Smart 3D Computer Animation based on Artificial Life

Francisco J. Seron [1], Diego Gutierrez [1], Alfredo Pina[2]

[1] Universidad de Zaragoza, CPS, c/ María de Luna
E-50015, Zaragoza, Spain, seron@posta.unizar.es, diegog@phoebe.cps.unizar.es

[2] Universidad Pública de Navarra, Campus de Arrosadía s/n
E-31006 Pamplona, Spain, pina@unavarra.es

Abstract. This paper describes the ALVW system, a high-level system for producing smart behavior-based 3D Computer Animation. The system allows the design and simulation of virtual worlds, environments and their inhabitants. Once the simulation of the virtual ecosystem is run, the results are transferred to a commercial 3D program, where a realistic animation can be produced based on the transferred data. The concatenation of all these processes allows us to produce a realistic 3D Computer Animation showing what happens in the artificial ecosystem, using the best of both worlds: behavior-modeling techniques plus all the capabilities of existing 3D software. As a pilot project, we have developed Kukasim, a simulator that models a cockroach living in a kitchen, and produced a four-minute 3D animation based on the simulation data obtained from it.

1 Introduction and Background

We present here ALVW, a 3D animation system based on behavior modeling techniques, that acts as an interface between the animator and an existing commercial 3D package, and produces realistic animation without keyframing. Obviously, the key to a realistic animation starts with the correct modeling of the behavior of the characters, creating believable synthetic actors.

In recent years there has been a great evolution in the techniques used to model behavior in Computer Animation. The tools for modeling behavior have traditionally been connected to AI although recently they have been extensively used in relation to Artificial Life. Actors are able to show complex behaviors that emerge from the combination of several simple ones.

The ALVW system is versatile enough and can work by itself, without the explicit need of connection to a 3D software, being useful in a number of other different fields. In fact, there is an increasing interest from ecologists, physicians, biologists, etc… in the use of biologically inspired computer based systems for simulation and evolution of natural systems [1]. This interest is present in the Artificial Life domain, where computer scientists and biologists (among other scientists) have been co-operating over the last several years. As in the case of 3D animation, one of the main problems is to provide powerful and easy-to-use tools to accomplish the demanded tasks, so that the user (a biologist, entomologist…) may use these tools with little or no assistance from the designer [2].

Computer animation is a field that comprises a broad range of techniques, which have little in common in most cases. For a more complete work on the state of the art of computer animation the reader can refer to [3][4].

Synthetic actors not only have to move in the most realistic possible way, according to the laws of physics, but they have to be able to accomplish tasks and take decisions as well. Going from reactive behavior to cognitive behavior (from a pure reactive behavior to a more complex one where the actor responds to his/her wishes and experience, while having the capability of learning), has been a huge step forward in the behavior modeling area. On the other hand, the ever-increasing computing power, along with more advanced parallelization techniques, have allowed scientists to face the complexity of this type of animation system.

We outline in the rest of this section the most relevant works focusing on behavior modeling for computer animation.

The first works that provide animation systems with behavior and autonomy characteristics (applied to flock of birds and a school of fish) is owed to [5]. It is the first time that a bottom-up approach is realized. Another pioneering work in human behavior simulation is [6], where the "Human Factory" animation system is presented. Several other works arise from this moment.

A classic reference for behavior modeling both in robotics and animation has been the works of [7], [8] and [9]. An action-perception scheme is proposed, as opposed to the classic planning scheme.

Several perception-action schemas have been used to model behavior for animation of fishes [10][11], for animation of autonomous animated characters [12], for synthetic actors [13], for agent based actors [14], for designing synthetic actors that express emotions [15] or that use natural language to communicate [16], and for guided actors and interactive autonomous actors on the net [17]. An example of the kind of application of these systems can be found in [18] where a real-time, interactive virtual oceanarium based on a perception-action scheme is presented. Different kinds of fish are modeled, each with unique simulated perception systems that match their real-world counterparts. There are a general set of behaviors, such as fleeing, schooling or feeding, that can influence the fish's inner state, for example making it less hungry after feeding. Behaviors imply locomotive actions, ruled by an underlying kinematic model. In order to meet real-time constraints, some aspects of the real world creatures, such as the ability to learn, needed to be left out of the simulation.

Other lines of work in the field of synthetic actors study the art of acting by real actors, to then create believable synthetic actors. Hayes-Roth et al. [19] study the personality in a given context of artificial agents that work as actors. From this point of view, the authors define concepts such as role, behavior or improvisation for synthetic actors. Other works related to this area are the Oz project [20], with the goal of building interactive story worlds, or Teatrix which is a collaborative virtual environment where virtual actors play roles [21]. Also the project IMPROV [22] proposes the use of scripts to provide a group of behaviors to a given actor.

Cognitive modeling is introduced in [23], with the goal of directing autonomous characters to perform certain tasks, something still difficult to achieve with behavioral modeling. The authors develop a Cognitive Modeling Language (CML), specifying actions, their preconditions and their effects, and goals for the characters.

Artificial Intelligence along with Artificial Life techniques are also starting to be used to express behavior and to provide control to the user. This kind of techniques have been used in modeling both complex behavior of synthetic main actors and changing environments for such actors [24]. Based on these works, [25] describes a system designed for automating character animation and producing photorealistic 3D animation.

In [26] the CreatureViewer system for designing 3D actors and complex behavior modules is described. The approach combines Artificial Intelligence and multiagent systems, and the design includes appearance, physical abilities and animations.

The ALVW system presents the novelty of combining the best of two worlds: behavior modeling for synthetic actors and 3D commercial software. The combination has proven to be a powerful one: it takes advantage of the enormous capabilities of existing 3D software while providing an intuitive interface for adding behavior modeling to it, something commercial programs lack.

2. Behavior modeling for synthetic actors

2.1 The ALVW system

The system for modeling Artificial Life and Virtual Worlds (ALVW) is a generic platform that allows the user to design an environment as well as the inhabitants who may live within it.

The system is divided in two parts. The first module deals with the creation of a virtual ENVironment (ENV), composed of Special Elements, Dynamic and Static Objects. The second module is specialized in modeling the Behavior of the Inhabitants (actors in this case) (BIN) and is based on a perception-analysis-reaction loop.

Inhabitants (principal actors) are designed and controlled by a perception-analysis-reaction agent (based on a Fuzzy Expert System and on an Action Selection Mechanism), providing an autonomous and adaptive behavior. In the environment, the Special Elements (extra actors) are controlled by Genetic Algorithms. Static and dynamic objects use time varying attributes and procedural motion.The behavior of an inhabitant responds to his own perception of the environment. As a result of the assessment of sensor values (FES module) a computational model determines which behavior should be activated. A selection algorithm determines among all the candidate behaviors which one is preferred. As a consequence several motor commands are executed reflecting in the environment the chosen behavior of the actor (see figure 1).

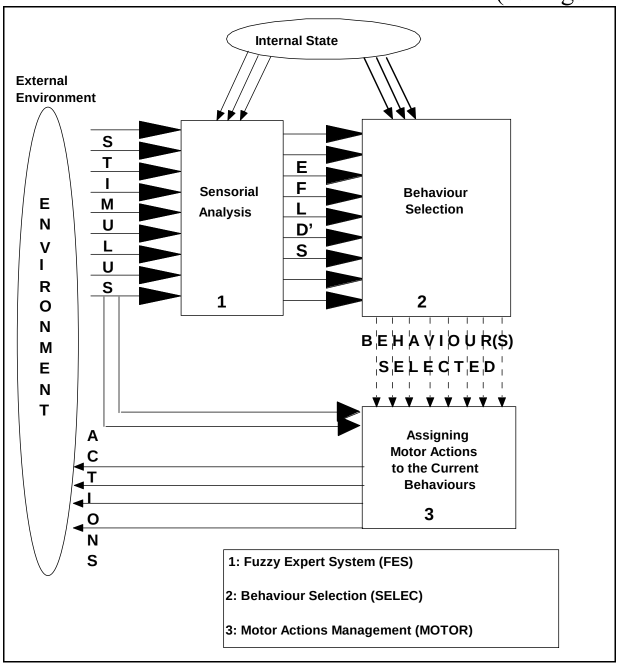

Figure 1. Detail of the BIN module.

2.2 How the Animator defines behavior for principal actors

When the user wants to set the global behavior for a principal actor, the following stages have to be accomplished:

- Specifying sensors
- Specifying parameters for the fuzzy expert system
- Specifying behaviors. Every behavior may be of one of the 3 main types (normal, opportune, simultaneous) which in turn may be composed of further sub-behaviors.
- Specifying parameters for the computational model
- Specifying motor commands

At the current stage, the system provides the animator with a basic set of sensors and motor commands. The main task he/she has to accomplish is to define the set of behaviors of an actor and specifying parameters for the fuzzy expert system and for the computational model of the behaviors.

2.2.1.- Different types of behavior

We distinguish between several types of behavior, which in turn may be composed of other behaviors. We distinguish between several types of behaviors, which in turn may be composed of other behaviors: Normal behaviors (the main block of behaviors an inhabitant uses), Opportune behaviors (they interrupt a normal behavior to give way to the realization of an advantageous one) and Simultaneous behaviors (parallel to the activation of a compatible normal behavior).

Normal and Opportune behaviors can be composed of one or more sub-behaviors which in turn may be either temporal (activation for a finite amount of time) or conditional (activation lasts until the final condition is fulfilled). Every type of behavior may have one or more associated motor commands according to the function that should be performed. The main motor commands of a behavior are called maintenance motor commands; other motor commands are used to start or to end a change in behavior.

Every behavior has an associated computational mechanism which gives it a value, the degree of activation. A selection algorithm decides which behavior is executed at each time taking into account this degree, goals, current values of other behaviors, possible inhibitions between behaviors, etc….

2.2.2.- Parameters for the fuzzy expert system

The first aspect the animator has to define are the variables (input and output) and rules of the fuzzy expert system. The syntax for defining them is as follows:

```
INPUTS DEFINITION
 <num> INPUTS
 {<name> <num_input> <num_linguistic_var>
 { <name_ linguistic_var> <value1> < value2> < value3> < value4> } }
 INPUTS DEFINITION
 <num> OUTPUTS
{<name> <num_output> <num_linguistic_var>
{ <name_ linguistic_var> <value1> < value2> < value3> < value4> } }
 RULES DEFINITION
 <num> RULES
 IF        < linguistic_expresion > { AND < linguistic_expresion > }
THEN       < linguistic_expresion > WEIGHT <val_weight> }
```

An input could be the temperature, possible linguistic variables of the temperature could be cold, warm, etc…; an output variable could be ambiance, possible linguistic variables of the ambiance could be bad, good, etc…; a rule could be "IF temperature cold THEN ambiance bad weigh 0,5".

2.2.3.- Behavior selection mechanisms

Every behavior has an associated computational mechanism which gives him a value, the degree of activation. A selection algorithm decides which behavior is executed at every time taking into account this degree, goals, current values of other behaviors, possible inhibitions between behaviors, etc….

The syntax for defining the behaviors is as follows (here we show the case of normal behaviors):

```
    <num> NORMAL BEHAVIORS
{<name_behav> <par1> <par2> <par3> <par4> etc… <par7>
{<num>  SUBbehaviors  // if <par3> is 1
{<name_subbehav> <par1> <par2> <par3> <par4>}}
```

< num> GOALS{<name_goal> <num_goal>}
< num > STIMULUS{<name_stimulus> <num_stimulus>}
<num> OPPORTUNE {<name_oppor> <num_ opport>}
<num> SUMULTANEOUS{< name_simul> <num_ simul>}

All the necessary information needed by the computational model has to be expressed (through the different parameters) in this file. We are working on a graphical interface for helping the animator in the task of defining all these data.

There is a flow of information between all the modules as it is shown in figure 2.

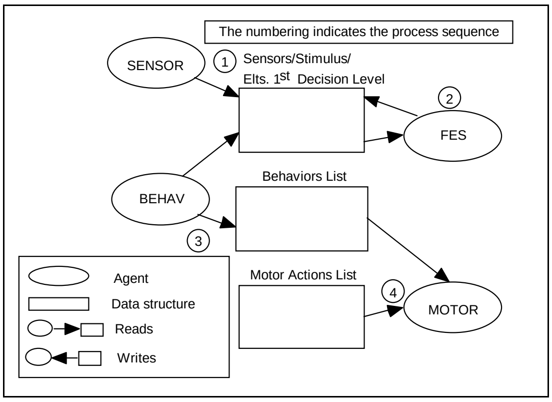

Figure 2. Shared information between the different modules.

A sequential process makes the main loop of perception-analysis-reaction, and the shared information between the different modules is stored on different data structures. The current implementation has been carried out sequentially, but the chosen architecture allows a distributed implementation, using the data structures as a common space of information for exchange and sharing purposes.

2.3 KUKASIM: MODELING A BIOLOGIC SYSTEM

The ALVW system is a generic and open system which may be used to define any virtual scenario and any actor(s) based on a perception-action schema. In the example we present, it has been used to model the behavior of cockroaches. The simulator Kukasim allows the animator to define a scenario (a kitchen which is the cockroach's habitat), to design principal actors (cockroaches) who perceive, think and react in this environment. The use of the simulator enables to follow the evolution of the artificial scenario.

Implementing the system has required the design of simulated sensors (similar to real ones). Special Elements and Static and Dynamic objects provide a dynamic and not predictable scenario. A behavior model controls the cockroaches, which are the principal actors; they interact with the environment and with other actors through several motor commands. Table 1 show the different kinds of elements modeled within *Kukasim*.

Table 1. Different elements modeled in Kukasim.

Special elements	Food, Temperature, Humidity, Danger, Light
Sensors	Light sensors, Chemical sensors, Special Sensors
Behaviors	Staying quiet, Wandering, Looking for food, Reproducing, Escaping from danger, Obstacle detecting/avoiding
Motor Commands	moving towards/away, Setting velocity, Mating, Eating

3 From a virtual world simulator to a realistic 3D computer animation

In this paper, we have tested a novel approach for producing 3D animations that starts with the simulation ran in the Kukasim system. Figure 3 shows an overview of the production of a 3D animation from the stored data of the execution of a simulation. On the left of the figure (step 1) the simulator is running, and any relevant change of the state of the virtual ecosystem is stored on disk. Once a simulation is over, this data is integrated with a complete 3D geometric, visual and motion modeling of the cockroach and its environment, stored in a commercial 3D package (steps 2 and 3).

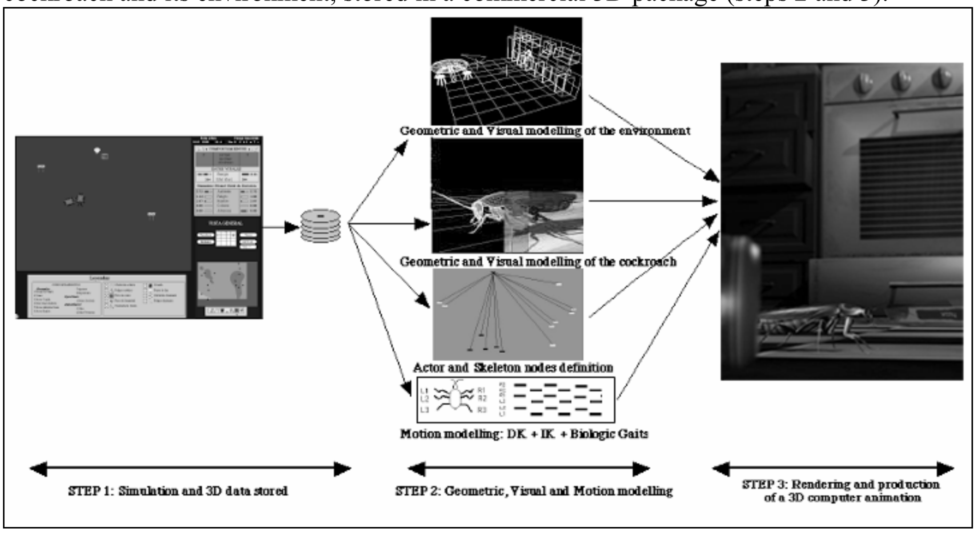

Figure 3. Steps required to produce a 3D computer animation using Kukasim

The animator can interact with the simulator, defining different environments and conditions before running the simulation. In a logical, intuitive way, the animator can place obstacles, food or several dangers. He (or she) can also define local humidity or temperature levels, as well as light sources for the scene. All this elements, plus the geometric data, define the environment the cockroach lives in. The roach's behavior rules themselves can also be modified as explained in the ALVW system. In the current version of the system, the geometric data of the environment can not be interactively changed, and needs to be predefined according to the 3D geometry stored in the commercial animation software. The approach of combining this type of research with a commercial software has also been addressed in [27] [28].

```
FRAME REGISTER
    <num> cockroach
    { cockroach <num_ cockroach >
            { FRAME <num_frame> <x> < y> < z> }}
    <num> GENETIC ELEMENTS
    { GENETIC ELEMENTS <name_gen_elt>
            { FRAME <num_frame> <x> < y> < z> }}
    <num> STATIC OBJECTS
    { STATIC OBJECTS <name_sta_obj>
            { FRAME <num_frame> <x> < y> < z> }}
    <num> DYNAMIC OBJECTS
    { DYNAMIC OBJECTS <name_dyn_ obj >
            { FRAME <num_frame> <x> < y> < z> }}
```

Figure 4. Stored data file syntax.

Once the environment and the actor have been defined, the animator runs the simulation. The system produces the data of the animation based on the A-Life engine, and displays a rough, 2D representation of the results for visual inspection. The data is stored in a file, which syntax is shown in Figure 4. Once the animator is satisfied with the results, the system outputs this data file to the commercial 3D package. In our case, the chosen package is Softimage|3D. The possibility of combining behavior-based animation with the power of state-of-the-art animation software is what sets ALVW apart from other systems.

A four-minute, computer generated animation has been created that proves the feasibility of the proposed approach. (http://giga.cps.unizar.es/Kukasim).

4 Conclusions and future work

We have shown a system that allows to produce smart behavior-based 3D computer animation. The result was a semi-autonomous system that received the results of the *Kukasim* simulation and used that data, plus the character setup built in Softimage 3D, to reproduce the animation in a more realistic way. A video has been produced in order to prove the validity of the system.

The user is able to tune the system expressing behavior for the objects within the environment as well a for the inhabitants. However several lines of work remain open for future development.

In this version, the simulator only outputs the position of the center of gravity of the character on a flat plane. More sophisticated simulations would allow the roach to climb walls or walk on little objects, instead of avoiding them.

All secondary animation needs to be done (either procedurally or keyframed) inside the 3D commercial software. Again, a more advanced simulation and plug-in would be required to add that on top of the main animation.

Looking towards the future, one of the major ideas behind this system is to provide to the user (either a computer animator or an entomologist) with libraries of behaviors, libraries of inhabitants or actors, libraries of special elements and enough control to construct and manipulate life-like artificial words with their life-like artificial inhabitants.

The new computer animation packages should provide this kind of tools to the user. This would require an effort in defining an adequate interface, easy to use and complete.

5 Acknowledgments

We would like to thank Jorge del Pico, co-author of the animation, and Javier Barroso, who helped with the final editing

References

[1] Beer R.D., Quinn R.D., Chiel H.J., Ritzmann R.E., Biologically inspired approaches to Robotics, Commun. ACM, Vol. 40, Number 3, pp. 31-38 (1997)

[2] Kumar D., A new life for AI Artifacts, Intelligence: New Visions of AI in Practice, Vol. 10, No. 3, pp. 13 (Sep. 1999)

[3] Cerezo E., Pina A., Serón F.J., Motion and behavior modelling: state of art and new trends, The Visual Computer, 15, 3, pp. 124-146, (1999).

[4] Pina A., E. Cerezo E., Serón F.J., "Computer Animation: From Avatars to Unrestricted Autonomous Actors (A survey on replication and modelling mechanisms)". Computers & Graphics, Vol. 24, No. 2, section "Surveys", pp. 297-311 (2000)

[5] Reynolds, C.W. "Flocks, Herds, and Schools : A distributed Behavioral Model". Computer Graphics, vol. 21, nª4 (July 1987)

[6] Thalmann D., Magnenat Thalmann N.: The direction of Synthetic Actors in the film Rendez-vous á Montrèal. Computer Graphics & Applications December 1987, 9-19.

[7] Maes P. "Situated Agents can Have Goals" Designing Autonomous Agents: Theory and Practice from Biology to Engineering and back. MIT-Elsevier, (1990)

[8] Maes P. "Modeling Adaptive Autonomous Agents.". Artificial Life VOL 1, Number 1/2 (1994) 135-162.

[9] Maes P. "Artificial life meets Entertainment: Lifelike Autonomous Agents". Communications of the ACM VOL 38, Number 11 (1995) pp.108-114.

[10] Tu X., Terzopoulos D. "Artificial Fishes: Physics, Locomotion ,Perception, Behavior". Computer Graphics Proceedings (SIGGRAPH'94)

[11] Terzopoulos, D., Tu, X., Grzeszczuk, R. "Artificial fishes: autonomous locomotion, perception, behavior and learning in a simulated physical world". Artificial Life, 1 (4), pp. 327-351. Dec. 1994

[12] Blumberg B. "Multi-Level Control for Animated Autonomous Agents: Do the Right Thing...Oh, Not That". Creating Personalities for Synthetic Actors (1997) Trappl R., Petta P. (Eds)

[13] Thalmann D., Noser H., Huang Z.: "Autonomous Virtual Actors based on Virtual Sensors". Creating Personalities for Synthetic Actors (1997), Trappl R., Petta P. (Eds)

[14] Badler N.I., Reich B.D., Webber B.L. "Towards Personalities for Animated Agents with Reactive and Planning Behaviors". Creating Personalities for Synthetic Actors (1997) Trappl R., Petta P. (Eds)

[15] Bates J. “The role of Emotion in Believable Agents”. Communications of the ACM Vol. 37, Number 7 (July 1994) pp.122-125.
[16] Loyall A.B. “Some Requirements and Approaches for natural Language in a Believable Agent”. Creating Personalities for Synthetic Actors (1997), Trappl R., Petta P. (Eds)
[17] Noser H., Pandzic I.S., Capin T.K., Magnenat Thalmann N., Thalmann D. “Playing games through the Virtual Life Network”. Proc. Artificial Life V Nara, Japan (1996)
[18] Fröhlich, T. “The virtual oceanarium”. Communications of the ACM, Vol. 43, Num. 7, pp. 95-101. July 2000
[19] Hayes-Roth B., Van Gent R., Huber D. “Acting in Character”. Creating Personalities for Synthetic Actors (1997) Trappl R., Petta P. (Eds)
[20] Mateas, M. “An Oz-Centric Review of Interactive Drama and Believable Agents”. CMU Report CMU-CS-97-156, Carnegie Mellon University, June 1997
[21] Machado, I., Paiva, A., Prada, R. “Is teh wolf angry or… just hungry?”. Proceedings of AGENTS’01. Montreal, Canada. 2001
[22] Goldberg A. “IMPROV: a system for Real-Time Animation of Behavior-based Interactive Synthetic Actors”. Creating Personalities for Synthetic Actors (1997), Trappl R., Petta P. (Eds)
[23] J. Funge, X. Tu, D. Terzopoulos. "Cognitive Modeling: Knowledge, reasoning and planning for intelligent characters" Proc. ACM SIGGRAPH 99 Conference, Los Angeles, CA, August, 1999, in Computer Graphics Proceedings, Annual Conference Series, 1999, 29-38.
[24] Pina A., Serón F.J. “Modelling Behavior and Motion Control in Computer Animation with Intelligent Objects”. International Symposium on Engineering of Intelligent Systems, EIS’98. University of la Laguna, Tenerife, Spain. February 11-13, 1998. ICSC.
[25] Gutiérrez, D., Pina, A., Serón, F.J., del Pico, J. “Character animation: agents, animats and synthetic actors”. Proceedings of the IASTED International Conference. Visualization, Imaging and Image Processing. Marbella, Spain 2001 pp. 145-150
[26] Geiger, C., Latzel, M. "Prototyping of complex plan based behavior for 3D actors”. AGENTS 00, pp. 451-458. Barcelona, Spain, 2000.
[27] Chi D., Costa M, Zhao L, Badler N, “The EMOTE Model for Effort and Shape”. Proceedings of the SIGGRAPH 2000. ACM Computer Graphics Annual Conference, New Orleans, Louisiana, pp. 173-182. July, 2000
[28] Cassell, J., Vilhjálmsson, H., Bickmore, T. "BEAT: the Behavior Expression Animation Toolkit" Proceedings of SIGGRAPH 2001. ACM Computer Graphics Annual Conference, Los Angeles, California, pp. 477-486. August, 2001.