The ANIMUS Project: A Framework for the Creation of Interactive Creatures in Immersed Environments

Daniel Torres
Department of Computing Science
University of Alberta
T6G 2E8, Edmonton, Alberta, Canada
dtorres@cs.ualberta.ca

Dr. Pierre Boulanger
Advanced Man-Machine Interface Laboratory
Department of Computing Science
University of Alberta
T6G 2E8, Edmonton, Alberta, Canada
pierreb@cs.ualberta.ca

ABSTRACT

This paper describes the architecture of the ANIMUS framework. This framework facilitates the creation of synthetic characters that convey the illusion of being *alive*. The components of ANIMUS are inspired by observations made in biological organisms, and provide means for creating autonomous agents that mimic awareness of their environment, of other agents, and of human audience. They also show particular roles, personality, and emotions, active and reactive behavior, automatic reflexes, and selective attention; use temporal memory and learning capabilities to evolve in their dynamic virtual worlds, and express their thought and emotions with a flexible animation system while they interact with the user in immersive 3D environments. ANIMUS creatures follow artistic conceptual designs and constraints that determine the way they behave, react and interact with other creatures and the user, allowing the designer to create meaningful and interesting characters. The framework can be applied to complex immersive environments like CAVE systems or other interactive applications like video games and advanced man-machine interfaces, providing high level tools for creating a new generation of responsive believable agents.

Keywords

Artificial intelligence, artificial perception, agent architectures, believable agents, broad agents, human-machine interaction, inmersive environments, interactive art, intelligent agents, synthetic creatures, VR entertainment.

1. INTRODUCTION

The challenge of creating artificial forms of life has generated a lot of interest in the scientific community. Today we can find agents with relatively good performance in areas like medicine, enterprise logistics, industrial engineering, travel and entertainment, to mention a few. Still, the place where we have seen the best artificial creatures that mimic lifelike features like emotions, personality, intention, autonomous and non-deterministic behavior is science fiction and animation films. Numerous research projects have done valiant efforts to create characters that behave like living organisms, applying the best AI and automation techniques, producing agents that look more *intelligent* than *believable* in the sense that they certainly learn and behave logically and efficiently (some of them even move or react like their real counterparts), but their behavior is usually cold, predictable, too perfect or intelligent, and hints the user into realizing that the character is nothing more than a lifeless puppet controlled by a computer program. Other attempts try to eliminate deterministic and repetitive behavior by the integration of more or less complex noise functions, but the actions of an interesting character are seldom based on noise, instead they follow a defined personality and emotional structure. What do artificial creatures need to actually give the illusion of life? Perhaps the missing ingredient can be described by Thomas and Johnston[22]:

> "What's funny about a hat sneaking up on a dog?"
> "It depends on the hat's personality –what kind of a guy he is"
> "Then how do you make a hat into a *believable character*?"
> "The same way you made a crude cartoon dog into Pluto. By showing the emotions. How else do you get life into anything?...(p. 505)"

A believable artificial creature shows its own thoughts and emotions, plans and beliefs; it can be tricked; it is not necesarely intelligent or efficient, nor always makes the best decisions, but acts within the bounds of its role and personality; Their animation may not be realistic, but succeeds at expressing its internal state and makes it interesting to the audience; as Reilly[18] points out: *"It turns out that sometimes being more realistic can decrease believability. For example, watching extremely realistic animation of human faces, like that of Terzopoulus[21], can be somewhat disturbing, whereas watching unrealistic animation, like Charlie Brown, can be very satisfying…(p. 11)"*.

The challenge of the ANIMUS project is to create believable creatures. To achieve this goal, we have broken the problem into three subgoals: How does a believable creature see the world? What happens inside his/its/her mind? how to

VRST'03, October 1-3, 2003, Osaka JAPAN.

successfully make the user perceive the internal emotions of our characters? This paper describes the approach we have taken to solve those questions.

The rest of the paper introduces the framework and illustrates its functionality with an example project named *Alebrije*. We begin by reviewing some related work. Section 4 briefly describes the inmersive VR environment, some basic definitions and the dynamics of information flow. The framework itself is detailed in section 5, where we describe its three main components. We finally conclude and provide insights about future work.

2. RELATED WORK

In this section, we briefly describe several projects that have either been a source of inspiration to ANIMUS or are closely related to our research. One of the first projects about interactive believable agents is the Oz Project[2, 13], where the term *believable* was taken from the art community and applied to computer agents. As explained also in Reilly[18], creating believable characters has strong artistic implications: Artists know how to conceive characters such that the audience can easily *suspend his/her disbelief* and consider them real. By letting the hand of art guide AI expertise, personality and emotions can be integrated to create the agents for interactive fiction and drama found in the OZ project. Reilly[18] adds functionality for *social behavior*, where agents engage in complex activities like negotiation, making friends, or helping other agents reach their goals. The simulated emotions are goal-based and depend on the various ways in which the environment and the behavior of the agent succeeds or fails to satisfy its objectives (The possibility of failing might, for example, produce *frustration*). These projects use a text based interface or a graphic world with simple 2D images to interact with the user.

Rizzo[19] also proposes a goal based model where agents have a taxonomy of goals and priorities which are always tried to be satisfied. A particular goal can be pursued in different ways, and several actions and plans are set with different priorities depending on the agent situation, context and personality. Interaction with the user is also text-based.

An interesting architecture that makes use of planning and explores the relation between emotions and decision-making is found in Gratch[9, 10], where graphs of cells containing preconditions or constraints, actions, and world state predictions are organized to form a particular plan. Plans can be extrapolated to predict the outcome of an action and compared with those of other agents to determine if they conflict or favor each other. This project is oriented toward modeling realistic human behavior in a high-end VR training environment for military personnel, and is among the few that include an immersive environment as part of the project. C4, by Isla[6, 11] is a layered brain architecture that perceives information from the world through a set of percepts (anatomical classification and data extraction units) organized in a tree. Each percept has its own algorithms for determining what and how much is known about entrant data. Percepts can be created or refined as needed by the character in real time. The C4 framework also provides an action system that evaluates aspects like what to do, when to do it, how worth is a given action, among others. The user is presented with a 3D environment where the agent performs a set of previously hand-made animations in response to its chosen actions.

Figure 1: Alebrije from several angles and poses

3. ALEBRIJE: A SIMPLE ANIMUS CREATURE

To illustrate the functionality of the framework, we show how it can be applied to bring a simple creature to life. The artist designs a character named **Alebrije**[1], a lizard–like creature approximately the size of a large dog pictured in Fig. 1. We will be making extensive mention of this creature when providing practical examples and explanations for the following sections.

4. THE VIRTUAL ENVIRONMENT

Before describing the ANIMUS framework, we detail the characteristics of the immersive environment in which our virtual creatures live. Our simple world model is inspired by Isla[6] and its main functions are:

1. To keep a list of the characters and objects currently interacting with the user.
2. To facilitate the interchange of information by means of a specialized blackboard.
3. To synchronize different events like how often are creatures sensing the world.
4. To administrate graphic rendering tasks.

[1]Some Mexican artisans use the term Alebrije when referring to a particular kind of imaginary creatures that are said to live in alternate worlds.

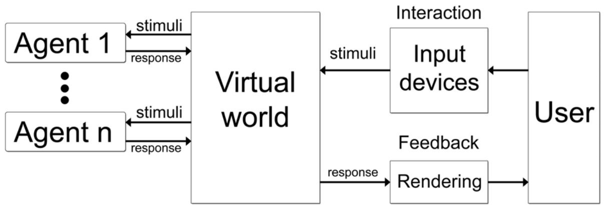

Figure 2: Layout of the inmersive virtual environment

The Virtual Environment layout is shown in Fig. 2. This setup can be implemented in a VR immersive system with an array of complex sensors like position, movement, sound, breathing sensors, visual gesture recognition or any other required device. The framework is ready to include any kind of information in the environment and pass it on to the virtual creatures, which are always designed with specific types of interaction techniques in mind. The basic information primitive is called *feature* and consists of a concept–value pair. A *concept* represents something that can be sensed by our character and range from physical properties like color, weight, temperature or size, to more complex abstractions like pain, hunger and fatigue. The value is a scalar magnitude that does not need to be within a particular range, but must reflect the nature of its corresponding concept. When a feature is produced by a character or the user we call it *stimulus*. A *stimuli* is the set of stimulus perceived by a creature at a given time.

Alebrije characters in a CAVE–like environment will implement vision and audition capabilities by using magnetic and sound sensors (If desired by the artist, Alebrije could, for instance, be sensible to the user's excitation state by including heart monitoring and galvanic skin resistance sensors). The perceivable features are therefore *sound intensity* and the *position* of other creatures and the user. If the user moves from one place in the VR room to another, he/she *emits* a stimuli consisting of the sound intensity detected by the sensors and his/her new position.

Creatures are able to *sense* stimuli from the user and other creatures by reading a specialized *blackboard* system administrated by the virtual world containing all perceivable stimuli at time t. To support a realistic behavior, perceived stimuli are modified to make it *relative* to the character *before* it is accessed. In other words, if the user moves behind a wall in the VR room, Alebrije should not be able to recognize his/her position, therefore creating an opportunity for the character to feel curiosity and maybe walk to a place that provides a better angle. If the user is too far away from a character and produces a noise, the intensity perceived by that character will be less than the one sensed by closer creatures. This principle is called *perceptual honesty*[2], and it is is of great importance to maintain a believable behavior. Creatures are also able to directly query other creatures (In a scene with two Alebrijes, if one moved behind a wall, the other could actually ask its partner for its position); nevertheless, creatures *must* assume that information might (and often will) be noisy, transformed or absent at all (if the intention of the other Alebrije was to hide, it won't reveal its new position when the searching Alebrije asks). ANIMUS world model and information interchange is described with more detail in Torres and Boulanger[23].

[2]Defined with more detail in [6, 11]

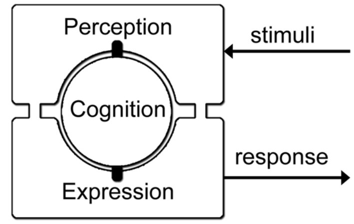

Figure 3: The three main components of an ANIMUS creature

Once ANIMUS creatures have perceived information about the world, they produce a response that may become a stimuli for other creatures. The virtual world synchronizes this perception cycle a given number of times per second and signals all creatures to update their inner states and output new stimuli for the duration of the simulation.

5. THE ANIMUS ARCHITECTURE

We will now describe in detail the ANIMUS architecture and exemplify each module with the help of our Alebrije creature. It has already been stated that our framework works closely with the artistic definition of the characters, therefore it is important to remember that such definitions *must* exist before attempting to implement the creature. The artist must define concepts like appearances, moods, attitudes and personality; a unique way of walking, gazing or waving its hand; the senses possessed by the character (is it blind? can it smell and/or hear?), characteristic actions and reflexes that add to the personality (the character might have a tendency to scratch its head when thinking, cover its mouth when scared or being continuously checking the time when nervous). This conceptual design is very similar to the one created for animated characters in feature films.

One of the main differences between ANIMUS and other frameworks consists on its modularity and the way its functionality is distributed among its different elements. Fig. 3 shows the three main layers of our framework. Stimuli from sensors in the immersive environment, other creatures, and world objects enter the creature through the *Perception* layer. The *Cognition* layer handles *high level* cognitive processes and constitutes the actual *brain* of the creature. It is here where temporal memory, personality, emotions and goals are modeled and applied. Finally, the *Expression* layer is the animation engine in charge of showing the inner state of the creature to the audience. Each of these layers will be now analyzed with more details.

5.1 Perception Layer

We have implemented an *early selective attention and perception layer* that performs the following tasks:

1. Serves as an entry-point for the world, enabling the character to be aware of its environment within the bounds of its perceptual definitions (if the creature is supposed to be deaf, no sound will be sensed even when other creatures emmit it. If its hearing abilities are finer than those of others, sound resolution will also be more accurate).

2. Segments, discriminates and prioritizes information to bias the attention of the creature toward salient stimuli *without* needing to understand the meaning or nature of the perceived information.

3. Implements reflexes as an *automatic* response to certain stimuli.

4. Performs low-level conditioned *learning*

Most architectures perform semantic analysis of information as data is perceived [6, 20, 19, 24, 2] in order to generate an output. Our architecture proposes a method for prioritizing information *before* any cognitive or semantic analysis is applied. This presents three major advantages: first, because no complex AI algorithms are involved in the attention shifting mechanisms, it is possible to implement *instinctive behavior* and *automatic reflex response* in a simple way, which greatly adds to the believability and realism of the character (if the creature touches a very hot surface, it moves its hand away before having to decide that burning its hand is not desirable. Suddenly bringing a strong source of light on its face might cause it to turn away or close its eyes). Second, it optimizes subsequent analysis of the information by biasing the creature's attention toward what seems to be interesting about the world (as an example, a loud noise or an object that suddenly begins to move will automatically call the attention of the creature, probably making it turn to face the source of distraction). Third, an interesting type of *low-level conditioned* learning can be produced. A creature that burns its hand when touching a hot surface instinctively avoids touching it again, or jumps back if suddenly discovers that it is standing over the same surface, even if it is not hot. This system is similar to conditioned learning observed in humans and animals.

The possibility of discriminating information without having to analyze its meaning was inspired from the concepts of *early* and *late* selection on human perceptual processing[8]. Early selection suggests that attention is directed before information is actually understood by blocking some stimuli and preserving other. Late selection states that attention comes from the correct evaluation of the information by high-level cognition systems. It was later proposed that early selection would indeed perform *a priori* discrimination of the information, not by blocking, but by degrading or attenuating stimuli before it would reach higher levels of cognition. ANIMUS creatures can get distracted or center their attention on the various events of an immersive environment instinctively because it is imprinted on the constitution of their Perception Layers, which reflect their artistic and conceptual design, as we will explain later. The architecture of the Perception Layer is shown in Fig. 4. We now briefly describe its functionality. For a more detailed explanation please refer to Torres and Boulanger[23].

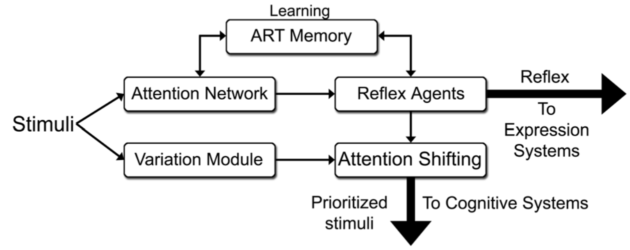

Figure 4: Components of the Perception Layer

Let N be the total number of virtual creatures in the inmersive environment. At a given time t, the world blackboard contains the stimuli set $S = \{s_1, ..., s_n\}, 1 \leq n \leq N$, that is, one stimuli for each creature that produces an output. A given creature $C_i, 1 \leq i \leq N$ reads a stimuli set $S'_i \in S$ such that $s_i \notin S'_i$ which becomes the input to its Perception Layer. The output will be a prioritized stimuli set S''_i based on the results produced by the layer. At the same time, the layer will perform low level learning and reflex triggering.

Each stimuli in S'_i is send into two different channels: an *Attention Network* and a *Variation Module*. The Attention Network is inspired by Koch[15] and contains a set of M carefully chosen *attention neurons*. An attention neuron j contains an initial weight $w_j, 0 \leq w_j \leq 1$ and a threshold value θ_j initially set to 0, and is sensible to a single feature that might or might not be present in the input stimuli. When a stimuli enters the network, neurons react to their matching stimulus with net input $v_j(t) = x_j(t)w_j$, where $x_j(t)$ is the input magnitude of stimulus j at time t, and the output is the excitation level determined by the sigmoid function $I_j = (v_j(t), \theta_j)$ if $v_j(t) \leq \theta_j$ and $I_j = 1$ otherwise. Some features can only be produced by combination of other features (like pain or some specific kind of pleasure, for example), therefore the output of some neurons will constitute the input to others. Both the weight and threshold of each neuron will change as a function of time and input magnitude, giving the neuron short–term memory characteristics. The neuron threshold θ_j adapts to perceived stimulus with a learning rate $\eta_j, 0 \leq \eta_j \leq 1$, and the weight w_j decreases with a function $\gamma(w_j(t), \sigma_j)$ (usually a parabolic or sigmoid decay function) in a way that, if input remains constant, its output will converge to 0 in the next σ_j steps. In other words, *the creature's senses adapt to the magnitude of a constant stimulus, which will cease to elicit attention after a certain time.* The output of the Attention Network is the vector containing the excitation level of all excited neurons and tells how *interesting* in terms of magnitude are each one of the stimulus contained in the input stimuli.

The Variation Module is based on the *Inhibition of Return*[8] effect observed in human perception, which simply states that recently attended stimuli is inhibited over time. We say that a stimulus belonging to a specific creature at time t and $t-1$ shows variation if the absolute difference of their magnitude is greater than a certain threshold. The output of the variation module sensible to this stimulus is 1 if the *variation rate* is less than a hand coded frequency factor, and 0 otherwise. This inhibits interest in repetitive phenomena like the sound of a ticking clock, and highlights more unique events like a radio that suddenly stops playing. Please note that while the Attention Network measures *intensity* by comparing input magnitude against a threshold, the Variation Module detects *change* regardless of its magnitude. An oscillating signal will make both the Variation and Attention modules lose interest after a while, but if the signal becomes stronger, the Attention module will recognize the change, while the variation module will remain the same. The output of the Variation Module is considered along with that of the Attention Network in order to attenuate the input stimuli, prioritize all input stimuli in S'_i and suggest an attention target to the high level cognitive systems.

The Reflex Triggering module is a set of *Reflex Agents*. A Reflex Agent is an independent process formed by a reflex stimuli and a reflex function. Reflex agents try to match their reflex stimuli against the perceived stimuli (a heat–pain agent, for instance, monitors output from heat–sensible neurons belonging to different parts of the body). If they are similar, the original magnitudes of the input stimuli become the parameters to the reflex function. The reflex function reviews those parameters and constructs an adequate response (taking the last example, if heat is detected with magnitude greater than a tolerated amount, the heat–pain agent produces a reaction oriented to get the affected body part out of danger). Reflex agents have direct access to the Expression Layer, so ANIMUS creatures execute reflex reactions *before* the stimuli gets to their *brain* (our pain agent decides that the hand is getting burned and orders the body to move it away from the fire before the creature's brain *realizes* what is happening). Different creatures may have different reactions to the same stimuli, according to their conceptual design.

The ART[3] Memory module holds a unique ART model for each creature that has triggered a reflex. When the reflex is produced, a corresponding ART model synthesizes a sample of the original stimuli. If a creature produces some stimuli that results similar to the one that elicited the reflex, a similar reaction with attenuated magnitude can be triggered. To illustrate this with an example consider the case of training a virtual dog. Imagine that the dog contains a reflex agent that produces joy when the creature is fed (our dog can show joy as a reflex by wagging its tail). If the user feeds the animal and sounds a bell at the same time, the ART module synthesizes a stimuli that contains both the sound of the bell and the sensation of the food. After some reinforcement made by repeating the same stimuli, the user can cause the dog to wag its tail by just sounding the bell. This happens because the bell's stimuli matches to certain extent the sample synthesized when feeding occurs. In other words, the dog "suspects" it will be fed.

The functionality of this conditioned learning module is inspired in work by Mumford[16] and assumes that events in the virtual world are not arbitrarily produced but are due to hidden variables. Synthesizing coded signals for the creation of comparison points in a feed-forward, feed-backward loop create knowledge about these hidden variables and *teach* the character to react to events in the inmersive environment.

[3]Adaptive Resonance Theory. For more information please refer to [3]

Creating the perception layer for Alebrije is quite simple. Since our creature is sensible to sound intensity and position, we can create a minimalistic Attention Network module with a neuron devoted to sensing sound, and 3 neurons to detect the x, y and z coordinates of sensed characters or the user inside the CAVE. We provide Alebrije with a simple reflex agent sensible to noise: If the user or other Alebrije makes a strong noise, the agent uses the coordinates to turn and face the source. The cognition layer can choose to remain tracking the source of the noise or turn to face other interesting stimuli.

5.2 Cognition Layer

The Cognition Layer is the place where models for personality, goals and emotions are put to work to create a high level behavior for the character. At this point we can take advantage of the prioritized information provided by the perception layer analyze the semantics of data that either seems to be important or has already been chosen as relevant by our cognitive systems; remember that nothing is preventing us from directing the attention toward any objective chosen by the creature's mind. As an example, imagine that Alebrije decides to track the movements of the user. If other Alebrije suddenly sings or generates other forms of noise, our creature might be distracted for an instant and turn to face it because of the reflex triggered by the Perception Layer, but will immediately return to its original task of facing the user in the inmersive environment. It is desirable that such conflict occurs because it favors a more natural and realistic behavior; in fact, frustration or anger as an emotion might even be elicited as a response to an scenario where too many creatures are interrupting our Alebrije and preventing it from paying attention to the user.

This layer is quite flexible in the sense that it provides a base for testing many different models. We might use a general rule–based approach for simple animal–like creatures like Alebrije, we can consider other models like the EM architecture proposed by Reilly[18], a more complex plan–based architecture like the one used in Gratch[10], an adaptation for real physical robots like the ones described in Velázquez[24] and in general any kind of model that best adapts to the needs of our character. In other words, the Perception and Expression layers provide a propitious body and resources for the Cognition model to be easily adapted.

We are currently testing several models for this layer. As different creatures have different requirements, it is hard (and not necessarily appropriate) to find a *one–size–fits–all* cognition model. Nevertheless, we provide three different tools that help considerably in the task of controlling high–level cognition operations. These are the Driver System, a Dictionary of Characters and a Temporal Memory model.

5.2.1 Driver System

Many emotional and behavioral architectures are substantially based on the use of drivers[4]. A driver is essentially an accumulator that keeps track of a certain magnitude or represent the state of a particular feature. Drivers can be used alone or within a weighted network, they can be controlled

[4]Examples include Velázquez[24], *The Sims*[1], and Kline[14]

Table 1: Callback messages for ANIMUS drivers

Message	Driver Magnitude Events
DM_THRESHOLD	Surpasses threshold
DM_SIGNCHANGE	Shifts sign
DM_UPDOWN	Increasing to decreasing
DM_DOWNUP	Decreasing to increasing
DM_UP	Increases
DM_DOWN	Decreases
DM_HOMEOSTASIS	Is within homeostasis value

by an external function or possess their own grow and decay behavior. They can simulate emotions like joy, anger or frustration, sensations like hunger, pain or tiredness, and virtually anything that can be measured. An excellent example of a practical driver system is shown by the creatures of *The Sims*[1]. In this videogame, virtual characters have a set of needs that increase as a function of time or other factors, and a wide array of actions the user can perform to satisfy drivers with dangerously high magnitudes (the user must feed a hungry character, or entertain a bored one). More complex driver–based systems can be seen in [24] and [14].

The ANIMUS framework provides a flexible driver implementation. A driver is defined as $D_i(\phi_i, \upsilon_i, \theta_i)$ where ϕ_i is the *homeostasis* state or equilibrium value, υ_i is an *umbral value* that defines an area around ϕ_i in which the magnitude of the driver is considered to be normal or desirable, and θ_i is a threshold value that usually indicates a maximum tolerance point. If drivers are interconnected in a network or group, the magnitude of the driver D_i at time t is simply written as:

$$D_{it} = \sum_{n} (D_{nt} \times w_{ni}) \qquad (1)$$

where D_{nt} is the magnitude of drive D_n at time t, w_{ni} is the weight from drive D_n to D_i and n denotes all drivers connected to D_i.

An important difference with respect to other systems is that our drivers are programmable and include a callback messaging system architecture that make it straightforward for the cognitive systems to react to different events, without having to be constantly monitoring their magnitudes. Table 1 shows some of the messages that can be set in a generic driver.

Messaging simplifies code writing and scripting because instead of searching for particular conditions we basically wait to be notified when they happen, giving our architecture an event–oriented approach. As an example, consider a function that makes Alebrije look for food when a certain hunger driver reaches its threshold. The driver can be set to automatically call this function or add some *look–for–food* task to the character's agenda whenever the creature is hungry. This driver architecture also brings the possibility of creating *specialized* agents that receive events from certain drivers and control their behavior in an interesting way, instead of just letting the magnitude of a driver linearly increase or decay by itself in time. Consider a driver that measures *interest* in a certain activity like hunting a prey. If the character is not making any progress (the prey is hidden somewhere and won't come out) an agent might decrease or increase interest as a function of variables in the hunting scenario (like hunger level, probability of catching the pray, observed behavior of the pray...) instead of just letting it linearly decrease as long as the prey is hiding until the character loses interest.

5.2.2 *Dictionary of Characters*

To favor high–level knowledge representation and modeling, two special data structures are provided. They consist of *Entities* and *Characters*. An Entity is anything that our creature recognizes as an individual object (living or otherwise, user included) inside the immersive virtual world. A Character is defined here as the *particularized* version of an Entity from the point of view of our creature. In other words, an Entity is like the picture of a creature, which is always the same for everyone, while a Character is the picture plus our own knowledge, beliefs, emotions, annotations and experience about such creature, which might (and hopefully will) change in time. In a simulation with two Alebrije creatures and the human user, both creatures recognize the user as the same Entity, but each one might have a different concept of him/her given their personal experience and therefore react in a different way to his/her presence. This simple principle enables two otherwise exact instances of the same object (Alebrije) to have different knowledge and behavior, and create their very own representation and beliefs about the world.

The dictionary of characters provides ANIMUS creatures with a database that stores particularized knowledge about other creatures and human users[5]. In practice, references to entries in the dictionary of characters can be used in other cognitive processes to represent creatures, as demonstrated in section 5.2.3.

5.2.3 *Temporal Memory System*

Inspiration for a temporal memory object comes from Mumford[16] and Damasio[5] and is mainly based on the theory that knowledge can be acquired, represented, and maintained by coding temporal signals of events that happen in the world. A basic representation of the memory module elements is illustrated in Fig. 5.

We reconsider the statement done in section 5.1 related to the theory that events in the environment are not arbitrarily produced, but due to specific hidden variables. To find such variables and reconstruct the world state we create a model that allows to *learn from experience*, detecting patterns as a function of both time and stimuli data. The model itself is very simple: For each known character[6], we synthesize perceived stimuli in a progressive timeline structure, adding an entry each time the world is sensed. We can now analyze two kinds of information: the set of stimuli belonging to all perceived characters at some point in time (called a *time frame*, which also represents the known state of the world at that time) or the behavior history for specific Characters within a specific lapse of time. Specialized pattern analysis

[5]From the point of view of any ANIMUS creature, the user is *also* a creature

[6]Here, we refer to the concept of *Character* defined in section 5.2.2.

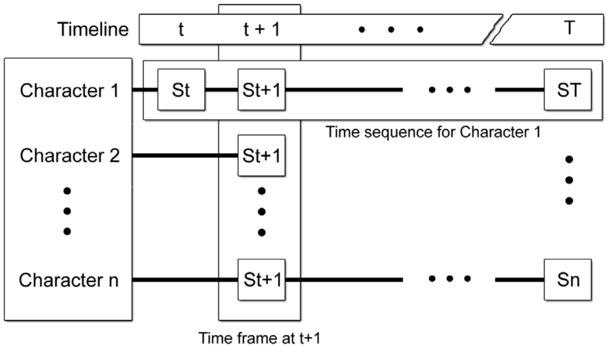

Figure 5: The ANIMUS Temporal Memory model

agents can read the temporal memory both vertically and horizontally and respond questions like: which Characters where present when I entered the room? Given observed events, what is the most probable thing to happen if I perform certain action? What does creature X do every time Y sings? What are the common characteristics of the world and other creatures every time the user seems to be angry? What is the average speed of creature Y when it is walking over mud? What is the set of conditions that seem to be required for some phenomena to happen? High–level cognition systems can take advantage of information stored in temporal memory objects to create knowledge and learn about the world and its creatures.

Let us try an example: Imagine that our virtual character finds that the user is sad and asks itself what to do to cheer him/her up. Searching through its temporal memory, it recalls 6 different occasions in which the user was happy. Comparing those 6 time frames, it discovers that common factors include the presence of object X and its position in a place of the virtual world. It also discovers that this candidate condition is not currently met, therefore tries to create it by looking for object X and placing it in its correct place. If the user is now happy, our character has succeeded; otherwise different analysis techniques can be applied, like reviewing its own behavior sequence and that of other creatures prior to the time frames where the user was happy.

Stimuli for a given creature might or might not be present at a given time, and the system's memory constraints will determine how many time frames to keep. Multiple temporal memory objects can be constructed, updated and consulted in real time for organizing information, and given physical memory constraints an active window may keep only the last n time frames and store information that proves to be relevant in other less frequently updated temporal memory modules. Temporal memory modules are powerful structures that also allow interesting operations like interpolation and extrapolation of stimulus. Alebrije might try to predict the behavior of another creature by extrapolating past similar actions, or use interpolation to deduct what happened between two points in time where its behavior was unknown.

5.3 Expression Layer

Probably the most critical aspect about creating a life–like creature resides on its animation. A perfect architecture will be hopelessly doomed without an effective animation system that conveys to the audience the character's internal emotions, thoughts, beliefs, personality, goals and desires. Nobody knows this better than traditional character animators. Good animators can take almost *any* sort of creature and make it interesting, whereas a poor animator will fail to interest the audience into the most carefully designed character. The simple act of waving a hand must be charged by the character's personality and style.

Thomas and Johnston[22] mention on their classic masterpiece, *The Illusion of Life*, 12 fundamental principles of animation that are followed by today's classic and computer animators when bringing to life the impressive creatures that we see in today's feature films. Such principles have nothing to do with real physics, mechanics or kinematics, but with artistic elements closely related to the character's personality, role, intention, and emotional state. Any well animated character succeeds at the questions proposed by Ommanney[17] for real actors: are the characters interesting, lifelike, and vivid? Do you become emotionally involved with them? Do the gestures and movements seem sincere, convincing, clear and properly motivated? Does all of the action help to delineate the characters and their situation for you?

Common approaches to 3D animation are realtime physics–based controllers and graphs of keyframed animation sequences. The first case is expensive because one needs to perform operations like inverse kinematics and apply mechanical and physical constraints over the various degrees of freedom of the creature's body at each frame. It also fails to produce an interesting animation since mathematics can hardly encode the artistic requirements mentioned in previous paragraphs. This problem is partially solved in the second case (widely used in video games), where a number of animation sequences are created *a priori* by character animators that carefully handle artistic constraints and make sure the character's movement is adequate to its unique style and personality. The motor system concatenates or selects different sequences of these *canned* animations to make the character react to different situations in the environment. The major disadvantage of this approach is that such sequences are unmodifiable and play in exactly the same way every time. After a short while they become repetitive, drastically eliminating the character's believability.

A better option is the use of *pose–based* or *blend–based* motor systems. The idea is similar to the techniques used in traditional animation: Two reference poses are drawn by the artist to picture the character in a particular situation. Intermediate poses are then added to create the animation. The flow of motion is not necessarily linear and the artist has complete control over the expression of the character. Pose–based motor systems contain a set of artistically designed poses. At a given time, two or more poses can be blended together to create the animation. Parameters in the blending algorithm and the choice of reference poses give the programmer flexibility to create new animations in realtime that adapt to different situations, generating flexible animation sequences while preserving the character's personality and style. Excellent examples and more detailed information systems can be found in Downie[7] and Rose[4].

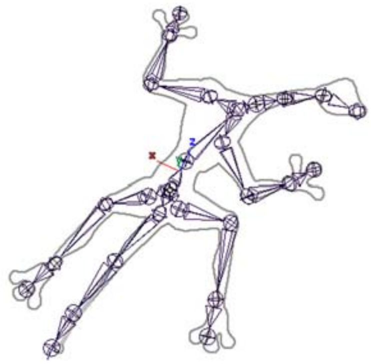

Figure 6: Skeleton structure for Alebrije

The ANIMUS pose–based motor system is composed by the creature's geometry (which can be modeled using any commercial package), an internal skeleton structure and an organized set of bone "snapshots". A skeleton is a hierarchical set of transformations (bones) that affect certain regions of the creature's geometry. Instead of taking a snapshot of the whole geometry, we only record the skeleton's transformations. Fig. 6 shows the skeleton for Alebrije's body.

Each pose is assigned a numeric *weight* and placed inside an uniquely named *node*, which keeps a collection of poses that are usually related. For example, we can create a node called "Walking" that contains five poses used to create walking cycles. Each node contains at least one weighted pose. When creating an animation sequence, we simply select two poses (an initial and final state) by indicating the name of their nodes and a certain weight. We can select poses from the same or different nodes, and the weight may not necessarily correspond to an existing pose, in which case a new *virtual pose* is created by interpolation in realtime as shown in Fig. 7. The skeleton is blended across selected poses to animate the creature. If an animation needs to be changed or *redirected* to another pose while blending, the current state of the skeleton is considered as the initial pose of the second animation and gets blended against the *new* final pose. Because of pose redirection and automatic virtual pose generation, new animations can be created in realtime and existing sequences are flexible and may change given the environment circumstances. In addition, the main skeleton of the creature can be divided into smaller bone chains (i.e., left arm, right arm, and such) and poses can specify transformation for particular sets of chains. this way, more than one blended animation can be executed at the same time to animate different parts of the body, like walking while waving the hand. A limitation to this approach is that reference poses must have the same bone chains and two concurrent animation threads must not be allowed to blend the same chain.

Pose weights can indicate features with relevant meaning. For example, a "Hand_reach" node might have two poses with weights of 10 and 40 respectively. The first pose represents the arm of the creature stretched at 10 cm of its body, while in the second pose the arm reaches out at 40 cm. If the creature detects and object at 37 cm, an animation directive like `grab_object(37)` might blend from the character's current state to a virtual pose that effectively moves the arm to the desired place, *simulating* inverse kinematics and maintaining the character's style of movement.

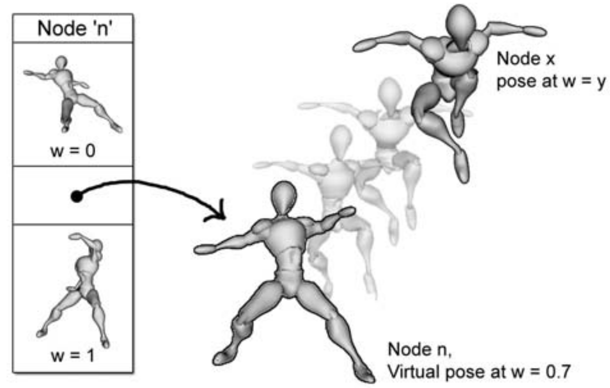

Figure 7: Blended animation from $Node = x, w = y$ to $Node = n, w = 0.7$. As n only has two poses (at $w = 0$ and $w = 1$), a virtual pose at $w = 0.7$ is created

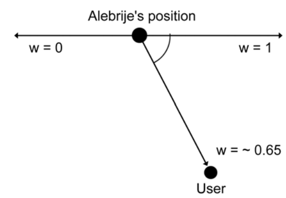

Figure 8: Using the Alebrije and user's position to calculate weight for a gazing virtual pose

It is desired to give Alebrije the ability to turn and face the user as he/she moves inside the CAVE. As Alebrije turns around, its whole body must move to enhance the intention and expression of the gesture. To create this functionality we define a node called "Gaze" with two poses: one with Alebrije facing left ($w = 0$) and another facing right ($w = 1$). When gazing at the user, it is easy to find the angle between a vector defined by the position of the user and Alebrije. From there we map the value to a $0 - 1$ range and simply blend from Alebrije's current position to the obtained weight as shown in Fig. 8. It is important to mention that, even when the obtained information could be used to simply create a rotation matrix that would make a geometric object face the user, the virtual pose created by the engine is not just facing in the right direction: the whole body will be transformed to reflect Alebrije's unique "gazing" style.

6. CONCLUSIONS AND FUTURE WORK

We have described a new system to design and create virtual creatures, which are capable of giving an illusion of life within an immersive environment. Dynamics of information flow and basic information units were also defined. We have described the three main modules of the ANIMUS framework and showed how they contribute to the creation of believable creatures with artistic behavior and personality. We have outlined the advantages of our early perception and attention system and exemplified its constitution with an example. We have described the three high–level cognition tools provided by the Cognition Module, namely the Driver system, the Dictionary of Characters and the Temporal Memory module, which is particularly useful for advanced time and magnitude–based pattern analysis. The Pose–based engine of the Expression Module was explained and its construction for a simple task (making Alebrije gaze at the user as he/she moves within the CAVE system) served as an example of its functionality.

Current research is focusing on the design of a complementary architecture for the Cognition Module to integrate an emotion model, a social behavior model and a goal–planning agent similar to the one described by Gratch[9]. At this time, several demos with Alebrije and other similar creatures are being prepared for public evaluation, these creatures are created with the ANIMUS framework and contain simple rule–based behavior in their Cognition Modules in a fashion similar to Funge[12]. Future research will continue to develop advanced tools for high–level cognition based on mind theory and research on biological systems.

We are about to apply these concepts to videogames as testbed for the ANIMUS framework. Several special features like scripting and character design tools might be implemented to facilitate the creation and evaluation of interactive creatures in this domain.

7. REFERENCES

[1] E. Arts. The sims©.

[2] J. Bates and Others. Integrating reactivity, goals and emotions in a broad agent. *14th Annual Conference of the Cognitive Science Society*, July 1992.

[3] G. Carpenter and S. Grossberg. Adaptive resonance theory (art). *The Handbook of Brain Theory and Neural Networks, MIT Press*, pages 79–85, 1995.

[4] B. B. Charles Rose and M. F. Cohen. Verbs and adverbs: Multidimensional motion interpolation using radial basis functions. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.

[5] A. R. Damasio and H. Damasio. Cortical systems for retrieval of concrete knowledge: The convergence zone framework. *Large–Scale Theories of the Brain, MIT Press*, pages 61–74, 1995.

[6] R. B. Damian Isla et al. A layered brain architecture for synthetic creatures. *International Joint Conferences on Artificial Intelligence*, August 2001.

[7] M. Downie. Animation, music: The music and movement of synthetic characters. Master's thesis, Media Arts and Sciences, MIT, 2001.

[8] M. Gazzaniga and Others. *Cognitive Neuroscience, The Biology of The Mind*. Norton, New York, 1998.

[9] J. Gratch. Modeling the interplay between emotion and decision–making. *Ninth Conference on Computer Generated Forces and Behavioral Representation*, 2000.

[10] J. Gratch and S. Marsella. Modeling emotions in the mission rehearsal exercise. *10th Conference on Computer Generated Forces and Behavioral Representation*, May 2001.

[11] D. Isla and B. Blumberg. New challenges for character–based ai for games. *AAAI Spring Symposium on AI and Interactive Entertainment*, March 2002.

[12] X. T. John Funge and D. Terzopoulos. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. *ACM Computer Graphics Procedures*, 1999.

[13] B. L. Joseph Bates and W. S. Reilly. An architecture for action, emotion and social behavior. Technical report, School of Computer Science, Carnegie Mellon University, 1992.

[14] C. Kline and B. Blumberg. The art and science of synthetic character design. *Proceedings of the AISB Symposium on AI and Creativity in Entertainment and Visual Art*, 1999.

[15] C. Koch and F. Crick. Some further ideas regarding the neuronal basis of awareness. *Large–Scale Theories of the Brain, MIT Press*, pages 93–109, 1995.

[16] D. Mumford. Neuronal architectures for pattern-theoretic problems. *Large–Scale Theories of the Brain, MIT Press*, pages 125–152, 1995.

[17] Ommanney and Schanker. *The Stage and the School*. McGraw-Hill, New York, 1972.

[18] W. S. N. Reilly. *Believable Social and Emotional Agents*. PhD thesis, Carnegie Mellon University, 1996.

[19] P. Rizzo et al. Personality–driven social behaviors in believable agents. *AAAI Fall Symposium on Socially Intelligent Agents*, 1997.

[20] D. Rousseau and B. Hayes-Roth. Improvisational synthetic actors with flexible personalities. Technical report, Stanford University, 1997.

[21] D. Terzopoulus and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, 1993.

[22] F. Thomas and O. Johnston. *Disney Animation: The Illusion of Life*. Abbeville Press, New York, 1981.

[23] D. Torres and P. Boulanger. A perception and selective attention system for synthetic creatures. *To appear in proceedings of the Third International Symposium On Smart Graphics*, March 2003.

[24] J. D. Velázquez. When robots weep: Emotional memories and decision–making. *AAAI Proceedings*, 1998.