Emotionally controlled inter-agent communication

Luís M. Botelho
Department of Information Sciences and Technologies
ADETTI/ISCTE
Tel: 351 21 7903906 Fax: 352 21 7903099
Luis.Botelho@iscte.pt

Pedro N. Ramos
Department of Information Sciences and Technologies
ADETTI/ISCTE
Tel: 351 21 7903906 Fax: 352 21 7903099
Pedro.Ramos@iscte.pt

Abstract

One of the roles played by emotion is to detect possible problems in human interaction and to signalise the need for an adaptive reaction. Similar problems affecting the interaction among artificial agents may be originated by the processes of receiving or sending messages, and may arise while the agent is receiving or sending messages. These problems became more accurate in dynamic and unpredictable environments, like in intelligent vehicle scenarios. These problems may happen even in drastically restricted environments in which there are only a small number of possible types of interactions. The paper shows that the preliminary theory of artificial emotions proposed by [Botelho and Coelho 2000] may provide the means required for detecting such interaction problems and signalling the need for an adaptive reaction. Finally the paper shows that the usual multi-processing, signalling, and inter-process communication capabilities provided by operating systems are enough to build an agent with the kinds of artificial emotions discussed and analysis the implementation of a generic platform for building those agents. The ideas discussed in the paper are exemplified in an intelligent vehicle scenario.

Keywords Inter-agent communication, vehicle-agent, appraisal, emotion, emotion-signal.

1 Introduction

This paper proposes that artificial emotions play an important control role in inter-agent communication, especially in dynamic and unpredictable environments such as in intelligent vehicle scenarios.

When autonomous artificial agents interact with each other in agent societies it is to be expected that some of the problems specific to the interaction in human societies may also take place. In this paper we analyse the role of emotion to detect communication-related problems and to signalise the possible need for an adaptive action both in natural and in artificial agents.

As a typical example of such a situation, consider the case in which a computer science teacher is fully absorbed trying to figure the best way to explain how computers may become conscious while his son suddenly appears and the following dialogue takes place:

- *Dad, will you give me a motorbike next Christmas?*
- *Yes. Now, go play with your sister... Wait a minute? What did you say? A motorbike? You must be out of your mind! I thought we had already discussed this motorbike issue.*

In the above conversation, the computer science teacher answers without fully understand his son's question, trying to focus on his thoughts. Then, some form of appraisal takes place and an emotion-signal is generated warning him that he should allocate some more cognitive resources to the processing of the question. The teacher then processes the question more carefully and reacts as described. Meanwhile his thoughts about computers and consciousness (temporarily) go away.

The role of emotion in this kind of situation was to detect that an important message has been received (without being fully processed) and to generate a signal that the person should pay attention to it. Whether and how the teacher reacts to that signal is not determined by the emotion-signal itself.

This kind of interaction control problem may also occur in inter-agent communication settings, such as an intelligent vehicle scenario in which an intelligent agent driving a vehicle communicates with other vehicle agents, with traffic information agents and with traffic sign agents. Suppose that, while a vehicle agent is trying to ask a yellow pages agent what is the name of a traffic monitoring system, it receives a message from another vehicle-agent reporting an accident. If this message is not processed in time, it is possible the vehicle agent gets in trouble. Following the preliminary theory of artificial emotion presented in [Botelho and Coelho 2000], one of the roles of emotion would be to detect the risk of undesirable situations like this and to generate an emotion-signal to warn the agent.

This paper presents a general platform for building emotionally intelligent agents according to that preliminary theory of emotion. It further describes the architecture of a vehicle agent built on top of that platform.

We have chosen the intelligent vehicle application to capitalise on work already done by the Modest European ACTS Project on a multi-agent system for video-based traffic surveillance [Botelho et al 1999]. The emotion component of the paper reflects exploratory work about emotions in the scope of the SAFIRA European IST Project.

Our work contrasts with the mainstream research on emotions in Autonomous Agents, which aims at developing believable agents [Bates et al 1992][Reilly 1995]. They use emotion with the purpose of creating the illusion of life in synthetic characters in interactive drama environments.

There are also some groups that try to mimic the processes involved in emotion in animals and humans. Some of these groups work at a cognitive level (Aaron

Sloman and the Cognition and Affect group [Sloman 1987][Sloman and Logan 1999]). Others approach emotions at the neural level [Balkenius and Móren 1998] and at the biochemical level [Cañamero 1997]. Other authors have been using emotion to improve the performance of agents at the task level, namely Clark Elliot and the Affective Reasoning Project [Elliot 1994] and [Numao et al 1997] that have used a model of the emotion of listeners to improve the performance of a system for automatic music composing.

None of the mentioned approaches address the role of emotion in inter-agent communication.

Section 2 describes the scenario that will be used to exemplify the approach taken in this paper. Section 3 describes our general architecture for artificial autonomous agents, with special emphasis on emotion. Section 4 presents requirements and specifications for a concrete implementation of the architecture described in section 3. Finally, section 5 concludes and points further developments.

2 An application scenario

This section describes an imagined application scenario used as examples through out the paper. This scenario is made up by several agents and multi-agent systems with road traffic applications. Some vehicles have navigation and driving assistants. There are local area agents that manage information about traffic signs and semaphores. There are local area multi-agent systems that monitor the traffic conditions including global level of service (e.g., traffic jams, sparse traffic) and abnormal events (e.g., stopped vehicles, accidents). There are also yellow pages agents that keep record of all agents and multi-agent systems in the society.

Figure 1 is a local map of a city with some streets. Small integers identify street intersections (1, 2, 3, …). Streets are referred to by the letter s with a subscript (s_1, s_2, …). r_1 and r_2 are two alternative routes that could be taken by a vehicle to reach the destination indicated by the world "Goal" in the figure. Agents and multi-agent systems are identified by capital letters (A, B, …).

A is an agent that drives a vehicle following instructions specified by the user. The user specifies its desired destination and the agent has to drive the vehicle from its current position to the user-specified destination. The position of the vehicle is determined through a satellite system (GPS, Global Positioning System) and a digital map of the city inside the vehicle.

B is a traffic-monitoring multi-agent system [Botelho et al 1999] that covers the traffic to the left of the double dashed line in the middle of Figure 1. C is a similar traffic monitoring system that covers the right region of the map.

D and E are responsible for keeping vehicles informed about the traffic signs and semaphores existing in the left and right regions of the map, respectively.

Besides agents A, B, C, D and E, there are some yellow pages agents that maintain descriptions of all agents and multi-agent systems. Each city map is associated with a global yellow pages agent (Global DF) that keeps record of all other yellow pages agents relevant to traffic applications in the city area. Agents DF_1 and DF_2, contain a record of all the agents that provide traffic related services in their region, including vehicle-agents, traffic monitoring multi-agent systems, traffic signs and semaphore agents.

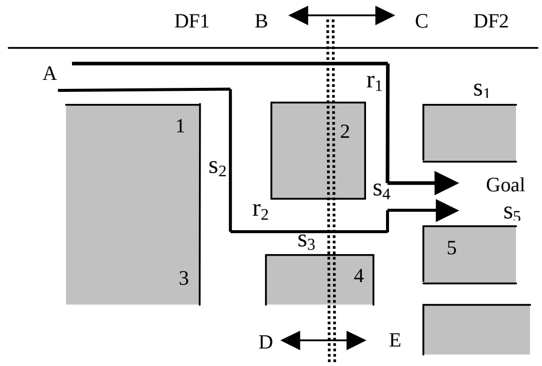


Figure 1 - An agent society in the traffic domain

This paper focuses on communication management in agent A, while it drives its vehicle to a desired destination.

2.1 General approach and technical options

In terms of inter-agent communication, the general approach taken in this paper is based on the information subscription protocol described in [Botelho et al 1999] which, in turn, is a development and an application of [D'Inverno, Kinny and Luck 1998]. According to this protocol, in the initial phase of their lifecycle, agents request information providers to send them the desired information. Agents also receive information requests from other agents. The first step of the information subscription protocol is to request the yellow pages agents to inform it of the names of the providers of the desired types of information. In the second step of the protocol, the agent requests the providers to send it the desired information whenever new instances are available. The requester doesn't need to repeat its requests, because providers will keep sending them the desired information whenever it is available.

This protocol has several advantages. First, agents ask each information class only once, not whenever they need it. Second, an agent only receives the information it has requested. Hence, it can make some simplifying assumptions regarding the format and the content-type of the received informative messages. Third, agents are kept open because they are not hardcoded to send information to a specific set of agents known *a priori* by the designer. Information providers only have to send information to the agents that request it.

All agents communicate with each other in ACL [FIPA 1998b]. All agents register their services with the corresponding yellow pages agent (generally known as Directory Facilitators [FIPA 1998a]). This means that if some service is offered by some agent or by some multi-agent system, it will be known by some directory facilitator (DF).

2.2 Problem solving strategy of vehicle-agent A

Agent A in Figure 1 is a vehicle control agent. A vehicle-agent uses information regarding its vehicle (e.g., position, speed), information about traffic conditions and information about the street in order to drive the vehicle to the user specified destination. This subsection examines the problem solving strategy taken by vehicle-agent A.

As the vehicle moves along the city streets, the agent uses its current position and the city map to plan (or to re-plan) some alternative routes to reach the user-desired destination. After computing a few alternative routes, the vehicle-agent asks the global yellow pages agent (Global DF) to tell it the names of the specific directory facilitators of the regions traversed by the best alternative route. Then, it asks the regional directory facilitators whose names were determined by the Global DF to tell it the names of the agents that provide relevant traffic services (traffic-monitoring multi-agent systems, traffic-signs agents, and vehicle-agents of the nearest vehicles).

The vehicle-agent subscribes information regarding the traffic-signs of its nearest vicinities, subscribes state information about the traffic semaphores and subscribes traffic information regarding the best route.

While the information regarding the traffic, the current position of the vehicle and the user specifications do not warrant abandoning the current route, the vehicle-agent must actually drive the vehicle along the route. Otherwise, if there is a planned alternative route the vehicle-agent selects the next best alternative route and plans a possibly empty transition path from the current route to the alternative route. If there is no more planned alternative routes, the vehicle-agent must re-plan some more alternatives.

Whenever vehicle-agent A decides to change to a different route or in case of re-planning, the vehicle-agent cancels the requests it has made to the traffic information agents, seeks the traffic information agents appropriate for its new situation and the new best route and sends them the requests.

2.3 Agent Architecture

A vehicle-agent is made up of several subagents: *planner*, *driver* and *communication manager*. Those are the agents specific for the vehicle domain. In a general agent architecture another subagents should be considered, namely, subagents that generates emotional signals.

The *planner* determines the vehicle main route, monitors its current position and actual progress, and re-plans the vehicle main route whenever its position is no longer compatible with the original route or any external recent event prevents the original route from being followed (e.g., one of the paths of the plan becomes blocked). The *planner* requests the driver to execute the individual steps specified in the route.

The *driver* receives and executes the steps specified by the *planner*. Its task is to perform lower level driving steps, avoiding obstacles and keeping the movement smooth.

The *communications manager* subscribes desired information with the information providers and manages the communication with the Global DF.

2.4 Agent interaction control problems

In the described scenario several control problems may arise regarding inter-agent communication from the point of view of vehicle-agent A, even though the technical options made (e.g., using ACL) and the general approach taken (i.e., using an information subscription protocol) ensure a relatively restricted communication situation.

Inter-agent communication control problems are situations involving inter-agent communication that require signalling and interruption mechanisms. These mechanisms play a role similar to that played by natural emotions in human beings. There are four classes of such situations:

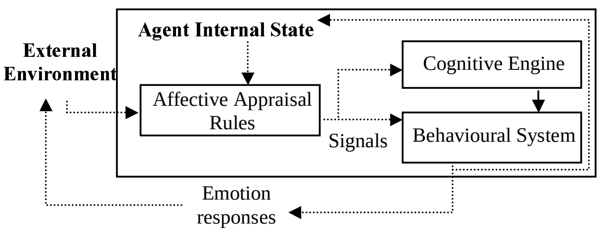
1. Communication control problems with/due to incoming messages (this class includes communication control problems due to the message's content, namely because the message is incomprehensible);
2. Communication control problems that arise while the agent is receiving and/or processing incoming messages (but are not caused by the reception of the message);
3. Communication control problems with/due to messages sent by the agent;
4. Communication control problems that arise while the agent is sending messages.

The agent current problem solving may have to be interrupted so that the agent can attend an important and urgent message. Examples:

- *(interrupt at the level of the driver* – Class 1*)* For instance if the *planner* requests the driver to turn right and then it receives new information regarding traffic conditions (for instance, an accident) and it decides not to turn, the *driver* activity of *"turning right"* must be interrupted so that the agent may receive the new request to move forward. It is worth noting that this is not a problem due to resource limitations. Even if the agent had infinite resources, the "*turning right*" would have to be replaced by a "*moving forward*" action.
- *(interrupt at the level of the communications manager* - Class 4*)* The *communications manager* may be busy in an information subscription protocol cancelling previous requests, seeking new traffic-information agents and sending them the information requests. A new important and urgent message regarding lost of oil may have come which entails a new action. Since this message must be attended before the protocol is completed, the communication agent may decide to suspend its activity so that the processor may be assigned to some other agent.

3 General agent architecture with emotion

Although we are not ready yet to provide an accurate definition of emotion, we will present our own view, hoping it will constrain the use of the concept in sensible and useful ways. Emotion is a process that involves appraisal stages, generation of signals used to regulate the agent's behaviour, and emotion responses. In the first stage, a set of appraisal structures is used to evaluate the global state of the agent (internal state plus external environment). If certain conditions hold, an emotion-signal is generated informing the agent of the result of the appraisal stage. A signal can have positive or negative valence. If the agent global state is found to conflict with the motives of the agent (e.g., instincts, goals, intentions, values, attitudes), a negative emotion-signal is generated. If the agent global state is specially favourable according to the agent motives, a negative signal is generated. The quality of the signal is related to the behavioural response that will be produced. Signals generated during an appraisal stage are sent to the behavioural system of the agent possibly giving rise to a behavioural response – the emotion response. In different situations, the same emotion-signal may give rise to different emotion-responses, depending on the internal state of the agent. Among other things, an emotion-response may be a reflexive action, it may increase or decrease the resources assigned to satisfy a given motive, it may create a new motivator (e.g., a new goal), it may set a new criteria to be used for selecting plans (or other methods) to achieve current motives. After an emotion-signal has been generated and an emotion-response has been performed, the global state of the agent changes and a new appraisal is done possibly generating a new emotion-signal. As sketched in Figure 2, this continuous process mirrors the concept of "feedback loop" of the control theory. The emotion process goes on in parallel with the cognitive process of the agent.



Only the emotion process and the components of the agent architecture relevant to it are depicted. Other interactions involving the cognitive engine are not shown. The process by which emotion is learned is also omitted from the figure.

Figure 2 - Emotion process

A simple metaphor for our model is an operating system, where the exceptions play the role of the emotion-signals, the actions performed by the exception-handlers play the role of the emotion-responses and the appraisal stage is played by all if-statements in the operating system code that generate exceptions. The difference is that once an exception is generated in an operating system, the exception handler will handle it in a predictable way, whereas, in our model, when an emotional signal is generated no one can tell how it will be handled.

For similar reasons there is an important difference between motivation an emotion, which are both control processes directed at the satisfaction of the agent motives (but see also [Aubé 1998]). In the emotion process, there is no direct coupling between the signals generated by the appraisal stage and the behaviours enacted in response to those signals. In the motivation process, the cognitive system of the agent analyses the agent global state and selects the appropriate behavioural response (e.g., it makes decisions, builds plans and selects actions). One could argue that this distinction might be related to the distinction between automatic (non-conscious) and deliberative (conscious) processing. The feeling of an emotion (which makes the agent aware of it and hence conscious) would be a result of the way signals are generated in specific architectures (e.g., the release of certain transmitters) or a result of the emotion-response (e.g., muscle contraction).

We believe this "preliminary definition" preserves much of the common sense understanding of emotion and identifies precise features that enable us to distinguish it from other concepts.

Among other things, emotion can be used as a basis for attention control, for performance evaluation and regulation, for identification of and recovery from malfunctioning-components, and for adaptive learning processes.

4 Implementation analysis

This section presents the analysis of an implementation of the general ideas described in sections 2 and 3, for the specific case of a vehicle-agent. Although the present analysis has been done for the case of a specific agent, nothing relevant is dependent on this agent.

An agent may be composed of several subagents with specific tasks sharing the same address space in a computational environment. All the subagents communicate between them through the inter-process communication and synchronisation mechanisms available in the specific operating system in which they are implemented. Only one global agent is registered in the yellow pages agents and in the platform management agents. All subagents are implemented as independent sub processes of the agent process.

Some of the subagents of a specific agent are designed to perform application specific tasks. The exact number and nature of task-subagents depend on the concrete application. One or more subagents are responsible for generating emotion-signals that are sent to some or to all the task-subagents. The task-subagents may use the emotion-signals generated to control any aspect of their operation. Some emotion-signals enable the agent to react in real-time, therefore they must be generated and processed very rapidly. Finally, some subagents are responsible for reading the messages from the input channel (e.g., a tcp/ip socket), convert them to an appropriate internal format and deliver them to the appropriate task-subagents (possibly with priority information attached). In order to allow the emotion-subagents to generate emotion-signals in early processing stages (possibly based on very little information), the message translation and delivery process may have to be accomplished by several message-processor subagents.

Emotion-signals are implemented through a software interrupt mechanism that is used only to signal the subagents that an emotion-signal was generated. The contents of the emotion-signal are written in a shared memory available to all subagents. The task-subagents that want to receive emotion-signals must catch the software interrupt and process the emotion-signal written in the emotion shared memory. If a task-subagent does not want to receive any emotion-signal, it disables the software interrupt. If a task-subagent wants to decide whether or not to ignore each emotion-signal, it catches the interrupt, and decides what to do with the emotion-signal received in the emotion shared memory.

In Figure 3 we present the agent architecture of a vehicle-agent. The circles represent the different subagents of the vehicle-agent (sub processes). The boxes correspond to information recipients and the arrows represent the message flow between processes and recipients. In order to simplify the reading of the diagram we have not presented the complete message flow, namely the messages sent by the agent to the external environment were not depicted.

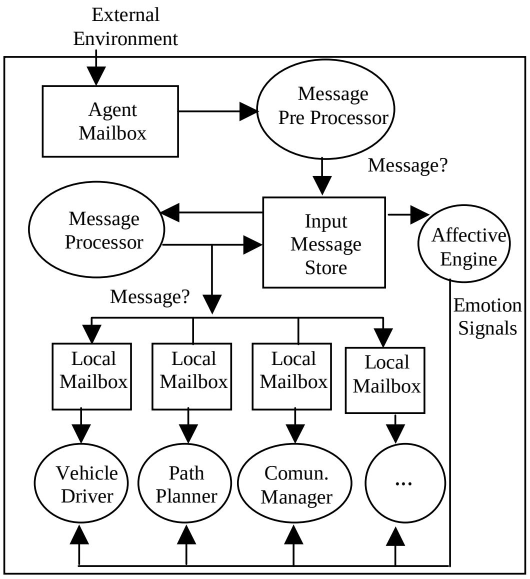


Figure 3 Internal Agent Architecture (vehicle-agent)

The translation of the incoming messages into an internal representation and their distribution to appropriate subagents is ensured by the Message Pre-Processor and the by Message Processor subagents. The reason for having two subagents to achieve those two goals is to allow the emotion-subagents to react sooner to the incoming messages, even before the content of the message is completely known.

The Message Pre-Processor is responsible for reading the messages from the Agent Mailbox and writing them in the Input Message Store (IMS). When the message is first written in the IMS, a minimum set of information sufficient to activate the Affective Engine is extracted from the message and explicitly stored with it (e.g., message type, *:sender* and *:reply-by*). In often cases the identification of the sender is sufficient to generate an emotion-signal. For instance, messages coming from the car driving in front. If we want to ensure a quick reaction, the Affective Engine cannot wait until the message is completely interpreted. Notice that, since the messages in FIPA standard transport format are received in ASCII representation, the translation is not a simple de-serialization process. It involves parsing. The Message Processor is responsible for the complete translation of the messages and their distribution to the appropriate subagents (local mailboxes).

Translated messages are not copied to the local mailboxes. Instead, the message stored in the IMS is replaced by its new format and the index of the message in the IMS is placed in the local mailboxes of the subagents that must receive it. IMS is implemented as a shared memory, therefore messages can be accessed in any strategy (FIFO, LIFO or direct access).

The Mailbox (based on a socket) manipulated by the Message Pre-Processor can only be accessed in a FIFO strategy. Notice that one message can be *"delivered"* to several mailboxes. It may also happen that some messages stored in the IMS do not have a receiver (e.g., messages processed only by the Message Pre-Processor).

The Affective Engine reads all messages available in the IMS and generates emotion-signals in certain circumstances. The emotion-signal generated at each instant of time depends on the amount of translated information available, i.e., depends on what the Message Processor has achieved so far. In order to avoid that the Affective Engine processes the same message twice, the Affective Engine attaches a status information to the message when it process it. There are three different message statuses: i) not processed ii) partially processed (before the Message Processor) and iii) completely processed (after the Message Processor). When the Affective Engine selects a message with status *partially processed*, it processes it again because some new inferences may be drawn. The Affective Engine never processes messages with the status *completely processed*. The Affective Engine is basically a set of simple rules with the following format *If condition then signal*. In order to ensure a quick response, the left-hand side of the rules must be simple, i.e., it should not involve complex information. The conditions of the rules, among other things (see example 2 of section 2.4), check information about the messages. Some rules depend on only one message at a time while others depend on several messages. For example:

```
If    message(ID) and sender(ID, Sender) and
      class(Sender, 'Traffic Surveillance') and
      content(ID, Content) and
      matches(Content, 'Accident')
Then  Signal (Signal_Description(
         Type: 'Attention-shift',
         Receiver: 'Planner',
         Intensity:'High',
         Object: message_id(ID)))
If    number_of_messages_not_processed(Number)
      and is-high(Number)
Then  signal(Signal_Description (
         Type: 'Performance evaluation',
         Valence: negative,
         Receiver: ALL,
         Intensity: 'Low))
```

In the first rule, the Affective Engine sends an Attention-shift signal to the Planner subagent because a Traffic Surveillance multi-agent system had sent a message related to an accident. Since that message can be crucial for the planner, the intensity of the signal is high. The message Id is sent to the planner as the object of the attention-shift warning so that the planner can quickly retrieve the message.

In the second rule a negative Performance-evaluation signal is sent because there are too many messages in the message recipient. The signal is sent to all task-subagents.

The task-subagents of the vehicle-agent automatically receive the emotion-signals sent by the Affective Engine. The main idea is that the operation of the subagents is interrupted so that the emotion-signal is interpreted and, if necessary, some response can be performed. However there are situations in which a task-subagent cannot spend time to interpret the signal (e.g., when the driving subagent is trying to avoid a collision). In those situations, emotion-signals are disabled. When the subagent completes the execution of a critical sequence of actions, emotion-signals must be enabled again.

After receiving an emotion-signal, the emotion-signal handler decides if the signal is relevant and strong enough to justify the interruption of the current action. If so, the action is interrupted and new actions are carried out (e.g., reading a new message from the local mailbox or from the IMS if the Message Processor has not yet delivered it to the process mailbox). If, after the emotion-signal is processed, the goal the agent was previously pursuing becomes obsolete, the goal is aborted. Otherwise the agent resumes its goal.

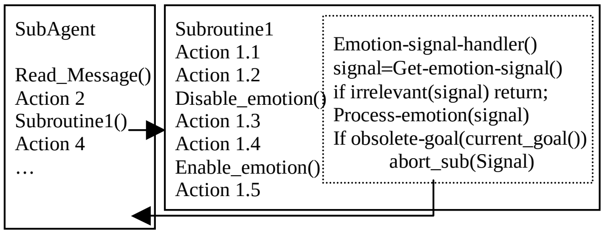


Figure 4 An example of an interrupt mechanism based on an Emotion-signal

If the emotion-signal received is not relevant or not strong enough, the subagent ignores it. A new message will be read from the local mailbox when the subagent plans to do so.

In Figure 4, actions 1.3 and 1.4 of the sub routine 1 of a specific subagent cannot be interrupted. If, during the execution of action 1.2 an emotion-signal is received and processed by the subagent and, as a consequence of the emotion-signal, the execution of the sub routine 2 becomes useless (or undesired) it may be aborted and the control returns to the main program of the subagent.

5 Conclusion and Future Work

We have analysed specific examples of four classes of situations in an intelligent vehicle scenario in which we could use artificial emotions to detect some agent interaction problems and signalise the possible need for some adaptive reaction of the system.

The paper describes a vehicle agent architecture containing several behavioural and cognitive subagents (driver, planner, communication manager) and an affective engine that detects possible problems related with communication (e.g., it detects situations in which many messages were lost, it detects situations in which a possibly important message has been received by the system). This particular architecture is an instantiation of the general architecture for artificial emotion presented in section 3.

Finally, the paper analyses the implementation of a general platform to be used to develop emotionally intelligent agents. We conclude that the usual inter process communication and signalling capabilities of general multi-processing operating systems constitute sufficient requirements for the kind of the described emotion-based control mechanisms.

In the near future, the platform will be implemented and several agent applications will be developed. This will enable us to draw more accurate conclusions regarding the usefulness of emotion as a component of the agent control system and about the capabilities of the proposed theory and platform.

6 References

[Aubé 1998] Aubé, M. (1998) "Designing adaptive cooperating animats will require designing emotions: expanding upon Toda's urge theory", in Canamero, D.; Numaoka, C; and Petta, P. (eds) *Proc. of the Workshop "Grounding emotions in adaptive systems"* International Conference of the Society for Adaptive Behaviour (SAB'98), p7-13

[Balkenius and Móren 1998] Balkenius, C. and Morén, J. (1998) "A computational model of emotional conditioning in the brain", in Canamero, D.; Numaoka, C; and Petta, P. (eds) *Proc. of the Workshop "Grounding emotions in adaptive systems"* International Conference of the Society for Adaptive Behaviour (SAB'98), p13-18

[Bates et al 1992] Bates, J., Loyall, A.B. and Reilly, W.S. (1992) "An architecture for action, emotion and social behavior", *Proc. of the 4th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'92*

[Botelho and Coelho 1998] Botelho, L.M. and Coelho, H. (1998) "Artificial agents with artificial emotions", in Sycara, K. and Wooldridge, M. (eds) *Proc. of 2th International Conference on Autonomous Agents (Agents98)*, p449-450, ACM Press

[Botelho and Coelho 2000] Botelho, L.M. and Coelho, H. (1999) "Machinery for Artificial Emotions", *Cybernetics and Systems*. Submitted.

[Botelho et al 1999] Botelho, L.M.; Lopes, R.; Sequeira, M.M.; Almeida, P.; and Martins, S. (1999) "Inter-agent communication in a FIPA compliant intelligent distributed dynamic-information system", in Callaos, N; Nada, N; Cherif, A; and Aveledo, M (eds) *Proc. of the 5th International Conference Information Systems Analysis and Synthesis (ISAS99).* International Institute of Informatics and Systemics.

[Cañamero 1997] Cañamero, D. (1997) "Modeling motivations and emotions as a basis for intelligent behavior", *Proc. of the First International Conference on Autonomous Agents (Agents'97)*

[Elliott 1994] Elliott, C. (1994) "Research problems in the use of a shallow artificial intelligence model of personality and emotion", *AAAI'94,* p9-15

[D'Inverno, Kinny, and Luck 1998] D'Inverno, M.;. Kinny, D.; and Luck, M. (1998) "Interaction protocols in Agentis", *Proc. of the ICMAS98,* p112-119

[FIPA 1998a] Foundation for Intelligent Physical Agents (1998) "FIPA 97 Specification, Part 1, Version 2.0, Agent Management"

[FIPA 1998b] Foundation for Intelligent Physical Agents (1998) "FIPA 97 Specification, Version 2.0, Part 2, Agent Communication Language"

[Numao et al 1997] Numao, M., Kobayashi, M. and Sakaniawa, K. (1997) "Acquisition of human feelings in music arrangement", *IJCAI'97,* p268-273

[Reilly 1995] Reilly, W.S.; and Bates, J. (1995) "Natural negotiation for believable agents", Technical Report CMU-CS-95-164, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA

[Sloman 1987] Sloman, A. (1987) "Motives, mechanisms and emotions", *Cognition and Emotion,* 1:217-234

[Sloman and Logan 1999] Sloman, A. and Logan, B. (1999) "Building cognitively rich agents using the SIM_AGENT toolkit", in Communications of the Association of Computing Machinery, 43(2):71-77