	Time in K units
1	2	10556 ms	2640 K
2	3	20173 ms	5043 K
3	4	5443 ms	1361 K
4	5	34256 ms	8564 K
5	6	29654 ms	7413 K
Value of K : 4			

Figure 4: Simulation of the algorithms

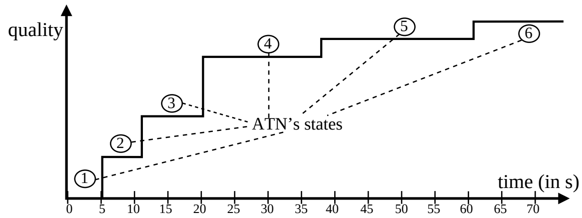

Figure 5: ANYMAS and profil performance

4 Anytime extraction of information in a marketplace

In order to validate ANYMAS model, an application of marketplace management is used (cf figure 6). In an electronic marketplace, actors use computer system to buy and sell objects, say firms. When a seller wants to sell a firm, he proposes his price to the other actors of the electronic marketplace. In order to know the moment to buy the firm, a buyer need to have some information about the firm (for example its number of employees, its economic situation) and about the seller (seriousness, ...). To help the buyer to take his/her decision, we have designed a computer system (electronic marketplace) which has 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 earlier as possible even if he/she does not have the whole information needed to take the best decision. So, the computer system must be able to give him/her the maximum amount of information as possible in a minimum amount of time.

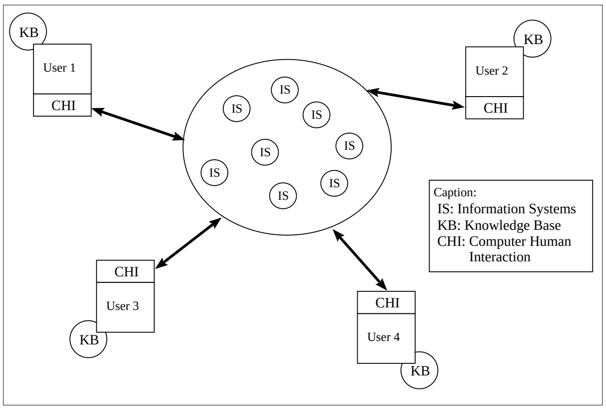

Figure 6: Multi-users marketplace management

In our electronic marketplace (see figure 6), we use several information systems to store information about firms (employees, financial aspects, position on the marketplace, geographical situation), and information about actors, in the form of Oracle databases. Users use a computer system (which is based on ANYMAS model) which communicate with the distributed information systems through the JDBC (Java DataBase Connectivity) interface. Moreover, our computer system is based on a multi-agent system in order to extract, analyze and filter the information which are available in the information systems. This application was implemented using the MadKit platform [5] where we have had to increase functionnalities of the agents. As we can see in the figure 7, an ATN, a clock and a function of discretization are added to the initial structure of the agent. We note the the instropection component was implemented with an ATN and a disretisation function.

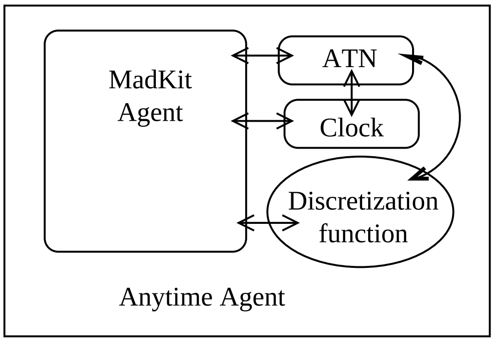

Figure 7: Construction of an anytime agent with MadKit

The anytime agents are used to extract information from distributed information systems. More the time allowed to the query, more one precise the results or more quantities of information we can retrieve. The anytime agents of ANYMAS are used to extract intelligently information from distributed information systems.

5 Conclusion and future work

In this paper, we have presented ANYMAS model designed to take into account real-time aspects in multiagent systems. This model based on anytime techniques allows to construct real-time complex applications. We have chosen to implement ANYMAS model an electronic marketplace application. This application has allowed to validate a part of the 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.

Our perspective is to implement ANYMAS model in hard real-time application [12], that is, in sirens'manager system of a town (Le Havre, France). Indeed, ANYMAS need to be completed in order to take into account emergence aspects in anytime component of multiagent systems. Other aspects need to be detailled or completed (anytime query manager, RT-MAS design methodologies, time constraints specification, ...).

An other aspect of our work is to use distributed multiagent systems in order to build a computer aid based system working on either wired or wireless networks. Therefore, in wireline networks we can use the CORBA norm [10] to distribute anymas multiagent system. In wireless networks, one may use system's agents with adapted anytime algorithms in order to aleviate the problem of frequent connexions and deconnexions of the network. Indeed, each agent must define steps where it may provide useful partial results according to the frequency of the network disconnections.

References

[1] T. Dean and M. Boddy. Solving time-dependant planning problems. In *Proceedings of the 7^{th} National Conference of Artificial Intelligence*, pages 419–454, Minneapolis, Minnesota, 1988.

[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 14^{th} 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 *The 8^{th} 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] P. Haddawy. Focusing attention in anytime decision-theoretic planning. *SIGART Bulletin*, 7(2), 1996.

[7] E. Horvitz. Reasonning about beliefs and actions under computational resource constraints. In *Proceedings of the Workshop on Uncertainty in Artificial Intelligence*, Seattle, Washington, 1987.

[8] A. Mouaddib. Anytime coordination for progressive planning agents. In *Proceedings of the 16^{th} National Conference on Artificial Intelligence and 11^{th} Conference on on Innovative Applications of Artificial Intelligence*, pages 564–569, Orlando, Florida, 1999. AAAI Press / The MIT Press.

[9] M. Occello and Y. Demazeau. Building real time agent using parallel blackboards and its use for mobile robotics. In *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, 1994.

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

[11] 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.

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

[13] M. Wooldridge and N. 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.

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