

Uma Arquitetura para Animação de Humanos Virtuais com Raciocínio Cognitivo

Jorge Alberto Rangel Torres¹, Anderson Maciel², Luciana Porcher Nedel¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²Virtual Reality Lab – Swiss Federal Institute of Technology (EPFL)
CH 1015 – Lausanne – VD – Switzerland
{jorget,nedel}@inf.ufrgs.br, anderson.maciel@epfl.ch

Abstract. *This paper proposes an architecture to implement autonomous synthetic actors in virtual worlds. These actors are represented by an articulated structure and have cognitive reasoning using BDI logic. Such architecture uses an interpreter for AgentSpeak language developed by the Artificial Intelligence group at UFRGS, and an articulated body animation model. This work suggests the development of a communication interface between the two systems described above, using sockets.*

Resumo. *Este artigo propõe uma arquitetura para implementar atores sintéticos autônomos em ambientes virtuais que combinam raciocínio cognitivo usando lógica BDI e representação e movimentação do corpo usando uma estrutura articulada. A arquitetura apresentada utiliza um interpretador para a linguagem AgentSpeak, desenvolvido pelo grupo de Inteligência Artificial da UFRGS, e um modelo de corpo articulado com animação e propõe uma interface de comunicação entre os dois sistemas utilizando sockets.*

1. Introdução

Humanos virtuais são modelos computacionais de pessoas, usados tanto como substitutos para pessoas reais em avaliações ergonômicas de projetos baseados em computador, quanto para embutir, em tempo real, representações de participantes reais em ambientes virtuais. Se necessário, humanos virtuais podem apresentar uma aparência bastante realista, baseada em princípios fisiológicos e biomecânicos. Restrições funcionais podem ser aplicadas a eles, a fim de que seus movimentos satisfaçam as limitações humanas. Além disso, ao invés de ficarem restritos a movimentos previamente definidos, humanos virtuais são capazes de comportar-se de forma autônoma e inteligente em ambientes dinâmicos, podendo apresentar até mesmo individualidade e personalidade [Badler *et al.* 1999].

Em aplicações como jogos ou na produção de animação, humanos virtuais podem ser utilizados como atores sintéticos. Um ator sintético, também chamado virtual ou digital, é definido como um ator autônomo, com aparência humana e completamente gerado por computador [Thalmann 1995]. Aplicações desse tipo requerem humanos virtuais autônomos e inteligentes, com capacidade de adaptação, percepção e memória. Além disso, para transmitir ilusão de vida, eles devem estar aptos a agir livremente e se

expressar emocionalmente, além de ser conscientes e imprevisíveis [Thalmann e Noser 1999].

Muitos projetos propostos implementam humanos virtuais com aparência e funcionalidade bastante realistas, usando superfícies deformáveis e mecanismos de animação baseados em Física, mas não se importam com o comportamento do agente. Outras abordagens propõem agentes cognitivos capazes inclusive de apresentar reações emocionais e personalidade própria, porém com representação gráfica pobre (textual em alguns casos). Existem ainda alguns trabalhos que procuram combinar a aparência realista com a credibilidade do comportamento, mas a maioria destes utiliza abordagens reativas para simular o comportamento.

Este artigo propõe uma arquitetura que combina um modelo articular bastante flexível para representação e animação de humanos virtuais com um interpretador para linguagem de programação de agentes cognitivos, o *AgentSpeak*. O sistema proposto permite implementar atores sintéticos autônomos que combinam raciocínio cognitivo usando lógica BDI com representação e movimentação realista em ambientes virtuais. É importante salientar que este ainda não é um sistema completo, visto que alguns fatores importantes, tais como percepção limitada do ambiente, personalidade e expressão de emoções, não estão sendo considerados. O objetivo deste trabalho limita-se, assim, ao desenvolvimento e teste de uma nova camada sobre o modelo de animação comportamental, oferecendo uma escolha cognitiva de ações usando lógica BDI. Na área de sistemas multiagentes, a arquitetura proposta pode servir para incorporar interpretação visual realista às simulações. Em aplicações gráficas, principalmente em jogos, possibilita implementar personagens com aparência realista contendo raciocínio cognitivo, visando melhorar a simulação do comportamento de participantes reais.

2. Definições

Agentes inteligentes têm sido usados em diversos projetos na implementação de personagens virtuais. Um aspecto fundamental nesses sistemas é a noção de credibilidade dos personagens, que está associada à capacidade que possuem de transmitir ilusão de vida. Em um primeiro momento, eram consideradas somente características ligadas à área de agentes, mas atualmente os projetos aliam o comportamento dos agentes a uma representação gráfica convincente dos mesmos.

O projeto Oz da Universidade Carnegie Mellon [Bates et al. 1992] introduziu o termo "agentes credíveis" (believable agents) na área de agentes como ele é usado na arte, significando que um observador pode ter a sensação de que um personagem fictício é real. O projeto Virtual Theater [Rousseau e Hayes-Roth 1997] desenvolve agentes que atuam como atores inteligentes, improvisando seu comportamento sem planejamento detalhado. De acordo com este projeto, um ator sintético possui uma "mente" e um "corpo" (Figura 1). A primeira é implementada através de um agente, o qual atualiza o conhecimento do ator baseado em sua percepção, controla suas decisões e ordena ao corpo que execute ações para mudar o mundo virtual. O corpo, por sua vez, corresponde à representação na tela das ações executadas pelo ator.

Existem diversas arquiteturas disponíveis para planejar e implementar agentes, desde os reativos até os cognitivos. Os agentes reativos são muito simples e não possuem representação do ambiente nem do seu conhecimento, ficando este implícito em suas regras de comportamento. Estes agentes não possuem memória de suas ações

passadas nem podem planejar seu comportamento futuro. Suas reações dependem unicamente da percepção do ambiente a cada instante. Os agentes cognitivos, ao contrário, mantêm uma representação explícita de seu ambiente e dos outros agentes presentes no sistema, têm memória do passado, podem comunicar-se diretamente e possuem um mecanismo de controle deliberativo, ou seja, raciocinam para decidir os objetivos a alcançar, os planos a seguir e as ações a executar [Alvares e Sichman 1997].

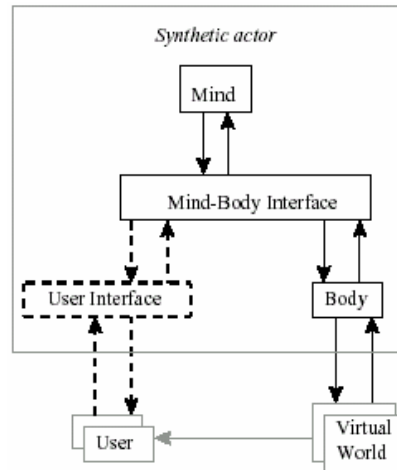


Figura 1. Arquitetura de um ator sintético [Rousseau e Hayes-Roth 1997]

Agentes BDI (**B**elief, **D**esire, **I**ntention) são um tipo particular de agente cognitivo, possuindo atitudes mentais como crenças, desejos e intenções (Figura 2). O embasamento filosófico do modelo BDI vem da teoria do raciocínio prático humano, que envolve dois processos importantes: decidir quais objetivos busca-se atingir (deliberação) e como agir para atingi-los [Wooldridge 1999]. Intuitivamente, crenças correspondem às informações que o agente possui sobre seu ambiente, desejos representam opções de estados futuros disponíveis ao agente (motivação) e intenções são os estados futuros que o agente escolheu e com os quais comprometeu-se (deliberação). O raciocínio prático do agente envolve repetidamente: atualizar as crenças a partir da percepção do ambiente; decidir quais opções estão disponíveis; "filtrar" estas opções para determinar novas intenções e agir baseado nestas intenções [Rao e Georgeff 1995].

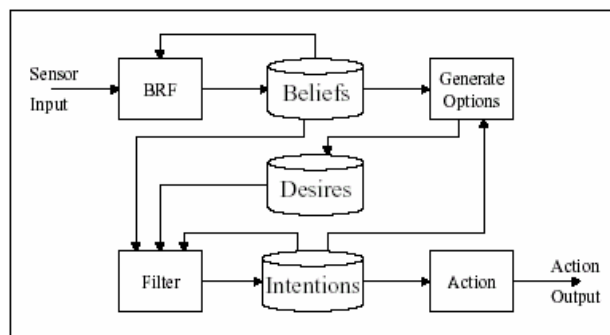


Figura 2. Arquitetura BDI (adaptada de [Wooldridge 1999])

A animação do corpo humano é uma tarefa bastante complicada dada sua forma irregular e de difícil representação. Além disso, a especificação e computação de seus movimentos não é trivial, posto que suas articulações complexas envolvem mais de 200

graus de liberdade. Normalmente, humanos virtuais são estruturados como corpos articulados definidos por um esqueleto composto por um conjunto de segmentos (correspondendo aos membros) conectados por articulações. Cada articulação pode ter de um a três graus de liberdade, sendo que seus ângulos determinam a posição de um segmento em relação a outro. O corpo é movimentado através da alteração dos ângulos das articulações e da sua posição global. O esqueleto pode ser coberto por uma superfície rígida ou então deformável durante os movimentos. Esta última opção é mais realista, mas acrescenta carga extra de modelagem e computação [Badler et al. 1999].

Humanos virtuais animados tanto podem ser controlados por pessoas reais, sendo chamados avatares, como podem representar figuras puramente sintéticas. No primeiro caso, os ângulos das articulações são capturados (usando vídeo ou captadores magnéticos ou óticos) e convertidos em rotações para o corpo virtual. No segundo caso, é preciso implementar programas para gerar as seqüências e combinações corretas e assim criar os movimentos desejados. Os procedimentos para alterar ângulos de articulações e posição do corpo são chamados geradores motores ou habilidades motoras. Algumas dessas habilidades típicas são [Badler et al. 1999]: mudanças de postura e ajuste de equilíbrio; alcançar, agarrar e outros gestos do braço; caminhar, correr, subir; observar; expressões faciais; pular, cair, balançar; combinar um movimento com o seguinte (*blending*).

Um humano virtual deve ser capaz de executar várias dessas atividades de forma simultânea, assim como fazem seres humanos reais. Badler [Badler et al. 1999] propõe uma máquina virtual paralela que anima modelos gráficos, chamada Parallel Transition Networks (PaT-Nets). Nessa máquina, os nodos da rede representam processos e os arcos contêm predicados, condições, regras ou outras funções que causam transição para outros nodos. Uma PaT-Net fornece um modelo de animação não-linear e possibilita capacidades de reatividade e tomada de decisão.

A animação a nível de tarefa (*task-level animation*) utiliza o mesmo conceito das habilidades motoras. Os comandos task-level precisam ser transformados em instruções de baixo nível para que o movimento correspondente seja executado. Comandos motores devem ser traduzidos para que a animação do personagem possa ser efetivamente executada através da alteração dos ângulos das articulações de seu esqueleto.

O movimento de caminhar, assim como agarrar, é uma tarefa essencial para a animação de um ator sintético. O ciclo do andar é geralmente dividido em duas fases que se alternam, uma delas correspondendo ao tempo em que o pé está em contato com o chão, e outra representando o balanço da perna para a frente. Cada braço balança para a frente acompanhando a perna oposta, e para trás enquanto a mesma está em contato com o chão. Multon [Multon et al. 1999] apresenta um estudo sobre diversas abordagens utilizadas para implementar esse movimento. Giang [Giang et al. 2000] salienta que movimentos cíclicos, tais como o caminhar, obtidos pela maioria dos mecanismos apresentam aparência artificial por natureza, pois não conseguem representar os pequenos defeitos humanos, necessários para transmitir realismo. Funções geradoras de ruído podem ser aplicadas para provocar perturbações, tentando simular esses defeitos.

A técnica de animação comportamental [Reynolds 1987] permite que um personagem seja animado a partir da modelagem de seu comportamento sem que o animador precise ocupar-se com a especificação dos movimentos que levam a tal resultado. Os comportamentos podem variar desde simples planejamento de caminho até interações emocionais complexas entre personagens. Em seu trabalho, Reynolds estudou em detalhe o comportamento de grupos de animais, tais como bandos de pássaros, cardumes de peixes e manadas de animais terrestres. A animação desse tipo de comportamento usando técnicas tradicionais é praticamente impossível. Na abordagem adotada por Reynolds, cada pássaro decide sua trajetória sozinho, sem intervenção do animador, e o comportamento do bando é o resultado da interação dos comportamentos individuais.

Um sistema de animação comportamental, em geral, é composto por um ambiente virtual e por atores autônomos que se movimentam nesse ambiente. O comportamento de um ator é influenciado pelas informações capturadas do ambiente sobre os objetos e sobre os outros atores participantes. Essa informação é usada pelo modelo comportamental para selecionar qual ação tomar, e esta resulta em um procedimento de controle motor [Thalmann 1996]. Modelos comportamentais podem ser usados para produzir animação de alto nível envolvendo seres humanos e animais. Alguns exemplos de comportamentos baseados em percepção a partir de sensores virtuais são locomoção dirigida pela visão, manipulação de objetos e resposta a sons e expressões.

3. Modelo de Corpo

A representação gráfica de humanos virtuais prevê uma estrutura hierárquica capaz de representar o conjunto de articulações que representa o corpo humano e um conjunto de funções utilizadas para gerar movimento sobre estas articulações.

Cada junta representa um conjunto de possibilidades de movimento, ou em outras palavras, um conjunto de graus de liberdade (DOF). Um grau de liberdade de uma junta é quem determina a presença de um movimento e como esse movimento se realiza. Uma junta com 2 DOFs pode realizar dois tipos de movimento, por exemplo: flexão/extensão e adução/abdução.

Para que se possa construir um corpo humano a partir de um conjunto de juntas, também é necessário determinar a topologia de tal conjunto. A solução utilizada aqui foi adotar a topologia de árvore, que se aproxima bastante da topologia articular humana, e estabelecer relações entre juntas-pais e juntas-filhas por meio de matrizes homogêneas 4x4, denominadas LIM¹. Desta forma, cada junta está definida com base no sistema de referência da sua junta-pai, à exceção da junta-raiz da árvore, definida com base no sistema de referência do universo (SRU) (ou qualquer outro sistema externo ao esqueleto, dependendo da aplicação). Caso seja necessário obter a posição e orientação de uma junta qualquer em relação ao SRU – para ser utilizada em algum procedimento de visualização, por exemplo – podem ser feitas multiplicações sucessivas das LIMs desde a raiz até a junta em questão, conforme mostra a equação abaixo:

$$M_{[J_n \rightarrow SRU]} = (LIM_{J_n} \times (LIM_{J_{n-1}} \times \dots \times (LIM_{J_2} (LIM_{J_1} \times LIM_{J_0}))))$$

¹ LIM = Matriz de Instanciação Local, do inglês: *Local Instance Matrix*.

Na equação, J_0 representa a junta-raiz e J_n representa uma junta qualquer do corpo. M é a matriz que determina a posição e a orientação da junta J_n em relação ao SRU, e é obtida pela multiplicação das LIMs de todas as juntas do ramo da árvore entre J_0 e J_n .

As juntas, propriamente ditas, não têm representação geométrica. Elas são importantes porque representam os relacionamentos matemáticos entre partes do corpo, sendo necessário utilizar outras estruturas para representar ossos, músculos e pele. Neste modelo, essas estruturas são representadas por uma abstração denominada *objeto gráfico* (primitiva geométrica ou malha de polígono que pode ser associada a qualquer junta da árvore que compõe um corpo). Cada um desses objetos também possui uma LIM, que serve para associá-lo a uma junta e definir sua posição e orientação em relação a essa junta. Desta forma, as estruturas representadas pelos objetos gráficos mudam de posição e/ou orientação de acordo com a posição da junta. Quando suas coordenadas em relação a um sistema de referência global são necessárias – para detecção de colisão ou no momento da sua exibição, por exemplo – elas podem ser obtidas pela multiplicação das LIMs desde a junta raiz até a junta onde está o objeto e, ainda, pela LIM do objeto. Este processo gera uma GIM^2 do objeto gráfico:

$$GIM_{O_m} = M_{[J_n \rightarrow SRU]} \times LIM_{O_m}$$

Na equação acima, O_m é um objeto gráfico qualquer, GIM_{O_m} é a matriz que representa a posição e orientação do objeto O_m em relação ao SRU, a matriz M é a mesma da equação anterior e LIM_{O_m} é a matriz que determina a posição e orientação do objeto O_m em relação à junta J_n .

O modelo articular apresentado acima [Maciel et al. 2002], foi implementado usando o paradigma de orientação a objetos, o que facilita a criação de novas classes que possam estendê-lo. Para representar a hierarquia articular do corpo, cada junta, retratada na classe *VPJoint*, possui uma referência para cada uma de suas juntas-filhas. Além dessa lista, outros atributos importantes de uma junta são um conjunto de DOFs, um conjunto de objetos geométricos associados a ela e uma matriz LIM. A classe *VPJoint* representa uma articulação genérica, contendo as propriedades que são comuns a todas as juntas humanas e os métodos necessários para movimentá-las. Outras classes mais especializadas são derivadas desta e representam tipos específicos de juntas humanas, como por exemplo: *VPPolyaxialJoint* e *VPPlanarJoint*. De acordo com os diferentes tipos, são definidos a quantidade e tipo dos DOFs, os limites angulares e outros parâmetros de movimento da junta.

Uma vez que se tenha definido e implementado um modelo de representação de corpos, a necessidade de aplicar movimentos sobre este corpo acaba surgindo naturalmente. Duas classes foram desenvolvidas para realizar esse trabalho. A classe *VPBodyMotion* permite associar um conjunto de movimentos a um corpo e controlar a execução desses movimentos. Portanto, um objeto desta classe tem associados a ele um corpo e os parâmetros da animação, como tempo atual, tempo final e intervalo de tempo entre quadros, além de uma lista de objetos de outra classe, *VPMotion*. Esta última representa cada uma das ordens individuais de movimentação, onde são armazenados o seu tempo inicial e final, a junta em questão, o tipo de movimento a ser realizado

² GIM = Matriz de Instanciação Global, do inglês: *Global Instance Matrix*.

(flexão, adução, rotação, etc.), o parâmetro que define o novo estado final da junta e a fração do parâmetro a ser aplicada em cada intervalo de tempo do movimento (ΔS).

Durante a execução do movimento, o tempo é contado em intervalos de tamanho definidos na *VPBodyMotion*. Toda vez que o tempo atinge o tempo inicial de um objeto da *VPMotion*, seu ΔS , que inicialmente tem o valor zero, é alterado para a fração adequada do parâmetro que define o estado final. A partir de então, a cada intervalo, o estado da junta é modificado de acordo com o ΔS até que o tempo seja igual ao tempo final do objeto *VPMotion*, quando ΔS volta a ser zero. Todas as informações necessárias para a representação do corpo articulado e também para a especificação dos movimentos das articulações são transmitidas ao modelo articular através de um arquivo XML.

O controle de movimento implementado através das classes *VPBodyMotion* e *VPMotion* exige especificação individual para cada junta, o que torna bastante ineficiente a geração de animações complexas. Para solucionar este problema, foram incorporadas duas novas classes com a finalidade de encapsular conjuntos de movimentos. Uma delas, *VPTask*, armazena os movimentos básicos (nível de tarefa), enquanto a outra, *VPBehavior*, reúne conjuntos destes movimentos, compondo comportamentos mais complexos e vinculados ao ambiente virtual. *Caminhar*, por exemplo, é um método de *VPTask*, enquanto *Visitar (azul)* é um método de *VPBehavior*.

4. Modelo Mental

Rao e Georgeff (1995) propõem uma arquitetura abstrata que segue a teoria do modelo BDI. Salientam, entretanto, que não é um sistema prático para agentes racionais, uma vez que não há indicação de como tornar os procedimentos de geração de opções e deliberação rápidos o suficiente para aplicações em tempo real. O tempo de raciocínio é potencialmente ilimitado, destruindo assim a reatividade, essencial para o agente.

Para resolver este problema, usam-se representações que limitam o poder do modelo, tornando-o em contrapartida, mais prático para agentes racionais. Entre outras medidas, as informações sobre os meios para atingir estados futuros do ambiente (objetivos), além das opções disponíveis ao agente, podem ser representadas por *planos*. O corpo de um plano descreve as ações primitivas que o agente deve executar ou outros objetivos que deve atingir para que o plano tenha sucesso. A satisfação de uma pré-condição e a ocorrência de um evento "disparador" (*triggering event*) são as condições para que um plano seja escolhido. Baseada nesta representação, foi criada a linguagem *AgentSpeak* [Rao 1996] para implementação de agentes BDI.

As definições de sintaxe de *AgentSpeak* são as seguintes:

- crenças são formadas por um predicado e um conjunto de termos;
- objetivos de realização são representados usando o símbolo “!” antes do predicado, e objetivos de teste utilizam o símbolo “?”;
- adição ou remoção de crenças ou objetivos são eventos disparadores e podem ser representados utilizando os símbolos “+” e “-”;
- o agente pode executar ações que modificam o estado do ambiente;

- planos são compostos por um cabeçalho e um corpo. O cabeçalho contém um evento disparador e um contexto separados por ":". O corpo é uma seqüência de objetivos ou ações.

Na arquitetura proposta, o raciocínio cognitivo dos humanos virtuais é implementado com o auxílio de um interpretador para a linguagem *AgentSpeak* desenvolvido pelo grupo de IA da UFRGS [Bordini et al. 2002], o qual obedece à sintaxe original da linguagem. O interpretador recebe como entrada inicialmente um conjunto de crenças e planos, que compõem um programa *AgentSpeak*. A cada passo do ciclo de execução, o interpretador pode receber ou não percepções do ambiente, as quais geram respostas de acordo com a lógica do programa.

5. Proposta de Interface *Mind-Body*

A comunicação entre corpo (articulado) e mente (raciocínio cognitivo) do humano virtual é implementada através de uma interface capaz de:

- receber informação de percepção no ambiente gráfico;
- transferir a informação para o interpretador *AgentSpeak* no formato desta linguagem;
- receber o resultado e interpretá-lo para decidir qual deve ser o comportamento gráfico adotado;
- executar este comportamento gráfico pré-definido;
- a partir de nova percepção, reiniciar o ciclo.

O interpretador *AgentSpeak* é executado em um processo servidor e sua comunicação com o sistema gráfico ocorre através de *sockets*. Esta solução evita a necessidade de incorporar o interpretador ao sistema, e assim os mantém independentes.

Uma vez que a percepção não faz parte do foco do projeto atual, o humano virtual terá acesso direto a informações internas do ambiente. As informações obtidas serão mapeadas para a sintaxe de *AgentSpeak* e enviadas ao interpretador. Neste, serão recebidas como *triggering events*, capazes de desencadear a execução de planos e a alteração do estado do mundo (crenças e objetivos). Esta alteração será então transferida de volta ao sistema gráfico e servirá para que este escolha qual comportamento deve ser executado. Nos sistemas de animação comportamental em geral, a decisão sobre o comportamento adotado face a determinada percepção é puramente reativa. Na arquitetura proposta, o processo de escolha de comportamento é dividido em duas etapas - o interpretador *AgentSpeak* é responsável pelo raciocínio cognitivo (lógica BDI) e gera um resultado, o qual serve de entrada para o sistema gráfico executar uma escolha reativa de comportamento.

6. Estudo de Caso

A seguir é apresentado um exemplo de execução do sistema. Um agente inserido em um ambiente virtual contendo cubos coloridos (Figura 5) precisa obter uma pista que inicia oculta em um dos cubos, o qual encontra-se trancado. A chave necessária para abri-lo está em um dos outros três cubos, que estão abertos. Portanto, para ter acesso à pista, além de descobrir em qual cubo ela está guardada, o agente precisa saber onde encontrar a chave.

As Figuras 3 e 4 ilustram o movimento de caminhar do humano virtual (em nível de tarefa). Os objetos gráficos escolhidos para representar o agente foram ossos, formando um esqueleto. Contudo, outros objetos podem ser aplicados sobre a mesma definição de articulações, bastando alterar informações no arquivo XML que define a representação e a movimentação do agente. A Figura 6 mostra o comportamento de caminhar até um cubo do agente inserido no ambiente.

A Figura 7 apresenta um exemplo simples de programa em *AgentSpeak*, usado para implementar a "mente" do agente. Este programa não faria muito sentido se não trocasse informações com o modelo de animação. As informações de entrada que recebe (em formato de crenças) avisam quando o agente chegou a determinado cubo, ou ainda quando um cubo contém uma chave ou uma pista. Sendo o objetivo inicial deste trabalho testar a funcionalidade e eficiência da interface, a lógica implementada está bastante simplificada. Em uma versão posterior, a descoberta da localidade da chave e da pista farão parte do raciocínio do agente.



Figura 3. Visão frontal



Figura 4. Visão lateral

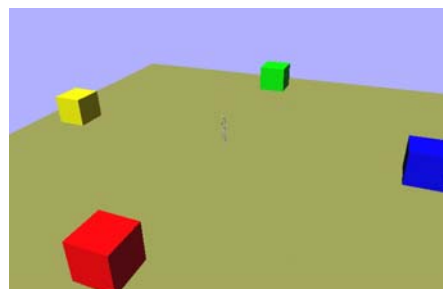


Figura 5. Ambiente virtual

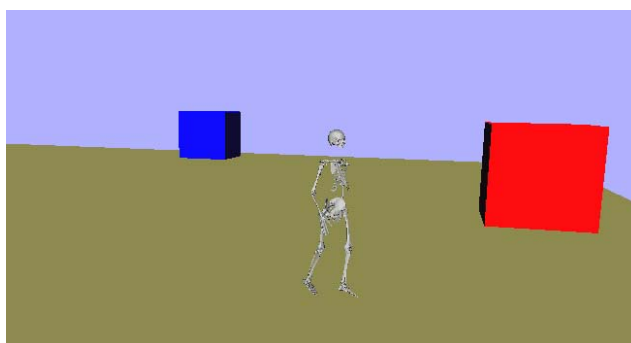


Figura 6. Humano virtual inserido no ambiente virtual

Visit e *take_track* são ações passadas ao sistema gráfico, o qual adota os comportamentos correspondentes. O último plano é o único disparado internamente, a partir de outro plano; os demais são acionados por entradas externas. *Cube* é uma variável que recebe a cor do cubo envolvido na entrada de percepção. Considerando uma situação com a chave guardada no cubo verde e com a pista escondida no cubo amarelo, a seqüência de passos executados é a seguinte:

- sistema gráfico passa ao interpretador o evento disparador *+contains_track (yellow)*;

- o segundo plano não pode ser executado, pois a condição *has_key* não é verdadeira. Assim, o terceiro plano é executado e cria a crença *track_in (yellow)*;
- o sistema gráfico passa ao interpretador o evento disparador *+contains_key (green)*;
- o primeiro plano é executado, e a ação *visit (green)* é transmitida ao sistema gráfico;
- o agente caminha até o cubo verde; ao chegar, transmite ao interpretador o evento *+arrived (green)*;
- o quinto plano é executado, cria a crença *has_key* e ativa o sexto plano;
- no sexto plano, o teste obtém a crença *track_in (yellow)* como resultado e transmite a ação *visit (yellow)* ao sistema gráfico;
- o agente caminha até o cubo amarelo e transmite ao interpretador o evento *+arrived(yellow)*;
- as duas condições do quarto plano são satisfeitas, e a ação *take_track* é transmitida ao sistema gráfico;
- o agente executa a ação de pegar a pista e encerra o exemplo.

```

+contains_key (Cube):
    not has_key ←
    +key_in (Cube);
    visit (Cube).
+contains_track (Cube):
    has_key ←
    +track_in (Cube);
    visit (Cube).
+contains_track (Cube):
    true ←
    +track_in (Cube).
+arrived (Cube):
    track_in (Cube) & has_key ←
    take_track.
+arrived (Cube):
    key_in (Cube) ←
    +has_key.
+has_key:
    track_in (Cube) ←
    visit (Cube).

```

Figura 7. Exemplo de Programa AgentSpeak

7. Conclusões e Considerações Finais

Este trabalho discute uma proposta de arquitetura para simulação de humanos virtuais, considerando agentes com movimentos articulados e realistas, e tomadas de decisão baseadas em lógica BDI. Apresentou-se uma pequena revisão bibliográfica tanto na área de animação como em sistemas multiagentes, sendo que o foco do trabalho centrou-se na interface entre estas duas áreas. Apesar de se tratar de trabalho em desenvolvimento,

alguns resultados preliminares foram apresentados com o objetivo de demonstrar a viabilidade da arquitetura.

A escolha do modelo BDI para implementar a tomada de decisão dos agentes, deve-se ao fato desta ser a mais importante arquitetura para simulação de agentes cognitivos na área de sistemas multiagentes. Atualmente, a grande maioria das tentativas de criação de linguagens de programação próprias para agentes (*agent-oriented programming*) são baseadas na arquitetura BDI. *AgentSpeak*, por sua vez, é uma dessas linguagens e está sendo utilizada por um grupo na UFRGS, cujo trabalho vem obtendo ótima receptividade junto a comunidade mundial. Em relação ao estudo de caso apresentado, para tirar maior proveito do modelo BDI, pretende-se criar um ambiente com pelo menos dois agentes e aumentar a complexidade de seu raciocínio, a fim de permitir que entrem em competição na busca pela pista escondida.

Outro ponto importante para análise é o tempo de resposta do sistema. Considerando que a arquitetura foi projetada para uso em jogos e ambientes virtuais interativos, faz-se necessária a execução em tempo real. O uso de *sockets* para comunicação entre os processos permite separá-los, inclusive executando os processos em máquinas diferentes. Considerando que ambos sistemas (animação e tomada de decisão) apresentam desempenho satisfatório, serão então avaliados executando em uma única máquina. Se a resposta utilizando *sockets* não for satisfatória, será avaliada outra alternativa, provavelmente envolvendo a incorporação do interpretador *AgentSpeak* no código do sistema responsável pela animação.

Uma sugestão de trabalho futuro é a implementação de um modelo de percepção como alimentador do *AgentSpeak*, a fim de tornar os humanos virtuais mais realistas e convincentes. Diversos modelos de percepção foram apresentados, nos quais os agentes precisam interpretar os sinais recebidos a partir de seus sensores virtuais, ao invés de simplesmente receber informações prontas do ambiente [Terzopoulos e Rabie 1997], [Noser e Thalmann (1995)], [Thalmann e Noser 1999].

Referências

- L. O. Alvares, J. S. Sichman, "Introdução aos Sistemas Multiagentes", Jornada de Atualização em Informática (JAI 97), XVII Congresso da Sociedade Brasileira de Computação, Brasília, 1997.
- N. Badler, R. Bindiganavale, J. Bourne, J. Allbeck, J. Shi and M. Palmer. "Real time virtual humans", International Conference on Digital Media Futures, Bradford, UK, abril de 1999.
- J. Bates, A. B. Loyall, W. S. Reilly, "An Architecture for Action, Emotion and Social Behavior", Technical Report CMU-CS-92-144, School of Computer Science, Carnegie Mellon University, 1992.
- R. H. Bordini, A. L. C. Bazzan, R. O. Jannone, D. M. Basso, R. M. Vicari and V. R. Lesser, "AgentSpeak(XL): Efficient Intention Selection in BDI Agents via Decision-Theoretic Task Scheduling", Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002, featuring 6th AGENTS, 5th ICMAS and 9th ATAL), 15-19 July, 2002, Bologna, Italy, to appear.

- T. Giang, R. Mooney, C. Peters, C. O'Sullivan, "Real-Time Character Animation Techniques", Image Synthesis Group Trinity College Dublin, Technical Report TCD-CS-2000-06, fevereiro de 2000.
- A. Maciel, L. P. Nedel, C. M. D. S. Freitas. "Anatomy Based Joint Models for Virtual Humans Skeletons. In: IEEE COMPUTER ANIMATION, 2002, Geneva.
- F. Multon, L. France, M-P. Cani-Gascuel, G. Debunne, "Computer animation of human walking: a survey", *The Journal of Visualization and computer Animation*, Vol. 10, pp 39-54, 1999.
- H. Noser, D. Thalmann, "Synthetic Vision and Audition for Digital Actors", *Proc. Eurographics '95*, Maastricht, agosto de 1995, pp. 325-336.
- A. S. Rao, "AgentSpeak (L): BDI Agents Speak Out in a Logical Computable Language", *Proceedings of the Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '96)*, 22-25 de janeiro, Eindhoven, Holanda, Springer Verlag, 1996, pp. 42-55, LNAI 1038.
- A. S. Rao, M. P. Georgeff, "BDI Agents: From Theory to Practice", *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, São Francisco, CA, 312—319, 1995.
- C. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model", *Proc. SIGGRAPH 1987, Computer Graphics*, Vol. 21, Nº 4, 1987, pp. 25-34.
- D. Rousseau, B. Hayes-Roth, "Improvisational Synthetic Actors with Flexible Personalities", *Knowledge Systems Laboratory, Report nº KSL 97-10*, Stanford, CA, dezembro de 1997.
- D. Terzopoulos, T. Rabie, "Animat Vision: Active Vision in Artificial Animals", *Videre: Journal of Computer Vision Research*, 1(1):2-19, 1997.
- D. Thalmann, "Autonomy and Task-Level Control for Virtual Actors", *Programming and Computer Software*, Nº 4, Moscou, Rússia, 1995.
- D. Thalmann, "Physical, Behavioral, and Sensor-Based Animation", *Proc. Graphicon 96*, St Petersburg, Russia, 1996, pp 214-221.
- D. Thalmann, H. Noser, "Towards Autonomous, Perceptive, and Intelligent Virtual Actors", in: *Artificial Intelligence Today, Lecture Notes in Artificial Intelligence*, Nº 1600, Springer, pp. 457-472, 1999.
- M. Wooldridge, "A Modern Approach to Distributed Artificial Intelligence", *Capítulo I - Agentes Inteligentes*, editado por G. Weiss, 1999.
- M. Wooldridge, N. R. Jennings, "Intelligent Agents: Theory and Practice", *The Knowledge Engineering Review*, 10(2):115-152, 1995.