

# Plataforma de Suporte à Geração de Cenas Animadas com Agentes Inteligentes

M. Beatriz Carmo, A. Paula Cláudio, João D. Cunha, Helder Coelho, Miguel Silvestre

FCUL-DI, Campo Grande 1749-016 Lisboa

{bc,apc,jdc,hcoelho}@di.fc.ul.pt,  
msilvestre@gmail.com

Maria Pinto-Albuquerque

DCTI, Instituto Sup. de Ciências do Trabalho e da Empresa

Av. das Forças Armadas Edifício ISCTE 1649-026 Lisboa

maria.albuquerque@iscte.pt

---

## Resumo

*Neste artigo apresenta-se uma plataforma de suporte à geração de cenas animadas com agentes inteligentes. É proposta uma arquitectura que separa o processamento gráfico do processamento inteligente. A plataforma que a concretiza incorpora várias aplicações que são articuladas por dois módulos independentes que comunicam entre si: a camada de processamento gráfico e a camada de processamento inteligente.*

**Palavras Chave:** *humanos virtuais, ambientes virtuais, animação comportamental.*

---

## 1. INTRODUÇÃO

A construção de ambientes virtuais onde se movimentam humanos virtuais inteligentes é cada vez mais comum. Estes ambientes têm aplicação em áreas tais como entretenimento, educação, treino em ambientes industriais, reconstituição de ambientes históricos, aplicações médicas, tratamento de fobias.

A representação de humanos virtuais, quer em ambientes virtuais, quer em ambientes reais, como acontece na Realidade Aumentada, é um tema de investigação onde se combinam contribuições de várias disciplinas, nomeadamente, da Computação Gráfica, da Inteligência Artificial e das Ciências Sociais. O trabalho descrito neste documento enquadra-se no projecto MAGO2, Modelação de Agentes em Organizações. Este projecto, desenvolvido no LabMAg, Laboratório de Modelação de Agentes [<http://LabMAg>], integra várias tarefas relacionadas com a estrutura e dinâmica de sociedades de agentes. A tarefa do MAGO2 que corresponde ao trabalho descrito propõe-se concretizar uma biblioteca de representações gráficas para agentes inteligentes, que possa vir a suportar estudos sobre o comportamento de agentes individualmente ou integrados em sociedade.

Têm sido apresentadas diversas concretizações de plataformas para a geração de cenas animadas com agentes inteligentes. Na secção 2 são referidas algumas destas soluções. Na secção 3 apresenta-se o trabalho desenvolvido e na secção 4 conclui-se descrevendo o trabalho em curso e o trabalho futuro.

## 2. TRABALHO RELACIONADO

A geração de ambientes virtuais com agentes virtuais autónomos implica a integração de ferramentas gráficas com módulos de processamento inteligente. Referem-se

em seguida trabalhos que explicitamente se debruçam sobre a integração destas duas componentes.

O trabalho de Tu et al. [Tu94] sobre a simulação artificial de peixes é um caso particular da animação de agentes inteligentes. Neste caso a modelação de cada agente incorpora três componentes: o sistema motor, o sistema de percepção e o sistema de comportamento. O sistema motor baseia-se em modelos físicos para gerar o movimento dos peixes. O sistema de percepção capta a informação sobre o meio envolvente através de um sensor de visão e de um sensor da temperatura da água. O sistema de comportamento faz a mediação entre os sistemas sensorial e motor. A modelação do comportamento tem em atenção a definição dos hábitos do peixe, do seu estado mental e a informação obtida através dos sensores. Os hábitos do peixe são definidos interactivamente pelo utilizador através do ajustamento de um conjunto de parâmetros. Funge et al. [Funge99] estendem este trabalho, incluindo no modelo de comportamento, um modelo cognitivo que permite aos agentes virtuais tomarem decisões baseando-se em conhecimento adquirido sobre o mundo. O modelo de Funge utiliza o formalismo de cálculo de situações, utilizado em Inteligência Artificial, para a descrição de alterações no mundo envolvente.

O sistema desenvolvido por Perlin et al. [Perlin96], Improv, permite a criação de actores animados com comportamentos próprios e que interagem entre si e com o utilizador. Este sistema integra duas componentes fundamentais. A primeira, *animation engine*, permite a criação de animações, incluindo ferramentas para gerar movimentos realistas. Além do recurso a técnicas de *inverse kinematics* e de deformação de superfícies, usa funções de ruído para combinar diferentes movimentos

numa só acção [Perlin95]. Os actores, quer sejam humanos, animais ou seres imaginários, são constituídos por segmentos ligados por articulações. A segunda componente, *behavior engine*, controla as regras de comportamento dos actores. Os diferentes tipos de comportamento dos actores são fornecidos numa linguagem de *script* própria. A selecção do comportamento de um actor baseia-se em regras de decisão e no peso atribuído pelo autor a cada comportamento. O sistema Improv recorre à biblioteca gráfica SGI GL e é implementado como um conjunto de programas distribuídos num ambiente Unix comunicando entre si através de *sockets*, protocolos *multicast* e *pipes*.

Devillers et al. [Devillers02] construíram uma plataforma para a animação de agentes inteligentes que designaram por GASP (*General Animation and Simulation Platform*). Esta plataforma inclui vários módulos que contemplam a modelação de diferentes aspectos dos agentes, tais como, controlo do movimento, visão, comportamento. A execução pode ser distribuída por vários processadores recorrendo ao PVM (*Parallel Virtual Machine*) para realizar a comunicação entre eles [Donikian98].

Outra proposta de arquitectura para a animação de humanos virtuais com raciocínio cognitivo é apresentada em [Torres02]. Esta arquitectura inclui uma componente gráfica, uma componente de raciocínio e uma interface entre estas duas componentes [Torres03]. A representação gráfica dos humanos é definida por um modelo hierárquico que representa o conjunto de articulações do corpo humano e um conjunto de funções para gerar movimentos nessas articulações. São usados ficheiros XML para definir a representação e a movimentação dos agentes. A componente de raciocínio é implementada através da linguagem AgentSpeak(XL) [Bordini02]. Esta linguagem é uma extensão de uma linguagem de programação de agentes AgentSpeak(L) proposta em [Rao96] para implementar agentes de acordo com o modelo BDI (*Beliefs, Desires, Intentions*). A interface faz a comunicação entre o “corpo articulado” e a mente. A interface recebe do ambiente gráfico informação sobre o estado do ambiente e comunica-a à componente de raciocínio. Esta, por sua vez, decide qual o comportamento adequado para o agente e fornece à interface esta informação. A componente de raciocínio é independente da componente gráfica e comunica com esta através de *sockets*.

Daniel Thalmann et al. [Thalmann04] propõem também uma arquitectura baseada na separação entre a simulação física e a simulação comportamental. Neste caso são identificadas duas componentes, uma para cada um dos dois tipos de simulação: o *Agents' Common Environment* (ACE) e o *Intelligent Virtual Agent* (IVA). O ACE contém um conjunto de comandos para controlar a simulação do mundo físico como, por exemplo, comandos para criar e animar objectos 3D e humanos virtuais capazes de exprimir emoções [Kallmann00].

Engloba ainda animações básicas, como por exemplo vários tipos de locomoção de humanos virtuais. As linguagens usadas para este nível são o C++ e a linguagem de *scripting* Python. Esta última é usada para a criação de comandos de animação dos agentes a incluir no cenário. Cada agente corresponde a uma *thread* de modo a poder interagir de modo independente com o ambiente 3D que o rodeia. Os agentes comunicam entre si e sincronizam-se com os restantes elementos do cenário, usando uma área comum de memória que é gerida por uma entidade designada por *Agents Controller*. Esta área comum é especialmente útil quando os agentes estabelecem entre si comunicação verbal. A animação em tempo-real dos humanos virtuais está a cargo da aplicação AgentLib [Boulic97], integrada no ACE. A animação produzida explora várias técnicas: sequências de *keyframes*, *inverse kinematics* e captura de movimentos (*motion capture*). O IVA baseia-se na arquitectura BDI [Caicedo00]. Esta componente foi desenvolvida em Java [Monzani01]. Cada entidade da simulação que seja dotada de inteligência corresponde a uma *thread* no IVA. Cada uma destas *threads* comunica, através de uma ligação TCP/IP, com a *thread* que lhe corresponde no ACE. Contudo, tal como os humanos não comunicam entre si usando ligações directas entre os seus cérebros, as *threads* do IVA não estabelecem qualquer tipo de comunicação entre si.

Vosinakis et al. [Vosinakis05] desenvolveram uma ferramenta, *SimHuman*, para a construção de ambientes virtuais com agentes autónomos. O *SimHuman* é composto por uma biblioteca escrita em C++ e recorre à biblioteca gráfica OpenGL. Uma aplicação pode incorporar chamadas a esta biblioteca para fazer o *rendering* e animação de um ambiente virtual interactivo 3D com um número arbitrário de objectos e humanos virtuais, quer com comportamentos próprios, quer controlados pelo utilizador. O *SimHuman* inclui ainda dois utilitários: um que ajuda na criação e animação de agentes (*Agent Designer*) e outro que facilita a colocação de objectos numa cena e a visualização de sequências de acções dos agentes (*Scene Designer*). O movimento e acção dos agentes tem subjacente a utilização de modelos baseados na Física (*Physically Based Modeling*), *Inverse Kinematics*, detecção de colisões e visão dos agentes. O modelo geométrico dos agentes pode ser definido utilizando uma aplicação comercial, *Curiouslabs Poser*. Por outro lado, a geometria dos objectos que compõem a cena pode ser definida em qualquer *software* de modelação que exporte ficheiros no formato VRML97.

Nos trabalhos referidos é nítida a separação entre a componente gráfica e a componente de inteligência artificial. Esta foi também a abordagem adoptada na arquitectura que propomos para a construção de ambientes virtuais habitados por seres dotados de comportamento inteligente.

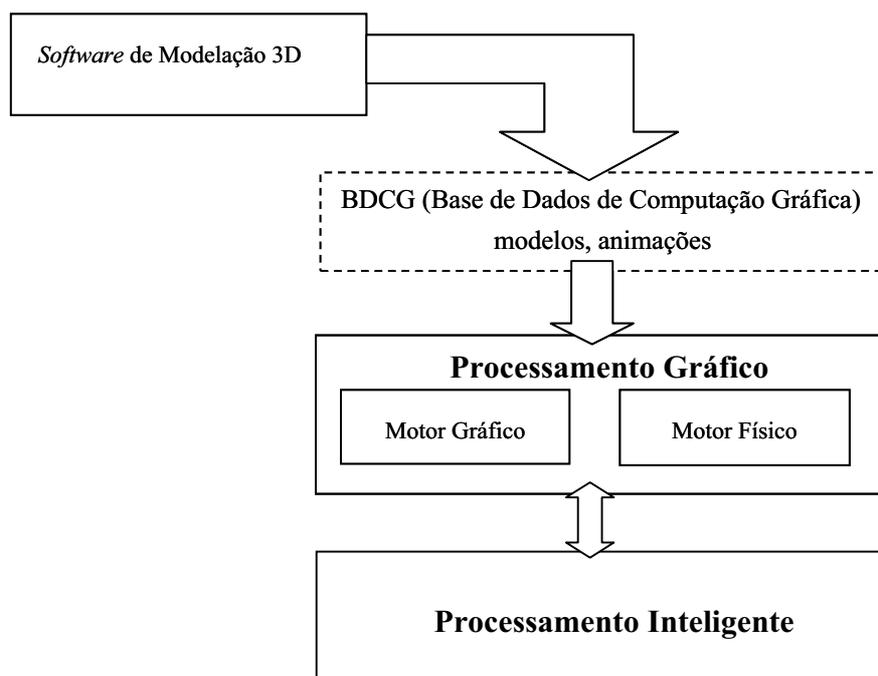


Fig. 1 – Arquitectura de suporte à geração de cenas animadas com agentes inteligentes

### 3. ARQUITECTURA ADOPTADA E PLATAFORMA DESENVOLVIDA

A arquitectura que propomos para suportar a geração de cenas animadas com agentes inteligentes tem duas componentes fundamentais: uma componente gráfica e uma componente de inteligência artificial. O parágrafo seguinte explica em traços gerais o que levou à concepção da arquitectura proposta e que a figura 1 ilustra.

Relativamente à componente gráfica, um dos elementos fundamentais é um *software* de modelação 3D para criar os elementos da cena e gerar as animações pretendidas. Com o objectivo de escolher uma abordagem que permita a geração de imagens em tempo-real, optou-se por juntar a geração de imagens em tempo-real, optou-se por juntar um motor gráfico. Como o motor gráfico nem sempre resolve alguns aspectos da simulação do comportamento físico dos objectos da cena como, por exemplo, as colisões ou a aplicação de forças, considerou-se necessária a inclusão de um motor físico.

Uma vez que o motor gráfico importa o modelo criado pelo *software* de modelação através da leitura de um ficheiro, esta arquitectura é independente do *software* de modelação, o que consideramos ser uma vantagem considerável. Além do programa de modelação, do motor gráfico e do motor físico, esta arquitectura fica aberta à inclusão de outras ferramentas.

Quanto à componente de processamento inteligente optou-se pela ligação a uma bancada de agentes. Depois de definida a arquitectura houve que tomar decisões sobre o *software* a utilizar nas duas componentes. O processo de desenvolvimento de ferramentas gráficas e de inteligência artificial é normalmente moroso e exigente do ponto de vista de recursos humanos. Para reduzir o tempo de criação de uma plataforma de trabalho optou-se por incorporar aplicações já existentes e de

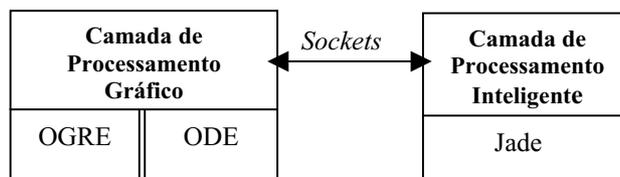
domínio público. Nas subsecções seguintes justificamos as escolhas efectuadas e descrevemos os detalhes mais relevantes da concretização da plataforma.

#### 3.1 Componente Gráfica

Como ponto de partida e com o objectivo de escolher de forma ponderada as ferramentas a adoptar, iniciou-se uma pesquisa sobre as ferramentas de modelação existentes. Compararam-se as diferentes ferramentas com base no seguinte conjunto de características: capacidades de modelação e de animação disponibilizadas, facilidades de exportação e de importação de ficheiros, quantidade de ajuda acessível através da internet e, por último, contextos efectivos de utilização da ferramenta [Silvestre04].

O Blender [httpBlender] foi o *software* de modelação 3D escolhido. Trata-se de um *software* gratuito, com uma grande comunidade de utilizadores e que tem estado em contínuo desenvolvimento. Apesar de se ter considerado a interface menos amigável do que a de outras ferramentas, concluiu-se que possui fortes capacidades de modelação gráfica e permite importar ficheiros de vários tipos.

Relativamente ao motor gráfico foi adoptado o OGRE [httpOgre]. Trata-se de um motor genérico, isto é, não foi desenvolvido para um tipo particular de aplicação. O OGRE é um motor bastante flexível, modular, eficiente, que conta com uma grande comunidade de desenvolvimento e de utilizadores. As imagens geradas pelo OGRE são de qualidade superior às de outros motores gráficos analisados. Uma vez que o OGRE não inclui um sistema de simulação de fenómenos físicos ou de colisões, incluiu-se no processamento gráfico um motor físico que permite simular estes fenómenos.



**Fig. 2 Ligação entre a componente gráfica e a componente inteligente**

O ODE [httpODE] foi o motor físico escolhido uma vez que é muito usado, bastante referenciado, com uma grande comunidade de utilizadores e com bons materiais de apoio, incluindo vários exemplos sobre como usar o ODE em conjunto com o OGRE.

Para construir uma aplicação usando estas componentes criam-se primeiro os modelos gráficos dos agentes e as suas animações básicas, como andar, correr, entre outras, utilizando o *software* de modelação 3D. Em seguida, estes modelos são exportados para um formato conhecido do motor gráfico, criando-se assim uma Base de Dados de modelos animados ou inanimados, como é o caso dos elementos estáticos, por exemplo, cadeiras, mesas, objectos de arte.

Para garantir a ligação entre as aplicações OGRE e ODE desenvolveu-se um conjunto de classes C++ designadas por camada de processamento gráfico [Silvestre05a]. Esta camada contém igualmente classes que suportam a comunicação com a componente de inteligência artificial.

### 3.2 Componente de Inteligência Artificial

A fim de simular os comportamentos inteligentes dos agentes adoptámos uma bancada de agentes já anteriormente utilizada no contexto de outros trabalhos desenvolvidos no LabMAg: JADE (*Java Agent Development Framework*).

O JADE é uma plataforma de agentes que facilita a tarefa de criação de novos agentes, bem como toda a comunicação feita entre agentes. É uma plataforma distribuída, sendo transparente para o utilizador a máquina onde a *thread* de cada agente é executada. Deste modo podemos ter vários agentes a serem executados em máquinas distintas.

Desenvolveu-se uma camada de *software* sobre o JADE, designada por camada de processamento inteligente, de forma a estabelecer a ligação entre o JADE e a camada de processamento gráfico.

O JADE e a camada de processamento inteligente estão implementados em JAVA, enquanto o OGRE, o ODE e a camada de processamento gráfico estão implementados em C++. A solução adoptada para fazer a ligação entre as duas camadas passou pelo recurso a *sockets*. A utilização de *sockets* permite distribuir por diferentes computadores o processamento relativo a cada uma destas camadas.

Apesar de ser possível ligar cada agente JADE directamente à camada de processamento gráfico, havendo assim tantas ligações quantos os agentes existentes, optámos por criar apenas uma conexão simples entre a camada de processamento gráfico e a

camada de processamento inteligente, usando apenas uma ligação TCP/IP (Fig.2). A ideia principal é ter um agente na bancada JADE responsável pela comunicação e coordenação entre os agentes JADE e a camada de processamento gráfico.

A adopção de *sockets* para fazer a ligação entre as camadas de processamento gráfico e de processamento inteligente tornou necessário criar um protocolo de comunicação que fosse entendido por ambas as camadas. As duas camadas comunicam através de uma sequência de pedidos e respostas que respeitam o protocolo de comunicação definido. A camada de processamento gráfico efectua um pedido à camada de processamento inteligente, esta recebe o pedido, executa-o e envia a resposta à camada de processamento gráfico.

### 4. CONCLUSÕES E TRABALHO FUTURO

Têm sido propostas diferentes soluções para a criação de plataformas para a geração de agentes autónomos em ambientes virtuais, como referimos atrás. Nas soluções referidas grande parte da componente gráfica é construída de raiz e, na maior parte dos casos, a plataforma desenvolvida não está disponível para outros utilizadores.

O desenvolvimento de uma solução própria com base em bibliotecas gráficas elementares, apesar de ser muito mais flexível, é muito mais demorado e exigente em termos de recursos humanos. Optou-se assim por criar uma plataforma modular baseada na integração de aplicações já com algum grau de especialização. A escolha das aplicações teve em atenção o custo, a documentação acessível e a existência de grupos de utilizadores activos.

A arquitectura está aberta à adopção de um *software* de modelação distinto do escolhido o que permitirá utilizar outras ferramentas de modelação. Tendo em conta que algumas ferramentas comerciais oferecem um período de utilização gratuita ou uma versão incompleta do produto, esta hipótese é perfeitamente viável e não acarreta custos financeiros.

Até ao momento foi criado um núcleo restrito de operações que permitiu a execução de uma demonstração para exemplificar a viabilidade da arquitectura. Nesta demonstração a movimentação do agente é feita interactivamente pelo utilizador através do teclado ou do rato [Silvestre05b].

O interesse do trabalho descrito reside na sua potencial utilidade nos ambientes de investigação (LabMAg) e de docência (disciplinas dos cursos do departamento de Informática da FCUL) em que se insere. Destacamos a

sua utilização na exploração de conceitos no contexto da animação comportamental (*behavioral animation*). Como exemplo podemos referir a implementação do conceito de *smart object* [Thalmann04], um objecto que encapsula uma espécie de guia de utilização que deve ser seguido pelos agentes durante a interacção com o referido objecto.

Esta plataforma pode servir também de suporte experimental para projectos a realizar no contexto das disciplinas de Inteligência Artificial ministradas no departamento e, eventualmente, para trabalhos de investigação de maior envergadura.

Além do enriquecimento das camadas do processamento gráfico e de inteligência artificial, a incorporação de um sistema de visão é um ponto chave para a continuação do trabalho. Através de um sistema de visão, um agente pode reconhecer o ambiente e os agentes que o rodeiam permitindo a interacção entre os vários intervenientes da cena. Outro aspecto a considerar é a realização de testes sobre a adequação da plataforma para a concepção de uma biblioteca de expressões faciais capazes de transmitir emoções e estados de espírito dos agentes inteligentes.

## 5. REFERÊNCIAS

- [Bordini02] Bordini, R.H., Bazzan, A.L.C., Jannone, R.O., Basso, D.M., Vicari, R.M., and Lesser, V.R., AgentSpeak(XL): Efficient intention selection in BDI agents via decision-theoretic task scheduling, in Castelfranchi, C., and Johnson, W.L., eds., Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002), Bologna, Italy, pp 1294-1302, New York, NY: ACM Press, 2002
- [Boulic97] Boulic, R., Beicheraz, P., Emering, L., Thalmann, D., Integration of Motion Control Techniques for Virtual Human and Avatar Real-Time Animation, Proceedings of VRST'97, pp. 111-118, ACM Press, 1997
- [Caicedo00] Caicedo, A., Thalmann, D., Virtual Humanoids: Let Them be Autonomous without Losing Control, Proc. 3IA 2000, Limoges, France, 2000
- [Devillers02] Devillers, F., Donikian, S., Lamarche, F., Taille, J.-F., A programming environment for behavioral animation, The Journal of Visualization and Computer Animation, 13:263-274, 2002
- [Donikian98] Donikian, S., Chauffaut, A., Duval, T.Z., Kulpa, R., GASP: from Modular Programming to Distributed Execution, Computer Animation' 98, Philadelphia, USA, 1998
- [Funge99] Funge, J., Tu, X., Terzopoulos, D., Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters, SIGGRAPH 99, Los Angeles, CA, 1999
- [httpBlender] <http://www.blender3d.org/cms/Home.2.0.html>
- [httpJADE] <http://jade.tilab.com/>
- [httpLabMAg] <http://labmag.di.fc.ul.pt/>
- [httpODE] <http://ode.org>
- [httpOGRE] <http://www.ogre3d.org/>
- [Kallmann00] Kallmann, M., Monzani, J.-S., Caicedo, A., Thalmann, D., ACE: A Platform for the Real Time Simulation of Virtual Human Agents, EGAS'2000, 11th Eurographics Workshop on Animation and Simulation, Interlaken, Switzerland, 2000
- [Maciel02] Maciel, A., Nedel, L.P., Freitas, C.M., Anatomy-Based Joint Models for Virtual Humans Skeletons. In: Computer Animation 2002, 2002, Lausanne. Computer Animation, v.1 p.110 – 116, 2002
- [Monzani01] Monzani, J.-S., Caicedo, A., Thalmann, D., Integrating Behavioural Animation Techniques, Eurographics 2001, A. Chalmers and T.-M. Rhyne (eds), Computer Graphics Forum, vol 20, nº 3, 2001
- [Perlin95] Perlin, K., Real Time Responsive Animation with Personality, IEEE Transactions on Visualization and Computer Graphics, pp 5-15, vol. 1, nº1, 1995,
- [Perlin96] Perlin, K., Goldberg, A., Improv: A System for Scripting Interactive Actors in Virtual Worlds, Proceedings of SIGGRAPH'96, pp205-216, 1996
- [Rao96] Rao, A.S., AgentSpeak(L): BDI Agents speak out in a logical computable language, Proceedings MAAMAW-96, Springer Verlag, LNAI 1038, pp 42-55, 1996
- [Silvestre04] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-04-7, Julho 2004
- [Silvestre05a] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Concretização de uma Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-05-6, Março 2005
- [Silvestre05b] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Platform for the Generation of Virtual Environments Inhabited by Intelligent Virtual Humans, *student poster* ACM ITiCSE'05, Monte da Caparica, Junho 2005
- [Thalmann04] Handbook of Virtual Humans, edited by N. Magnenat-Thalmann, D. Thalmann, John Wiley & Sons, 2004
- [Torres02] Torres, J.A.R., Maciel, A., Nedel, L.P., Uma Arquitetura para Animação de Humanos Virtuais

com Raciocínio Cognitivo In: SVR 2002 – Symposium on Virtual Reality, 2002, Fortaleza - CE Proceedings, p.341-352. Fortaleza: SBC, 2002

[Torres03] Torres, J.A., Nedel, L.P., Bordini, H.R., Using the BDI Architecture to Produce Autonomous Characters in Virtual Worlds, Interactive Virtual Agents (IVA 2003), Irsee, Germany, 2003

[Tu94] Tu, X., Terzopoulos, D., Artificial fishes: Physics, locomotion, perception, behavior, Proc. ACM SIGGRAPH'94 Conference, Orlando, FL, 1994, in Computer Graphics Proceedings, Annual Conference Series, 43-50, 1994

[Vosinakis05] Vosinakis, S. and Panayiotopoulos, T., A Tool for Constructing 3D Environments with Virtual Agents, Multimedia Tools and Applications, 25, 253-279, 2005