EMOTIONAL CHARACTERS IN VIRTUAL ENVIRONMENTS

Tiago Sepúlveda, Carlos Martinho, José Proença and Mário Rui Gomes
INESC - Instituto de Sistemas e Computadores
R. Alves Redol, 9, 6º, 1000 Lisboa, Portugal
e-mail: {Tiago.Sepulveda, Carlos.Martinho, Jose.Proenca, mrg}@inesc.pt

ABSTRACT

The design of believable emotional characters is the goal of the architecture described in this paper. The novelty of our approach relies on the distribution of the synthetic characters' architecture in two complementary world models: semantic and physical. The characters emotional state is considered in the semantic world model, implemented as stand-alone Artificial Intelligence module, where general behaviour directives are generated. The agent physical architecture, part of the physical world model, carries out those directives taking advantage of its layered structure. The proposed architecture was used to build the interactive application *S3A, a story of dolphins*, exhibited in Lisbon at *EXPO 98* and featuring two synthetic dolphins.

Keywords: virtual environments, computer graphics, synthetic actors, behaviour simulation.

1 INTRODUCTION

Portugal was the host country of the last world fair of the century – *EXPO'98*. The exhibition was held between May 20 and September 30, and its motto was: *"The Oceans, an Heritage for the Future"*. Aiming at the promotion of innovative scientific projects related with the exhibition theme, the *EXPO'98* Territory Pavilion invited our lab – *GAIVA*, and *Projecto Delfim*, a non-profitable organisation devoted to the preservation of the Sado dolphin community, to create an interactive application featuring artificial dolphins.

The challenge was to present a real-time simulation of organic believable software agents exhibiting emotional behaviour. *S3A, a story of dolphins* is the product of this challenge, presenting the visitor with a virtual window to an enchanted world where humans and dolphins can communicate and share feelings and emotions.

This paper presents the *S3A* project and the architecture developed to support the characters inhabiting the *S3A* scenario. In section 2, we introduce *S3A* and present an overview of the story behind the project. The general architecture of *S3A* is then described in section 3, emphasising the division of the world model into two distinct blocks: the mind and the body. These blocks together with the application sound server are further described in the respective subsections. Section 4 focuses on the characters "physical" architecture. The architecture is composed of a control model, built around the concept of behaviour block, and a physics model. In section 5, to complete the description of *S3A*, a simplified example of how character behaviour control is performed, is presented. Then, in section 6 we present the results gathered while developing and exhibiting *S3A* before drawing the conclusion that the presented architecture provides a very simple way of introducing believable characters in real-time synthetic environments.

2 S3A, A DOLPHINS STORY

S3A is a 3D interactive application that enables a user to observe the underwater ecosystem of the portuguese river Sado and communicate with an artificial dolphin named *Isolda*.

The interaction between *Isolda* and the public is done through a porcelain apparatus instrumented with four pressure buttons modifying *Isolda*'s emotional levels (happiness, sadness, passion and fright).

But *Isolda* is not alone in the synthetic environment. Besides sharing the virtual environment with several specimens of Sado's fishes, she has the company of, *Tristão,* a typical bottlenose Sado dolphin, shy and suspicious of men activities. However, and unlike *Isolda*, he is not able to feel the apparatus action.

Isolda, *Tristão* and Humans constitute the vertexes of a dramatic triangle, introduced to capture the visitors' attention to the *S3A* plot.

The communication apparatus was placed in front of the large screen where *S3A* was displayed. *S3A* was a non-stop application, displayed, from May to September, to thousands of people (10.000) every day.

3 SYSTEM ARCHITECTURE

The novelty of S3A architecture resides in the two-level modelling of the virtual world (Fig. 1):

At the semantic level, the module **mind** evaluates qualitatively world events and produces behavioural directives to be fulfilled at the physical level.

At the physical level (the **body** module), each agent executes the actions required to carry out the task implied by the current directive(s).

As we can see in Fig. 1, besides **body** and **mind**, the general architecture of *S3A* is extended with the **3D Sound Server 2**(**3DSS2**), a remote application responsible for generating all the sound effects of *S3A*.

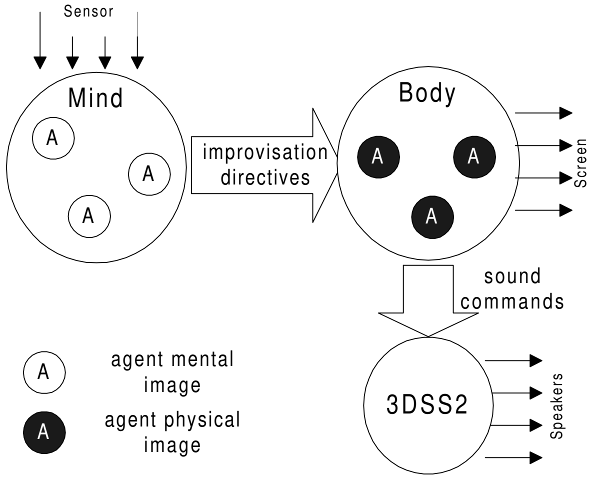

Fig. 1 - *S3A* architecture.

3.1 MIND

Metaphorically, the **mind** module can be seen as a dynamic scriptwriter that issues improvisation directives to the **body** module in narrative time.

Narrative time is a subjective time span that represents the time for something relevant for the storyboard to happen. In *S3A*, the narrative time span is limited. There is a lower bound of 5 seconds to guarantee that the users have time to identify the action that the agents are performing, and there is an upper bound of 30 seconds to prevent the system from being stalled in the same narrative cycle. This is the case when an agent isn't capable of carrying out the task implied in its improvisation directive.

When the narrative time span ends, a new set of improvisation directives are fetched from the **mind** module, which takes into account all the sensor data read since the last query. The only existing external events in *S3A* are sensor data from user interactions. Since the sensor changes only values of the **mind** module emotional variables (Fig. 1), their effect is accumulated, during the narrative cycle, not interrupting the performance of current actions.

Improvisation directives fetched from the **mind** module constitute a set of commands that the **body** module must fulfil. Again metaphorically, the **body** can be seen as a set of actors. They must follow accurately the script but can improvise on it as long as they still convey to the viewer the actions specified by the improvisation directives. This "one-way" approach allows the **mind** module to be integrated in real time virtual environments with minimum performance degradation.

3.1.1 The Model

The model implemented by the **mind** module follows the cognitive theory of emotions of *Ortony*, *Clore* and *Collins* [Orton88] for the goal and emotional reaction structure. As proposed in this theory, emotional reactions are divided into three categories:

- Emotions based on aspects of objects, launched when a relevant object is nearby (e.g. *Tristão* hates to be near the plane);
- Emotions based on actions of agents, launched when a specific action pattern is recognised based on the agent standards (e.g. *Tristão* admires *Isolda* for her courage to explore the plane);
- Emotions based on consequences of events, launched when a specific goal state changes (e.g. *Tristão* feels hope as he starts to follow *Isolda* but soon get disappointed as he realises she is too fast for him), or when a prospect irrelevant event happens (e.g. *Tristão* feels distress as he ears a boat approaching).

3.1.2 Implementation

The *core* of the **mind** module uses a conceptual framework to represent the virtual world abstraction. The world is topologically represented by a set of linked areas where entities live. The entities are the characters (the agents) of *S3A*.

The narrative simulation cycle of the *core* of the **mind** module uses a "perceive-react-think-act" approach. First, the characters inspect the area surrounding them, filtering the relevant entities and events, discarding others. This information is then used to update each agent model of the world. After that, goals and emotional reactions are matched against the perceived events and updated accordingly. All the active emotions are mapped onto behaviour features [Reill96]. Finally, a set of rules is fired based on the active goals and the behaviour features intensities. The actions produced by those rules are then sent to the world where they will be perceived by the other characters. Details can be found in [Marti98].

3.2 BODY

The **body** is the application central module. Besides keeping the agents database, **body** controls the application frame rate and manages the simulation cycle.

The simulation cycle has the following steps:

1. **Goal fulfilment**: analysis of the level of accomplishment of the current improvisation directives.
2. **Agent set up phase**: identification of the agents who must fulfil the directives (directive subjects) and activation of the behaviours matching the directive actions.
3. **Agent execution phase**: management of the simulation step for the agent physical instances.
4. **Drawing**: image generation.

In phase 1, all agents are checked for the conclusion of behavioural directives assigned to them by the **mind** module. If an agent specific directives are accomplished it must improvise while waiting for new directives. The improvisation must be a neutral behaviour, in the sense that it must not interfere with the actual world state in order to preserve the accuracy of the **mind** model, and should be consistent with recent behaviour

The **body** fetches a new packet of directives when all agents have fulfilled their objectives or when the maximum narrative time span as been reached. In this last case, the coherence between the **mind** world model and the **body** world model must be maintained by forcing the current behaviours to finish satisfactorily. For example, if a dolphin cannot eat its prey during the maximum narrative time span, the fish is forced to leave the scene because the next narrative cycle will assume that that fish does not exist.

When a new packet of **mind** directives arrives, the **body** module enters in the agent set up phase (step 2). The agent set up phase assign the directives to the correct subjects. This phase consists in identifying directive subjects and activating the behaviour(s) that are matched with the directive being processed.

In step 3, the behaviours that are active in the agent control structure are performed and the scene geometry is updated.

Finally, in step 4, the visualisation pipeline is processed and the generated image is sent to the screen (Fig. 1).

3.3 SOUND SERVER

The **3DSS2** is a remote server capable of generating simultaneous spatial positioned sound. It is not only responsible for satisfying agent sound requests, but also responsible for the generation of the *S3A* ambience sound.

The communication (performed through sockets) is based on the attribution and manipulation of sound channels (client ids). The implementation uses *Microsoft Direct Sound* API to drive all sound hardware and deform the sound according to the source position. *S3A* has 5 sound channels: 1 ambience and two for each dolphin. Therefore the dolphins can emit two sounds simultaneously (vocal and sonar), an approach based on ethology.

The ambience sound is a pre-programmed sound sequence that is launched at the beginning of the application. The dolphin sounds are requested accordingly to the performed behaviour and emotional state using a table that produces a sequence id. The dolphin client id, the sequence id, the dolphin location and sound parameters, like volume and reverb, are concatenated in a command line that is sent to the **3DSS2** socket.

With the resource-oriented architecture proposed and further described in section 4, dolphin sound requests are done transparently. The sound is just another agent physical resource encapsulated in a behaviour layer. Only the behaviours involved in using sound need to know the **3DSS2** socket. That will become more obvious after the explanation of the next section.

4 AGENT PHYSICAL ARCHITECTURE

Since 1987, when the seminal work of *Reynolds* [Reyno87] was published, the simulation of lifelike agents is based on the modelling of responses of each individual to external stimulus. Several researchers (e. g. [Tu94]) followed his work establishing a first standard architecture for the simulation of lifelike behaviour that consists in building the ethologic model of the pretended entity.

We took this approach a step further creating two distinct internal models of the virtual environment, implying the distribution of the character architecture. The character is composed of:

- A mental architecture, residing in the **mind** module, providing support for high-level behaviour command generation (improvisation directives) [Marti98].

- A physical architecture, residing in the **body** module, responsible for the management of the character activity and of the fulfilment of the goals implied by the current improvisation directives.

As proposed in [McKee90] and depicted in Fig.2, the agent physical architecture is divided into a *control* model and a *physics* model.

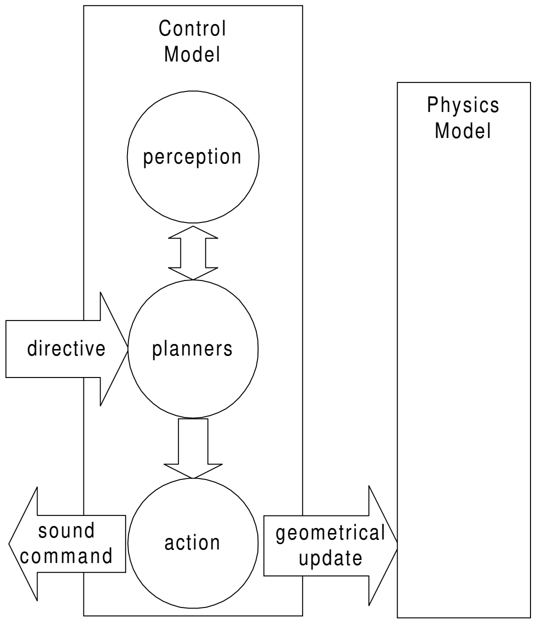

Fig. 2 - Agent physical architecture.

4.1 CONTROL MODEL

4.1.1 Implementation Concepts

The units that constitute the control model (*perception, planners* and *action*) were implemented using a common data structure called behaviour block. Behaviour blocks are composed of (Fig. 3):

- a command parser, that receives directives and maps them into behaviour activations.

- an array of behaviour layers, the recipient of the block functionality.

A behaviour layer is a mechanism developed to manage mutually exclusive resources. Like *Blumberg*'s groups [Blumb95], a layer supports only an active behaviour, inhibiting the simultaneous execution of concurrent behaviours, that is, behaviours that act over a same resource.

As an example of the importance of the mechanism, imagine a layer associated with the dolphin's head. Behaviour inhibition would not permit the simultaneous activation of *look*, a behaviour that would point the dolphin's head to a precise location, and *shake_head* a behaviour that would cause the dolphin to shake its head, since both are in the same behaviour layer (the head layer).

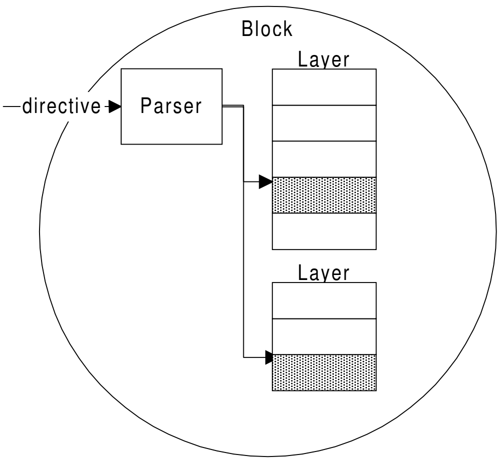

Fig. 3 - Internal block structure.

The working procedure of a behaviour block is very simple. Based on the received directive, the parser checks for the presence of the intended behaviour (through a behaviour id) in the behaviour layers. If this operation is successful, the behaviour becomes active in the layer where it belongs replacing the last (if any) activated behaviour. Later, all activated behaviours will be executed.

The agent control model is fully configurable by a text file. In this file the blocks, layers and behaviours can be organised through the use of a very simple syntax.

Each layer is semantically linked with one specific character ability, classifying the behaviours according to the resource they address. Using blocks and layers, behaviours are classified according to (1) the type of action they perform (2) the resources they use.

4.1.2 The Model

As it can be observed in Fig. 2, the agent control model is modelled according to 3 functional units, implemented as behaviour blocks: a decision nucleus – *planners*, a perception centre – *perception* and an action unit – *action*.

- Planners

The *planners* are the top-level block of the control model. The guidelines from the **mind** module are interpreted by the *planners* parser and, according to the improvisation directive, a set of *planners* behaviours is activated.

The *planners* layers are structured in such a way that exclusive tasks will not be performed at the same time. As an example we can think of tasks implying the movement of a dolphin to different locations on the scene, like following other dolphin and chasing a fish. The two tasks must be included in the same *planners* layers (a global motion layer) to prevent the execution of the two behaviours at the same time.

Each *planners* behaviour can become active for a period of several simulation steps, until the next call to the **mind** module. In each simulation step, it can send directives either requesting detection of the physical environment (to the perception block) or demanding the execution of an "atomic" action (to the action block).

- Perception

The *perception* block includes behavioural layers that inquire the physical world, and, that can be seen as sensors filtering the pertinent data needed to perform the demanded actions.

- Action

The *action* behaviours are responsible for carrying out, frame by frame, planner directives and to build a report of geometrical changes, in the form of transformation matrixes, to be applied to the physics model. The behaviours are organised in layers that are directly related with available agent physical resources, like the dolphin’s head or fins.

4.2 PHYSICS MODEL

Spring models, introduced by Miller [Mille88] and developed by Terzopoulos [Tu94], were initially the best option for dolphin motion simulation. But the spectrum of motions required, turned the task of defining and tuning the complex dynamic (spring) models too time consuming for the development time available.

Hence, the methodology adopted to develop the agent *physics model*, was based on the one proposed in [Sepúl96] that enables the quick prototyping of undulatory motion using a modified articulated structure.

The *physics model* is constituted by:

- Sections defined as sets of mesh vertexes.

- Control points defining the origin of the local coordinate systems for the section that the control points refer to.

In Fig. 4 we can observe that control points are the *physics model* counterparts to the *control model action* block resources.

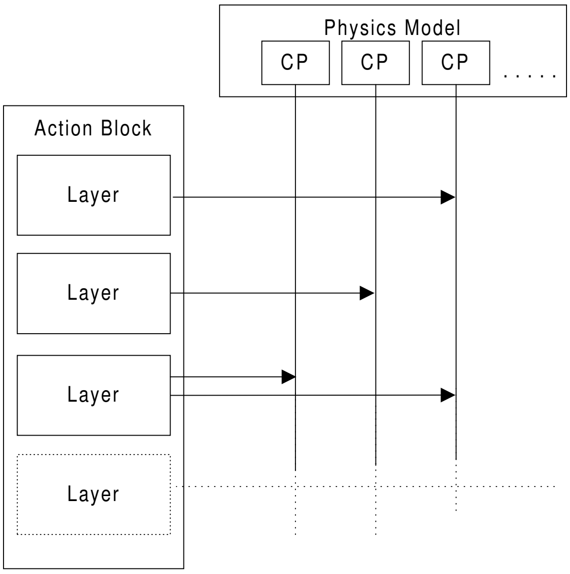

Fig. 4 - Action behaviours sending geometrical updates to the *physics model* control points (CP).

A good example of how the *physics model* works is the generation of the undulatory motion by simulation of muscle contracting waves. As depicted in Fig. 5, the dolphin 3D model is divided in longitudinal sections. To define the new positions of these sections, when the dolphin is moving, we keep the last points of the dolphin's trajectory, use them to build a *spline* and then distribute the section control points over the constructed curve. This operation together with the attitude calculus necessary to prevent unrealistic torsion of the dolphin body enables the construction of scene geometrical transformations that are applied to the referred sections.

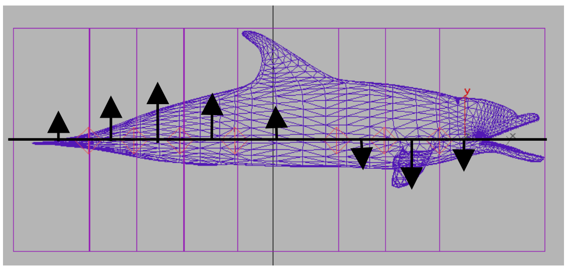

Fig. 5 - The dolphins longitudinal section oscillations simulate a muscular contraction wave.

To obtain the desired motion [Webb96] the *oscillation* behaviour (from *action* block) sends the value of section oscillations (Eq. 1) to the *physics model*, as control point local translation matrixes, that afterwards will be superimposed to the relevant sections to obtain a smooth organic motion. The frequency of the oscillation must be adjusted with the dolphin speed, a task involving planning, that, consequently, must be performed by the planner that activated the *oscillation* behaviour.

$$z = A \cdot \sin(2\pi f t + \Psi) \qquad \text{(Eq. 1)}$$

f is the oscillators frequency.
A is the maximum amplitude
Ψ is the initial phase.

The definition of the *physics model* is the trickiest task of the creation of a character since it requires expertise in the division of the geometric model in sections and control points. This operation can be performed with the help of a commercial 3D-modeling tool as long as the designer follows certain nomenclature rules. An ASE file parser was developed, to introduce the information regarding the *physics model* automatically. By this way, designers are able to fully prototype the characters physical attributes without needing any kind of programming knowledge, according to ethological data.

5 A WORKING EXAMPLE

At a given time step, imagine that the **mind** module generated the following improvisation directive:

ISOLDA EAT FISH-XX

After the identification of the directive subject (*Isolda*), in the agent set-up phase, the command ***EAT FISH-XX*** is mapped by *Isolda*'s *planners* parser onto the behaviour *eating*.

The planner *eating* requests the activation of the *modified neighbour detection* and the *target detection* sensors. The *modified neighbour detection* behaviour is responsible for the detection of all neighbours pertinent to the collision avoidance algorithm, while *target detection* keeps track of target data.

In subsequent application simulation cycles, *eating* will stay active until the arrival of an improvisation directive that deactivates it. Until then, *eating* will follow its plan, consisting of:

- *Isolda* approaching the target: in this phase, *Isolda* must follow and look to the target. The emission of intense sonar sounds to knock out targets (called *bangs*) is also simulated.

- Capturing prey: When *Isolda* enters in "physical contact" with the target, she opens her mouth and the fish is removed from the scene.

- Wandering: after eating *Isolda* is told to improvise (*wander* behaviour) in the area where the action took place until all improvisation directives are accomplished.

Fig. 6 - *Isolda* is preparing to eat a fish.

Hence, the arrival of the *EAT* improvisation directive causes:

-The activation of (*Isolda set-up phase*):
eating FishXX (*motion* layer of *planners* block)
modified neighbour detection and *target detection* (*perception* block).

-In the approach phase:
Fish-XX bang (*motion* layer of *Fish-XX planners* block)
sound bang (*sound* layer of *action* block)
look Fish-XX (*head* layer of *action* block)
move x y z (*position* layer of *action* block)
oscilate f (*oscilation* layer of *action* block)

-When capturing the prey:
mouth (mouth layer of *action* block)
move x y z (position layer of *action* block)
oscilate f (oscilation layer of *action* block)

-After eating (improvisation):
wander (motion layer of *planners* block)

6 RESULTS

During the short development time of *S3A*, about 4 months, the **mind** and **body** modules were developed independently proving the **flexibility** and **modularity** of the global architecture.

In the development, we observed that several architectural features were an important contribution for quick prototyping believable characters.

- The division of the world model into the **mind** and **body** module.
- The agent behaviour library structured in behaviour blocks.
- The external configuration of behaviour blocks (through text files).
- The external definition of the agents physical model (through ASE files).

Biologists accompanied the project to ensure that the motion and behaviour generated were **realistic**. The development team emphasised the organic aspect refining the dolphins *physics model* sections as needed.

In *S3A*, **Intelligibility of action** and **usability** are two related features: there is a direct link between user interaction and the behaviors performed. They became our major drawbacks. Although the textual output of the **mind** module embodied coherent agent behaviours given the emotion variables involved, the users did not understand what was happening. Several reasons contributed for the less intelligibility of dolphins' actions. The constantly overcrowded room did not provide any information or cue about the sensor effect. Therefore, the average visitor had, as in real life, a hard time figuring what was going on the dolphins' mind, and what was the sensor part in the mental process underlying *Isolda*'s behaviour. A deeper discussion of the problems and solutions can be found in [Marti99].

To solve the reported problems, we "domesticated" *Isolda*, and assigned, to each interaction, besides the corresponding change in the value of the emotional variable, an action script (Fig. 7). Even though, many people did not react well to this solution because they had to wait a few seconds till the scripts were triggered, due to the discontinuity of calls to *mind*. It was only with the help of an assistant, which explained the interaction procedure and the dolphins' typical reactions, that this problem was solved.

Besides the problems regarding interaction, we evaluated positively the public response to the **likelihood** of dolphin behaviour. The behaviours that dolphins chose were seldom identified, but the unpredictability and emotionality that, nevertheless, still transpired, instilled in those that were watching them empathy and involvement.

Fig. 7 – When the happiness button is touched *Isolda* executes an helicoidal motion.

7 CONCLUSION

The application *S3A, a story of dolphins* features software agents that can present lifelike motion and behaviour. The coupling of (user) actions and temperamental (dolphin) reactions makes credible the assumption that the observed characters are experiencing emotions. The use of an Artificial Intelligence module (**mind**) assured the behavioural reasoning of the agents, enabling the assignment of personality and emotional features independently of the agent physical attributes. The directives generated by the AI module are improvisation guidelines for the agent physical instances residing in the "physical" module (**body**).

The independence of **mind** from the rest of the application offers the opportunity to replace it easily for another behaviour generator module and to prototype the mental and physical characters instances separately.

The use of a multi-layered architecture for controlling independently the agent physical resources offered an increase of flexibility, indispensable for the real-time animation of characters with such complexity like the dolphins present in *S3A*: *Tristão* and *Isolda*.

8 ACKNOWLEDEMENTS

I would like to thank Ana Costa, Ana Torres, José António Conde and Joaquim Antunes for their laborious work during the development of *S3A*.

This work was supported by PRAXIS XXI, BD/9375 and BM/12597.

9 REFERENCES

[Blumb95] Blumberg B., Galyean T.: Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments, SIGGRAPH'95 Conference Proceedings, pp. 47-54, 1995.

[Marti98] Martinho, C., Paiva, A., Gomes, M.: Tristão and Isolda Synthetic Dolphins with Emotions, Proceedings of the Workshop on AI/Alife and Entertainment, ECAI'98, 1998.

[Marti98] Martinho, C.and Paiva, A.: Pathematic Agents, (to be published in) Proceedings of the Autonomous Agents Conference, 1999.

[McKee90] McKeena, M., Pieper, S., Zeltzer, D.: Control of a Virtual Actor, Proceedings of Symposium on Interactive 3D Graphics, pp.165-174, 1990.

[Mille88] Miller, G.: The Motion Dynamics of Snakes and Worms, SIGGRAPH'88 Conference Proceedings, pp. 169-173, 1988.

[Orton88] Ortony, A., Clore G., Collins A.: *The Cognitive Structure of Emotions*, *Cambridge University Press*, 1988.

[Reill96] Reilly, W.: Believable Social and Emotional Agents, Carneggie Mellon University PhD Thesis, 1996.

[Reyno87] Reynolds, C.: Flocks, Herds and Schools: A Distributed Behavioral Model, SIGGRAPH'87 Conference Proceedings, pp.25-34, 1987.

[Sepúl96] Sepúlveda, T.: *Actores Sintéticos em Ambientes Virtuais*, Msc Thesis, Techical University of Lisbon - IST, 1996.

[Tu94] Tu, X., Terzopoulos, D.: Artificial Fishes: Physics, Locomotion, Perception, Behavior, SIGGRAPH'94 Conference Proceedings, pp.43-49, 1994.

[Webb96] Webb, P.: Form and Function in Fish Swimming, Scientific American, Vol.5, No.4, pp.58-68, 1996.