USING COGNITION AND LEARNING TO IMPROVE AGENTS' REACTIONS[*]

Pedro Rafael Graça and Graça Gaspar
Department of Computer Science
Faculty of Sciences of the University of Lisbon
Bloco C5 - Piso 1 - Campo Grande, 1700 Lisboa, Portugal
E-mail: prafael@di.fc.ul.pt, gg@di.fc.ul.pt

Abstract

This paper proposes an agent-architecture to deal with real-time problems where it is important both to react to constant changes in the state of the environment and to recognize the generic tendencies in the sequence of those changes. Reactivity must satisfy the need for immediate answers; cognition will enable the perception of medium and long time variations, allowing decisions that lead to an improved reactivity. Agents are able to evolve through an instance-based learning mechanism fed by the cognition process that allows them to improve their performance as they accumulate experience. Progressively, they learn to relate their ways of reacting (reaction strategies) with the general state of the environment. Using a simulation workbench that sets a distributed communication problem, different tests are made in an effort to evaluate: the utility of the multi-agent system architecture and the importance of the individual features of agents, the utility of using a set of different strategies, and the significance of the learning mechanism. The resulting conclusions point out the most significant aspects of the generic model adopted, helping to put it in perspective as a solution for other problems.

1 Introduction

1.1 Motivation

Reaction, cognition and the ability to learn are among the most fundamental aspects of human behaviour. Daily, we react to a non-deterministic and constantly changing world, often facing unknown situations that nevertheless need immediate answer (for example, crossing an unknown street for the first time); we constantly rely on our cognitive ability to classify the surrounding environment (for example, choosing the best place to cross the street); we use our experience to select actions for specific situations (for example, quickly crossing the street when the sign turns red). Generally, cognition and the ability to learn lead to the accumulation of experience, allowing better decisions that improve the selection of actions. This is the central idea of the agent-architecture proposed in this paper: the agents have a reaction module that allows them to answer in real-time, a cognition module that successively captures and classifies images of the environment, and a learning module that accumulates experience that progressively allows a better selection of actions.

[*] This work was supported by the LabMAC unit of FCT

1.2 Environment

A real-time group communication simulated environment (reproduced by a simulation workbench) supported the development and study of the proposed architecture. Prone to non-deterministic and continuous changes (traffic fluctuations), such an environment emphasizes the need for immediate reactions. At the same time, the cyclic occurrence of similar environment states (for example, periods with low traffic level) and the repetition of changing patterns (for example, brisk increases of traffic level) point to the utility of a cognitive system that enables a form of learning, allowing the use of accumulated experience.

In this distributed communication environment, where the traffic level is variable and messages can be lost, each agent is responsible for sending and eventually resending (when losses occur) a set of messages. Each agent's goal is to optimise the timeout interval for resending lost messages, in such a way that the sending task is completed as soon as possible and the communication system is not overloaded by unnecessary resending. The agent chooses from a set of tuning strategies that over time it learns to relate to the state of the communication system, progressively improving its performance.

1.3 Related work

In the context of concrete applications of multi-agent systems in traditional telecommunication problems, our main goal is to put in perspective the relationships between the generic problems observed in a distributed application and the generic answers that a multi-agent system is able to offer, in an abstraction level more concerned with properties than with detail. Even though no studies similar to ours were found (involving reaction, cognition and machine learning in a telecommunication problem), many other investigations in the field of multi-agent systems and telecommunications address problems in real-time environments that share many of the essential characteristics. A wide diversity of studies address problems such as routing, traffic congestion, scalability, fault location, and cooperative work, to mention only a few. Work on this area can be found in (Albayrak, 1999) and (Hayzelden et al, 1999).

(Mavromichalis and Vouros, 2000) and (Malec, 2000) propose layered agent-architectures to address the problem of controlling and balancing reactivity and deliberation in dynamic environments requiring real-time responsiveness. These perspectives show some similarities to our work, but they don't incorporate a machine learning mechanism.

(Weiβ, 2000) discusses the relationship between learning, planning and reacting, proposing an extension to a single-agent architectural framework to improve multi-agent coordination. The learning mechanism is used in order to determine the best way of alternating between reaction-based and plan-based coordination. In this particular, our study is significantly different: our learning mechanism concerns how to react in environments where reaction is a necessity rather than an option.

Work concerning a learning approach in some regards close to ours can be found in (Prasad and Lesser 1999). They propose a system that dynamically configures societies of agents, using cognition and/or communication as the basis for learning specific-situation coordination strategies.

2 A group communication problem

2.1 Motivation

The communication problem used in this investigation was idealized following two main requirements:

- the preservation of the generic properties of a real-time distributed application;
- the avoidance of complex situations that could make the interpretation of results a more difficult task.

To meet the first requirement, we selected a group communication problem, a typical and intuitive real-time distributed situation, offering a high degree of versatility concerning the desired complexity level involved. To meet the second requirement, we used a simulation workbench that reproduced the selected problem, simplifying accessory aspects and preserving all the essential properties that ensure the accuracy and expressiveness of the results.

As a good example of the benefits introduced by the simplifications that took place, it is considered that, although normal messages can be lost in the communication process, acknowledgments cannot. Since from the message sender point of view both of these losses are equivalent and undistinguishable, the occurrence of acknowledgment losses would increase complexity without enriching the study or its results.

2.2 Description

The problem in question was conceived in order to serve a generic purpose, but the description of a real and specific situation will help to clarify its outlines. Imagine a team of stockbrokers, each of them working on a different stock market. Suppose that, in order to coordinate the team's global action, there is a synchronization rule that establishes that each individual can only perform an action after being informed of every other team member's intention. Consider that it is important to perform as many operations as possible and that the communication between stockbrokers takes place in a telecommunication system where the traffic level is variable and messages can be lost. This scenario corresponds to the distributed communication problem adopted in this investigation.

Each agent is associated to a communication node, having the responsibility of sending and resending (when losses occur) a message to each other node. When a message arrives to its destination, an acknowledgment is sent back to the message sender. In each communication *season*, the users (each associated to a node) exchange messages with each other. One season ends when the last message (the one that takes more time to successfully arrive) reaches its destination.

The traffic level on the communication network varies over time, influencing the reliability: an increase of traffic level causes a decrease of reliability, increasing the occurrence of message losses; a decrease of traffic level has the opposite effect. The better the agents are able to adapt to the sequence of changes, the more accurate becomes the chosen instant for resending lost messages. Increased accuracy improves the communication times, causing the duration of seasons to decrease.

It is important to notice that the communication problem described takes place at application level. In environments where the sequence of events is very fast (imagine a millisecond time scale) the ability for reacting very quickly is often more important than the ability for choosing a good reaction. The time needed to make a good choice can actually be more expensive than a fast, even if worse, decision. Because of this, the agent-architecture proposed in this paper is better suited for problems where the time scale of the sequence of events justifies efforts such as cognition or learning. This does not mean that quick answers to the environment are not possible: deliberation (recognising, learning and deciding) can easily become a background task, only showing its influence on the quality of reactions when there is enough time to identify environmental states and use previously acquired experience. On the worst case (millisecond time scale) this influence will tend to be null and agents will react without deliberating. The better the deliberation process can accompany the sequence of events, the greater will this influence be.

3 Agent-architecture

3.1 Introduction

Considering the communication problem adopted, the agents' task is to tune the timeout interval for resending lost messages, so that the duration of the communication seasons and the occurrence of unnecessary resending are both minimised. In their tuning task, agents must deal with information at different time scale perspectives: they must immediately react to constant variations in the state of the environment and also be able to recognize tendencies in the sequence of those variations so that a learning mechanism can be used to take advantage of accumulated experience.

To react to constant variations, each agent uses one of several *tuning strategies* at its disposal. To evaluate the quality of a tuning strategy in a communication context (for example, during a low traffic period) the period during which that strategy is followed cannot be too short; to allow the search for better strategies, this period should not last too long. These opposing desires led to the introduction of the *satisfaction level* concept, a varying parameter that regulates the probability of a strategy change decision (the lower the agent's satisfaction level is, the more likely it will decide to adopt a new strategy). As it will be briefly explained below, this satisfaction level depends on two additional aspects:

- the detection of changes in the state of the environment (communication conditions);
- the self-evaluation of the agents' performance.

To recognize non-immediate tendencies in the sequence of environment state changes, the agent uses its cognition system. The information collected in each communication season is gathered in a memorization structure. This structure is periodically analysed in order to abstract from the details of basic states, fusing sequences of those basic states into generic states classified in *communication classes* and detecting important variations in the state of the communication system (for example, a transition from a traffic increase tendency to a traffic decrease tendency). The result of this analysis influences the satisfaction level; for example, a change of the communication class, or the detection of an important variation,

can cause the satisfaction level to decrease, motivating the agent to choose a new tuning strategy (possibly fitter to the new conditions).
Progressively, agents learn to relate the tuning strategies and the communication classes. The establishment of this relationship depends on two classification processes: the classification of the agents' performance and the classification of the generic state of the environment (the communication classes processed by the cognitive system).

A scoring method was developed in order to classify the agents' performance. As the duration of the timeout interval depends on the tuning strategy in use, the qualification of an agent's performance in a sequence of seasons is a measure of the fitness of the strategy used to the communication class observed during those seasons.

The diversity of states of the environment emphasizes the utility of studying a multi-agent system where different individuals may have different characteristics. While an *optimism level* regulates the belief in message losses (the more pessimistic the agent is, the sooner it tends to conclude that a message was lost), a *dynamism level* regulates the resistance to stimulation (the less dynamic the agent is, the less it reacts to changes, the longer it keeps using the same strategy). Each different agent has a specific behaviour and interprets the surrounding environment in a different way.

In this section, after introducing some terminology (subsection 3.2), the details of the proposed agent-architecture are presented in the following order:

- the tuning strategies (subsection 3.3);
- the scoring method (subsection 3.4);
- the cognitive system (subsection 3.5);
- the learning system (subsection 3.6).

Finally, a diagram (subsection 3.7) and an illustration of an operating agent (subsection 3.8) give a global perspective of the architecture.

A more detailed description of this agent-architecture can be found in (Graça, 2000).

3.2 Terminology

The period of time needed to successfully send a message (including eventual resending) and to receive its acknowledgement is called *total communication time*. When a message successfully reaches its destination at the first try, the *communication time* is equal to the total communication time; otherwise, it is the time elapsed between the last (and successful) resending and the reception of the acknowledgement.

The ending instant of the timeout interval is called *resending instant*. It is considered that the *ideal resending instant* of a message (the instant that optimises the delay) is equal to the communication time of that message.
The difference between the resending instant and the ideal resending instant is called *distance to the optimum instant*.

A high increase or decrease of traffic level immediately followed by, respectively, a high decrease or increase is called a *jump*. A high increase or decrease of traffic level immediately followed by stabilization is called a *step*.

3.3 Tuning strategies

To follow the fluctuations of the communication system, each agent constantly (every communication season) adjusts the resending instant. It is possible to imagine several ways of making this adjustment: following the previous communication time, following the average of the latest communication times, accompanying a tendency observed in the succession of communication times, etc. A *tuning strategy* is a way of adjusting the resending instant. It is a simple function whose arguments include the communication times observed on previous seasons and the optimism level, and whose image is the resending instant to be used on the following season.

A set of ten tuning strategies is available to the agents, including for example: a *reactive* strategy (according to this strategy, the communication time observed in season *t* is used as resending instant in season *t+1*), an *average* strategy (as shown in figure 1, each resending instant is defined according to the average of all previous communication times), a *semi-reactive average* strategy (the last communication time is weighted by one tenth in the average calculus), an *almost reactive average* strategy (as shown in figure 2, the last communication time is weighted by one third in the average calculus), a *reactive ignoring jumps* strategy (works like the reactive strategy but keeps the same resending instant when jumps occur). A *TCP* strategy, reproducing the real procedure adopted by the TCP/IP protocol (see (Peterson and Davie, 1997) for details), was also included in this set. According to this strategy (figure 3) the more unstable the communication conditions are, the bigger is the safety margin used (higher resending instants).

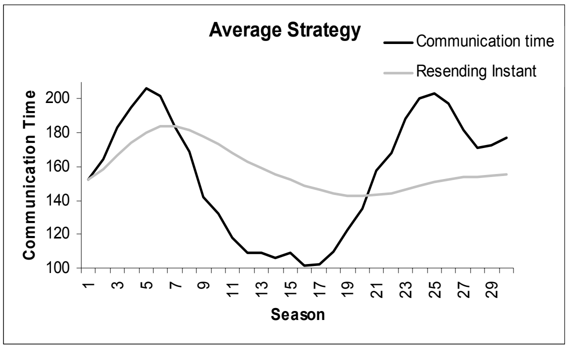

Figure 1: Average tuning strategy

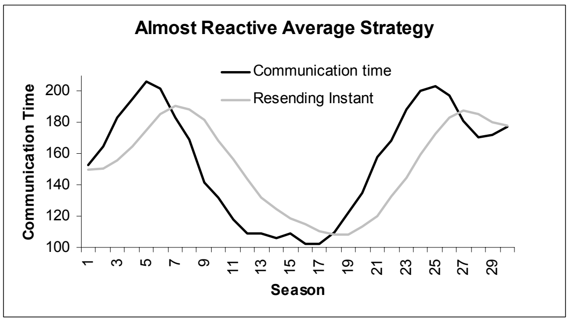

Figure 2: Almost reactive average tuning strategy

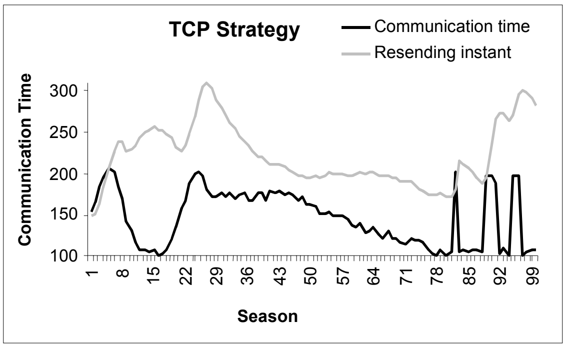

Figure 3: TCP tuning strategy

It is expected that the diversity of strategies helps to match the diversity of environmental states: different tuning strategies will be differently fitted to the different communication classes. For example, when traffic fluctuations are unpredictable, a reactive strategy will probably produce better results than an average-based strategy; the opposite will probably occur if the traffic level tendency is to vary within a sufficiently well determined interval. The goal of the learning system is precisely to select for each communication class the strategies that produce better results.

As mentioned before, the optimism level regulates an agent's belief in message losses: the more pessimistic the agent is, the sooner it tends to conclude that a message was lost, the sooner it tends to resend it. Agents with different optimism levels use the same tuning strategy differently: based on the same unmodified resending instant given by the strategy, a pessimistic agent shortens the delay and an optimistic agent widens it (figure 4 shows this effect).

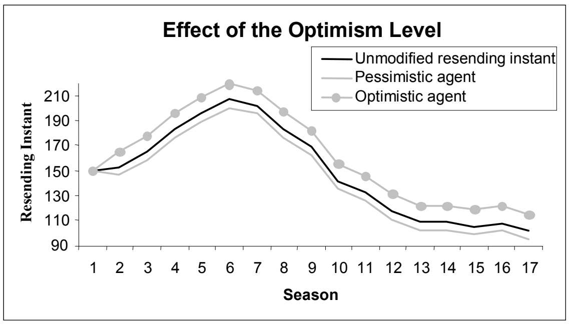

Figure 4: Effect of the optimism level on strategies

This differentiation (along with the dynamism level) opens the possibility of establishing relations between certain types of agents (for example, highly pessimistic) and certain communication conditions (for example, constant traffic level).

3.4 Performance evaluation

The evaluation of the agents' performance has three main goals:

- to measure the significance of different individual features: it may be possible to determine a measure of the fitness of certain agents' characteristics to the communication conditions, if a specific type of agent (for example, very optimistic) tends to receive higher or lower scores under those conditions;
- to allow the agents to adjust their procedure: poor performance causes a decrease of the satisfaction level, and eventually leads to a change of strategy;
- to support the classification of training examples: this will be detailed further ahead, in the learning system subsection.

At the end of each season, each agent has information that allows it to qualify its performance. The score obtained helps each individual to answer the following question: how accurate were the resending instants chosen? As it will be detailed in the learning system subsection, the successive scores will help to answer another question: how fitted is the chosen strategy to the communication class currently observed?

The *main* performance evaluation score is obtained considering the sum of distances to the optimum[1] instants, adding a penalty for each message unnecessarily resent (the lower the score, the better the performance).

To clarify this system, consider the following example:

- agent A sends 3 messages during a communication season, to agents B (*mAB*), C (*mAC*) and D (*mAD*);
- the resending instant for *mAB* is 150 time units;
- the resending instant for *mAC* is 180 time units;
- the resending instant for *mAD* is 140 time units;
- the acknowledgement from *mAB* arrives after 146 time units;
- the acknowledgement from *mAC* arrives after 183 time units (an unnecessary resend occurred);
- the acknowledgement from *mAD* arrives after 270 time units (this means the first message wasn't acknowledged, and the second one's acknowledgement arrived after 130 time units).

Given this situation, agent A would get:

- 4 points for *mAB* (150-146);
- 6 points for *mAC* (183-180=3, penalty for unnecessary resend[2]: 3*2=6);
- 10 points for *mAD* (140-130);

Agent A's final score is 20 points. For example if the resending instant for *mAB* were 155 time units (further away from the optimum instant by 5 time units), the score would have been 25 points (worse than 20).

[1] The concept of optimum instant was presented at section 3.2.

[2] The penalty for unnecessary resend equals the distance to the optimum instant: the larger the distance the bigger the penalty.

Two additional *auxiliary* scoring mechanisms are also used in order to evaluate the agents' performance[3]. Each of these mechanisms ranks the agents on each season (from first to last place), according to each of the following criteria:

- the necessary time to complete the sending task (the best one is the first to receive all acknowledgements);
- the quality of the chosen resending instants (same criteria as the main method described above).

At the end of each season, each agent is scored according to its rank (the first of *n* agents gets 1 point, the second 2 points, and so on until the last that gets *n* points).

The information about every agent's performance is gathered and processed at the end of each season in order to establish the rank. On a simulation workbench this is a trivial task because all the information is locally available. On a real situation, a way of gathering the information and broadcasting the results would have to be studied.

The purpose for these auxiliary mechanisms is to allow the agents to compare each other's performance. When the communication conditions are unstable the resending instants are more difficult to set and, although the agent's performance may be good (considering the complexity involved), the main score (determined in an absolute way) will tend to be lower. In these cases, the auxiliary scores (determined in a relative way) can help each agent to correctly evaluate the quality of its performance.

3.5 Cognitive system

The information memorized after each season is continuously analysed and processed in order to provide the agent with an image of the world. The memorization structure, more than just an information repository, is a fundamental part of the cognitive system; among other information (arguments for the tuning strategies), it stores:

- a *short memory block*: contains each ten consecutive average communication times and the average performance score during those ten seasons (state of the environment in the last few seasons);
- a *global memory block*: the result of a process of synthesis of past short memory blocks (state of the environment during a wider period).

Every ten seasons, a short memory block is processed in order to obtain information that is then added to the global memory block. This synthesised information includes: a set of parameters that characterize the traffic oscillation (for example, how many increases of traffic level were observed during the ten seasons), the communication class observed, and the average performance score.

A *communication class* is a generic classification of the state of the environment. Such a classification is determined from the parameters that characterize the traffic conditions (obtained from each short memory block), and has three dimensions: the *traffic level* (high, medium, low, variable), the *traffic variation tendency* (increase, decrease, constant, alternating, undetermined) and the *sudden variations occurrence* (jumps, steps, jumps and steps, none).

[3] These auxiliary scoring mechanisms weren't considered in the classification of training examples, but they also influence the satisfaction level.

The detection of variations in the communication system is based on the following principle: the greater the difference between the global communication class (the communication class of the global memory block) and the communication classes of the last short memory blocks, the more likely it is that a significant variation is occurring. A metric of three-dimensional distance between communication classes was developed in order to apply this idea (considering, for example, that the difference between a high and a medium traffic level is smaller than the difference between a high and a low traffic level). The distance between two communication classes is obtained by adding the distances between each dimension's members.

The detection of variations causes the satisfaction level to progressively decrease, motivating the agent to choose a new tuning strategy more adequate to the new communication class. When a variation is first detected, the decrease of satisfaction is generally small; in this way, if the variation is merely transitory its impact will be minimal. However, if the variation is progressively confirmed, the decrease in the satisfaction level is continuously accentuated: the more obvious and significant the variation is, the larger becomes the satisfaction level decrease.

3.6 Learning system

The agents must be able to adapt to the conditions of the communication system, selecting the best strategies for each communication class. This requirement appeals for a learning mechanism that builds up and uses accumulated experience. When its performance isn't satisfactory (bad score), an agent must learn that the strategy used in the current communication class is probably not adequate. If the performance is good, the agent must learn the opposite.

The learning mechanism is based on the following cyclic process associated to internal state transitions:

- Finalization of the previous state (a new training example is stored);
- Prevision of a communication class and selection of a tuning strategy for the next state;
- Beginning of the new state.

When the satisfaction level is low (bad performance and/or variation detected), an agent may decide to finalize its current state. The *dynamism* level associated to each agent makes it possible for two agents to act differently when facing the same conditions: a more dynamic individual is more likely to feel dissatisfied and will consequently change its state more often then a more conservative individual.

An agent's state (from the learning mechanism perspective) consists of a sequence of communication seasons, characterized by a global communication class, a strategy in use, and a performance score. When an agent decides to end the current state, this characterization is used to build a new *training example*, a triple <*communication class, tuning strategy, performance score*>. The training examples are stored into a two-dimensional table (the *experience table*) that contains the average score for each pair <communication class, strategy>, recalculated every time a correspondent training example is added. The more training examples (concerning a particular communication class) an agent stores, the higher is its *experience level* (in that class).

This form of case based learning has some specific characteristics: the learning cases are processed instead of being stored (only necessary information is preserved); it admits an initially void base of cases; it is dynamic, in the sense that new learning cases cause previous information to be progressively adjusted (recalculation of the average score). This learning mechanism has therefore some resemblance to a simple way of reinforcement learning[4].

Before initiating a new state, the agent must predict a communication class and choose a new strategy. The prediction of a communication class for the next state is based on the following complementary ideas: if the state transition was mainly motivated by bad performance, the communication class remains the same; if it was motivated by the detection of variations, then those variations are analysed in order to predict a new communication class. This analysis considers the latest three short memory blocks, using their data to determine a new communication class (assuming that the lately detected variations characterize the new state of the communication environment). If transition patterns were to be found in the sequence of states, this prediction process could be enhanced by the use of another learning mechanism that were able to progressively determine which transitions were more likely to occur.

To select a new strategy an agent may consult the experience table (selecting the best strategy according to the predicted communication class), choose randomly[5] (when it has no previous experience or when it decides to explore new alternatives), or consult a shared blackboard (described ahead). The random selection of strategies is the preferred option when the agent has a low experience level, being progressively abandoned when the experience level increases (even when experience is very high, a small probabilistic margin allows random selections).

The shared *blackboard* is used as a simple communication method for sharing knowledge between agents. Every ten seasons, the most successful agents (those who receive better scores) use it to register some useful information (strategy in use and communication class detected), allowing others to use it in their benefit. When the agents do not use this communication mechanism, learning is an individual effort; if it is used in exclusivity (as the only way to select a new strategy), learning does not occur[6]. When it is optional, it enables a simple way of multi-agent learning: by sharing their knowledge, the agents allow others to benefit from their effort, eventually leading each other to better solutions earlier than it would happen otherwise.

When a new state begins, the agent's memory (short and global blocks) is initialised. If, in a short period of time (first thirty seasons), the predicted communication class proves to be a wrong prevision (because the analysis was wrong or because the conditions changed), the

[4] Our initial idea was indeed to use reinforcement learning, but a more careful study revealed incompatibilities between this learning mechanism and the addressed problem: on the state transition process, there is no necessary relationship between the action an agent selects (new tuning strategy), and the following state (that includes the communication class, which is generally not influenced by the agent's choice); moreover, a state is only identified at the moment of its conclusion. These facts oppose the most basic principles of reinforcement learning.

[5] Random selection could be replaced by another specific distribution (such as Boltzmann distribution).

[6] An agent whose only method of strategy selection is consulting the blackboard is considered opportunistic: he develops no learning effort and always relies on other agents' work.

agent may choose to interrupt the new state to correct it. In this case, regarding the interrupted state, no training example is considered.

3.7 Agent-architecture diagram

The diagram shown in figure 5 summarizes the agent-architecture presented in this paper.

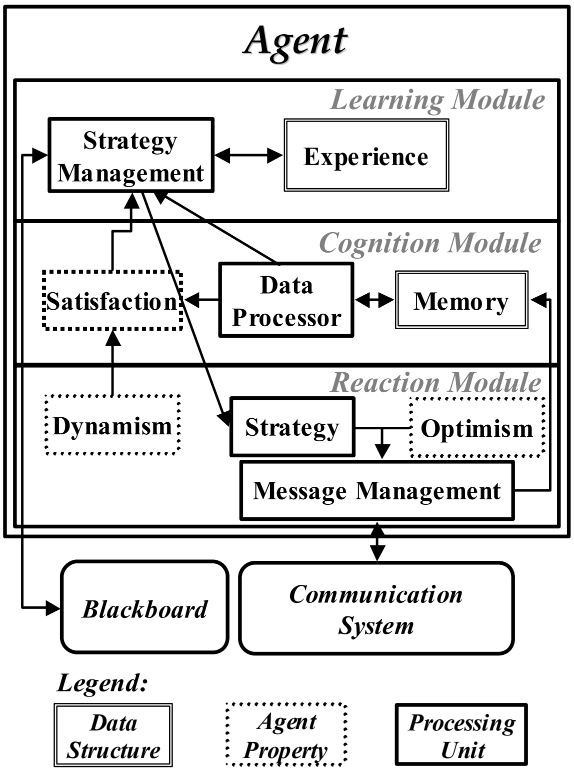

Figure 5: Agent-architecture diagram

The *reaction module* includes a message management unit, responsible for sending and resending messages according to the strategy in use (modified by the optimism level). The strategy is changed when the strategy management unit produces such a decision.

The data processor unit included in the *cognition module* is responsible for analysing the information that is constantly memorized, evaluating the environment (communication classes) and the performance (score). Conditioned by the dynamism level, the result of this analysis influences the satisfaction level. Whenever a state transition occurs, the data processor sends necessary information to the learning module so that a new training example can be created and stored.

The strategy management unit included in the *learning module* is responsible for the state transition decision (conditioned by the satisfaction level) and its related operations. It stores the training examples in the experience data unit.

3.8 Learning with agent Adam

The following description illustrates a sequence of operations of a specific agent during a learning state. This example can help to clarify the concepts presented along this section, showing how they are applied in a practical situation.

Agent Adam, a very optimistic and very dynamic individual, was using an average-based strategy to tune the resending instants. Since he is very optimistic, the resending instants are delayed accordingly (a less optimistic agent following the same strategy would resend earlier than Adam). The strategy was producing good results (he was receiving high scores) in a communication environment characterized by a low and constant traffic level, with no sudden variations; this communication class was being observed in the last 50 seasons (that far, the duration of the current state). Adam's satisfaction level was high and a change of strategy was not in consideration.

After ten more seasons had passed, the latest short memory block was analysed and some jumps were detected. Initially, the detection of this new condition caused only a small decrease of the satisfaction level; but when it was confirmed by two additional short memory blocks and reinforced by low scoring (the adopted strategy was now producing bad results), it motivated a quick decrease of the satisfaction level that led to a strategy change decision. If Adam were less dynamic, the decrease of the satisfaction level would have been slower and such decision would probably take a longer time to occur.

Following the strategy change decision, a new training example was stored, describing the good results of the previous strategy under the previous communication class (low and constant traffic level, with no sudden variations). If the same conditions were met in the future, this information could then be helpful for the selection of a tuning strategy.

Considering the latest short memory blocks, a new communication class was predicted: low and alternating traffic level, with jumps. Since Adam had no memory of operating under such conditions, he couldn't rely on previous experience to select a new tuning strategy. So, putting aside a random selection alternative, he decided to consult the blackboard. Understanding that Amy, a very successful agent that had been receiving the highest scores, had detected the same communication class, Adam selected the same tuning strategy that she was using, and then begun a new state.

Even if Amy's strategy were a bad choice for Adam, he would have in the future (when the same communication class were detected and a random selection of strategy decided) the opportunity for testing other strategies (explore other possibilities) and find out which one would serve him better in this situation. Moreover, he would (given enough opportunities) eventually select untested strategies even if a good strategy were already found (this would allow him to escape local minimums in local search). However, if Amy's strategy were a good choice for Adam, it would allow him not only to perform better but also to accelerate his learning effort (a good reference point in terms of tuning strategy would allow him to quickly discard worst alternatives).

4 Tests and results

4.1 Introduction

Using a simulation workbench for the group communication problem described, a significant number of tests were made. In this section we describe the most relevant of those tests and discuss their results. To support these tests, several traffic variation functions were created. Some reflect traffic variation patterns as they are found in real communication situations; others set interesting situations that help the evaluation of the different aspects of the agent-architecture.

Each simulation is composed by a sequence of one thousand communication seasons. Each test is composed by a sequence of five hundred simulations. The average of the agents' added distances to the optimum instants (from hereon referred simply as *distance*) is the value chosen to express the results.

4.2 Tuning strategies and cognitive system

The initial tests were focused on the tuning strategies. The original set of strategies was tested separately (no cognition or learning) under different traffic conditions. These tests led to the introduction of additional strategies (to cover important specific situations) and to an important (and expected) conclusion: different communication classes have different more adequate strategies.

The tests made to the cognitive system allowed its progressive refinement. In its final version, the system was able to classify the communication conditions in a very satisfactory manner. The possibility of capturing the essential aspects of a complex real-time environment in a classification system opened the door to the introduction of the learning mechanism.

4.3 Learning mechanism

The learning mechanism produced a clear and important result: the progressive decrease of the *distance*. The more complex the traffic variation function is (in other words, the greater the number of communication classes needed to capture the evolution of traffic conditions), the slower is this decrease.

In simple situations, where a single tuning strategy is highly adequate to the traffic function, the learning curve tends to approach the results that would be obtained if only that strategy was used[7] (figure 6). In more complex situations, when the diversity of the traffic variation function appeals to the alternate use of two or more strategies, the advantage of the learning mechanism becomes evident (figure 7).

[7] When we mention that only a single strategy is used, we mean that the same tuning procedure is kept throughout the simulations. In these cases, the learning mechanism remains inactive.

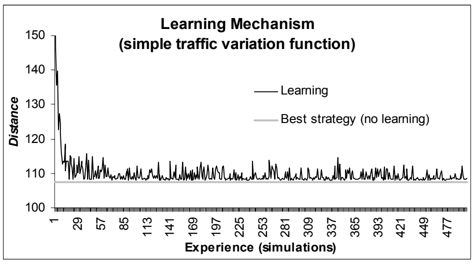

Figure 6: Learning in a simple traffic variation context

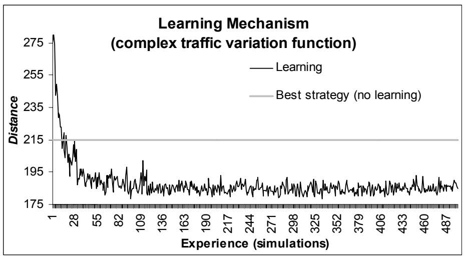

Figure 7: Learning in a complex traffic variation context

This last figure emphasizes an important result: the alternated use of different strategies can match the diversity of communication classes, producing, in some situations, better results than those produced by any single strategy available.

To confirm this result, special tests were made. Three different sets of available tuning strategies were considered in these tests: *set A* included the best strategy (the one producing better global results if used in exclusivity on the chosen context) and four other strategies (chosen randomly); *set B* included the five remaining strategies; a *full set* always included all ten. In each test, each of these sets was used separately and the respective results were compared. These tests showed clearly that, in some situations, the diversity of strategies is more important then the global fitness of any particular strategy. The *full set* often produced the best results (especially on complex traffic variation contexts) and, in some cases (as the one in figure 8), *set A* produced the worst results (even though it contained the best strategy).

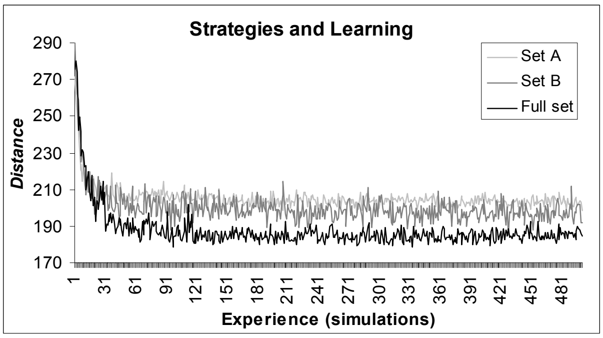

Figure 8: Comparing the performance of different sets of strategies on the same traffic variation context

From these results emerges the idea that, in certain traffic variation contexts, there is no single strategy that can be used to match the performance of alternating a set of different strategies.

4.4 Optimism and dynamism levels

Various tests showed that the optimism level could clearly influence the agents' performance. When the traffic level has a low variation or while it continuously decreases, pessimistic agents usually have a better performance; when the traffic level has a high variation or while it continuously increases, an optimistic posture tends to be better.

The next two figures show the results of testing five different sets of agents grouped by their optimism levels (all having neutral dynamism levels). Figure 9 refers to testing on a traffic variation context where the traffic level predominantly increases: periods of traffic increase are intercalated with abrupt and instantaneous traffic level decreases, producing a typical sawtoothed pattern. Since optimistic agents tend to delay their resending instants, they have better chances to avoid unnecessary resending under such conditions[8] and achieve a better performance.

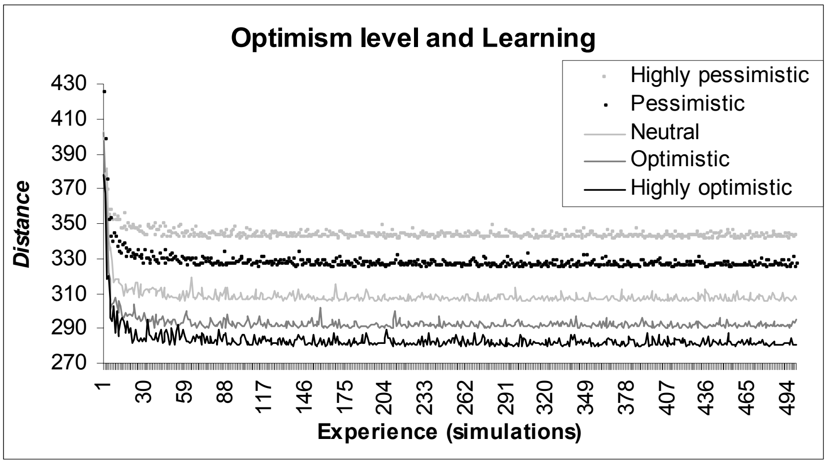

Figure 9: The influence of the optimism level (sawtooth traffic variation pattern)

When a traffic variation context in which there is no predominant variation is considered, high optimism or pessimism postures are usually not adjusted (figure 10).

[8] A pessimistic posture, according to which delays are believed to result from message losses, tends to anticipate the resend. Such a posture is generally penalized when the delay is a consequence of a traffic level increase. In these cases, it pays off to wait a little longer before resending (optimistic posture).

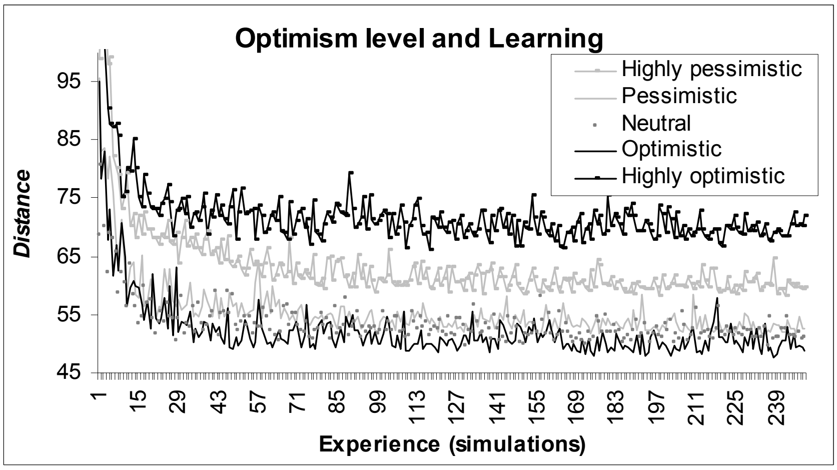

Figure 10: The influence of the optimism level (traffic context with no predominant variation)

As it is evidenced by figure 10, the relation between the optimism level and performance can be complex: although an optimistic posture produces the best results, a highly optimistic posture is worse than a highly pessimistic posture.

The effect of the dynamism level on the agents' performance wasn't made clear by the tests. It was observed that extremely dynamic agents had more difficulty in learning (they eventually achieved the same *distance* of others but took more time to do so).

4.5 Communication between agents

As described before, the agents may use a simple communication method (blackboard) as an alternative way of selecting a new strategy. To test and evaluate the impact of introducing this alternative, we considered two different sets of agents: agents that only used their individual experience table (no communication), and agents that alternated between their experience table and the blackboard. Results showed that the alternation between consulting the experience table and consulting the blackboard could improve the agents' performance (figure 11).

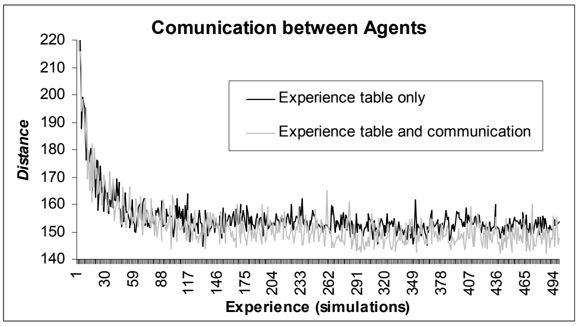

Figure 11: Learning with and without communication

Encouraged by this observation, new tests were made in order to compare the performance of a set of agents that learn without using the blackboard with the performance of an opportunistic agent that only uses the blackboard (only uses the experience of others and doesn't learn by itself). These tests showed that, regardless of the traffic function in use, the opportunistic agent's performance was never worse then the learning agents' performance. Moreover, when complex traffic variation functions were used, the opportunistic agent clearly beat the agents that were able to learn (figure 12).

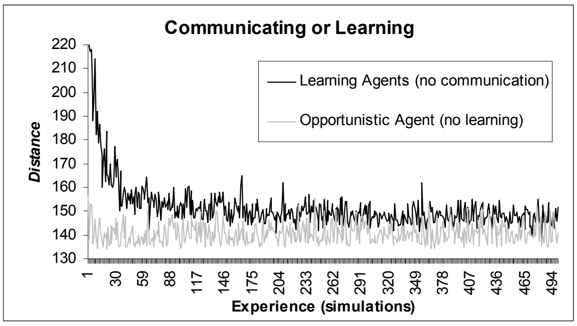

Figure 12: Performance of an opportunistic agent

It is important to notice that, in this case, the opportunistic agent achieves a clearly better performance since the first simulation, and reaches its best performance level within the first ten simulations. This is a clear sign that the learning task could be optimised, that the interactions between agents (namely the knowledge sharing) can improve global performance (even at early stages), and that having agents with different tasks or roles can benefit the collective effort and enhance the results.

5 Conclusions

The good results achieved with the proposed agent-architecture in a simulated group communication problem showed its adequacy to a real-time environment. Reacting through the use of different tuning strategies, classifying the environment through a cognitive system, and progressively determining the fitness of those strategies to specific communication classes, the agents were able to significantly improve their performance. Furthermore, in some situations, the alternation of strategies allowed by the learning mechanism achieved results that were clearly superior to those obtainable using a single strategy, leaving the idea that having an expressive variety of options can be a good way of addressing the complexity and dynamism of the environment.

The optimism and dynamism levels added versatility and adaptability to the agents. The optimism level revealed a special relevance, significantly influencing the agents' performance in various situations. The accurate characterization of these situations could motivate the online variation of this level, allowing increased adaptation to the environment.

The use of the blackboard as a knowledge sharing method improved overall performance. Furthermore, especially under complex traffic variation functions, opportunistic non-learning agents had better performance than learning non-communicating agents.

6 Final discussion

The success achieved by opportunistic agents indicates that it would be interesting and potentially useful to study the use of a mixed agent society in which the existence of different roles could lead to an improvement of collective performance. More than that, the members of this society could be exchanged according to their performance (from time to time, the worst active agents would be replaced) or even according to the collective experience (for example, under unknown conditions agents with the ability of learning would be preferred, but under well known conditions the opportunistic agents would become predominant).

The study of ways of coordination and interaction between agents to optimise the learning task is a promising field of development of the proposed architecture towards further improvement of global performance. The expressive results obtained with a simple communication mechanism suggest that an additional effort towards basic coordination could easily introduce a distributed learning perspective into the proposed model. This, along with the introduction of specific agent roles, could allow the reduction of the collective learning cost.

The generic properties of a multi-agent system successfully matched the generic problems found in a typical telecommunication problem, reinforcing the idea that the affinities between Distributed Systems and Distributed Artificial Intelligence justify further research. Globally, more than showing the utility of the proposed agent-architecture to the problem in question, the encouraging results indicate that the generic model is a potentially adequate solution for similar problems, namely for those where a real-time environment constantly demands immediate reactions and continuously appeals for cognition.

To help to put in perspective the generic aspects of the architecture, consider the following real-time communication problem. Imagine a multimedia conference where it is important that the participants keep visual contact with each other. During the conference, the video image frames are continuously transmitted on a communication system prone to traffic fluctuations. This problem is also concerned with the imperfection of the telecommunication system in a group communication situation. In this case it becomes important to set an adequate frame transmission rate so that the video image's quality is as good as possible (it is expected and possibly unavoidable that on high traffic situations this quality decreases, being advisable to decrease the frame transmission rate so that congestion doesn't worsen). To apply the proposed agent-architecture to this problem, a set of *transmission strategies* (for example, frame repetition strategies, frame skipping strategies, fixed transmission rate, etc.) and a method of performance evaluation (based on the quality of the video image) would have to be defined. Other than that, the essential aspects of the architecture would be easily applicable.

On a first glance, and as an example of a problem belonging to a different area of study (not centred on the communication process), our architecture seems to match the generic properties of the Robocup environment. Robocup sets a constantly changing environment that requires real-time responsiveness and, at the same time, strongly appeals for cognition. The alternative ways of reacting (running towards the ball, stopping, shooting at goal, passing, etc.) could be converted into strategies, and a learning mechanism could progressively determine their fitness to specific states of the environment. To determine the real extent of this simplistic and superficial analogy, further investigation is obviously required.

If reaction, cognition and the ability to learn are among the most fundamental aspects of human behaviour, they may well emerge as fundamental aspects of artificial agents that dwell on artificial worlds that become more and more similar to our own.

References

Albayrak , S. (Ed.) (1999). Intelligent agents for telecommunication applications. *Springer*.

Graça, P. R. (2000). Performance of evolutionary agents in a context of group communication. *M. Sc. thesis, Department of Computer Science of the University of Lisbon (in Portuguese).*

Hayzelden, A. L. G., Bigham, J., Wooldridge, M., Cuthbert, L. (Eds.) (1999). Software agents for future communication systems. *Springer.*

Malec, J. (2000). On augmenting reactivity with deliberation in a controlled manner. *In proceedings of the workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Fourteenth European Conference on Artificial Intelligence, Berlin. 89-100.*

Mavromichalis, V. K. and Vouros, G. (2000). ICAGENT: Balancing between reactivity and deliberation. *In proceedings of the workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, *Fourteenth European Conference on Artificial Intelligence, Berlin. 101-112.*

Peterson, L. L. and Davie, B. S. (1997). Computer networks: a systems approach. *Morgan Kaufmann Publishers.*

Prasad, M. V. N. and Lesser, V. R. (1999). Learning situation-specific coordination in cooperative multi-agent systems. *Autonomous Agents and Multi-Agent Systems,* 2:173-207.

Sutton, R. S. and Barto, A. G. (1998). Reinforcement learning: an introduction. *The MIT Press.*

Weiβ, G. (2000). An architectural framework for integrated multiagent planning, reacting, and learning. *In proceedings of the Seventh International Workshop on Agent Theories, Architectures, and Languages, Boston.*