

***BeE MODEL* – MODELO DE COMPORTAMENTO E DE
INTERAÇÃO DE ENTIDADES EM AMBIENTES DE
REALIDADE VIRTUAL E EM JOGOS**

por

Glaudiney Moreira Mendonça Júnior

Dissertação apresentada ao
Mestrado em Ciências da Computação da
Universidade Federal do Ceará

Orientadores:
Creto Augusto Vidal, Ph.D.
Joaquim Bento Cavalcante Neto, Dr.

Setembro, 2005

Índice

Agradecimentos	i
Sumário.....	ii
Summary.....	iii
Índice	iv
Lista de Anexos	vii
Lista de Figuras	viii
Lista de Tabelas	x
Capítulo 1	1
Introdução	1
1.1 Motivação	1
1.2 Objetivos.....	3
1.3 Estrutura da Dissertação	3
Capítulo 2	5
Comportamento e Interação em Ambientes Virtuais	5
2.1 Introdução.....	5
2.2 Entidades	5
2.2.1 Tipos	5
2.2.2 Características.....	11
2.3 Comportamento	15
2.4 Interação	16
2.4.1 Definições.....	16
2.4.2 Aspectos de Interação.....	22
2.5 Considerações Finais	34
Capítulo 3	36
Modelos de Comportamento e Interação.....	36
3.1 Introdução	36
3.2 CORGI.....	36
3.2.1 Descrição	36
3.2.2 Categorização	38
3.3 MASSIVE.....	39
3.3.1 Descrição	39
3.3.2 Categorização	40

3.4 SENDA.....	41
3.4.1 Descrição	41
3.4.2 Categorização	49
3.5 <i>Rich Interaction Model</i>	49
3.5.1 Descrição	49
3.5.2 Categorização	54
3.6 Considerações Finais	54
Capítulo 4	56
Modelo de Entidades Comportamentais (BeE <i>Model</i>).....	56
4.1 Introdução	56
4.2 Definição	56
4.2.1 Entidades	56
4.2.2 Comportamento	58
4.2.3 Interação	59
4.3 Cenários.....	59
4.3.1 Ação Passível de Interação.....	59
4.3.2 Entrada de Nova Entidade	60
4.3.3 Comportamento Individual.....	60
4.3.4 Entrada, Permanência e Saída em uma Aura.....	60
4.3.5 Alteração de Aura, Foco ou Nimbo.....	60
4.3.6 Perca de Consciência.....	61
4.3.7 Interação entre Três ou mais Entidades.....	61
4.4 BeE <i>Model</i>	61
4.4.1 Núcleo.....	61
4.4.2 Extensões	67
4.5 Exemplo de Utilização.....	69
4.5.1 Comportamentos e Estados	69
4.5.2 Mecanismo de Decisão.....	69
4.5.3 Auras, Focos e Nimbos.....	69
4.5.4 Consciência.....	70
4.5.5 Diagrama de Estados	70
4.5.6 Especificação dos Comportamentos.....	71
4.6 Considerações Finais	71

Capítulo 5	73
Tecnologias.....	73
5.1 Introdução.....	73
5.2 Objetos Distribuídos.....	73
5.2.1 CORBA	75
5.2.2 SOAP	75
5.3 Linguagens de Script	76
5.3.1 Características das Linguagens de <i>Script</i>	76
5.3.2 Exemplos de linguagens de <i>Script</i>	79
5.4 Máquina de Estado Finito.....	80
5.5 Considerações Finais	82
Capítulo 6	83
Caso de Uso: Conservação de Alimentos.....	83
6.1 Introdução	83
6.2 Arquitetura CRAbCVE.....	83
6.3 SINTEGRA	86
6.4 Caso de Uso	89
6.4.1 CRAbCVE.....	90
6.4.2 Entidades	91
6.5 Considerações Finais	95
Capítulo 7	96
Conclusões e Trabalhos Futuros.....	96
Trabalhos Futuros	97
Referências	99

Lista de Anexos

Anexo I	109
Identificação de entidades e interações	109
Anexo II.....	113
Entidade Abelha – Código-Fonte	113
Anexo III	114
Diagramas do BeE Model	114
Diagramas de Casos de Uso	114
Diagrama de Classes.....	114
Diagramas de Seqüência.....	114

Lista de Figuras

Figura 2.1: Exemplos de avatares. Retirados, respectivamente, dos jogos Ragnarok Online, Splinter Cell e Need for Speed Underground 2.	8
Figura 2.2: Exemplos de simulóides. Respectivamente, Marilyn (MIRALab), STEVE (ISI) e BIA (CRAb).	9
Figura 2.3: Exemplos de objetos. Respectivamente, Armas (Doom 3), Móveis (The Sims) e Espadas (Neverwinter Nights).	10
Figura 2.4: Exemplos de artefatos. Respectivamente, escuridão (Splinter Cell), água (Final Fantasy X), fumaça (Wolfeinstein: Enemy Territory) e neblina (Doom 3). ...	11
Figura 2.5: Exemplo de subauras. Espaços baseados no grau de intimidade.	12
Figura 2.6: Exemplo de aura, foco, nimbo e consciência em um meio.	14
Figura 2.7: Componentes da interação homem-computador.	20
Figura 3.1: Funcionamento do gerente de aura.	40
Figura 3.2. Padrão de comportamento perceptível reativo.	44
Figura 3.3. Padrão de comportamento perceptível pró-ativo.	45
Figura 3.4. Padrão de comportamento perceptível passivo.	45
Figura 3.5. Padrão de comportamento perceptível pró-ativo e reativo.	46
Figura 3.6. Padrão de comportamento imperceptível reativo.	47
Figura 3.7. Padrão de comportamento imperceptível pró-ativo.	48
Figura 3.8. Padrão de comportamento imperceptível pró-ativo e reativo.	48
Figura 3.9: Hierarquia de Interação e um exemplo de sua utilização.	54
Figura 4.1: Diagrama de Classes do núcleo do modelo.	62
Figura 4.2: Diagrama de Classes das extensões do modelo.	67
Figura 4.3: Comportamentos e Estados da entidade abelha.	69
Figura 4.4: Diagrama de estados da entidade abelha.	70
Figura 5.1: Trading.	74
Figura 5.2: Representação gráfica de uma máquina de estado finito.	81
Figura 6.1: Componentes da arquitetura CRAbCVE.	84
Figura 6.2: Página Web do curso Boas Práticas na Manipulação de Alimentos.	87
Figura 6.3. Ambiente SOLAR.	88
Figura 6.4: Cliente de Interação e Edição em Realidade Virtual do CRAbCVE.	90
Figura 6.5: Gerenciador de Eventos do CRAbCVE.	91
Figura 6.6: Representação do usuário no caso de uso.	92

Figura 6.7: Comportamentos e estados da entidade alimento.	93
Figura 6.8: Diagrama de estados da entidade alimento.	93
Figura 6.9: Representação gráfica dos simulóides utilizados no caso de uso.	95

Lista de Tabelas

Tabela 2.1: Categorias de comunicação não-verbal.	23
Tabela 3.1. Modelos de formas de interação e suas categorias	50
Tabela 4.1. Condições para a realização de cada transição da entidade abelha	71
Tabela 6.1. Condições para a realização de cada transição da entidade alimento.....	94
Tabela 6.2. Informações específicas dos simulóide do caso de uso.	94

Capítulo 1

Introdução

1.1 Motivação

Segundo Çapin et al. (1999), Realidade Virtual (RV) é uma tecnologia capaz de transportar um indivíduo para um ambiente sintético, construído computacionalmente, sem movê-lo fisicamente, enquanto manipula informações destinadas aos seus órgãos sensoriais, com o objetivo de fazê-lo perceber o ambiente virtual sintetizado como se fosse um ambiente real.

O termo “realidade virtual” significa, de uma forma bastante genérica, a união de diversas tecnologias, oriundas das ciências da computação e da eletrônica, capazes de oferecer a um ser humano um alto grau de interação com a máquina e, possivelmente, com outros indivíduos. Segundo Hamit (1993), o termo “realidade virtual” foi criado no *Massachusetts Institute of Technology* – MIT, no final da década de 70, para expressar a idéia de presença humana em um espaço gerado por computador. Ela foi transferida para o *Atari Labs*, onde muitos ex-alunos do MIT foram trabalhar no início dos anos 80. Assim, o termo foi comumente utilizado na indústria de jogos de computador para descrever os cenários de vários *videogames*. Mais tarde, foi usado por Jaron Lanier para descrever a imersão num mundo virtual pelo uso de um HMD (*Head-Mounted Display*) e uma *dataglove*. Nos anos 90, o emprego da realidade virtual como uma forma de interação com o computador, tornou-se bastante popular devido à sua utilização em jogos. Isso foi possível, graças à rápida evolução dos processadores empregados em microcomputadores comuns.

Mais tarde, com o avanço da internet e dos sistemas de telecomunicações, esse tipo de jogo passou a oferecer a possibilidade de vários participantes interagirem entre si simultaneamente, criando assim a noção de ambientes virtuais em rede (NVE – *Networked Virtual Environments*).

O desenvolvimento desses aplicativos acabou permitindo a geração de toda uma nova gama de aplicações voltadas para entretenimento e que foram facilmente estendidas para o uso em: educação e cultura (Vidal et al., 2003; Kaufmann et al., 2000; Cardoso et al., 2001; Kirner et al., 2001; Wazlawick et al., 2001), treinamento corporativo (Netto & Oliveira, 2001), arquitetura (Silva et al., 2003), ciências da saúde (Drummond et al., 2001) e muitos outros fins (Musse et al., 2000).

A arquitetura CRAbCVE (Gomes, 2005) é uma arquitetura desenvolvida na Universidade Federal do Ceará para disponibilizar ambientes virtuais em rede a um baixo custo (utilizando computadores comuns) através da internet. É uma evolução da arquitetura Ataxia (Leite-Júnior, 2000) incluindo elementos para permitir o trabalho colaborativo.

De maneira geral, em todos os casos de aplicação da realidade virtual (multiusuária ou não), sua utilização possui uma finalidade principal (entretenimento, ensino, entre outras) e reúne usuários que interagem entre si (quando NVEs) e com o mundo virtual, para realizarem tarefas. Estes usuários são representados no mundo virtual por entidades que possuem características próprias (por exemplo, aparência e gestos) e realizam ações (por exemplo, andar e mover objetos). Os ambientes de realidade virtual são povoados por entidades que agem e interagem entre si e com o ambiente, modificando-se e alterando o ambiente (ou parte dele) e outras entidades.

Além dos usuários, existem outras entidades nesses ambientes que são controladas pelo próprio sistema. Elas possuem funções pré-definidas, podendo interagir com os usuários conectados ao ambiente (por exemplo, objetos manipuláveis), ou agir segundo uma configuração definida (por exemplo, personagens controlados por computador). A interação entre as entidades e os demais comportamentos que elas possuem, possibilitam que as pessoas inseridas nesses ambientes possam realizar seus objetivos de maneira mais rica e eficaz.

Comportamento expressa o conjunto de ações que podem ser realizadas por uma entidade de um ambiente virtual, em resposta a um estímulo interno ou externo, podendo alterar o seu estado. Essas ações podem envolver outras entidades, o que a caracteriza como uma interação; ou envolvem apenas a própria entidade e, assim, expressa desejos e intenções inerentes a ela.

Para possibilitar a realização de tarefas em um ambiente de realidade virtual, é necessário disponibilizar ao usuário ferramentas com as quais ele possa alcançar seus objetivos. Leontjev (1978) define a execução de tarefas em três níveis: atividade, ação e operação. Atividade corresponde à motivação necessária para se executar algo e pode ser dividida numa sequência de ações, cada uma tendo seus próprios objetivos. As operações, por sua vez, são consideradas processos automáticos ou subconscientes correspondentes a condições correntes e a ajuste das ações. Essas operações compõem as formas de interação.

É muito importante disponibilizar formas de interação ricas em aplicações de realidade virtual (multiusuárias ou não), bem como agregar mecanismos de tomada de decisão e de reação a estímulos às entidades existentes em um ambiente virtual.

1.2 Objetivos

Visando proporcionar comportamentos ricos às entidades existentes em um ambiente de realidade virtual, esta dissertação propõe um modelo, chamado *BeE Model*, para modelagem de comportamentos de entidades em ambientes de realidade virtual e em jogos.

Para atingir tal objetivo, as seguintes etapas foram realizadas neste trabalho:

- ◆ Classificação das entidades existentes em um ambiente virtual;
- ◆ Caracterização de comportamento e interação;
- ◆ Identificação e avaliação dos modelos existentes na literatura e proposição de um modelo de comportamento abrangente e flexível (*BeE Model*);
- ◆ Identificação de tecnologias que podem ser utilizadas no desenvolvimento do modelo proposto; e
- ◆ Desenvolvimento de um caso de uso com a utilização do modelo proposto em uma aplicação que utiliza um sistema de realidade virtual, baseado na arquitetura CRAbCVE.

O modelo não se propõe a inserir elementos de Inteligência Artificial em um ambiente de realidade virtual, e sim, proporcionar uma plataforma que possa agregar uma inteligência desse tipo.

1.3 Estrutura da Dissertação

No Capítulo 2, são apresentados conceitos sobre entidades, comportamento e interação, necessários ao entendimento dos tópicos abordados nos capítulos subsequentes. No Capítulo 3, quatro modelos de comportamento são apresentados e, para cada um deles, é feita uma caracterização de como os modelos trabalham com os conceitos abordados no Capítulo 2. No Capítulo 4, com base nos conceitos e nos modelos apresentados nos Capítulos 2 e 3, é definido o modelo de entidades comportamentais, *BeE Model*. Ainda no Capítulo 4, são apresentados a estrutura do modelo, alguns cenários e um exemplo de utilização. No Capítulo 5, tecnologias que podem ser utilizadas no desenvolvimento do modelo proposto são descritas. No Capítulo 6, um caso de uso é apresentado incluindo o sistema de realidade virtual ao qual o modelo será integrado e o contexto da

aplicação realizada. Por fim, no Capítulo 7, as conclusões desta dissertação são apresentadas, bem como algumas sugestões de trabalhos futuros.

Capítulo 2

Comportamento e Interação em Ambientes Virtuais

2.1 Introdução

Neste capítulo serão abordados alguns conceitos importantes no universo de comportamento e interação em ambientes virtuais. Como o foco deste trabalho é um modelo que suporte comportamento e interações entre entidades de ambientes virtuais, algumas questões são relevantes: Quais entidades podem existir nesses ambientes? E o que significa comportamento e interação?

A definição desses termos deve ser feita de maneira apropriada antes de se apresentar e avaliar modelos de interação e comportamento. Assim, as próximas Seções descrevem quais os tipos de entidades que podem aparecer em ambientes virtuais e suas características, bem como apresentam definições de comportamento e interação encontradas em diferentes áreas de pesquisa, além de suas peculiaridades e importância nos ambientes virtuais.

2.2 Entidades

Ambientes de realidade virtual são povoados por entidades que agem e interagem entre si e com o ambiente, modificando-se e alterando o ambiente (ou parte dele) e outras entidades. Inúmeros exemplos de entidades podem ser encontrados nos mais variados ambientes virtuais como, pessoas, animais, objetos, seres místicos, espíritos, água, neblina, entre outros.

2.2.1 Tipos

As entidades podem ser classificadas em cinco tipos:

- ◆ **Avatares:** representações dos usuários no sistema de ambiente de realidade virtual (considerado ambiente virtual). O tipo de representação depende do ambiente que se deseja criar, como por exemplo: pessoas em ambientes similares ao real, animais em ambiente que representam um habitat selvagem, seres sem corpo físico quando a representação não for importante ou atrapalhar o objetivo do ambiente, entre outras;

- ◆ **Simulóides:** entidades controladas pelo computador. Por exemplo, bibliotecária virtual, entidades que apenas povoem um ambiente, máquinas complexas, simuladores, entre outras;
- ◆ **Mestres:** entidades que possuem permissão de alterar o estado interno de outras entidades. Como, por exemplo, um superusuário, uma entidade mantenedora do ambiente virtual, entre outras;
- ◆ **Objetos:** elementos que podem ser manipulados por outras entidades e que fazem parte do ambiente. Por exemplo, porta, copo, bola, entre outras; e
- ◆ **Artefatos:** atributos do ambiente que podem interagir com outras entidades, como por exemplo, neblina, água, luminosidade, entre outros. O ambiente em si não interage com outras entidades, pois ele é apenas o lugar onde as entidades estão inseridas.

A divisão das entidades em tipos tem o intuito de apresentar as principais características dessas entidades existentes em um ambiente virtual e auxiliar o desenvolvedor na elaboração destes ambientes. Assim, alguns exemplos de entidades podem se encaixar em diferentes categorias, tendo o desenvolvedor, total liberdade de escolher o tipo de entidade a qual o exemplo específico irá pertencer, ou considerá-lo como pertencente a mais de um tipo, pois eles não são excludentes.

A seguir é apresentada uma discussão mais detalhada de cada tipo de entidade, suas características e sua importância.

Avatares

Segundo o Houaiss (2001), Avatar é, na crença hinduísta, a descida de um ser divino a terra de forma materializada. Em realidade virtual, avatar é um ícone gráfico que representa um participante do sistema de realidade virtual (Webopédia, 2004). Ele pode ser desde um nome escrito na tela como nos jogos estilo MUD (*Multi-User Dungeons*), a uma estrutura tridimensional articulada e deformável como no jogo *Splinter Cell* (Ubisoft, 2005). Seu uso aprimora o sentido de presença do participante (Slater & Usoh, 1994), localizando-o no espaço virtual, e informando aos demais participantes do sistema sua posição, quando este for um ambiente virtual em rede (NVE – *Networked Virtual Environment*). Utiliza-se o avatar também para possibilitar a interação do participante com o mundo virtual e a comunicação deste com outros participantes através de gestos e expressões.

Em NVE's, o seu emprego ganha ainda mais importância, uma vez que estes ambientes devem representar, de maneira autêntica e coerente, os diversos participantes e permitir a presença compartilhada dos mesmos. Os avatares desses ambientes apresentam ainda mais funções do que aquelas encontradas em ambientes virtuais monousuários (Çapın, et al., 1999), tais como:

- ◆ Percepção (saber se alguém está próximo): o emprego de avatares para a representação dos participantes de um NVE torna possível a qualquer indivíduo reconhecer se há mais algum participante no ambiente. Nesse caso, quanto mais realista for o avatar, mais simples a tarefa de distinguir um determinado participante de um objeto qualquer.
- ◆ Localização (saber onde outro participante está): o uso de um avatar torna disponível, de uma forma bastante simples, as informações sobre posicionamento e orientação de participantes no interior do NVE.
- ◆ Identificação (reconhecer um participante): sempre que um participante possuir certa liberdade de escolha de seu respectivo avatar no NVE, a tarefa de diferenciação de indivíduos torna-se um processo simples de visualização e reconhecimento por parte dos próprios usuários.
- ◆ Visualização do foco de interesse de outros participantes (saber onde a atenção de um participante está direcionada): a orientação de avatares no interior do mundo virtual compartilhado pode definir qual o foco de interesse no qual se concentra o usuário. Isso é crucial para a manutenção de interações entre os participantes. Em situações específicas de comunicação não-verbal, o direcionamento do avatar pode, inclusive, servir como um modo bastante eficiente de se especificar um determinado local no mundo virtual.
- ◆ Visualização das ações de outros participantes (saber o que outro participante está fazendo e o que quer dizer com seus gestos): processos de manipulação de objetos no interior do NVE podem ser percebidos através da simples observação do comportamento de um determinado avatar. Isso é particularmente interessante para a execução de tarefas de forma colaborativa. Do mesmo modo, é possível, inclusive, a transmissão de emoções entre participantes, através da visualização de feições e movimentos.

- ◆ Representação social do indivíduo através de adornos (saber o estado e a função de outro participante): o acoplamento de adornos (roupas, acessórios, entre outros) permite que os participantes expressem reações ou sentimentos de uma forma mais completa. Vale salientar que nem sempre tais recursos encontram-se disponíveis no NVE, uma vez que opções de seleção de diferentes vestimentas e o uso de objetos de decoração e de trabalho aumentam bastante a complexidade da implementação desses ambientes.

Em NVEs que almejam aproximar-se do mundo real, o uso de avatares realistas, que representam o visual e o comportamento do ser humano, acarreta um aumento na complexidade e no número de possibilidades de interação entre os participantes, aumentando assim a quantidade de informações que transitam na rede. Muitas vezes, devido à necessidade de se diminuir esse tráfego, diminui-se a complexidade dos avatares e, com ela, as interações entre os participantes. Exemplos de avatares são mostrados na Figura 2.1.



Figura 2.1: Exemplos de avatares. Retirados, respectivamente, dos jogos Ragnarok Online, Splinter Cell e Need for Speed Underground 2.

Simulóides

Csordas (2000) define simulóides como entidades gerenciadas por *software* que não possuem uma contraparte consciente no mundo real. Ou seja, são entidades controladas por computador, que não precisam de intervenção humana para realizar suas funções.

Simulóides podem ser animais, máquinas automáticas, monstros em ambientes virtuais, ou também ter o aspecto de um avatar, diferindo apenas quanto à característica de serem controlados por uma máquina. Eles podem ser elementos simples inseridos no ambiente com o propósito de povoá-lo, como por exemplo, animais povoando uma fazenda virtual, e transeuntes de uma cidade virtual. Podem também ser entidades com comportamento complexo e próximo ao comportamento humano, como por exemplo, o simulóide Marilyn, desenvolvido por Nadia Thalman e considerado por Csordas

(2000) como um dos mais avançados. Ele simula a atriz Marilyn Monroe, incluindo características como as expressões faciais, a fala, as roupas, o cabelo, e a habilidade de responder ao usuário do computador. Outro exemplo de simulóide é o STEVE - *Soar Training Expert for Virtual Environments* (Rickel & Johnson, 1999), um agente pedagógico com o objeto de ensinar tarefas procedimentais como a operação e o reparo de máquinas complexas. Os simulóides podem também disponibilizar ferramentas externas ao ambiente virtual, como, por exemplo, o simulóide BIA, uma bibliotecária virtual que disponibiliza o serviço de tradução para os usuários de um ambiente (Santos, 2001). Na Figura 2.2 são apresentados alguns exemplos de simulóides.



Figura 2.2: Exemplos de simulóides. Respectivamente, Marilyn (MIRALab), STEVE (ISI) e BIA (CRAB).

Mestres

O termo mestre veio dos jogos de interpretação (RPG – *Role Playing Game*). No RPG, o mestre é um jogador diferente dos outros por não possuir um personagem específico, pois representa todo o mundo, exceto os personagens dos jogadores, incluindo outros personagens que por ventura sejam necessários (PNJ – Personagem Não Jogador), como vilões, pessoas comuns, monstros, entre outros. É ele que define a trama e o ambiente que os personagens serão inseridos. Sua principal tarefa é criar uma boa trama.

Mestre, neste trabalho, é uma entidade que pode alterar as outras entidades, incluindo seu estado, comportamento e propriedades. O mestre pode impedir uma entidade de realizar alguma ação e também alterar essa ação. Um exemplo de entidade mestre é um superusuário que gerencia o ambiente. É importante observar que qualquer entidade pode ser um mestre. Por exemplo, um simulóide pode ser também uma entidade mestre.

Objetos

São entidades inseridas no ambiente virtual com o propósito de preencher o espaço. Eles são divididos em estáticos e dinâmicos (Vidal et al., 2004). Os objetos estáticos não possuem comportamentos e são simples elementos integrantes do cenário. Já os objetos dinâmicos possuem comportamento vinculado, ou seja, eles podem reagir às

ações de outras entidades, como ser manipulado, além de possuir um estado interno. Este trabalho está interessado apenas nos objetos dinâmicos e, por isso, todos os objetos citados serão deste tipo.

Quando uma entidade possuir um comportamento muito complexo e autônomo, pode ser classificado como um simulóide, mesmo que se pareça com um objeto. Por exemplo, uma máquina automática, com um funcionamento bastante complexo e que exija cálculos matemáticos, pode ser visto como um simulóide e não um objeto.

Objetos não precisam possuir uma representação visível no mundo virtual, pois não necessariamente precisam ser vistos por outras entidades. Outros podem até ser visíveis, mas podem não possuir existência física, ou seja, entidades não entram em contato com eles. Pode-se representar uma informação como objeto, a fim de se criar mecanismos de interação com essa informação. Na Figura 2.3 são mostrados alguns exemplos de objetos.



Figura 2.3: Exemplos de objetos. Respectivamente, Armas (Doom 3), Móveis (The Sims) e Espadas (Neverwinter Nights).

Artefatos

Manninen (2004) define o conceito de artefato com uma entidade que representa um atributo do ambiente, tal como, escuridão, água, gás, neblina, entre outros. Ele interage com outras entidades e podem alterar a maneira delas agirem. Por exemplo, se o ambiente possuir um artefato água, as entidades devem se locomover através desse artefato utilizando alguma forma de nado, ou serão levadas pela correnteza. Na Figura 2.4 alguns exemplos de artefatos são apresentados. No artefato escuridão, só é possível observar nitidamente as entidades da cena devido a um recurso do jogo de visão no escuro. No artefato água, a entidade está interagindo com uma bola numa espécie de jogo aquático de futebol.



Figura 2.4: Exemplos de artefatos. Respectivamente, escuridão (*Splinter Cell*), água (*Final Fantasy X*), fumaça (*Wolfenstein: Enemy Territory*) e neblina (*Doom 3*).

2.2.2 Características

As entidades, embora de tipos diferentes, possuem características comuns que necessitam de uma explicação sobre seu uso. Elas serão apresentadas nesta seção.

Aura

Aura é utilizada para se determinar quais entidades podem interagir, em um dado momento. Fahlen et al. (1993) define aura como um subespaço que representa a existência de um objeto em um determinado meio e que atua viabilizando uma possível interação. Greenhalgh & Benford (1995) explicam que aura representa a área de sensibilidade do objeto à interação. Cada entidade possui uma aura para cada meio que deseje possibilitar interação, como o meio visual, sonoro, entre outros. Isso pode permitir, por exemplo, que uma entidade seja capaz de ouvir outra, antes de vê-la.

As entidades podem carregar suas auras consigo quando se movimentam (como no caso de um avatar que carrega sua aura visual, possibilitando que outras entidades possam vê-lo), ou posicionar sua aura em uma região específica, dependendo do seu propósito (como um robô que possui uma aura de alerta em determinada região e, caso alguém entre na área, perseguirá o intruso). Uma aura pode ter qualquer forma e tamanho, e não precisa estar em redor da entidade relacionada, como também não necessita ser contínua no espaço.

Uma aura pode possuir subauras, representando aspectos mais especializados daquela aura. Por exemplo, a aura de proximidade de um avatar pode ser dividida, segundo a classificação de distância conectada com o grau de intimidade dos indivíduos, apresentada por Çapin et al. (1999), em quatro espaços: público, social, pessoal e íntimo. Esses quatro espaços e suas posições são apresentados na Figura 2.5

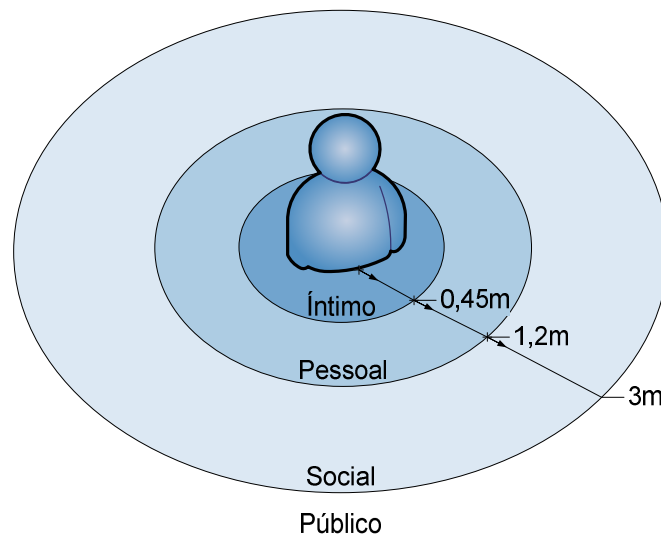


Figura 2.5: Exemplo de subauras. Espaços baseados no grau de intimidade.

Quando duas auras colidem, a interação entre as entidades naquele meio se torna possível, não significando que sempre ocorrerá. A decisão de existência de interação é feita pelo foco e pelo nimbo apresentados a seguir.

Foco e Nimbo

Uma vez que duas auras colidem, as duas entidades envolvidas na interação devem analisar se realmente uma interação deve acontecer. Cada entidade especifica uma região de atenção e uma região de presença para aquele meio onde houve colisão de aura.

A região de atenção, chamada de **foco**, representa o local no espaço onde a atenção da entidade está voltada. Já a região de presença, chamada de **nimbo**, define um subespaço no qual a entidade torna algum aspecto seu disponível às outras entidades. Nimbo pode significar presença, identidade, atividade, ou uma combinação destes, e ele permite que as entidades possam influenciar outras, ou seja, projetarem-se, ou suas atividades, para tentar serem ouvidas, vistas, entre outros (Çapin et al., 1999).

Foco e Nimbo são representados da mesma forma que aura, e as entidades envolvidas em uma colisão de aura devem analisar se também existe uma colisão de foco e nimbo para que, assim, possa existir uma interação. Aura, foco e nimbo podem ser funções multivaloradas que mapeiam atributos geralmente espaciais das entidades (posição, orientação, entre outros). Além disso, possuem forma e extensão arbitrárias, e são específicos do meio.

Segundo Greenhalgh & Benford (1995), aura, foco e nimbo podem ser manipulados de três formas diferentes: implícita, explícita e via entidades adaptadoras.

A manipulação **implícita** dá-se através de ações que causam modificações involuntárias desses elementos. Por exemplo, ao colocar um fone de ouvido, a aura auditiva do avatar é modificada. Na manipulação **explícita**, a entidade escolhe alterar seu foco, nimbo ou aura. Assim, um usuário utiliza a manipulação explícita caso deseje alterar a forma ou a extensão do foco de seu avatar. **Entidades adaptadoras** alteram a aura, o foco ou o nimbo de uma entidade que esteja interagindo com eles. Um palanque para discursos, por exemplo, pode aumentar a aura e o nimbo de áudio das entidades que subirem nele, aumentando assim a potência da sua voz; uma mesa para reuniões pode expandir a aura, o foco e o nimbo daqueles que se sentam em volta dela, de modo a alcançar todos os presentes na mesa.

Consciência

A colisão entre foco e nimbo possibilita, para a entidade que possui o foco, uma consciência da entidade que possui o nimbo. “A consciência de um objeto a respeito de outro quantifica a importância subjetiva ou relevância do outro objeto em um dado meio” (Greenhalgh & Benford, 1995). Ela pode, por exemplo, corresponder ao volume de um canal de áudio entre dois usuários, ou ao nível de detalhe de um gráfico. Portanto, mais atenção será dada às entidades que possuírem um maior valor de consciência.

A medida de consciência entre duas entidades não é necessariamente simétrica, ou seja, a consciência de uma entidade A sobre uma entidade B não é necessariamente igual à consciência de B sobre A.

Na interação, ambas as entidades desejam afetar seus níveis de consciência mútuos – a entidade observadora deseja focar sua atenção em regiões particulares ou entidades particulares (foco), enquanto a entidade observada deseja controlar sua visibilidade ou “acessibilidade” de modo que certas entidades estejam mais cientes dele que outras (nimbo).

Assim como a aura, o nível de consciência é específico para cada meio, e depende sempre do foco e do nimbo. Quanto mais uma entidade A encontra-se dentro do foco de uma entidade B, mais a entidade B torna-se consciente de A; e quanto mais A encontra-se dentro do nimbo de B, mais A torna-se consciente de B. Na Figura 2.6 são ilustrados a aura, o foco, o nimbo e a consciência de entidades simples para apenas um meio. Foram omitidos o foco do elemento observado e o nimbo do observador.

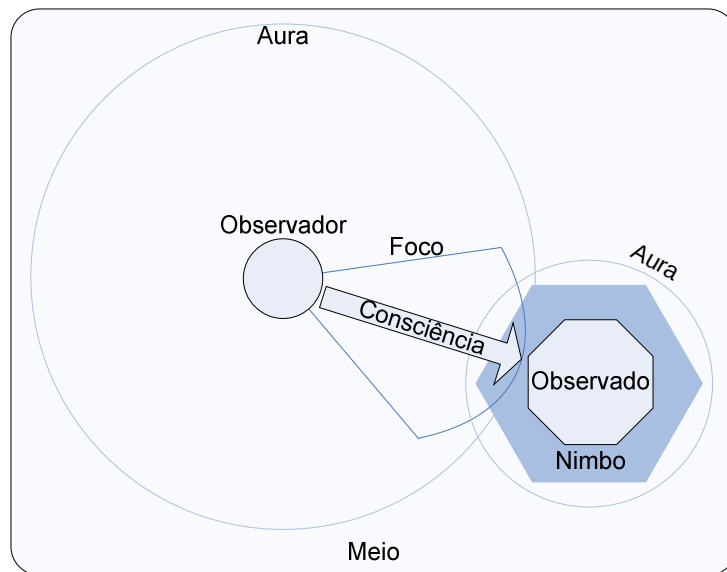


Figura 2.6: Exemplo de aura, foco, nimbo e consciência em um meio.

Estado

Campos et al. (2003) define três características principais em objetos de ambientes virtuais: existência, atividade e visibilidade. Essas características serão estendidas a todas as entidades. Essas características definem o estado da entidade num determinado momento. A combinação dessas características gera oito tipos diferentes de estados, que podem ser utilizados por qualquer tipo de entidade:

- ◆ **Ator:** existe, está ativo e é visível. Exemplo: um carro em movimento.
- ◆ **Cenário:** existe, não está ativo e é visível. Exemplo: um carro parado em um posto de gasolina. Embora o estado da entidade seja chamado de “cenário”, ele não é uma entidade estática, pois está apenas temporariamente sem realizar uma ação.
- ◆ **Espião:** existe, está ativo e não é visível. Exemplo: entidades em uma cidade virtual que não interessa ao usuário e que este selecionou para que não fossem visíveis a ele.
- ◆ **Camuflagem:** existe, não está ativo e não é visível. Mesmo exemplo de espião, porém não está realizando uma ação.
- ◆ **Fantasma:** não existe, está ativo e é visível. O fato de uma entidade fantasma não existir, não significa que não possa ser visto. A existência exprime uma condição “física” e por isso, outras entidades não poderão estabelecer contato “físico” com eles, embora possa vê-los. O mesmo vale para todos os tipos que não existem. Por exemplo, num sítio arqueológico,

ser possível visualizar a população da época em que as ruínas estavam intactas.

- ◆ **Miragem:** não existe, não está ativo e é visível. Por exemplo, num sítio arqueológico, ser possível visualizar como eram as ruínas, antes de serem destruídas.
- ◆ **Fábula:** não existe, está ativo e não é visível. Por exemplo, num sítio arqueológico, o usuário não quer mais ver a população das ruínas.
- ◆ **Mito:** não existe, não está ativo e não é visível. Por exemplo, num sítio arqueológico, o usuário não quer mais ver as construções originais.

Além dos estados principais apresentados, uma entidade possui estados inerentes ao seu comportamento. Esses estados adicionais herdam características dos estados principais, além de possuírem características próprias. Não existe limite para a quantidade de estados que uma entidade pode possuir, mas, quanto mais estados, mais complexo é o seu comportamento, o que torna difícil o seu desenvolvimento. Os estados podem ser compostos de subestados, elaborando assim uma hierarquia.

2.3 Comportamento

Segundo Houaiss (2001), comportamento é:

- ◆ Uma reação peculiar de uma coisa em determinadas circunstâncias;
- ◆ Tudo que um organismo, ou parte dele, faz que envolva ação e resposta à estimulação;
- ◆ Procedimento de alguém perante estímulos ou sentimentos e necessidades, ou uma combinação de ambos.

A enciclopédia virtual Wikipédia (2004) define comportamento como qualquer coisa que um organismo faça. A palavra geralmente é empregada como um substituto para “resposta”, mas, no uso coloquial, é um termo coletivo. Em Vidal et al. (2004), comportamento é um conjunto de ações que um objeto dinâmico (uma entidade) executa quando exposto a um determinado estímulo, podendo ou não mudar o seu estado.

Klüsener et al. (2002) cita alguns autores e suas considerações sobre comportamento no âmbito da comunicação social. Comportamento social é definido por Skinner (1981) como o comportamento de pessoas em relação a uma outra, em um ambiente comum. Assim, o comportamento social surge quando um organismo é importante para outro, como parte do seu ambiente. O comportamento de um organismo

gera estímulos que afetam o comportamento de outros organismos. Botomé (2001) desmembra o comportamento em três momentos: situação anterior, ação humana, e consequência ou situação decorrente. Assim, uma entidade encontra-se em uma situação ou estado; recebe um estímulo proveniente de outras entidades, do ambiente ou de si mesmo; decide se deve reagir em resposta àquele estímulo e, caso reaja, quais ações deve tomar e se essas ações mudarão o seu estado interno.

Baseado no trabalho de Sánchez-Segura et al. (2004), o comportamento pode ser dividido em: Pró-ativo, Reativo e Pró-ativo e reativo. Um comportamento pró-ativo gera estímulos que podem acionar outros comportamentos. Por exemplo, um despertador que toca para acordar outra entidade. Um comportamento reativo responde a um estímulo. Por exemplo, uma porta que se abre à aproximação de um avatar. E um comportamento pró-ativo e reativo pode tanto gerar estímulos, como ser estimulado.

No contexto desta dissertação, comportamento expressa o conjunto de ações que podem ser realizadas por uma entidade de um ambiente virtual, em resposta a um estímulo interno ou externo, podendo alterar o seu estado. Essas ações podem envolver outras entidades, caracterizando-se numa interação, que será definida com mais detalhes na próxima seção. Porém, algumas ações envolvem apenas a própria entidade e expressam desejos e intenções inerentes a ela.

2.4 Interação

Segundo Manninen (2004), a definição de interação difere de acordo com o domínio de pesquisa. Ciências comportamentais e educacionais usam a palavra de maneira diferente das áreas de engenharia e processamento de informação. Nesta seção serão apresentadas definições relacionadas com interação, encontradas na literatura e apresentadas por Manninen em seu trabalho. Após essas definições, serão abordados alguns aspectos de interação que apresentam características peculiares desse tipo de comportamento.

2.4.1 Definições

O dicionário Collins (Sinclair, 1995) identifica três categorias de interação:

- ◆ Quando pessoas interagem com outras, elas se comunicam à medida que trabalham ou passam tempo juntas;
- ◆ Quando pessoas interagem com máquinas, informações ou instruções são trocadas; e

- ◆ Quando uma coisa interage com outra, as duas coisas afetam condições ou comportamento delas.

Interação, portanto, é o processo ou situação onde duas ou mais entidades têm contato com outras. A palavra interação é relacionada principalmente com comunicação. Wiio (1996) divide a comunicação em três dimensões: social, factual e temporal. Na dimensão social, a comunicação é um processo de interação; na dimensão factual, é um processo de manipulação de fatos; e na dimensão temporal, é um processo de modificação estrutural.

No mundo real, quando se interage com outra pessoa, geralmente existe a intenção de comunicar algo e, mesmo que a intenção não exista, o processo de interação pode gerar informações que podem ser percebidas pela contraparte da interação, ou por outros que estejam observando. Os aspectos de comunicação da interação serão discutidos mais adiante neste Capítulo. Por hora, é suficiente saber que as teorias de comunicação existentes descrevem a natureza da comunicação e o papel da interação nesse processo. Assim, uma definição clara de interação é ambígua, pois, existem diferentes definições para comunicação (Wiio, 1996) e como interação está estreitamente relacionada com comunicação, também possui várias definições. Para reduzir esta ambigüidade no foco, escopo e limite do termo, a seguir serão apresentadas definições que contribuem na direção do conceito de interação.

Interatividade

É o grau pelo qual o usuário sente-se convencido do efeito mútuo que ele e o ambiente têm um sobre o outro. Melhor interatividade gera uma interação mais agradável e melhor controlada com o ambiente virtual (Nunamaker, 1997). Uma definição alternativa é dada por Robinett (1998) que descreve interatividade em jogos de ramificação¹:

“Ser **interativo** significa que, quando você faz algo, altera o que acontece a seguir. Uma experiência interativa imita a natureza responsiva [que reage positivamente] da experiência do mundo real – quer dizer, suas ações produzem resultados. Você pode alterar o mundo (uma pequena parte dele, pelo menos) estendendo a mão e manipulando alguma coisa”.

(Robinett, 1998, p. 332)

¹ Nesse tipo de jogo o usuário experimenta segmentos de história lineares e curtos que disponibilizam um pequeno número de escolhas, cada uma direcionando a um novo segmento linear com novas escolhas e assim por diante. O caminho a percorrer pode convergir, divergir ou retornar.

Mecanismos de interação

São características que devem ser suportadas em um ambiente virtual. No domínio dos jogos de computador, podem ser definidos também como **mecânicas de jogo**:

“Mecânicas de jogo consistem na natureza do ambiente de jogo e como o jogador pode interagir com ele. Correndo, dirigindo, voando, atirando, pegando chaves, e adicionando ingredientes em uma sopa são exemplos de mecânicas possíveis. Dominar as mecânicas de jogo pode ser uma grande fonte de prazer”.

(Clanton, 2000)

Os mecanismos de interação geralmente são livres do contexto onde serão utilizados, ou seja, são desenvolvidos para dar suporte às convenções sociais do ambiente (Bullock et al., 1993). O modo como o usuário movimenta-se e quais as ações que pode executar são exemplos de mecanismos de interação.

Interação Virtual

Interação, no contexto de ambientes virtuais colaborativos (*Collaborative Virtual Environment* – CVE), tem muitas similaridades com a interação do mundo físico. Robinett (1998) classifica ações importantes de interação humana com as coisas e pessoas no mundo:

- ◆ Observar (uma pessoa pode olhar sua vizinhança);
- ◆ Mover-se (uma pessoa move-se através do mundo e o vê de diferentes pontos de vista);
- ◆ Agir (uma pessoa pode executar ações que modificam o mundo); e
- ◆ Conversar (uma pessoa pode comunicar-se verbalmente com outra).

Essa classificação altamente abstrata confere, além dos aspectos comunicativos da interação, sua natureza espacial (movendo-se no espaço, reorientando a direção e modificando o ambiente). A importância da interação espacial é introduzida por Wann & Mon-Williams (1996) em seu trabalho. Eles argumentam que a interação entre um usuário e o ambiente envolve o uso de informações que refletem modificações espaciais e temporais do ambiente relativo.

É importante que o observador possa: determinar para onde ele está se dirigindo quando se mover através do mundo; estimar como o contato com um objeto pode ser feito ou evitado; modificar seu ponto de vista, ou seja, passear pela estrutura tridimensional e se posicionar em áreas de interesse; discernir detalhes visuais

relevantes na vizinhança; e ativar e manipular uma estrutura com algum dispositivo de entrada.

A natureza virtual de CVEs faz o conceito de interação difícil. O mundo virtual é o espaço onde as ações e interações acontecem. Contudo, o processo total de interação inclui entidades que povoam esse espaço. Para definir uma contraparte relevante à interação do mundo real, a comunidade de CVE tem definido conceitos como interação virtual.

Interação virtual (Jensen, 2001b) se refere a interação especificamente dentro de um *framework* de mundos virtuais. Esta interação tem sua base no conceito social, estendendo a situação social para cobrir pessoas que estão geograficamente distribuídas, sem proximidade física imediata. A interação em si é mediada, e somente a representação das pessoas, ou os avatares, participam face a face na comunicação virtual, podendo utilizar expressões faciais, gestos e linguagem corporal de uma forma mediada e virtual. Então, em CVEs, comunicação interpessoal e o conhecimento que representa tornam-se relevantes para o mundo de computador.

Técnicas de Interação

São utilizadas para mapear a entrada do usuário, capturada através de dispositivos, em ações e comandos de controle correspondentes. Essa entrada pode ser a posição das partes do corpo, comandos de voz, gestos das mãos, movimentos de um *mouse*, entre outros. O sistema de ambiente virtual responde a entrada modificando o estado do ambiente, isto é, modificando a forma, posição, cor, e outras propriedades de várias entidades existentes no ambiente. Dispositivos de resposta provêem uma resposta sensorial aos usuários, estimulando seus sistemas visuais, auditivos, e outros sistemas de percepção (Poupyrev & Ichikawa, 1999).

Formas e Técnicas de Interação

Formas de interação são canais ou instâncias de ações, enquanto que **técnicas de interação** podem ser vistas como conceitos de nível mais alto, como atividades. Por exemplo, se a função do usuário é pintar uma parede, as formas podem incluir “mergulhar o pincel na tinta” e “arrastar o pincel na parede”.

Uma divisão similar de níveis de ação pode ser encontrada, por exemplo, na Teoria da Atividade (Leontjev, 1978) onde são definidos três níveis: atividade, ação e operação. Atividade corresponde à motivação necessária para se executar algo (por

exemplo, pintar uma parede), e ela pode ser dividida numa seqüência de ações, cada uma tendo seus objetivos (por exemplo, mergulhar o pincel para pegar um pouco de tinta). O nível mais baixo consiste de operações que são consideradas processos automáticos ou subconscientes correspondentes a condições atuais e ajuste das ações, essas operações compõem as formas de interação.

Os limites desta pesquisa podem ser apresentados através da diferenciação de formas de interação com técnicas de interação. A Figura 2.7 mostra os componentes da interação homem-computador. A seqüência de interação inicia de uma ação humana (atividades disparadas pelo usuário) que se dá a partir de um dispositivo de entrada (*mouse*, teclado, entre outros). Técnicas de interação serão utilizadas para mapear a entrada do usuário, do dispositivo para a aplicação do computador (Norman, 1986). Finalmente, existem as formas de interação executadas que ocorrem dentro do ambiente virtual. Essa tarefa pode, por exemplo, ser realizada pelo avatar do usuário. Essa fase final, ou a representação da interação, é o interesse principal dessa dissertação.

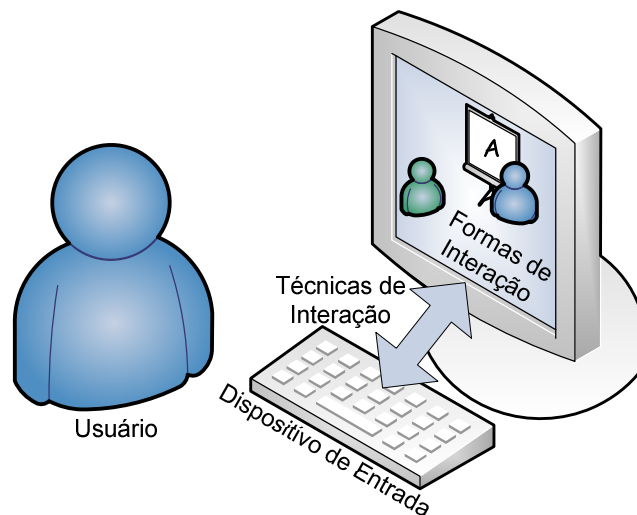


Figura 2.7: Componentes da interação homem-computador.

Autonomia, Interação e Presença

Zeltzer (1992) propôs uma taxonomia de sistemas de simulação gráfica baseada em três componentes: autonomia, interação e presença.

Autonomia reflete a medida no qual o ambiente age sobre si. Em um CVE, o nível de autonomia é muito alto quando se consideram as ações dos outros participantes. Baixa autonomia pode ser vista quando as entidades do ambiente permanecem inalteradas até o usuário ativar uma ação que o envolva. Ambientes altamente autônomos, como o mundo real, seguem seus próprios objetivos, evoluindo e desenvolvendo-se, quer o usuário faça algo ou não.

Em seu *framework*, Zeltzer define **interação** como a ação recíproca entre participante e ambiente, seguindo logicamente as leis que os governam. O ambiente possui muitos aspectos de interação, e disponibilizar o acesso direto do participante a todos eles exige uma grande preocupação com o controle. Isso é referenciado como o **problema dos graus de liberdade**. Quanto mais liberdade é dada ao participante, em relação a sua interação com o ambiente, mas controle é exigido dele, tornando sua interação mais complexa. Em um CVE, onde existem muitas entidades e ações, as operações possíveis devem ser organizadas de modo a reduzir funcionalmente o número de graus de liberdade que devem ser diretamente controlados pelo participante. Um importante aspecto dessa organização é a visibilidade de interações possíveis (ou seja, quais entidades podem interagir e como).

Presença, por Slater & Wilbur (1997), é o estado de consciência ou senso (psicológico) de estar no mundo virtual. Heeter (1992) descreve presença como “o sentimento subjetivo de existência dentro de um dado ambiente”. No modelo de Zeltzer, a presença é definida pela quantidade e fidelidade dos canais sensoriais de entrada e saída disponíveis. Por não ser possível simular o mundo físico em todos os seus detalhes e complexidade, é importante identificar os detalhes sensoriais que devem ser fornecidos ao participante para que ele execute a tarefa corretamente. Também é importante suprir o ambiente com ferramentas que possibilitem as performances motora e perceptiva necessárias à tarefa, da maneira mais representativa possível. O uso de habilidades sensores-motor familiares para manipular objetos virtuais diretamente, através de dispositivos de entrada naturais, contribui para o senso de presença do participante de ambientes virtuais. Assim, a presença fornece uma medida do grau de adaptação dos canais de entrada e saída do sistema com o participante humano. Isso sugere que um ambiente, na definição de presença de Zeltzer, depende em grande parte da interface e de questões tecnológicas como campo de visão, precisão de rastreamento e velocidade dos dispositivos de entrada, e a taxa nas quais as visões do ambiente são mostradas ao participante.

Considerando a perspectiva de sistemas distribuídos, interação implica em latência mínima, um senso de presença e a habilidade de acessar e modificar conteúdo (Brutzman, 1998). A velocidade de uma ação e o correspondente retorno fornece um forte sentimento de estar no controle e também aumenta o sentido de presença. No intuito de ampliar a presença em CVEs, também é importante aumentar a qualidade das interações físicas e sociais dos participantes entre si e com o ambiente. Um ambiente

virtual deve dar aos participantes um sentimento de presença, fornecendo melhor interação e uma interface intuitiva (Pandzic et al., 1996).

2.4.2 Aspectos de Interação

A complexidade e o amplo escopo do conceito de interação tornam difícil seu desenvolvimento dos pontos de vista de análise e projeto. Nesta seção serão mostrados alguns aspectos da interação que afetam a construção de modelos conceituais de interação, a fim de elucidar todas as facetas existentes na interação.

Comunicação

Com relação à intenção de comunicação, Burgoon & Ruffner (1978) definem que as interações podem se apresentar sem intenção de comunicação e com intenção de comunicação. Geralmente, interações com objetos inanimados não possuem a intenção de comunicar algo, como, por exemplo, abrir uma porta. Devido a essa intenção, a interação pode até ser considerada como pré-requisito à comunicação, mas o oposto não é verdade.

Algumas tarefas e atividades incluídas em ambientes virtuais não são planejadas, ou usadas, como ferramentas de comunicação. Bogh-Andersen & Callesen (2001) definem comportamentos como partes classificadas e segmentadas de um fluxo contínuo de ações. Comportamentos são representados através de ações, e eles possuem aspectos contínuos e discretos. Nem todos os comportamentos são usados para significar algo, apesar da maioria ser. Por exemplo, caso um participante esteja checando sua localização em um ambiente virtual através de um mapa, não existe necessariamente aspectos de comunicação nessa ação. Porém, se a ação for apresentada aos outros participantes, existe a possibilidade de ocorrer uma comunicação não intencional, uma vez que os demais participantes podem pensar que o personagem está perdido ou procurando o caminho para algum lugar.

Já na interação com intenção de comunicação, pode-se apresentar a comunicação de maneira verbal ou não-verbal. A comunicação verbal utiliza-se de meios de comunicação tradicionais como a fala, ou meios alternativos como a escrita. Porém, os aspectos não-verbais são de extrema importância no processo de comunicação. Riva & Calimberti (2001) consideram que a comunicação expressa uma relação psico-social tão importante quanto sua relação lingüística. Comunicação é, portanto, vista como um sistema global nos quais os participantes constroem significados utilizando e

interpretando não somente palavras, mas também entonações, gestos, posturas, atitudes e o espaço entre eles. Uma vez que interação pode possuir uma natureza comunicativa não-verbal, é importante discutir sua teoria.

“Comunicação não-verbal preocupa-se com posturas e suas indicações do que as pessoas estão sentindo. Posturas são um meio de se comunicar e são definidas através da posição específica dos braços e pernas, e ângulos do corpo. Essa comunicação não-verbal é essencial para conduzir a interação entre as pessoas com ou sem contato”.

(Thalmann 2001, p. 32)

Mensagens não-verbais é uma parte poderosa do processo comunicativo. Devido o comportamento não-verbal operar de modo inconsciente, a maioria das pessoas falha em reconhecer o amplo leque de funções que tais mensagens revelam (Burgoon & Ruffner, 1978). A importância da comunicação não-verbal em todas as interações face-a-face sugere que sua utilização em jogos e CVEs possa melhorar a interação das entidades desses ambientes.

De acordo com Fiske (1982), comunicação não-verbal é transmitida através de códigos de apresentação como gestos, movimento dos olhos ou qualidade da voz. Ele declara que “esses códigos podem dar somente mensagens do aqui e agora. Meu tom de voz pode indicar minha atitude atual para meu alvo ou ouvinte: não pode enviar uma mensagem sobre meus sentimentos da semana passada. Códigos de apresentação são limitados à comunicação face-a-face ou quando o comunicador está presente”.

A Tabela 2.1 apresenta as formas, ou códigos, de comunicação não-verbal. As categorias foram combinadas por Manninen (2004) por aspectos relacionados e os autores correspondentes das formas individuais, identificados. A literatura de comunicação possui variações nas categorias como quantidade, limites e definições. Essa apresentação é uma combinação de vários modelos.

Tabela 2.1: Categorias de comunicação não-verbal.

Categoria	Fontes
Tátil: reflete o uso do tato em situações de comunicação. Contatos físicos ou corporais como aperto de mãos.	Burgoon & Ruffner, 1978 Argyle, 1975 Fiske, 1982
Aparência Física: define os atributos de imagem e apresentação. Contém os aspectos visuais da apresentação de alguém.	Burgoon & Ruffner, 1978 Argyle, 1975 Fiske, 1982
Linguagem Corporal: inclui todos os movimentos corporais, exceto toque. Por exemplo, sinais com a cabeça, posturas e gestos.	Burgoon & Ruffner, 1978 Argyle, 1975 Fiske, 1982 Allbeck & Badler, 2001

Categoria	Fontes
Expressão Facial: pode ser dividida em posição das sobrancelhas, posição dos olhos, forma da boca, e tamanho do nariz. Por exemplo, envergonhar-se, transpirar, entre outros.	Argyle, 1975 Fiske, 1982 Birdwhistell, 1970
Comportamento Espacial: consiste de proximidade, orientação, comportamento territorial e movimento no sentido físico. Proxêmica inclui ações relacionadas ao uso do espaço pessoal.	Burgoon & Ruffner, 1978 Argyle, 1975 Fiske, 1982
Paralinguagem: é a parte auditiva não-verbal da fala e inclui o uso da voz na comunicação. Isto é, vocálicos, aspectos não-verbais da fala, vocalização não-verbal.	Burgoon & Ruffner, 1978 Argyle, 1975 Fiske, 1982
Visão: são movimentos na área facial e olhos, isto é, olhar, movimento e contato dos olhos, orientação visual.	Argyle, 1975 Fiske, 1982 Allbeck & Badler, 2001
Detalhes Ambientais: define a aparência da vizinhança dando marcas contextuais. Isto é, artefatos, manipulação do ambiente físico, entre outros.	Burgoon & Ruffner, 1978 Argyle, 1975
Cronologia: envolve o uso e percepção do tempo. Como as pessoas percebem, organizam e reagem ao tempo.	Burgoon & Ruffner, 1978
Olfato: reflete o efeito comunicativo não-verbal de cheiro e odor.	Jacob et al., 1993 Davide et al., 2001 Masterson, 1996

Como as categorias mencionadas são da área de comunicação, elas não foram desenvolvidas em ambientes virtuais contemporâneos. Contudo, existem descrições de experimentos em laboratório da maioria delas: Benford et al. (1997), Vihjálmsson e Cassell (1998), Cassell (2000), Thalmann (2001), Allbeck e Badler (2001).

Snowdon et al. (2001) aponta a importância da comunicação não-verbal no contexto de CVEs:

“Estudos sobre conversação têm demonstrado que gestos de mãos e sinais com a cabeça, olhar, e elevação das sobrancelhas, determinam o meio pelas quais as expressões são interpretadas. Bem como os gestos de resposta, que enfatizam e dão significado a muitas expressões de conversação, indicação de vestimenta, postura e (muitas vezes determinado pela cultura) maneirismos, fornecem mais contexto secundário para nossa interpretação de negociações verbais”.

(Snowdon et al., 2001, p.10)

De acordo com Rokeby (1998), o movimento do corpo pode ser lido de dois níveis diferentes. Existe o conteúdo semântico do gesto, no qual o movimento é interpretado simbolicamente (por exemplo, o sinal de OK), e existe a pura experiência visual do gesto (por exemplo, dinâmicas do movimento). Na comunicação interpessoal, as pessoas estão simultaneamente interpretando gestos em muitos níveis. Essa

combinação é a base de uma comunicação rica. Para que essa amplitude e qualidade de comunicação sejam levadas a ambientes virtuais, os desenvolvedores devem estar cientes da importância desses múltiplos modos e estar aptos a criar ferramentas e equipamentos para suportá-los.

Vilhjálmsón & Cassell (1998) enfatizam que a comunicação entre humanos faz uso extensivo do canal visual para a administração da interação, onde muitos detalhes sutis e mesmo involuntários são percebidos da postura, olhar e gesto. Eles argumentam que a modelagem e animação desses comportamentos são cruciais para a credibilidade e efetividade da interação virtual.

De acordo com Benford et al. (1997), o gesto é uma parte importante da conversação e alcança desde um acompanhamento quase inconsciente do discurso à completa e bem formada linguagem de sinais dos surdos. Em complemento aos gestos, as expressões faciais também têm um papel importante na interação humana como a mais poderosa representação externa das emoções, seja consciente ou inconsciente. O problema com expressões faciais, no contexto de CVEs, é a fina granularidade de detalhes envolvidos, além dos problemas técnicos inerentes na captura e na representação.

Controle é especialmente problemático no contexto de CVEs. Quando o corpo do participante é puramente virtual, não existem meios convencionais de controlar os códigos sutis da comunicação não-verbal. Porém, o mundo físico contém numerosos exemplos de processos automáticos, ou subconscientes, que não necessariamente são controlados diretamente.

“Oradores escolhem parceiros de diálogos, mas não escolhem elevar suas sobrancelhas junto com a ênfase em uma palavra, ou olhar para uma pessoa que tropeça e cai. Embora nós também nos preocupemos com esses detalhes como ouvintes”.

(Vilhjálmsón & Cassell, 1998, p. 271)

Badler & Allbeck (2000) citam o jogo *The Sims* (Máxis & Electronic Arts, 2000) como um bom exemplo de uso de vocalizações não-verbais. Os personagens vivem suas vidas participando desde conversas educadas até discussões zangadas. Porém, eles não falam uma linguagem compreensível, em vez disso, comunicando-se através de gestos, balões com símbolos e vocalizações não-verbais. Mesmo assim, é fácil distinguir uma conversa educada de uma conversa esquentada, através do volume e frequência de vocalizações não-verbais. Rouse (2000) cita o desenvolvedor de *The Sims*, Will Wright:

“E nós decidimos que a voz seria inteiramente para conteúdo emocional: você pode dizer se o personagem está flertando, chateado, descontraído, ou cansado pelo tom da voz e cadência. Mas da maneira que é desenvolvida, porque não existe a semântica, porque você não está ouvindo as palavras, você naturalmente fica lá e imagina as palavras claramente fluindo”.

(Rouse, 2000, p. 460-461)

O conhecimento teórico de comunicação não-verbal é relativamente forte quando se consideram as interações face-a-face do mundo real. Embora a aplicação dessas teorias no contexto de CVEs não possa ser replicada diretamente, existe uma grande parte do conhecimento que pode ser transformado para desenvolver ambientes virtuais mais ricos.

Níveis de Interação

Laurel (1993) define níveis de interatividade caracterizando-o parcialmente em três variáveis: Frequência (quão freqüente se pode interagir), Alcance (quantas escolhas estão disponíveis) e Significância (o quanto importa afetar as escolhas).

Nível de interação também pode ser visto em função da velocidade de resposta, do leque de possíveis interações e do mapeamento do controle (Nunamaker, 1997). Um tempo de resposta rápido promove interatividade, fornecendo reação imediata às operações executadas pelo usuário. Jogos de tiro e de ação em primeira pessoa, como *Wolfenstein: Enemy Territory* (Splash Damage & Activision, 2003), representam o mais alto nível de interatividade atualmente alcançado através da evolução dos dispositivos de entrada e saída, aumentando a velocidade de resposta.

A segunda variável, ou o número de escolhas que o usuário pode fazer em certo momento no tempo, reflete-se muito nas tecnologias de comunicação. No domínio dos jogos, essa dimensão do nível de interatividade é geralmente dominada pelos jogos de interpretação (*Role Playing Game* – RPG) como *Neverwinter Nights* (Bioware Corp. & Atari, 2002) e simuladores como *Flight Simulator* (Microsoft, 2004).

A terceira dimensão, ou o mapeamento de controles, é a relação entre as interações físicas do usuário com os mecanismos de entrada e a modificação do ambiente virtual. Devido às limitações nas interfaces de usuário, essa dimensão é vista como a mais problemática na busca de um alto nível de interatividade. Jogos de ação e simuladores são geralmente controlados através de teclado e mouse, necessitando assim de mapeamentos complexos de teclas para proporcionar o acesso a todas as ações. O que está longe do ideal apresentado por Steuer (1992): interação eficaz para o mapeamento de controles deve ser natural e causar resultado previsível.

A necessidade de interação rica e de formas correspondentes origina-se da natureza da percepção humana.

“Humanos desejam informações paralelas. As pessoas fazem uso dos estímulos sensoriais para ajudá-los a reduzir a ambigüidade. O som que uma carta faz ao cair na caixa de correio informa o quanto essa caixa de correio está cheia. O eco em uma sala nos informa sobre o material dos objetos e das paredes. Nós usamos o movimento da cabeça para melhorar nossa interpretação direcional do som. Nós usamos o toque para determinar a suavidade de uma superfície. Múltiplas modalidades nos dão uma janela combinatória rica de nosso ambiente que nós usamos para definir e refinar nossa percepção do ambiente. É a nossa maneira de reduzir a ambigüidade”.

(Furness, 2001, p. 85)

A busca de interações cada vez mais ricas é um fator importante em sistemas de colaboração (*Computer Supported Cooperative Work – CSCW*). A consciência do espaço de trabalho é importante para o usuário do sistema que precisa estar ciente das interações das outras pessoas: onde elas estão trabalhando, o que estão fazendo, as alterações que realizam e as futuras intenções.

Wloka (1995) descreve fatores que enriquecem a interação em ambientes virtuais: é mais interessante que a interface com o objeto seja o próprio objeto, em vez de utilizar métodos indiretos como menus de seleção; é importante utilizar múltiplos dispositivos de entrada, simultaneamente ou não; e os dispositivos de entrada, utilizados em realidade virtual, devem possuir um número grande de graus de liberdade.

Um grande número de graus de liberdade tem sido desejado pelos desenvolvedores de ambientes virtuais que almejam tornar essa experiência tão interativa quanto possível (Joiner, 1998). Contudo, isso pode causar sérios problemas, mesmo em situações onde as técnicas de interface podem tratar as interações entre humano e máquina. Rokeby (1998) desenvolveu uma experiência com um grande número de parâmetros de comportamento refletindo as ações dos usuários, e o que percebeu foi que as pessoas se perdiam: “Ironicamente, o sistema foi interativo em tantos níveis que a interação se tornou indigesta”.

A experiência claramente indica a necessidade de uma seleção cuidadosa das ações modeladas. Rouse (2000) descreve os benefícios de se reduzir os graus de liberdade no contexto do desenvolvimento de um jogo de computador chamado *Loom* (LucasArts, 1991) que é um exemplo completamente focado no que se deseja alcançar. Ao invés de tentar incluir todas as mecânicas de jogo possíveis, o desenvolvedor gastou seu tempo em descobrir quais as mecânicas eram necessárias para contar a sua história.

O resultado é uma grande simplificação do jogo, tornando-se mais elegante e fácil de navegar.

Caso seja necessário o aumento do grau de liberdade, isto pode ser conseguido através do uso de interação hierárquica, ou tarefas. Gabbard & Hix (1997) dizem que é possível enumerar tarefas de baixo nível dentro das tarefas específicas da aplicação e então fazer uso dessas tarefas em outros projetos. Pode-se projetar e desenvolver interações utilizando-se de um conjunto de blocos construídos em diferentes níveis de complexidade. Os desenvolvedores estabelecem um conjunto de regras físicas e limitantes de como o universo funciona.

Nesse propósito, o uso de simulações encaixa-se bem. Por exemplo, o uso de simulação física, composta de peso de objetos, gravidade, entre outros, é um sistema que oferece possibilidades infinitas para interação apenas desenvolvendo um conjunto básico de mecanismos para interagir com o sistema. A filosofia de criar simulações é constante nos trabalhos de Weisman (1998) que diz: “Já não fazemos jogos, nós fazemos caixas de areia e damos aos jogadores algumas ferramentas divertidas que eles usam para criar o jogo”.

Espaço

Outros aspectos importantes para a interação são os aspectos espaciais de interação e as categorias de comunicação não-verbal de Comportamento Espacial e Detalhes Ambientais. Esses aspectos espaciais relacionam as ações e as atividades realizadas pelos participantes no espaço e lugar disponibilizados pelo ambiente virtual. Churchill et al. (2001) descreve a diferença entre espaço e lugar. **Espaço** é um volume físico (ou virtual) que pode conter artefatos e onde eventos podem acontecer. O espaço só se torna um **lugar** quando uma atividade é realizada. Um espaço pode se tornar vários lugares diferentes dependendo do momento. Por exemplo, uma quadra de esportes pode ser um lugar para praticar esportes, realizar festas, lecionar aulas de dança, entre outras. Apesar de o espaço ser o mesmo, o entendimento de como o espaço pode ser utilizado depende da ocasião. Os tipos de interação disponíveis irão depender do lugar e não do espaço.

Laurel et al. (1998) diz que a experiência humana é um fenômeno relacionado com o lugar. O homem chega a conhecer o lugar, utilizando todos os seus sentidos, em virtude das ações que executa no local, e de um ponto de vista. Cabe ao ambiente registrar a presença e as ações dos participantes, como também as marcas que os participantes deixam.

O espaço pode ser utilizado para aprimorar a interação entre os usuários e o mundo. Destaques devem ser inseridos no espaço para facilitar a navegação e ressaltar objetos especiais ou de interesse (Clark-Willson, 1998). Essas marcas podem ser geradas também pelas ações do usuário, possibilitando uma melhoria em sua orientação e a possibilidade de retornar a um ponto anterior (Bates, 2001, p. 61). O próprio ambiente deve servir de mapa àqueles que estão navegando. Além disso, o ambiente pode sugerir determinadas ações aos usuários. Por exemplo, o ambiente de um *shopping center* aumenta as chances de ocorrer uma negociação de venda.

Outro aspecto que deve ser levado em consideração é a limitação do ambiente. Existem várias indicações de benefícios obtidos através da restrição seletiva. Clarke-Willson (1998) observa que os melhores jogos desenvolvem-se num espaço limitado e restrito. Em *Myst* (Cyan Worlds Inc. & Ubisoft, 1993), o jogador fica preso em uma série de ilhas; em *7th Quest* (Trilobyte & Virgin Interactive, 1992), o jogador está preso em uma mansão; e em *Neverwinter Nights* (Bioware Corp. & Atari, 2002), onde é possível criar seus próprios ambientes, o criador define limites.

O problema com ambientes sem limites, ou muito grandes, é a falta de conteúdo. No desenvolvimento e criação de ambientes virtuais deve-se ter certeza de que, qualquer que seja o lugar para onde os usuários se dirijam, o local será interessante o suficiente para que ele não desperdice seu tempo. O resto do mundo deve estar fora dos limites do usuário. Esses limites devem ser dados de maneira contextual. Por exemplo, uma ilha deve estar cercada pelo mar, onde os usuários não podem ter acesso.

Jää-Aro & Snowdon (2001) afirmam que um espaço vazio não possui informação útil. São os artefatos, as entidades no espaço e suas ações que determinam a informação. Assim, são as entidades e as ações (comportamentos e interações) que tornam o ambiente significativo e útil.

De acordo com Huxor (2001), um aspecto importante da interação, e que falta em muitos ambientes, é a habilidade dos usuários construírem seus próprios espaços dentro do mundo. Essa funcionalidade permite a introdução de conteúdo desenvolvido pelo usuário e de interesse específico. Sistemas que possibilitam a construção de espaços, como o *ActiveWorlds* (Activeworlds Inc., 1997) e MUDs, mostram a importância da construção e da propriedade em sua popularização e seu uso.

A posição do participante no espaço e o resultado de sua aproximação, junto com a orientação, formam a maior parte da categoria de comunicação não-verbal chamada de Comportamento Espacial. Essa categoria disponibiliza informações sobre

identidade, interesse e ações do usuário (Harrison & Dourish, 1996). Mas, a locomoção também é uma parte importante do Comportamento Espacial. Existem muitas maneiras de se locomover e o uso de artefatos e objetos pode estender essas maneiras, mas o importante é que essa locomoção disponibiliza uma abundância de informações sobre as ações dos participantes.

Som

A parte auditiva da comunicação não-verbal, ou Paralinguagem, é relacionada, na literatura de comunicação, principalmente com atributos da fala e do discurso. Porém, em ambientes virtuais, principalmente em CVEs, existem outros elementos sonoros que fazem parte do processo interativo.

O diálogo falado é o meio comum de comunicação no mundo real. Johnson & Leigh (2001) revelam que o áudio é o canal de comunicação mais importante em um espaço colaborativo compartilhado. O suporte a som espacial, que indica o ponto ou pelo menos a direção da origem do som, pode aumentar a colaboração em ambientes virtuais. Para isso, a qualidade do som é importante, pois se o som se deteriora, o processo de colaboração pode desaparecer. Uma das razões da importância do canal de áudio pode ser seu meio natural de comunicação, pois toda, ou pelo menos a maior parte, da informação contida na fala é conduzida através do canal da voz, sem nenhuma necessidade de dispositivos de controle adicionais.

Latta (1998) apresenta a importância da boa qualidade do áudio como um mecanismo de transmissão de mensagens, emoções e humor:

“O segundo estímulo mais importante [depois da visão], e um dos melhores meios de se comunicar emoção e humor, é o som... Um ambiente virtual efetivo precisa possibilitar som ambiente, fala e efeitos sonoros causados pelo uso de objetos virtuais”.

(Latta, 1998, p. 501-502)

Clanton (2000) afirma que o som é a arma secreta para se controlar as emoções. Ele pode ser utilizado para antecipar eventos futuros como, por exemplo, a mudança da música ambiente pode alertar o usuário de um futuro ataque, e o som de passos pode indicar a direção de onde um inimigo está vindo. Ele pode criar sentimentos mais poderosos no usuário através de antecipação e ansiedade. Bacon (2000) acrescenta que a música narrativa apresenta características da natureza e do estado do personagem, e a importância do evento. Nesse sentido, a música de acompanhamento possui certo efeito narrativo da situação.

O som deve responder a mudanças de tensão e humor, informar objetivos alcançados ou a ocorrência de eventos importantes na história. Cada avatar pode ter seu acompanhamento sonoro audível somente para ele, que o acompanha e reflete as mudanças em sua vizinhança (Goldberg, 1998).

Assim, existe a necessidade de se criar uma verdadeira trilha sonora parecida com a de um filme. Porém, os eventos em ambientes virtuais não seguem uma ordem definida como nos filmes, sendo necessários sons de duração variável e efeitos de suavização de transição.

Sons podem chegar a substituir movimentos e animações em muitos ambientes. Clanton (2000) descreve essa característica no jogo *Myst* (Cyan Worlds Inc. & Ubisoft, 1993): “A imagem do oceano é estática, mas o som de ondas faz parecer que ele se move. O jogador ouve as portas se fecharem atrás de si, após passar por elas. Esses sons substituem as animações. A floresta parece viva por causa do som”. Kau (2001) descreve o universo sonoro de *Riven* (Cyan Worlds Inc. & Ubisoft, 1998): ele observa que as impressões do jogador são aumentadas com o uso dos efeitos sonoros e musicais. Portas de metal e pedra, elevadores e máquinas movem-se com ecos, barulhos, retinidos e rangidos que, literalmente, se torna possível ouvir seu peso e material. Um bom som pode auxiliar na imersão do usuário no ambiente.

Deve-se ter um cuidado especial ao utilizar o som para comunicação, pois existem situações onde a quantidade de sons pode ser tão grande que sua compreensão se torna impossível. Por exemplo, quando muitos avatares estão em uma sala conversando, o barulho gerado por essas múltiplas conversas paralelas pode comprometer a comunicação. Pode-se contornar esse problema utilizando-se dois canais de áudio, um para os sons ambientes e o outro para uma conversa específica do avatar.

Jensen (2001a) complementa o termo som cobrindo quatro tipos de fenômenos: fala, música, efeito sonoro e silêncio. O som possui dimensões espaciais e temporais. A dimensão espacial existe devido a sua fonte de emissão, e a dimensão temporal caracteriza o aspecto eventual do som. Na dimensão espacial, o som pode ser dividido quanto à identificação da fonte do som. Existem sons que são oriundos de fontes que não podem ser localizadas na cena, no espaço e nem no mundo, como por exemplo, um narrador. Quando a fonte existe, o som pode ser dividido ainda em visível e não-visível, dependendo da possibilidade de se observar a fonte de som naquele momento. Os sons que possuem fonte existente no mundo apresentam aspectos de profundidade, direção e sentido.

Tempo

De acordo com Singhal & Zyda (1999), ambientes virtuais em rede ou CVEs incorporam um sentido compartilhado de tempo aos participantes de uma tarefa colaborativa. Eles podem observar o comportamento dos outros participantes no momento em que se manifestam, possibilitando assim a interação em tempo real. Esse recurso está presente na categoria de comunicação não-verbal chamada de Cronologia.

A percepção e o uso do tempo são essenciais no desenvolvimento e experimentação de jogos de computador. Momentos de tranquilidade e de movimento, repetições, estruturas de tempo cíclicas e induções de limitação são usos do tempo em jogos contemporâneos (Wolf, 2001). Essa preocupação com o tempo existe em CVEs devido a seu caráter colaborativo, através da coordenação de atividades e da discussão de turnos de trabalho.

Restrições temporais possuem um papel importante na interação. Certas respostas de eventos são esperadas em um determinado espaço de tempo, como por exemplo, olhar em uma direção que está sendo apontada (Goodwin, 1981). Além disso, em jogos e simulações, o próprio ambiente pode forçar restrições temporais para os participantes. Por exemplo, o jogador ter que resolver um quebra-cabeça em determinado intervalo de tempo, ou algumas ações só serem realizadas em determinado período do dia.

O processo do tempo pode ser simulado através de convenções do mundo real, como a passagem do sol pelo céu. Porém, em ambientes virtuais pode-se representar o fluxo do tempo de maneira diferente. Ele pode ser acelerado como no jogo *The Sims* (Maxis & Electronic Arts., 2000) ou retardado como o efeito “*bullet time*” do jogo *Enter The Matrix* (Shiny & Atari, 2003).

Um problema que pode ocorrer quando se imita interações do mundo real é a ligação do tempo e do espaço. Na navegação do mundo virtual, quando se deseja ir de um local a outro, há um gasto de tempo que, às vezes, torna o processo enfadonho. Isso pode ser resolvido através de mecanismos de tele-transporte. Porém, quando retardamos o tempo, alteramos o fluxo temporal para todos os participantes do ambiente.

Drozd et al. (2001) apresenta mais uma utilidade do aspecto temporal da interação: o registro tridimensional e o mecanismo de rever um momento passado. Todas as ações do ambiente são registradas e, quando se deseja rever algum intervalo de

tempo, os participantes podem ouvir o som e até seguir personagens, como se vissem um filme tridimensional.

Representação

Nesta seção serão apresentados aspectos de representação da interação e da aparência das entidades. Jensen (2001b) apresenta que uma das características decisivas de CVEs é serem mundos simbólicos, exclusivamente constituídos de representações, símbolos e sinais digitais. A representação diz respeito a todos os aspectos da aparência do ambiente virtual, incluindo a aparência das entidades.

É importante que o usuário do sistema possa escolher a aparência de sua representação, porém algumas características podem ser utilizadas para fins específicos do próprio ambiente como, por exemplo, o uso de diferentes cores de camisas para distinguir dois times de futebol.

Benford et al. (1997) define a personificação do usuário como a atribuição de uma imagem corporal apropriada que o represente para os outros e para ele mesmo. Essa personificação é relevante em sistemas de cooperação, porém muitos sistemas vêem o usuário como um ser que observa de fora e não se preocupam com sua representação dentro do ambiente. Assim, o usuário só tem conhecimento das ações dos outros usuários através de suas ações não-personificadas.

Benford apresenta ainda questões que devem ser consideradas pelos projetistas de corpos virtuais, como representação pessoal (identidade, informações, personalização) e atividade expressiva (presença, localização, pontos de vista, pontos de ação, expressões faciais, gestos, relação com objetos e informação, histórico de atividades).

O objetivo principal da imagem corporal é expressar o sentido de presença no ambiente virtual. E quanto mais complexo é o modelo virtual, maior será a interação dentro do ambiente. Uma percepção maior das entidades aumenta o sentido de presença compartilhada no ambiente (Thalmann, 2001).

“A motivação para utilizar usuários personificados em sistemas colaborativos torna-se clara quando consideramos o papel de nossos corpos na comunicação do dia-a-dia. Nossos corpos fornecem informações imediatas e contínuas sobre nossa presença, atividade, atenção, disponibilidade, humor, estado, localização, identidade, capacidade e muitos outros fatores. Nosso corpo pode ser utilizado para se comunicar, como demonstra os inúmeros gestos da linguagem de sinais, ou pode fornecer um acompanhamento importante a outras formas de comunicação, auxiliando a coordenação e gerência de interações (também chamado de linguagem corporal)”.

(Benford et al., 1997, p. 93)

Allbeck & Badler (1998) listam sete características em avatares: aparência, vestimenta e acessórios, etnia, locomoção, ações corporais, formas de comunicação, e emoções e personalidades. Jensen (2001b) revela dois aspectos importantes que o avatar do usuário tem: aparência e funcionalidade. Aparência define como o avatar parece-se e como se apresenta no ambiente virtual. Funcionalidade define todas as funções diferentes que um avatar pode ter incluso, por exemplo, animação e padrões de comportamento, emoções via expressões faciais e capacidade de carregar objetos.

Uma funcionalidade que deve existir em ambientes virtuais é a customização de personificações. O usuário deve estar apto a escolher qual a aparência de sua representação, ou poder alterar um modelo padrão. Com esse grau de flexibilidade, encoraja-se o sentido de individualidade e a identificação do usuário.

Outro aspecto da representação do usuário em ambientes virtuais é a utilização da visão em primeira ou terceira pessoa. Na visão em primeira pessoa, o usuário vê o que sua representação vê e permite uma maior imersão do usuário, uma vez que ele sente-se inserido no mundo. A terceira pessoa fornece uma visão externa que permite ao usuário ver sua representação em ação. Ela diminui a imersão do usuário, mas permite uma melhor visualização do ambiente que o cerca. A escolha de qual visão é mais adequada depende da tarefa a ser realizada pelo usuário.

2.5 Considerações Finais

Uma avaliação dos tipos de entidades existentes em um ambiente virtual com suas características, e das diferentes definições de comportamento e interação, assim como suas peculiaridades e preocupações, estruturam um alicerce de conhecimento importante para a observação dos elementos a serem disponibilizado num modelo de interação e comportamento. A consciência dessa diversidade de fatores é crucial quando se deseja planejar um modelo abrangente e eficaz.

Foram apresentados cinco tipos diferentes de entidades, a definição de comportamento como a forma dessas entidades agirem em ambientes virtuais e os múltiplos aspectos importantes de interação como comunicação, espaço, som, entre outros. Assim, com esse conhecimento, torna-se possível apresentar alguns modelos de interação encontrados na literatura a fim de buscar contribuições para o desenvolvimento do modelo proposto. Porém, como os modelos são diferentes entre si, serão feitas comparações e posicionamentos dos elementos dos modelos com as definições de entidade, comportamento e interação apresentadas neste capítulo.

Capítulo 3

Modelos de Comportamento e Interação

3.1 Introdução

Neste capítulo são apresentados quatro modelos de comportamento e interação disponíveis na literatura. Cada modelo possui estrutura própria, e por isto é difícil compará-los. Os aspectos que não são de interesse deste trabalho serão apresentados superficialmente, enquanto que os aspectos de interação e comportamento serão discutidos em maior profundidade. A fim de se estudar esses aspectos de interação e comportamento, nas próximas Seções são apresentados os modelos CORGI, MASSIVE, SENDA e RICH INTERACTION MODEL. Após cada modelo, é apresentada uma categorização dos elementos do modelo aos conceitos apresentados no Capítulo 2. O estudo desses modelos é de fundamental importância para o desenvolvimento de um novo modelo de interação e comportamento, que é o objetivo dessa dissertação.

3.2 CORGI

Desenvolvido na universidade de Rhodes, na República Checa, CORGI (Rorke & Bangay, 1999) é um sistema de interação genérico que tanto dá suporte às técnicas de interação atuais quanto está preparado para incorporar novas técnicas. Esse sistema utiliza um modelo de fluxo de dados simplificado que descreve uma aplicação em termos dos seus dados de entrada, os quais são transformados pelos nós do sistema, atualizando a aplicação e produzindo uma saída.

3.2.1 Descrição

Esse modelo é semelhante àquele utilizado pelo VEDA – *Virtual Environment Dialogue Architecture* (Steed & Slater, 1996) e ao modelo adotado pelo VRML 2.0 – *Virtual Reality Modeling Language* (Steed, 1997). Ele possui seis tipos de componentes: Componente de Sistema, Componentes de Entrada, Componentes de Interação, Intermediários, Entidades e *Widgets*, descritas a seguir através da ilustração de um jogo de tênis de mesa, usado para melhor compreensão dos componentes.

Componente de Sistema

Cada aplicação desenvolve esse componente de maneira diferente, o que justifica a apresentação apenas de como ele interage com os demais componentes. Ele desenvolve

a semântica única de uma aplicação particular, geralmente englobando todos os outros componentes, e é responsável por produzir uma saída para o usuário.

No exemplo do jogo de tênis de mesa, o Sistema é responsável por implementar a física necessária para possibilitar movimentos realistas. Assim, implementará não apenas o sistema de colisão entre a bola, as raquetes e a mesa, como também as leis gravitacionais e as demais regras necessárias ao jogo.

Componentes de Entrada

Esses componentes são responsáveis por acessar os diferentes dispositivos de entrada e por repassar todos os dados capturados aos componentes de interação correspondentes. São implementados como pares de processos – ator e servidor de entrada – que se comunicam através de uma conexão de rede. O servidor reside em um computador dedicado e comunica-se com os atores associados a cada dispositivo de entrada utilizado. Essa estrutura é necessária por existirem diferentes tipos de dispositivos de entrada, utilizando arquiteturas distintas.

A aplicação de tênis de mesa necessita de um rastreador de movimento para o usuário controlar a posição da raquete no jogo. Ele é conectado ao servidor de entrada que se comunica com o ator da aplicação.

Componentes de Interação

Recebem dados do componente de entrada e os interpretam, passando informações relevantes a outros componentes. Cada aplicação pode possuir diferentes componentes de interação (por exemplo, um para implementar uma mão-virtual e outro para a posição da cabeça). Cada componente de interação pode utilizar diferentes componentes de entrada (por exemplo, uma mão-virtual utiliza um rastreador de movimento e uma luva de dados). O sistema possui diferentes componentes de interação, onde cada um implementa uma técnica de interação imersiva.

No jogo de tênis de mesa há dois componentes de interação: a raquete, que o usuário utiliza para interagir com a bola, e a mão-virtual, que o usuário utiliza para lançar bolas que caem da mesa.

Intermediários

Correspondem a nós no diagrama de fluxo de dados que realizam algum processamento. Eles recebem dados de um componente (geralmente dos componentes de interação, dos

widgets ou dos atores de entrada), processam esses dados, e os repassam às entidades ou aos componentes de interação.

No jogo de tênis de mesa, um componente intermediário é utilizado para calcular a velocidade da raquete. Ele recebe a posição do rastreador de posição, passada por um ator de interação, e utiliza essa informação e o tempo no cálculo da velocidade da raquete. Essa velocidade é passada para o componente de interação da raquete (junto com a posição e orientação) que será usada para calcular a nova trajetória quando interagir com alguma bola no sistema.

Entidades

Representam objetos do sistema. Cada entidade conhece os detalhes sobre o objeto que representa. Com isso, o sistema não precisa conhecer cada tipo particular de objeto, mas sim um conjunto de comandos comuns a todas as entidades, que sabem como cada comando se aplica a si.

No exemplo apresentado, as entidades são: bola, mesa, rede e chão. Os atributos da bola, por exemplo, são forma, cor e elasticidade.

Widgets

São formas especializadas de entidades. Diferente das entidades comuns, que simplesmente respondem a comandos de um componente de interação, os *widgets* geram dados dependendo do comando que recebam. Por exemplo, um *widget* botão pode gerar um número inteiro ao ser pressionado.

O jogo de tênis de mesa possui um único botão que pode ser pressionado para reiniciar o jogo.

3.2.2 Categorização

O sistema CORGI considera como entidade qualquer elemento que interage no sistema. As entidades possuem ações padrões conhecidas pelo sistema, e cada uma implementa a sua ação de maneira específica. Assim, fica fácil inserir novas entidades sem ter que alterar o sistema como um todo. Considerar como entidades os elementos de interface, especificamente os *Widgets*, cria uma ligação entre as entidades de realidade virtual e a interface, o que facilita a interação entre esses elementos.

Muito do comportamento das entidades do sistema CORGI é definido no Componente de Sistema, como o comportamento físico. Porém, podem ser utilizados

Intermediários para comportamentos particulares que necessitam de cálculos, ou ações na própria entidade.

A interação, no sistema CORGI, é tratada pelos Componentes de Interação, porém está limitada àquela realizada pelo usuário com o sistema. A interação entre as demais entidades é feita por Intermediários ou pelas próprias entidades.

3.3 MASSIVE

Desenvolvido na universidade de Nottingham, na Inglaterra, MASSIVE – *Model, Architecture and System for Spatial Interaction in Virtual Environment* (Greenhalgh & Benford, 1995) – é um sistema de realidade virtual experimental que dá suporte a atividades colaborativas, buscando escalabilidade e heterogeneidade. Ele tem como objetivo permitir a interação entre usuários de ambientes virtuais que utilizam equipamentos de diferentes capacidades e que, portanto, empregam estilos de interface de usuário radicalmente diferentes. Assim, o sistema possibilita que, desde usuários de terminais de texto, até usuários que utilizam equipamentos de realidade virtual imersiva, como capacetes e luvas, possam interagir num mesmo ambiente virtual.

3.3.1 Descrição

As funcionalidades mais relevantes do sistema são:

- ◆ Suporte à utilização simultânea por milhares de usuários sem degradação do desempenho;
- ◆ Utilização de um modelo espacial de interação que restringe as interações do usuário aos elementos que estão próximos a ele no mundo virtual;
- ◆ Suporte à utilização de múltiplos mundos virtuais através do uso de portais;
- ◆ Navegação com 6 graus de liberdade;
- ◆ Conversação via voz em tempo real;
- ◆ Suporte à execução de sons pré-gravados.

O seu modelo espacial de interação viabiliza formas ricas de interação parecidas com comportamentos do mundo real. Ele pode ser dividido em dois componentes principais: um baseado no conceito de **aura** (Çapin et al., 1999) e que facilita a escalabilidade e outro que utiliza o conceito de **consciência** e que controla a interação espacial. Esses conceitos foram apresentados no Capítulo 2.

Utilizando a teoria de sistemas distribuídos, cada objeto possui interfaces que disponibilizam os tipos de interação que eles suportam. Então, para uma interação

ocorrer, são necessárias duas condições: os dois objetos devem possuir interfaces compatíveis, e suas auras precisam estar próximas o suficiente para que haja colisão (os objetos possuem uma aura separada para cada interface). Essas condições refletem o conceito de **negociação espacial**, que combina a técnica de detecção de colisão da realidade virtual com o conceito de serviço de nomes baseado em atributo (ANSA, 1989) existente em sistemas distribuídos.

Quando um objeto entra no mundo, ele estabelece um contato com o negociador principal, o gerente de aura, declarando suas interfaces como serviços oferecidos, com suas respectivas auras. O gerente de aura fica responsável por analisar a ocorrência de alguma colisão de auras, quando, então, repassa as interfaces dos objetos envolvidos para que possam estabelecer conexões ponto-a-ponto entre si (Figura 3.1). Cada gerente de aura é responsável por um ou mais mundos e quando um objeto muda de mundo, através dos portais, ele também muda de gerente de aura.

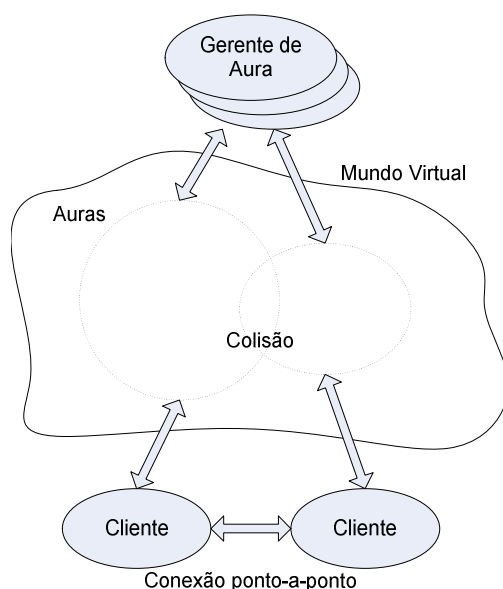


Figura 3.1: Funcionamento do gerente de aura.

Quando uma colisão de auras ocorre e a conexão entre os objetos se estabelece, o cálculo do nível de consciência mútuo usando foco e nimbo é de responsabilidade dos objetos.

3.3.2 Categorização

Chamadas de objetos, as entidades no sistema MASSIVE seguem a definição dada neste trabalho. O uso de técnicas de sistemas distribuídos contribui para a implementação dessas entidades como elementos ativos no sistema e registrados em um servidor de nomes, possuindo interfaces de interação.

O comportamento mais focado em MASSIVE é a interação, mas acredita-se que, utilizando os mesmos recursos da interação, é possível inserir comportamentos diferentes de interação nas entidades desse sistema.

A interação é possibilitada através da colisão de auras, que é detectada pelo gerente de auras. Uma vez que essa colisão existe, é estabelecida uma conexão entre as entidades para que as próprias decidam se realmente houve uma interação. Essa decisão é baseada no cálculo de consciência feito pelas entidades.

Utilizando-se dos recursos de sistemas distribuídos, MASSIVE trata cada entidade como um elemento ativo e registrado no sistema distribuído, o que facilita o desenvolvimento de novos tipos de entidades.

3.4 SENDA

Desenvolvido na universidade Carlos III, na Espanha, o *framework* SENDA (Sánchez-Segura et al., 2004) foi definido para facilitar e fornecer um guia rigoroso ao desenvolvimento de ambientes virtuais, focando principalmente nas fases de análise e projeto.

3.4.1 Descrição

Baseado nos padrões gerais de interação de Kaur (1999), SENDA define:

- ◆ Taxonomia de componentes que permite uma rápida identificação e classificação dos componentes a serem desenvolvidos;
- ◆ Arquitetura de três camadas (percepção, decisão e ação) para esses componentes; e
- ◆ Padrões de comportamento interativo para cada tipo de componente da taxonomia.

Em SENDA, interação refere-se não somente à relação entre o usuário e o computador, mas também entre os componentes existentes em um ambiente virtual. O estudo da interação, considerando todos os componentes, é útil para a análise e o projeto desses sistemas. Assim, **mecanismo de interação** é o meio através do qual os componentes agem e comunicam-se com outros componentes, ou com o usuário, em situações específicas. Durante o projeto e a implementação de ambientes virtuais, cada componente tem que ser dotado de um conjunto de mecanismos que combinados de maneira específica são chamados de **padrão de comportamento**.

Os mecanismos de interação podem-se classificar nos seguintes tipos: Ação, Percepção e Decisão. A **Ação** é um mecanismo que pode ser subdividido em: Reflexão, Interação interna e Interação externa. A **Reflexão** modifica o estado do componente sem informar isso aos outros componentes. A **Interação interna** possibilita que mensagens sejam enviadas de um componente a outro. A **Interação externa** de um componente permite ao usuário percebê-lo. Existem muitos tipos de mecanismos de interação externa, dependendo da interface utilizada. Por exemplo, a **Animação** é um mecanismo de interação externa que altera a representação do componente de forma que o usuário percebe a alteração através da visão; já o **Som**, permite que o usuário perceba alterações através do estímulo auditivo. O outro mecanismo de interação, a **Percepção**, é um mecanismo através do qual o componente capta informações oriundas do usuário ou de outros componentes que tenham ativado seus mecanismos de interação (interna ou externa). Finalmente, o último mecanismo de interação, a **Decisão**, é o mecanismo utilizado por um componente para reagir a uma situação específica.

Segundo SENDA, qualquer componente encontrado em um ambiente virtual pode ser classificado segundo dois critérios: baseado na existência de mecanismos de interação externa e baseado na atividade do componente.

A classificação baseada na existência de mecanismos de interação externa considera os seguintes tipos de componentes:

- ◆ **Componentes Imperceptíveis**, os quais não interagem ou não podem ser diretamente percebidos pelo usuário (não possuem mecanismo de interação externa); e
- ◆ **Componentes Perceptíveis**, que estão fisicamente presentes no ambiente e podem interagir ou ser percebidos pelo usuário.

A classificação baseada na atividade do componente define componentes:

- ◆ **Passivos**, como aqueles que podem possuir apenas um mecanismo de interação externa associado;
- ◆ **Reativos**, como os que nunca iniciam uma interação com outro componente ou com o usuário, mas são capazes de reagir a ações de outros componentes ou dos usuários, podendo ser dotados de qualquer ação que não inicie uma interação;

- ◆ **Pró-ativos**, como aqueles que iniciam interações com outros componentes ou com os usuários e podem possuir qualquer ação, mesmo que não inicie uma interação; e
- ◆ **Pró-ativos e reativos**, como os que podem iniciar uma interação com outro componente ou com os usuários e podem reagir a uma possível interação, podendo ter qualquer ação, mesmo que não inicie uma interação.

As combinações entre essas duas classificações geram diferentes padrões de comportamento que são apresentados a seguir. Esses padrões são apresentados em diagramas de atividade UML (Booch et al., 1999) para serem facilmente entendidos e adaptados sempre que se desejar projetar qualquer novo tipo de componente em um novo ambiente virtual.

Padrão de Comportamento Perceptível Reativo

Esse comportamento é iniciado pela percepção de algo no ambiente virtual ou por uma ação do usuário (através do mecanismo de percepção). Tão logo algo relevante seja detectado, o componente analisa de que forma ele é afetado por esse evento, utilizando seu mecanismo de decisão para disparar ações compatíveis. Quando o mecanismo de decisão detecta a necessidade de algum cálculo, mecanismos de ação são ativados. O evento pode afetar somente o componente ou pode também afetar outros componentes. Caso o usuário esteja envolvido na ação, mecanismos de interação externa são executados. Se, além disso, o evento afeta outros componentes, um mecanismo de interação interna é acionado. Somente componentes reativos e pró-ativos e reativos podem receber mensagens através de seus mecanismos de percepção. A representação desse padrão de comportamento na forma de um diagrama está ilustrada na Figura 3.2.

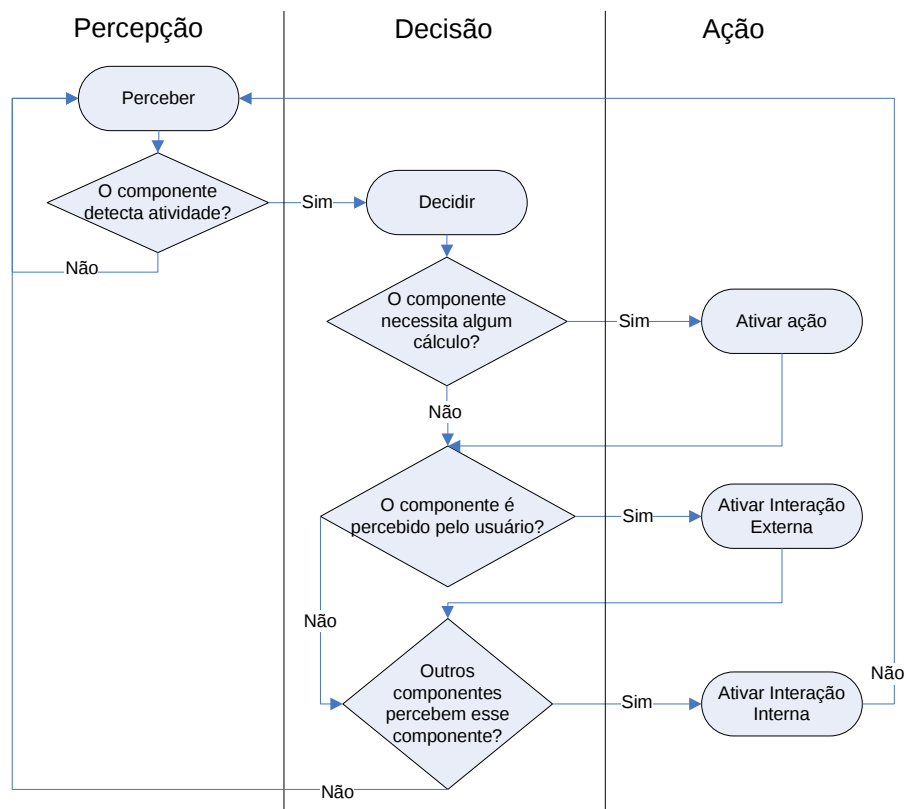


Figura 3.2. Padrão de comportamento perceptível reativo.

Um exemplo desse tipo de componente é uma porta em uma sala virtual que se abre quando tocada por um avatar.

Padrão de Comportamento Perceptível Pró-ativo

Um componente desse tipo é continuamente ativado por seus mecanismos de decisão. Se uma atividade interna (cálculo, verificação, etc.) resultou de um mecanismo de decisão, o mecanismo de ação correspondente será ativado. Se o usuário estiver envolvido, o mecanismo de interação externa correspondente será acionado. Se outros componentes estiverem envolvidos, mecanismos de interação interna serão ativados para estabelecer comunicação com outros componentes. O padrão é mostrado na Figura 3.3

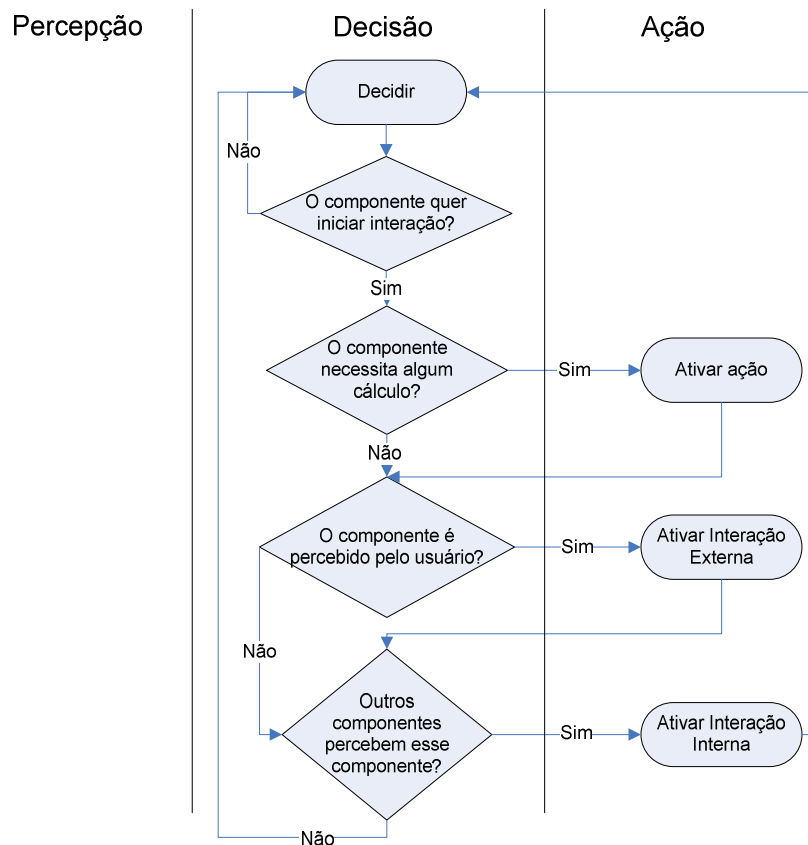


Figura 3.3. Padrão de comportamento perceptível pró-ativo.

Um exemplo típico desse tipo de componente é um quadro em uma sala virtual, mostrando informações específicas dependendo dos avatares que estão na sala.

Padrão de Comportamento Perceptível Passivo

Esse tipo de componente pode ter somente mecanismos de interação externa, então o usuário pode perceber sua presença embora seja estático. O padrão é apresentado na Figura 3.4. Exemplos desse tipo de componente são os itens de decoração de um ambiente virtual.

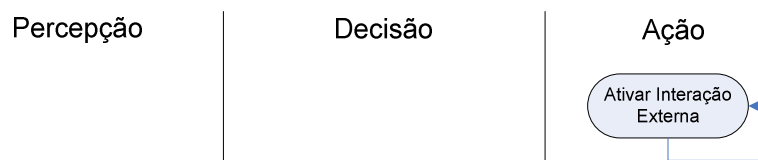


Figura 3.4. Padrão de comportamento perceptível passivo.

Padrão de Comportamento Perceptível Pró-ativo e Reativo

Esse padrão inclui os padrões perceptível pró-ativo e perceptível reativo. A Figura 3.5 apresenta o padrão.

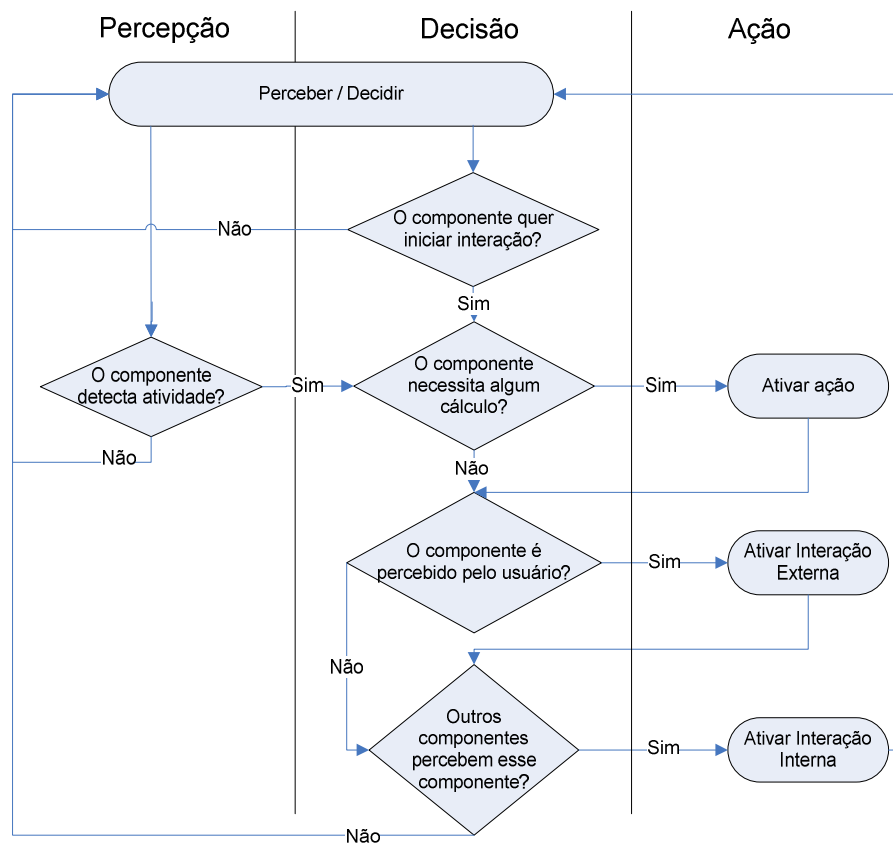


Figura 3.5. Padrão de comportamento perceptível pró-ativo e reativo.

Um exemplo desse tipo de componente pode ser uma porta em uma sala virtual que se abre quando um avatar se aproxima e também abre automaticamente a cada minuto.

Padrão de Comportamento Imperceptível Reativo

Tão logo o componente detecte algo no ambiente virtual, seu mecanismo de decisão analisa como o evento percebido o afeta e dispara ações compatíveis. Se outros componentes estão envolvidos, mecanismos de interação interna serão ativados, estabelecendo comunicação. Somente componentes reativos ou pró-ativos e reativos podem receber essa comunicação através de seus mecanismos de percepção. Como o usuário não percebe esses componentes, eles não possuem mecanismos de interação externa. O padrão desse componente é apresentado na Figura 3.6.

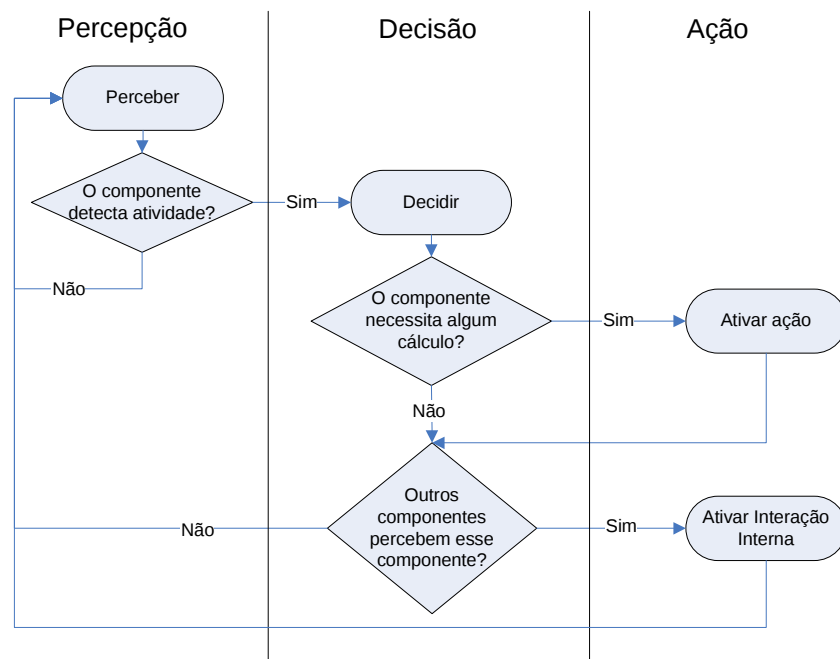


Figura 3.6. Padrão de comportamento imperceptível reativo.

Um exemplo desse tipo de componente pode ser um *software* de rastreamento que registra as posições dos avatares no ambiente.

Padrão de Comportamento Imperceptível Pró-ativo

Um componente com esse padrão de comportamento toma decisões de interação sem que seja necessária a ocorrência de qualquer evento externo. Se alguma ação for necessária, o mecanismo de ação correspondente é acionado. Se outros componentes estão envolvidos, mecanismos de interação interna são ativados. O diagrama desse padrão de comportamento é apresentado na Figura 3.7.

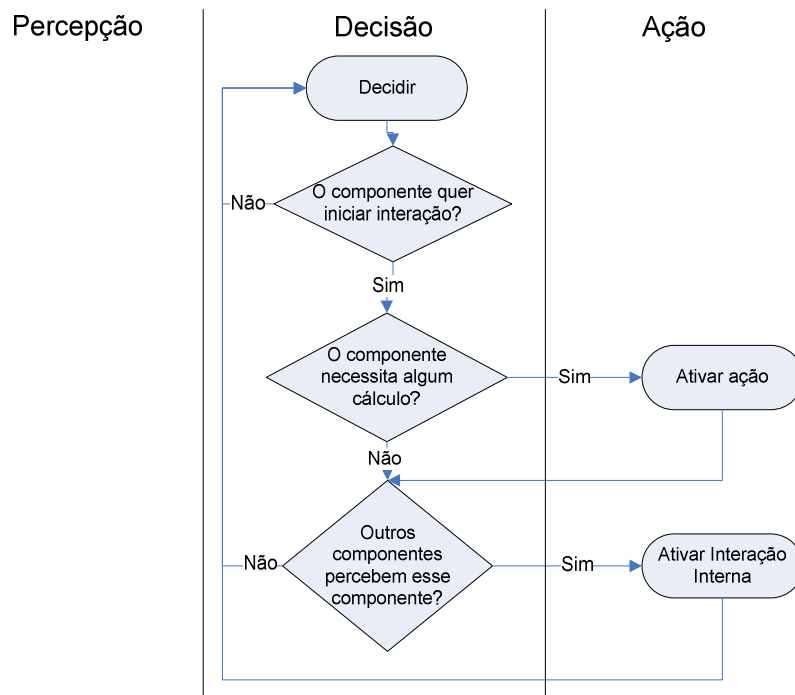


Figura 3.7. Padrão de comportamento imperceptível pró-ativo.

Um exemplo desse tipo de componente pode ser um *software* de simulação que envia mensagens para os objetos no ambiente, atualizando seus estados de acordo com o estado da simulação.

Padrão de Comportamento Imperceptível Pró-ativo e Reativo

Esse padrão inclui os padrões imperceptível pró-ativo e imperceptível reativo. A Figura 3.8 apresenta o padrão.

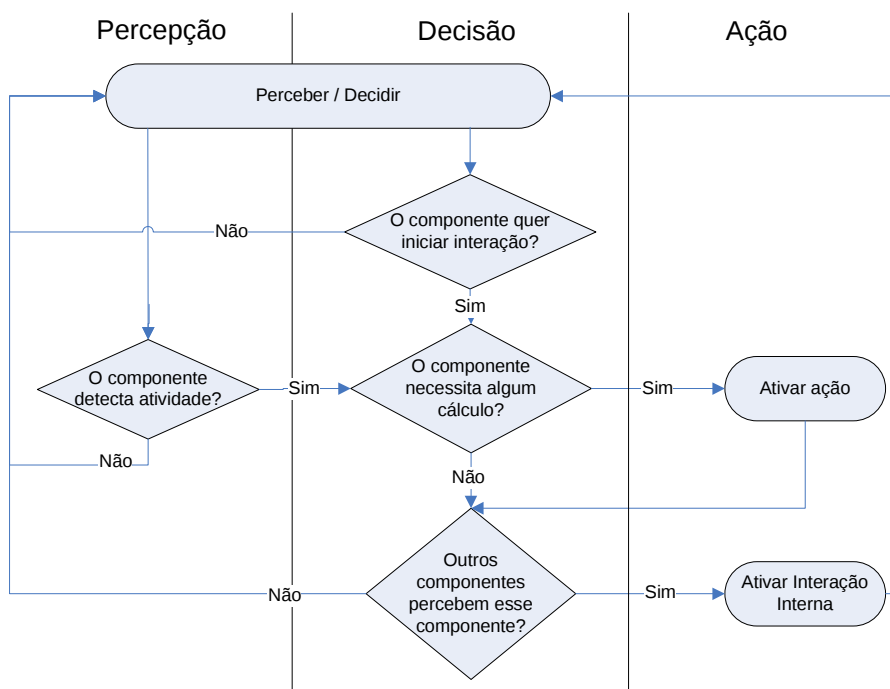


Figura 3.8. Padrão de comportamento imperceptível pró-ativo e reativo.

3.4.2 Categorização

Em SENDA, as entidades são denominadas componentes e representam tanto elementos que são visíveis no ambiente virtual, como elementos utilizados apenas para a manutenção do sistema, como softwares de simulação, de rastreamento e de informação.

As entidades em SENDA são divididas de acordo com sua visibilidade (perceptível e imperceptível) e suas funções de interação (passiva, pró-ativa e reativa). Isso é diferente, por exemplo, da definição de entidades nessa dissertação, que são divididas de acordo com o papel que representam no ambiente virtual (avatar – um usuário, simulóide – um elemento externo não-usuário, mestre – um elemento de controle, objeto – preenchimento do ambiente, artefato – aspecto do ambiente).

Os componentes de SENDA possuem ações de resposta específicas ao usuário, como se as entidades tivessem ações específicas para entidades avatares.

SENDA não faz distinção entre comportamento e interação. Todos os padrões apresentados podem representar um comportamento exclusivo da entidade (quando não ativam mecanismos de Interação – interna ou externa), ou um comportamento de interação. Os comportamentos pró-ativos são aqueles que mais se aproximam da idéia de comportamento exclusivo.

A interação em SENDA é definida para todos os componentes existentes no ambiente virtual. Assim, utiliza a mesma definição dessa dissertação. Os componentes reativos são os que mais se assemelham às ações de interação. Porém, qualquer comportamento pode se envolver em uma interação.

3.5 *Rich Interaction Model*

Desenvolvido pela universidade de Oulu, na Finlândia, o *Rich Interaction Model* (Manninen, 2004) foi baseado numa análise de jogos e nas formas de comunicação não-verbal apresentadas no Capítulo 2. Sua atenção está na conceituação e delimitação das formas de interação disponíveis em CVEs baseados em avatares.

3.5.1 Descrição

Manninen procurou descrever uma série de categorias baseadas na taxonomia de formas de interação encontrada em jogos, e uniu essas categorias com o modelo de comunicação não-verbal. A Tabela 3.1 mostra uma comparação entre os dois modelos estudados por Manninen e o modelo proposto por ele. Na primeira coluna da tabela são apresentadas as categorias encontradas na taxonomia de formas de interação dos jogos. Na segunda, vêm-se as categorias do modelo de comunicação não-verbal. Manninen

criou seu próprio modelo baseado na união desses outros dois, o qual é apresentado na terceira coluna. A taxonomia de formas de interação chama de Linguagem Corporal o que, para o modelo de comunicação não-verbal, são três categorias diferentes: Expressões Faciais, Visão e Cinética. Além disso, não possui as categorias de Olfato e Cronologia encontradas no segundo modelo e possui duas categorias que o outro modelo não possui: Comunicação Baseada em Linguagem e Autônomos / IA.

Tabela 3.1. Modelos de formas de interação e suas categorias

Taxonomia Preliminar de Formas de Interação	Modelo de Formas de Comunicação Não-verbal	Modelo de Interação Rica
Aparência do Avatar	Aparência Física	Aparência do Avatar
Linguagem Corporal	Expressões Faciais	Expressões Faciais
	Visão	Visão
	Cinética	Cinética
Gestos	Detalhes Ambientais	Detalhes Ambientais
Baseado em objeto	Comportamento Espacial	Comportamento Espacial
Modificação do Mundo	Tato	Contato Físico
Contato Físico	Paralinguagem	Áudio Não-verbal
	Olfato	Olfato
	Cronologia	Cronologia
Comunicação Baseada em Linguagem		Comunicação Baseada em Linguagem
Autônomos / IA		Autônomos / IA

Além disso, outras contribuições, como o modelo de interação hierárquico, foram propostas em seu escopo.

Categorias

O modelo de interação rica define várias categorias. A seguir são apresentadas breves descrições dessas categorias, com seus aspectos e particularidades. .

Autônomos / IA & Automação

Esta categoria inclui um conjunto de ações pré-programadas e comportamento reativo que se assemelha com ações subconscientes e intuitivas no mundo físico. Algumas dessas ações são controladas pelo sistema e outras modificadas pelos participantes. Esse grupo de formas de interação envolve-se com a maioria das outras categorias. Contudo, essas ações são consideradas em separado devido à sua importância em termos de projeto e, à natureza específica da maioria das ações.

Exemplos desse grupo são as reações automáticas e as reflexões para um conjunto de situações e critérios, e os altos níveis de abstração no controle da forma de

interação. Essas interações podem ser utilizadas, por exemplo, como meio de transmitir intenções de comunicação:

“A interação entre as escolhas do usuário e os comportamentos comunicativos autônomos permite que os usuários se concentrem no controle de alto nível e locomoção, enquanto depende do avatar para transmitir o sinal comunicativo que representa as intenções comunicativas do usuário”.

(Vilhjálmsón & Cassell, 1998, p. 277)

Nesse caso, o papel das interações autônomas é superar o problema de interface do usuário. Churchill et al. (2001) argumentam que deve ser dado ao computador o controle de atividades mundanas e banais, que não necessariamente necessitem da intervenção total do participante. Deve ser possível escolher o nível de comportamento automático de acordo com a capacidade do terminal ou interface disponível ao participante.

Aparência do Avatar

Define os atributos de imagem e apresentação do usuário. A aparência contém aspectos visuais da apresentação de alguém, tais como, roupas, pele, adornos, cabelo, tamanho.

A utilização da aparência como forma de interação em CVEs extrapola o limite físico do mundo real. Por exemplo, o tamanho e a forma do avatar podem ser alterados pelo usuário e são utilizados para transmitir alguma mensagem. O avatar, que tanto pode ser realista (uma representação aproximada de uma entidade do mundo real) quanto imaginário, deve ser passível de ser modificado ao gosto do usuário.

Cronologia

Essa categoria engloba o uso e a percepção do tempo em interações síncronas. O tempo pode ser utilizado em interações, por exemplo, pausas e velocidade de movimento podem antecipar ações ou chamar a atenção dos usuários para alguma ação.

Observar a passagem do tempo é uma experiência pessoal e depende do que se está fazendo. O ritmo de um jogo é definido por suas ações automáticas e pelas ações dos usuários. Então, formas de interação temporais podem ser vistas como os mecanismos de representação e controle da passagem de eventos, como um relógio tique-taque, limite de tempo, dia e noite, entre outros, e também afetam as atividades e a comunicação interpessoal como, por exemplo, pausas, respostas rápidas e hesitações.

Expressões Faciais

Representam o canal de comunicação não-verbal mais significativa, apesar dessas expressões não serem totalmente similares às do mundo real, pois, geralmente, a face não é bem definida em ambientes virtuais. Alguns ambientes, devido a limitações de resolução gráfica, restringem as opções de expressões faciais.

As expressões faciais mais comuns são obtidas pelas deformações labiais em combinação com as deformações das sobrancelhas. Essas combinações podem expressar alegria, tristeza, raiva, indiferença, dúvida, surpresa, medo, susto, entre outros.

Segundo Thalmann (2001), existem pelo menos quatro meios de se projetar e desenvolver expressões faciais em CVEs: captura da face do participante através de vídeo, códigos baseados em modelos de expressões, síntese do movimento dos lábios na fala, expressões ou animações pré-definidas. Independentemente do meio utilizado, as expressões faciais não podem ser esquecidas.

Detalhes Ambientais

Definem a aparência do ambiente para uma melhor contextualização. Para esse fim, são utilizadas as entidades artefato e objeto. Os detalhes ambientais, tais como limites físicos (paredes), iluminação (sombra) e meio de imersão (ar, água), têm impacto direto na maneira como os participantes se comportam dentro do ambiente virtual.

Geralmente, o participante tem permissão restrita para modificar os detalhes ambientais (acrescentar ou remover elementos).

Cinética

Inclui os movimentos do avatar tais como, movimentos da cabeça, posturas e gestos. É possível utilizar essa categoria em ambientes virtuais, mas ela possui problemas de controle que limitam sua utilização.

Comunicação Baseada em Linguagem

É o maior canal de informação interpessoal da maioria dos CVEs. Pode-se utilizar essa categoria de maneira textual, sonora, por imagens, entre outras.

Áudio Não-verbal

Essa categoria inclui o uso de voz na comunicação, que é referenciada com Paralinguagem. Os aspectos não-verbais da fala possuem códigos prosódicos e paradigmáticos. Os prosódicos estão relacionados à intenção da mensagem (isto é, tempo, tom e intensidade da voz), enquanto que os paradigmáticos são independentes dessa intenção (isto é, qualidade da voz, sotaque, emoção e distúrbios). Ainda pode conter

efeitos de som e músicas de ambiente que podem ser utilizados com manifestações de interação.

“A manipulação da origem do som e sua localização aparente no espaço é um dos mais velhos e poderosos efeitos dramáticos para se elaborar personagens convincentes na mente da audiência que pode vê-los em cena” (Foster & Schneider, 1998).

Visão

Nessa categoria estão os movimentos oculares. O movimento dos olhos e o contato visual dão idéia do foco, direção e duração do olhar em relação aos outros participantes.

Com relação à direção do olhar, existem duas abordagens sobre a atividade de um personagem: ponto-de-vista e ponto-de-ação. O ponto-de-vista reflete a direção do olhar em si, enquanto o ponto-de-ação refere-se ao alvo da ação do personagem. O uso dessa categoria necessita da modelagem do movimento dos olhos.

Olfato

Ainda em desenvolvimento, essa categoria relaciona os efeitos comunicativos de cheiros e odores. Existem limitações de equipamento para o uso dessa categoria. Além disso, o conhecimento dos componentes de odores e o mecanismo do olfato ainda são primitivos.

Contato Físico

Reflete o toque virtual em situações de comunicação. Esta categoria inclui movimentos como aperto-de-mão e “tapinhas”. O contato pode responder a toque, pressão, calor e frio, e dor.

Comportamento Espacial

Comportamento espacial consiste de posição, proximidade, orientação, comportamento territorial e movimento. No mundo real, à medida que alguém se aproxima de uma pessoa, esta apresenta um comportamento de acordo com a relação que possuem.

Modelo Hierárquico

O modelo hierárquico, apresentado no trabalho de Manninen (2004), define camadas de interação. Elas dividem e classificam as ações de interação de tal forma que se cria uma estrutura que vai desde um nível baixo, onde as ações são espécies de sinais, até níveis mais altos, onde os objetivos e metas definem o propósito da interação. Áreas como

Robótica e Inteligência Artificial têm utilizado esse tipo de estrutura (Albus et al., 1985).

O número de possibilidades de interação aumenta com os níveis de interação, aumentando assim os graus de liberdade. Os níveis de interação mais altos combinam as possibilidades dos níveis inferiores. Assim, com um pequeno número de ações básicas é possível construir várias variações e combinações de seqüências de ações. Na Figura 3.9, é apresentada a pirâmide invertida dos níveis do modelo hierárquico, juntamente com um exemplo de sua utilização.

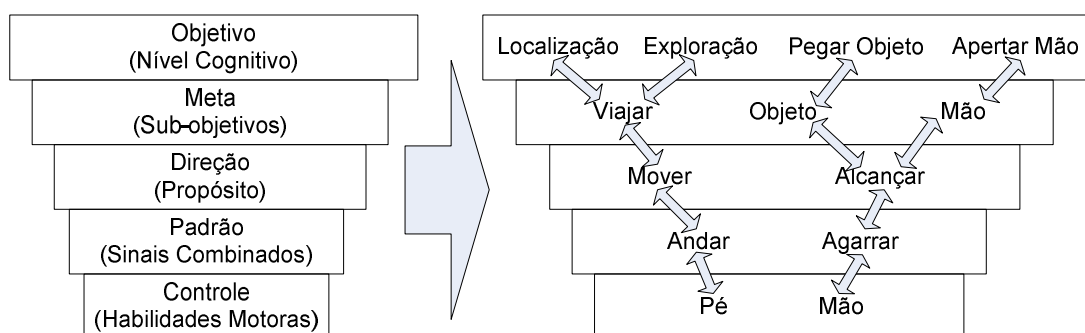


Figura 3.9: Hierarquia de Interação e um exemplo de sua utilização.

3.5.2 Categorização

Como o modelo de interação rica foi criado para expandir as interações entre os usuários e fazê-las da forma mais próxima das interações reais, esse modelo foca entidades avatares. Apesar disso, algumas recomendações são extensíveis a simulóides (como aparência do avatar) e objetos (como automação).

O modelo SENDA foca os comportamentos definidos como interações, porém alguns aspectos, como IA & Autonomia, Cronologia, entre outros, podem contribuir para um melhor desenvolvimento dos demais comportamentos.

As categorias definidas em SENDA apresentam diversos aspectos importantes para a definição e o desenvolvimento das interações das entidades.

3.6 Considerações Finais

Cada um dos modelos apresentados pôde contribuir com suas idéias ao modelo proposto nessa dissertação. Nenhum modelo sozinho foi considerado suficiente para a modelagem de interação e comportamento, mas cada um possui características importantes que foram levadas em consideração no modelo final proposto.

O sistema CORGI contribuiu com a utilização de encapsulamento das ações das entidades, possibilitando assim a inserção de novos tipos sem comprometer o sistema

em si, desde que esse novo tipo utilize-se dos comandos básicos do sistema em seu funcionamento.

MASSIVE contribuiu com a sugestão de utilização de aura, foco, nimbo e consciência. Utilizando auras, esse modelo apresentou a importância de disponibilizar meios de interação diferentes para cada aspecto no qual se deseje interagir, além de utilizar objetos distribuídos para implementar as entidades do ambiente virtual.

O *framework* SENDA indicou a arquitetura de três camadas: percepção, decisão e ação, e fez uma classificação interessante dos tipos de entidades. Também, com seus padrões, ofereceu uma visão do que deve ser levado em consideração no desenvolvimento das entidades.

Por ultimo, *Rich Interaction Model* mostrou os aspectos importantes das interações como também seu caráter comunicativo. Além de contribuir com a idéia de interação hierárquica.

Levando em consideração os aspectos positivos de cada modelo, esta dissertação propõe um modelo diferente dos demais e que procura englobar o maior número de características e possibilidades para ambientes virtuais e jogos. Este modelo é apresentado no Capítulo 4.

Capítulo 4

Modelo de Entidades Comportamentais (BeE Model)

4.1 Introdução

O modelo proposto nesta dissertação, chamado de Modelo de Entidades Comportamentais (*Behavioral Entities Model* - BeE Model), buscou inspiração em alguns modelos importantes encontrados na literatura, descritos no Capítulo anterior, agregando suas características importantes, além de abordar as entidades existentes em ambientes virtuais de uma maneira mais intuitiva, facilitando assim a análise e a especificação do ambiente virtual projetado.

O modelo proposto objetivou apresentar uma base para o desenvolvimento de entidades comportamentais que mudam de estado e reagem a ações realizadas por outras entidades. O modelo pode ser utilizado em sistemas de um único usuário (como jogos *single-player*) ou sistemas distribuídos (como CVEs, jogos *on-line*, entre outros).

Neste Capítulo será apresentada uma abordagem geral de como o modelo funciona e seus principais aspectos. A Seção 4.2 apresenta aspectos relacionados com as entidades de ambiente virtual, o comportamento que elas possuem e a interação que realizam. A Seção 4.3 apresenta alguns cenários que podem ocorrer na utilização do sistema e como eles são tratados. A Seção 4.4 apresenta como o modelo funciona. Finalmente, a Seção 4.5 demonstra a utilização do modelo para o desenvolvimento de uma entidade.

4.2 Definição

Nesta seção serão apresentadas as definições de entidade, comportamento e interação sob a ótica do BeE Model. Esses elementos são fundamentais em um modelo de comportamento, como é o caso do modelo proposto.

4.2.1 Entidades

BeE Model utiliza o conceito de entidades apresentado no Capítulo 2. Os cinco tipos de entidades estão disponíveis:

- ◆ Avatares: usuários do sistema;
- ◆ Simulóides: entidades externas ao sistema que são controladas por computador, geralmente possuindo um mecanismo de decisão associado;

- ◆ **Mestres:** entidades que alteram diretamente outras entidades (podendo ser utilizada em conjunto com outro tipo de entidade);
- ◆ **Objetos:** entidades utilizadas para preencher o ambiente e geralmente são manipuladas por outras entidades, não apresentando mecanismos de decisão; e
- ◆ **Artefatos:** entidades que representam aspectos do ambiente que interagem com outras entidades.

Internamente, as entidades possuem mecanismos de percepção, decisão e ação, como no sistema MASSIVE. A percepção utiliza o conceito de aura apresentado no Capítulo 2, recebendo informações relevantes e analisando se alguma colisão de auras ocorreu. A decisão analisa as informações e decide o que a entidade deve fazer, alterando o seu estado, quando for o caso, e ativando comportamentos; ou disparando comportamentos próprios da entidade. As ações, nesse contexto, são as respostas dadas pela entidade ao ocorrido, ou seja, os comportamentos.

Sempre que uma entidade realiza uma ação passível de desencadear uma interação, as outras entidades devem testar uma possível colisão de auras com essa entidade. Por exemplo, a movimentação de uma entidade no mundo virtual pode gerar uma colisão de uma aura que a acompanhe e, por isso, qualquer tipo de movimentação gera uma informação relevante ao processo de interação. A entidade que causou a ação também deve avaliar uma possível colisão de auras com as outras entidades do sistema.

As entidades possuem uma aura para cada tipo de meio que deseja interagir: aura visual (entidade pode ver ou ser visualizada), aura sonora (entidade pode produzir ou ouvir sons como fala, áudio não-verbal, entre outros), aura de proximidade (perceber entidades que se aproximem ou ser percebida), aura funcional (que representem alguma funcionalidade da entidade), entre outras. É importante que a forma utilizada pela aura possua um cálculo de colisão simples, pois será realizada com muita frequência e em situações que não existe realmente uma interação.

O foco e o nimbo de uma entidade geralmente ficam dentro dos limites de sua aura respectiva, pois serão acionados somente depois que uma colisão de aura acontecer. Eles podem possuir qualquer formato e estar em qualquer lugar, apesar de geralmente se encontrarem ao redor da entidade. A consciência será utilizada pelo processo de decisão da entidade quando reagir à presença da entidade que colidiu.

O processo de decisão avalia situações da entidade, reagindo às ações e possivelmente alterando o estado da entidade. Ele pode ser desde uma simples lista de ações, até um modelo complexo de inteligência artificial. Os estados representam a situação de uma entidade num dado momento: por exemplo, uma abelha pode se encontrar em diversos estados, dependendo do seu propósito, ou seja, voando, atacando alguém, em estado de alerta, entre outros. Os estados podem possuir subestados que especificam ainda mais a situação da entidade: no mesmo exemplo da abelha, ela pode estar no estado “voando” e no subestado “rápido”. Na Seção 2.2.2 foram apresentados tipos de estados que serão utilizados como base para a criação dos estados de cada entidade, ou seja, é importante que cada estado identifique se a entidade existe, está visível e está ativa.

As ações são comandos que devem ser realizados para alterar a entidade devido ao comportamento acionado. Essas ações, representadas por um comportamento, alteram as propriedades da entidade e são acionadas através da uma mudança de estado ou disparadas diretamente por uma entidade, passando sempre pelo processo de decisão.

Ao projetar um ambiente virtual é importante identificar as entidades que este ambiente possui e quais os comportamentos (e interações) que irão desempenhar. O levantamento desses elementos indica: quais eventos relevantes que as entidades devem observar, quais os estados (e subestados quando necessário) que as entidades devem possuir, quais meios de interação (auras) possuem, quão complexo seus mecanismos de decisão devem ser, e quais as ações necessárias aos comportamentos e interações.

4.2.2 Comportamento

Alguns aspectos sobre os comportamentos, que não são considerados interação por envolver apenas uma entidade, são importantes no modelo proposto. Esses comportamentos geralmente reagem a estímulos da própria entidade e dizem respeito somente a ela, apesar de alguns desses comportamentos poderem incentivar uma interação mesmo sem intenção. Por exemplo, uma entidade que resolva sentar-se em um determinado momento, por mais que não almeje atingir nenhuma outra entidade, pode incentivar algumas entidades a aproximarem-se e iniciarem um diálogo. O processo de decisão ativa esses comportamentos que podem ou não alterar o estado da entidade. Esses comportamentos geralmente inserem individualidade nas entidades e conferem uma atitude mais realista.

4.2.3 Interação

O processo de interação começa quando uma entidade realiza uma ação passível de ocorrer essa interação. Nesse caso, as outras entidades analisam se houve alguma colisão de auras com aquela entidade, ou se ela saiu de uma colisão. Se nenhuma colisão ocorreu não houve interação, mas, se uma colisão for detectada, as entidades envolvidas calcularão colisão entre o foco e o nimbo e, a partir daí, calcularão o nível de consciência uma da outra. Com o nível de consciência calculado, cada entidade envolvida utiliza seu processo de decisão para resolver quais ações devem executar para responder àquela interação, podendo nesse processo alterar seu estado.

Em sistemas mais simples, pode-se utilizar um elemento que fique responsável por detectar colisão de auras de todas as entidades e, só quando houver alguma colisão, ativar as entidades envolvidas na interação. Isso facilita o desenvolvimento do sistema e simplifica o processo de interação, evitando que todas as entidades estejam sempre avaliando possíveis interações. Porém, em ambientes virtuais mais complexos como CVEs, isso já não é muito interessante pois o elemento seria sobrecarregado com a quantidade e a complexidades das entidades. É interessante, nesse caso, que cada entidade se responsabilize pela avaliação da colisão de auras e que seja utilizado os conceitos de sistemas distribuídos (apresentado no Capítulo 4) para disponibilizar as entidades como elementos ativos no sistema.

4.3 Cenários

Nesta seção são apresentados alguns cenários de ocorrências possíveis quando da utilização do modelo de entidades comportamentais.

4.3.1 Ação Passível de Interação

Existem algumas ações passíveis de causar uma interação entre duas entidades, como por exemplo, a movimentação de um avatar. Nesses casos, as demais entidades devem avaliar se a ação ocasionou uma interação e a própria entidade deve também avaliar se não gerou uma interação com alguém.

Cada entidade avalia se existe alguma interação sua com outra entidade e trata como deve reagir. Exceto pelas entidades mestres, nenhuma entidade pode alterar o estado ou propriedades de outras, apenas as suas.

4.3.2 Entrada de Nova Entidade

Quando uma entidade nova surge, ela avisa as demais entidades sobre seu estado inicial. A entrada de uma entidade é uma ação passível de causar interação.

Entidades podem ter como ação própria criar novas entidades. Por exemplo, uma entidade “flor” pode criar entidades “pólen” quando for o momento de realizar uma polinização.

4.3.3 Comportamento Individual

Os comportamentos individuais de uma entidade são disparados por seus sistemas de decisão e geralmente não causam interação. Ocasionalmente podem existir entidades que observam outras entidades e que podem iniciar uma interação com elas caso um comportamento individual seja observado. Nesse caso, a entidade observadora irá iniciar a interação e não a entidade que gerou o comportamento individual.

4.3.4 Entrada, Permanência e Saída em uma Aura

A análise de colisão de auras deve ser feita não somente quando uma entidade entra em uma aura, mas também quando ela sai ou permanece na aura. Cada entidade sabe quais entidades entraram, saíram e permaneceram em suas auras e pode reagir diferentemente para cada uma dessas ações. Essa informação de movimento da entidade é incluída na consciência que passa a ter dois valores vinculados a ela: valor de consciência e tipo de movimentação que a entidade realizou.

4.3.5 Alteração de Aura, Foco ou Nimbo

Uma entidade pode alterar a forma, tamanho e posição de suas auras, nimbos e focos. Essa alteração pode ser feita devido a uma escolha pessoal da entidade, uma ação resultante de uma interação, ou uma ação de uma entidade mestre.

A escolha pessoal geralmente é feita por avatares, ou seja, o usuário decide alterar sua aura. Entretanto, o mecanismo de decisão de qualquer entidade pode fazê-lo. Uma ação pode alterar a aura como resultado de uma interação: por exemplo, uma ação que torne a entidade surda, como uma explosão, altera a aura auditiva daquela entidade ou, uma interação com um objeto como um microfone aumenta sua aura sonora. Entidades mestres podem interceder em uma entidade alterando sua aura, como por exemplo, uma entidade controladora, como um superusuário, que resolve retirar a permissão de falar de uma entidade que violou alguma regra.

4.3.6 Perca de Consciência

Caso uma entidade percebida afaste seu nimbo do foco de uma entidade que possui consciência dela, de tal forma que não exista mais colisão entre eles, a entidade que foca perderá a consciência dela. Apenas quando o nimbo da entidade voltar a colidir com o foco é que a consciência retornará.

4.3.7 Interação entre Três ou mais Entidades

Cada entidade recebe uma ação passível de interação e avalia como ela deve reagir àquela ação. Assim, quando três ou mais entidades desejam realizar uma interação, cada entidade decide suas ações pessoais, não interferindo nas outras entidades. A única exceção é para as entidades mestres, mas esses casos devem ser projetados com bastante cuidado para evitar ações indevidas.

4.4 BeE Model

Esta seção apresenta a estrutura do modelo proposto nessa dissertação. O modelo de entidades comportamentais é constituído de classes que definem como os comportamentos de entidades de ambientes virtuais devem ocorrer. O modelo foi desenvolvido na linguagem de programação C++, utilizando-se de estruturas de dados disponíveis na STL². A escolha da linguagem deu-se por sua flexibilidade, por ser utilizada na grande maioria dos projetos relacionados às áreas de realidade virtual e jogos, e por ser facilmente utilizada em diferentes plataformas.

Cada classe está localizada em um arquivo de mesmo nome. Os arquivos *Globals.h* e *Globalsx.h* definem tipos de dados utilizados pelo modelo. Esses arquivos podem ser alterados caso necessário, mas recomenda-se isso apenas se for extremamente necessário, a fim de evitar erros e funcionamentos indevidos.

Como se trata de um modelo, algumas classes são abstratas, representando as diretivas que a classe descendente deve seguir, e possuindo as assinaturas dos métodos que devem ser desenvolvidos. As classes estão divididas em dois grupos: núcleo e extensões.

4.4.1 Núcleo

O núcleo consiste no conjunto de classes principais do BeE Model que especificam todas as características necessárias ao comportamento de entidades. A Figura 4.1 mostra

² STL – *Standard Template Library* – é uma biblioteca de *software* que faz parte do C++ *Standard Library* e que possui algoritmos genéricos e estruturas de dados de uso geral. Foram concebidos com a idéia de uma programação genérica e uma abstração sem perda de eficiência.

respectivamente). Na classe *Reason*, encontram-se os estados e os comportamentos possíveis a uma entidade, representados respectivamente pelas classes *State* e *Behaviors*: *State* representa cada estado individualmente e *Behaviors* representa todos os comportamentos da entidade. Na classe *Perception*, encontra-se a classe *EntitiesAwareness* que é responsável por calcular e guardar a consciência que a entidade tem das outras que realizam interação com ela (a consciência é representada pela classe *Awareness*). Em *Perception* encontram-se também as auras (classe *Aura*) e, nelas, o foco (classe *Focus*) e o nimbo (classe *Nimbus*). Esses três últimos elementos possuem uma forma espacial representada pela classe *Shape*. Maiores detalhes sobre todas essas classes do modelo proposto serão apresentados nas Seções seguintes.

Entity

A classe *Entity* representa, de maneira genérica, uma entidade. Ela é abstrata e, portanto, não pode ser utilizada para criar instâncias. Para isso, podem ser utilizadas as cinco subclasses de *Entity* que representam os tipos de entidade que pode existir em um ambiente de realidade virtual: *AvatarEntity* (avatares), *ObjectEntity* (objetos), *ArtefactEntity* (artefatos), *SimuloidEntity* (simulóides), e *MasterEntity* (mestres).

Entity possui diversos métodos de manipulação do mecanismo de percepção, como a inserção e remoção de auras, e do mecanismo de decisão, como a inserção e remoção de estados. Possui também um método chamado *aurasCollide* que deve ser chamado sempre que uma ação passível de interação ocorrer (embora esse método não seja implementado nessa classe e sim na classe *Perception*).

Na construção de qualquer uma das cinco classes que representam uma entidade, deve-se informar qual o mecanismo de decisão escolhido para aquela entidade (classe *Reason*).

Reason

A classe *Reason* representa o mecanismo de decisão da entidade. É uma classe abstrata e, portanto, não possui realmente os métodos de decisão. Essa classe define como uma classe que deseje realizar o processo de decisão de uma entidade deve se comportar. Na seção de extensões, será apresentada uma classe que se propõe a desenvolver a classe *Reason* utilizando para a decisão uma máquina de estados finitos.

Esta classe possui os estados e os comportamentos de uma entidade. Possui todos os métodos necessários à manutenção dos estados que são reconhecidos pela

classe através de identificadores (recomenda-se o uso de uma enumeração a fim de tornar simples o seu reconhecimento). Os comportamentos devem ser definidos em uma classe que estenda a classe *Behaviors*.

Reason possui um método abstrato chamado *reason* que define a assinatura do método que irá realizar o processo de decisão. Para esse método, devem ser informadas a entidade e a aura que ocasionaram a interação.

State

Essa classe define o estado que uma entidade pode assumir. Como apresentado na Seção 2.2.2, o estado possui informação sobre sua existência, visibilidade e atividade, e pode ainda possuir subestados vinculados a ele. Os estados e subestados são referenciados através de seus identificadores (id).

Behaviors

A classe *Behaviors* representa os comportamentos que uma entidade pode realizar. Cada comportamento possui uma lista de ações que devem ser realizadas quando for ativado. Essa classe é abstrata e, portanto, deve-se desenvolver uma subclasse que possua os métodos que representam os comportamentos da entidade.

Em sua construção, deve-se informar o objeto externo ao modelo que representa a entidade no sistema de realidade virtual. Essa informação é guardada no atributo *externalEntity* e será utilizada pelos comportamentos para alterar a entidade. Podem-se utilizar *scripts* para representar as ações das entidades, a fim de tornar a manutenção das mesmas mais simples e eficiente.

O método abstrato *execute* deve ser desenvolvido na subclasse que estende a classe *Behaviors*, para que selecione um comportamento utilizando o identificador do comportamento desejado.

A fim de possibilitar a passagem de parâmetros do processo de decisão para o método que representa o comportamento, o método *execute* possui um número variável de parâmetros e deve resgatar os parâmetros para passá-los aos comportamentos. Um exemplo de como a passagem variável de parâmetros funciona e como tratá-la encontra-se na documentação do modelo.

Perception

Essa classe representa o mecanismo de percepção da entidade e possui as auras da entidade. Só é possível existir uma aura de cada tipo, sendo por isso as auras identificadas por seu tipo e não por um identificador. No arquivo *Globals.h*, são especificados alguns tipos de auras comumente utilizados, porém esses tipos podem ser expandidos facilmente.

A classe possui métodos de gerenciamento de auras, e um método chamado *auraCollision* que deve ser chamado sempre que uma ação passível de ocasionar uma interação ocorrer.

Aura

A classe *Aura* representa uma aura da entidade e possui o foco e o nimbo da entidade para aquela aura. Além disso, possui uma forma representada pela classe *Shape*. A aura pode ser desabilitada alterando-se o valor do atributo *enabled*, deixando assim de ser testada quando algum evento passível de interação ocorrer.

Essa classe possui um método *collide* que testa a colisão da aura com outra invocando o método *collide* de *Shape*. Caso seja detectada colisão entre as auras, o método *collide* do foco é chamado.

Na criação de *Aura*, devem ser informados: a forma da aura, do foco e do nimbo, como também o cálculo de consciência que o foco utilizará (classe *EntitiesAwareness*).

Focus

Representa o foco da entidade para uma determinada aura. Assim como aura, possui uma forma (representada pela classe *Shape*) e pode ser desabilitado através do atributo *enabled*. Na sua criação, a classe *Aura* informa qual o cálculo de consciência que o foco utilizará (armazenado no atributo *entitiesAwareness*).

Nimbus

Representa o nimbo da entidade em uma determinada aura. Assim como aura, possui uma forma (representada pela classe *Shape*) e pode ser desabilitado através do atributo *enabled*.

Shape

Representa a forma que a aura, o foco ou o nimbo podem possuir. Essa classe é abstrata e deve ser desenvolvida de acordo com o tipo de forma geométrica que o desenvolvedor necessite em seu sistema e que esteja disponível no sistema gráfico utilizado. *Shape* possui um ponteiro para o objeto que representa a forma no sistema gráfico (*externalShape*).

O método *collide* representa o teste de colisão entre duas formas e, possivelmente, fornece informação para o cálculo da consciência entre as entidades, uma vez que o valor de consciência está intimamente ligado ao volume colidido. Deve-se utilizar o sistema de colisão disponível no sistema de realidade virtual, ou desenvolver um sistema de colisão para os tipos de formas desejadas no ambiente virtual. O método *collide* vai utilizar *externalShape* e as rotinas de colisão do sistema gráfico para calcular a colisão.

EntitiesAwareness

Esta classe guarda a consciência da entidade sobre todas as outras que interagiram com ela e possui o método de cálculo da consciência. Ela é abstrata e deve ser desenvolvida por uma subclasse utilizando uma forma de cálculo para o valor de consciência. Ela também guarda informações de qual movimento a entidade que está interagindo realizou: entrando, movendo-se dentro, ou saindo da aura.

Para o cálculo de consciência, recomenda-se o uso do valor do volume colidido para esse cálculo, mas cálculos mais simples podem ser utilizados. Na Seção 4.4.2, será apresentada uma classe que desenvolve a maneira mais simples de cálculo de consciência.

Awareness

Essa classe representa a consciência que o foco de entidade possui sobre o nimbo de outra entidade. Possui o valor da consciência (representado pelo atributo *value*) e também o tipo de movimento que a entidade percebida realizou no foco da entidade que percebe, podendo ser: de entrada, de saída, ou de permanência (representado pelo atributo *movement*).

4.4.2 Extensões

O BeE *Model* disponibiliza também um conjunto de classes que expandem as classes do núcleo, disponibilizando assim uma maneira, mesmo que simples, de utilizar o modelo. A Figura 4.2, mostra a relação das classes estendidas com as classes do núcleo. As classes em vermelho representam as classes abstratas que devem ser estendidas e desenvolvidas pelas entidades do sistema; as classes em amarelo representam extensões de outras classes; os elementos em cinza representam constantes utilizadas no modelo e sugeridas para a utilização do modelo no sistema; e as classes em cinza claro representam as classes do núcleo que as extensões fazem referência. As classes estendidas estão disponíveis em um pacote (*namespace* em C++) chamado *beex*.

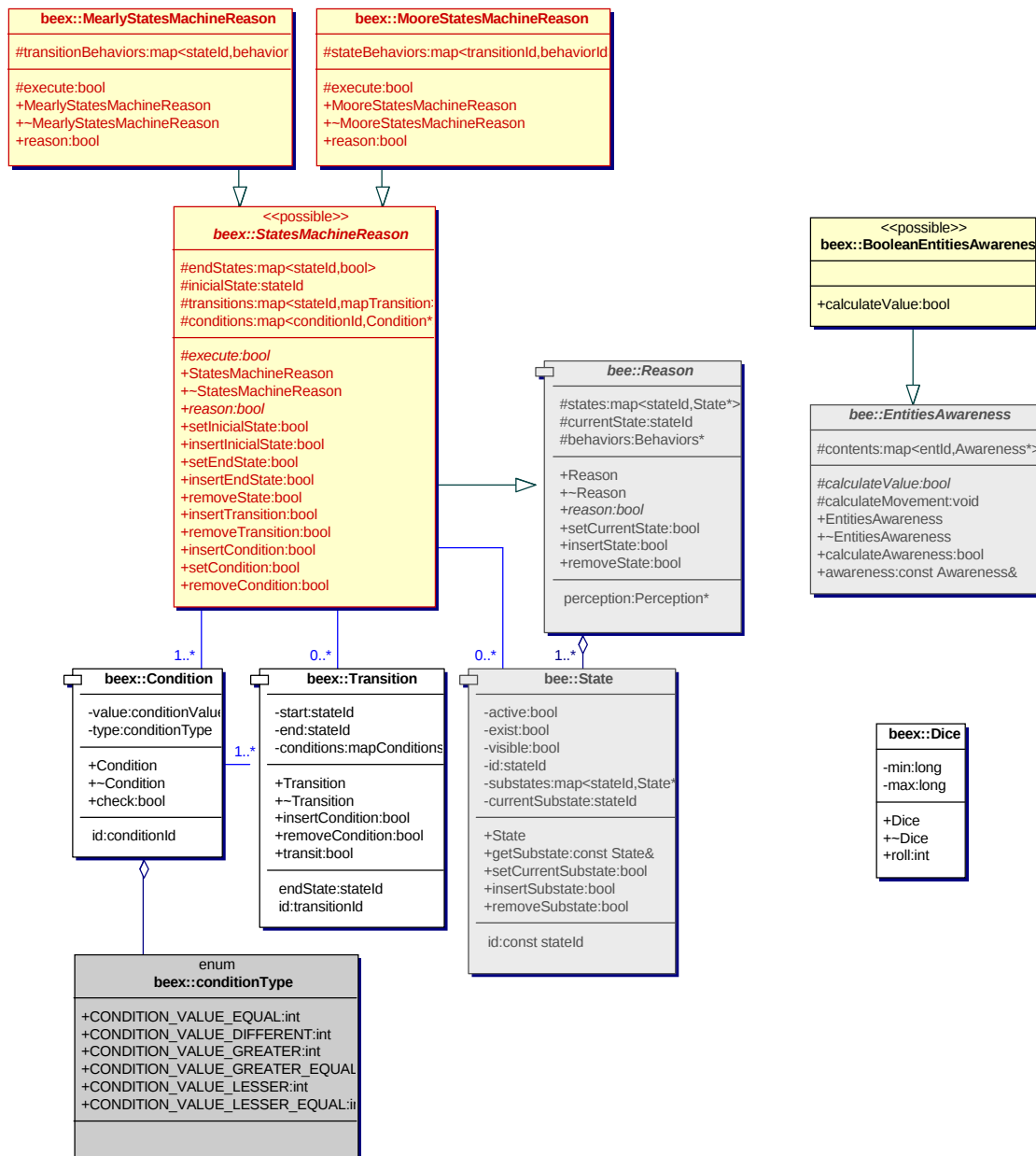


Figura 4.2: Diagrama de Classes das extensões do modelo.

StatesMachineReason

Essa classe estende a classe *Reason* utilizando, para o mecanismo de decisão, uma máquina de estados finitos (a ser apresentada na Seção 5.4). Possui as transições que a máquina pode realizar, as condições para que cada transição ocorra, e os estados inicial e finais utilizados na máquina de estados.

Mesmo sendo uma subclasse de *Reason*, essa classe continua sendo abstrata, pois existem dois tipos de máquinas de estados: máquinas de Mearly e máquina de Moore. Essas máquinas também estão disponíveis nas extensões.

MearlyStatesMachineReason

Subclasse de *StatesMachineReason*, essa classe representa uma máquina de Mearly que realiza as ações relativas a mudança de estado da máquina em suas transições. Esse tipo de máquina de estados finitos é apresentado na Seção 5.4 .

MooreStatesMachineReason

Subclasse de *StatesMachineReason*, essa classe representa uma máquina de Moore que realiza as ações relativas a mudança de estado da máquina em seus estados. Esse tipo de máquina de estados finitos é apresentado na Seção 5.4 e um exemplo de sua utilização, na Seção 4.5.2 .

BooleanEntitiesAwareness

Subclasse de *EntitiesAwareness*, essa classe representa a maneira mais simples de se calcular a consciência entre duas entidades. Quando houver colisão entre o foco e o nimbo de duas entidades, a entidade que possui o foco terá consciência total da outra entidade; caso a colisão não exista, a entidade que possui o foco não terá nenhuma consciência.

Dice

A classe *Dice* representa um dado eletrônico para fornecer valores randômicos. Sua implementação é bastante simples e ela gera números inteiros entre um intervalo definido na sua criação.

4.5 Exemplo de Utilização

Esta seção apresenta os passos necessários para se definir uma entidade de um ambiente virtual utilizando o BeE *Model*. Como exemplo, a entidade criada será uma abelha. A ordem dos passos pode mudar de acordo com o desenvolvedor, mas alguns passos devem ser realizados antes de outros. O bom-senso do desenvolvedor vai definir qual a sequência de passos a serem realizados e quais passos são realmente necessários.

4.5.1 Comportamentos e Estados

Inicialmente, define-se, em linhas gerais, como a entidade deve se comportar. Para o exemplo, deseja-se que a entidade abelha possua comportamentos simples: que ela possa pousar em uma determinada posição ou iniciar um voo em momentos aleatórios, que fique em alerta caso alguma entidade aproxime-se dela, e que ela ataque o alvo mais próximo caso sinta-se ameaçada.

Depois, utilizando uma estratégia como a apresentada no Anexo I, identificam-se quais comportamentos a entidade deve possuir, bem como os estados em que pode se encontrar. Na Figura 4.3, é mostrado o gráfico final gerado pela estratégia do Anexo I para a entidade do exemplo, apresentando os comportamentos que a entidade deve possuir e seus estados.

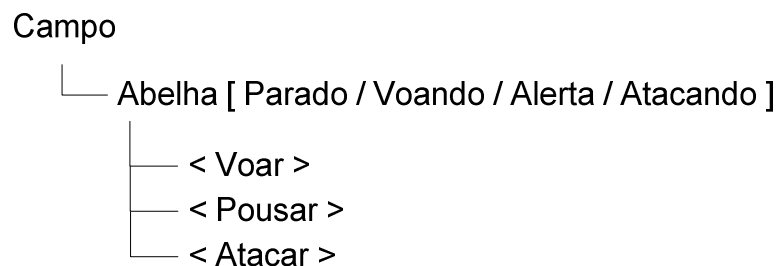


Figura 4.3: Comportamentos e Estados da entidade abelha.

4.5.2 Mecanismo de Decisão

É necessário, então, decidir qual mecanismo de decisão será utilizado pela entidade. No exemplo apresentado, será utilizada uma máquina de estados finitos com ativação das ações nos estados (Máquina de Moore) disponível nas extensões do Modelo de Entidades Comportamentais.

4.5.3 Auras, Focos e Nimbos

Uma vez que já foi apresentado o que a entidade deve fazer, agora se definem os tipos de auras disponíveis na entidade, com seu respectivo foco e nimbo. Definem-se também a forma que aura, foco e nimbo devem possuir.

O *BeE Model* não possui nenhum tipo de forma disponível, pois essas formas dependem do sistema gráfico utilizado. Para o exemplo, utilizaremos uma forma esférica de um sistema gráfico fictício chamado *ExternalSphereShape*, e desenvolveremos uma extensão da classe *Shape* (chamada *SphereShape*) para essa forma. *SphereShape* aponta para o tipo correto do sistema gráfico (*ExternalSphereShape*) e desenvolve o método *collide*, que é utilizado pelo *BeE Model* para testar colisão, utilizando o método de mesmo nome disponível na forma esférica.

No exemplo, a entidade possui uma aura de aproximação para interagir com entidades que se aproximarem dela.

4.5.4 Consciência

Uma vez que as auras, e seus respectivos focos e nimbos, foram definidos, deve-se definir também qual o cálculo de consciência que vai ser utilizado. Esse cálculo pode estar ligado ao tipo de forma que está disponível a entidade, pois se podem utilizar informações de intercessão entre as formas.

A expansão do modelo apresenta um tipo simples de cálculo de consciência representado pela classe *BooleanEntitiesAwareness*.

4.5.5 Diagrama de Estados

Uma vez que os estados possíveis já foram identificados, o mecanismo de decisão escolhido e as auras definidas, é necessário agora desenvolver o diagrama de estados da entidade baseado nos estados detectados e no tipo de mecanismo de decisão utilizado. Nele, informam-se as possíveis transições entre os estados.

Na Figura 4.4, é apresentado o diagrama de estados da entidade abelha com as transições entre os estados.

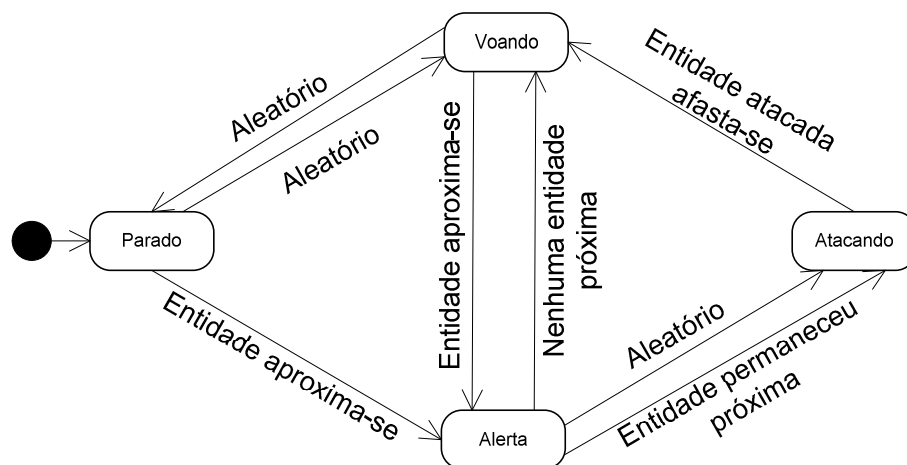


Figura 4.4: Diagrama de estados da entidade abelha.

Uma vez que todas as transições foram definidas, é preciso identificar as condições que devem ser satisfeitas para que a transição possa ocorrer. Na Tabela 4.1 são apresentadas as condições necessárias para que cada transição da entidade exemplo possa ocorrer.

Tabela 4.1. Condições para a realização de cada transição da entidade abelha

Transição	Condições
Aleatório	Valor randômico.
Entidade aproxima-se	Colisão de aura de proximidade. Movimento de entrada da entidade na aura de proximidade. Contador de alerta diferente de zero.
Nenhuma entidade próxima	Movimento de saída da entidade na aura de proximidade. Não existir mais nenhuma entidade próxima. Contador de alerta igual a zero.
Entidade permaneceu próxima	Colisão de aura de proximidade. Movimento de permanência da entidade na aura de proximidade.
Entidade atacada afasta-se	Movimento de saída da entidade na aura de proximidade. Contador de alerta igual a zero.

4.5.6 Especificação dos Comportamentos

Por último, é necessário definir as ações que cada comportamento deve executar. Para fazer isso, é necessário estender a classe *Behaviors* do modelo, criando métodos que representam os comportamentos e implementando o método abstrato *execute*, que ativa o método correto para cada comportamento.

No exemplo, nenhum dos comportamentos necessita de parâmetros e, por isso, não foi necessário tratar a passagem variável de parâmetros.

4.6 Considerações Finais

Esta dissertação utilizou os conceitos de entidades, comportamento e interação, juntamente com as características mais importantes dos modelos analisados, para criar um modelo que fosse: simples de utilizar, genérico suficiente para ser incorporado em sistemas de realidade virtual diferentes, capaz de possibilitar a inserção de novas funcionalidades e capaz de deixar o desenvolvimento de entidades de realidade virtual o mais intuitivo possível.

O modelo é independente do mecanismo de decisão, possibilitando a utilização de diferentes técnicas. Utiliza o conceito de aura, foco e nimbo para caracterizar a ocorrência de interação, como também, o cálculo de consciência. É independente da forma como o cálculo da consciência é realizado, disponibilizando informação de

movimentação da entidade além do valor da consciência. Independe do cálculo de colisão, podendo utilizar a biblioteca disponível no sistema gráfico. E, por fim, abstrai os comportamentos, podendo utilizar métodos específicos, ou *scripts*.

Também está disponível no BeE *Model* um conjunto de tecnologias alternativas a fim de possibilitar sua utilização: um mecanismo de decisão baseado em máquina de estado finito (possuindo dois tipos de máquinas: Moore e Mearly) e um cálculo de consciência simples.

Para o exemplo de entidade dado, foi necessário criar quatro classes para que a utilização do modelo fosse possível: *Bee* (subclasse de *SimuloidEntity*) representando a entidade, *BeeMooreSMReason* (subclasse de *Reason*) que define o método de decisão, *BeeBehaviors* (subclasse de *Behaviors*) que implementa os comportamentos da entidade, e *SphereShape* (subclasse de *Shape*) que representa a forma esférica da aura, foco e nimbo. Além disso, foram definidas quatro enumerações representando os identificadores das condições, das transições, dos estados e dos comportamentos para tornar mais claro a utilização desses identificadores. Apenas uma aura foi necessária, e ela utiliza o cálculo de consciência disponível nas extensões do modelo.

Com o exemplo foi possível apresentar como desenvolver uma entidade para ser utilizada no modelo e assim, apresentar uma pequena noção do trabalho que um desenvolvedor terá ao utilizar o modelo. Esse trabalho, em comparação com o trabalho de se desenvolver uma estrutura de controle de interação e comportamento, é bastante simples e possui tarefas bem definidas. Assim, o desenvolvedor poderá dedicar-se a definição minuciosa da das entidades, abstraindo as preocupações de controle que o modelo proporciona. Com isso, entidades cada vez mais complexas e ricas podem ser desenvolvidas utilizando a mesma base de interação.

Capítulo 5

Tecnologias

5.1 Introdução

Neste capítulo são apresentadas tecnologias que podem ser utilizadas para desenvolver alguns conceitos apresentados no Modelo de Entidades Comportamentais (*BeE Model*) desta dissertação.

O modelo define, de maneira abstrata, tanto o funcionamento do comportamento de entidades em um ambiente de realidade virtual, quanto o modo como os componentes do modelo interagem entre si. O modelo proposto é suficientemente genérico para poder se adaptar a diferentes sistemas de realidade virtual. No entanto, é interessante que sejam apresentadas algumas tecnologias a fim de nortear o leitor que ainda não possui um sistema de realidade virtual elaborado.

Na Seção 5.2 , são apresentados o conceito de objetos distribuídos e algumas das arquiteturas que utilizam tal conceito. Na Seção 5.3 , é apresentado o conceito de linguagem de *scripts*, e suas vantagens e desvantagens são apontadas. Na Seção 5.4 , conceitos de máquina de estado finito são apresentados.

5.2 Objetos Distribuídos

Objetos distribuídos são componentes de um sistema distribuído. Cada objeto é autônomo e oferece seus serviços ao sistema através de interfaces que especificam como esses serviços podem ser solicitados pelos demais objetos do sistema. Os objetos distribuídos não precisam ser homogêneos. Eles podem funcionar em sistemas operacionais diferentes e até serem desenvolvidos em linguagens de programação diversas. Podem existir também componentes que são utilizados exclusivamente por um único usuário (eles são conhecidos como componentes não-compartilhados).

Como consequência da autonomia dos componentes, os sistemas distribuídos executam-nos concorrentemente, e por essa razão, geralmente, há um grande número de processos simultâneos no sistema. Os sistemas distribuídos, freqüentemente, são multitarefa, e criam novas tarefas (*threads*) ao iniciar um serviço para um usuário ou para outro componente, evitando que o componente prestador do serviço seja bloqueado. Além disso, os processos de um sistema distribuído não precisam ser

executados no mesmo processador, envolvendo assim comunicação entre máquinas através de uma rede.

Em sistemas distribuídos, se um componente está sobrecarregado por causa do número excessivo de solicitações por parte dos usuários ou dos outros componentes, um novo componente capaz de realizar o mesmo serviço pode ser inserido no sistema para ajudar a aliviar a carga do componente sobrecarregado.

Uma plataforma distribuída provê um mecanismo para estabelecer contato entre objetos, enquanto negocia o uso de interfaces e a solicitação de serviços. Esse conceito é chamado de *trading* (ISO/IEC, 1991). Assim, um objeto provedor de serviços (*provider*) exporta suas interfaces, registrando-as em um objeto *trader*. O objeto *trader* registra o tipo de cada interface e também o contexto relacionado ao serviço. Dessa maneira, quando um objeto *customer* deseja usar uma interface, ele examina o *trader*, verificando tanto o tipo de interface desejada como também o contexto designado. O *trader*, então, procura uma associação, e, se ela existir, retorna uma referência de interface para o *customer*. Essa referência de interface pode ser então usada pelo *customer* para solicitar serviços ao objeto *provider*. Os conceitos de objetos, interface, serviços e *trading* são resumidos na Figura 5.1.

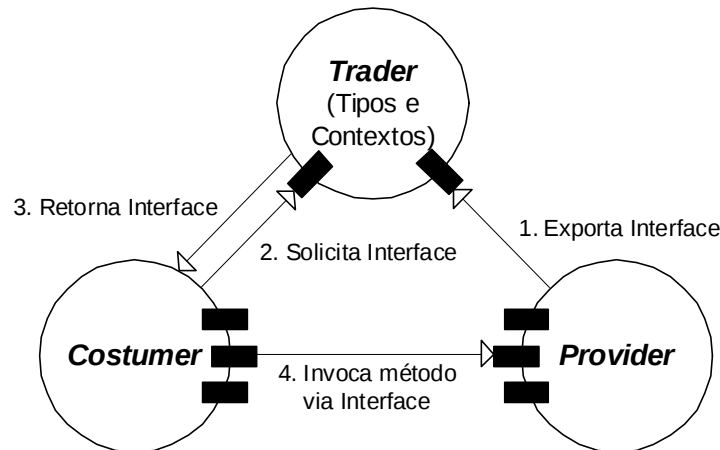


Figura 5.1: Trading.

Embora os sistemas distribuídos apresentem uma série de vantagens, eles podem falhar por diversas razões: falha por mau funcionamento de um componente; falha de rede ou, falha por demora de um componente em responder a uma solicitação do sistema.

Há diversas plataformas de desenvolvimento de sistemas distribuídos. Aquelas mais comumente utilizadas são apresentadas a seguir.

5.2.1 CORBA

CORBA – *Common Object Request Broker* (OMG, 1997) – é uma especificação desenvolvida pelo *Object Management Group* (OMG), e tem a finalidade de proporcionar interoperabilidade entre elementos de sistemas distribuídos. O objetivo do grupo foi definir uma arquitetura que permitisse a comunicação entre ambientes heterogêneos, utilizando a idéia de objetos, independente de quem projetou os dois elementos da aplicação distribuída. Por se tratar de uma arquitetura, diversas implementações foram desenvolvidas por diferentes empresas de *software*, entre elas, duas se destacam: Visibroker (Borland, 1994), desenvolvido pela Borland e disponível para C++ Builder, Delphi e JBuilder; e omniORB 2 (Apasphere, 2002), desenvolvido atualmente pela Apasphere e disponível para C++ e Python.

Em CORBA, um objeto que deseje interagir com outro objeto distribuído não necessita conhecer a localização real do alvo, e sim sua identificação ou referência. Para tornar ainda mais flexível esse modelo de localização, essas referências de objetos podem ser associadas a nomes, de maneira que um objeto possa ser encontrado a partir de seu nome (único) no sistema. Para disponibilizar esse recurso, *middlewares* baseados na arquitetura CORBA fornecem o chamado Serviço de Nomeação (*Naming Service*) que é um mecanismo para publicação de referências de objetos, possibilitando a identificação e a localização de objetos no sistema a partir de seus nomes (Balen, 2000). Logo, com o uso de um serviço de nomeação, é possível desenvolver um sistema sem a necessidade de conhecer previamente as localizações na rede dos diversos componentes, bastando referenciá-los pelos seus nomes.

5.2.2 SOAP

Segundo Davis & Parashar (2002), SOAP (*Simple Object Access Protocol*) é um protocolo leve para troca de informação em sistemas distribuídos e descentralizados. É baseado em XML e consiste de três partes: um envelope que descreve o que pode existir em uma mensagem e como processá-la; um conjunto de regras de codificação para expressar instâncias de tipos de dados definidos pela aplicação; e uma convenção para representar chamada de procedimentos remotos e suas respostas. Embora SOAP não especifique um mecanismo de transporte, a maioria de suas implementações utiliza HTTP.

Duas das muitas implementações SOAP disponíveis são: o Apache SOAP (1999), e o XSOAP (2002).

5.3 Linguagens de Script

Desde os primeiros sistemas operacionais, os *scripts* têm sido grandes aliados dos usuários de computadores por servirem como uma forma rápida e segura de agilizar tarefas muito repetitivas, aumentando a produtividade. Um *script* é uma seqüência de comandos eventualmente repetidos seguidamente (Nunes & Pereira, 1999). Segundo Webopédia (2004), *script* é uma lista de comandos que pode ser executada sem a interação do usuário, e uma linguagem de *script* é uma linguagem de programação com a qual a escrita de *scripts* é possível. O objetivo desse tipo de linguagem é combinar componentes desenvolvidos em outras linguagens, e por essa razão, são também chamadas de linguagens de colagem (*glue languages*) ou de linguagens de integração de sistemas (*system integration languages*).

As linguagens de *script* têm a filosofia de desenvolver sistemas de *software* a partir de conjuntos de bibliotecas pré-definidas. Programar nessas linguagens significa definir a maneira como os objetos devem interagir entre si, e não o processo de especificação desses objetos. Conseqüentemente, a tarefa de programar torna-se mais independente das características específicas do hardware e mais dependente da formulação dos próprios problemas (Nunes & Pereira, 1999).

5.3.1 Características das Linguagens de Script

As linguagens de *script* possuem características que lhes conferem robustez e tornam seu uso aconselhável. Algumas dessas características são apresentadas a seguir.

Tipagem Dinâmica

As linguagens de *script* são de tipagem dinâmica, isto é, não existem declarações explícitas de variáveis, nem construção de estruturas de dados com tipos pré-definidos. Ao invés disso, elas possuem algumas estruturas de dados polimórficas auto-adaptáveis que reconhecem as estruturas de dados a elas referenciadas (Nunes & Pereira, 1999). Esse recurso evita que certos valores sejam tratados com tipos incorretos, facilitando a adaptação dos programas às diversas formas de dados que podem ser manipulados. A verificação dos tipos é feita em tempo de execução. Erros de tipos em tempo de execução podem ocorrer em qualquer linguagem, com tipagem dinâmica ou não.

Também dessa tipagem dinâmica vem o conceito de passagem de parâmetros por falta, que é a possibilidade de ocultarem-se parâmetros de entrada em componentes. Dessa maneira, valores padrões serão atribuídos a parâmetros não informados. Por

exemplo, o valor padrão da cor de um botão pode ser cinza, e por tanto, a menos que se deseje uma cor diferente, esse valor não precisa ser informado quando da criação desse elemento.

Interpretação

As linguagens de *script* são linguagens interpretadas, isto é, durante a execução do programa o interpretador da linguagem avalia o código-fonte e executa as operações nele descritas. Assim, ao contrário das linguagens compiladas, um programa executável não é gerado.

As linguagens interpretadas em geral têm uma eficiência muito menor do que as linguagens compiladas tradicionais. Entretanto, a linguagem interpretada permite a auto-referência, o que possibilita a criação de partes do código-fonte de uma aplicação durante sua execução. O PTUI – *Python Tkinter User Interface* (Roadhouse, 1998) – é um editor de scripts na linguagem Python (PSF, 1991), que utiliza auto-referência. O PTUI é uma aplicação escrita em Python, e, assim, através do próprio interpretador que o executa, avalia as aplicações enquanto elas estão sendo escritas, elaborando esquemas gráficos sobre a hierarquia dos objetos que estão sendo definidos. Quando o editor executa o programa, ele incorpora o novo código-fonte a si mesmo, executando conjuntamente com ferramentas de *debugging*, ao invés de criar um novo processo com o programa que está sendo escrito.

Outra característica positiva das linguagens interpretadas é a facilidade de localização e correção dos erros, já que a mensagem de erro emitida pelo interpretador está associada ao trecho de código que está sendo interpretado naquele instante. Essa característica tem um impacto direto na redução do tempo gasto com testes e correções da aplicação.

Reutilização de Código

A produtividade no desenvolvimento de aplicações está intimamente relacionada ao reaproveitamento de código, que, por sua vez, é uma consequência direta da qualidade da especificação dos processos. Quando, devido a falhas de engenharia de software, a qualidade dessas especificações é baixa, não há como evitar a necessidade de grandes esforços na normalização de dados e na digitação de longos trechos de programa repetidas vezes. As linguagens de *scripts* por sua vez utilizam seus tipos auto-adaptáveis

para simplificar a normalização de dados, encorajando a reutilização de código-fonte (Nunes & Pereira, 1999).

Essa utilização de recursos já disponíveis inclui aqueles que estão em outros formatos ou aqueles que foram desenvolvidos em uma outra linguagem. Por exemplo, a linguagem de *script* Python não possui um módulo gráfico, entretanto a linguagem Tcl (Ousterhout, 1988), também de *script*, tem um módulo bastante portátil (chamado Tk) que serve para criar aplicativos que utilizem recursos de GUIs (*Graphical User Interface*). Assim, através da reutilização de código, é possível desenvolver aplicações em Python com recursos gráficos.

Portabilidade

O desenvolvimento das linguagens de *script* foi um passo importante na direção da portabilidade. Algumas dessas linguagens têm interpretadores escritos na linguagem Java, assim é possível que os mesmos programas que executem em um interpretador feito para um sistema operacional possam funcionar como um *applet*. Isto pode simplificar consideravelmente a construção de softwares para ambientes cooperativos envolvendo diferentes plataformas.

Software Livre

A maioria das linguagens de *script* em desenvolvimento utiliza um regime de comunidades *open-source*. Isto significa que são desenvolvidas por um grande grupo de voluntários, espalhados pelo mundo, e estão disponíveis gratuitamente através da internet, contando também com uma vasta documentação e um suporte informal baseado em listas de discussões.

Gerência Automática de Memória

As linguagens de *script* geralmente gerenciam automaticamente a memória alocada, utilizando para isso estratégias como: coleta de lixo (*garbage collection*, como nas linguagens Java, Lua e Ruby) e contadores de referências (como na linguagem Python).

Essa característica tanto simplifica a programação, pois o desenvolvedor não precisa lidar com o problema de gerência de memória; quanto evita o surgimento de erros comuns como, por exemplo, o vazamento de memória.

Facilidade para Estruturação de Dados

As linguagens de *script* geralmente oferecem uma implementação padrão de diversos tipos e estruturas de dados, como por exemplo, cadeias de caracteres, números complexos, listas, tabelas, conjunto, entre outros. Dessa maneira, o desenvolvedor não precisa preocupar-se com tais tipos e estruturas, podendo reutilizar as construções padrões que a linguagem oferece. Isso simplifica a definição de interfaces de classes e tipos de dados customizados. A utilização dessas estruturas é facilitada ainda mais devido à gerência automática de memória que essas linguagens realizam.

Segurança

A semântica completa dessas linguagens, aliada às características de interpretação, tipagem dinâmica e gerência automática de memória, contribui para que sejam linguagens seguras. Nessas linguagens, os erros e exceções em trechos de código dificilmente podem ser explorados como falhas de segurança. Em particular, a interpretação de código impede a invasão de regiões de memória, pois a mesma não é endereçada diretamente pelo código interpretado.

5.3.2 Exemplos de linguagens de Script

Existem diversas linguagens de *script* disponíveis. A seguir serão apresentadas algumas das mais conhecidas e utilizadas.

Python

Python (PSF, 1991) é uma linguagem de programação de altíssimo nível, criada pelo holandês Guido Van Rossum sob o ideal de “Programação para todos”. Esse ideal norteou seu desenvolvimento, garantindo: liberdade (gratuita e de código aberto), disponibilidade (disponível para as plataformas Windows, Linux, Mac, Palm, entre outras) e clareza de sintaxe.

Python é uma linguagem orientada a objetos, o que facilita o controle da estabilidade dos projetos de grandes proporções. Com essa linguagem, também pode ser utilizado o paradigma procedimental. Ela é uma linguagem altamente modular, permitindo o aproveitamento de funcionalidades já desenvolvidas (Labaki, 2003).

Lua

Desenvolvida pelo TecGraf (PUC-Rio), a linguagem Lua (Tecgraf, 2004) é uma linguagem de programação poderosa e leve, projetada para estender aplicações, ou seja,

pode ser acoplada a programas maiores que precisem ler e executar programas escritos pelos usuários. Lua é uma linguagem embutida, com sintaxe semelhante ao Pascal, mas com construções modernas, como funções anônimas (inspiradas no paradigma funcional) e poderosos construtores de dados. Isso faz com que Lua seja uma linguagem de grande expressão com uma sintaxe familiar.

Lua é utilizada atualmente pela indústria de jogos. Dois dentre os vários exemplos de jogos que utilizam Lua estão: o Baldur's Gate (Bioware & Interplay, 1998), e o FarCry (Crytek & Ubisoft, 2004).

5.4 Máquina de Estado Finito

Uma máquina de estado finito (*Finite State Machine* - FSM) é um sistema que possui um número limitado de estados. Um exemplo real pode ser um interruptor de luz que possui dois estados: ligado e desligado. Qualquer sistema que possui um número limitado de possibilidades, definidas por estados, pode ser representado por uma máquina de estado finito (Howland, 1999).

Máquina de estado finito é uma técnica de Inteligência Artificial que se originou na Matemática e na Teoria da Computação e que inicialmente foi utilizada para representação de linguagens. É intimamente ligada a outras técnicas fundamentais de representação de conhecimento, como as Redes Semânticas e sua extensão chamada de Espaço de Estado (Brownlee, 2000).

Segundo Brownlee (2000), as FSM são compostas de quatro elementos principais: os estados, que definem comportamentos e podem produzir ações; as transições de estado, que são movimentos de um estado para outro; as regras ou condições, que precisam ser conhecidas e atendidas para que uma transição de estado possa ocorrer; e os eventos de entrada, que são gerados externamente ou internamente, e que podem ativar regras que conduzam a transições.

A Figura 5.2 mostra uma representação gráfica de uma máquina de estado finito. Os estados estão representados pelos retângulos de cantos arredondados, o círculo preto simples informa o inicial, os círculos pretos duplos informam os estados finais, as setas representam as transições e os nomes que acompanham as setas representam as condições para aquela transição ocorrer.

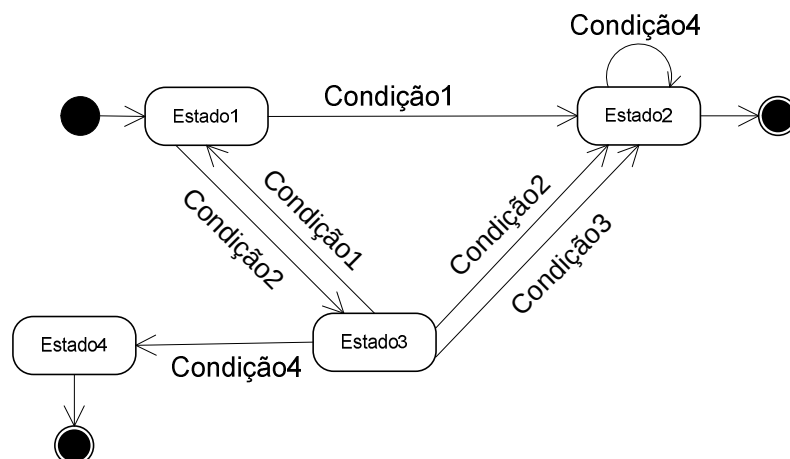


Figura 5.2: Representação gráfica de uma máquina de estado finito.

Uma máquina de estado pode possuir uma transição que leve ao mesmo estado original; pode possuir mais de uma transição entre os mesmos estados, desde que as condições de ocorrência das transições sejam diferentes, ou uma transição seja de saída e outra de entrada; pode também possuir mais de um estado final; e pode utilizar uma mesma condição em transições diferentes. Ela não pode possuir duas transições saindo de um mesmo estado e com as mesmas condições, e não pode possuir mais de um estado inicial.

FSM é tipicamente usada como um sistema de controle onde o conhecimento é representado nos estados e as ações são controladas por regras. Como qualquer sistema baseado em regras, se todos os antecedentes de um nó são verdadeiros, então a regra é ativada. É possível que múltiplas regras sejam disparadas ocorrendo um conjunto de conflitos. Somente uma transição por vez pode ser ativada, sendo exigida uma estratégia de resolução de conflitos consistente para selecionar uma única regra de ativação que executará uma transição de estado.

Existem dois tipos principais de máquina de estado: a determinística e a não-determinística. A máquina determinística garante que, dado uma entrada e um estado atual, a transição pode ser sempre prevista. Já a não-determinística não permite prever a transição. As máquinas determinísticas são preferidas, pois são mais simples de desenvolver.

As ações de uma FSM podem ser geradas de duas maneiras diferentes: são produtos dos estados alcançados (Máquina de Moore) ou são produtos das próprias transições (Máquina de Mearly). A Máquina de Moore relaciona as ações com os estados, não importando qual transição foi utilizada. A Máquina de Mearly permite que

transições diferentes executem ações diferentes, mesmo que essas transições saiam de um mesmo estado inicial e alcancem o mesmo estado final.

Brownlee (2000) apresenta algumas vantagens e desvantagens na utilização de FSM. As vantagens são: simplicidade, previsibilidade, rapidez e baixo processamento, além de serem simples e rápidas de se projetar e desenvolver. Sua previsibilidade (em máquinas determinísticas) permite testar facilmente sua execução e, como apenas o processamento referente ao estado atual é executado, o processamento é reduzido.

As FSMs também apresentam algumas desvantagens: previsibilidade e adaptação. E ainda existem domínios onde o comportamento do sistema não pode ser decomposto em estados separados, com condições bem definidas para as transições.

5.5 Considerações Finais

Neste capítulo, buscou-se mostrar as tecnologias possíveis de serem utilizadas no desenvolvimento do BeE *Model*. A gama de opções é muito maior do que as apresentadas, no entanto, estas podem ser consideradas uma boa amostra do todo, cabendo ao desenvolvedor escolher as tecnologias que melhor se adaptem ao seu sistema específico. O BeE *Model* não se limita a tais tecnologias, pois sua concepção teórica e sua especificação lógica se abstraem o máximo possível dos meios utilizados para implementá-lo.

A tecnologia de objetos distribuídos é uma excelente opção para a especificação e o desenvolvimento das entidades, uma vez que seus comportamentos podem ser expressos através de serviços. Assim, as entidades ficam transparentes ao sistema, podendo ser inseridas e removidas facilmente. Entretanto, a latência de execução para solicitação de um serviço, somada à latência gerada pelo resto do sistema, pode inviabilizar a interatividade e o sentimento de presença dentro do sistema.

Linguagens de *scripts* são fortes candidatas na especificação e execução das ações das entidades (comportamentos), tornando a inserção, alteração e remoção de tais ações mais simples, uma vez que apenas os arquivos de *scripts* deverão ser alterados.

Máquina de estado finito é uma ótima escolha para o mecanismo de decisão das entidades, por ser bastante rica e de simples desenvolvimento. Métodos de inteligência artificial podem se caracterizar como uma boa opção também para o mecanismo de decisão, embora não tenham sido abordados neste capítulo.

Capítulo 6

Caso de Uso: Conservação de Alimentos

6.1 Introdução

Para uma observação prática da aplicação do BeE *Model*, foi escolhido um caso de uso dividido em um sistema de realidade virtual onde o modelo pudesse ser inserido, bem como uma aplicação prática que usasse o sistema e necessitasse de uma interação mais rica entre as entidades presentes. O sistema utilizado, chamado CRAbCVE, é um sistema de realidade semi-imersiva (que utiliza computadores comuns e uma conexão simples de internet) desenvolvido na Universidade Federal do Ceará. Já a aplicação prática escolhida está contida no contexto do projeto SINTEGRA.

Uma explicação mais detalhada sobre a arquitetura CRAbCVE é apresentada na Seção 6.2, e o projeto SINTEGRA definido na Seção 6.3. Na Seção 6.4, o caso de uso é mostrado, especificando as alterações que foram necessárias na arquitetura, bem como o que foi definido na aplicação prática. No final, algumas considerações são desenvolvidas na última seção.

6.2 Arquitetura CRAbCVE

A arquitetura CRAbCVE (Gomes, 2005) foi projetada para viabilizar ambientes virtuais colaborativos de baixo custo (utilizando computadores comuns) através da internet. É uma evolução da arquitetura Ataxia (Leite-Júnior, 2000) que inclui elementos para permitir o trabalho colaborativo.

Para o desenvolvimento da arquitetura CRAbCVE, as funcionalidades dos elementos da Ataxia foram redefinidas, um novo componente foi adicionado e um novo protocolo de comunicação foi especificado. Essas modificações visam: prover suporte aos processos de colaboração, permitir a publicação de mundos virtuais e garantir uma maior acessibilidade. A maior acessibilidade é garantida pela CRAbCVE através de um novo componente que permite ao usuário ter conhecimento de todos os CVEs disponíveis no sistema, funcionando como um portal de acesso, além de ter o cadastro de todos os usuários do sistema. Na Figura 6.1, são mostrado os componentes da arquitetura CRAbCVE e a relação entre eles.

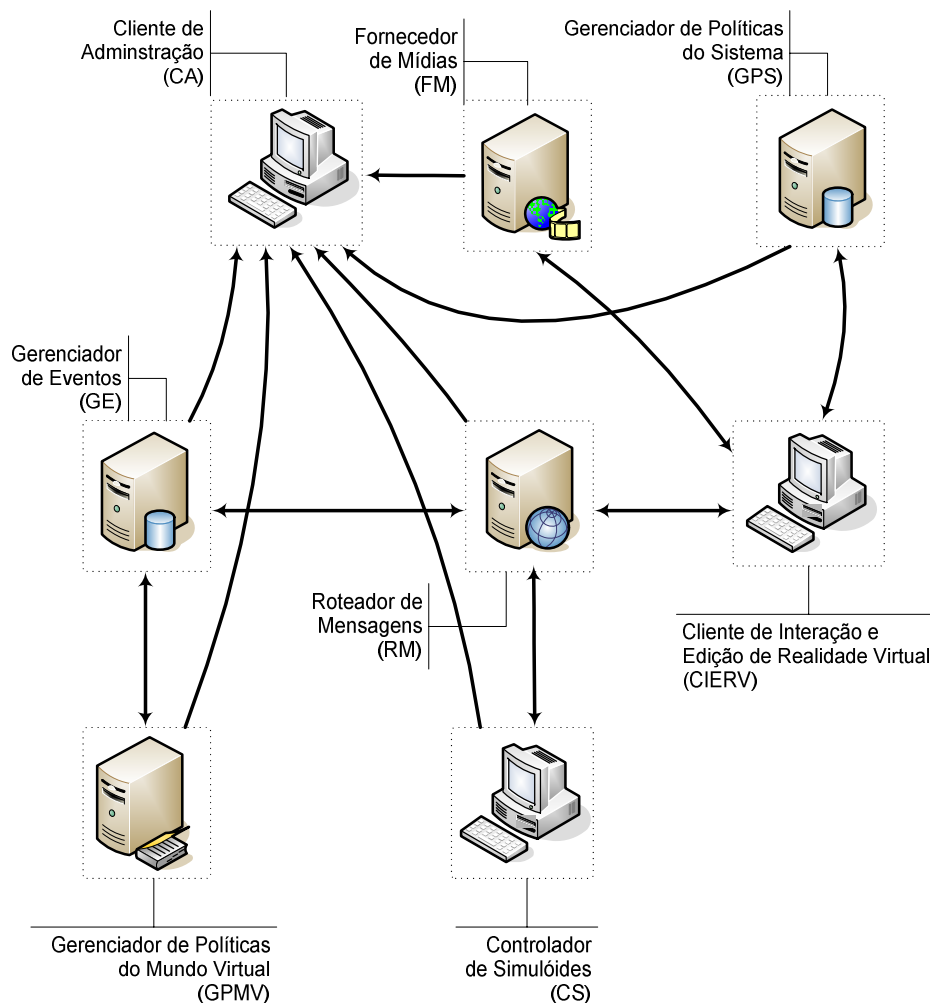


Figura 6.1: Componentes da arquitetura CRAbCVE.

Gerenciador de Políticas do Sistema (GPS)

Para ter acesso aos CVEs, todos os participantes precisam passar antes pelo GPS, que pode ser considerado um componente centralizador. Existe uma única instância desse componente no sistema, ao contrário dos outros componentes que podem ser instanciados várias vezes. O GPS apresenta três importantes funcionalidades: registro de usuários e entidades, controle automático de instâncias de CVEs e localização de CVEs.

Cliente de Interação e Edição em Realidade Virtual (CIERV)

Os recursos do ambiente virtual estão disponíveis para os usuários através deste cliente que se conecta diretamente ao Roteador de Mensagens. Essa aplicação cliente faz uso de uma interface gráfica tridimensional para a realização de todo o processo de geração e apresentação de um mundo virtual.

Roteador de Mensagens (RM)

É o componente responsável por tornar as conexões de um determinado CVE disponíveis às diversas entidades. Esse componente é empregado no repasse de informações a outros elementos da arquitetura CRAbCVE, a fim de manter o estado compartilhado do CVE e de viabilizar a colaboração entre participantes.

Gerenciador de Eventos (GE)

É responsável por iniciar o funcionamento do ambiente virtual e por tratar eventos oriundos das interações de seus participantes e da edição de mundos virtuais, gerando mensagens de resposta sempre que necessário. Com esse fim, o GE armazena os estados de cada uma das entidades do CVE, as definições de portais³ e nascedouros⁴ para interconexão de ambientes, e também as informações referentes às entidades.

Gerenciador de Políticas do Mundo Virtual (GPMV)

Este componente é especializado no armazenamento das restrições de ações e acesso de participantes no CVE. O conjunto de direitos de acesso permitirá ou não que um usuário interaja com determinadas entidades do CVE, ou tenha acesso a determinados subambientes. Juntamente com o Roteador de Mensagens e o Gerenciador de Eventos, o GMPV é um dos três componentes essenciais para o funcionamento de um CVE.

Fornecedor de Mídias (FM)

Torna disponível aos participantes do CVE o conjunto de arquivos utilizados em interações no ambiente. O FM é bastante simples, funcionando basicamente como um sistema de armazenamento de arquivos de mídias. Entre essas mídias estão presentes todos os elementos tridimensionais do ambiente virtual incluindo: sua estrutura arquitetônica, imagens, sons, vídeos, entre outros. O FM armazena ainda arquivos que descrevem a formação dos mundos virtuais e das entidades, e listas de permissões das entidades em cada mundo virtual.

Controlador de Simulóide (CS)

Este componente permite a existência de entidades controladas por computador como os simulóides. Ele conecta-se diretamente ao RM e, através dele, pode enviar e receber informações para todos os demais participantes do CVE.

³ Portais são ligações de um mundo virtual para outro, facilitando assim o traslado entre eles.

⁴ Nascedouros são posições no mundo virtual onde objetos ou personagens virtuais podem aparecer.

Cliente de Administração (CA)

Atua conectando-se diretamente aos outros componentes da arquitetura, podendo tanto realizar alterações nas configurações do componente, como acessar as informações neles armazenadas, a fim de realizar análise de funcionamento do sistema.

6.3 SINTEGRA

O projeto SINTEGRA (Sistema Integrado de Educação a Distância com Suporte a Avaliação de Competências) tem como objetivo principal o desenvolvimento de um ambiente computacional integrado para aprendizagem a distância através da *Web* (Sintegra, 2005). Um importante diferencial desse sistema é a inclusão de serviços de avaliação de competências de participantes, uma necessidade premente no processo de capacitação de pessoal para diversas áreas de trabalho e que vem sendo enfrentado pelo Serviço Nacional de Aprendizagem Comercial (SENAC) e, em particular, o SENAC/CE. Esses serviços merecem destaque especial por estabelecerem novas considerações sobre modelos de educação a distância. O sistema proposto também faz uso de recursos multiusuários de realidade virtual voltados para treinamento, somados às ferramentas de educação a distância baseadas na *Web*. Esses recursos, atuais objetos de estudo da Universidade Federal do Ceará (outro parceiro do Projeto SINTEGRA) adequam-se perfeitamente ao ensino profissionalizante praticado pelo SENAC/CE.

Para atingir o objetivo proposto, respeitando as peculiaridades do sistema, serão criadas metodologias pedagógicas específicas, que contemplem o ensino a distância através da internet e de modelos baseados em realidade virtual. Da mesma forma, também será necessária a criação de uma metodologia especial para a geração de conteúdos direcionados a esse tipo de sistema. O desenvolvimento de tais metodologias, abordando inclusive modelos teóricos para avaliação de competências, também é um dos objetivos do projeto.

O projeto propõe a geração de três ferramentas voltadas para a simplificação e a otimização do uso do sistema de realidade virtual. A primeira delas consiste em um sistema de autoria de mundos virtuais voltada para a criação de ambientes de realidade virtual por alunos e professores; a segunda ferramenta é um simulador de ambientes para gerenciar o comportamento dos diversos participantes do modelo de realidade virtual abrangendo, inclusive, interações com objetos e possíveis reações a determinados estímulos (esta ferramenta é essencial para o realismo das simulações propostas aos participantes dos cursos); e a terceira, um gerenciador de cargas, visa

controlar o fluxo de dados transmitidos na rede de comunicação empregada, modificando configurações e, caso seja necessário, incluindo novos computadores para o correto balanceamento do poder computacional envolvido. Para avaliar o sistema proposto, serão especificados e desenvolvidos dois módulos pertencentes a um curso que será aplicado a turmas de alunos do SENAC/CE.

O curso Boas Práticas na Manipulação de Alimentos é o curso proposto pelo SINTEGRA e será desenvolvido completamente a distância, perfazendo um total de 20 horas/aula. Destina-se ao aperfeiçoamento de profissionais ou estudantes de Nutrição, Engenharia de Alimentos e Economia Doméstica que trabalham com a manipulação de alimentos. Uma amostra do curso pode ser vista na Figura 6.2.

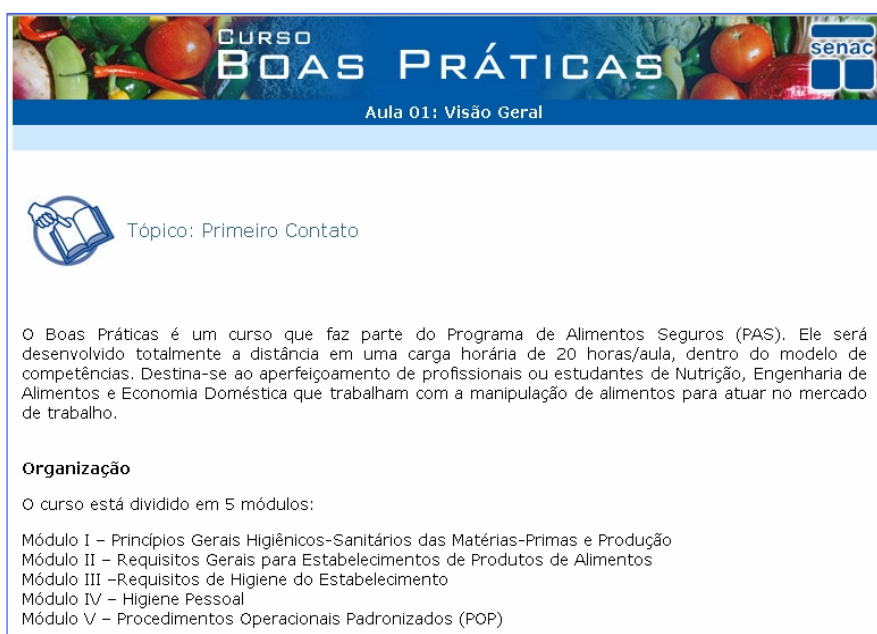


Figura 6.2: Página Web do curso Boas Práticas na Manipulação de Alimentos.

Este curso estabelece um conjunto de princípios e regras para o correto manuseio de alimentos, abrangendo desde as matérias-primas até o produto final, de forma a garantir a saúde do consumidor. Ele é realizado através da metodologia de educação a distância, envolvendo as seguintes estratégias com o objetivo de gerar colaboração: leitura de textos *on-line*, análise de situações-problema, estudo de casos, dinâmicas de grupo, fóruns de discussão, formação de comunidade virtual, pesquisa orientada, e projeto.

A avaliação será feita por meio da observação do desenvolvimento de competências ao longo do curso. Serão observados os seguintes critérios: participação no ambiente virtual, utilização das ferramentas, colaboração, criatividade, clareza e assiduidade na entrega das tarefas. Às atividades, serão atribuídos valores quantitativos

e qualitativos expressos em: tarefas corretas; tarefas concluídas no prazo; resultados para competências: desenvolvida (D), não desenvolvida (N) e parcialmente desenvolvida (P); e conceitos: muito bom (MB), bom (B), regular (R) e insuficiente (I). Para os resultados de competência parcialmente e não desenvolvidas, serão propostas correções e atividades complementares com o objetivo de atingir a competência desenvolvida, buscando dar um caráter formativo à avaliação.

O curso é ministrado através do SOLAR – Sistema *Online* de Aprendizagem (SOLAR, 2005) – que é um ambiente virtual de aprendizagem desenvolvido pelo Instituto UFC Virtual (UFC Virtual, 2005), da Universidade Federal do Ceará. Ele é orientado ao professor e ao aluno, possibilitando a publicação de cursos e a interação com esses cursos. O SOLAR foi desenvolvido para potencializar o aprendizado a partir da relação com a própria interface gráfica do ambiente. Essa interface permite que o usuário tenha rapidez no acesso às páginas e ao conteúdo, além de oferecer fácil navegação e compatibilidade com diversos navegadores do mercado. Na Figura 6.3, é apresentado o ambiente SOLAR.



Figura 6.3. Ambiente SOLAR.

Dentre as aulas do curso de Boas Práticas na Manipulação de Alimentos, destaca-se, para este estudo, a lição de recepção e armazenagem de alimentos que aborda os procedimentos necessários ao processo de recebimento e armazenagem de um variado conjunto de alimentos advindos de um ou mais fornecedores. Tais conhecimentos permitirão ao aluno saber como separar os alimentos recebidos em categorias e armazená-los em estoques de maneira correta. O aluno, após participar de

aulas sobre cuidados com o processo de contaminação dos alimentos e o armazenamento dos mesmos, irá treinar sua competência em um ambiente de realidade virtual que representa um restaurante.

A lição inicia com a chegada dos alimentos em um caminhão. Os alunos, em equipes de dois participantes, devem recepcionar os alimentos, avaliando suas condições e separando para retorno os que possuem características de recusa como: prazo de validade vencido, produto amassado, embalagem rasgada, em condições de conservação inadequadas, entre outras. Em seguida, os alunos devem separar os alimentos e armazená-los de acordo com sua categoria: refrigerados, congelados ou secos. O armazenamento possui certas regras que devem ser seguidas pelos alunos, como por exemplo, os produtos devem ser ordenados por prazo de validade, ou seja, os produtos com validade mais próximas de seu vencimento devem encontrar-se na frente dos que possuem um prazo maior; certos produtos não devem encontrar-se próximos para não se alterarem; entre outras.

Ao final do processo, um relatório, com a atuação dos participantes, é emitido por um supervisor virtual. Caso haja algum problema no processo, este supervisor informa a equipe participante e a mesma poderá percorrer novamente os espaços virtuais utilizados na avaliação para tentar perceber o que aconteceu de errado. As dicas sobre as condições de falha dos alunos serão dadas pelos atores Garçom, Cliente ou Cozinheiro, que farão observações sobre algum alimento proveniente do setor de armazenamento. Após o aluno tentar perceber seu erro, ele deverá se comunicar com seu professor para discutir o exercício e poderá então repetir todo o processo, tendo sido modificados os alimentos recebidos e as possíveis problemáticas encontradas.

6.4 Caso de Uso

Para o estudo da utilização do BeE *Model*, a arquitetura CRAbCVE deverá ser alterada para suportar o modelo. Essa alteração ocorrerá em apenas dois dos componentes da arquitetura e não irá causar um impacto grande em suas funcionalidades. No contexto da aula de recepção e armazenagem de alimentos, serão escolhidas e simplificadas algumas entidades para facilitar o desenvolvimento do caso de uso. Isso, porém, não interfere na validação do modelo uma vez que a complexidade das entidades está ligada a sua especificação, ocasionando alteração em seus mecanismos de decisão, na quantidade das auras utilizadas, no cálculo de consciência e na especificação dos comportamentos e não na maneira como o modelo trabalha.

6.4.1 CRAbCVE

As alterações necessárias na arquitetura do CRAbCVE para a integração do modelo foram: a inserção do conjunto de classes do BeE *Model* no Cliente de Interação e Edição em Realidade Virtual, e a inserção da funcionalidade de gerenciamento de entidades no Gerenciador de Eventos.

Cliente de Interação e Edição em Realidade Virtual

O Cliente possui agora uma lista de entidades que estão presentes no ambiente virtual. Caso alguma ação passível de interação ocorra, ele deve testar alguma colisão de auras com as entidades do ambiente e quando alguma mensagem de ativação de comportamento chegue, ele deve executar o comportamento daquela entidade localmente. Na Figura 6.4, é apresentado o Cliente com um usuário manipulando um alimento.

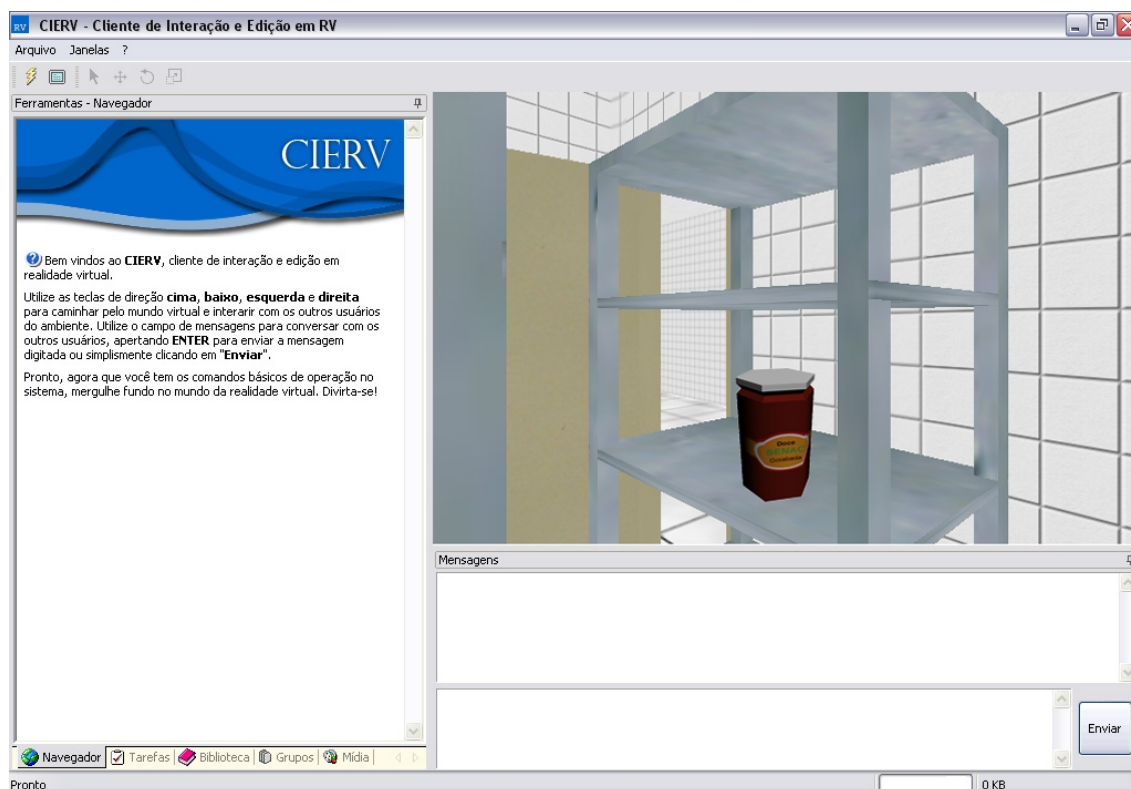


Figura 6.4: Cliente de Interação e Edição em Realidade Virtual do CRAbCVE.

Gerenciador de Eventos

O Gerenciador de Eventos fica observando os acontecimentos que ocorrem no ambiente e, quando alguma ação passível de ocorre uma interação for executada, ele aciona o teste de colisão de auras das entidades que gerencia. Caso alguma interação ocorra, ele enviará uma mensagem de ativação do comportamento que aquela entidade executou

como resposta à interação para todas as entidades do sistema. Na Figura 6.5, é apresentado o Gerenciador de Eventos com algumas entidades que ele está administrando.

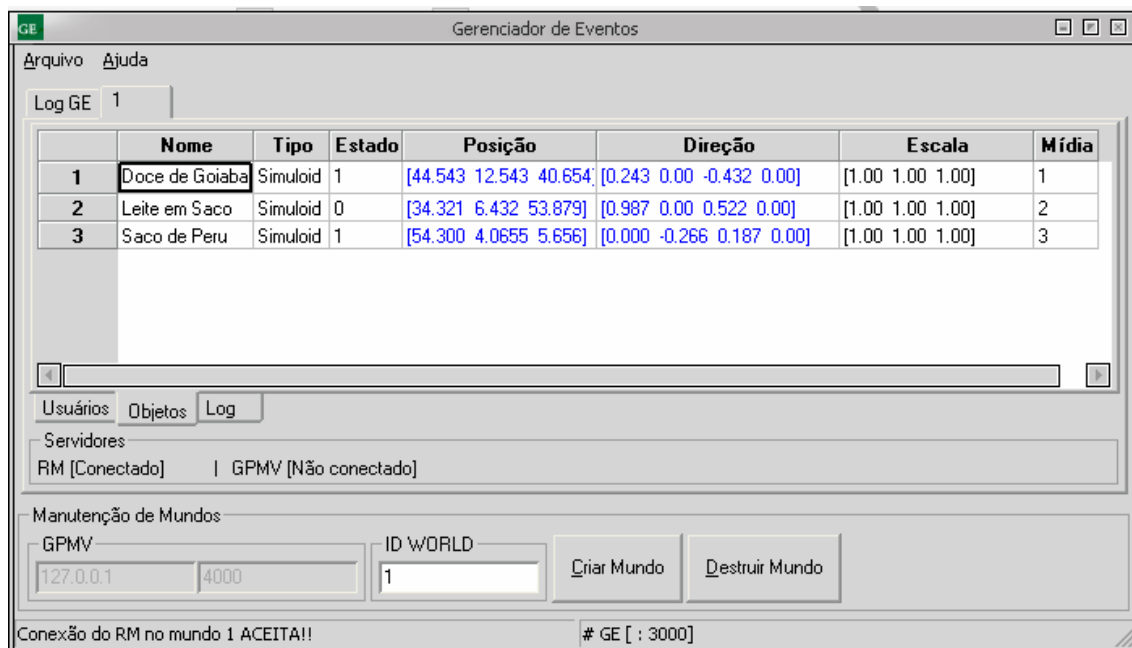


Figura 6.5: Gerenciador de Eventos do CRABCVE.

6.4.2 Entidades

Para o caso de uso, foram escolhidas cinco entidades a serem desenvolvidas: um avatar para representar o usuário; três simulóides representando um alimento de cada categoria: saco de frango (congelado), leite em saco (refrigerado) e doce de goiaba (seco); e um artefato representando a refrigeração de um estoque.

Para todas as entidades foi utilizada uma forma geométrica simples (esférica) a fim de simplificar os testes de colisão, e somente uma aura foi necessária (aura de proximidade).

Avatar

A entidade avatar irá representar o usuário podendo movimentar-se pelo ambiente de realidade virtual e manipular objetos livremente (no caso, os alimentos). Quanto aos estados, o avatar do usuário não precisará mudar de estado, uma vez que não irá utilizar essa informação em nenhum momento.

O avatar não necessita de mecanismo de decisão, já que será controlado pelo usuário, não precisando decidir sobre nenhuma interação ou comportamento e nem mudando de estado.

O avatar irá possuir apenas uma aura de aproximação, definida ao redor do avatar e de forma esférica.

Apenas a especificação dos dois comportamentos será desenvolvida: a movimentação, que será realizada da maneira convencional de sistemas de realidade virtual, e não será considerada uma ação passível de ocorrer interação pois as entidades não necessitarão interagir com o usuário; e a manipulação de entidades, que é uma ação passível de interação e que não afetará o avatar, e sim a entidade que está sendo manipulada, deixando assim o tratamento desse evento com a própria entidade.

A Figura 6.6, apresenta o modelo tridimensional que representa a entidade no ambiente de realidade virtual.



Figura 6.6: Representação do usuário no caso de uso.

Simulóides

As entidades simulóides, que serão definidas para o caso de uso, são alimentos que podem ser recebidos por um restaurante. Assim, será definida a entidade “alimento” que generaliza os três simulóides em questão, deixando as características específicas nos simulóides para serem apresentadas apenas quando necessário.

Cada alimento deve possuir a habilidade de estragar caso alguma situação que o estragaria no mundo real ocorra. Então, o comportamento “estragar” está disponível nas três entidades, mas as ações realizadas nesse método são diferentes em cada alimento. Além disso, os alimentos devem possuir um comportamento chamado de “informar situação” onde apresentam ao usuário informações de temperatura, cheiro, textura, entre outras, referentes a sua condição. Os estados que um alimento pode encontrar-se são:

bom, estragando ou estragado. Na Figura 6.7, encontra-se os comportamentos e os estados dos alimentos.

Restaurante

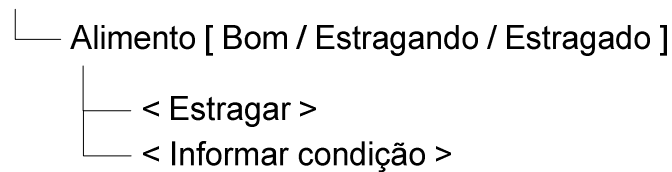


Figura 6.7: Comportamentos e estados da entidade alimento.

O mecanismo de decisão utilizado será uma máquina de estados finitos com ações nos estados (Máquina de Moore).

A entidade alimento possuirá apenas uma aura de aproximação, uma vez que alguns alimentos estragam ao ficarem muito próximos de outros e, no caso dos refrigerados e congelados, começam a aumentar sua temperatura caso distanciem-se demais da refrigeração do estoque. O cálculo de consciência utilizado será o disponível nas extensões do modelo (Booleano).

O diagrama de estados da entidade alimento é apresentado na Figura 6.8. As transições “Saída Refrigeração”, “Retorno Refrigeração” e “Temperatura alta por muito tempo” só serão válidas para os alimentos congelados ou refrigerados, e a transição “Aleatório” representa fatores aleatórios que podem estragar o alimento.



Figura 6.8: Diagrama de estados da entidade alimento.

As transições e as respectivas condições para que a transição ocorra são apresentadas na Tabela 6.1. Os termos em *itálicos* na tabela relacionam-se aos valores específicos do alimento representado.

Tabela 6.1. Condições para a realização de cada transição da entidade alimento.

Transição	Condições
Alimento inadequado próximo	Movimento de entrada na aura de proximidade, Entidade próxima = <i>entidade inadequada</i> .
Alimento inadequado afastado	Movimento de saída na aura de proximidade, Entidades inadequadas próximas = 0.
Alimento inadequado próximo por muito tempo	Movimento de permanência na aura de proximidade, Tempo de proximidade entre as entidades = <i>tempo proximidade</i> .
Saída refrigeração	Tipo de entidade = artefato, Movimento de saída da entidade na aura de proximidade.
Retorno refrigeração	Tipo de entidade = artefato, Movimento de entrada da entidade na aura de proximidade.
Temperatura alta por muito tempo	Temperatura limite alcançada = <i>temperatura limite</i> , Tempo de permanência na temperatura limite = <i>tempo permanência</i> .
Aleatório	Valor randômico.

Na tabela 6.2, são apresentadas as informações utilizadas nas condições e que são específicas de cada simulóide desenvolvido (representado pelos termos em *itálico*).

Tabela 6.2. Informações específicas dos simulóides do caso de uso.

Valores	Saco de Frango	Leite em Saco	Doce de Goiaba
Entidades inadequadas	Doce de Goiaba	-	Saco de Frango
Tempo de proximidade	3min	-	4min
Temperatura limite	20°C	15°C	-
Tempo de permanência	4min	3min	-

Por último, os comportamentos dos simulóides devem ser especificados. O comportamento “Estragar” deve ser ativado quando a entidade passar para o estado “Estragando” e assim, começar a contabilizar o tempo necessário para o alimento estragar sob as condições ocorridas. Caso alguma ação mude novamente o estado da entidade para “Bom”, esse comportamento deve, gradativamente, voltar a entidade ao seu estado normal. Se nada for feito que mude a entidade para o estado “Bom” e o comportamento “Estragar” chegar ao limite de tempo, o método mudará o estado da entidade para “Estragado”. O comportamento “Informar Condição” deve retornar informações importantes ao usuário como temperatura, prazo de validade, textura, entre outras.

A Figura 6.9 apresenta as representação gráficas dos simulóides do caso de uso. Respectivamente, saco de frango, leite em saco e doce de goiaba.

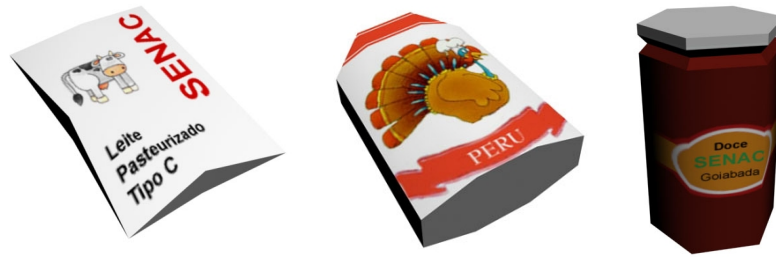


Figura 6.9: Representação gráfica dos simulóides utilizados no caso de uso.

Artefato

A entidade artefato representa a refrigeração do estoque onde os alimentos devem ser guardados. Ela não possui comportamentos, uma vez que não muda seu estado durante sua execução, e nem mecanismo de decisão. A única aura presente é a de proximidade, para que os alimentos possam interagir com o artefato. Também não necessita de cálculo de consciência, uma vez que não precisa estar consciente das demais entidades.

6.5 Considerações Finais

Com o caso de uso foi possível analisar o nível de dificuldade de se utilizar o *BeE Model* em uma aplicação real e em um sistema de realidade virtual existente. A utilização do modelo para desenvolver o caso de uso mostrou a complexidade de se especificar entidades em um ambiente de realidade virtual, mesmo que simples. Uma vez que essa especificação é complexa, é interessante que o desenvolvedor possa dedicar-se a ela, abstraindo-se da maneira de implementá-la. Então, utilizando o modelo, o desenvolvedor pode adicionar a entidade especificada sem muitos problemas no sistema de realidade virtual, uma vez que todos os recursos necessários encontram-se desenvolvidos. Ou, no pior caso, necessitará apenas desenvolver componentes que agreguem novas funcionalidades às existentes.

Quanto ao CRAbCVE, não existiram grandes problemas na inserção do modelo na arquitetura, pois apenas dois componentes foram alterados e as alterações não foram muito significativas. O Gerenciador de Eventos teve sua carga de processamento aumentada devido à inclusão do tratamento da colisão de auras e dos testes de colisão. O Cliente também aumentou seu processamento devido ao teste de colisão de auras que o avatar deve realizar sempre que ocorrer uma ação passível de interação (tanto gerada por ele, quanto advinda de outra entidade). Seria interessante, para a utilização efetiva do modelo na arquitetura CRAbCVE, desenvolver um Controlador de Simulóides para realizar o papel que o Gerenciador de Eventos está realizando no momento, para que a carga de processamento fosse retirada deste componente.

Capítulo 7

Conclusões e Trabalhos Futuros

Nesta dissertação, foi feito um levantamento dos conceitos de entidades, comportamento e interação, para avaliar o universo de possibilidades existentes em aplicações de realidade virtual que possuem entidades participativas, com mecanismos de decisão e reação a estímulos. A partir desse levantamento, foram identificados os aspectos relevantes que as aplicações de RV devem disponibilizar. Essa análise, por si só, é útil para os desenvolvedores de entidades de ambientes virtuais, no que diz respeito à melhor definição dessas entidades e à criação de novas possibilidades na utilização da realidade virtual.

Foi feita também uma análise dos modelos de comportamento existentes na literatura, extraindo-se os aspectos cruciais que um modelo deve atender. A partir da análise dos pontos positivos e negativos dos modelos estudados, foi proposto um modelo de comportamento abrangente e flexível, o Modelo de Entidades Comportamentais (*BeE Model*). Esse modelo, portanto, pode ser considerado mais robusto do que todos os modelos analisados.

O *BeE Model* é simples de utilizar, é genérico suficientemente para ser incorporado em sistemas de realidade virtual diferentes, possibilita a inserção de novas funcionalidades e torna o desenvolvimento de entidades de realidade virtual o mais intuitivo possível.

O modelo é independente do mecanismo de decisão, possibilitando a utilização de diferentes técnicas. Utiliza o conceito de aura, foco e nimbo para caracterizar a ocorrência de interação, como também, o cálculo de consciência. É independente também do cálculo de consciência utilizado, podendo ser desenvolvidas novas formas de realizar esse cálculo, como também, disponibiliza informação de movimentação da entidade além do valor de consciência. Indepeinde também do cálculo de colisão, podendo ser utilizada a biblioteca disponível no sistema gráfico. E, por fim, abstrai os comportamentos, podendo ser utilizado métodos específicos, ou *scripts*.

Também está disponível no *BeE Model* um conjunto de tecnologias alternativas a fim de possibilitar sua utilização: um mecanismo de decisão baseado em máquina de estado finito (possuindo dois tipos de máquinas: Moore e Mearly) e um cálculo de consciência simples.

O modelo foi apresentado por meio da identificação das classes componentes, da forma como elas se relacionam, e de quais são suas possíveis utilizações. O modelo possui dois grupos de classes: o núcleo e as extensões. O núcleo contém as classes essenciais do modelo e descreve como ele funciona de maneira genérica; as extensões ampliam as classes do núcleo, disponibilizando mais ferramentas e deixando o modelo mais robusto.

A apresentação de tecnologias que podem ser utilizadas em conjunto com o modelo proposto indicou possibilidades interessantes de uso. Uma delas (máquina de estado finito) já se encontra disponível nas extensões do modelo, e as demais são apenas sugestões a serem utilizadas no sistema de realidade virtual que irá agregar o *BeE Model*.

Por fim, a elaboração de um caso de uso mostrou a utilização do modelo em uma aplicação real e em um sistema de realidade virtual. Foi notado o grau de dificuldade da especificação de entidades de ambientes virtuais, sendo assim importante que um modelo de comportamento, como o *BeE Model*, esteja disponível no sistema de realidade virtual utilizado, pois assim o desenvolvedor pode concentrar-se nessa especificação e abstrair-se do desenvolvimento das entidades.

Trabalhos Futuros

O modelo proposto precisa ser testado em diversas aplicações de RV para que as eventuais deficiências encontradas possam ser superadas e novas funcionalidades relevantes sejam incorporadas.

Novos mecanismos de decisão devem ser desenvolvidos, dando mais opções aos sistemas que utilizem o modelo. Um mecanismo de decisão baseado em redes neurais e outro em programação evolutiva são idealizados para um futuro próximo, conferindo, assim, inteligência às entidades.

Cálculos de consciência mais robustos são necessários para a evolução do *BeE Model*. Um cálculo de consciência baseado no volume colidido promete ser interessante na determinação da consciência das entidades.

É interessante também a inserção de uma biblioteca de colisão no modelo, a fim de disponibilizá-lo em sistemas ou componentes que não possuam essa funcionalidade. Para que esse sistema seja efetivo, é importante que sejam desenvolvidas também formas para representar as auras, focos e nimbos, e que utilize a biblioteca de colisão.

Quanto ao caso de uso, é necessário que as entidades envolvidas possam ser mais trabalhadas a fim de torná-las mais robustas, e que novas entidades sejam especificadas e inseridas. Possibilitando que o curso de Boas Práticas seja totalmente desenvolvido.

Quanto a arquitetura CRAbCVE, um trabalho bastante pertinente, e que enriqueceria essa arquitetura, é a união do BeE *Model* com a edição proposta em Gomes (2005). Isso possibilitaria a edição de entidades, comportamentos, estados, mecanismos de decisão e cálculos de consciência, para que seja simplificado ainda mais o processo de especificação de entidades, permitindo que até mesmo os usuários do sistema realizem essa tarefa. Ainda quanto à arquitetura CRAbCVE, é importante a utilização do Controlador de Simulóides, e não do Gerenciador de Eventos, no gerenciamento de entidades, para que o GE não seja sobrecarregado. A utilização do Controlador de Simulóides para esse fim poderia alterar seu nome para Controlador de Entidades, que estaria mais conivente com sua nova funcionalidade.

Referências

- Activeworlds Inc. (1997). *ActiveWorlds*. <http://www.activeworlds.com>
- Allbeck, J. M. & Badler, N. I. (2001). *Consistent Communication with Control*. Proceedings of the Workshop on Multimodal Communication and Context in Embodied Agents, Montreal, Canadá.
- ANSA (1989). *The Advanced Network Systems Architecture Reference Manual*. Castle Hill, Cambridge, Inglaterra: Architecture Project Management.
- Apache SOAP (1999). *Apache SOAP*. <http://ws.apache.org/soap/>.
- Apasphere, Ltd. (2002). *omniORB*. <http://omniorb.sourceforge.net/>.
- Argyle, M. (1975). *Bodily Communication*. International Universities Press Inc., New York.
- Bacon, H. (2000). *Audiovisuaalisen kerronnan teoria*. [In Finnish], SKS, Helsinki.
- Badler, N. I. & Allbeck, J. M. (2000). *Towards Behavioral Consistency in Animated Agents*. Proc. Deformable Avatars 2000, Geneva, Switzerland, p.191-205.
- Bates, B. (2001). *Game Design: The Art & Business of Creating Games*. Prim Publising, Roseville, CA.
- Benford, S.; Bowers, J.; Fahlen, L. E.; Greenhalg, C.; Mariani, J.; Rodden, T. (1995). *Networked Virtual Reality and Cooperative Work*. Presence: Teleoperators and Virtual Environments, Vol. 4, No. 4.
- Benford, S.; Bowers, J.; Fahlen, L. E.; Greenhalgh, C. & Snowdown, D. (1997). *Embodiments, Avatars, Clones and Agents for Multi-User, Multi-sensory Virtual Worlds*. Multimedia Systems, 5(2): 93-104.
- Bioware & Interplay (1998). *Baldur's Gate*.
http://www.bioware.com/games/baldurs_gate/.
- Bioware Corp. & Atari. (2002). *Neverwinter Nights*. RPG. <http://nwn.bioware.com/>.
- Birdwhistell, R. L. (1970). *Kinesics and Context*. University of Pennsylvania Press, Philadelphia.

- Bogh-Andersen, P. e Callesen, J. (2001). *Agents as Actors*. In: Qvortrup, L. (ed). *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*. Springer-Verlag, London, p. 132-165.
- Booch, G.; Rumbaugh, J. & Jacobson, I. (1999). *The Unified Modeling Language*. Addison-Wesley, Reading, MA.
- Borland (1994). *Visibroker*. <http://www.borland.com.br/corba/>.
- Botomé, S. P. (2001). *Sobre a Noção de Comportamento*. In: Feldes, H. P. M. & Ziles, U. (orgs.). *Filosofia: Diálogo de Horizontes*. Caxias do Sul: EDUCS, Porto Alegre: EDIPUCRS. v. 1, p. 685-708.
- Bowman, D.; Hodges, L.; Allison, D. & Wineman, J. (1999). *The Educational Value of an Information-Rich Virtual Environment*. *Presence*, 8(3): 317-331.
- Brutzman, D. (1998). *Graphics Internetworking: Bottlenecks and Breakthroughs*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 61-95.
- Bullock, A.; Benford, S. & Bowers, J. (1993). *Requirements of Interaction in Large Virtual Spaces*. In: Benford, S. & Mariani, J. (eds). *Requirements and Metaphors of Interaction – COMIC Deliverable 4.1*. Lanxaster University, Lancaster, p. 61-75.
- Burgoon, M. & Ruffner, M. (1978). *Human Communication*. Holt, Rinehart and Winston, New York.
- Campos, J.; Hornsby, K. & Egenhofer, M. (2003). *A Model for Exploring Virtual Reality Environments*. *Journal of Visual Languages and Computing*, 14(5), pg 469-492.
- Çapin, T. K.; Pandzic, I. S.; Magnenat-Thalmann, N. e Thalmann, D. (1999). *Avatars in Networked Virtual Environments*. John Wiley & Sons, LTD. ISBN 0-471-98863-4.
- Cassell, J. (2000). *Embodied Conversational Interface Agents*. *Communications of the ACM*, 43(4): 70 - 78.
- Churchill, E. F.; Snowdon, D. N. & Munro, A. J. (2001). *Collaborative Virtual Environments – Digital Places and Spaces for Interaction*. In: Deaper, D. & Sanger, C. (eds). *Computer Supported Cooperative Work*. Springer-Verlag, London, p. 316.

- Clanton, C. (2000). *Lessons from Game Design*. In: Bergman, E. (ed). *Information Appliances and Beyond – Interaction Design for Consumer Products*. Morgan Kauffman, San Francisco, p. 299-334.
- Clark-Willson, S. (1998). *Applying Game Design to Virtual Environments*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Tecnology*. ACM Press, New York, p. 229-239.
- Crytek & Ubisoft (2004). *FarCry*. <http://www.farcry-thegame.com/>.
- Csordas, T. J. (2000). *Computerized Cadavers: Shades of Being and Representation in Virtual Reality*. Case Western Reserve University.
<http://www.focusing.org/compucad.html>.
- Cyan Worlds Inc. & Ubisoft. (1993). *Myst*. Aventura.
<http://www.cyanworlds.com/games/myst.php>
- Cyan Worlds Inc. & Ubisoft. (1998). *Riven*. Aventura. <http://www.riven.com/>
- Davide, F.; Holmberg, M. & Lundström, I. (2001). *Virtual Olfactory Interfaces: Electronic Noses and Olfactory Displays*. In: Riva, G. & Davide, F. (eds). *Communications Through Virtual Technology: Identity Community and Technology in the Internet Age*. IOS Press, Amsterdam, p.193-220.
- Droz, A.; Bowers, J.; Benford, S.; Greenhalgh, C. & Fraser, M. (2001). *Collaboratively Improvising Magic: An Approach to Managing Participation in an On-Line Drama*. In: Jarke, M.; Rogers, Y. & Schmidt, K. (eds). *Proc. Seventh European Conference on Computer-Supported Cooperative Work*. Bonn, Germany. Kluwer Academic Press, Amsterdam, p. 159-178.
- Fiske, J. (1982). *Introduction to Communication Studies*. Routledge, London.
- Furness, T. A. (2001). *Toward Tightly Coupled Human Interfaces*. In: Earnshaw, R.; Guejd, R.; van Dam, A. & Vince, J. (eds). *Frontiers of Human-Centred Computing, Online Communities and Virtual Environments*. Springer-Verlag, London, p. 80-98.
- Gabbard, J. L. & Hix, D. (1997). *Taxonomy of Usability Characteristics in Virtual Environments*. Virginia Polytechnic Institute and State University, Blacksburg, VA.

- Goldberg, A. (1998). *Avatars and Agents, or Life Among the Indigenous Peoples of Cyberspace*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 161-180.
- Gomes, G. A. M. (2005). *CRAbCVE – Uma Arquitetura para Viabilização de CVE's Através da Internet*. Dissertação de Mestrado, Mestrado em Ciência da Computação, UFC, Brasil, Defesa em setembro.
- Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*. Academic Press, New York.
- Greenhalgh, C. e Benford, S. (1995). *MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading*. In Proceedings of the 15th International Conference on Distributed Computing Systems (DCS'95), Vancouver, Canada, pg. 27-34, IEEE Computer Society Press.
- Gutwin, C.; Greenberg, S. & Roseman, M. (1996). *Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation*. In: Sasse, R. J.; Cunningham, A. & Winder, R. (eds). *People and Computers XI*. Proceedings of HCI'96 Conference. Springer-Verlag, London, p. 281-298.
- HAMIT, F. (1993). *Realidade Virtual e a Exploração do Espaço Cibernético*. Berkeley Brasil Editora, Original Sams Publishing.
- Harrison, S. & Dourish, P. (1996). *Re-place-ing Space: the Role of Places and Space in Collaborative Systems*. Proc. ACM Conference on Computer Supported Cooperative Work, Boston, MA. ACM Press, New York, NY, p. 67-76.
- Heeter, C. (1992). *Being There: The Subjective Experience of Presence*. Presence, Teleoperators and Virtual Environments, 1: 262-271.
- Houaiss, A. (2001). *Dicionário Houaiss da Língua Portuguesa*. Editora Objetiva. CD-ROM.
- Howland, G. (1999). *Basics of Game AI*. Disponível em:
<http://www.gamedev.net/reference/articles/article784.asp>
- Huxor, A. (2001). *The Role of the Personal in Social Workspaces: Reflections on Working in AlphaWorld*. In: Churchill, E. F.; Snowden, D. N. & Munro, A. J. (eds). *Collaborative Virtual Environments – Digital Places and Spaces for Interaction*. Springer-Verlag, London, p. 282-296.

- ISO/IEC (1991). *Basic Reference Model of Open Distributed Processing*. Working Document on topic 9.1 - ODP Trader. December 1991. Available through national standards bodies.
- Jää-Aro, K. M. & Snowdon, D. (2001). *How not to Be Objective*. In: Churchill, E. F.; Snowdon, D. N. & Munro, A. J. (eds). *Collaborative Virtual Environments – Digital Places and Spaces for Interaction*. Springer-Verlag, London, p. 143-159.
- Jacob, R. J. K.; Leggett, J. J.; Myers, B. A. & Pausch, R. (1993). *Interaction Styles and Input/Output Devices*. Behaviour and Information Technology, 12(2): 69-79.
- Jensen, J. F. (2001a). *Film Theory Meets 3D: a Film Theoretic Approach to the Design and Analysis of 3D Spaces*. In: Qvortrup, L. (ed). *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*. Springer-Verlag, London, p. 311-328.
- Jensen, J. F. (2001b). *Virtual Inhabited 3D Worlds: Interactivity and Interaction Between Avatars, Autonomous Agents and Users*. In: Qvortrup, L. (ed). *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*. Springer-Verlag, London, 23-47.
- Johnson, A. & Leigh, J. (2001). *Tele-Immersive Collaboration in the CAVE Research Network*. In: Churchill, E. F.; Snowdon, D. N. & Munro, A. J. (eds). *Collaborative Virtual Environments - Digital Places and Spaces for Interaction*. Springer-Verlag, London, p. 225-243.
- Johnson, J.; Roberts, T.; Verplank, W.; Smith, D.; Irby, C.; Beard, M. & Mackey, K. (1989). *The Xerox Star: A Retrospective*. IEEE Computer, 22(9): 11-29.
- Johnston, R. S. (1987). *The SIMNET Visual System*. Proc. Ninth Interservice/Industry Training Equipment Conference (ITEC). Washington, DC, p. 264-273.
- Joiner, D. (1998). *Real Interactivity in Interactive Entertainment*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 151-159.
- Kau, E. V. (2001). *Shaping Meaning: On Action and Content in Unreal Worlds*. In: Qvortrup, L. (ed). *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*. Springer-Verlag, London, 329-344.
- Kaur, K. (1999). Interacting with Virtual Environments: An Evaluation of a Model of Interaction. Interacting with Computers 11, 403-426.

- Klüsener, C. S., Pinheiro, P. L. & Cruz, R. M. (2002). *Decomposição do Processo de Comunicação na Ação de Comunicar nas Organizações*. Revista Virtual Comunicação Organizacional. Faculdade de Comunicação Social. Pontifícia Universidade Católica do Rio Grande do Sul. Disponível em: <http://www.pucrs.br/famecos/geacor/texto6.html>
- Labaki, J. (2003). *Introdução a Python – Módulo A*. Grupo Python, Universidade Estadual Paulista Júlio de Mesquita Filho. Disponível em: http://geocities.yahoo.com.br/grupopython/modulo_a.pdf.
- Latta, J. N. (1998). *Virtual Communities: Real or Virtual?* In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 501-511.
- Laurel, B. (1993). *Computers as Theatre*. Addison-Wesley Publishing Company, Inc., Reading, MA.
- Leite-Júnior, A.J.M. (2000). *Ataxia: Uma Arquitetura para a Viabilização de NVE's Voltados para a Educação a Distância Através da Internet*. Dissertação de Mestrado, Mestrado em Ciência da Computação, UFC, Brasil.
- Leontjev, A. N. (1978). *Activity, Consciousness and Personality*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- LucasArts. (1991). *Loom*. Aventura. <http://members.chello.at/theodor.lauppert/games/loom.htm>
- Manninen, T. (2004). *Rich Interaction Model for Game and Virtual Environment Design*. Academic Dissertation, Department of Information Processing Science, University of Oulu, Oulu, Finland.
- Masterson, J. (1996). *Nonverbal Communication in Text Based Virtual Realities*. Dissertação de Mestrado, University of Montana.
- Maxis & Eletronic Arts. (2000). *The Sims*. Estratégia. <http://thesims.ea.com/>
- Microsoft (2004). *Flight Simulator 2004: A Century of Flight*. Simulação. <http://www.microsoft.com/games/flightsimulator>

- Norman, D. A. (1986). *Cognitive Engineering*. In: Norman, D. A. & Draper, S. W. (eds). *User Centered System Design*. Lawrence Erlbaum Associates, Hillsdale, NJ, p. 31-61.
- Nunamaker, J. F. (1997). *Future Research in Group Support Systems: Needs, Some Questions and Possible Directions*. *International Journal of Human-Computer Studies*, 47(3), p. 357-385.
- Nunes, M. P. & Pereira, T. P. (1999). *Uso de Linguagens de Script como Ferramentas de Apoio ao Ensino*. II Semana Acadêmica de Engenharia de Computação. Fundação Universidade Federal do Rio Grande. Disponível em: <http://lula.dmat.furg.br/~python/scriedu.html>
- OMG (1997). *CORBA. Object Management Group*. <http://www.corba.org>.
- Ousterhout, J. (1988). *Tcl. Tool Command Language*. <http://tclbrasil.cipsga.org.br/>.
- Pandzic, I. S.; Çapin, T. K.; Magnenat-Thalmann, N. & Thalmann, D. (1996). *Motor Functions in the VLNET Body-Centered Networked Virtual Environment*. Proc. Eurographics Workshops: Virtual Environments and Scientific Visualization '96, Monte Carlo, p. 94-103.
- Poupyrev, I. & Ichikawa, T. (1999). *Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques*. *Journal of Visual Languages and Computing*, 10(1), p. 19-35.
- PSF (1991). *Python 2.4.1*. - Python Software Foundation. <http://www.python.org>.
- Rees, D. (2000). *Mostly Armless: Grabbing the 3D World*. Proc. Game Developer's Conference [online]. San Jose, CA.. Disponível em: <http://www.gdconference.com/archives/proceedings/2000/rees.doc>.
- Rickel, J. & Johnson, W. L. (1999). *Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control*. *Applied Artificial Intelligence*. v13, no 4. pg 343-382.
- Riva, G. e Galimberti, C. (2001). *Virtual Communication: Social Interaction and Identify in an Eletronic Environment*. In: Riva, G. e Davide, F. (eds). *Communication Through Virtual Technology: Identify Community and Technology in the Internet Age*. IOS Press, Amsterdam, p. 23-46.

- Roadhouse, Z. (1998). *PTUI: Python/Tkinter User Interface (Software)*. Turtles Research Group, LEMS Laboratory, Division of Engineering, Brown University, Providence, RI. Disponível em: <http://starship.python.net/crew/zack/ptui/>
- Robinet, W. (1998). *Interactivity and Individual Viewpoint in Shared Virtual Worlds: the Big Screen Versus Networked Personal Displays*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 331-342.
- Rokeby, D. (1998). *The Construction of Experience: Interface as Content*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 27-47.
- Rorke, M. & Bangay, S. (1999). *A Generic Virtual Reality Interaction System and its Extensions Using the Common Object Request Broker Architecture (CORBA)*. Disponível em: www.citeseer.nj.nec.com/rorke99generic.html.
- Rouse, R. (2000). *Game Design: Theory & Practice*. Wordware Publishing, Inc., Plano, Texas.
- Sánchez-Segura, M. I.; Antonio A. e Seco, A. A. (2004). *Interaction Patterns for Future Interactive Systems Components*. *Interacting with Computers*, 16(2), 331-350.
- Santos, E. M. (2001). *VEGA – Um Modelo de Integração de Aplicações Externas a Ambientes Virtuais Através do Uso de Aplicações Mediadoras e de Simulóides*. Dissertação de Mestrado, Mestrado em Ciência da Computação, UFC, Brasil.
- Shannon, C. E. e Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL.
- Shiny & Atari. (2003). *Enter The Matrix*. Ação. <http://www.enterthematrixgame.com>
- Sinclair, J. (1995). *Collins COBUILD English Dictionary*. HarperCollins Publishers, London.
- Singhal, S. & Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*. ACM Press, New York, NY.
- Sintegra (2005). *Sistema Integrado de Educação a Distância com Suporte a Avaliação de Competências*. <http://www.vdl.ufc.br/sintegra>.
- Skinner, B. F. (1981). *Ciência e Comportamento Humano*. São Paulo: Martins Fontes.

- Slater, M.; Ush, M. & Steed, A. (1994). *Depth of Presence in Virtual Environments*. Presence: Teleoperators and Virtual Environments, Vol. 3, No. 2.
- Snowdon, D. N.; Churchill, E. F. & Munro, A. J. (2001). *Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction*. In: Churchill, E. F.; Snowdon, D. N. & Munro, A. J. (eds). *Collaborative Virtual Environments – Digital Places and Spaces for Interaction*. Springer-Verlag, London, p. 3-17.
- SOLAR (2005). *Sistema Online de Aprendizagem*. <http://www.vdl.ufc.br/solar>.
- Splash Damage & Activision. (2003). *Wolfeinstein: Enemy Territory*. Ação. <http://www.enemy-territory.com>
- Square Enix. (2001). *Final Fantasy X*. RPG. <http://www.square-enix-usa.com/games/FFX/>
- Steed, A. & Slater, M. (1996). *A Dataflow Representation for Defining Interaction Within Immersive Virtual Environments*. Proceedings of VRAIS 96, IEEE Computer Society.
- Steed, A. *Dataflow Languages for Immersive Virtual Environments, Virtual Environments on the Internet, WWW and Networks*. National Museum of Photography, Film & Television, Bradford, UK.
- Steuer, J. (1992). *Defining Virtual Reality: Dimensions Determining Telepresence*. Journal of Communication, 2(42), p. 73-93.
- Tecgraf (2004). *Lua 5.0.2*. Grupo de Computação Gráfica da PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro). Departamento de Informática. <http://www.lua.org>.
- Thalmann, D. (2001). *The Role of Virtual Humans in Virtual Environment Technology and Interfaces*. In: Earnshaw, R.; Guejd, R.; van Dam, A. & Vince, J. (eds). *Frontiers of Human-Centred Computing, Online Communities and Virtual Environments*. Springer-Verlag, London, p. 27-38.
- Trilobyte & Virgin Interactive. (1992). *7th Guest*. Aventura. CD-ROM.
- Twidale, M. (1993). *Rules of Etiquette at the Global Cocktail Party: Issues of Metaphor in CSCW Design*. In: Benford, S. & Mariani, J. (eds). *Requirements and Metaphors of Interaction - COMIC Deliverable 4.1*. Lancaster University, Lancaster, 121-129.

- Ubisoft. (2005). *Tom Clancy's Splinter Cell Chaos Theory*. Ação.
<http://www.splintercell.com/>.
- UFC Virtual (2005). *Instituto UFC Virtual*. <http://www.ufc.br/virtual>.
- Vidal, C. A.; Gomes, G. A. M.; Mendonça-Junior, G. M., Gomes, H. O. O. & Cavalcante-Neto, J. B. (2004). *Uma Ferramenta de Autoria de Ambientes Virtuais Adaptável a Diferentes Motores Gráficos*. Proceedings of the 7th Symposium on Virtual Reality, São Paulo, SP, Brasil, p. 15-26.
- Vilhjálmsson, H. H. & Cassell, J. (1998). *BodyChat: Autonomous Communicative Behaviors in Avatars*. Proc. Second International Conference on Autonomous Agents. Minneapolis, MN, US, 269-276.
- Wann, J. & Mon-Williams, M. (1996). *What Does Virtual Reality NEED?: Human Factors Issues in the Design of Three-dimensional Computer Environments*. International Journal of Human-Computer Studies, 44(6): 829-847.
- Webopédia (2004). Online Computer Dictionary for Computer and Internet Terms and Definitions. <http://www.webopedia.com>
- Weisman, J. (1998). *The Stories We Played: Building BattleTech and Virtual World*. In: Dodsworth, C. (ed). *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York, p. 463-478.
- Wiio, O. A. (1996). *Information and Communication – A Conceptual Analysis*. University of Helsinki, Department of Communication Publications, Helsinki.
- Wikipédia (2004). A Enciclopédia Livre. <http://pt.wikipedia.org/wiki/Comportamento>
- Wloka, M. M. (1995). *Interacting with Virtual Reality*. In: Rix, J.; Haas, S. & Teixeira, J. (eds). *Virtual Prototyping: Virtual Environments and the Product Development Process*. Chapman & Hall.
- Wolf, M. J. P. (2001). *Time in the Video Game*. In: Wolf, M. J. P. (ed). *The Medium of the Video Games*. University of Texas Press, p. 77-91.
- XSOAP (2002). XSOAP. <http://www.extreme.indiana.edu/xgws/xsoap/>.
- Zeltzer, D. (1992). *Autonomy, Interaction and Presence*. Presence, 1: 127-132.