

Ambientes Virtuais Contextualizáveis Baseados em Jogos de Computador em Educação¹

Flavio Shigeo Yamamoto

Flávio Soares Corrêa da Silva

Instituto de Matemática e Estatística – Universidade de São Paulo

Rua do Matão, 1010 – CEP – São Paulo – SP – Brasil

{fsy, fcs}@ime.usp.br

Abstract. *This paper reports an experiment in the course of Artificial Intelligence using the Gun-Tactyx as an educational tool for practical learning of advanced concepts and techniques in Artificial Intelligence. We also present a virtual contextualizable environment based on multi-agents, with the main idea being the interaction of Non-Player Characters. This environment can be immersed in different scenarios. Moreover, we propose a formal structure that integrates this environment which can be used in different kinds of application, like education and arena for investigating, experimenting, validating (in Software Engineering sense) and fast-prototyping of synthetic characters based on advanced concepts and techniques in Artificial Intelligence.*

Key-words: *computer games, digital entertainment, interactive spaces, multi-agent systems, situated agents.*

Resumo. *Este texto traz o relato de um experimento na disciplina de Inteligência Artificial utilizando o Gun-Tactyx como ferramenta para aprendizado prático de conceitos e técnicas avançadas em Inteligência Artificial. Também apresenta a proposta de um ambiente virtual contextualizável baseado em multi-agentes que pode ser revestido de diferentes cenários, permitindo a construção de uma gama variada de jogos centrados na interação de Personagens Não Jogadores. Além disso, propomos uma estrutura formal que integra o ambiente virtual permitindo desde usos mais amplos em educação até aplicações específicas como arena de teste para validação (no sentido da Engenharia de Software) e prototipações-rápidas de personagens sintéticos baseados em conceitos e técnicas avançadas em Inteligência Artificial.*

Palavras-chave: *agentes-situados, ambientes interativos, entretenimento digital, jogos de computador, sistemas multi-agentes.*

¹Projeto financiado parcialmente pela *Microsoft Research*, através do LIDET - *Laboratory of Interactivity and Digital Entertainment Technology*. DCC – IME – USP.

1. Introdução

As atividades descritas neste texto fazem parte dos projetos gerenciados pelo LIDET - *Laboratory of Interactivity and Digital Entertainment Technology* [Silva 2005 (a)]. Os projetos estão centrados em estudos teóricos em Inteligência Artificial (IA) e no desenvolvimento de aplicações em interatividade e tecnologias de entretenimento digital. A indústria de jogos digitais se direciona para inovações que estejam além da riqueza da *arte gráfica*, as tendências para as novas gerações de jogos se mostram favoráveis a verdadeiras aplicações de conceitos e técnicas avançadas de IA [Laird 1999, 2000; Blumberg 2002; Sloman 2000]. As ações do LIDET estão de conformidade com essa tendência e suas propostas e projetos a colocam numa posição de que visa aproximar a área acadêmica de IA à indústria de jogos e entretenimento digital.

Para alcançar essa meta, os projetos do LIDET vão desde novas práticas nos cursos de *Engenharia de Software* (ES) [Nakanishi 2005] e IA [Silva 2005 (a)] a desenvolvimento de ambientes virtuais contextualizáveis baseados em multi-agentes [Yamamoto 2005] (incluindo, estudos de combinações e extensões das lógicas modais (como as lógicas híbridas) para prover os agentes com características lógico-dedutivas - Raciocínio Espacial Qualitativo (REQ) [Forbus 1983, 2001]). Tais ambientes virtuais, com agentes providos ou não de REQ, podem ser revestidos de diferentes cenários, permitindo a construção de uma variedade de jogos centrados na interação de Personagens Não Jogadores. Esse ambiente virtual pode ser utilizado como arena de teste para validação, no sentido da ES, e prototipações-rápidas de personagens sintéticos baseados em conceitos e técnicas avançadas em IA. O uso desses recursos, combinado às práticas das disciplinas de ES e IA cria uma expectativa maior em formar futuros cientistas da computação que produzam jogos cujos personagens sintéticos tenham credibilidade em aspectos relativos à IA.

Este texto traz o relato do uso de um ambiente virtual baseado em jogo de computador, o *Gun-Tactyx Programming Game* [Boselli 2005], como ferramenta para aprendizado prático de conceitos e técnicas avançadas em IA. Também, apresentamos uma proposta de um ambiente virtual contextualizável baseado em multi-agentes [Silva 2005 (b)] e propomos uma estrutura formal que integra tal ambiente virtual permitindo que os agentes possuam REQ, maiores detalhes sobre essa proposta podem ser encontradas em [Yamamoto 2005].

O texto segue o seguinte esquema: na Seção 2 descrevemos o uso do *Gun-Tactyx* como um ambiente para observar implementações de técnicas avançadas em IA. O *Gun-Tactyx* pode ser utilizado, por exemplo, como um Sistema Multi-Agente (SMA) baseado em conhecimento para tratar tópicos como *arquitetura bottom-up*, *revisão de crenças* (versão simplificada de um TMS – *Truth Maintenance System*) [Jorge 2005 (a) e (b)] e *Agentes Celulares Situados* (ACS) [Bandini 2002, 2005; Vizzari 2004]. Na Seção 3 apresentamos um ambiente que é mais que uma simples extensão do *Gun-Tactyx*, o projeto encontra-se em fase intermediária de desenvolvimento [Silva 2005 (b)]. Na Seção 4 introduzimos os conceitos básicos de uma estrutura formal para integrar o ambiente proposto na seção anterior, tornando-o uma poderosa ferramenta de uso tanto para o desenvolvimento de personagens sintéticos com características de IA quanto para um uso mais amplo em educação [Yamamoto 2005]. Finalmente, na Seção 5, apresentamos resultados obtidos em relação ao uso do *Gun-Tactyx* e tecemos um breve

comentário sobre as potencialidades do uso de ambientes virtuais contextualizáveis baseados em jogos para promover novas práticas na área do ensino.

2. O Gun-Tactyx

O *Gun-Tactyx* possui vários módulos de jogo (FIGHT, SOCCER, RACE), no nosso caso optamos pelo modo FIGHT – exhibe uma situação de combate. Cada time (2 a 4 times) é formado por *agentes* (*warriors* ou *bots* – 1 a 64). Há diversos cenários (de com múltiplos andares, rampas, paredes e pontes), nós escolhemos a ARENA. Cada *agente* possui uma CPU de 10KHz e 32KB RAM e tem a sua disposição armas que disparam balas e lançam granadas. A percepção dos agentes ocorre através de sensores para *visão* (*sight* e *watch*) e *audição* (*hear* e *listen*). Por exemplo, `hear(&int:item,&int:sound,&float:yaw=0.0,&float:pitch=0.0,int:id=0)` é tal que *item* é um objeto do tipo `ITEM_GUN`, `ITEM_WARRIOR`, ou 0 (para o caso de não ouvir nada); *sound* identifica o som percebido (e.g. `SOUND_SHOT` ou uma *palavra* sendo dita, neste caso, *id* identifica o *warrior* que proferiu a *palavra*); *yaw* e *pitch* identificam o local (coordenadas dentro da arena) de origem do som. Cada *bot* pode executar um conjunto básico de *movimentos* rotacionar o torso (a arma está anexa ao torso), podem andar, correr, ficar parados (*walk*, *run*, *standing*), etc. Um jogo pode durar de um minuto a uma hora, mas, em geral, termina assim que um time elimina todos os outros.

Cada soldado é programado individualmente, usando-se uma linguagem de *script* chamada SMALL, (atual PAWN, vide em www.compuphase.com/small.htm) com funções pré-definidas (*sight*, *watch*, *walk*, *run*, *standing*, etc).

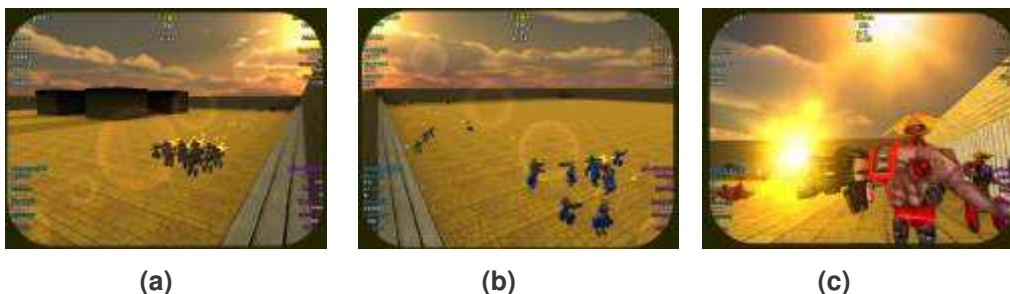


Figura 1. Cenas de ação no Gun-Tactyx: em (a) vemos o modo Arena, em (b) times em confronto e (c) mostra um *warrior* disparando sua arma. Fonte: gameprog.it/hosted/guntactyx/infolt.php.

As duas subseções a seguir descrevem duas atividades que foram construídas sobre o *Gun-Tactyx* e que exploram conceitos e técnicas avançadas em IA. A plataforma mostrou-se interessante, justamente pela linguagem SMALL oferecer pouquíssimos recursos as implementações exigiram muita criatividade, característica fundamental para qualquer bom profissional.

2.1. Agentes que Agem Baseados em Crenças

Apresentamos nesta seção uma versão simplificada de um TMS, o *Assumption-based TMS* (ATMS) [Jorge 2005 (a), 2005 (b)] implementado no *Gun-Tactyx*. Através dos sensores os *agentes percebem e avaliam* (confrontam suas *observações* com suas *crenças* – *fazem a manutenção ou revisão de suas crenças*) a situação atual do mundo

de acordo com aquilo que eles supõem ser verdade (*crenças*). O resultado de tal avaliação é a escolha (*deliberação*) de uma ação adequada à representação ideal (*ambiente*) que possuem do mundo. A Figura 2 exibe a estrutura geral do ATMS [Jorge 2005 (a), 2005 (b)].



Figura 2. Esquema geral da implementação.

Uma *suposição* (*assumption*) é uma *proposição primitiva* p que é suposto ser *verdadeira* pelo agente. Cada proposição primitiva estabelece uma *crença*. Dado um conjunto S de suposições, qualquer subconjunto é chamado de *ambiente*, cada suposição retrata a *imagem interna* que um agente possui do mundo, nym dado instante. Esta imagem refere-se a aquilo que o agente percebe através de seus *sensores*. Assim, um conjunto de uma ou mais suposições descreve um cenário no qual o agente deve deliberar uma ou mais ações.

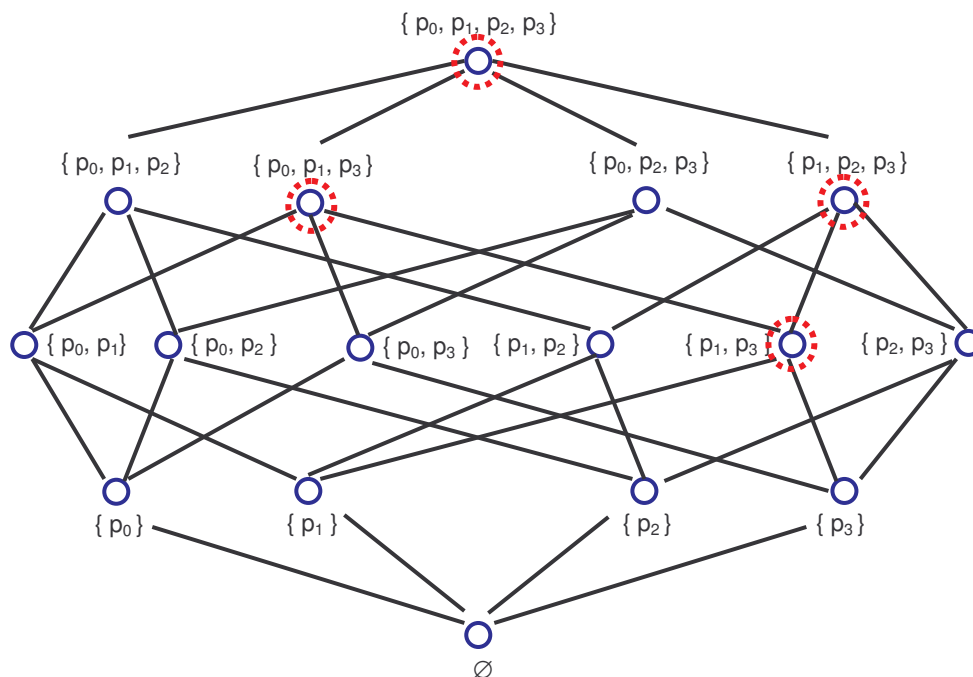


Figura 3. Estrutura relacional entre ambientes onde p_1 e p_3 não são *plausíveis*. Isto é, p_1 e p_3 levam a uma situação implausível. Assim, se um agente avaliar uma dada situação e estabelecer o ambiente $\{p_0, p_1, p_3\}$ como base para suas ações ele deve obrigatoriamente rever sua avaliação, a escolha de acordo com o ATMS seria $\{p_0, p_1\}$ ou $\{p_0, p_3\}$ uma vez que $\{p_1, p_3\}$ é implausível [Jorge 2005 (a), 2005 (b)].

A percepção dos agentes pode estabelecer a *veracidade* ou *falsidade* de certas suposições, o que pode levar a uma representação conflitante, isto é, o agente pode

chegar a um *ambiente implausível* (grosso modo, é um ambiente onde duas ou mais informações se *contradizem*). Para que o agente possa deliberar uma ação ele deve atingir um ambiente *consistente, livre de contadições*. Caso a avaliação resulte num ambiente inconsistente a , o agente escolhe um dos a_{i_s} tal que $a_i \subseteq a$ e a_i é *maximal*. Essa escolha pode ser vista como uma *reavaliação que o agente faz a partir de suas crenças*. A Figura 3 exibe um exemplo.

Seguindo o esquema apresentado das figuras 2 e 3 o que é percebido pelo agente é associado às suposições. A percepção se dá pelos sensores de *visão* (*sight* e *watch*) e *audição* (*hear* e *listen*) que estimulam o agente a confrontar o mundo descrito contra a imagem do mundo estabelecida a partir das suposições estabelecidas. Aquilo que o agente assume determina cada uma das ações. As ações (*walk*, *run*, *standing*, *shoot*, etc) descrevem o comportamento do *bot*. Ou seja, uma vez que os sensores determinaram quais suposições são verdadeiras, o *bot* delibera suas ações, porém antes verifica se, de fato, o que foi observado é plausível. A implementação desse cenário está disponível em [LIDET 2005] (há outra versão em [Jorge 2005 (a)]).

2.2. Agentes Celulares Situados

Simular situações de ações coordenadas sobre entidades autônomas, que se orientam pela percepção que extraem do mundo não é tarefa fácil. Uma forma de construir tais simulações é através de *Agentes Celulares Situados*. Esta seção expõe o uso dessa técnica aplicada à construção de *bots* para o *Gun-Tactyx*.

Um modelo via ACS é uma estrutura computacional para caracterizar sistemas onde entidades autônomas interagem entre si e com o ambiente considerando a percepção que extraem do mundo. Um *modelo ACS* é uma tripla $\langle Space, F, A \rangle$, intuitivamente a leitura para esta estrutura é: *os agentes em A, situados em Space, interagem de forma autônoma através de ações que se baseiam na percepção dada pela propagação dos campos em F*.

Vamos detalhar os elementos do modelo ACS. Um *Space* é um grafo $\langle S, E \rangle$ (uma estrutura relacional, como aquela da Figura 3) não orientado cujos nós são chamados de *sites*. O *Space* caracteriza o ambiente onde o agente está imerso, cada agente está situado em um único *site*. As arestas em E indicam a relação de vizinhança que se estabelece entre os *sites*. Todo $s \in S$ contém no máximo um agente. Um *site* é uma tripla $\langle a_s, F_s, S_s \rangle$ definido como segue: (i) $a_s \in A \cup \{\perp\}$ é um agente situado em s ($a_s = \perp$ significa que s está vazio); (ii) $F_s \subseteq F$ é o conjunto de campos que estão ativos em s ($F_s = \emptyset$ significa que não há campos ativos em s). Cada elemento de F é gerado por influência de um ou mais agente; e (iii) $S_s \subseteq S$ é o conjunto de *sites* adjacentes à s .

O termo A representa um conjunto de agentes. Um agente é uma tripla $\langle e, s, \tau \rangle$ com τ determinando o *tipo* do agente, $e \in \Sigma_\tau$ denota o *estado* do agente (isto é, assume um dos valores especificados pelo seu tipo) e $s \in S$ é o *site* do *Space* onde o agente está situado. O tipo de um agente é determinado por aspectos relacionados à *percepção*, *habilidades* e *comportamento*, é uma tripla $\langle \Sigma_\tau, Perception_\tau, Action_\tau \rangle$: (i) $\Sigma_\tau \subseteq \Sigma$ é o conjunto de *estados* que um agente do tipo τ pode assumir; (ii) $Perception_\tau : \Sigma_\tau \rightarrow [N \times W_{f_1}] \dots [N \times W_{f_{|F|}}]$ é uma função que associa a cada *estado do agente* um vetor de pares representando o *coeficiente de percepção* e *pontos de sensibilidade* para cada tipo

de campo existente; e (iii) $Action_t$ o comportamento de um agente é determinado pelas ações que executa, e elas podem ser caracterizadas por *primitivas* como (cada uma das primitivas exige uma especificação das pré-condições a serem cumpridas para que possam se estabelecer): (iii.a) $emit(e, f, s)$: emite um campo f do site s , onde está situado, num dado estado e ; (iii.b) $react(e, a_{s_1}, a_{s_2}, \dots, a_{s_n}, e')$: se uma ação é compatível com todos os agentes em $s_1, s_2, s_3, \dots, s_n$ (*sites vizinhos*), então os agentes $a_{s_1}, a_{s_2}, a_{s_3}, \dots, a_{s_n}$ devem modificar seu estado de e para e' (retrata uma espécie de comportamento coordenado, coordena a mudança de estado desses agentes); (iii.c) $trigger(e, e')$: primitiva que especifica a mudança de estado do agente de e para e' . Deve-se certificar das condições para esta mudança (avaliar os *sites vizinhos* e *campos atuantes*). É igual ao $react()$ porém somente para o agente em questão; e (iii.d) $transport(s, s')$: move o agente de s para s' . Evidentemente, verifica se s' é vizinho de s e que s' está vazio.

Todas as primitivas acima determinam o comportamento dos agentes, tudo isso deve ser especificado na construção dos agentes (*e.g.* especificar o campo a ser emitido por $emit()$ ou as condições para identificar o *site* destino em $transport()$).

O termo F representa um conjunto de direfentes *tipos de campos*. Para que seja possível a *percepção*, cada agente é provido de um conjunto de sensores que permitem aos agentes interagir entre si e com o meio em que está imerso. A interação pode originar um ou mais *tipos de campos* que atuam e se propagam pelo ambiente (podem existir *tipos de campos* que são naturais do próprio ambiente, por exemplo, a *força gravitacional*. No caso do *Gun-Tactyx* ela modifica a trajetória de qualquer projétil). Um *tipo de campo* definido por uma quadrupla $\langle W_t, Diffusion_t, Compare_t, Compose_t \rangle$: (i) W_t é o conjunto de *valores* que os campos do tipo t podem assumir; (ii) $Diffusion_t : S \times W_f \times S \rightarrow (W_t)^+$ é uma função de *difusão* de um campo que associa o valor de um campo ao valor com que foi inicialmente gerado, levando-se em conta o campo atuando em um *site* $s \in S$ específico; (iii) $Compose_t : (W_t)^+ \rightarrow W_t$ para um dado tipo de campo, expressa como este é combinado (*e.g.* como obter o valor de um campo do tipo t num dado *site*); e (iv) $Compare_t : W_t \times W_t \rightarrow \{ true, false \}$ para um dado tipo de campo, compara seus valores. Útil para verificar se um agente pode perceber um campo comparando a sensibilidade a ele com o coeficiente de percepção ao campo.

Cada um dos itens acima definidos devem ser interpretados no *Guntactyx*. O *Space* refere-se ao módulo ARENA, uma área quadrada de 131,5 metros de comprimento (ou 4208 unidades) - valores (em ponto-flutuante). Um modo de delimitar espacialmente um *site* seria converter a arena num reticulado, por exemplo, de 42×42 unidades, onde cada unidade equivale a uma área de 3,131×3,131 m² – um *site* está espacialmente localizado de acordo com essas limitações. Assim, o grafo $\langle S, E \rangle$ não orientado refere-se à arena subdividida em *sites* e as arestas em E indicam a relação de *vizinhança (adjacência espacial)* que se estabelece entre os *sites*. Cada $s \in S$ deve conter no máximo um *bot*.

Pelas limitações do SMALL a implementação do ACS no *Guntactyx* pode ser simplificada. Por exemplo considerar a capacidade de *percepção* e *sensibilidade* (*e.g.* de um campo sonoro ou visual) igual para todos os *bots*; ou a não perda na *propagação dos campos* pelos *sites*, isto é, os valores para os campos serão 0 ou 1 (se o *bot* perceber (isto é, ouvir ou ver) o campo assume valor 1, caso contrário assume 0). Logo, as

funções *difusão* e *composição* são tais que $W_i^+ = W_i$, e se comportam como a *função identidade*. Analogamente, a função *comparação* não retrata nenhum tipo de interferência. A implementação está disponível em [Silva 2005 (a)].

3. Um Ambiente Virtual Baseado em Jogo de Computador

O *Gun-Tactyx* mostrou-se um ambiente muito interessante para desenvolver atividades onde se pode simular e observar o comportamento de múltiplos agentes. Porém alguns conceitos sofisticados em IA como simular tomadas de decisão via redes bayesianas ou técnicas baseadas na teoria de Dempster-Shaffer não puderam ser desenvolvidos. Para suprir nossas necessidades didáticas estamos trabalhando no desenvolvimento de um ambiente virtual onde os alunos podem efetivamente implementar e visualizar agentes inteligentes em ação.

O sistema que está sendo desenvolvido consiste em um jogo estilo RPG – *Role Playing Game*, nesse tipo de jogo a idéia é imergir o jogador em um mundo imaginário onde ele tenha que interpretar papéis. Em geral o jogador tem a possibilidade de escolher entre vários tipos de personagens possíveis, como um guerreiro ou um mago [Silva 2005 (b)]. Basicamente, o ambiente virtual é povoado por personagens (agentes) cujo comportamento é orientado externamente por programas Prolog, acessados por meio de interfaces pré-definidas em tempo de execução. A escolha do estilo RPG propicia um amplo leque de aplicação da IA em relação a outros estilos como ação, onde papel central da IA está em fazer com que os *bots* ganhem do jogador humano. Problemas sofisticados podem abordados, por exemplo, a tridimensionalidade, que é um atrativo nesse tipo de ambiente, é uma forma de abordar conceitos complexos relacionados a visão dos agentes.

Para a construção de um jogo tridimensional estamos investigando a *engine 3D Ogre* [Streeting 2005], que é orientada a cenas. Escrita em C++ e projetada para que o desenvolvimento de aplicações que utilizem aceleração 3D de hardware seja mais fácil e intuitivo. A *Ogre* abstrai todos os detalhes de utilização das bibliotecas *Direct3D* e *OpenGL* e pode ser compilada para *Windows* e sistemas baseados em *Unix*. Não se trata de uma *game engine*, mas sim uma solução genérica para renderização em tempo real.

As atividades de implementação sobre o ambiente virtual seriam feitas de modo que o programa elaborado pelo aluno seja integrado ao ambiente em tempo de execução, dessa forma o aluno não precisaria tocar no código do jogo em si, e sim apenas escrever seu código-fonte em uma linguagem como Prolog [Silva 2005 (b)].

Esse ambiente virtual que descrevemos está em fase intermediária de desenvolvimento, os trabalhos estão sendo coordenados pelo Prof. Flávio S. C. Da Silva e executados por seu orientado de mestrado Filipe C. L da Silva, maiores detalhes podem ser encontrados em [Silva 2005 (b)]. De fato, esse ambiente serve de base para outras extensões, a seção seguinte trata de uma delas.

4. Ambientes Contextualizáveis Baseados em Agentes Situados

O raciocínio dos agentes que povoam o ambiente proposto na Seção 3 é baseado no *motor de inferência* do Prolog. A idéia da contextualização está em integrar arcabouços à estrutura do ambiente virtual que ofereçam recursos para processos de *inferência* de diferentes tipos (emocional, afetiva, qualitativa, espacial, etc). Nesta seção

descrevemos de modo breve uma proposta para um arcabouço que dê suporte ao Raciocínio Espacial Qualitativo.

A interação, seja do personagem com o ambiente, entre personagens ou com usuários externos ao mundo virtual é um aspecto chave em qualquer *Sistema Multi-Agente* (SMA). A essência de um SMA reside no fato da dinâmica global do sistema emergir de um comportamento local e interações entre as partes que a compõem [Bandini 2002]. Em nosso caso, estamos interessados em construir um arcabouço que comporte ambientes virtuais baseados em agentes situados que se orientem influenciados pelas *relações espaciais*, entre as entidades espaciais que caracterizam o cenário em que o agente está imerso.

Nossa proposta utiliza a *linguagem modal* para capturar a noção de *conhecimento* [Yamamoto 2003]. Baseado nessa linguagem, inserimos características que permitam o raciocínio qualitativo - *obter resultados significativos a partir de pouca quantidade de informações*. Ainda, em termos implementacionais buscamos o *raciocínio contextualizado* [Ghidini 2001] para reforçar o conceito de *Agentes Situados* [Bandini 2002, Vizzari 2004]. Não descartamos a possibilidade de uma *lógica intensional* [Chrz 1982; Wadge 1999], pelas características do raciocínio qualitativo seria interessante que as partes (segundo o critério da *composicionalidade*) fornecessem mais conteúdo do que aquilo que eles simplesmente denotam.

Note-se que, fixado um cenário (*estado de mundo*), devemos saber como compor as representações de diferentes aspectos de um cenário ou como compor um fenômeno nesse cenário, já que o agente necessita, a partir deles, criar sua própria representação interna [Forbus 1983]. No caso da semântica das lógicas quase tudo é feito a partir do *critério de composicionalidade* de Frege: *a semântica de uma expressão, numa dada linguagem, é baseada na sua respectiva construção sintática, a semântica das partes determina o que a expressão diz*. A *lógica intensional* enriquece as linguagens formais, pois leva em conta a *intensão* das sub-expressões (não somente suas *extensões*), tal como ocorre na *linguagem natural* [Chrz 1982].

Consideramos o raciocínio dos agentes como uma atividade local (utiliza somente parte do que está potencialmente disponível, aquilo que é conhecido ou que é obtido por inferência). Evidentemente, esta atividade local deve ser *compatível* em cada cenário distinto em que o agente irá se situar, tal *compatibilidade* poderá ser estabelecida através de um conjunto de crenças (que pode sofrer revisões) e/ou em *pontos de vistas* que o agente tem de um dado estado de mundo. Daí, *raciocínio contextual = localidade + compatibilidade* [Ghidini 2001].

A proposta acima descrita está em fase de estudo, estamos desenvolvendo os primeiros estudos sobre as lógicas modais (epistêmica, espacial e híbridas) para podermos alcançar uma sistematização formal que nos permita expressar o conhecimento espacial de modo que agentes situados possam ser providos de REQ.

5. Comentários Finais

O fato dos alunos poderem visualizar os diferentes comportamentos dos agentes que eles próprios codificaram, baseados em conceitos e técnicas de IA, é um fator que auxilia no processo de aprendizado. Dependendo das características dos jogos é possível elaborar torneios entre alunos (a competitividade é um fator que estimula o

aprendizado), trabalhar as melhorias em personagens “ruins”, etc. Evidentemente, é possível potencializar o uso de tais ambiente virtuais através da adoção de metodologias de ensino diferenciadas. Por exemplo, esse tipo de ambiente virtual oferece recursos para explorar a interdisciplinaridade, ou seja, projetos pedagógicos que se orientem por trabalhar nessa linha seriam altamente favorecidos: baixo custo, motivação, facilidade na elaboração e gerenciamento de projetos interdisciplinares, interatividade, etc. Ainda, podemos utilizar o ambiente como arena de simulações, o que favorece em muito algumas práticas baseadas em estudos observacionais. Um estudo interessante sobre o processo de aprendizado é apresentado em [Rocha - Cap. 10, 2004], os autores analisam experimentos sobre a *aquisição de conhecimento pela observação*, em particular estudam a relação *visio-motor*.

Para o caso de cursos em Ciência da Computação há o relato de outros experimentos como os elaborados por Natvig e Line, que utilizaram um jogo baseado em interface *web* para tratar conceitos fundamentais de computadores - relatam que a resposta dos alunos foi muito positiva [Natvig 2004]. Ian Parberry, que trabalha a mais de 12 anos com um cursos de graduação em programação de jogos na universidade de North Texas, em recente pesquisa constatou que a existência do curso foi um fator de escolha para 37% dos alunos da faculdade [Parberry 2005]. Esses relatos e dados acerca do uso de jogos e ambiente virtuais como ferramentas de aprendizado corroboram com as propostas e atividades executados pelo LIDET, o investimento no estudo e construção de ambientes virtuais como os aqui propostos são poderosos recursos que só vêm a enriquecer o acervo dos recursos pedagógicos hoje disponíveis.

6. Agradecimentos

Os autores agradecem à Microsoft Research pelo apoio financeiro. Nossos mais sinceros agradecimentos ao membros do LIDET: Tiago M. Jorge, Filipe C. L. Silva e Sérgio H. Nakanishi. Este texto não seria possível sem os trabalhos desenvolvidos por esses jovens e talentosos pesquisadores.

Referências

- Bandini, S., Federici, M. L., Manzoni, S. and Vizzari, G. Towards a Methodology for Situated Cellular Agent Based Crowd Simulations. Sixth IWESA World, 2005.
- Bandini, S., Manzoni, S., and Simone, C. *Modelling Heterogeneity in Multi Agent Systems*. LNCS, p.685-692, 2002.
- Blumberg, B., Isla, D. *New Challenges for Character-Based AI for Games*. AAAI, 2002.
- Boselli, L. “Gun-Tactyx Programming Games”. <http://gameprog.it/hosted/guntactyx/>
Acesso: agosto de 2005.
- Chrz, T. *Application of Intensional Logic to Knowledge Representation*. Proc. 9^o International Conf. on Computational Linguistics, Coling 1982.
- Forbus, K. D. *Qualitative Reasoning About Space and Motion*. In Gentner, D. and Stevens, A. (Eds.), *Mental Models*, LEA Associates, Inc., 1983.
- Forbus, K. D., Mahoney, J. V. and Dill, K. *How Qualitative Spatial Reasoning can Improve Strategy Game AIs*. AAAI Spring Symp. on AI and IE, 2001.

- Ghidini, C. and Giunchiglia, F. *Local Models Semantics, or Contextual Reasoning = Locality + Compatibility*. AI 127, p. 221-259, 2001.
- Jorge, T. M. and Silva, F. S. C. *A Truth Maintenance System Implementation*. IME-USP, 2005 (a). Disponível em [<http://www.linux.ime.usp.br/~tigod/monograph/>].
- Jorge, T. M. and Silva, F. S. C. *Towards Complex Reasoning Agents for Action Games*. WJOGOS - SBGAMES, 2005 (b).
- Laird, J. E., Lent, M., *et al.*, *Intelligent Agents in Computer Games*. AAAI, 1999.
- Laird, J. E., van Lent, M. *Human-Level AI's killer Application: Interactive Computer Games*. Proc. of the 70^o National Conf. on AI, pp. 1171-1178, 2000.
- Nakanishi, S. H. and Silva, F. S. C. *Game-engine as an Educational Tool in Software Engineering Courses*. WJOGOS – SBGAMES, 2005.
- Natvig L. and Line S. *Age of computers: game-based teaching of computer fundamentals*. In ITiCSE '04, pp. 107–111, 2004.
- Parberry, I. et al. *Experience with an Industry-Driven Capstone Course on Game Programming*, ACM SIGCSE Bulletin, Vol. 37, Issue 1, pp. 91-95, 2005.
- Rocha, A. F., Massad, E. and Pereira Jr., P. *The Brain: Fuzzy Arithmetic to Quantum Computing*. Springer, 2004.
- Silva, F. S. C. LIDET – *Laboratory of Interactivity and Digital Entertainment Technology* – <http://www.ime.usp.br/~lidet>, 2005 (a).
- Silva, F. C. L. and Silva, F. S. C. *Um Ambiente Virtual Baseado em Jogos para o Aprendizado de Inteligência Artificial*. WJOGOS – SBGAMES, 2005 (b).
- Sloman, A. *Architectural Requirements for Human-Like Agents Both Natural and Artificial: What sorts of machines can love?* In HCSAT, pp. 163-196, 2000.
- Streeter, S. *Ogre: Object-Oriented Graphics Rendering Engine*. <http://ogre3d.org>. (Acesso: 26/08/05).
- Vizzari, G. *Dynamic Interaction Spaces and Situated Multi-Agent Systems: From a Multi-Layered Model to a Distributed Architecture*. Ph.D. Thesis. Università degli Studi di Milano-Bicocca, 2004.
- Wadge, W. W. *Intensional Logic in Context*. World Scientific, November, 1999.
- Yamamoto, F. S. and Silva, F. S. C. *Ambientes Virtuais Contextualizáveis para Validação de Técnicas de Inteligência Artificial*. WJOGOS-SBGAMES - São Paulo, 2005.
- Yamamoto, F. S. *Sistemas Modais de Conhecimento*. Dissertação de Mestrado, IME USP, 2003.