An Extendable Multiagent Model for Behavioural Animation

Oliver Häger
German Aerospace Center (DLR)
Braunschweig site, Institute Automotive
oliver.haeger@dlr.de
http://www.dlr.de/fs/

Soraia Raupp Musse
CROMOS Laboratory
PIPCA - Universidade do Vale do Rio dos Sinos (UNISINOS)
São Leopoldo, Brazil
soraiarm@unisinos.br
http://www.inf.unisinos.br/~cromoslab/

Abstract

This paper presents a framework for visually simulating the behaviour of actors in virtual environments. In principle, the environmental interaction follows a cyclic processing of perception, decision, and action. As natural life-forms perceive their environment by active sensing, our approach also tends to let the artificial actor actively sense the virtual world. This allows us to place the characters in non-preprocessed virtual dynamic environments, what we call generic environments. A main aspect within our framework is the strict distinction between a behaviour pattern, that we term model, and its instances, named characters, which use the pattern. This allows them sharing one or more behaviour models. Low-level tasks like sensing or acting are took over by so called subagents, which are subordinated modules extendedly plugged in the character. In a demonstration we exemplarily show the application of our framework. Therefore we place the same character in different environments and let it climb and descend stairs, ramps and hills autonomously. Additionally the reactiveness for moving objects is tested. In future, this approach shall go into action for a simulation of an urban environment.

Keywords: behavioural animation, computer animation, multiagent system, framework

1 Introduction

Nowadays, in *Computer Graphics* it is possible to generate worlds with an astonishing degree of realism. Especially the entertainment industry takes advantage of those technologies and tries to capture the consumers with their more or less believable virtualities. But even the most detailed and most realistic virtual world remains lifeless without interacting inhabitants. The visual representation of such inhabitants like humans is already quite advanced, but still advancing within the technological progress. But only the visual representation is not enough to create credible entities as long as they do not act in a believable manner.

The discipline *Behavioural Animation* deals with visually simulating believable behaviour of such entities, which is still a challenge. Putatively simple actions such as grasping objects or climbing stairs in virtual worlds are not that simple to realize. Beside solving substantial problems like graphical representation, accurate collision detection, and simulating other physics, also the (autonomous) interaction process for the entity needs to be realized. Mallot[1] explicitly named such a process the *perception-action cycle*, where perceiving environmental information, extracting the needful data, deciding proper actions, and executing them, are the main aspects. A vital distinction is the manipulation of the environment's state through actions and their sequential (re-)actions. Following this principle we present an approach of a framework that can be used to simulate an entity's behaviour for such dynamic manipulable environments, which also shall be

- Adaptive for different non-preprocessed environments
- Reusable and easily extendable
- Usable for various types of virtual actors (humanoids, animals, vehicles etc.)
- Applicable for a larger amount of entities

With a small case study we also demonstrate the active senses and its reactive behaviour of humanoid entities in virtual environments. Those actors are placed in a non-preprocessed environment following predefined paths and autonomously react to some exemplarily encounters, like stairs, ramps, hills, and moving objects.

2 Related Work

Within the discipline *Behavioural Animation* a large community deals with the visual simulation of life-forms' behaviour. Since the first remarkable milestone *Boids* in 1987 [2] the continuous progress led to today's impressive results. Between Reynolds' *Boids* and the awarded thesis of Tu [3] about artificial fishes with physically realistic locomotion, perception, and believable behaviour in 1996 mainly fundamental research, for instance on perception, reasoning and animation, was done. Later on several works merged and refined those found results.

An interesting approach in that area is about non-verbal interpersonal communication among virtual actors by postures was shown by Bécheiraz and Thalmann [4]. Blumberg presented an autonomous acting dog with cognitive learning capabilities [5]. Noser and Thalmann demonstrated perception based behaviour in a tennis game simulation [6]. A way of planned navigation in dynamic environment using a vision sensor was shown by Kuffner [7]. He extended his approach in a recently presented work focussing additionally on flawless motion of characters [8]. While some researchers are interested in imitating individual believable behaviour with realistic physics-based models as it was already impressively demonstrated [9, 10], others are more focused on efficient solutions to be applicable for simulation of larger amounts of individuals [8, 11, 12, 13, 14].

Amongst them we found only a few, who underlined the usage of specifically developed frameworks for their purposes [9, 11]. Beside those other frameworks were presented in the past. The library *AGENTLib* offers the integration of differently generated motion types, like keyframing animation or inverse kinematics, of virtual humans [15]. It was extended by a flexible perception module, which allows information filtering of sensor-based input signals in virtual environments [16]. Millar proposed in his hypothesis a common way to model Behavioural Animation systems [17], which also was examined and generally acknowledged [18]. Another framework designed for agent-object interaction based on smart objects was developed by Peters et al. [19]. Furthermore Monzani [20] worked out a framework for “creating autonomous agents, capable of taking decisions based on their simulated perceptions and internal motivations”.[1]

3 Model

Here we present our approach of a flexible and efficient framework for autonomous characters in virtual environments. Its clearly structure allows a high extendability of the character’s behaviour. To show this capability we used our framework for a simple demonstration. A virtual human is placed within a *non-preprocessed dynamic environment*, which we use to call generic environment. When closing stairs, hills or ramps the character autonomously starts to climb or descend these objects using a believable animation. Also the reactive behaviour on dynamic objects will be demonstrated.

As others [1, 7, 12], already emphasized the basic principle for spatial cognition, our system bases on cyclic repetition of perception, decision and action illustrated in figure 1.

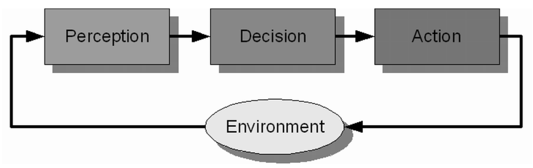

Figure 1: The basic principle of our framework bases on a *Perception-Decision-Action*-Loop.

For sensing the environment a virtual character shall be equipped with a set of active sensors. The sensors extract environmental information which is going to be processed to determine a proper action to be executed. These actions possibly can affect the environmental state. Due to the character’s sensorial activity these changes can be noticed.

[1] See in [20], page 3, second paragraph.

The aim was to offer a flexible, extendable, and efficient framework for applying behavioural animation in generic virtual environments. It shall allow to plug-in various modules for different low-level tasks. For that we first organized the framework in three operational layers, which are connected to each other by easy-to-use interfaces:

- Application
- Model
- Character

The *application-layer* contains the whole application and is – from a programmer’s point of view – inside the global scope. Herein we have the global objects like the once instanced message handler or the data structure of the world. The distinction between a *model* and its instance, which we call *character*, is vital for the concept of the framework. A character uses a set of predefined data of animation and behaviour, offered by the model. This feature allows a centralized handling of reusable data. So we can clump those data as we want to use this framework also for group and crowd simulation. Because each instance’s current state is hold by the character itself, but also necessary for reasoning, we need a communication structure between the separated layers. Hence we use the global message handler, who imparts message objects between related modules. The character not only maintains its own current state, but also has a set of subordinated modules, that we use to call *sub-agents*. An interface connects the extensions with their instance. These sub-agents are responsible for character’s low-level tasks like perceiving the environment or executing an animation.

The schematic overview of the described architecture is presented in figure 2. The stippled boxed indicates the different layers. It is valid that the system-layer is the superset of the model-layer, which on his part is the superset of the character-layer. This architecture allows the data exchange not only between characters and

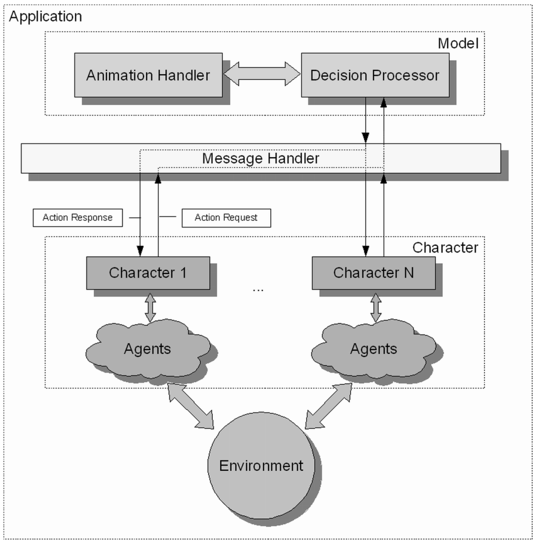

Figure 2: Our framework in a schematic overview. The connection between the model and its characters is done via the global message handler. The agents are the character's subordinated modules for low-level tasks, which – if necessary – perceive the information from the environment.

its model, but also between any character and any model.

For our work we revert to the free graphics toolkit *OpenSceneGraph*[2] to keep the virtual environment and also the animated virtual humans in an organized structure. The scene graph simplifies the access to the virtual world's object dramatically. Actually the animation data are processed by the *Character Animation Library 3D* (*Cal3D*).[3]

As mentioned before the main aspects are perception, decision, and action. Perception, like others, is a low-level task, which we handle by a specific interface connected to the character. The interface allows the exchange of data amongst the sub-agents for those low-level tasks. Figure 5 demonstrates the realized character concept. The subordinated modules in this illustration are part of our exemplary demonstration. Through the interface named *Extension-*

[2]`http://www.openscenegraph.org/` (2006-03-14)

[3]`http://cal3d.sourceforge.net/` (2006-03-14)

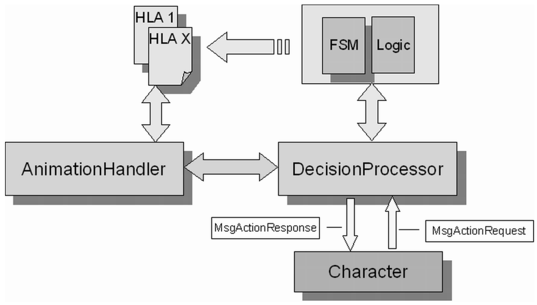

Figure 3: The decision processor for reasoning bases on a finite state machine. It is supported by an animation handler who keeps the animation data for different behaviours.

Handler it is easy to integrate other modules, which may process any low-level tasks.

It shall be noticed that we simulate a visual perception by using a geometric field of view. Potential visible objects – identified by their nodes' names – within this volume are checked for their distance to the character and eventually trigger a perception event. Furthermore we use another module that can be interpreted as a tactile sensor. Contact bones of the animation skeleton inform the character about ground contact and keep it in the right height relative to the ground height level. For the feedback between a character and an attached sub-agent we follow the principle, which we also use for the communication between the decision processor (as part of the model) and the characters:

A requesting event from a sub-agent makes the character ask the decision processor for a behaviour change.

That means, that we forward an initialized event to the decision processor, who then tries to resolve the problem. For that, the character passes its actual state with the data from the requesting sub-agent to the decision unit.

Actually the decision unit uses a finite state machine (FSM) to decide which behaviour shall be executed. The input data from the sub-agent (here: the visual perception unit) of the requesting character lead to a specific action. Combined with the character's actual state the FSM returns eventually a new state. Dependent on this state a predefined behaviour as a so called high-level animation (HLA) is going

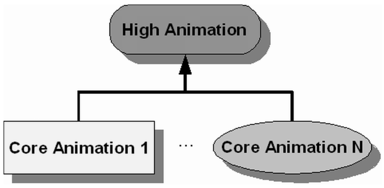

Figure 4: Several core animations can be blended in a more complex high-level animation.

to be returned to the character. For this a separated handler for animation data supports the decision processor. All necessary information about *Cal3D*'s core animations and the HLAs are maintained inside here. Both units together represent a *model*, which is depicted in figure 3. For the exchange of data between the characters and the models a global message handler is used, as we has described it already in the previous subsection. This method guarantees a true encapsulation between models and characters.

Basically, each character needs a module that is responsible for the action, that has to be executed after a decision for a behaviour was made. A behaviour means the execution of a suitable animation for the current situation. The subagent *Animator* takes up this task. The *Animator* receives a package of parameters from the decision processor, which we call high-level animation (HLA), and extracts its data to let *Cal3D* blend into a new animation. The animation to be executed is a compound of differently graded core animations based on keyframing animation, like figure 4 illustrates. In our example we let the decision process calculate the grades for each core animation. These grades depend on the received input data from the perceived objects, like the stair height of the next step to be climbed. It shall be noticed that the visual results highly depend on the quality of the used (core) animations. Because we use keyframing animation we cannot control the limbs directly, but only know – by extrapolation – where they could be in the next frame(s).

To build up a model and to produce its instances we developed a kind of a factory. Once a factory is constructed the model has also been built. Due to the model-character principle an instance can be inserted in and deleted from the scene at runtime. Each factory creates one model. Using for example the factory pattern of Gamma et al. [21], it is easily possible to control different model's and instances' creation with a simple interface.

We point out that this framework can be used for various forms of interaction between virtual environments and its inhabitants. Reducing its architecture to the here presented elements offers a large functionality. Based on already quite efficient toolkits for animation and visualization the user just has to care about his own implementation.

4 Applying the framework: A case study

For demonstration purposes we chose to simulate the behaviour for climbing and descending continuous and discrete height level variances. In general, representative objects within a virtual environment are stairs, hills and ramps. As this is still a classical, but not unresolved problem in robotics, we could not find much about this subject for virtual characters. For sure, successful approaches on physically realistic animations, for example [9], offers the capability for climbing stairs. In some.approaches [19, 22, 23, 24] cover this subject by using so called smart objects, for which a scene needs to be preprocessed. Primarily, our aim is that our framework can also be used for any kind of (nearly) non-preprocessed virtual environments. So the usage of smart objects is not suitable for our approach. Using physically realistic animation models also may not be suitable as we plan to apply the framework for a larger amount of virtual characters. The high computational costs of such models denies the usage for our future plans.

To notice environmental changes it is necessary to let the characters know about possible events. For reasons of true autonomy, but also for the use in generic environments, we equipped the virtual human with active sensors, which simulate vision and touch. The active sensors are subordinated modules of a model's instance. These sensors collect the necessary environmen-

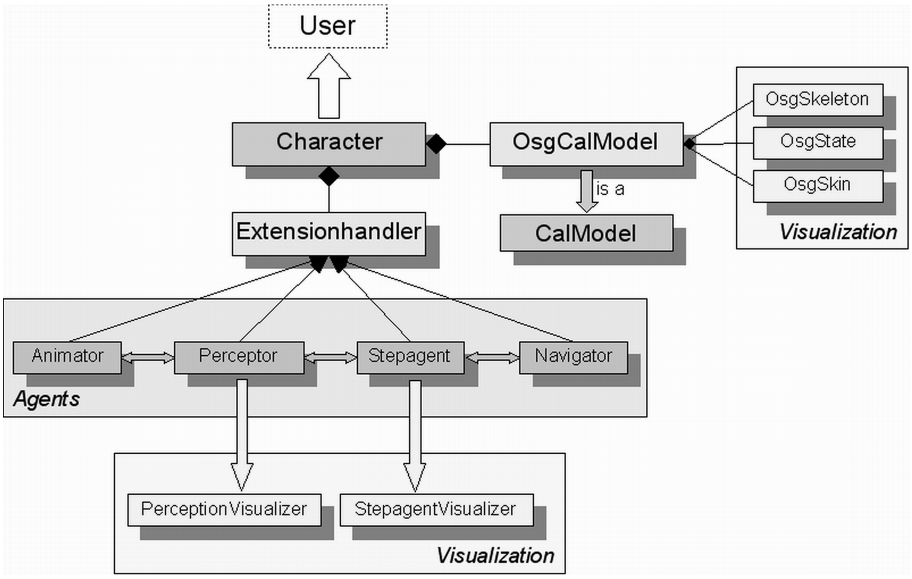

Figure 5: The applied framework for an exemplary demonstration. The four sub-agents connected to the extension handler take over the different low-level tasks.

tal data of potential visible and touchable objects, like stairs, ramps and moving scene participants. The received information is going to be sent to the separated decision unit. For reasoning we revert to a finite state machine that decides whether a new action shall be executed. A character communicates with the decoupled decision processor via message objects that keep the data needed for further processing. If it is necessary to change the behaviour the decision unit responds the character's request and returns the animation data needed for execution. Each character can be treated as an autonomous unit, that explores its environment by itself. The demonstration just shows reactive behaviour of a character, but the extendibility offered by our framework allows a rapid integration of further modules, that can take over more advanced functions like memorizing, learning and goal-oriented planning. The figure 5 illustrates the scheme of the exemplarily applied framework. The four integrated sub-agents are separately connected to the character. Each is responsible for its own low-level task. The *Animator* receives the animation data, and via the linkage to *Cal3D* it blends into a new animation. The *Perceptor* simulates the synthetic vision of the character, while the module *Stepagent* is responsible for the calculations regarding the vertical and horizontal translation. The *Navigator* lets the character follow a predefined path of navigation points. All these modules are bound to the character via the interface named *Extensionhandler*, that facilitates the access to each of these modules. As it is shown, the visual representation of the character is completely decoupled of the sub-agents. Calculated data of sub-agents, which need to be shown during the developing process can optionally be activated and therefore remain also independent of the character's visualization.

We placed the same character in different environments and let it act within these scenes. In our demonstration a character for example arrives at a stair, checks the first step's height and – if within a passable height – changes to a proper animation to climb or descend the stair. If reached at the end of the stairs it returns to his walking animation. It is also capable to recognize ramps and hills and changes his behaviour in a proper manner. When reaching close to unknown or non-passable objects the character smoothly stops in front of the obstacle. In case of moving hazards, it continues his way after the object has passed. The figure 6 shows some excerpts from our example. Among others it is illustrated that the visual perception bases on a collision detection method with a geometric volume, which represents the field of view. It can be

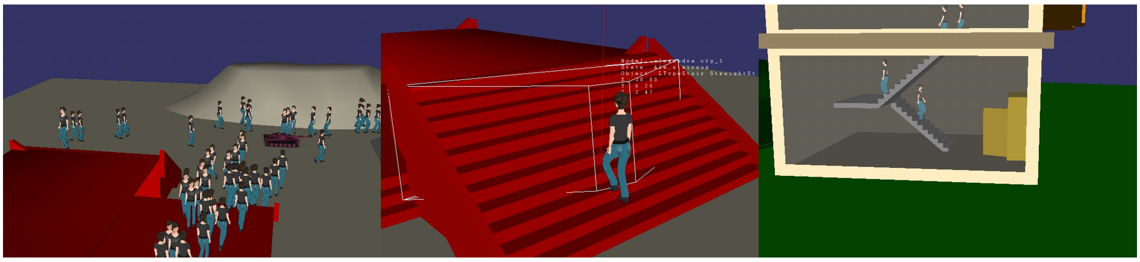

Figure 6: Excerpts from the case study.

seen, that our framework also works with many instances of a model. In our demonstration we use a humanoid model with nearly 3500 polygons in a simple environment. We can simultaneously draw about 60 instances in real-time (> 24 frames per second). It shall be noticed that we did not really optimized the drawing and calculation routines yet. By lowering the level of detail of the models, we surely can draw more than 200 characters even in real-time with the actual framework. Theoretically there's no limit for instances in a non-real-time application, as we already checked with about 600 characters.[4]

5 Conclusions and Future Work

We presented a framework for behavioural animation based on a *Perception-Decision-Action-*Loop. A main aspect is the decoupling between a model and its instances, the so called characters. The model maintains all reusable data, as there are the animation data and the decision mechanism, which together can be seen as a behavioural pattern. These parts are connected with a global messaging system. This feature will allow not only the data exchange between a character and its model, but also the exchange amongst characters of different models. Furthermore the character's capabilities easily can be extended by using an easy-to-use plug-in interface. Those extensions are subordinated, but still encapsulated, modules, which we call sub-agents and which are responsible for low-level tasks, like sensing or executing animations. Hence it is easy to integrate a once developed sub-agent, that for example take over the task for hearing, smelling, or even simulating the character's behaviour for waiting lines at service points.

As the aim was to create a reusable system for virtual life-forms in non-pre-processed virtual environments active sensing is a vital component. In a case study we presented the abilities of an active visual and tactile sensor on letting a character climb and descend stairs, ramps and hills. Even moving objects – embedded in the used scene graph – are recognizable by the humanoids and cause a proper behaviour.

We want to apply this framework for different aspects in a simulated urban environment. The extension of the model for more complex behaviour, like using emotional and intentional states, shall also be part of future works. Using the messaging system an "interpersonal" communication can easily be assimilated. Additionally, the integration of advanced mechanisms for navigation, like path-planning and autonomous collision avoidance are planned. Furthermore a character will be equipped with more active senses like smell and hearing. The flexibility of the framework also allows us to use it for the simulation of vehicles in the urban environment. Although the actual state of our framework needs some improvements regarding optimization and still better usability, we are confident that our desired prospective aims can be realized with the presented approach.

Acknowledgements

We would like to thank the German Academic Exchange Service for the granted scholarship.

[4]The demo runs on a commercially usual notebook with an Intel® Pentium® M 730 CPU at 1.6 GHz, 512 MB RAM and an ATI® MOBILITY™RADEON™X600 display controller with 64 MB VRAM.

References

[1] J. M. Loomis H. A. Mallot, S. D. Steck. Mechanisms of spatial cognition:behavioral experiments in virtual environments. *Künstliche Intelligenz*, (4):24–28, 2002.

[2] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 25–34, New York, NY, USA, 1987. ACM Press.

[3] X. Tu. *Artificial Animals for Computer Animation*. Springer, Berlin, Germany, 1999.

[4] D. Thalmann P. Bécheiraz. A model of nonverbal communication and interpersonal relationship between actors. In *Proceedings of Computer Animation'96*, pages 58–67. IEEE Computer Society Press, June 1996.

[5] B. Blumberg. *Old Tricks, New Dogs: Ethology and Interactive Creatures*. PhD thesis, Massachusetts Institute of Technology, 1997.

[6] H. Noser D. Thalmann. Sensor based synthetic actors in a tennis game simulation. *The Visual Computer*, 14(4):193–205, 1998.

[7] J. J. Kuffner. *Autonomous Agents For Real-Time Animation*. PhD thesis, Stanford University, Stanford, CA, USA, December 1999.

[8] M. Lau J.J. Kuffner. Behavior planning for character animation. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '05)*. ACM Press, 2005.

[9] D. Terzopoulos P. Faloutsos, M. van de Panne. Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH'01)*, pages 251–260. ACM Press, 2001.

[10] D.C. Brogan J.F. O'Brien J.K. Hodgins, W.L. Wooten. Animating human athletics. In *Proceedings of the 22nd International Conference on Computer Graphics and Interactive Techniques*, pages 71–78. ACM Press, 1995.

[11] D. Terzopolous W. Shao. Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 19–28, New York, NY, USA, 2005. ACM Press.

[12] S. Donikian. Htps: A behaviour modelling language for autonomous agents. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 401–408, New York, NY, USA, 2001. ACM Press.

[13] S. R. Musse. *Human Crowd Modeling with Various Level of Behavior Control*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2000.

[14] D. Thalmann S. R. Musse. Hierarchical model for real time simulation of virtual human crowds. In *IEEE Transactions on Visualization and Computer Graphics*, volume 7, pages 152–164, 2001.

[15] Luc Emering Daniel Thalmann Ronan Boulic, Pascal Bécheiraz. Integration of motion control techniques for virtual human and avatar real-time animation. In *VRST '97: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 111–118, New York, NY, USA, 1997. ACM Press.

[16] D. Thalmann C. Bordeux, R. Boulic. An efficient and flexible perception pipeline for autonomous agents. *Proceedings of EUROGRAPHICS'99*, 18(3):23–30, 1999.

[17] S.M. Kealy R.J. Millar, J.R.P. Hanna. A review of behavioural animation. In *Computers and Graphics*, volume 23, pages 127–143, 1999.

[18] W.M. Johnston J. R. P. Hanna, R. J. Millar. Examing the generality of a behavioural animation framework. In *WSCG'2001:*

9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2001, volume 9, 2001.

[19] B. MacNamee C. O'Sullivan C. Peters, S. Dobbyn. Smart objects for attentive agents. In *Short Paper Proceedings of the Winter Conference on Computer Graphics 2003*. UNION Agency Science Press, February 2003.

[20] J.-S. Monzani. *An architecture for the Behavioural Animation of Virtual Humans*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, July 2002.

[21] R. Johnson J. Vlissides E. Gamma, R. Helm. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, July 1997.

[22] R. Boulic D. Thalmann, N. Farenc. Virtual human life simulation and database: Why and how. In *International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, pages 471–480. IEEECS Press, 1999.

[23] D. Thalmann N. Farenc, R. Boulic. An informed environment dedicated to the simulation of virtual humans in urban context. In *Proceedings of EUROGRAPHICS'99*, volume 18, pages 309–318, 1999.

[24] D. Thalmann M. Kallmann. Modeling objects for interaction tasks. In *Proceeedings of EGCAS'98, Eurographics Workshop on Animation and Simulation*, pages 73–76, 1998.