

Onivaldo Rosa Junior

**LRVCHAT3D,  
DESENVOLVIMENTO DE UM AMBIENTE  
VIRTUAL TRIDIMENSIONAL MULTIUSUÁRIO  
PARA INTERNET.**

Dissertação apresentada ao  
Programa de Pós-Graduação em  
Engenharia de Produção da  
Universidade Federal de Santa Catarina  
como requisito parcial para obtenção do grau  
de Mestre em Engenharia de Produção.

Orientador: Ricardo Miranda Barcia, Ph.D.

Florianópolis

2003

R788l Rosa Junior, Onivaldo

LRVCHAT3D, desenvolvimento de um ambiente virtual tridimensional multiusuário para Internet / Onivaldo Rosa Junior; orientador Ricardo Miranda Barcia. – Florianópolis, 2003.

109 f. : il. ; gráfs. ; tabs.

Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, 2003.

Inclui bibliografia

1. Realidade virtual. 2. Ambientes virtuais. 3. Internet (Redes de computadores). 4. Sistemas Multiusuário. 5. VRML (Linguagem de programação de computador). I. Barcia, Ricardo Miranda. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Produção. III. Título.

CDU: 681.31

Onivaldo Rosa Junior

LRVCHAT3D,  
DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL  
TRIDIMENSIONAL MULTIUSUÁRIO PARA  
INTERNET.

Esta dissertação foi julgada e aprovada para a obtenção do grau de **Mestre em Engenharia de Produção** no **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina.

Florianópolis, 11 de Julho de 2003.

---

Prof. Edson Pacheco Paladini, Dr.  
Coordenador do Programa

**BANCA EXAMINADORA:**

---

Prof. Ricardo Miranda Barcia, Ph.D.  
(Orientador)

---

Prof. Eugenio Andrés Díaz Merino, Dr.

---

Prof. Fabiano Luiz Santos Garcia, Dr.

---

Prof. Rodolfo Pinto da Luz, Dr.

## **Dedicatória**

*Ao meu pai, que mesmo não estando mais fisicamente presente neste mundo nunca me deixou sozinho, acompanhando-me sempre durante esta longa jornada, chamada “vida”.*

*A minha mãe e meu irmão que sempre tiveram a paciência de compreenderem meu gênio, e pelo incentivo generoso tão necessário na realização deste trabalho.*

## **Agradecimentos**

*Graça ao Senhor pela oportunidade divina.*

*A Universidade Federal de Santa Catarina por ser fornecedora de conhecimento e ao PPGEP - Programa de Pós-Graduação em Engenharia de Produção por ter acolhido a execução deste trabalho.*

*Ao Professor Ricardo Miranda Barcia que sempre apoiou e confiou na concretização deste trabalho e o aceitou sob sua orientação.*

*Aos amigos e companheiros, Rodolfo e Irla pelo constante apoio e motivação, mesmo quando nos encontrávamos separados por uma longa distância, demonstrando que amizade não é medida pela presença física e sim por elos de mais alto valor.*

*Aos amigos do LRV - Laboratório de Realidade Virtual, pelo companheirismo, paciência e compreensão que permitiram a discussão e a geração de idéias, que culminaram neste trabalho, e por estarem sempre em busca do desenvolvimento da Realidade Virtual, mesmo em condições adversas.*

*Agradeço aos professores do PPGEP as lições recebidas, e por persistirem no desenvolvimento de um ensino de qualidade nesta nação.*

*Enfim, a todos aqueles que de alguma forma, direta ou indiretamente foram responsáveis pela viabilidade deste trabalho.*

*“Singular entre as ciências, ciberespaço já foi uma árvore plantada firmemente nas terras ricas da mitologia humana. Os três autores que definiram este campo: Vernor Vinge<sup>1</sup>, William Gibson<sup>2</sup> e Neal Stephenson<sup>3</sup>; Confiaram exclusivamente em uma fundação mitológica para descrever as complexidades incompreensíveis das redes de computador. Isto dificilmente é uma coincidência; na realidade, isto ecoa num padrão tão antigo quanto a consciência humana”.<sup>4</sup>*

**Mark Pesce.**

---

<sup>1</sup> Autor de *True Names* (1978).

<sup>2</sup> Autor de *Neuromancer* (1984), *Count Zero* (1986) e *Mona Lisa Overdrive* (1988).

<sup>3</sup> Autor de *Snow Crash* (1992).

<sup>4</sup> Trecho de *Ritual and the Virtual* (PESCE, 1997), traduzido livremente pelo autor.

## RESUMO

ROSA Jr., Onivaldo. **LRVCHAT3D, Desenvolvimento de um Ambiente Virtual Tridimensional Multiusuário para Internet**. 2003. 109f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis, 2003.

Vivemos em tempos onde informação é a palavra chave no desenvolvimento pessoal e profissional. Existe uma constante necessidade de sistemas de conhecimento flexíveis e exploratórios que permitam a transmissão de conhecimento, a exploração de formas de apresentação, e, principalmente, formas de colaboração. Este trabalho apresentará uma solução para se ter um espaço interativo tridimensional no qual a colaboração seja o ponto chave e onde as pessoas interajam entre si de uma forma amigável e próxima de seu cotidiano. A solução encontra-se na utilização de conceitos de Realidade Virtual associados com ferramentas clássicas de desenvolvimento de sistemas computacionais, criando ambientes virtuais multiusuário colaborativos para Internet. Desta maneira será apresentada uma visão global dos temas: Realidade Virtual e Ambientes Virtuais. Além disso, serão apresentados o desenvolvimento e implementação de um protótipo de ambiente virtual multiusuário tridimensional para Internet e exemplos de aplicação. O Sistema será denominado de LRVCHAT3D.

**Palavras-chaves:** Realidade Virtual, Ambientes Virtuais, Internet, VRML, Sistemas Multiusuário, Chat.

## ABSTRACT

ROSA Jr., Onivaldo. **LRVCHAT3D, Desenvolvimento de um Ambiente Virtual Tridimensional Multiusuário para Internet**. 2003. 109f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis, 2003.

We live an era where information is the key for personal and professional improvement. There is a need for flexible knowledge systems that allows exploratory activities and knowledge broadcast besides presentation methods, but mainly collaboration forms. This project intends to find a solution for interactive three-dimensional environments where collaboration is the crucial point, and, at the same time, supported by a place where people interact among each other in a friendly way and close to their daily routine. The solution can be found by using virtual reality concepts associated to classical tools for computational system development, throughout the creation of multi-user and collaborative virtual environments to the Internet. Thus it will be shown a global vision that clarifies the following issues: Virtual Reality and Virtual Environments. In order to demonstrate the effectiveness of this technology it will be presented the development and implementation of a multi-user three-dimensional virtual environment prototype to the Internet and examples of applications created from the prototype system. This system is entitled LRVCHAT3D.

**Keywords:** Virtual Reality, Virtual Environments, Internet, VRML, Multi-user Systems, Chat.



## Sumário

<i>Lista de Figuras</i> .....	11
<i>Lista de Quadros</i> .....	12
<b>1. INTRODUÇÃO</b> .....	<b>13</b>
1.1 MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA .....	14
1.2 OBJETIVOS .....	15
1.3 ESTRUTURA DO TRABALHO .....	15
<b>2. REALIDADE VIRTUAL</b> .....	<b>18</b>
2.1 DEFINIÇÕES .....	18
2.2 HISTÓRICO .....	20
2.3 REALIDADE VIRTUAL IMERSIVA E NÃO IMERSIVA .....	21
2.4 EQUIPAMENTOS ESPECIAIS PARA REALIDADE VIRTUAL .....	21
2.5 EXEMPLOS DE APLICAÇÕES QUE UTILIZAM REALIDADE VIRTUAL .....	26
2.5.1 <i>Arquitetura</i> .....	27
2.5.2 <i>Treinamento</i> .....	28
2.5.3 <i>Ensino à distância</i> .....	29
2.5.4 <i>Cultura</i> .....	30
2.5.5 <i>Comércio eletrônico</i> .....	31
<b>3. AMBIENTES VIRTUAIS</b> .....	<b>32</b>
3.1 MUNDOS E COMUNIDADES VIRTUAIS .....	34
3.1.1 <i>Habitat</i> .....	35
3.2 AMBIENTES VIRTUAIS EM REDE .....	36
3.2.1 <i>Comunicação e Distribuição de Dados em AVR</i> .....	38
3.2.2 <i>Desenvolvimento de Ambientes Virtuais em Rede</i> .....	41
3.2.3 <i>Avatar</i> .....	43
3.2.4 <i>Comunicação não Verbal em AVR</i> .....	45
3.3 EXEMPLOS DE AMBIENTES VIRTUAIS PARA INTERNET .....	47
3.3.1 <i>ActiveWorlds</i> .....	48
3.3.2 <i>CyCo Entrance</i> .....	49
3.3.3 <i>Blaxxun Community Plataform</i> .....	50
3.3.4 <i>ParallelGraphics Island</i> .....	51
3.3.5 <i>There</i> .....	52
3.4 AMBIENTES VIRTUAIS COLABORATIVOS .....	54
3.4.1 <i>Projetos e aplicações</i> .....	54
3.5 CONSIDERAÇÕES FINAIS .....	56
<b>4. VRML – LINGUAGEM PARA MODELAGEM EM REALIDADE VIRTUAL</b> .....	<b>57</b>
4.1 ORIGENS E DESENVOLVIMENTO .....	58
<b>5. LRVCHAT3D</b> .....	<b>61</b>
5.1 FASE I .....	64
5.2 FASE II .....	66
5.2.1 <i>Estrutura básica de rede</i> .....	69
5.2.2 <i>Formatos de Arquivos de Ambiente</i> .....	69
5.2.3 <i>Formatos de Arquivos de Avatares</i> .....	70
5.2.4 <i>Formatos de Arquivos de Objetos</i> .....	71

5.2.5	<i>Protocolo de Mensagens</i> .....	74
5.2.6	<i>Programa Servidor</i> .....	78
5.2.7	<i>Programa Cliente</i> .....	79
<b>5.3</b>	<b>FASE III</b> .....	<b>88</b>
<b>5.4</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>92</b>
<b>6.</b>	<b>CONCLUSÕES E RECOMENDAÇÕES</b> .....	<b>93</b>
<b>7.</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>95</b>
<b>8.</b>	<b>APÊNDICES</b> .....	<b>100</b>
<b>8.1</b>	<b>APÊNDICE A - EXEMPLOS COMPLETOS DE CÓDIGO DE OBJETOS</b> .....	<b>100</b>
8.1.1	<i>Objeto 1 - Cone de fuga</i> .....	100
8.1.2	<i>Objeto 2 - Globo girante</i> .....	102
<b>8.2</b>	<b>APÊNDICE B – COMPARATIVO ENTRE A CODIFICAÇÃO VRML97 E X3D</b> .....	<b>105</b>
<b>8.3</b>	<b>APÊNDICE C – NET INTERFACES</b> .....	<b>107</b>

## Lista de Figuras

Figura 1 - Triângulo da Realidade Virtual.....	19
Figura 2 - VFX3D Headgear.....	22
Figura 3 - CyVisor .....	22
Figura 4 - CyberGlove .....	23
Figura 5 - CyberTouch .....	23
Figura 6 - CyberForce .....	24
Figura 7 - CyberGrasp.....	24
Figura 8 - CAVERNA.....	26
Figura 9 - Exemplo de RV na Arquitetura.....	27
Figura 10 - Captura de tela, Monta PC.....	28
Figura 11 - Captura de tela, Laboratório virtual de química. ....	29
Figura 12 - Captura de tela, CD-ROM Oscar Niemeyer, Vida e Obra. ....	30
Figura 13 - Captura de tela, StarLight Express. ....	31
Figura 14 - Cena típica do Habitat.....	35
Figura 15 - Topologias de comunicação. ....	38
Figura 16 - Banco de Dados replicado com comunicação centralizada. ....	39
Figura 17 - Banco de Dados replicado com comunicação distribuída.....	40
Figura 18 - Base de dados replicada com comunicação distribuída. ....	40
Figura 19 - Avatar caricato. ....	44
Figura 20 - Avatar realístico. ....	44
Figura 21 - Captura de tela do ActiveWorlds.....	48
Figura 22 - Captura de tela do Cyco Entrance. ....	49
Figura 23 - Captura de tela do Blaxxun Contact.....	50
Figura 24 - Captura de tela do ParallelGraphics Island.....	51
Figura 25 - Captura de tela do There. ....	53
Figura 26 - Protótipo do programa cliente da Fase I. ....	65
Figura 27 - Protótipo teste design da Fase II.....	67
Figura 28 - Protótipo da Fase II, mostrando um avatar animado. ....	68
Figura 29 - Estrutura básica de rede no sistema.....	69
Figura 30 - Programa Servidor.....	78
Figura 31 - Tela de inicialização.....	79
Figura 32 - Tela indicativa de componente VRML não instalado. ....	80
Figura 33 - Tela principal.....	80
Figura 34 - Menu Configuração.....	81
Figura 35 - Menu Animações. ....	81
Figura 36 - Configuração de avatar e conexão. ....	82
Figura 37 - Visualizando animação do avatar. ....	83
Figura 38 - Seleção de render.....	83
Figura 39 - Janela de seleção de Objetos.....	85
Figura 40 - Adicionando um objeto ao ambiente.....	86
Figura 41 - Menu auxiliar do objeto. ....	87
Figura 42 - Menu auxiliar do Avatar. ....	87
Figura 43 - Captura de tela do VIRTUALLAB.....	89
Figura 44 - Captura de tela do VIRTUALNET. ....	90
Figura 45 - Captura de tela da interface WEB do MediaServer. ....	91
Figura 46 - Captura de tela do MediaChat3D.....	91

## Lista de Quadros

Quadro 1 - Fatores no desenvolvimento de sistemas distribuídos.....	42
Quadro 2 - Resumo da Fase I.....	65
Quadro 3 - Resumo da Fase II.....	68
Quadro 4 - Código base de avatar do sistema.....	71
Quadro 5 - Código estrutural de objetos do sistema.....	72
Quadro 6 - Sensores de interação.....	84

## 1. INTRODUÇÃO

No início, a navegação na Internet era um passeio estático e solitário, então elementos dinâmicos foram adicionados progressivamente, mas mesmo assim continua sendo uma atividade solitária na maioria dos casos. A navegação é bidimensional e, em geral, somente ocorre através de atalhos chamados de *hyperlinks*<sup>5</sup>. Com o advento da VRML (*Virtual Reality Modeling Language*, ou Linguagem para Modelagem em Realidade Virtual) (WEB3D, 2003) surgiu uma forma diferenciada de apresentação de conteúdo na Internet (CAREY, 1997). A VRML possui como característica a ilusão de espaço tridimensional e mecanismos que permitem que o usuário interaja com este espaço. Mas a natureza solitária da experiência de navegação na Internet não foi eliminada.

Mecanismos de comunicação textual, como IRC<sup>6</sup> (IRC, 2003) e ICQ<sup>7</sup> (ICQ, 2003) entre outros, além de ferramentas integradas de comunicação como o Microsoft Netmeeting<sup>8</sup> (MICROSOFT, 2001), contribuíram na expansão do fator interativo da Internet, permitindo comunicação em tempo real.

Mas como fazer para se ter um espaço interativo tridimensional, no qual a colaboração seja o ponto chave, no qual as pessoas interajam entre si de uma forma amigável e próxima de seu cotidiano?

A resposta para esta pergunta pode estar na utilização de conceitos de Realidade Virtual<sup>9</sup> (RV) (BURDEA, 1994; BYRNE, 1995), em associação com ferramentas clássicas de desenvolvimento de sistemas computacionais, criando os chamados Ambientes Virtuais Colaborativos<sup>10</sup> (AVC) (BENFORD, 2001). Sendo este o enfoque deste trabalho, que apresentará uma visão global sobre o tema e um estudo de caso prático, representado pelo desenvolvimento de um protótipo de ambiente virtual multiusuário tridimensional para Internet, denominado de LRVCHAT3D (ROSA Jr., 2001).

---

<sup>5</sup> "Hyperlink", também conhecido simplesmente por "link". Não há tradução definida em português, embora algumas pessoas usem "apontador", "ponteiro" ou "vínculo". É o texto ou imagem que pode ser "clicado" com o mouse, levando o usuário a uma outra localidade (pagina ou documento) na Internet.

<sup>6</sup> *Internet Relay Chat*, programa de bate-papo, baseado em salas temáticas.

<sup>7</sup> Programa de comunicação através de mensagens instantâneas.

<sup>8</sup> Programa de comunicação que permite o uso de texto, voz, vídeo e o uso colaborativo de aplicativos.

<sup>9</sup> Do termo original: Virtual Reality (VR).

<sup>10</sup> Do termo original: Collaborative Virtual Environment (CVE).

A palavra chave desta nova era é **Informação**, e a Internet é a nova fronteira de disseminação da informação. A utilização racional destes recursos será a questão decisiva no desenvolvimento e manutenção das instituições.

O uso racional de recursos e a busca constante do aprimoramento através da educação serão elementos constantes em nossa sociedade, e neste aspecto a Realidade Virtual fornece um novo, porém ilimitado, ambiente colaborativo que permite a formação de alunos e profissionais de diversos níveis, além de sistemas de simulação complexos.

Muito é especulado acerca da interface com o usuário dos sistemas de informação, mas ainda não se tem uma visão clara da interface **definitiva**. A proposta, neste sentido, é que esta interface esteja o mais próximo possível da realidade das pessoas e que ela possa ser ajustada e adaptada para a realidade socioeconômica e cultural do usuário, utilizando a Realidade Virtual como base desta nova interface.

Aproximando a interface computacional de uma visão mais realística, permitindo assim a exploração de uma faixa maior de nossa capacidade sensorial, este é um fator determinante no quociente de absorção do usuário para com os sistemas de informação que se torna essencial devido a crescente complexidade e volume de informação transmitidas ao usuário em sistemas desse tipo. Qualquer empreendimento que venha a melhorar o grau de cognição do usuário é válido, o que tem sido comprovado através da criação e desenvolvimento de sistemas multimídia cada vez mais completos e complexos. A Realidade Virtual só tem colaborado com estes sistemas, agregando visão do espaço tridimensional, livre exploração em tempo real, não linearidade de navegação, apresentação de conteúdo usando formas reais, entre outros. Este conjunto de possibilidades permite um aumento da capacidade cognitiva dos indivíduos.

## 1.1 Motivação e Definição do Problema

O que motivou o desenvolvimento deste trabalho é a constante necessidade de sistemas de conhecimento flexíveis e exploratórios, que permitam a transmissão de conhecimento, a exploração de formas de apresentação, e principalmente formas de colaboração.

Diversos sistemas já possuem estas características, mas são normalmente sistemas proprietários ou de custo elevado, limitando ou mesmo não permitindo uma flexibilização no desenvolvimento de aplicações próprias, e gerando uma dependência excessiva.

O problema então consiste na pesquisa e desenvolvimento de um sistema que auxilie e estimule a interação colaborativa entre os indivíduos em um processo de absorção de conhecimento. Este sistema deve ser flexível e expansível, de forma a crescer em conjunto com as suas aplicações, de forma a possuímos o controle sobre uma parcela considerável da tecnologia utilizada, ou que as dependências tecnológicas sejam minimizadas.

## **1.2 Objetivos**

O objetivo geral deste trabalho é desenvolver e implementar um ambiente virtual tridimensional multiusuário para Internet. Para isto serão utilizados conceitos de Realidade Virtual e Ambientes Virtuais. O Resultado deste trabalho viabiliza o desenvolvimento de projetos relacionados com a utilização da Realidade Virtual como interface de aplicações de integração entre usuário e sistemas computacionais. Espera-se obter um sistema de Realidade Virtual que auxilie e estimule a interação colaborativa entre indivíduos no processo de absorção do conhecimento.

Como objetivos específicos serão apresentados (1) elementos conceituais de Realidade Virtual, Ambientes Virtuais e da linguagem VRML utilizada no desenvolvimento do sistema protótipo, (2) descrição sobre a implementação de um protótipo funcional do sistema e (3) demonstração de sua utilização através de aplicações do sistema protótipo.

## **1.3 Estrutura do Trabalho**

A dissertação está organizada de maneira a iniciar o leitor em conceitos e considerações sobre Realidade Virtual, Ambientes Virtuais e VRML, que formam a base do conhecimento fundamental na compreensão e assimilação do sistema

desenvolvido: O LRVCHAT3D.

O conteúdo específico desta dissertação inicia com o **capítulo 2**, que apresenta conceitualmente a Realidade Virtual, um resumo de seu desenvolvimento histórico e diferencia os conceitos de Realidade Virtual imersiva e não imersiva. Ainda neste capítulo são apresentados alguns dos principais equipamentos especiais para RV assim como alguns exemplos de Realidade Virtual aplicada.

No **capítulo 3** são apresentados conceitos e definições de Ambiente Virtual, Mundos Virtuais e Comunidades Virtuais. É apresentado ainda um enfoque progressivo do tópico citado, passando por Ambientes Virtuais em Rede e chegando ao estado da arte em Ambientes Virtuais Colaborativos. O item 3.4 (Ambientes Virtuais em Rede) é bem detalhado, pois o sistema protótipo gerado ao final do trabalho corresponde a uma típica aplicação deste conceito. Alguns tópicos especiais também tratados neste capítulo são estrutura de rede, desenvolvimento de ambientes virtuais em rede, avatares<sup>11</sup>, comunicação não textual e aspectos construtivos.

A linguagem VRML, que é apresentada no **capítulo 4**, é uma das linguagens mais conhecidas e padronizadas, quando se fala em RV para Internet em equipamentos *desktop*<sup>12</sup>. Esta linguagem é adotada como interface tridimensional, que em conjunto com uma linguagem de programação tradicional irá compor a tecnologia de construção do sistema protótipo, LRVCHAT3D, apresentado no capítulo 5. Sobre este tópico são apresentadas ainda informações históricas e acerca de seu desenvolvimento, além de exemplos de codificação em seu formato nativo que para efeito comparativo será apresentado o mesmo exemplo codificado em sua nova versão chamada de X3D<sup>13</sup> (WEB3D, 2003).

No **capítulo 5** é apresentado o desenvolvimento e implementação de um ambiente virtual multiusuário tridimensional para Internet. O projeto é apresentado e suas fases de desenvolvimento detalhadas. As duas primeiras fases foram realizadas ao longo do período de realização do curso de mestrado do autor e a terceira fase do projeto foi parcialmente concluída, sendo alguns resultados da terceira fase apresentados também neste capítulo.

---

<sup>11</sup> Resumidamente, avatar é a representação do usuário em um ambiente virtual, mais detalhes no item 3.2.3 deste trabalho.

<sup>12</sup> Computadores de mesa.

<sup>13</sup> X3D é um abreviação de eXtensible 3D, ou 3D extensível.



As conclusões e recomendações são apresentadas no **capítulo 6**, bem como sugestões e trabalhos futuros. As referências utilizadas neste documento se encontram no **capítulo 7**.

## 2. REALIDADE VIRTUAL

Neste capítulo serão apresentados conceitos e definições de Realidade Virtual (RV) na visão de diversos autores, um resumo de seu histórico e uma explicação sobre a diferenciação da RV imersiva e não imersiva. Fechando o capítulo são apresentados alguns equipamentos especiais utilizados em RV e algumas aplicações da Realidade Virtual em diversos campos.

### 2.1 Definições

*“É difícil dizer o que é impossível, pois a fantasia de ontem é a esperança de hoje e a realidade de amanhã”.*<sup>14</sup>

(Robert H. Goddard, 1882 - 1945)

Latta (1994) cita Realidade Virtual como uma avançada interface homem-máquina que simula um ambiente realístico e permite que participantes interajam com ele: Realidade Virtual envolve a criação e experimentação de ambientes.

Pimentel (1995) define Realidade Virtual como o uso da alta<sup>15</sup> tecnologia para convencer o usuário de que ele está em outra realidade - um novo meio de “estar” e “tocar” em informações: Realidade Virtual é o local onde os humanos e computadores fazem contato.

Segundo Luz (1997, p. 24), Realidade virtual é:

“A utilização de artifícios para a reprodução da realidade, sendo que atualmente o meio mais utilizado é o digital, através do uso dos computadores. Para se criar à realidade virtual, é necessário aguçar o maior número de sentidos do usuário, sejam eles visual, auditivo, tátil, dentre outros. Assim, o usuário sente-se inserido, ou seja, imerso em um ambiente no qual pode interagir com objetos e outras pessoas.”

Uma aplicação que utiliza Realidade Virtual deve proporcionar imersão e interação, e ainda estimular a imaginação, formando assim uma trilogia (BURDEA, 1994), representada pelo **Triângulo da Realidade Virtual** (figura 1). Mas isto nada mais é que o transporte para o contexto “virtual” destes três importantes fatores que

---

<sup>14</sup> Tradução do Autor para a frase “It is difficult to say what is impossible, for the dream of yesterday is the hope of today and the reality of tomorrow” do Físico e pioneiro na engenharia de foguetes, Robert H. Goddard.

<sup>15</sup> No sentido de moderna, isto é, se referindo as tecnologias modernas ou em desenvolvimento.

representam nossa realidade, nosso cotidiano. Através de uma imersão total em um ambiente completamente interativo, constantemente modificado pela nossa imaginação.



Figura 1 - Triângulo da Realidade Virtual.

Podemos assim traçar um paralelo entre o significado de nossa realidade e o conceito de Realidade Virtual. Quanto maior o nível de imersão, interação e de imaginação aplicado a um sistema, mais próximo chegamos da sintetização de uma nova realidade, a Realidade Virtual.

Mas como podemos explorar estes fatores em sistemas de RV? Novamente recorreremos ao paralelo com nossa realidade: estamos imersos no mundo real através de nossos sentidos e interagimos com ele através de inferências no meio utilizando nossa representação física. Desta maneira geramos, através do imaginativo, um moto<sup>16</sup> contínuo de ações e reações temporais que caracterizam esta realidade.

Através das considerações anteriores, podemos desenvolver os três fatores de interesse (Imersão, Interação e Imaginação). Estamos imersos no sistema através de interfaces que de alguma forma emulam nossos sentidos (atualmente concentrados na visão, audição e tato, mas com pesquisas avançando sobre os outros sentidos como o paladar e o olfato). Exploramos a visão através da projeção de ambientes no espaço virtual, e a audição através da geração de som com profundidade variável, isto é, posicional. O tato esta sendo explorado com a utilização de luvas que emulam a pressão e textura de contato.

Interagimos utilizando dispositivos de controle no espaço, que permitem a

---

<sup>16</sup> Ato ou efeito de mover; movimento.

mobilidade e a inferência no meio, podendo utilizar desde um simples teclado de computador até dispositivos de captura completa de movimentos corporais, gerando desta maneira a própria movimentação referencial do usuário ou manipulando o ambiente virtual.

A utilização de técnicas de modelagem espacial, textura, animação entre outras, estimulam nossa imaginação, completando assim um ambiente propício para a efetivação de nossa nova realidade, a RV.

## 2.2 Histórico

Em termos de área de estudo podemos dizer que a Realidade Virtual teve início com o aparecimento dos primeiros simuladores de vôo, que reproduziam o movimento de aeronaves através de um *cockpit*<sup>17</sup> que respondia a controles mecânicos de acionamento pneumático (SHERMAN, 2002).

Com o advento dos computadores eletrônicos nos anos 50, os simuladores passaram a usar equações matemáticas para a modelagem das características das aeronaves e passou-se a utilizar instrumentos com algum retorno de informação ao operador, aumentando assim sua percepção de realismo. Foi então desenvolvido um equipamento por Morton Heilig para a indústria de entretenimento chamado de *Sensorama* (RHEINGOLD, 1991), que criava a ilusão de realidade através de um filme em 3D e movimentos mecânicos.

Ivan Sutherland (1965) escreveu um artigo denominado *The Ultimate Display* que é considerado o ponto de partida para a Realidade Virtual. Neste artigo ele disserta sobre a idéia de simulação, com uma interface onde o computador controla a existência da matéria. Em 1968 foi criado o primeiro Capacete de visualização para 3D, usando dois televisores em miniatura montadas na frente dos olhos e um sensor de movimento (SUTHERLAND, 1968).

A década de 80 foi chave para o desenvolvimento da Realidade Virtual. Foram realizados vários estudos e projetos envolvendo hardware e sensoriamento remoto. Atualmente com o constante desenvolvimento dos computadores pessoais um número cada vez maior de pessoas tem acesso a esta tecnologia. Outro fator que

---

<sup>17</sup> Espaço onde se aloja o piloto nos aviões.

esta contribuindo para este desenvolvimento é a criação e padronização de técnicas e equipamentos. A popularização da Realidade Virtual esta acontecendo devido ao uso da VRML (CAREY, 1997) que foi criada para a Internet, e através dos jogos, que exploram cada vez mais a interação em ambientes tridimensionais.

### **2.3 Realidade Virtual Imersiva e Não Imersiva**

Podemos dizer que o fator imersão esta diretamente relacionado com a capacidade do sistema em RV de envolver os sentidos do usuário, fazendo com que as ações e reações da interação entre usuário e ambiente virtual sejam transparentes e que o usuário tenha a sensação de “realmente” estar dentro do ambiente, vivenciando plenamente esta realidade sintetizada pelo computador.

A questão da imersão em sistemas de RV é um pouco complexa, pois depende de diversos fatores como tipo de aplicação, percepção e imaginação do usuário. Em termos de hardware considera-se RV imersiva, quando utilizamos um conjunto de dispositivos chamados “imersivos” tais como: capacetes (*HMD - Head Mounted Display*), telas de projeção (*WorkBenchs*) ou salas de projeção (*CAVERNAS*), dispositivos rastreadores de movimento (sensores de movimento, luvas, etc.), som com posicionamento espacial (3D), entre outros.

A RV não imersiva é normalmente caracterizada pela simplicidade de equipamentos de interface homem-máquina, fazendo uso de monitores e dispositivos de interação simples (mouse, teclado, controle de jogos<sup>18</sup> e similares).

Existem aplicações que são facilmente classificadas nestes termos, mas em muitos casos o nível de imersão varia com a utilização de combinações de equipamentos típicos de RV imersiva e não imersiva. Além disso, o fator imersão não esta somente relacionado ao equipamento utilizado, mas também à qualidade representativa do ambiente virtual (gráficos, som, tempo de resposta e elementos ativos de interface, são alguns exemplos).

### **2.4 Equipamentos especiais para Realidade Virtual**

---

<sup>18</sup> Também conhecidos como joysticks e joypads, utilizados em vídeo games.

Dentre a variedade de equipamentos existentes, alguns são para uso genérico (luvas, capacetes, sensores de movimento, etc.), outros são de uso específico em determinadas aplicações, como por exemplo, uma pinça cirúrgica adaptada com sensores e atuadores de maneira a capturar a pressão dos dedos do médico e ao mesmo tempo simular a resistência de contato de um determinado tecido humano.

Não serão detalhados aqui os equipamentos comuns como teclados, mouses, monitores de Tubo de Raios Catódicos ou Telas de Cristal Líquido. As diversas funções de equipamentos de RV servem a tarefas de visualização, manipulação, rastreamento de movimento, retorno de sensibilidade (tato, calor, pressão, etc.).

A seguir serão detalhados alguns dispositivos utilizados em RV:



Figura 2 - VFX3D Headgear

Fonte: Interactive Imaging Systems, Inc.



Figura 3 - CyVisor

Fonte: Intersense, Inc.

**CAPACETE ou HMD (Head Mounted Display)** – dispositivo de visualização composto por duas telas montadas em um capacete, de forma a substituírem o campo de visão do usuário. Uma característica explorada nestes equipamentos é a utilização da visão estereoscópica (cada tela exibe uma imagem levemente deslocada) criando a ilusão de profundidade. Alguns equipamentos são providos com sensores para captura do movimento da cabeça, permitindo assim a atualização do ponto de vista da imagem apresentada nas telas de acordo com o posicionamento da linha de visão do usuário. A presença de fones de ouvido em alguns modelos também colabora com a sensação de imersão, possibilitando a utilização de som estéreo posicional. Nas figuras 2 e 3 podemos observar alguns

modelos comerciais de capacetes.

**LUVAS** – São dispositivos de manipulação, onde são capturados os movimentos dos dedos e a orientação espacial da mão do usuário; existe uma variedade de luvas as quais utilizam diversos tipos de sensores: magnéticos, ópticos, mecânicos, entre outros. Outras luvas possuem características especiais de sensibilidade que propiciam ao usuário a sensação de tato (*tactile-feedback*) (LYNCH, 1992) e força (*force-feedback*) (MARK, 1996).

As figuras de 4 a 7 apresentam ilustrativamente uma série de luvas comerciais (IMMERSION, 2001) com funções diferenciadas.



Figura 4 - CyberGlove

Fonte: Immersion Corporation.



Figura 5 - CyberTouch

Fonte: Immersion Corporation.

A luva *CyberGlove* (Figura 4) é uma luva leve com sensores flexíveis que medem a posição e movimento dos dedos e pulso.

A luva *CyberTouch* (Figura 5) se caracteriza por possuir pequenos estimuladores vibrotácteis em cada dedo e na palma da mão. Cada estimulador pode ser programado para variar a força de sensação de toque individualmente. Os estimuladores podem gerar sensações simples como pulsos ou vibração contínua e, em combinação, podem ser usados para produzir complexos padrões de avaliação táteis. Estas luvas possuem perfil de atuação programável para permitir o alcance da sensação tátil desejada e inclusive a percepção de tocar um objeto sólido em um mundo virtual simulado.



Figura 6 - CyberForce

Fonte: Immersion Corporation.



Figura 7 - CyberGrasp

Fonte: Immersion Corporation.

A luva *CyberForce* (Figura 6) é um dispositivo de *force-feedback* resistivo com armação de base fixa que transmite forças de base terra<sup>19</sup> para a mão de uma forma bem realística. Além desta função este dispositivo também serve como *tracker*<sup>20</sup> posicional realizando medidas de translação e rotação da mão no espaço tridimensional.

A luva *CyberGrasp* (Figura 7), é um dispositivo de *force-feedback*, formada por um exoesqueleto que transmite sensações de força de resistência para cada dedo. Esta resistência, independente para cada dedo, permite uma sensação de volume de objetos sólidos em ambientes virtuais.

Este conjunto de dispositivos aqui apresentados representam alguns exemplos do avanço significativo na interface homem-computador que habilita os usuários a manterem uma interação háptica<sup>21</sup> (RUSPINI, 1997). Tocando e sentindo objetos tridimensionais gerados por computador e experimentar uma interação realística através da interface mais natural possível - a mão humana. Os dispositivos e os métodos de interação que representam podem ser combinados de diversas maneiras, de forma a produzir os melhores efeitos para determinada aplicação.

**CAVERNA**<sup>22</sup> – É uma designação genérica para uma classe de dispositivos de apresentação de Realidade Virtual. O primeiro dispositivo do tipo foi construído

<sup>19</sup> Forças referenciais com base na terra, isto é, a força peso.

<sup>20</sup> Sensor de movimento.

<sup>21</sup> Relativo ao sentido do tato.

<sup>22</sup> Também conhecida pelo termo em inglês, CAVE.



pela equipe da Dra. Carolina Cruz-Neira no *EVL - Electronic Visualization Laboratory* (Laboratório de visualização eletrônica) da Universidade de Illinois em Chicago no ano de 1992 (CRUZ-NEIRA, 1993), e apresentado ao público na SIGGRAPH'92<sup>23</sup>. Desde então vários pesquisadores e institutos comerciais construíram sistemas semelhantes para pesquisas e aplicações em Realidade Virtual.

Uma CAVERNA é um dispositivo de alta tecnologia normalmente disponível apenas em laboratórios industriais ou de pesquisa. O sistema de exibição consiste de telas de projeção semitransparentes, projetores de vídeo estéreo, espelhos, e um sistema de som ambiente. A qualidade gráfica requerida por sistemas desse tipo é fornecida pelo que a há de melhor em termos de supercomputadores gráficos. Como a tecnologia ainda é nova, muita pesquisa é dirigida aos fundamentos da tecnologia de imersão: dispositivos de interação e conceitos de interface, técnicas de visualização, ambientes compartilhados, fatores psicológicos e usabilidade.

A tecnologia deste equipamento baseia-se na projeção estéreo para obter um efeito de percepção de profundidade que é muito importante no processo de ilusão para uma experiência imersiva. O usuário pode caminhar ao redor de um objeto que flutua claramente no meio da caverna ou pode vislumbrar horizontes distantes. Equipamentos de manipulação e controle podem ser usados em conjunto com os dispositivos CAVERNA de visualização para ampliar seu campo de ação.

O impacto de uma CAVERNA é originado principalmente pela representação de seus gráficos. São utilizadas grandes telas de projeção (Figura 8) que cercam o usuário com gráficos de alta resolução. Uma só tela quase cobre o campo de visão do espectador, mas um número variável de telas é utilizado, dependendo do projeto de cada equipamento. Um caso típico utiliza a interação de três paredes e uma projeção de solo, que é o suficiente para criar a ilusão de que o usuário está dentro de um ambiente, sendo este gerado por computador.

Laboratórios de simulação industriais e científicos adotaram avidamente sistemas do tipo caverna; a habilidade para visualizar e interagir com simulações de fluxo de gás, testes de impacto, projetos de design de produtos, entre outros, oferece aos cientistas e pessoal técnico industrial uma ferramenta nova para observação e avaliação de sistemas em um ambiente próximo do real.

---

<sup>23</sup> SIGGRAPH é uma conferência anual relativa à computação gráfica e técnicas interativas.



Figura 8 - CAVERNA

Fonte: The Virtual Reality Application Center (VRAC),  
Institute for Physical Research and Technology, Iowa State University.

## 2.5 Exemplos de Aplicações que Utilizam Realidade Virtual

Atualmente encontramos vários exemplos da utilização da Realidade Virtual em diversas áreas do conhecimento, não mais restrita aos laboratórios de pesquisa, mas sim integrada ao nosso dia-a-dia. Ela está sendo utilizada tanto para fins educacionais como para fins comerciais ou industriais.

A seguir são apresentados alguns exemplos ilustrativos do uso de RV não imersiva em diversas áreas: Arquitetura, Ensino à Distância, Cultura, Comércio Eletrônico e treinamento.

### 2.5.1 Arquitetura

**Projeto:** *VRND, A real-time virtual reconstruction, Notre Dame Cathedral* (Reconstrução virtual em tempo real da Catedral de Notre Dame) (VRND, 2003).

**Desenvolvido por:** Digitalo Studios, Inc.

**Tempo de desenvolvimento:** Oito meses.

**Objetivo:** Reconstrução arquitetônica virtual em tempo real do interior da famosa Catedral de Notre Dame localizada em Paris – França. O projeto foi desenvolvido visando o uso de equipamento de baixo custo.

**Aspectos tecnológicos:** Utilizou técnicas de modelagem tridimensional para tempo real. Utiliza o motor gráfico de um jogo chamado Unreal, produzido pela empresa Epic Games, Inc.



Figura 9 - Exemplo de RV na Arquitetura.

Fonte: VRND Project.

## 2.5.2 Treinamento

**Projeto:** Monta PC  
(LRV, 2003).

**Desenvolvido por:** Laboratório de Realidade Virtual, PPGEP - UFSC.

**Objetivo:** Criar uma aplicação WEB<sup>24</sup> que auxilie na compreensão das etapas de montagem e desmontagem de um computador pessoal assim como apresentar informações sobre cada componente. Demonstrando as dependências nas seqüências de montagem e desmontagem, ou seja, algumas partes devem ser montadas ou desmontadas antes de outras. É uma aplicação simples que visa demonstrar a aplicabilidade dos conceitos de RV em treinamento.

**Aspectos tecnológicos:** Utiliza a linguagem VRML associada à linguagem HTML<sup>25</sup> em uma aplicação típica para WEB.

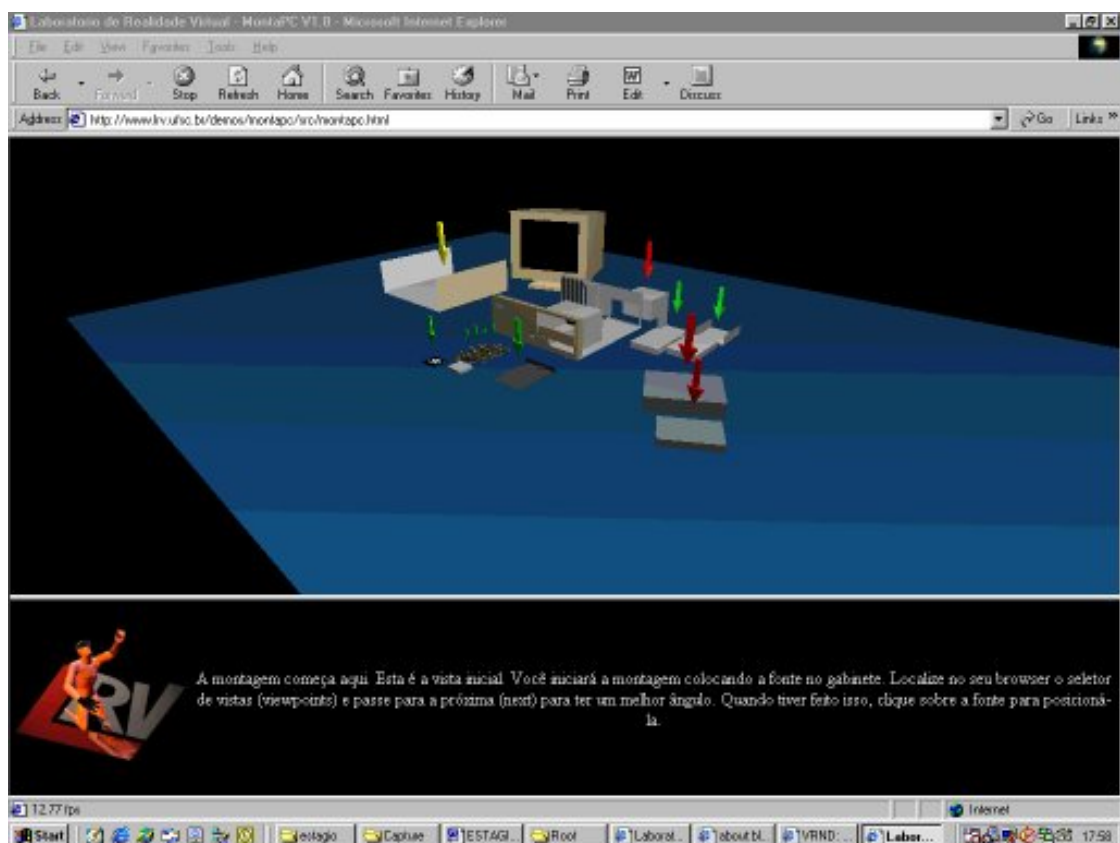


Figura 10 - Captura de tela, Monta PC.

<sup>24</sup> Nome pelo qual a rede mundial de computadores internet se tornou conhecida a partir de 1991, quando se popularizou devido à criação de uma *interface gráfica* que facilitou o acesso e estendeu seu alcance ao público em geral.

<sup>25</sup> HyperText Markup Language (HTML), é a linguagem comumente usada para a criação de páginas de informação na web.

### 2.5.3 Ensino à distância

**Projeto:** Virtual Chemical Laboratory (Laboratório Virtual de Química) (PARALLELGRAPHICS, 2001).

**Desenvolvido por:** ParallelGraphics.

**Objetivo:** Criar um laboratório virtual para o ensino de química. O exemplo apresentado pela ParallelGraphics possui somente uma experiência, a da síntese de sulfato de cobre hidratado. Todo o procedimento é controlado pelo aluno, que pode observar a estrutura molecular de qualquer elemento ou do composto, durante a execução da experiência. É uma aplicação simples que visa demonstrar a aplicabilidade dos conceitos de RV em ensino laboratorial.

**Aspectos tecnológicos:** Utiliza a linguagem VRML associada à linguagem HTML em uma aplicação típica para WEB.

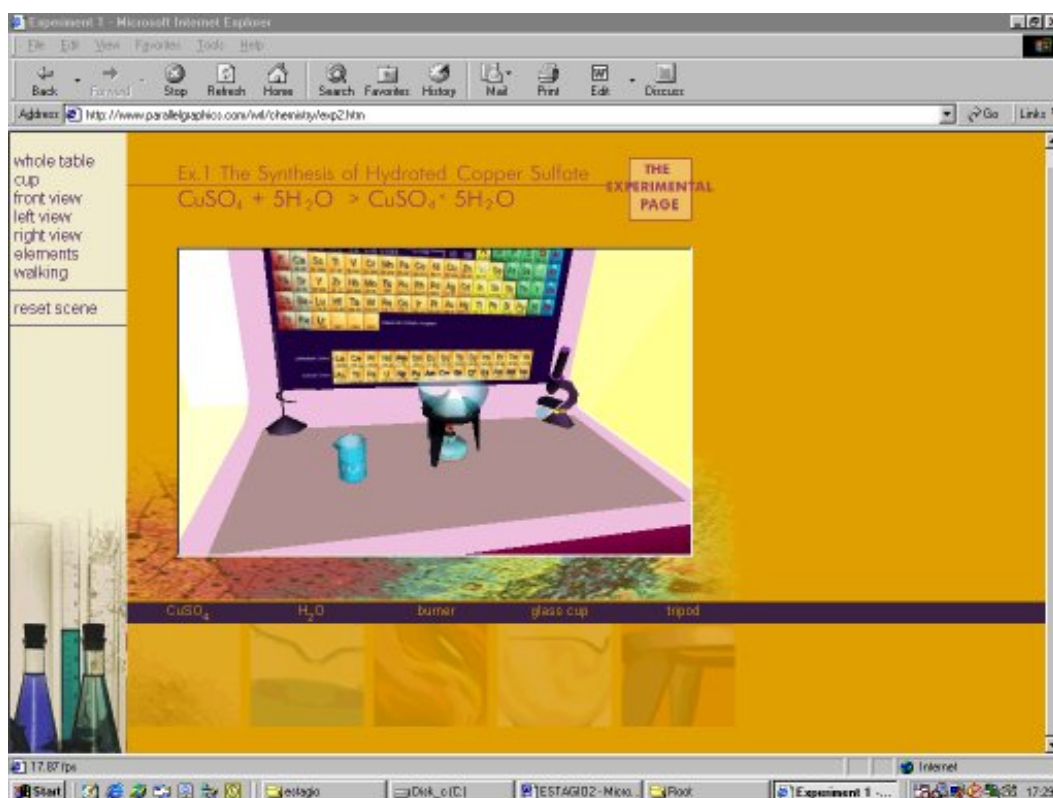


Figura 11 - Captura de tela, Laboratório virtual de química.

## 2.5.4 Cultura

**Projeto:** CD-ROM Oscar Niemeyer, Vida e Obra (LRV, 2003).

**Desenvolvido por:** Laboratório de Realidade Virtual, PPGEP - UFSC.

**Objetivo:** Apresentar através de um CD multimídia, a vida e obra do arquiteto Oscar Niemeyer (REBELO, 1998), que inclui um passeio em Realidade Virtual pelo MAC - Museu de Arte Contemporânea de Niterói – RJ. O Usuário pode passear pelos ambientes do museu, obtendo informações sobre sua arquitetura.

**Aspectos tecnológicos:** Foi desenvolvido em Borland C++Builder (BORLAND, 2001). O passeio em RV foi desenvolvido utilizando o pacote WorldUP (SENSE8, 2001) da empresa Sense8, integrado a aplicação multimídia desenvolvida em C++Builder.

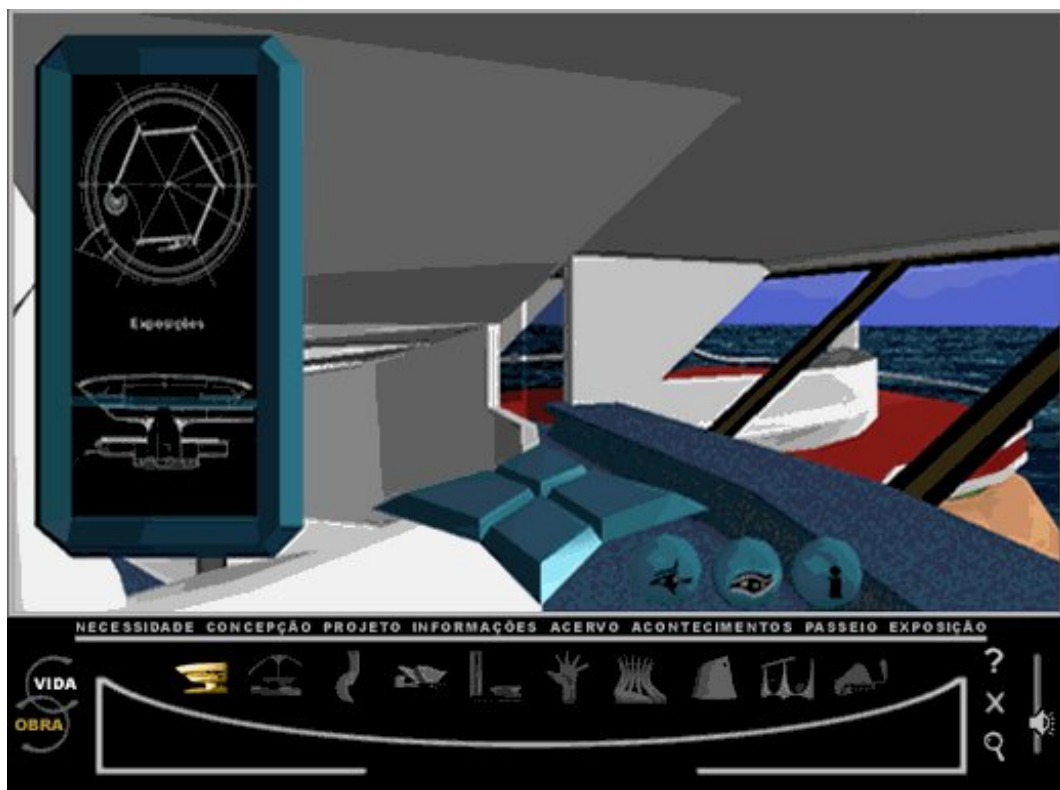


Figura 12 - Captura de tela, CD-ROM Oscar Niemeyer, Vida e Obra.

### 2.5.5 Comércio eletrônico

**Projeto:** StarLight Express (LUNATIC, 2003)

**Desenvolvido por:** Lunatic Interactive e Intel Europe para *Stella AG*.

**Objetivo:** Possibilitar a comprar de ingressos para o espetáculo StarLight Express através de um ambiente virtual, composto de uma representação virtual tridimensional do local onde ocorrerá o espetáculo. Através desta representação o cliente pode escolher seu assento e avaliar qual visão terá do palco, como se estivesse realmente lá.

**Aspectos tecnológicos:** Utiliza a linguagem VRML associada às linguagens HTML e JAVA<sup>26</sup> em uma aplicação típica para WEB.



Figura 13 - Captura de tela, StarLight Express.

Os exemplos apresentados são apenas uma pequena amostra ilustrativa de aplicações que utilizam técnicas de Realidade Virtual. Mas o suficiente para demonstrar sua aplicabilidade.

<sup>26</sup> Linguagem muito utilizada na web, principalmente em páginas com conteúdo dinâmico.

### 3. AMBIENTES VIRTUAIS

Ivan Sutherland (1965) introduziu o conceito de inserir pessoas em ambientes gerados por computador, e concretizou este conceito através da projeção de modelos tridimensionais em *wireframes*<sup>27</sup>, utilizando uma tela montada em um tipo de capacete (HMD), hoje um dos mais conhecidos dispositivos de RV.

Uma terminologia própria foi surgindo para definir o conceito de inserção de pessoas em ambientes sintéticos e seus modelos de interação, foram definidos termos como ambiente virtual, espaço cibernético, virtualização, presença virtual, entre outros. Alguns termos se confundem e outros possuem definições diferenciadas, mas mantendo uma congruência.

William Gibson (1984) fez uma descrição fictícia de **Ambiente Virtual**<sup>28</sup> em seu livro *Neuromancer*, o que ele chamou de **Espaço Cibernético**<sup>29</sup>:

*"Espaço Cibernético. Uma alucinação consensual experimentada diariamente por um bilhão de legítimos operadores, em toda a nação, por crianças que estão aprendendo conceitos matemáticos (...) Uma representação gráfica de dados abstraídos dos bancos de dados de cada computador no sistema humano. Uma complexidade inconcebível. Linhas de luz percorrem o vazio da mente, agrupamentos e constelações de dados. Como luzes de cidade, retrocedendo (...)".*<sup>30</sup>

Benedikt (1991) fez uma outra descrição do que seria Espaço Cibernético:

*"Espaço cibernético é globalmente ligado, computacionalmente gerado e sustentado, uma realidade multidimensional artificial ou virtual. Neste mundo sobre o qual toda tela de computador é uma janela, atual, onde a distância geográfica é irrelevante. Objetos vistos ou ouvidos não são físicos, nem, necessariamente, representações de objetos físicos, mas suficientes - em forma, caráter, e ação - compostos de dados, de informação pura. Esta informação é derivada em parte das operações do mundo natural, físico, mas é derivada principalmente do imenso tráfico de informações simbólicas, imagens, sons, e pessoas que constituem o empreendimento humano, em ciência, arte, negócio e cultura."*<sup>31</sup>

---

<sup>27</sup> Modelagem onde se utiliza uma malha de pontos ligados por linhas para representar tridimensionalmente um objeto de forma volumétrica.

<sup>28</sup> Do termo original *Virtual Environment (VE)*.

<sup>29</sup> Do termo original *Cyberspace*.

<sup>30</sup> Tradução livre do autor.

<sup>31</sup> Tradução livre do autor.



Steve Ellis (1991) definiu o termo **virtualização**, como sendo o processo pelo qual um espectador humano interpreta uma impressão sensória moldada para ser um objeto estendido em um ambiente diferente do qual ele existe fisicamente.

Ellis também dividiu o processo em três níveis:

- Espaço Virtual - Onde será projetada a imagem virtual;
- Imagem Virtual - Percepção de profundidade dos objetos;
- Ambiente Virtual - O usuário torna-se parte do mundo virtual.

Kalawsky (1993) definiu Ambiente Virtual como sendo uma experiência sensória sintética que comunica componentes físicos e abstratos a um operador humano ou participante. Esta experiência sensória sintética é gerada por um sistema computacional que um dia poderá apresentar-se como uma interface ao sistema sensório humano indistinguível do mundo físico real.

Segundo Sheridan (1994), a **presença virtual** é experimentada por uma pessoa quando esta recebe informações sensoriais geradas por um computador ou sistema computacional, e este compele um sentimento de estar presente em um ambiente diferente daquele onde a pessoa de fato esta.

Como se pode concluir, uma grande variedade e quantidade de ambientes e situações contextualizadas podem ser ajustadas de alguma forma aos conceitos de Ambiente Virtual, partindo das comunidades ambientadas em programas de comunicação textual, passando por ambientes de simulação em escala real, e chegando até aos ambientes sintéticos tridimensionais em rede.

No contexto computacional Ambiente Virtual (AV) é um ambiente sintético composto por um conjunto de elementos que estimulam os sentidos do usuário, causando a presença virtual. Compreende uma representação, que consiste na modelagem do ambiente e no conjunto de agentes de efeito compostos de interfaces físicas e lógicas que interagem com esta representação. Este é o contexto de AV que o autor utilizará no decorrer deste trabalho.

### 3.1 Mundos e Comunidades Virtuais

Comunidade virtual, um grupo de pessoas conectadas por um compromisso comum, ações mútuas, valores, interesses ou metas. Na verdade este é um conceito geral de comunidade e o termo virtual é acrescentado para diferenciar sua natureza, ou seu campo de atuação, que é o espaço virtual.

Mundo virtual, espaço virtual compartilhado que é visitado periodicamente por pessoas que possuem interesses comuns como alguma atividade, uma meta, um idioma ou uma ligação ética.

Uma comunidade virtual pode ou não formar um mundo virtual, comunidades virtuais de tamanhos e características variadas podem vir a fazer parte de um único mundo virtual onde seus membros podem fazer parte de comunidades diversas, interagindo entre si e com o meio. Tem-se então que um mundo virtual é uma construção virtual baseada nos mesmos princípios que norteiam nosso mundo real.

Nas comunidades virtuais tendem a surgir os mesmos agentes de regulamentação que temos no ambiente real, tais como: liderança, organização, priorização, domínio, etc. Os conceitos de mundo virtual e comunidade virtual surgiram através da literatura de ficção, como nas obras *Neuromancer* (GIBSON, 1984) e *Snowcrash* (STEPHENSON, 1992) e em aplicações do como o *Habitat* (MORNINGSTAR, 1991) e *MUD*<sup>32</sup> (CURTIS, 1992).

Inicialmente estes mundos rodavam em computadores padrões para a época, não requerendo nenhum hardware gráfico especial, ou mais que uma conexão de rede de baixa velocidade. Foram projetados principalmente, para comunicação e formação de comunidades de interesse, não representando ou visualizando ambientes gráficos.

Os Sistemas de Realidade Virtual neste período também não focalizavam a incorporação de representações dos usuários e freqüentemente eram aplicações monousuário.

A seguir, descreve-se um mundo virtual denominado *Habitat*, que teve um papel importante na evolução dos ambientes virtuais.

---

<sup>32</sup> MUD é a abreviação de *Multi-User Dungeons*. É um jogo de contar histórias (RPG) que pode ser jogado pela Internet, por vários jogadores ao mesmo tempo, todos interagindo entre si. Um MUD corresponde a um Mundo virtual. É um local que foi criado com as características próprias ao tipo de ambiente em que ele foi imaginado. Os personagens têm vida própria e atuam de forma explícita e direta.

### 3.1.1 Habitat

Habitat (MORNINGSTAR, 1991) foi o primeiro mundo virtual no qual as pessoas eram representadas como avatares e tinham a capacidade de se comunicar e formar uma comunidade virtual. Em 1985 rodava em um microcomputador Comodore64.

Uma cena típica de Habitat pode ser vista na figura 14.



Figura 14 - Cena típica do Habitat.

Fonte: LucasArts Entertainment Company.

Lucasfilm's Habitat foi criado pela Lucasfilm Games, uma divisão da LucasArts Entertainment Company em associação com a Quantum Computer Services, Inc.

Foi uma das primeiras tentativas de criação de um ambiente virtual multiusuário de larga escala com fins comerciais. Uma pequena iniciativa, perto dos muitos esforços em pesquisas de laboratório na época, baseados em interfaces de hardware sofisticadas e dezenas de milhares de dólares por usuário em poderosos computadores dedicados.

Habitat foi desenvolvido para rodar sob um serviço *on-line* comercial comum, e usava um computador pessoal de baixo custo, chamado por alguns de "brinquedo", para apoiar a interação do usuário. Apesar de aparentemente simples, Habitat é ambicioso em sua concepção, o sistema desenvolvido pode suportar uma população de milhares de usuários em um único ambiente cibernético compartilhado.

Habitat apresentava seus usuários em forma de desenho animado em tempo real, em um mundo simulado *on-line*, no qual era possível a comunicação textual, disputa de jogos, participação em aventuras, namorar, casar, iniciar negócios, entre outras interações.

O projeto Habitat provou ser uma fonte valiosa de informações na tarefa de implementar um ambiente virtual sério e comercialmente viável. A lição essencial que pode ser extraída deste projeto é que o espaço cibernético ou ambiente virtual é definido mais pelas interações entre seus atores do que pela tecnologia utilizada em sua implementação.

A principal idéia do espaço cibernético é que ele deve ser necessariamente um ambiente multiusuário. Os elementos importantes aos habitantes de tal ambiente são as capacidades disponíveis neste sistema, tais como, características dos usuários, modos de interação e integração entre usuários e com o meio. O Habitat original rodou durante seis anos no Japão e nos Estados Unidos, mas atualmente não se encontra mais ativo.

### 3.2 Ambientes Virtuais em Rede

Um Ambiente Virtual em Rede (AVR)<sup>33</sup> (MACEDONIA, 1997) é um sistema em que múltiplos usuários interagem entre si em tempo real, com a possibilidade de estarem localizados fisicamente em qualquer lugar. Este ambiente proporciona ao usuário um senso de imersão através da incorporação de gráficos tridimensionais, som estéreo (SINGHAL, 1999) e em alguns casos também com a utilização de dispositivos imersivos especiais, para captura de movimentos e resposta sensória.

Os AVR são distinguidos por cinco características comuns:

**a) Sensação de espaço compartilhado** – Todos os usuários do ambiente possuem uma sensação de estar no mesmo espaço compartilhando as mesmas sensações espaciais, como por exemplo, se o espaço virtual é composto por uma praça com jardim, a mesma praça deve ser observada por todos os usuários.

**b) Sensação de presença** – Todos os usuários devem ser visualmente disponíveis para os outros de alguma forma, a qual é comumente denominada de *avatar*<sup>34</sup> (DAMER, 1998). O avatar normalmente possui uma forma

---

<sup>33</sup> Do termo original: *Networked Virtual Environment* (Net-VE).

<sup>34</sup> Mais detalhes no item 3.2.3 deste trabalho.

geométrica qualquer (humanóide ou não) e pode possuir diversos graus de interação, onde a presença de um usuário deve ser notada pelos outros, seja através de modos de comunicação ou por intervenções no ambiente. A movimentação do avatar pelo ambiente contribui de forma decisiva para a efetiva sensação de presença.

**c) Sensação de tempo** – Os usuários devem ter noção de eventos temporais, isto é, as interações devem ser regidas pelo tempo, existindo passado, presente e futuro no ambiente. As interações devem ocorrer idealmente em tempo real, sendo que os usuários devem observar os comportamentos e intervenções dos outros usuários no mesmo momento em que forem geradas.

**d) Modos de comunicação** – A forma de comunicação entre os usuários é essencial em um ambiente virtual distribuído, e pode ser realizada de diversas formas, seja através de comunicação textual, gesto, ou em alguns casos através da voz do próprio usuário ou voz sintetizada. Os avatares também colaboram nos modos de comunicação quando possuem a capacidade de exprimir sentimentos e características do usuário através da representação gráfica no ambiente, como gestos, e ações de intervenção no meio.

**e) Modos de compartilhamento** – Talvez a mais interessante característica de um ambiente virtual em rede seja a possibilidade dos usuários interagirem com os outros e com o próprio ambiente. Eles podem ativar elementos, criar ou destruir partes do ambiente, entre outras possibilidades. Um exemplo deste compartilhamento é a ativação de uma calculadora, onde qualquer um dos usuários ativos no ambiente poderia efetuar operações de modo compartilhado clicando nas teclas da calculadora. Trocas de dados (arquivos) também são formas de compartilhamento possíveis nestes ambientes.

A habilidade de compartilhar objetos diferencia o AVR das tradicionais salas de bate-papo, e a interatividade em tempo real o diferencia das tradicionais navegações na rede ou do correio eletrônico. AVR é a solução mais apropriada para aplicações que necessitem da criação de **telepresença** (MAIR, 1995), pois simula um contato face a face.

São sistemas notoriamente de difícil implementação, complexos e possuem características de diversos sistemas tradicionais. Por exemplo, sistemas distribuídos, aplicações gráficas e aplicações interativas. Uma outra denominação comum a estes sistemas é a de Realidade Virtual Distribuída.

Muitos aspectos dos AVR são objetos de pesquisa continuada e apresentam resultados interessantes, tais como: presença (SCHUEMIE, 1999), colaboração (SHIRMOHAMMADI, 2000) e animação em tempo real (BABSKI, 2000a).

A visão de AVR apresentada e desenvolvida nesta dissertação é focada em sistemas de RV baseados em sistemas gráficos tridimensionais interativos, mas o princípio de AVR é mais amplo e outras perspectivas são igualmente exploradas

(ÇAPIN, 1999a), por exemplo, o já citado MUD (BELLMAN, 1997; CURTIS, 1992) faz uso de uma representação textual descritiva de ambientes virtuais em rede. Este tipo de representação de AVR não será detalhada neste trabalho.

### 3.2.1 Comunicação e Distribuição de Dados em AVR

Dois aspectos da maior importância (MCLEAN, 1997) no desenvolvimento de ambientes virtuais são a topologia<sup>35</sup> de comunicação e a forma de distribuição de dados. Duas topologias de comunicação comumente utilizadas são:

**a) Centralizada:** Esta é uma forma centralizada de rede, onde as informações (mensagens ou pacotes de dados) são enviadas para uma máquina central (Servidor) que por sua vez as distribui para as máquinas clientes (Figura 15a).

**b) Distribuída:** Cada máquina cliente possui um canal de comunicação com outra máquina cliente, e as informações são diretamente trocadas entre as máquinas da rede. (Figura 15b).

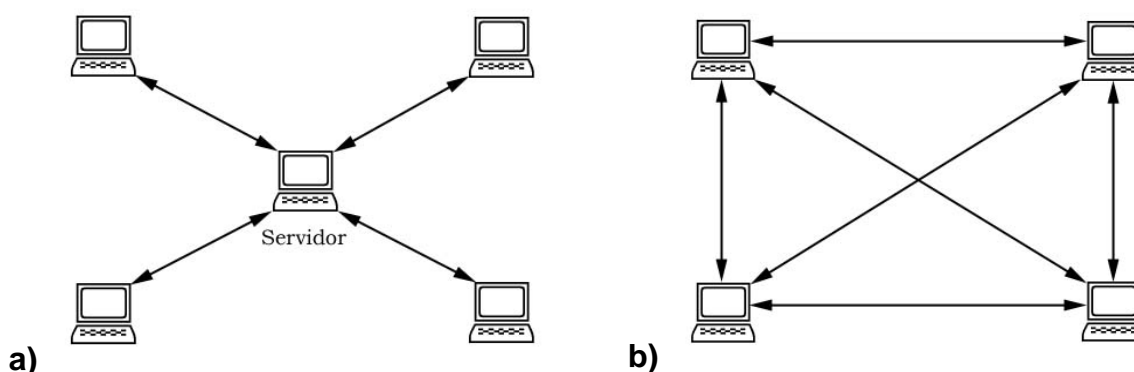


Figura 15 - Topologias de comunicação.

A distribuição de dados globais define onde os dados estão armazenados ou a maneira como são compartilhados, Bangay (1996) categoriza a distribuição de dados em:

**a) Banco de dados centralizado:** Os dados são armazenados em uma máquina central (Servidor). Os dados são atualizados nesta máquina central que é responsável por enviar informações atualizadas às máquinas cliente quando necessário.

**b) Banco de dados distribuído:** Os dados são armazenados em diversas máquinas, sendo sincronizados constantemente.

**c) Banco de dados replicado:** Os dados são armazenados em diversas

<sup>35</sup> Estudo sobre a colocação ou disposição.

máquinas e atualizados periodicamente para manter uma consistência razoável dos dados.

**d) Base de dados distribuída:** O Banco de dados é dividido em servidores múltiplos. Uma sincronização completa é uma possibilidade improvável de ser considerada.

**e) Base de dados replicada:** O Banco de dados é dividido em servidores múltiplos com segmentos de dados replicados em múltiplas máquinas.

Algumas combinações de topologias de comunicação e compartilhamento de dados (MCLEAN, 1997) são relevantes nos AVR:

**a) Banco de Dados centralizado com topologia de comunicação centralizada:** Pode ser representado como na figura 15a, com os dados armazenados na máquina central (Servidor). Não é uma arquitetura muito expansível, principalmente em ambientes com grandes quantidades de dados como sistemas de RV em tempo real. O servidor se torna o gargalo do sistema. É uma arquitetura viável para ambientes virtuais textuais, como por exemplo, os MUD.

**b) Banco de Dados replicado com topologia de comunicação centralizada:** Neste caso os dados do ambiente são replicados em todas as máquinas e a comunicação é mediada por uma única máquina (Servidor), como pode ser visto na figura 16. O problema nesta arquitetura é que o servidor central pode se tornar o gargalo do sistema, já que o servidor tem que receber e enviar as informações de atualização para todas as máquinas clientes. Em um sistema com muitos usuários, ou com um intenso fluxo de atualização isto pode se tornar bem complicado.

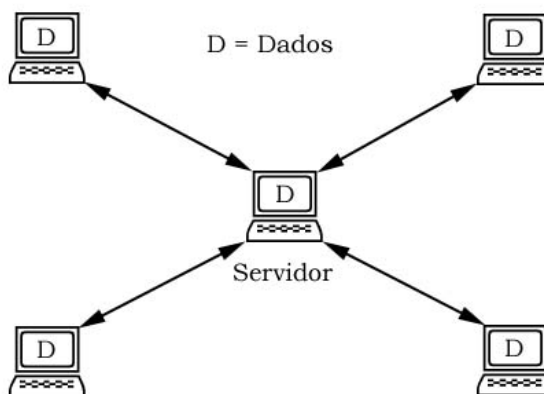


Figura 16 - Banco de Dados replicado com comunicação centralizada.

**c) Banco de Dados replicado com topologia de comunicação distribuída:** Cada participante pode se comunicar com outro sem precisar passar por um servidor central. Cada participante possui uma cópia dos dados do ambiente compartilhado. A figura 17 ilustra tal arquitetura. *Multicasting*<sup>36</sup> (GARCIA, 2003) seria o modo ideal para atualizações em um sistema desse tipo. Porém, a

<sup>36</sup> Multicasting é um método ou técnica de transmissão de um pacote de dados para múltiplos destinos ao mesmo tempo.

arquitetura selecionada necessita de alguma fonte de controle ou autoridade, que é exercida pelo servidor central nas arquiteturas apresentadas anteriormente.

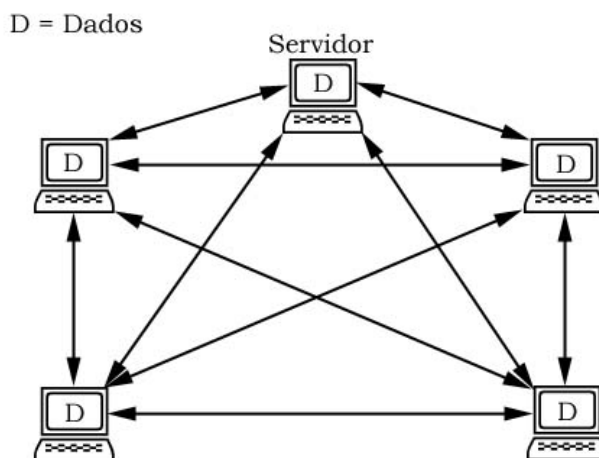


Figura 17 - Banco de Dados replicado com comunicação distribuída.

Também se torna necessário um ponto de armazenamento de dados, pois os participantes neste tipo de sistema podem entrar e sair a qualquer momento. Por isto é necessário que um dos participantes tenha ou assuma o papel de servidor no sistema, controlando a existência e consistência dos dados, administrando as entradas e saídas de participantes.

**d) Base de dados replicada com topologia de comunicação distribuída:**

Neste caso o banco de dados é dividido em múltiplos servidores, cada servidor é responsável por manter um segmento do ambiente virtual compartilhado. Quando um determinado participante deseja navegar em outro segmento, este se conectará no servidor responsável pelo segmento desejado, passando a fazer parte da rede composta por todos os participantes conectados no novo servidor. Este modelo permite criar ambientes em escala e maciçamente conectados, com inúmeros servidores administrando partes de ambientes diversos. A figura 18 representa esta arquitetura.

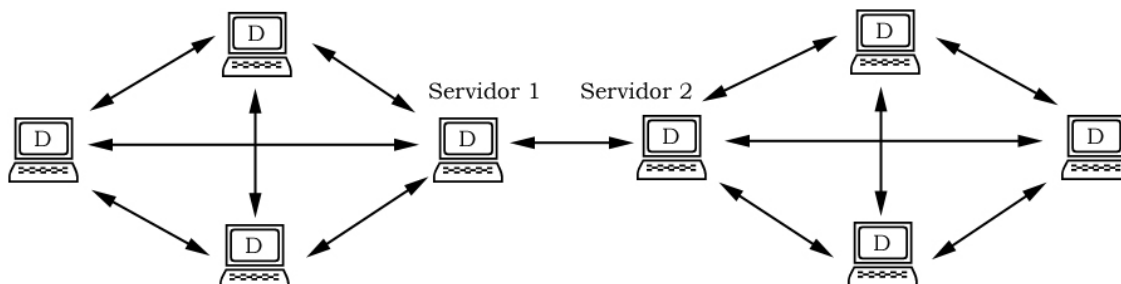


Figura 18 - Base de dados replicada com comunicação distribuída.



### 3.2.2 Desenvolvimento de Ambientes Virtuais em Rede

O desenvolvimento de ambientes virtuais é apresentado por Çapin (1999a) dividido em três elementos:

**1) Pré-condições:** São elementos os quais o projetista possui o mínimo de controle, e são definidas por condições iniciais pré-estabelecidas. Os elementos envolvidos nesta etapa são, por exemplo:

- O Tipo de aplicação alvo, isto é que tipo de serviço será oferecido e qual o público que se pretende atender;
- Qual infra-estrutura de rede vai suportar a aplicação desenvolvida, sua velocidade e qual seu atraso padrão (tempo entre o envio de um pacote de dados e o seu recebimento por outra máquina) da rede, quantidade esperada de clientes atendidos;
- Quais dispositivos de entrada e saída de dados serão utilizados pelos clientes do sistema.

**2) Escolhas de projeto:** Inclui ferramentas e parâmetros que o projetista pode controlar. Os elementos envolvidos nesta etapa são, por exemplo:

- Qual arquitetura<sup>37</sup> cliente será adotada;
- Esquema de distribuição de dados e tarefas;
- Topologia de rede (vide item 3.2.1);
- Como os participantes serão incorporados ao sistema, que tipo de representação dos usuários será adotada;
- Definição e padronização do protocolo de comunicação.

**3) Melhorias futuras:** Inclui ferramentas e técnicas que o projetista pode explorar para aumentar a eficiência, qualidade de um AVR. Os elementos envolvidos nesta etapa são, por exemplo:

- Colisão<sup>38</sup> entre objetos e participantes;
- Gerenciamento de permissões de compartilhamento de dados;
- Uso de dispositivos de entrada e saída do tipo *force-feedback*;
- Variação no nível de detalhamento dos ambientes e objetos;
- Utilização de filtros de mensagens para otimizar o uso da rede;
- Predicação de movimento e compressão de dados;
- Refinamento de algoritmos de sincronização.

---

<sup>37</sup> A palavra "Arquitetura" em computação relaciona-se diretamente com a funcionalidade dos computadores, isto é, o estudo de quais são as suas partes (CPU, memórias e periféricos), como funcionam.

<sup>38</sup> Tratamento de colisão em AVR tridimensionais é responsável em evitar que dois objetos tridimensionais ocupem o mesmo espaço ao mesmo tempo, da mesma forma que "colidimos" com uma parede no mundo real.

Os principais fatores no desenvolvimento de sistemas distribuídos em RV foram reunidos por Rodello (2001, p. 322) e são apresentados no quadro 1.

**Quadro 1 - Fatores no desenvolvimento de sistemas distribuídos.**

<b>Fatores</b>	<b>Descrição</b>
Número de usuários.	Quantidade de usuários que o ambiente pode suportar de forma simultânea.
Número e tamanho dos mundos virtuais.	Quantidade e tamanho de mundos que podem estar ativos no sistema.
Interconexão dos mundos.	Forma com que os diferentes mundos estão limitados e como o usuário ultrapassa esses limites.
Números de objetos.	Quantidade de objetos presente no ambiente.
Nível de detalhes dos objetos.	Nível de detalhamento presente nos objetos: textura, geometria, iluminação, etc.
Comportamento dos objetos.	Comportamento dos objetos dentro do ambiente virtual: Animação, interatividade, etc.
Interação dos usuários com o mundo virtual.	Operações disponíveis para que o usuário possa interagir com o mundo virtual e com os objetos.
Interações entre usuários.	Capacidade dos usuários interagirem entre si.
<b>Modelo de Comunicação</b>	<b>Descrição</b>
Modelo Cliente / Servidor (CS).	Comunicação direta somente do cliente com o servidor e vice-versa.
Modelo Ponto a ponto (P2P).	Comunicação direta entre clientes, sem intermédio de um ou mais servidores.
Computação distribuída orientada a objetos.	Comunicação do cliente com o serviço de objetos do solicitador e do receptor de serviços que são intermediados por um agente ( <i>broker</i> ).
Sistemas híbridos.	Combinação do modelo CS e P2P.
<b>Modelo de Armazenamento</b>	<b>Descrição</b>
Banco de dados centralizado e compartilhado.	Os dados estão centralizados em uma única máquina, que atua como servidor.
Banco de dados replicados de mundos homogêneos.	Cada máquina cliente conectada na aplicação possui uma réplica completa dos dados do ambiente.
Banco de dados distribuído e compartilhado.	Cada máquina cliente possui apenas uma parte do ambiente em sua base de dados.
<b>Avaliação de desempenho</b>	<b>Descrição</b>
Confiabilidade.	Quantidade de dados que são perdidos pela rede durante o trajeto do cliente origem para o cliente destino.
Atraso.	Tempo decorrido entre a atuação do usuário no sistema e a visualização do resultado na aplicação.
Escala.	Análise da capacidade de crescimento de um sistema em termos de usuários conectados ao mundo.
<b>Modelo de suporte para comunicação</b>	<b>Descrição</b>
<i>Sockets</i> .	Primitivas que estabelecem um canal de comunicação interprocessos por meio de chamadas de sistemas.
<i>Toolkits</i> .	Modelos desenvolvidos especificamente para um conjunto de desenvolvimento.
<i>Middlewares</i> .	Modelos independentes de plataforma.

Os três elementos de desenvolvimento: Pré-condições, Escolhas de projeto e Melhorias futuras são básicos no desenvolvimento de qualquer projeto e em conjunto com os fatores apresentados no quadro 1 formam as diretrizes de desenvolvimento adotadas pelo autor no desenvolvimento do sistema proposto nesta dissertação.

### 3.2.3 Avatar

Segundo definição baseada no antigo Vedas Indiano (uma das mais antigas literaturas espiritualistas), a palavra **avatar** é de origem antiga e significa “a descida de Deus” ou simplesmente “encarnação”. Em Inglês a palavra adquire o significado de “uma incorporação, uma manifestação corporal do Divino” (AVATARA, 2001).

Neal Stephenson (1992, p. 35) cita em seu livro de ficção *Snow Crash* o que seria avatar de uma forma interessante:

*"Hiro se aproxima da rua, ele vê dois casais de jovens, provavelmente usando os computadores de seus pais para um encontro de namoro no Metaverse, descendo da Porta Zero, que é o porto local de entrada e parada do monotrilho. Ele não está vendo pessoas reais, certamente. Isto é tudo parte da simulação desenhada por seu computador de acordo com as especificações que chegam pelo cabo de fibra óptica. As pessoas são peças de software chamadas avatares".<sup>39</sup>*

Em termos de RV, avatar é uma representação ou incorporação do usuário em um ambiente virtual, que é percebida pelos outros usuários deste ambiente. O usuário, por sua vez, também verá os outros usuários como avatares. Esta definição foi utilizada pela primeira vez por Chip Morningstar em 1985 quando sugeriu a utilização do termo avatar (DAMER, 1998).

Muitos avatares possuem ações e/ou gestos pré-definidos que podem ser disparados por eventos. Estes eventos são gerados pelo ambiente virtual ou pelo próprio usuário. Os gestos disponíveis são ações ou emoções tais como andar, dançar e sorrir. Gesticular não é o único modo de comunicar-se em espaços multiusuário. Os sistemas possuem comumente um sistema de comunicação via

---

<sup>39</sup> Tradução livre do autor.

texto, como o IRC. Existe freqüentemente, várias conversas em paralelo, os ambientes são complexos e localizar-se neste ambiente, dominando a ferramenta é algo que se adquire com a prática.

A escolha do avatar é freqüentemente feita pelo usuário, a qual dependendo do sistema que estiver sendo utilizado pode ser representações realísticas de seres humanos, caricaturas, ou qualquer representação abstrata. A figura 19 exemplifica um avatar caricato e a figura 20 um avatar realístico.



Figura 19 - Avatar caricato.



Figura 20 - Avatar realístico.

Além de fornecer um senso de presença em ambientes virtuais, o avatar é utilizado no auxílio da comunicação não verbal. Este assunto será discutido no item 3.2.4.

Diversos tópicos relacionados a avatares estão atualmente em estudo, tais como: transmissão em fluxo contínuo (*streaming*) de animações via rede (ÇAPIN, 1999b), animação facial (BABSKI, 2000b), expressões corporais (EMERING, 1999), animação em tempo real e captura de movimentos (BABSKI, 2000a).

A pesquisa de avatares humanóides, como o visto na figura 20 levou a criação de um padrão de movimentação chamado de H-Anim (H-ANIM, 2001), que é desenvolvido e mantido dentro de um grupo de trabalho do Web3D Consortium (WEB3D, 2003). O foco desta especificação é o VRML, mas seus princípios podem ser aplicados a outros sistemas.

H-Anim é baseado na modelagem das dimensões antropométricas e nos pontos de articulação, definindo modelos de esqueleto humanóide com diversos

graus de detalhamento.

Atualmente a especificação em desenvolvimento, chamada de versão 2001, foi incorporada a especificação da nova versão da VRML, chamada de X3D (WEB3D, 2003), que aguarda publicação definitiva como uma norma padrão ISO<sup>40</sup>.

A utilização de avatares é essencial para o estímulo visual e provê uma excelente representação do usuário em AVR, e esta é a forma pela qual os usuários serão representados no sistema desenvolvido nesta dissertação.

### 3.2.4 Comunicação não Verbal em AVR

Em AVR a comunicação é essencial e não se restringe a comunicação lingüística ou verbal (texto ou voz), mas busca uma análise das variantes não verbais da comunicação.

Com a utilização de RV existe uma representação complexa (através dos modelos tridimensionais, animações e sons) do ambiente e de seus usuários. Portanto deve-se explorar esta representação do usuário (avatar) como forma de comunicação. Explorando-se então os diversos canais pelos quais a informação é trocada entre indivíduos em uma interação face a face real, chega-se a utilização da comunicação não verbal em Ambientes Virtuais (MANIA, 1998).

Algumas formas de comunicação não verbal são explorados em AVR, tais como:

**a) Expressões faciais:** São muito representativas, indicando emoções e aspectos da personalidade do interlocutor. Sinaliza itens como o nível de interesse no assunto, descrença e surpresa, se devidamente sincronizada com a comunicação. Em AVR podem ser utilizados conjuntos de expressões pré-definidas associadas ao avatar do usuário, ou mesmo a utilização de equipamentos de captura de movimento facial<sup>41</sup>.

---

<sup>40</sup> ISO é uma abreviação de *International Organization for Standardization* (Organização Internacional para Padronizações).

<sup>41</sup> São equipamentos que podem capturar o movimento de pontos do rosto do usuário, por exemplo, utilizando reconhecimento de padrões sob uma imagem capturada por uma câmera de vídeo.

**b) Olhar:** O olhar é um indicador geral de atenção, o qual representa o interesse do usuário em algum aspecto específico do ambiente ou disponibilidade de conversação. Também é utilizado como forma de sincronização da comunicação verbal em interações face a face. Em AVR a representação do olhar através do avatar indica de alguma forma, a direção e sentido de interesse assumido pelo usuário. As representações auxiliares ficam por conta dos movimentos dos olhos e sobrancelhas.

**c) Gestos:** Mãos, cabeça e pés podem produzir uma vasta gama de gestos. São coordenados com a comunicação verbal e suportam varias funções de comunicação. Eles são usados como referencia a conteúdo, tendo um excelente alcance social. Podem ser utilizados gestos para indicar localizações físicas no ambiente, sejam pontos de referencia ou objetos, para demonstrar algum processo ou comportamento. A mobilidade também permite que o avatar represente os gestos dentro do AVR, sejam eles pré-definidos ou gerados pelo processo de captura de movimentos corporais<sup>42</sup>.

**d) Postura:** É um indicativo de comportamento social associado a estados emocionais através de representações corporais. A posição e orientação do corpo podem ser utilizadas para incluir ou excluir as pessoas em uma conversação de grupo. É representado em AVR diretamente pela posição e orientação do avatar.

**e) Ego-representação:** Esta diretamente associada à representação pessoal, isto é, indica vários estados físicos e emocionais, por exemplo: estado civil, ocupação, personalidade, agressividade, formalidade, etc. Em AVR esta representação é feita através da própria representação física do avatar (geometria, cor, textura, etc.) e por representações auxiliares indicando atividade ou condições especiais. Por exemplo, indicar que o usuário se encontra ocupado em alguma atividade ou indisponível para comunicação direta.

---

<sup>42</sup> São equipamentos que capturam os movimentos de partes do corpo, por exemplo, através de sensores magnéticos, que indicam posição e orientação dos pontos do corpo onde estão colocados.

**e) Contato Corporal:** O toque físico parece ter um significado primitivo de intimidade e produz um estímulo emocional aumentado. Esta forma de comunicações é utilizada para exprimir sentimentos ou alguma forma de saudação. O significado preciso de uma forma particular de toque depende da cultura. Normalmente não é representada em AVR, pois necessita obrigatoriamente de equipamento adicional. Mas em sistemas imersivos podem ser incluídas utilizando, por exemplo, luvas sensíveis ao toque.

No sistema desenvolvido pelo autor os itens de comunicação não verbal se concentram em gestos, postura e ego-representação, todos associados diretamente à representação do usuário no sistema, ou seja, através do avatar. Entretanto as outras formas de comunicação não verbal não estão excluídas do sistema, apenas não serão desenvolvidas no protótipo proposto, ficando sua implementação para trabalhos futuros.

### 3.3 Exemplos de Ambientes Virtuais para Internet

Através de uma pesquisa básica foram obtidas informações sobre alguns dos produtos ou projetos desenvolvidos por outros grupos espalhados pelo mundo.

Todos os itens apresentados têm como objetivo primário à formação de comunidades virtuais, sendo que alguns deles possuem características comerciais, isto é visam também à venda de produtos ou serviços. Esta característica depende também de como os ambientes disponibilizados nos produtos são criados. Um mesmo produto pode oferecer um ambiente educacional, e um ambiente comercial, sem que isto altere suas características básicas.

A seguir, apresenta-se uma listagem com algumas aplicações de ambientes virtuais para Internet, sendo destacada a aplicação denominada *There*, por ser o mais atual e completo exemplo de ambiente virtual para Internet: O *There* pode ser considerado uma das maiores iniciativas comerciais, no campo de ambientes virtuais para Internet, nenhum outro projeto anterior teve disponível equivalente montante de investimentos e pessoal.

### 3.3.1 ActiveWorlds

Usando o *ActiveWorlds* (ACTIVEWORLDS, 2001) (Figura 21) pode-se visitar cidades virtuais e interagir com outros cidadãos através de texto, além da possibilidade de construir ambientes. Alguns usuários do universo virtual podem espalhar diversos objetos no ambiente. Possui também características como música e outros efeitos sonoros, permitindo até que o usuário dance utilizando avatares articulados. Pode-se visitar os mundos criados como um usuário turista, ou pagar uma taxa anual para se tornar um cidadão e funciona como aplicação independente, isto é não roda dentro de um *Web browser*.

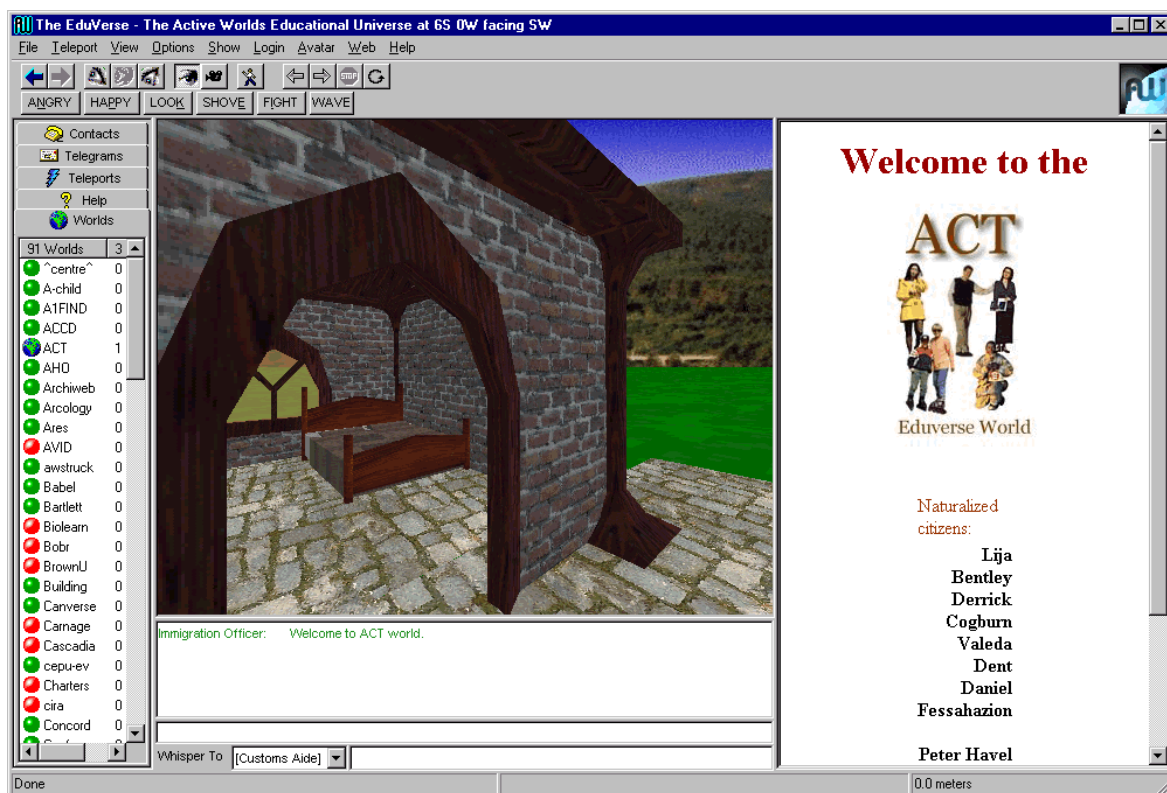


Figura 21 - Captura de tela do ActiveWorlds.



### 3.3.2 CyCo Entrance

*CyCo Entrance* (CYCO, 2001) (Figura 22) é um Chat 3D que utiliza o poder do motor gráfico *CyCo*. Possui *download*, *cache*<sup>43</sup> e *streaming*<sup>44</sup> inteligentes, compressão de dados de alta qualidade, e é altamente integrado aos principais *Web browsers*, o que permite uma instalação automática e fácil.

Altamente extensíveis permitindo um rápido desenvolvimento de conteúdo ativo, os ambientes podem facilmente mudar diariamente.



Figura 22 - Captura de tela do Cyco Entrance.

<sup>43</sup> Armazenamento local de arquivos do ambiente.

<sup>44</sup> Transmissão de arquivos e dados de forma contínua.

### 3.3.3 Blaxxun Community Plataforma

Os mundos da *Blaxxun* (BLAXXUN, 2001) (Figura 23) são construídos em VRML. Como VRML é uma arquitetura aberta, isto significa que é possível construir um mundo e/ou avatar próprio e conectá-los aos mundos da *blaxxun*. A conversa pode acontecer via texto, *cooltalk voice* (se estiver instalado dentro do *Web browser Netscape*) ou até mesmo iniciar o *Microsoft NetMeeting* para falar com outros cidadãos. Pode-se usar a visão em primeira ou terceira pessoa. Funciona dentro de um *Web browser*, utilizando o visualizador VRML da própria *Blaxxun* e um mini aplicativo (aplet) JAVA.

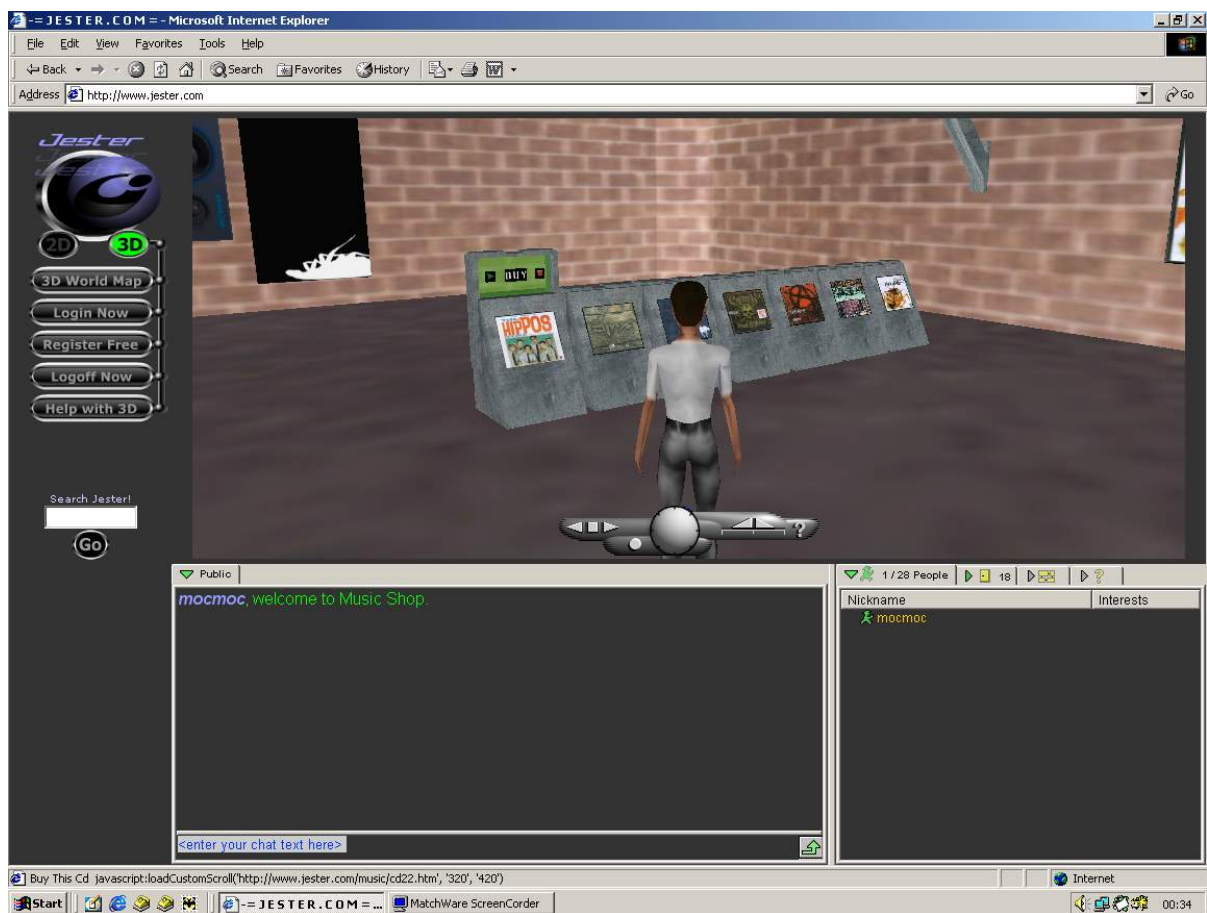


Figura 23 - Captura de tela do Blaxxun Contact.

### 3.3.4 ParallelGraphics Island

Este é o cliente multiusuário da *ParallelGraphics* (PARALLELGRAPHICS, 2001) (Figura 24). É produzido utilizando o mesmo SDK que foi utilizado no desenvolvimento do protótipo de AVR apresentado nesta dissertação. Utiliza avatares tridimensionais com movimentos e gestos, conversação em modo texto, múltiplos ambientes e inclusão de objetos no ambiente. Utiliza arquivos VRML como base e funciona como aplicação independente, isto é não roda dentro de um *Web browser*.

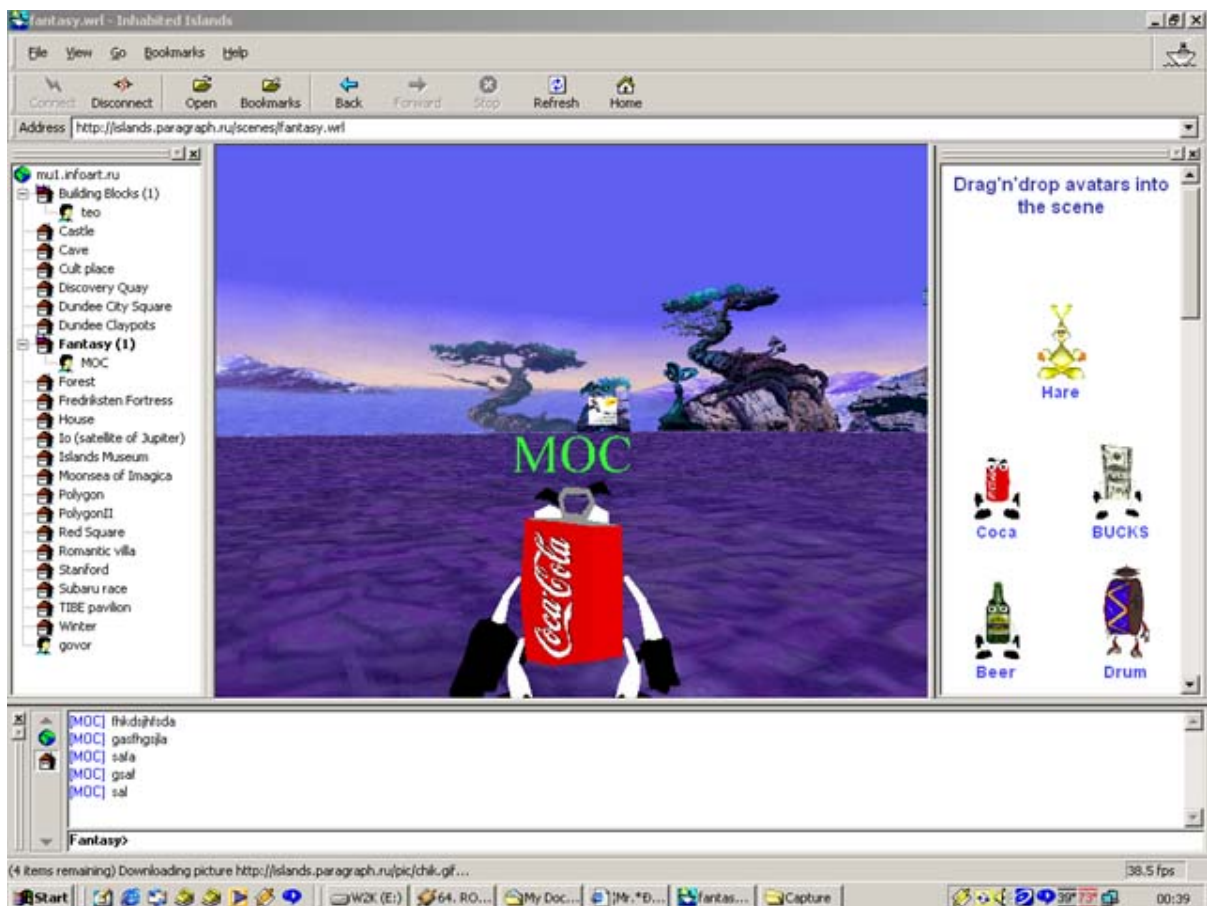


Figura 24 - Captura de tela do ParallelGraphics Island.

### 3.3.5 There<sup>45</sup>

O portal “*There*” (THERE, 2003) foi concebido com a intenção de oferecer ao público uma comunidade virtual onde seus usuários possam interagir entre si, participar de eventos ou brincadeiras e promover o comércio virtual.

O cenário em forma de *cartoon* (Figura 25) se parece muito com um desenho animado. O Sistema de navegação é de simples compreensão e exige apenas o uso do teclado. O uso do mouse se resume a escolha e clique de opções de interação com objetos do AVR e para uso da interface gráfica bidimensional de apoio. A Interface, um pouco carregada devido a grande variedade de opções de interações oferecida pelo sistema, possui muitas ferramentas e ações, consumindo tempo de estudo da interface.

O modelo de interação utilizado com os objetos virtuais do sistema consiste de um sinal em forma de seta que indica que o objeto possui uma interação. Ao clicar na seta aparece uma caixa de mensagem onde o usuário escolherá e clicará na opção que lhe convém. Estas interações podem ser a prova ou compra de uma roupa, a utilização de um aparelho, brincar com um cachorro, entre outras.

O diálogo funciona através de entradas diretas em balões situados acima do avatar e para iniciar uma conversa é só digitar e o se avatar soltar balões com o texto digitado em tempo real representando o que você disse ou quer dizer. Desta maneira qualquer avatar perto de você saberá o que você diz, a menos que você escolha uma conversa privada.

Os Avatares são criados pelo próprio usuário antes de entrar no sistema. O avatar vai possuir características particulares, pois são necessárias várias opções de configuração. As ações realizadas pelos avatares consistem de gestos que expressam sentimentos e estado de espírito. Enquanto o usuário utiliza outras funções fora do AVR seu avatar aparecerá usando um “óculos de RV” que significará que não está disponível, ou prestando atenção ao ambiente.

O usuário que entra pela primeira vez é contemplado com uma quantia em dinheiro que poder gastar como quiser nos pontos de venda encontrados no “*There*” (máquina de café, lojas de roupas, sapatos e perucas). A estratégia do “*There*” é

---

<sup>45</sup> O Autor não obteve acesso ao programa There e contou com a colaboração do usuário *beta tester*, Irla B. Rebelo, para a avaliação contida neste item.

oferecer ao público um espaço virtual onde possam ser encontradas opções de lazer e atividades de relação social, além da promoção de produtos que possam ser comercializados de maneira divertida e prática para consumidores e clientes.

Na versão beta hoje disponibilizada já podem ser encontradas algumas lojas, entre elas a *Nike*. Através de uma lista de email enviada com frequência pelo “*There*”, os usuários tomam conhecimento do que estará acontecendo (eventos) na próxima vez que entrar no sistema.

“*There*” é desenvolvido desde o final de 1999 com uma equipe de mais de 80 pessoas e com um investimento de 33 milhões de dólares. Até o momento se encontra em fase de testes.

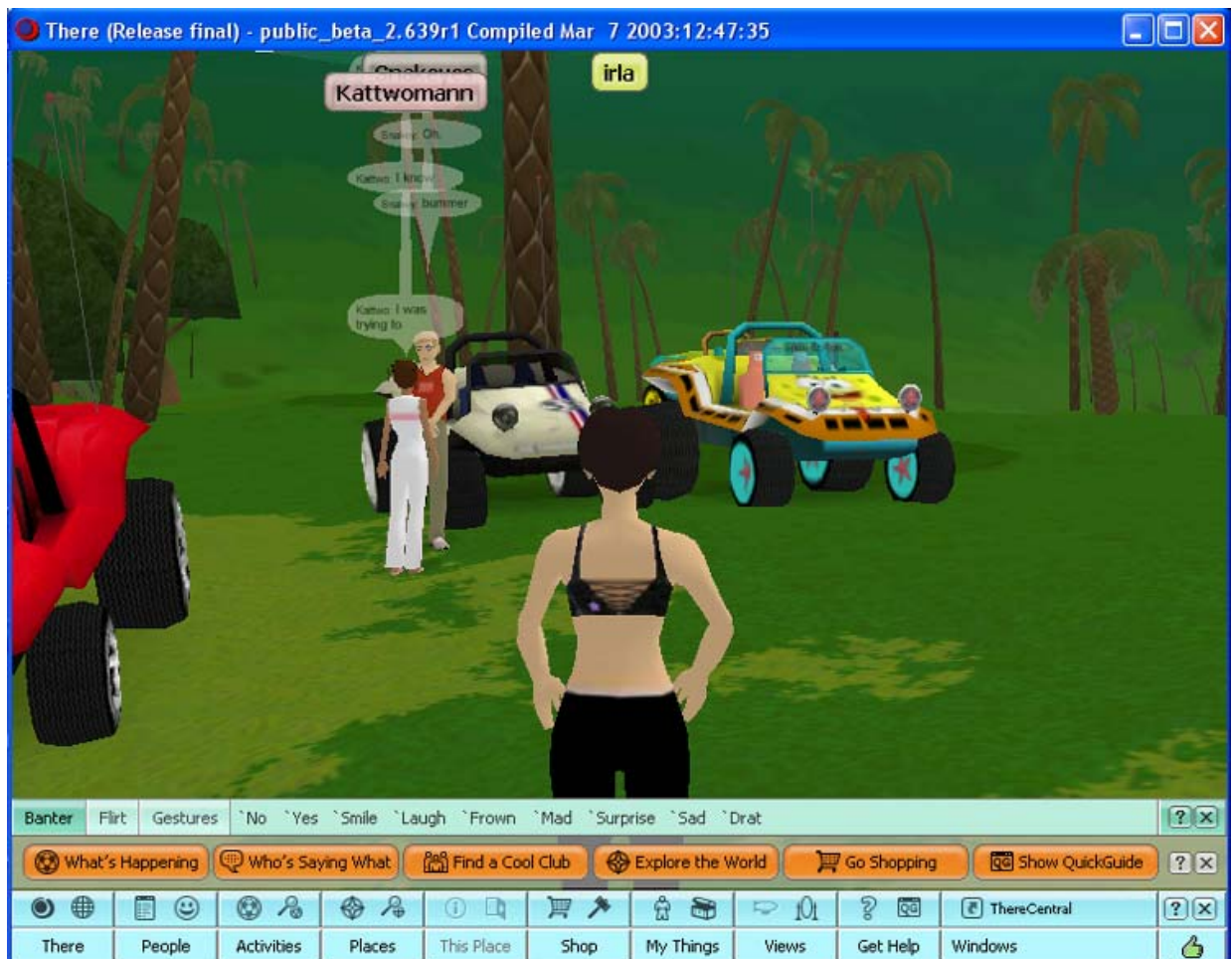


Figura 25 - Captura de tela do There.

Fonte: Irla B. Rebelo (Beta Tester).

### 3.4 Ambientes Virtuais Colaborativos

Ambientes Virtuais Colaborativos<sup>46</sup> (AVC) (BENFORD, 2001) envolvem o uso da tecnologia de realidade virtual distribuída para apoiar o trabalho de grupo. Uma condição necessária, mas não suficiente para um AVC, é a provisão de acesso multiusuário simultâneo à um sistema de realidade virtual. Porém, pode haver uma grande quantidade de diferenças entre um sistema multiusuário e um que de fato apóie o trabalho colaborativo. Assim uma segunda condição é que o sistema considere explicitamente o apoio às necessidades dos usuários que desejem realizar uma atividade em conjunto.

A essência de um AVC é que os usuários são representados explicitamente um ao outro dentro de um espaço compartilhado. Além disso, eles devem mover-se livremente dentro deste espaço encontrando outros usuários e, também, objetos e informações de interesse comum. A natureza interativa dos verdadeiros sistemas de realidade virtual permite aos usuários interagir entre si, com os objetos e informações ao seu redor. Os AVC estão intimamente ligados aos AVR, pois utilizam os mesmos princípios, apenas orientando o enfoque e criando características especiais para o trabalho colaborativo.

As habilidades sociais e noção espacial são naturais ao ser humano, já que todas as nossas atividades são realizadas e planejadas em nosso espaço-tempo real. Portanto nada mais natural do que colaborar em um ambiente virtual que possua características similares de espaço-tempo com nosso ambiente real.

#### 3.4.1 Projetos e aplicações

Vários projetos (BENFORD, 2001) têm abordado explicitamente o assunto de colaboração e realidade virtual. Estes incluem o *Comic Esprit III Basic Research Action* e o *UK's Virtuosi project* desenvolvido sob o *EPSRC/DTI CSCW Programme*. O primeiro apresenta técnicas gerais em desenvolvimento para AVC. O segundo tem focalizado na aplicação industrial, em projetos pilotos como a fábrica virtual para a indústria de manufatura e a passarela virtual para a indústria de moda. Esforços

---

<sup>46</sup> Do termo original *Collaborative Virtual Environments* (CVE).

---

japoneses incluem o *Collaborative Workspace* da NTT, que agora é open source<sup>47</sup>. O *Interspace project* da NTT apóia grupos de pessoas que navegam em uma cidade virtual, podendo fazer compras e se ocupar de atividades recreativas. O *Greenspace project* também é indicado como um exemplo de demonstração de um espaço virtual colaborativo entre o HITLab nos EUA e diversos locais no Japão.

Provavelmente o projeto mais influente nesta área foi o *DIVE*, desenvolvido pelo *Swedish Institute of Computer Science (SICS)* que resultou em um AVC disponível livremente, utilizado mundialmente como base para outros projetos. O *DIVE* provê um ambiente de desenvolvimento geral para AVC (FRECON, 1998).

Dentro do domínio especializado de simulação militar, a *US Navy* desenvolveu o *NPSNET* (MACEDONIA, 1994) (baseado em uma versão anterior denominada de *SIMNET*) que permite o envolvimento de grupos de várias centenas de pessoas participando de uma simulação de batalha em rede.

Vários outros projetos de pesquisa e sistemas agregam sistemas distribuídos e realidade virtual através de redes locais (embora com menos ênfase em colaboração). Estes projetos incluem o *MR Toolkit* e o *Aviary* da *Manchester University*.

A maioria dos sistemas de RV comercial também provê um apoio limitado por rede, como por exemplo, o *dVS* da *Division Ltd.* (Reino Unido), o *Elysium* da *Virtuality* (Reino Unido), o *Superscape* da *Dimension International* (Reino Unido), o *World Toolkit*, *WorldUp* e *World2World* da *Sense 8* (EUA) e o *Virtual Worlds Platform* da *Blaxxun Interactive* (Alemanha).

Outros projetos consideram o uso de metáforas tridimensionais do quadro de reunião para sistemas multimídia de conferência, incluindo ai o *Milan* e o *Electronic Office of Cook*.

---

<sup>47</sup> Open Source, de forma simplificada, quer dizer código aberto à comunidade, onde as pessoas podem fazer melhorias no programa e disponibilizar para o resto da comunidade.

### **3.5 Considerações finais**

Neste Capítulo foram apresentados os conceitos básicos necessários para o desenvolvimento de um ambiente virtual tridimensional multiusuário para Internet, denominado LRVCHAT3D, que é o assunto do capítulo 5.

Os conceitos de Ambiente Virtual em Rede (AVR) e Ambiente Virtual Colaborativo (AVC) são essenciais na medida que o sistema proposto neste trabalho é um AVR e se propõe a possuir características que permitam sua utilização como um AVC. Foi visto também neste capítulo os elementos e fatores relacionados com o desenvolvimento de um AVR, comunicação e distribuição de dados em AVR, exemplos de sistemas similares e conceitos auxiliares, como o conceito de avatar que é a forma de representação de usuários adotada no sistema proposto pelo autor.



## 4. VRML – Linguagem para Modelagem em Realidade Virtual

Neste capítulo é apresentada conceitualmente a linguagem VRML (*Virtual Reality Modeling Language*, ou Linguagem para Modelagem em Realidade Virtual) (WEB3D, 2003), seu histórico e desenvolvimento atual. Esta linguagem é utilizada como padrão de arquivos no sistema desenvolvido pelo autor como será detalhado no capítulo 5.

VRML é um formato de arquivo para descrever objetos e mundos interativos tridimensionais. Foi projetada para ser usada na Internet, Intranet e sistemas locais (CAREY, 1997). Também é esperado que a VRML seja adotada como um formato de intercâmbio universal para gráficos tridimensionais integrados e multimídia.

VRML pode ser usada em uma variedade de áreas de aplicação como engenharia, visualização científica, apresentações multimídia, entretenimento e títulos educacionais, páginas *web* e mundos virtuais compartilhados.

A VRML foi projetada para cumprir as seguintes exigências:

- **Autoria:** Habilidade no desenvolvimento de programas de computação capazes de criar, editar e manter arquivos VRML, assim como portabilidade e conversão entre outros formatos gráficos 3D e a VRML.
- **Composição:** Habilidade de combinação de objetos tridimensionais dinâmicos dentro de um mundo VRML, permitindo assim a reutilização.
- **Expansibilidade:** Habilidade para adicionar elementos novos não definidos explicitamente pelo padrão VRML.
- **Implementação:** Capacidade de implementação em uma gama extensiva de sistemas.
- **Desempenho:** Enfatiza o desempenho interativo em uma variedade larga de plataformas computacionais.
- **Escalabilidade:** Habilidade de manuseio de mundos tridimensionais dinâmicos arbitrariamente grandes.

A VRML é capaz de representar objetos tridimensionais estáticos e dinâmicos com *hyperlinks* para outras mídias como texto, sons, filmes, e imagens.

As ferramentas de autoria para a criação de arquivos em VRML são amplamente difundidas e extensamente disponíveis para diferentes plataformas.

Ferramentas tais como:

- **3DA IT ( Micronet Co., Ltd. )**
- **3DAnywhere ( Monfort Software Engineering Ltd. )**
- **3D Virtual Characters ( W Interactive SARL )**
- **VrmlIPad ( ParallelGraphics )**

- **Spazz3d ( Virtock Technologies )**
- **Internet Space Builder ( Parallel Graphics)**
- **Internet Scene Assembler ( Parallel Graphics )**

A VRML apóia um modelo expansível permitindo o desenvolvimento de extensões baseadas no padrão básico. Existem também formas de interação de aplicações com objetos VRML através do uso de interfaces de programação (API<sup>48</sup>), Este recurso é utilizado no desenvolvimento do protótipo de AVC apresentado neste trabalho, e serve como ligação entre a parte tridimensional do sistema e a parte de gerenciamento, rede e interface bidimensional.

## 4.1 Origens e Desenvolvimento

### a) Surge a Internet e a WWW (DAUB, 2001):

Em Outubro de 1969, a equipe de Leonard Kleinrock, professor de ciências computacionais na Universidade da Califórnia, em Los Angeles, conseguiu fazer a comunicação entre dois computadores, sendo um marco na história das redes de computadores.

Em 1970, a primeira rede de transmissão de informação por pacotes<sup>49</sup> nos Estados Unidos assentava em quatro nós: a Universidade da Califórnia em Los Angeles e em Santa Cruz, a Universidade de Stanford e a Universidade de Utah, em Salt Lake City. Surgia assim a ARPANet.

Em 1989 um grupo de cientistas do CERN (European Organization for Nuclear Research) decidiu trocar informações pela Internet, mas queriam algo com uma interface focada nos usuários e não no sistema de computadores. Então Tim Berners Lee, pesquisador do CERN, criou o primeiro Servidor para WWW (*world wide web* ou grande rede mundial) e *Browser* (Visualizador para a WWW).

Inicialmente a WWW foi baseada em uma interface puramente textual. Posteriormente a HTML (*Hyper Text Markup Language* ou Linguagem de marcação de *hypertexto*) fornecia um conjunto de melhorias na interface com o usuário, principalmente em seu aspecto gráfico.

Em 1993 o visualizador (*browser*) MOSAIC foi lançado nos EUA pelo NCSA

---

<sup>48</sup> API - Application Programming Interface.

<sup>49</sup> Blocos de dados.

(National Center for Supercomputer Applications), agora além do já tradicional texto, seria possível colocar quadros (imagens) interativos no corpo do documento, servindo inclusive como mapa. Agora a WWW era utilizada por um número maior de usuários não especialistas. A questão que surgiu era até onde esta interface bidimensional poderia resistir aos avanços tecnológicos e computacionais.

**b) A linguagem VRML (DAUB, 2001; CAREY, 1997; RECHTSTEINER, 2001):**

Em janeiro de 1994 Mark Pesce e Tony Parisi lançaram um equivalente tridimensional do HTML que funcionava em parceria com um visualizador WWW previamente existente. Eles denominaram esta aplicação de **Labirinto**. Este lançamento desencadeou grande interesse, pois possibilitaria o uso de uma interface em realidade virtual para a WWW. O termo VRML surgiu em abril de 1994 durante a primeira conferência internacional de WWW que ocorreu em Genebra, Suíça.

Tim Berners-Lee (CERN) e Dave Raggett (HP Labs) organizaram uma sessão para discutir Realidade Virtual associada a WWW. Os participantes apresentaram alguns projetos e concluíram que as ferramentas a serem desenvolvidas necessitavam ter um idioma comum para descrever o ambiente tridimensional.

A designação VRML (Virtual Reality Markup Language) foi estabelecida, e o grupo decidiu começar os trabalhos de especificação logo após a conferência. A palavra “Markup” foi alterada posteriormente para “Modeling” refletindo a natureza gráfica do VRML.

Em junho de 1994, a revista WIRED criou uma lista discussão via correio eletrônico para pessoas interessadas em VRML. As discussões públicas desencadeadas nesta lista determinaram o caminho inicial do que seria o princípio da VRML. Era ponto pacífico dos integrantes desta lista que não seria necessário e nem produtivo “reinventar a roda”, de modo que o trabalho deveria prever a adaptação do que já existia. Então o início do trabalho foi baseado no formato de arquivo 3D chamado de Open Inventor, criado pela Silicon Graphics (SGI). Gavin Bell, engenheiro da SGI, adaptou o Open Inventor criando o primeiro esboço do VRML 1.0 que foi anunciado em outubro de 1994.

Paul Strauss, também da SGI, escreveu o primeiro visualizador de VRML: o

"QvLib". Inicialmente o QvLib só rodava em computadores da SGI. Mas logo a empresa liberou o código fonte e *browsers* VRML foram então escritos para quase todos os sistemas operacionais conhecidos. Falhas secundárias foram consertadas e a versão final do VRML 1.0 foi liberada em maio de 1995.

Uma organização livre conhecida como VRML Advisory Group (VAG), liderada por Pesce, foi responsável pela direção geral no desenvolvimento da VRML. Mas todos podiam e participavam neste desenvolvimento, e a lista de discussão citada anteriormente foi decisiva neste aspecto, criando um consenso em torno das decisões.

O consenso e participação interativa em torno da VRML 2.0 foram ainda mais pronunciados. Em dezembro de 1995 a gigante Microsoft tentou impor seu próprio padrão para a VRML 2.0, mas foi combatida ferozmente pelos participantes da lista de discussão.

A segunda versão de VRML adicionou mais interatividade. Foi desenvolvida principalmente pela equipe VRML da SGI com contribuições da Sony, Mitra, e muitos outros. A VRML 2.0 foi revisada pelo grupo moderador da lista de discussão da VRML ([www-vrml@vrm1.org](mailto:www-vrml@vrm1.org)) e depois foi adotado e endossado por muitas companhias e indivíduos. Em dezembro de 1997 a VRML97 substituiu a VRML 2.0 e foi liberada formalmente como um padrão internacional (ISO/IEC 14772).

### **c) X3D, o Futuro? (WEB3D, 2003)**

Atualmente o grupo chamado Web3D Consortium, formado por diversas pessoas e instituições coordena os esforços no desenvolvimento de padrões e o foco atual é a especificação da nova versão da VRML, que agora se chama X3D (eXtensible 3D, ou 3D extensível), modernizando e aglutinando as melhores tecnologias na área. Seu padrão de formato de arquivo é o XML (Extensible Markup Language). A especificação X3D está em fase final de padronização ISO.

No apêndice B é apresentado a codificação de um exemplo simples em VRML97 e em X3D, para efeito de comparação da codificação. Este exemplo consiste simplesmente em uma esfera vermelha e um cubo azul, e tem como única finalidade a comparação entre os formatos de arquivo VRML97 e X3D.

## 5. LRVCHAT3D

Neste capítulo é apresentado o desenvolvimento do protótipo do Ambiente Virtual Tridimensional Multiusuário para Internet proposto pelo autor, denominado de LRVCHAT3D (ROSA Jr., 2001). Por definição este sistema é um Ambiente Virtual em Rede (AVR) e possui características como compartilhamento de dados e suporte a objetos interativos (que podem ser controlados de forma colaborativa pelos usuários conectados ao sistema), que o qualificam também como um Ambiente Virtual Colaborativo (AVC).

O objetivo deste projeto é ser uma ferramenta flexível e adaptativa que permita a interação de usuários, a troca efetiva de informação, o acesso à diversas mídias e que forneça um ambiente diferenciado para facilitar a integração entre usuários, podendo então ser utilizado em diversos projetos que venham a utilizar os conceitos de AVR e AVC, servindo de sistema hospedeiro<sup>50</sup> para diversos ambientes virtuais.

Seu público-alvo primário é o projetista de sistemas de informação que podem utilizar o sistema proposto como hospedeiro de seus ambientes virtuais, e assim poder concentrar seus esforços apenas nas metodologias (ensino, treinamento, mercadológica, etc.) e recursos (avatars, objetos e ambiente de fundo) que sejam necessários para que seus ambientes. Indiretamente o LRVCHAT3D atenderá o usuário final dos sistemas que o utilizarem como base, suprindo-os com toda a parte gerencial do sistema multiusuário, com o subsistema gráfico e com a manutenção da persistência<sup>51</sup> destes ambientes.

O LRVCHAT3D não é diretamente responsável pelo conteúdo apresentado ao usuário final do sistema, ao contrário, seu objetivo é gerenciar o ambiente virtual como um todo, ao mesmo tempo em que provê recursos para a construção de literalmente qualquer ambiente virtual que se baseie no conceito de objetos e eventos.

Neste sistema, objeto é qualquer elemento de interação que possua propriedades bem definidas e que possa ser tratado individualmente, em outras

---

<sup>50</sup> Sistema hospedeiro, isto é, o LRVCHAT3D fornece o suporte e a interface para que os ambientes virtuais criados pelos projetistas funcionem.

<sup>51</sup> Capacidade de manter as informações de um sistema, mesmo que seus utilizadores não estejam mais conectados ao sistema, por exemplo, um usuário criou um objeto no ambiente e depois deixou o sistema, quando este retornar ao sistema, o objeto ainda estará lá.

palavras, que seja auto contido. Evento é o elemento de interação que interliga os objetos entre si e com o sistema gerenciador de ambientes do LRVCHAT3D.

Exemplificando, temos um objeto que representa uma lâmpada, e tem como propriedades a cor de sua emissão luminosa e um indicador de ativação (ligada / desligada). Este objeto possui um evento de entrada (evento que chega ao objeto) que quando recebido alterna seu estado de ativação, ou seja, liga e desliga a lâmpada. Este objeto possui ainda um evento de saída (evento que sai do objeto) que indica a outros objetos ou ao gerenciador do sistema que uma de suas propriedades foi alterada. Desta forma temos um objeto auto contido, ou seja, que possui propriedades devidamente estabelecidas e um conjunto de eventos que permite sua interação com outros objetos e com o sistema gerenciador de ambientes e de forma similar o LRVCHAT3D pode gerenciar uma grande variedade de objetos e suas interligações. Veremos mais detalhes de como objetos e eventos são tratados no sistema LRVCHAT3D no item 5.2 deste capítulo.

A representação do usuário no sistema é feita utilizando-se avatares, que possuem características similares aos objetos, isto é, possuem certas propriedades e eventos, mas seguem um modelo mais rígido e são controlados diretamente pelo sistema gerenciador de ambientes do LRVCHAT3D.

O Sistema permite a inserção de agentes inteligentes sintéticos e outros sistemas de controle como se fossem usuários normais, desta forma a integração de agentes inteligentes pode ser desenvolvida em separado, utilizando os mesmos recursos que um usuário humano tem ao entrar no sistema.

O LRVCHAT3D utiliza VRML como base de desenvolvimento de um AVC (BROLL, 1996) e os avatares e objetos utilizados no sistema são construídos neste formato.

Desde o início do projeto procurou-se uma metodologia de projeto que permitisse a modularidade do programa de maneira a permitir a troca de uma solução tecnológica por outra, sem perda das características básicas. Assim, pode-se a qualquer momento utilizar os mesmos modelos e soluções estruturais, e modificar, por exemplo, a saída gráfica tridimensional de VRML para outra biblioteca gráfica baseada em OpenGL ou DirectX, ou ainda trocar a topologia de rede (vide item 3.2.1) de cliente/servidor para ponto a ponto sem grandes alterações na base

do programa. O desenvolvimento do sistema não foi estático e suas características finais foram moldadas no decorrer de seu desenvolvimento.

No item 3.3 deste trabalho apresenta-se diversos sistemas similares ao proposto, sendo que a análise destes sistemas foi determinante na opção de desenvolver um sistema próprio. A maioria dos sistemas similares é de arquitetura fechada, ou seja, não permitem uma margem de desenvolvimento de características novas, nem sua adaptação para uma realidade regional. Além deste, outros fatores determinaram o desenvolvimento de um sistema próprio:

- Possibilidade de estudo mais aprofundado do tema ambientes virtuais;
- Domínio de tecnologias associadas a ambientes virtuais em rede;
- Liberdade de desenvolvimento de sistemas AVR e similares;
- Possibilidade de criar sistemas com características especiais para cada tipo de aplicação;
- Necessidade de se ter uma plataforma que seja o mínimo possível dependente de sistemas proprietários, principalmente estrangeiros.

A única dependência tecnológica relevante do sistema se encontra no uso de um componente VRML para visualização, produzido por terceiros, mas como citado anteriormente, o sistema é modular e no futuro esta dependência pode ser substituída por uma solução própria. Mesmo assim o componente é utilizado para visualização de VRML que por sua vez é um padrão aberto.

A escolha de usar VRML no projeto foi baseada em dois fatores: 1) VRML ser um padrão aberto (sua especificação é pública), aceito mundialmente. 2) VRML ser de domínio do autor.

A escolha de usar um componente VRML de terceiros (*ParallelGraphics Cortona ActiveX*)(PARALLELGRAPHICS, 2001) foi baseada também na experiência prévia, na disponibilidade de uso, documentação e na capacidade do mesmo de ser utilizado com o pacote de desenvolvimento de software (Borland Delphi e Borland C++Builder) (BORLAND, 2001), também de domínio do autor.

O desenvolvimento do LRVCHAT3D foi dividido em três fases: A Fase I serviu basicamente para testar a integração entre o componente VRML e o pacote de desenvolvimento de software e desenvolvimento de características primárias do protótipo, a fase II é o desenvolvimento do protótipo propriamente dito e a fase III consistiu em aplicações do protótipo. Estas fases serão detalhadas no restante deste capítulo, com destaque para a fase II.

## 5.1 Fase I

O objetivo principal desta fase foi o aprendizado de aspectos de interface entre o *Parallel Graphics Cortona ActiveX* e o *Borland C++Builder*, além do levantamento de questões referentes a dependência temporal e sincronismo entre o módulo de rede e o módulo gráfico.

Esta fase consistiu na implementação de uma ferramenta de comunicação em tempo real (Chat), utilizando um ambiente virtual simplificado modelado em VRML.

Nesta primeira ferramenta o usuário pode circular livremente pelo ambiente tridimensional modelado em VRML, interagir com outros usuários através de mensagens texto e notar a presença e a movimentação espacial dos outros usuários através da visualização dos avatares destes usuários. A visão do usuário é em primeira pessoa, ou seja, ele não visualiza sua própria representação (avatar).

Os avatares utilizados nesta fase são modelados em VRML e não possuem nenhuma característica interativa, servindo apenas para localização (posição e direção/sentido de movimento) dos usuários no espaço tridimensional.

Nesta fase também foram realizados diversos testes com modelos de topologia de rede (vide item 3.2.1). Tanto o modelo cliente-servidor, utilizado como padrão para o desenvolvimento deste projeto, como o modelo ponto a ponto apresentaram resultados satisfatórios, sendo intercambiáveis conforme as necessidades específicas de cada aplicação e de seu público-alvo.

Um dos resultados mais importantes desta fase foi a criação de classes de programações que servem de base para a fase II, envolvendo principalmente a parte de rede e integração do VRML.

Nesta fase foram desenvolvidos um programa cliente e um programa servidor escrito em JAVA (para rodar em servidores UNIX) e também em C++Builder,

A Figura 26 mostra uma das versões do programa cliente desenvolvidas nesta fase.



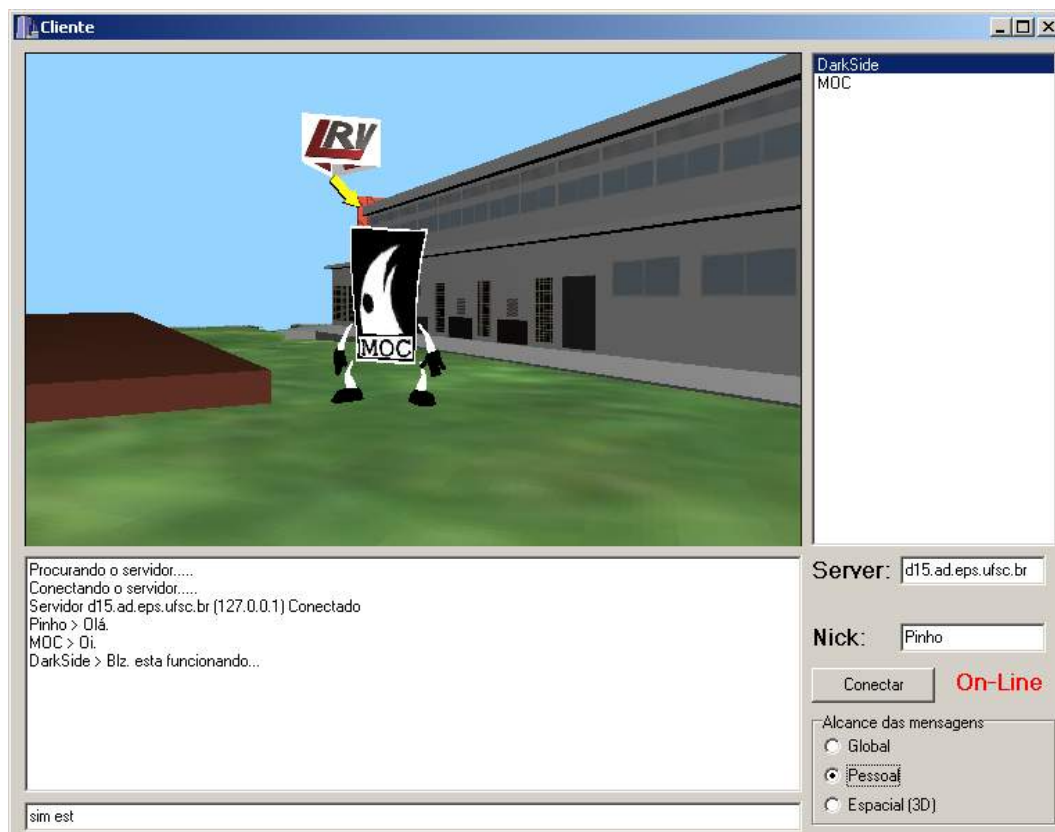


Figura 26 - Protótipo do programa cliente da Fase I.

O quadro 2 resume as características desta fase do desenvolvimento.

Quadro 2 - Resumo da Fase I.

<b>Características observadas</b>	- Ferramenta de comunicação básica (Chat); - Navegação espacial em um único ambiente tridimensional modelado em VRML; - Utilização de avatares (sem opção de escolha por parte do usuário) para a representação dos usuários conectados.
<b>Ferramentas utilizadas</b>	- Borland C++ Builder (Programação de rede, interface 2D); - ParallelGraphics Cortona SDK (VRML ActiveX).
<b>Investigações realizadas</b>	- Interação de mídias e interface básica; - Topologias de rede; - Exploração primária e estudos de agregação de funções.
<b>Resultados</b>	- Conhecimento sobre integração entre Cortona e C++Builder; - Definição do modelo cliente / servidor como topologia de rede; - Desenvolvimento de classes primitivas de programação; - Protótipos cliente e servidor de um Chat.

## 5.2 Fase II

O objetivo desta fase foi o desenvolvimento e implementação de um protótipo totalmente funcional de um Ambiente Virtual Tridimensional Multiusuário para Internet. A fase II consistiu na agregação de funções (compartilhamento de objetos dinâmicos, animação de avatares e portais entre servidores) ao protótipo obtido na fase I descrito no item anterior.

O modelo de rede adotado é o modelo cliente-servidor com um servidor central que é responsável pelo gerenciamento de todas as informações que trafegam no sistema. Embora o modelo cliente-servidor não seja o melhor modelo para ambientes virtuais de larga escala, este modelo foi adotado para o desenvolvimento deste sistema, em trabalhos futuros este servidor poderá compor uma rede distribuída com outros tipos de servidores de controle, estabelecendo o particionamento do ambiente, filtragem do fluxo de dados e predição de movimento.

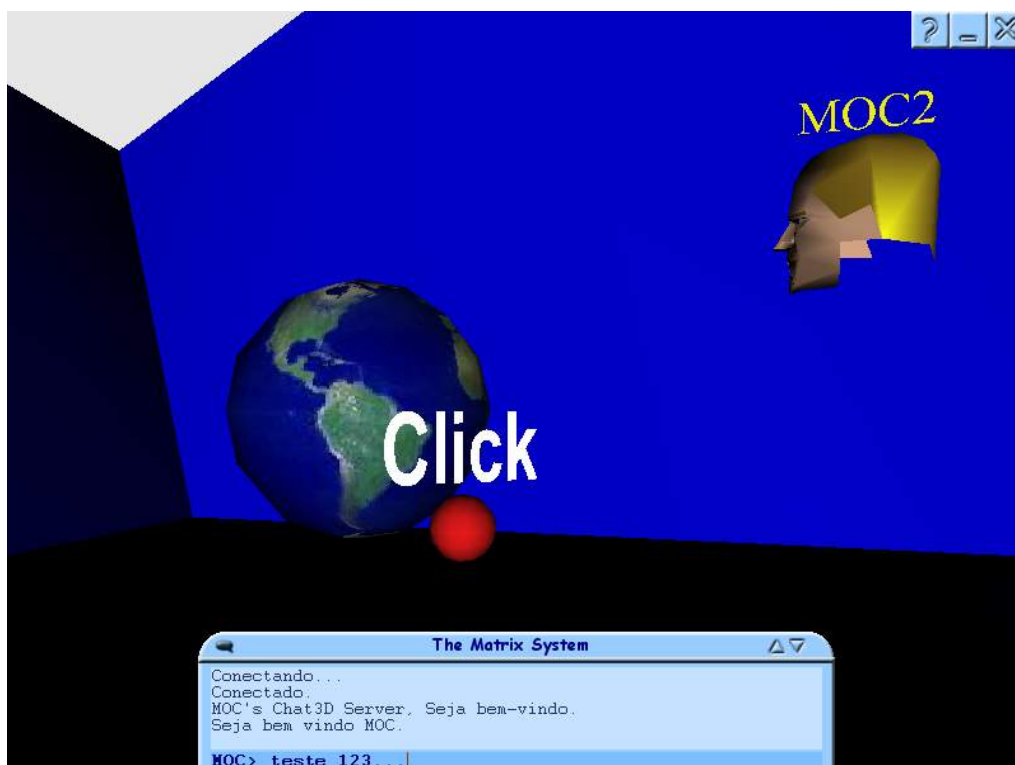
Além de mudanças de consolidação e consistência no protótipo da fase I, houve a migração do Borland C++Builder para Borland Delphi devido a questões de compatibilidade e suporte com a tecnologia VRML adotada (ParallelGraphics).

Os arquivos de recursos em VRML (Ambientes de fundo, Avatares e Objetos) são referenciados por endereços *web* (tipo <http://aaa.com>) o que aumenta ainda mais a flexibilidade do sistema. Os recursos utilizados podem estar em repositórios espalhados pela rede.

Nesta fase foram incluídas as funções de animação de avatares e implementados os objetos compartilhados, que representam a funcionalidade que permite aos projetistas de ambientes para o LRVCHAT3D explorarem a capacidade interativa e colaborativa do sistema.

A figura 27 apresenta um primeiro ensaio de objeto compartilhado. O Globo terrestre é um objeto do ambiente que pode ser manipulado (pode ser girado) através da ativação da esfera vermelha por qualquer usuário que esteja conectado ao sistema. Todos os outros usuários que estiverem visualizando o globo perceberão o movimento do mesmo.

A figura 27 também ilustra um dos testes de design de interface para o sistema, usando uma metáfora onde os elementos bidimensionais da interface são absorvidos pela interface tridimensional do ambiente, buscando uma melhor integração entre os elementos 2D e 3D, e desta forma obter uma melhora na sensação de imersão do usuário com o ambiente virtual.



**Figura 27 - Protótipo teste design da Fase II.**

A figura 28 apresenta um usuário acionando uma animação predefinida, sendo visualizada por outro usuário através da representação (avatar) utilizada pelo usuário que disparou a animação. Nesta figura é apresentada também a interface clássica (interface Windows padrão) adotada para a simplificação no desenvolvimento do protótipo. O ambiente utilizado nesta etapa já é mais elaborado e avatares humanóides com animação são utilizados na representação do usuário.



Figura 28 - Protótipo da Fase II, mostrando um avatar animado.

### Quadro 3 - Resumo da Fase II.

<b>Características observadas</b>	<ul style="list-style-type: none"> <li>- Ambientes virtuais múltiplos, selecionáveis e mais elaborados;</li> <li>- Inclusão de Objeto de Ação Compartilhada (ativação e visualização disponibilizadas para todos os usuários do sistema de forma compartilhada);</li> <li>- Múltiplos avatares com animação (gestos) disparada pelo usuário;</li> <li>- Portais entre Ambientes;</li> <li>- Elementos referenciados por endereços http.</li> </ul>
<b>Ferramentas utilizadas</b>	<ul style="list-style-type: none"> <li>- Borland Delphi (Programação de rede, interface 2D);</li> <li>- ParallelGraphics Cortona SDK (VRML ActiveX).</li> </ul>
<b>Investigações realizadas</b>	<ul style="list-style-type: none"> <li>- Animação de avatares;</li> <li>- Formatos de objetos;</li> <li>- Objetos dinâmicos;</li> <li>- Portais.</li> </ul>
<b>Resultados</b>	<ul style="list-style-type: none"> <li>- Comunicação via texto;</li> <li>- Navegação espacial em um ambiente tridimensional modelado;</li> <li>- Avatares animados, escolhidos pelo usuário, a partir de um conjunto disponibilizado de acordo com o ambiente;</li> <li>- Portais funcionais e ativação de objetos compartilhados.</li> </ul>

Os itens a seguir, entre 5.2.1 e 5.2.5, descrevem os diversos tópicos do desenvolvimento do sistema protótipo, desde estrutura de rede, formatos de arquivos, até uma descrição do protocolo de comunicação entre o programa servidor e os programas cliente. No item 5.2.6 será descrito o programa servidor e no item 5.2.7 uma descrição ilustrada e completa da funcionalidade do programa cliente.

### 5.2.1 Estrutura básica de rede

O sistema protótipo possui topologia de rede cliente-servidor, sendo o banco de dados replicado para cada cliente. A manutenção de sua integridade é efetivada através de trocas de mensagens entre os clientes e o servidor central por meio de um protocolo de comunicação simplificado que roda sobre o protocolo base TCP/IP<sup>52</sup>. Os arquivos de ambientes, avatares e objetos são acessados pelos clientes diretamente de repositórios em servidores convencionais via protocolo http. Existe ainda a possibilidade de navegação entre ambientes através do uso de portais entre servidores. A estrutura simplificada de acesso é apresentada na figura 29.

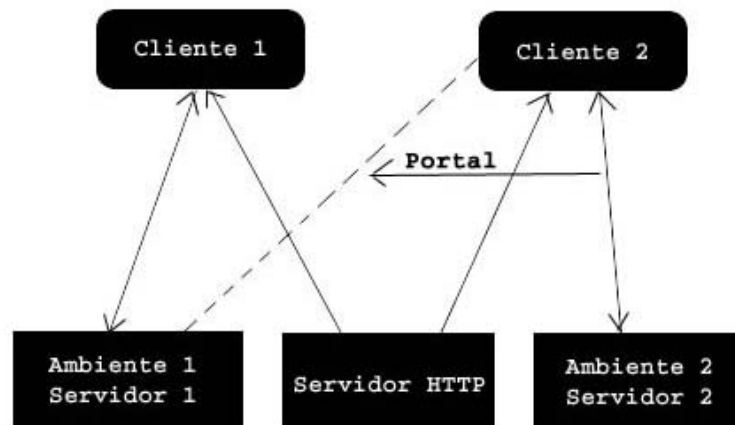


Figura 29 - Estrutura básica de rede no sistema.

### 5.2.2 Formatos de Arquivos de Ambiente

A estrutura dos arquivos utilizados, para descrição de ambientes no sistema, respeita o formato VRML97. Como base de referência de dimensões utiliza-se a escala base 1:1 para o dimensionamento dos ambientes, avatares e objetos, de

<sup>52</sup> Para saber mais sobre TCP/IP consulte HEDRICK (2003).

maneira a manter uma coerência entre arquivos de diferentes fontes.

Internamente o sistema utiliza prototipagem VRML (PROTO's) específica para facilitar a integração em sistemas multiusuários, utilizando alguns conceitos adaptados das especificações *Living Worlds* (HONDA, 2001) e *VNET* (WHITE, 2001). Estes *protos* são implementados diretamente no código da aplicação o que permite que o arquivo de ambiente seja o mais simples possível (VRML nativo).

Toda as interações multiusuário são realizadas através dos avatares e objetos do sistema. Cada arquivo de ambiente deve conter somente objetos interativos locais, sem que haja nenhuma distribuição de efeitos aos outros usuários conectados ao sistema. Exemplificando, um objeto estático no ambiente, quando acionado, informa a hora local. Sendo este objeto de ação local deve ser incluído diretamente no arquivo de ambiente, pois o resultado da ação não necessita uma sincronização do ambiente com outros clientes.

### 5.2.3 Formatos de Arquivos de Avatares

A estrutura dos arquivos de Avatares, assim como no caso dos ambientes citados anteriormente, faz uso do formato VRML97 utilizando conceitos de *protos*.

O sistema suporta, com poucas modificação, os formatos de avatares da *ParallelGraphics*, o formato padronizado H-Anim (H-ANIM, 2001), ou outro formato qualquer que suporte acionamento de gestos baseado em disparo de tempo.

Para que um avatar seja válido para o sistema basta que a definição do mesmo respeite o padrão de prototipagem VRML adotado. Este padrão consiste de um identificador (*\_type\_name\_*), de um campo indicando a altura do ponto de vista do usuário (*\_eye\_position\_*), de uma lista com o nome das animações suportadas pelo avatar (*\_animations\_names\_*) seguidos por uma seqüência de eventos acionadores (disparadores baseados em tempo) com nomes coerentes com a lista de animações (*\_animations\_names\_*).

A implementação das animações é transparente para o sistema, portanto qualquer implementação, seja ela baseada em Script ou classes JAVA, é suportada desde que esteja dentro das especificações de suporte do Cortona VRML ActiveX.

No quadro 4 é apresentado um código base de avatar do sistema.

#### Quadro 4 - Código base de avatar do sistema.

```

#VRML V2.0 utf8

PROTO ALEX [
  exposedField SFVec3f   _eye_position_   0 1.510 0
  exposedField SFString  _type_name_     "ALEX"
  exposedField MFString  _animations_names_ [
    "ani_stopAll",
    "ani_walkforward_do",
    "ani_wait1_do",
    "ani_wait2_do",
    "ani_wait3_do",
    "ani_wait4_do",
    "ani_speaking1_do",
    "ani_speaking2_do",
  ]
  eventIn      SFTIME   ani_stopAll
  eventIn      SFTIME   ani_walkforward_do
  eventIn      SFTIME   ani_wait1_do
  eventIn      SFTIME   ani_wait2_do
  eventIn      SFTIME   ani_wait3_do
  eventIn      SFTIME   ani_wait4_do
  eventIn      SFTIME   ani_speaking1_do
  eventIn      SFTIME   ani_speaking2_do
]
{
  ...Código da implementação do avatar inclui a forma geométrica e as animações
}

DEF AVATAR ALEX {}

```

#### 5.2.4 Formatos de Arquivos de Objetos

A estrutura dos arquivos de objetos utiliza o formato VRML97, também é baseada em conceitos de *protos* como nos casos anteriores. Agora, a criação de objetos é mais complexa, pois um novo elemento é adicionado: o suporte à programação interna de objetos utilizando scripts (VRMLScript, JavaScript) ou mesmo classes em JAVA. No quadro 5 é apresentado o código exemplo com a estrutura do *proto* de objeto e descrição de suas propriedades e funções. Exemplos completos de códigos de objetos utilizando diversas possibilidades de implementação encontram-se no apêndice A.

### Quadro 5 - Código estrutural de objetos do sistema.

```
#VRML V2.0 utf8

PROTO MOC_OBJECT [
  exposedField SFFloat   Version 1.0
  exposedField SFString  Type ""
  exposedField SFString  Desc ""
  exposedField SFBool    Private FALSE
  eventIn       SFBool   Properties
  eventIn       SFString  Command
  eventIn       SFString  Command_From_Net
  eventOut      SFString  Response
  eventOut      SFString  Command_To_Net
]
{... Implementação ...}

DEF OBJECT MOC_OBJECT {}
```

O *proto* é composto dos seguintes campos:

- **Version** que indica a versão da especificação de objetos do sistema.
- **Type** que indica o tipo de objeto (reservado para uso futuro)
- **Desc** que apresenta uma descrição do objeto.
- **Private** que indica se o objeto pode ser retirado de um ambiente ou ter suas propriedades alteradas por algum outro usuário que não seja o que o adicionou ao ambiente.

Os eventos que devem ser implementados no objeto são responsáveis pela comunicação entre o programa cliente e VRML. Com exceção do evento *Properties*, todos os outros eventos são do tipo *String*, ou seja, são mensagens texto com o caractere “|” como separador. O processamento interno de um evento de entrada pode gerar um evento de saída.

Os eventos de entrada são os seguintes:

- **Properties** é um evento gerado pelo programa cliente que envia um valor booleano “TRUE” para que o objeto responda com informações sobre as propriedades do objeto através do evento de resposta *Response*.

A resposta padronizada deve seguir o modelo:

```
%PROP%|Prop_Name!Prop_Value|...|Prop_Name!Prop_Value
```

Onde %PROP% é uma constante identificadora de mensagem seguida por todas as propriedades do objeto, sendo Prop\_Name o nome da propriedade e Prop\_Value o valor atual codificado em formato texto.

- **Command** é um evento utilizado internamente pelo arquivo de objetos para



tratar eventos, como um *click* no objeto, e processar este comando de forma apropriada. Recebe uma mensagem texto indicando o comando e gera um evento de saída do tipo *Response* ou *Comand\_To\_Net*, dependendo da programação do objeto.

- **Command\_From\_Net** é um evento gerado pelo programa cliente que envia uma mensagem texto para o Objeto. Esta mensagem é originária do programa servidor e é responsável pela consistência do ambiente virtual multiusuário gerado pelo sistema. Assim, quando algum usuário efetua alguma ação ou altera alguma propriedade de um objeto, todas as cópias replicadas deste objeto nas máquinas clientes devem ser atualizadas, e este evento indica então a chegada de uma mensagem de sincronismo.

A mensagem recebida está no formato:

*UserName|Name!Value*

Onde *UserName* identifica o usuário que efetuou a alteração seguido pelo nome da propriedade a ser alterada, ou do nome indicativo de ação, seguido do novo valor que a propriedade deve assumir, ou no caso de uma ação indica algum parâmetro adicional. Todos os valores estão codificados em formato texto.

Os eventos de saída são os seguintes:

- **Response** é um evento gerado pelo VRML para indicar ao programa cliente uma resposta à algum evento de entrada ou ação ocorrida no VRML com o objeto. Atualmente o programa cliente reconhece quatro tipos de resposta, mas outros tipos podem ser adicionados com facilidade:

**1) %PROP%|Prop\_Nam|Prop\_Value|...|Prop\_Name!Prop\_Value**

Indica as propriedades do objeto, como resposta a um evento *Properties*.

**2) %EXIT%**

Indica ao programa cliente que este deve efetuar uma desconexão do servidor atual.

**3) %PORTAL%|Servidor|Port**

Indica ao programa cliente que um portal foi ativado dentro do

ambiente VRML e que deve efetuar uma desconexão do servidor atual e efetuar uma nova conexão usando o IP dado pelo valor *Servidor* e porta indicada pelo valor *Port*.

#### 4) %URL%|Site

Que indica ao programa cliente que este deve abrir o *web browser* que estiver instalado na máquina cliente e carregar o endereço indicado pelo valor *Site*.

- **Command\_To\_Net** é um evento que indica ao programa cliente que alguma mensagem de sincronismo deve ser enviada a todos os outros clientes. Normalmente objetos que possuem alguma ação compartilhada usam este tipo de evento. Por exemplo, o ato de “clique” em um objeto o fazendo girar caracteriza uma ação é compartilhada. Então uma mensagem deve ser distribuída por toda a rede do sistema indicando que algum usuário efetuou a ação, para que desta maneira, o objeto gire para todos os clientes. A mensagem enviada está no formato: *Name!Value*, onde *Name* indica a ação e *Value* indica algum parâmetro adicional.

## 5.2.5 Protocolo de Mensagens

Neste item será apresentado o protocolo de comunicação utilizado para a criação e manutenção da rede necessária para o funcionamento do sistema multiusuário LRVCHAT3D. O sistema utiliza um protocolo simplificado sobre TCP/IP, baseado em mensagens texto do tipo: Comando + Parâmetros, com o caractere “[” como separador geral. Este é um protocolo de teste do protótipo, sendo que diversas otimizações podem ser aplicadas ao modelo.

Este sistema é facilmente expansível e adaptativo, pois como o sistema é modular, pode-se alterar completamente o protocolo de mensagens e adicionar filtros sem alterações nas interfaces 3D (VRML) e 2D (Delphi) do sistema.

O sistema possui *pilhas*<sup>53</sup> de mensagens, tanto a nível global para mensagens do sistema, como *pilhas* de comandos específicos, para cada objeto e avatar. Isto se faz necessário devido a interligação Delphi + Cortona que é assíncrona e o tempo

---

<sup>53</sup> As mensagens são empilhadas na memória, de forma que a primeira a entrar na pilha é a primeira a sair e ser processada. Isto evita perdas de mensagens devido ao processamento local.

necessário para completar uma operação é variável.

O protocolo será apresentado utilizando vários processos envolvidos na estrutura de em um sistema multiusuário deste tipo:

### 1) Cliente conectando um Servidor.

Cliente X	Servidor	Outros Clientes
	← BANNER	
ENTER(Nick) →		
	Se (Nick) existe ← REFUSED Senão ← WELCOME USER_ENTER (cliente X) → ← USER_ENTER Para todo Objeto Faça ← ADD_OBJ Fim Para	

BANNER = "banner|MOC Server|<Versão>|<Nome do Ambiente>"

ENTER = "enter|<Nick>|<URL\_Avatar>"

REFUSED = "refused"

WELCOME = "Welcome|<Nick>|<URL\_Base\_HTTP>|<Complemento\_URL\_Ambiente>"

USER\_ENTER = "user\_enter|<Nick>|<URL\_Avatar>|<X>|<Y>|<Z>|<A>"

ADD\_OBJ = "add\_obj|<Nick>|<Nome>|<URL\_Objeto>|<X>|<Y>|<Z>|<A>"

### 2) Desconectando de um Servidor.

Cliente X	Servidor	Outros Clientes
	USER_LEAVE →	

USER\_LEAVE = "user\_leave|<Nick>"

### 3) Adicionando um objeto.

Cliente X	Servidor	Outros Clientes
ADD_OBJ →		
	← ADD_OBJ →	
	← OBJ_STATUS(added)	

ADD\_OBJ = "add\_obj|<Nick>|<Nome>|<URL\_Objeto>|<X>|<Y>|<Z>|<A>"

OBJ\_STATUS(added) = "obj\_status|added"

### 4) Ativando alguma animação do avatar

Cliente X	Servidor	Outros Clientes
GESTURE →		
	← GESTURE →	

GESTURE = "gesture|<Nick>|<Animação>"

## 5) Enviando propriedades de um objeto.

Após a carga completa de um objeto pelo cliente, é necessário atualizar as propriedades, então cliente deve enviar este comando para o Servidor.

Cliente X	Servidor	Outros Clientes
GET_PRO(Obj X) →		
	Se (Obj X) não existe ← OBJ_STATUS(not_exist) Senão Para toda propriedade de Obj X ← PRO_NET Fim Para	

GET\_PRO(Obj X) = "get\_pro|<Nome>"  
 OBJ\_STATUS(not\_exist) = "obj\_status|not\_exist"  
 PRO\_NET = "pro\_net|SYSTEM|<Nome>|<valor>"

## 6) Transmitindo comandos de objetos compartilhados.

(Command\_To\_Net → Command\_From\_Net)

Cliente X	Servidor	Outros Clientes
COM_NET →		
	← COM_NET →	

COM\_NET = "com\_net|<Nick>|<Nome>|<valor>"

## 7) Renomeando um objeto.

Cliente X	Servidor	Outros Clientes
REN_OBJ (ObjX , NewN) →		
	Se (NewN) existe ← OBJ_STATUS(not_rename) Senão Se (Obj X) não existe ← OBJ_STATUS(not_exist) Senão Se (Cliente X é o dono do Obj X) ← OBJ_STATUS(not_owner) Senão ← REN_OBJ → ← OBJ_STATUS(renamed)	

OBJ\_STATUS(not\_rename) = "obj\_status|not\_rename"  
 OBJ\_STATUS(not\_exist) = "obj\_status|not\_exist"  
 OBJ\_STATUS(not\_owner) = "obj\_status|not\_owner"  
 REN\_OBJ = "ren\_obj|<Nick>|<Nome>|<NovoNome>"

## 8) Enviando mensagem texto.

Cliente X	Servidor	Outros Clientes
SEND →		
	← SEND →	

SEND = "send|<Nick>|<Mensagem>"

## 9) Enviando mensagem texto em modo privado.

Cliente X	Servidor	Cliente Y
PVT →		
	PVT →	

PVT = "pvt|<Nick>|<Para>|<Mensagem>"

## 10) Removendo um objeto.

Cliente X	Servidor	Outros Clientes
REM_OBJ (ObjX) →		
	Se (Obj X) não existe ← OBJ_STATUS(not_exist) Senão Se (Cliente X é o dono do Obj X) ← OBJ_STATUS(not_owner) Senão ← REM_OBJ → ← OBJ_STATUS(removed)	

OBJ\_STATUS(not\_exist) = "obj\_status|not\_exist"

OBJ\_STATUS(not\_owner) = "obj\_status|not\_owner"

OBJ\_STATUS(removed) = "obj\_status|removed"

REM\_OBJ = "ren\_obj|<Nick>|<Nome>"

## 11) Enviando informação de movimentação do avatar.

Cliente X	Servidor	Outros Clientes
MOVE →		
	# →	

MOVE = "move|<Nick>|<X>|<Y>|<Z>|<A>|<T>"

# = "#<Nick>|<X>|<Y>|<Z>|<A>|<T>"

Onde X,Y,Z é a posição do avatar, A é o ângulo em relação ao eixo Y e T um marcador de tempo.

## 12) Alterando alguma propriedade do objeto.

Cliente X	Servidor	Outros Clientes
PRO_OBJ (ObjX) →		
	Se (Obj X) não existe ← OBJ_STATUS(not_exist) Senão Se (Cliente X é o dono do Obj X) ← OBJ_STATUS(not_owner) Senão ← PRO_NET →	

OBJ\_STATUS(not\_exist) = "obj\_status|not\_exist"

OBJ\_STATUS(not\_owner) = "obj\_status|not\_owner"

PRO\_OBJ = "pro\_obj|<Nick>|<nome>|<valor>"

PRO\_NET = "pro\_obj|SYSTEM|<nome>|<valor>"

## 5.2.6 Programa Servidor

O programa servidor do protótipo atual foi escrito em *object pascal* (Borland Delphi) e funciona em conjunto com servidores *web* que são os repositórios dos arquivos de ambientes, avatares e objetos. A estrutura típica de diretórios em um servidor *web* utilizada no sistema é apresentada a seguir:

\AVATARS	(Diretório dos avatares)
\list.txt	(Lista de avatares disponíveis, formato nome=arquivo)
\avatar_n.wrl	(Arquivo principal do avatar n)
\OBJECTS	(Diretório dos objetos)
\list.txt	(Lista de objetos disponíveis, formato nome=arquivo)
\object_n.wrl	(Arquivo principal do objeto n)
\WORLDS	(Diretório dos ambientes)

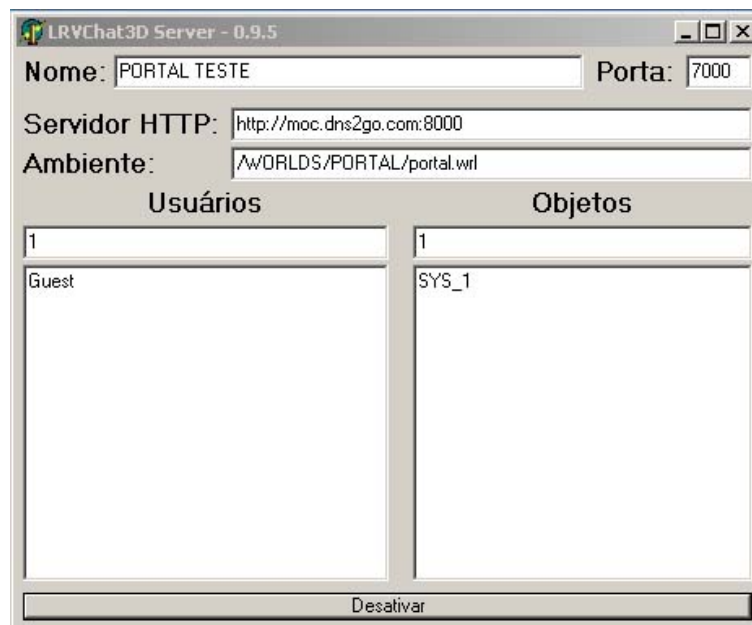


Figura 30 - Programa Servidor.

O Programa servidor é simples e controla um único ambiente, mantendo os dados necessários para a coerência do ambiente multiusuário. Na figura 30 podem ser observados os campos de configuração da interface do programa:

- **Nome:** Indica o nome do ambiente que aparecerá no topo da janela do programa cliente quando este estiver conectado ao servidor.
- **Porta:** Indica a porta utilizada por este servidor, podemos rodar vários servidores em uma mesma máquina, utilizando portas diferentes.
- **Servidor http:** Indica o endereço do servidor de repositórios onde os arquivos do ambiente servido poderão ser acessados.
- **Ambiente:** Indica o ambiente que será servido. Esta entrada indica também a localização relativa do arquivo de ambiente no servidor *web* citado acima. No exemplo da figura 31 o endereço completo do arquivo de ambiente ficaria sendo:  
`http://moc.dns2go.com:8000/WORLDS/PORTAL/portal.wrl`

Além dos campos de configuração, podem ser observados também dois campos informativos (Objetos e Usuários), que indicam a quantidade de usuários conectados e o número total de objetos ativos no ambiente, abaixo dos campos numéricos observa-se respectivamente as listas de usuários e de objetos.

Na parte inferior fica o botão de ativação e desativação do servidor.

## 5.2.7 Programa Cliente

Como já foi citado anteriormente, o protótipo atual do programa cliente foi desenvolvido em Object Pascal (Borland Delphi) utilizando o componente ActiveX Cortona VRML (ParallelGraphics) de modo a executar o protocolo apresentado em itens anteriores, assim como suportar os formatos definidos de arquivos de ambientes, avatares e objetos. Neste tópico será apresentada a interface e funcionalidade do programa cliente.

### 5.2.7.1 Inicialização

Ao se executar o programa cliente (arquivo LRVCHAT3D.exe) a primeira ação realizada pelo programa é a verificação da existência do controle ActiveX para VRML (ParallelGraphics Cortona 3.1 ou superior, para Internet Explorer). A figura 31 mostra a tela de inicialização. A parte inferior da imagem é a linha de *status* que provê o usuário com as informações de inicialização. As imagens apresentadas mostram textos em português, mas outros idiomas são suportados através da alteração dos arquivos de configuração. Uma tela informação será exibida caso o componente não seja encontrado, ou sua versão seja inferior a exigida pelo programa, como mostra a figura 32. Neste caso a versão requerida deve ser instalada para que o programa seja executado com sucesso.



Figura 31 - Tela de inicialização.

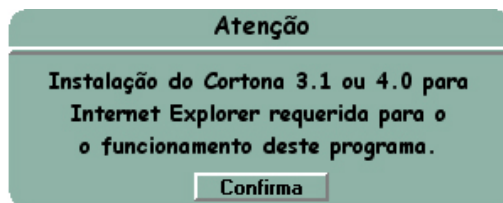


Figura 32 - Tela indicativa de componente VRML não instalado.

### 5.2.7.2 Tela principal

Caso a versão requerida do ParallelGraphics Cortona esteja devidamente instalada, o programa cliente completará sua carga e a tela ilustrada pela figura 33 será apresentada ao usuário.



Figura 33 - Tela principal.

A área grifada 1 da figura 33 é menu de acesso que apresenta diversas funções do programa que serão detalhadas na seqüência. A área 2 apresenta a lista de usuários conectados no mesmo ambiente. A região de navegação tridimensional, que é a janela de acesso ao ambiente virtual, está indicada pela área 3. É neste setor que o componente *Cortona VRML* aparecerá quando o programa estiver conectado a algum servidor. O item 4 indica a região do Chat texto onde haverá a troca de mensagens entre usuários e serve também como *display* de mensagens do



sistema. O setor marcado como **5** apresenta o *nickname* do usuário corrente e demarca a área de entrada de texto do programa.

O menu de acesso é composto dos itens:

**Conectar** – Conecta a um servidor previamente estabelecido em “Configuração / Usuário”. Na primeira que é utilizado, abre a janela “Configuração / Usuário” para que sejam escolhidas as características de acesso (*nickname*, avatar, servidor). Este item do menu muda para “Desconectar” quando o sistema estiver conectado, permitindo assim sua desconexão.

**Configuração** – Permite a alteração da configuração do usuário e do sistema gráfico de renderização (Render). Como é visto na figura 34, o menu apresenta dois subitens, Usuário e Render, os quais são detalhados no item 5.2.7.3 - *Configuração de conexão e avatar* e no item 5.2.7.4 - *Configuração de renderização*, respectivamente. Este item só é acessível quando o sistema estiver desconectado.

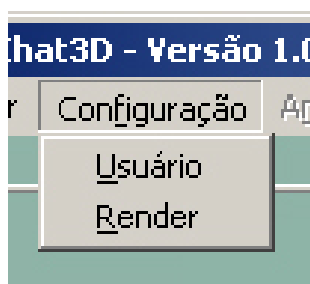


Figura 34 - Menu Configuração.

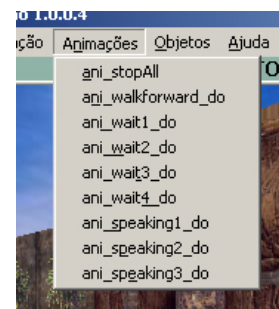


Figura 35 - Menu Animações.

**Animações** – Este menu (figura 35) possui como subitens, as animações permitidas pelo avatar selecionado pelo usuário no item 5.2.7.3 - *Configuração de conexão e avatar*. Caso o avatar não seja válido ou não possua animações, este item não apresentará subitens associados. Para acionar uma animação (ou gesto) basta clicar no item correspondente. Somente os outros usuários visualizarão a animação acionada pelo usuário, que por sua vez poderá ter uma prévia da animação acessando o menu *Configuração / Usuário*.

**Objetos** - Abre a janela de objetos que permite a seleção e inclusão de objetos no ambiente. (veja item 5.2.7.8 - *Manipulando Objetos*).

**Ajuda** – Abre o *web browser* padrão utilizado pelo usuário e mostra uma página de ajuda, que possui um conteúdo similar ao apresentado neste item (5.6).

### 5.2.7.3 Configuração de conexão e avatar

Quando é selecionado o item *Configuração / Usuário* no menu, a janela ilustrada na figura 36 é apresentada ao usuário, permitindo assim, a configuração do usuário (*nickname* e avatar) e do servidor que será acessado.

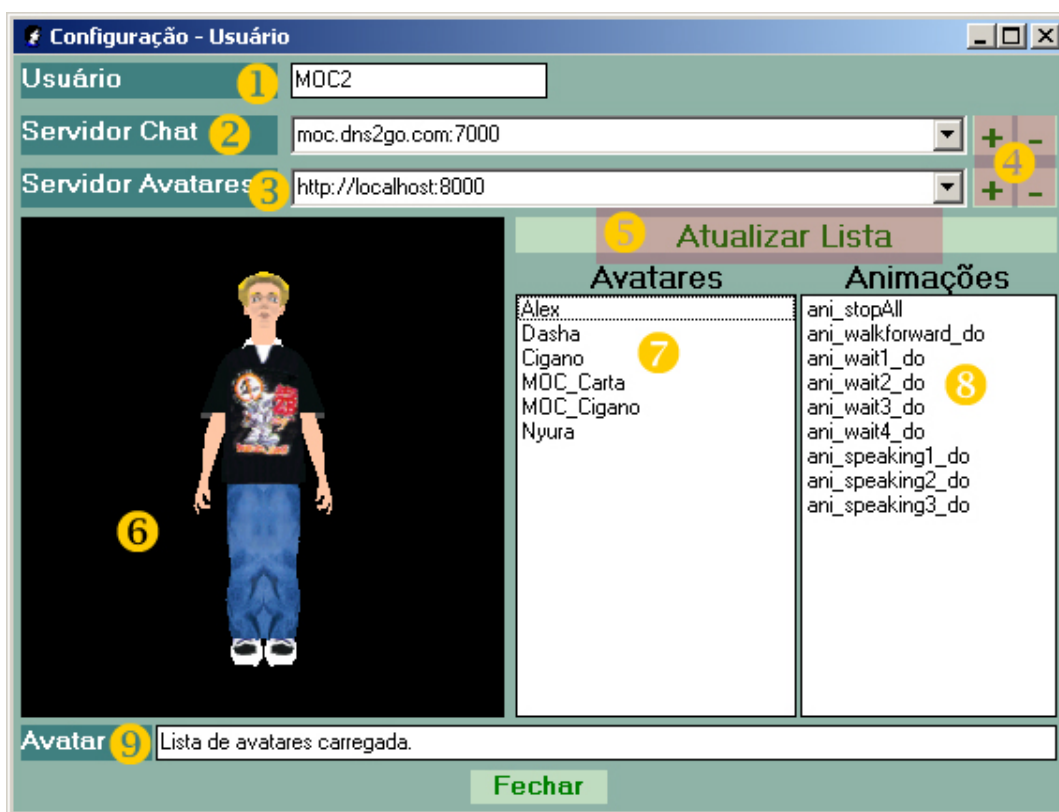


Figura 36 - Configuração de avatar e conexão.

Os itens grifados são os seguintes:

1. Nome do usuário (nickname).
2. Endereço do servidor de chat a ser conectado (no formato nome:porta ou IP:porta).
3. Endereço do servidor base para obtenção de avatares (no formato endereço base http).
4. Botões de inclusão e exclusão de endereços de servidores na lista de acesso rápido mantida pelo programa.
5. Botão para atualizar a lista de avatares disponíveis (item 7) no servidor base (item 3).
6. Janela VRML para visualização do avatar selecionado.
7. Lista de avatares disponíveis no servidor base.
8. Lista de animações disponíveis para o avatar selecionado.
9. Endereço real do avatar selecionado. Esta área também serve como área informativa, indicando procedimento de carga de avatar ou erros.

Caso o endereço do arquivo de avatar seja inválido, aparecerá um avatar padrão (cilindro com uma esfera no topo, ambos vermelhos). Ao selecionar uma animação, a janela de visualização VRML (item 6) apresentará o avatar realizando a animação escolhida, como demonstrado na figura 37 onde 1 indica a seleção da animação e 2 o avatar executando a animação.

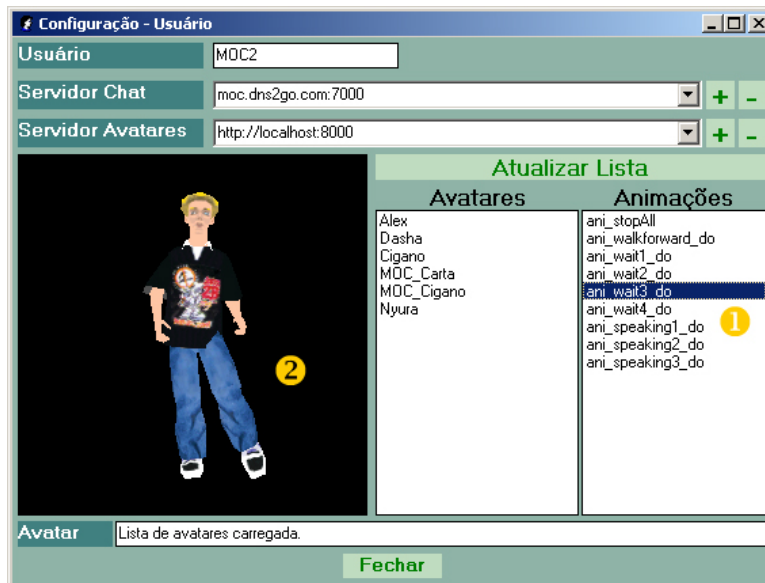


Figura 37 - Visualizando animação do avatar.

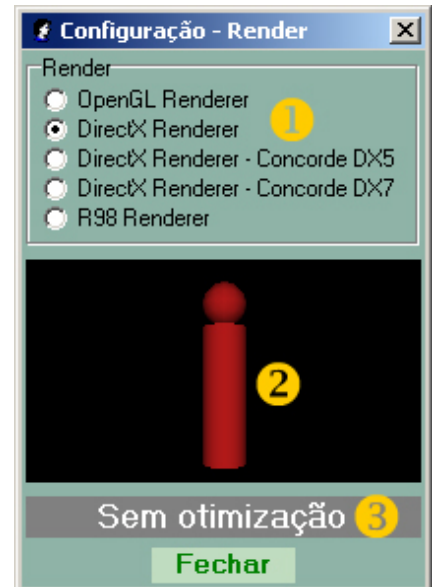


Figura 38 - Seleção de render.

#### 5.2.7.4 Configuração de renderização<sup>54</sup>

A configuração do tipo de renderização é acionada através do menu *Configuração / Render* que é apresentado ao usuário através da janela ilustrada na figura 38. O item 1 mostra os renderes disponíveis e qual está ativo no momento, o item 2 é uma janela VRML para inspeção do resultado da alteração de render, e finalmente o item 3 é um indicador que sinaliza a presença ou não de otimização para os processadores *Pentium 3* ou superiores. Esta otimização é ativada automaticamente pelo programa cliente.

<sup>54</sup> Renderização é o processo de criação de uma imagem de um objeto modelado tridimensionalmente, simulando sua aparência de forma realística, utilizando textura, cor, sombras de superfície, destaques e reflexões.







### 5.2.7.5 Explorando o ambiente

A navegação pelo ambiente é realizada utilizando o mouse e/ou teclado. Usando o mouse deve-se clicar e arrastar para que o deslocamento ocorra. Quanto maior a distância de arrasto do mouse maior vai ser a velocidade do movimento. O movimento também pode ser acelerado utilizando as teclas SHIFT, CTRL ou SHIFT+CTRL em conjunto com o movimento de arraste do mouse. A relação entre o movimento de arraste do mouse e o movimento de câmera é o seguinte:

- Arrasto para longe (frente) = movimento para frente
- Arrasto para perto (trás) = movimento para trás
- Arrasto para a direita = rotação para a direita
- Arrasto para a esquerda = rotação para a esquerda

Alguns objetos podem apresentar funções específicas, utilizando sensores de interação. O quadro 6 apresenta uma lista completa dos cursores correspondentes a estas interações:

**Quadro 6 - Sensores de interação.**

Cursor	Interação
	<b>Sensor de toque.</b> Detecta um click no objeto. O resultado desta ação depende da programação de funcionalidades do objeto adicionadas pelo autor do objeto.
	<b>Ancora.</b> Representa um link HTML, não é utilizada no LRVCHAT3D.
	<b>Sensor cilíndrico.</b> Transforma o movimento de clicar no objeto e arrastar em rotação sobre um eixo do objeto.
	<b>Sensor esférico.</b> Transforma o movimento de clicar no objeto e arrastar em rotação sobre dois eixos do objeto.
	<b>Sensor plano.</b> Transforma o movimento de clicar no objeto e arrastar em movimento do objeto sobre um plano.
	<b>Sensor de adição.</b> Utilizado em operações de arrastar e soltar. Não é utilizado no LRVCHAT3D.

### 5.2.7.6 Tela de Objetos

A tela de objetos (Figura 39) é acessível através do menu Objetos, segue uma descrição de suas partes componentes e respectivas funções:



Figura 39 - Janela de seleção de Objetos.

- 1 - Endereço do servidor base para obtenção de objetos (no formato endereço base http).
- 2 - Botões de inclusão e exclusão de endereços de servidores na lista de acesso mantida pelo programa.
- 3 - Botão para atualizar a lista de objetos (item 4) disponíveis no servidor base (item 1).
- 4 - Lista de objetos disponíveis no servidor base.
- 5 - Janela VRML para visualização do objeto selecionado.
- 6 - Endereço real do objeto selecionado. Esta área também serve como área informativa, indicando carga de objetos ou erros.

### 5.2.7.7 Adicionando um objeto ao ambiente

As telas apresentadas na figura 40 ilustram o processo de inclusão de um objeto no ambiente virtual.

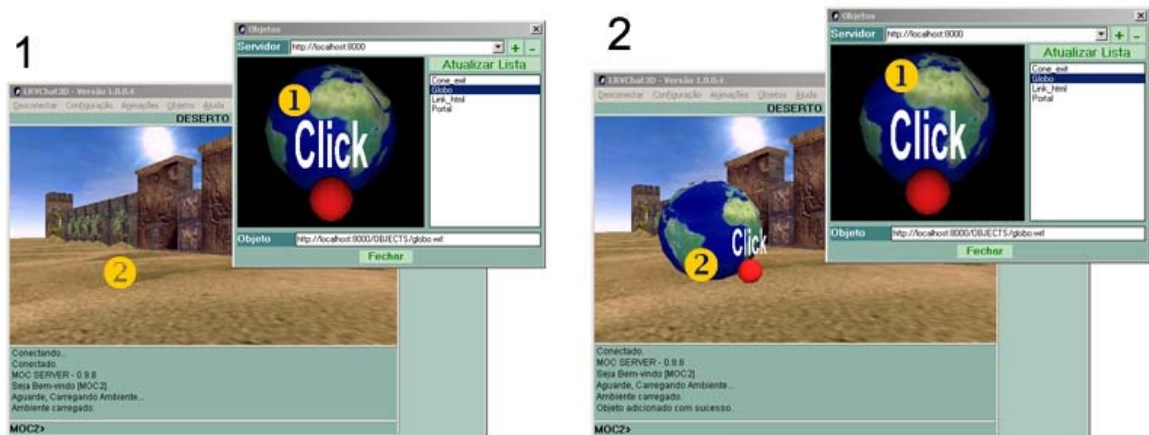


Figura 40 - Adicionando um objeto ao ambiente.

**Passo 1** - Clicar e manter pressionado o botão direito do mouse sobre o objeto selecionado (1).

**Passo 2** - Arraste e solte sobre um ponto base (2) no ambiente virtual. É neste ponto no espaço que a base do objeto estará inicialmente posicionada quando este for efetivamente incluído no ambiente.

*Observação:* Conforme o item 5.2.7.8 - Manipulando objetos, será adicionado um objeto com um nome genérico, que pode ser alterado em seguida.

### 5.2.7.8 Manipulando objetos

A manipulação de objetos no ambiente é realizada com o auxílio de menus flutuantes como o apresentado no item 2 da figura 41. Este menu é acessível através do click do botão direito do mouse sobre o objeto de interesse. Para facilitar a identificação dos objetos, um *hint* (pequeno texto flutuante) surge quando se movimenta o cursor do mouse sobre um objeto, indicando o nome do mesmo conforme apresentado no item 1 da figura 41.

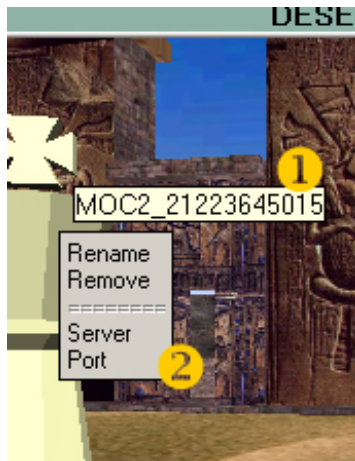


Figura 41 - Menu auxiliar do objeto.



Figura 42 - Menu auxiliar do Avatar.

O item 2 indicado na figura 41 aponta para os itens que representam as propriedades deste objeto específico. Para visualizar ou alterar o valor destas propriedades basta clicar no item escolhido. Para renomear ou remover um objeto basta selecionar as opções correspondentes no menu do objeto.

*Observação:* Alterações nas propriedades dos objetos, ou exclusão do mesmo só podem ser realizadas pelo usuário que adicionou o objeto ao ambiente (owner).

#### 5.2.7.9 Enviando mensagens privadas

Assim como no caso de objetos, um menu flutuante, como o apresentado no item 2 da figura 42, dá acesso às funções do avatar. Na versão da Fase 2 do sistema somente a opção de enviar mensagem para o usuário representado pelo avatar está disponível. Para facilitar a identificação dos objetos um *hint*, como o apresentado no item 1 da figura 42, surge quando se movimenta o mouse sobre um avatar indicando o *nickname* utilizado pelo usuário que este representa.

### 5.3 Fase III

A fase III do projeto originalmente previa aplicações do LRVCHAT3D e a utilização de agentes inteligentes, esta fase foi parcialmente desenvolvida durante o período de mestrado do autor, sendo que o desenvolvimento do uso de agentes inteligentes não foi efetivado. Por duas razões: Limitação do tempo disponível e a constatação que o atual desenvolvimento do LRVCHAT3D permite o desenvolvimento de agentes inteligentes sem a necessidade de alterações nos protocolos e processos envolvidos. Com isto outras pessoas podem em trabalhos futuros, efetivar a utilização de técnicas de agentes inteligentes no LRVCHAT3D. Duas aplicações acadêmicas do protótipo foram desenvolvidas e apresentadas nos trabalhos 1 e 2 detalhados a seguir, o terceiro item é uma extensão do LRVCHAT3D com fins comerciais.:

#### **1 - Proposta de Plataforma para Experimentos em Realidade Virtual.**

Este foi o trabalho de doutorado (LUZ, 2002) de Rodolfo Pinto da Luz, que apresenta a proposta de um projeto de uma plataforma para criação e realização de experimentos virtuais para ensino que emprega um conceito de reutilização de objetos. Essa plataforma permite que professores possam, sozinhos, preparar experimentos virtuais com o reaproveitamento de conhecimento. A validação da proposta de Luz foi feita com o desenvolvimento de um protótipo que contempla os principais aspectos de seu projeto: 1) reutilização de objetos e experimentos, 2) colaboração, 3) aplicação do modelo SCORM e 4) utilização de um padrão de objetos e experimentos virtuais. A solução para a criação de uma metodologia para experimentos distribuídos e colaborativos em realidade virtual foi aplicada ao protótipo, que provou ser possível o desenvolvimento de experimentos de laboratório e sua reutilização em diferentes sistemas de ensino on-line. O reaproveitamento acontece dentro da própria plataforma ou em sistemas de gerenciamento de ensino (LMS) em conformidade com o SCORM. O protótipo evidencia, também, que os alunos podem realizar os experimentos virtuais em caráter individual ou colaborativo.

O trabalho apresentado nesta dissertação (LRVCHAT3D) colaborou com o projeto de Luz através do desenvolvimento de uma versão especial do LRVCHAT3D,



onde foi adicionada comunicação por voz e suporte a objetos com interfaces sincronizadas em tempo real, denominados de *Net Interfaces* (Vide apêndice C).

O LRVCHA3D foi utilizado neste projeto como plataforma multiusuário suportando o ambiente de simulação de experiências proposto por Luz.

Na figura 43 temos dois usuários realizando uma experiência de física (cálculo do coeficiente de atrito através do ângulo crítico) dentro de um ambiente virtual utilizando o LRVCHAT3D. A imagem mostra a visão em primeira pessoa de um dos participantes, onde se vê o avatar do outro participante e o equipamento utilizado na experiência. Ambos os participantes podem interagir com o experimento.



Figura 43 - Captura de tela do VIRTUALLAB.

## 2 – VIRTUALNET, Laboratório Virtual 3D de Redes de Computadores.

VIRTUALNET (HASSAN, 2003) foi o trabalho de mestrado de Elizangela B. Hassan, que apresentou um protótipo de um ambiente virtual tridimensional para aprendizagem de Redes de Computadores. Através da utilização da linguagem VRML integrada com recursos multimídia e com uma ferramenta de comunicação para bate-papo em 3D, o VIRTUALNET foi desenvolvido com base nas abordagens de aprendizagem colaborativa e construtivista. Um dos ambientes de ensino de redes foi adaptado para o uso com uma versão do LRVCHAT3D, como pode ser

observado na figura 44, utilizando-o como plataforma multiusuário do projeto.

A figura 44 mostra um ambiente virtual de ensino funcionando dentro do LRVCHAT3D, onde o aluno monta topologias de rede, arrastando elementos como terminais, barramentos de rede e servidores, colocando-os sobre a mesa e interligando-os em diversas configurações conforme a teoria de redes de computadores, este ambiente possui um quadro onde o aluno pode obter informações teóricas sobre estas configurações.

O ambiente desenvolvido por Hassan apresenta ainda um sistema de avaliação que é ativado pelo aluno após terminar de montar uma configuração, este sistema avaliador verifica se a topologia solicitada ao aluno foi devidamente montada e fornece ao aluno informações relativas a esta avaliação.

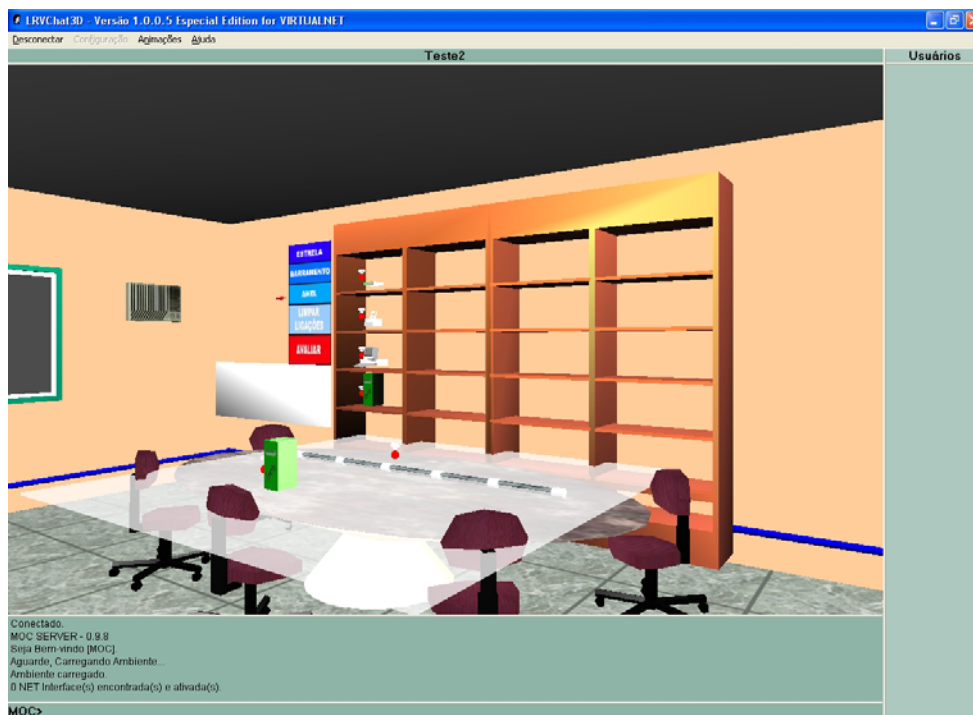


Figura 44 - Captura de tela do VIRTUALNET.

### 3 – MEDIAPLATAFORM.

Uma versão comercial está sendo derivada do LRVCHAT3D, e atualmente se encontra em desenvolvimento em parceria com a empresa Mediasoft – Softwares e Produções Multimídia Ltda (MEDIASOFT, 2003). Esta versão é denominada de *MediaPlataform* e consiste de um sistema servidor (Figura 45), controlado via Internet e uma aplicação cliente (Figura 46).

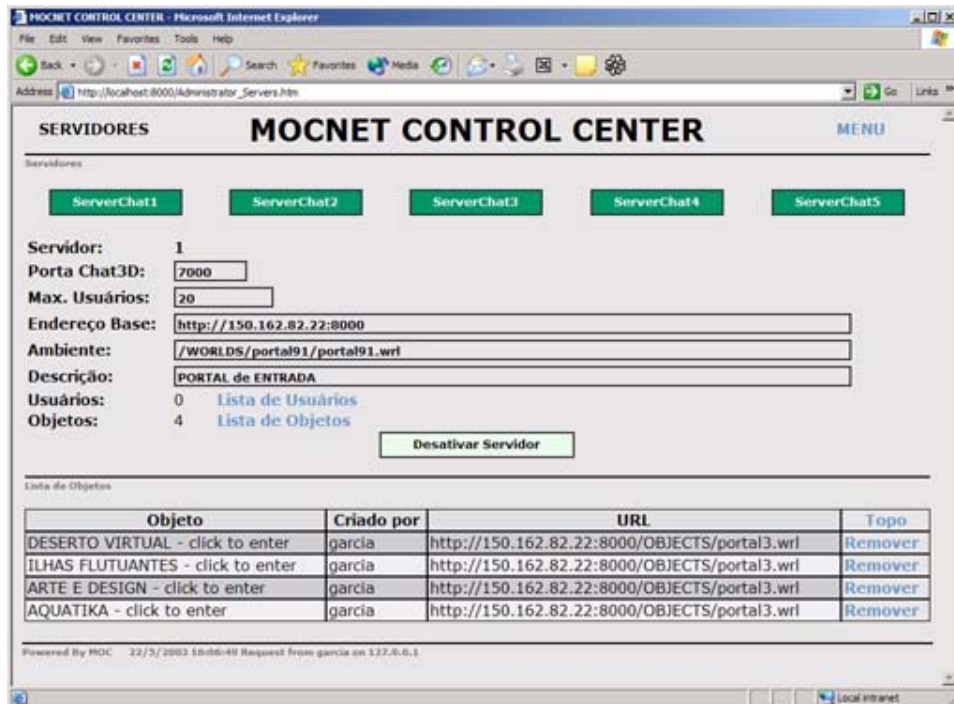


Figura 45 - Captura de tela da interface WEB do MediaServer.

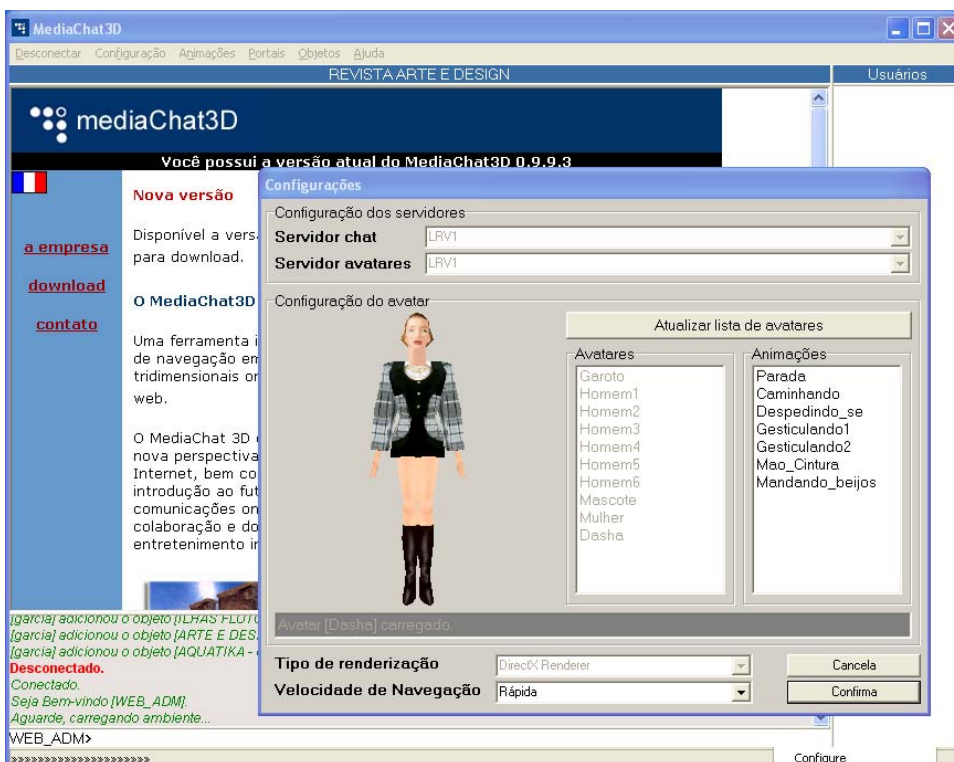


Figura 46 - Captura de tela do MediaChat3D.

Este é um sistema ferramenta, assim como é o LRVCHAT3D, pode-se, portanto explorar esta plataforma de diversas maneiras. O servidor atual suporta cinco ambientes, que podem ter objetivos diferentes, por exemplo, podemos ter um

ambiente que seja educacional, um ambiente que seja uma loja virtual, um ambiente que sirva para treinamento ou mesmo uma simples sala de reuniões.

O sistema possui as seguintes características:

- Abriga ambientes virtuais 3D - salas de aula, salas de reuniões, clubes, lojas, espaços culturais, empresas e páginas pessoais;
- Possui uma comunicação síncrona via texto;
- Possui uma comunicação assíncrona através de murais e e-mails;
- Colaboração entre os usuários através de objetos 3D, com aplicações compartilhadas;
- Administração dos usuários e controle de permissões de ações nos ambientes virtuais;
- Perfil de usuários, análise das atividades e estatísticas, através de banco de dados.

## 5.4 Considerações finais

Pode-se concluir que o desenvolvimento do protótipo e suas aplicações, apresentados neste capítulo foi bem sucedido na medida que foi demonstrado a viabilidade na construção de sistemas que suportam um ambiente virtual multiusuário tridimensional com interatividade e que possibilitem a colaboração entre os usuários. O desenvolvimento deste sistema foi de grande importância para a absorção do conhecimento teórico de ambientes virtuais em rede, servindo de base para futuras aplicações desta tecnologia nas diversas áreas do conhecimento.

As aplicações apresentadas e o sistema comercial que atualmente está em desenvolvimento comprovam a validade da iniciativa de se desenvolver um sistema próprio ao invés da utilização de um sistema de terceiros, normalmente baseados em arquitetura totalmente fechada. O fato de o LRVCHAT3D usar como padrão de arquivos o formato VRML facilitou a adoção do protótipo nas aplicações apresentadas, pois desta forma o desenvolvimento das aplicações e da ferramenta estiveram isolados, permitindo que os responsáveis pelo desenvolvimento das aplicações tivessem liberdade e independência.

Enfim, o desenvolvimento do LRVCHAT3D cumpriu suas metas iniciais e se mostrou uma ferramenta interessante para estudo e aplicação de ambientes virtuais, e se apresenta como uma ferramenta de uso potencial no EAD, conforme demonstrado nas duas aplicações acadêmicas apresentadas.

## 6. CONCLUSÕES E RECOMENDAÇÕES

No decorrer deste trabalho foi apresentada uma visão geral sobre Realidade Virtual e mais especificamente sobre Ambientes Virtuais em Rede (AVR). Por fim foram apresentados o desenvolvimento e a implementação de um protótipo de AVR (LRVCHAT3D). A apresentação teórica buscou não somente apresentar os conceitos base deste trabalho, mas também fornecer ao leitor uma visão atual sobre o assunto, mostrando sua enorme potencialidade em diversos campos. O LRVCHAT3D demonstrou sua viabilidade técnica e financeira, pois os recursos utilizados no seu desenvolvimento, software e hardware, são considerados de baixo custo em relação ao que normalmente se espera de um projeto em Realidade Virtual (RV). Sabe-se que parte da idéia, até preconcebida, que RV é considerada distante da realidade de uma nação como o Brasil, onde as verbas para pesquisa e desenvolvimento são poucas.

O uso da Realidade Virtual como suporte ao Ensino à Distância via Internet, por exemplo, já é uma realidade. Mesmo em sua fase inicial (Fase I) este projeto já poderia ser aplicado como ferramenta de suporte ao EAD, servindo como interface de sistemas de auxílio ao aluno onde monitores podem interagir com os alunos e estes entre si. Na fase I o sistema facilita a sociabilidade e torna mais agradável a utilização da monitoria. Uma outra aplicação seria a realização de palestras em um ambiente virtual.

A fase II do projeto ampliou os elementos de interação, permitindo a realização de aulas e a troca efetiva de informações não textuais. E as aplicações apresentadas na fase III, demonstraram o uso do LRVCHAT3D como ferramenta de suporte ao EAD. Estas aplicações são protótipos, sendo necessária uma confirmação de sua eficácia em sistemas reais de EAD, não sendo este o objetivo deste trabalho.

Outros trabalhos correlatos estão sendo desenvolvidos pela equipe do Laboratório de Realidade Virtual (PPGEP/UFSC) e laboratórios conveniados. Entre estes trabalhos podemos citar: pesquisa com agentes inteligentes na educação e desenvolvimento de uma ferramenta de criação de laboratórios virtuais para ensino a distância via Internet.

Conforme apresentado neste trabalho pode-se observar a importância dos

ambientes virtuais colaborativos, sua utilização em larga escala em diversas áreas parece inevitável, principalmente se ocorrer o desenvolvimento do poder de processamento gráfico e de rede que esperamos para este novo milênio.

Como recomendação para trabalhos futuros fica a implementação de agentes inteligentes como sugerido na terceira fase do projeto, além da sugestão de um trabalho de aplicação do protótipo gerado em um caso real de EAD, para uma avaliação preliminar de seu desempenho e eficácia, servindo de baliza para uma nova área de pesquisa e desenvolvimento.

Durante a execução deste trabalho observou-se a existência de pontos que merecem ser pesquisados individualmente, tais como, otimização nos protocolos de rede, avatares, captura de movimentos, comunicação não textual e metodologias de produção de conteúdo para uso como o sistema desenvolvido.

Um item ainda pouco explorado em AVR e AVC que merece uma dedicação especial em trabalhos futuros é estudo detalhado dos aspectos humanos, que vão desde o comportamento social, até a ergonomia e usabilidade nas interfaces homem-máquina.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

ACTIVEWORLDS Inc., Disponível em <<http://www.activeworlds.com>>. Acesso em: 1 nov. 2001.

AVATARA. What is an Avatar. Disponível em <<http://www.avatara.org/essay.html>>. Acesso em: 12 nov. 2001.

BABSKI, C.; THALMANN, D., **Real-Time Animation and Motion Capture in Web Human Director**, Proceedings of Web3D & VRML 2000 Symposium. ACM Press, Monterey, California, p. 139-145, 2000.

BABSKI, C., THALMANN, D., **Define Virtual Humans On The Web**, Software Focus, Wiley, Vol. 1, No1, p. 6-14, 2000.

BANGAY, S., **Modelling Parallel and Distributed Virtual Reality Systems for Performance Analysis and Comparison**, 1996. Phd Thesis, Department of Computer Science, Rhodes University. Grahamstown - South African, 1996.

BELLMAN, Kirstie L., **Playing in the MUD – Turning Virtual Reality into Real Education and Training** in Virtual Reality, Training's Future? - Perspectives on Virtual Reality and Related Emerging Technologies. Defense Research Series – Volume 6, Edited by Robert J. Seidel and Paul R. Chatelier. Plenum Press – New York and London. Published in cooperation with NATO Defense Research Group, p. 9-18, 1997.

BENEDIKT, Michael, **Cyberspace: First Steps**. Cambridge, MA: The MIT Press, 1991.

BENFORD, S. D., GREENHALGH, C., **Collaborative Virtual Environments**, Communications of the ACM, Vol. 44 No. 7 p. 79. ACM Press, MA, Julho 2001.

BLAXXUN Interactive, Disponível em <<http://www.blaxxun.com>>. Acesso em: 1 nov. 2001.

BORLAND Software Corporation, Disponível em <<http://www.borland.com>>. Acesso em 1 nov. 2001.

BROLL, Wolfgang, **Extending VRML to Support Collaborative Virtual Environments**, in Proceedings of CVE'96, Workshop on Collaborative Virtual Environments, University of Nottingham, UK, Sept. 19-20, 1996.

BURDEA, G.; COIFFET, P., **Virtual Reality Technology**. New York, NY: John Wiley & Sons, 1994.

BYRNE, C. M. **Use of Virtual Reality as Educational Tool**. Washington: Washington University, 1995.

ÇAPIN, T.; PANDZIC, I. et al. **Avatars in Networked Virtual Environments**. New York, NY: John Wiley & Sons, 1999.

ÇAPIN, T.K., ESMERADO, J., THALMANN, D., **Dead-reckoning algorithms for streaming virtual human data**, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Press, Vol.9, No3, p. 411-414, 1999.

CAREY, Rick; BELL, Gavin, **The Annotated VRML 97 Reference**, Addison-Wesley, 1997.

CRUZ-NEIRA, C.; Sandin, D.J.; DeFanti, T.A., **Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE**. In *Proceedings of SIGGRAPH 93 Computer Graphics Conference*, ACM SIGGRAPH, p. 135-142, Anaheim, CA, 1993.

CURTIS, P. Mudding: **Social Phenomena in Text-Based Virtual Realities**, Proceedings of Directions and Implications of Advanced Computing (DIAC'92), May 1992.

CYCO Entrance, Disponível em <<http://www.cycosys.com/entrance/index.html>>. Acesso em: 1 nov 2001.

DAMER, Bruce. **Avatars! : Exploring and Building Virtual Worlds on the Internet**. Berkeley, CA: Peachpit Press. 1998.

DAUB, Michael D. A Brief History of VRML. Disponível em <[http://www.astronomy.ohio-state.edu/~stephens/html/vrml/vrml\\_history.html](http://www.astronomy.ohio-state.edu/~stephens/html/vrml/vrml_history.html)>. Acesso em: 1 nov. 2001.

ELLIS, S.R. **Nature and Origin of Virtual Environments**: A Bibliographic Essay, Computing Systems in Engineering, Vol. 2, No. 4, p. 321-347, 1991.

EMERING, L., BOULIC, D., THALMANN, D., **Body Expression In Virtual Environments**, in Proc. ROMAN'99, Pisa, Italy, 1999.

FRECON, E.; STENIUS, M. **DIVE: Scalable Network Architecture for Distributed Virtual Environments**. Distributed Systems Engineering Vol. 5 No. 3. September 1998.

GARCIA, Plínio Silva de, BRUNO, Gaspare Giuliano Elias, **Introdução ao Multicast**. Disponível em <<http://prav.unisinos.br/tutoriais/multicast.php>>. Acesso em 1 mai. 2003.

GIBSON, William. **Neuromancer**. New York: Ace Books, 1984.

H-ANIM, Humanoid Animation Working Group. Disponível em <<http://H-Anim.org>>. Acesso em: 1 nov. 2001.

HASSAN, Elizangela B. **VIRTUALNET – Laboratório Virtual 3D de Redes de Computadores**. 2003. 146f. Dissertação (Mestrado em Ciências da Computação) – Instituto de Informática, Universidade federal do Rio Grande do Sul, Porto Alegre, 2003.

HEDRICK, Charles L. Introduction to the Internet Protocols. Disponível em <<http://oac3.hsc.uth.tmc.edu/staff/snewton/tcp-tutorial/>>. Acesso em: 1 mai. 2003.

HONDA, Yasuaki, MITRA, Y., ROCKWELL, B. Living Worlds, Making VRML 97 Applications Interpersonal and Interoperable. Disponível em <[http://www.web3d.org/WorkingGroups/living-worlds/draft\\_2/index.htm](http://www.web3d.org/WorkingGroups/living-worlds/draft_2/index.htm)>. Acesso em: 1 nov. 2001.

ICQ Inc., Disponível em <<http://web.icq.com>>. Acesso em 1 mai. 2003.

IMMERSION Corporation. Disponível em <<http://www.immersion.com>>. Acesso em 1 nov. 2001.

IRC, Internet Relay Chat. Disponível em <<http://www.irc.org>>. Acesso em 1 mai. 2003.

---

LRVCHAT3D, Desenvolvimento de um Ambiente Virtual Tridimensional Multiusuário para Internet.



KALAWSKY, R. **The Science of Virtual Reality and Virtual Environments**, Wokingham, UK: Addison-Wesley, 1993.

LATTA, J. N., OBERG, D. J. **A conceptual virtual reality model**. IEEE Computer Graphics & Applications, 14(1): p. 23-29, Jan. 1994.

LRV – Laboratório de Realidade Virtual. Disponível em <<http://www.lrv.ufsc.br>>. Acesso em 1 mai. 2003.

LUNATIC Interactive GmbH. Disponível em <<http://www.lunatic.de/ge/projekt.htm>>. Acesso em 1 mai. 2003.

LUZ, Rodolfo Pinto da. **Proposta de especificação de uma plataforma de desenvolvimento de ambientes virtuais de baixo custo**. 1997. 108f. Dissertação (Mestrado em Engenharia de Produção, Inteligência Aplicada) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1997.

LUZ, Rodolfo Pinto da. **Proposta de Plataforma para Experimentos em Realidade Virtual**. 2002. 102f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

LYNCH K. M., MAEKAWA H., TANIE K., **Manipulation and active sensing by pushing using tactile feedback**, Proc. of the IEEE International Conference on Intelligent Robots and Systems IROS'92, p. 416-421, Raleigh, NC, USA, July 1992.

MACEDONIA, M. R.; ZYDA, M. J.; PRATT, D. R.; BARHAM, P. T.; ZESWITZ, S., **NPSNET: A Network Software Architecture for Large Scale Virtual Environments**. Presence, Vol. 3, No. 4. Fall 1994.

MACEDONIA, M. R.; ZYDA, M. J.. **A taxonomy for networked virtual environments**. IEEE Multimedia, 4(1):p. 48--56, Jan.-Mar. 1997.

MAIR, G. M.; FRYER, R. J.; HENG, J. G., **An Experiment In Three Dimensional Telepresence**; (University of Strathclyde), G.Chamberlin (BT Laboratories); Proceedings of IEE Colloquium, London, 12 May 1995.

MANIA, Katerina; CHALMERS Alan, **A Classification for User Embodiment in Collaborative Virtual Environments**. In Proc. of the 4th International Conference on Virtual Systems and Multimedia, p. 177-182. IOS Press - Ohmsha, Ltd., November 1998.

MARK, W.R., RANDOLPH, S.C., FINCH, M., VAN VERTH, J.M., TAYLOR III, R.M., **Adding Force Feedback To Graphics Systems: Issues And Solutions**, in Proc. SIGGRAPH 96, New Orleans, p. 447-452, August , 1996.

MCLEAN, Clinton Brett. **Multi-User, Interactive Virtual Reality via the Internet**. 1997. Thesis for Degree of Honours in Computer Science of Rhodes University. Grahamstown - South African, 1997.

MEDIASOFT Softwares e Produções Multimídia Ltda. Disponível em <<http://www.mediasoft.com.br>>. Acesso em 1 mai. 2003.

MICROSOFT Corporation. Disponível em <<http://www.microsoft.com>>. Acesso em 1 nov. 2001.

MORNINGSTAR, Chip; FARMER, F. Randall. **The Lessons of Lucasfilm's Habitat**, in *Cyberspace: First Steps*, ed. Michael Benedict. Cambridge, MA: The MIT Press, 1991.

PARALLELGRAPHICS. Disponível em <<http://www.parallelgraphics.com>>. Acesso em: 1 nov. 2001.

PESCE, Mark. **Ritual and the Virtual**, *Consciousness Reframed*, Center for the Advanced Inquiry into the Interactive Arts, University of Newport, Wales, 1997.

PIMENTEL, K., TEIXEIRA, K. **Virtual reality – through the new looking glass**. 2. Ed. New York : MacGraw-Hill, 1995.

REBELO, Irla B., LUZ, Rodolfo Pinto da, **New Technologies Help to Enhance the Knowledge – Oscar Niemeyer Project**. Proceedings of the 4th International Conference on Virtual Systems and Multimedia: Future Fusion - Application Realities for the Virtual Age, Vol. One - p. 286-291, Gifu – Japan, 1998.

RECHTSTEINER, Betsy, VRML – History. Disponível em <<http://userpages.umbc.edu/~erecht1/hist.html>>. Acesso em: 1 nov. 2001.

RHEINGOLD, Howard. **Virtual Reality**. Secker & Warburg, London, 1991.

RODELLO, Ideberto A. et al, **Análise dos Principais Fatores Para o Desenvolvimento de Sistemas Distribuídos de Realidade Virtual**; Proceedings of 4th SBC Symposium on Virtual Reality, Florianópolis, p. 314-325, 16-19 October 2001.

ROSA Jr., Onivaldo, **Ambientes Virtuais Cooperativos, LRVCHAT3D, Um Estudo de Caso**; Proceedings of 4th SBC Symposium on Virtual Reality, Florianópolis, p. 1-11, 16-19 October 2001.

RUSPINI, D., KOLAROV, K., KHATIB, O., **Haptic Interaction in Virtual Environments**, Proc. of the IEEE International Conference on Intelligent Robots and Systems IROS'97, Grenoble, France, September 1997.

SCHUEMIE, M.J.; VAN DER MAST, C.A.P.G., **Presence: Interacting in VR?** in: A. Nijholt, O. Donk & B. van Dijk (Eds.) Proceedings Twente Workshop on Language Technology 15, TWLT 15, ISSN 0929-0672, p. 213-217, 1999.

SENSE8 Incorporated. Disponível em <<http://www.sense8.com>>. Acesso em 1 nov. 2001.

SHERIDAN, T.B. **Musings on Telepresence and Virtual Presence** in Presence: Teleoperators and Virtual Environments, Vol. 1, No. 1, p. 120-126. 1994.

SHERMAN, William R. CS397WRS Course Notes, NCSA Renaissance Experimental Lab. Disponível em <[http://archive.ncsa.uiuc.edu/VR/cs397wrs/Lectures/chapter\\_1.html](http://archive.ncsa.uiuc.edu/VR/cs397wrs/Lectures/chapter_1.html)>. Acesso em: 1 nov. 2002.

SHIRMOHAMMADI, S.; GEORGANAS, N. D., **Collaborating in 3D Virtual Environments: A Synchronous Architecture**, IEEE 9th Inter. Workshops on Enab. Technol. Infr. For Collabor. Entreprises (WETICE) Knowledge Media Networking workshop, NIST, Washington. 14 - 16 June 2000.

SINGHAL, S.; ZYDA, M. **Networked Virtual Environments: design and implementation**. MA: ACM Press, 1999.

SUTHERLAND, Ivan. **The Ultimate Display**. NY: IFIP, p.506-508, 1965.

SUTHERLAND, Ivan. **A Head-Mounted Three Dimensional Display** in Fall Joint Computer Conference 1968, Thompson Books, Washington, DC, p. 757-764, 1968.

STEPHENSON, Neal. **Snow Crash**, NY: Bantam Books, 1992.

THERE Inc. Disponível em <<http://www.there.com>>. Acesso em 1 mai. 2003.

VRND Project, Disponível em <<http://www.vrndproject.com>>. Acesso em 1 mai. 2003.

WEB3D Consortium. Disponível em <<http://web3d.org>>. Acesso em: 1 mai. 2003.

WHITE, Stephen F., VRML Interchange Protocol – Specification.  
Disponível em <<http://www.csclub.uwaterloo.ca/~sfwhite/vnet/VIP.html>>.  
Acesso em: 1 nov. 2001.

## 8. APÊNDICES

### 8.1 Apêndice A - Exemplos completos de código de Objetos.

#### 8.1.1 Objeto 1 - Cone de fuga

Este é um objeto em forma de cone, com a palavra “EXIT” sobre ele, quando clicado efetua a desconexão do usuário que o clicou do ambiente.

```
#VRML V2.0 utf8

PROTO MOC_OBJECT [
  exposedField SFFloat   Version1.0
  exposedField SFString  Type ""
  exposedField SFString  Desc ""
  exposedField SFBool    Private FALSE
  eventIn        SFBool   Properties
  eventIn        SFString Command
  eventIn        SFString Command_From_Net
  eventOut       SFString Response
  eventOut       SFString Command_To_Net
]
{
  Transform {
    translation 0 1 0
    children [
      Shape {
        appearance Appearance {
          material DEF CYAN Material {
            diffuseColor 0.2 0.7 0.8
            specularColor 1 1 1
            shininess 0.4
          }
        }
        geometry Cone {}
      }
      Transform {
        translation 0 1 0
        children Shape {
          geometry Text {
            fontStyle FontStyle {
              family "SANS"
              style "BOLD"
              justify "MIDDLE"
            }
            string "EXIT"
            length 2
          }
        }
      }
    ]
  }
  DEF TS TouchSensor {}
  DEF ACTION Script {
```

```

directOutput TRUE
mustEvaluate TRUE
field      MFString  Param []
eventIn    SFBool    Properties      IS Properties
eventIn    SFString  Command         IS Command
eventIn    SFString  Command_From_Net IS Command_From_Net
eventOut   SFString  Response        IS Response
eventOut   SFString  Command_To_Net  IS Command_To_Net
url "javascript:
    function Properties(value,timestamp)
    {
        Response = '%PROP%';
    }
    function Command(value,timestamp)
    {
        if (value=='Click') { Response = '%EXIT%'; }
    }
    function Command_From_Net(value,timestamp)
    {
    }
"}
DEF AUX Script {
    directOutput TRUE
    mustEvaluate TRUE
    eventIn  SFTIME  In
    eventOut SFString Out
    url "javascript:
        function In(value,timestamp)
        {
            if (value) Out = 'Click';
        }
    "
}
]
}
ROUTE TS.touchTime TO AUX.In
ROUTE AUX.Out TO ACTION.Command
}

MOC_OBJECT {Type "PORTAL" Desc "CONE DE FUGA"}

```

### 8.1.2 Objeto 2 - Globo girante

Este é um objeto em forma de um globo terrestre, que possui um elemento acionador (bolinha vermelha com a palavra “Click” em cima). Quando um usuário clica na bolinha o globo começa a girar, este é um tipo de objeto com ação compartilhada, o efeito de girar é enviado a todos os usuários conectados ao ambiente.

```
#VRML V2.0 utf8
PROTO MOC_OBJECT [
  exposedField SFFloat   Version 1.0
  exposedField SFString  Type ""
  exposedField SFString  Desc ""
  exposedField SFBool    Private FALSE
  eventIn        SFBool   Properties
  eventIn        SFString Command
  eventIn        SFString Command_From_Net
  eventOut       SFString Response
  eventOut       SFString Command_To_Net
]
{
  DEF GEOM Transform {
    translation 0 1.5 0
    children [
      Transform {
        translation 0 -1 2
        children [
          Shape {
            appearance Appearance {
              material Material {diffuseColor 0.9 0.1 0.1}
            }
            geometry Sphere {radius 0.3}
          }
          DEF TS2 TouchSensor {}
          Transform {
            children Shape {
              geometry Text {
                fontStyle FontStyle {
                  family "SANS"
                  style "BOLD"
                  justify "MIDDLE"
                }
                string "Click"
                length 1
              }
            }
            translation 0 0.4 0
          }
        ]
      }
    ]
  }
  DEF EARTH-XFORM Transform {
    children [
```

```

DEF EARTH-TIMER TimeSensor {
    startTime 0
    cycleInterval 5
    loop FALSE
}
DEF EARTH-ENGINE OrientationInterpolator {
    key [ 0, 0.5, 1 ]
    keyValue [ 0 0 1 0,0 1 0 3.14,0 1 0 6.28002]
}
Shape {
    appearance Appearance {
        material Material {diffuseColor 0 0 0.5}
        texture ImageTexture {url "earth-small.jpg"}
    }
    geometry Sphere {radius 1.5}
}
]
}
]
}

DEF ACTION Script {
directOutput TRUE
mustEvaluate TRUE
field SFNode ET USE EARTH-TIMER
field MFString Param []
eventIn SFBool Properties IS Properties
eventIn SFString Command IS Command
eventIn SFString Command_From_Net IS Command_From_Net
eventOut SFString Response IS Response
eventOut SFString Command_To_Net IS Command_To_Net
url "javascript:
function UnPack(value)
{
var P;
var l;
P = 0;
l = 1;
Param = "";
P = value.indexOf('=',0);
if (P===-1)
{
Param[0] = value;
}
else
{
Param[0] = value.substring(0,P);
value = value.substring(P+1,value.length);
while (true)
{
P = value.indexOf('=',0);
if (P===-1)
{
Param[l] = value;
break;
}
Param[l] = value.substring(0,P);
value = value.substring(P+1,value.length);
}
}
}

```

```

        I = I+1;
    }
}
function Properties(value,timestamp)
{
    Response = '%PROP%';
}
function Command(value,timestamp)
{
    UnPack(value);
    if (Param[1]=='Click')
    {
        Command_To_Net = 'Click='+Param[2];
    }
}
function Command_From_Net(value,timestamp)
{
    UnPack(value);
    if (Param[1]=='Click')
    {
        if (Param[2]=='TRUE')
        {
            ET.startTime = timestamp;
        }
    }
}
"
}

DEF AUX Script {
    directOutput TRUE
    eventIn SFTime In
    eventOut SFString Out
    url "javascript:
        function In(value,timestamp)
        {
            if (value)
                Out = 'Local=Click=TRUE'
            else
                Out = 'Local=Click=FALSE';
        }
    "
}

ROUTE TS2.touchTime TO AUX.In
ROUTE AUX.Out TO ACTION.Command
ROUTE EARTH-TIMER.fraction_changed TO EARTH-ENGINE.set_fraction
ROUTE EARTH-ENGINE.value_changed TO EARTH-XFORM.set_rotation
}

MOC_OBJECT {Type "SHARED" Desc "GLOBO GIRANTE"}

```



## 8.2 Apêndice B – Comparativo entre a codificação VRML97 e X3D.

### Código em VRML97:

```
#VRML V2.0 utf8
Transform {
  children [
    NavigationInfo { headlight FALSE }
    DirectionalLight {
      direction 0 0 -1
    }
    Transform {
      translation 3 0 1
      children [
        Shape {
          geometry Sphere { radius 2.3 }
          appearance Appearance {
            material Material { diffuseColor 1 0 0 }
          }
        }
      ]
    }
    Transform {
      translation -2.4 .2 1
      rotation 0 1 1 .9
      children [
        Shape {
          geometry Box {}
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }
          }
        }
      ]
    }
  ]
} # end of children for world
}
```

### Código em X3D:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D SYSTEM "file:///www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd">
<X3D>
  <Scene>
    <Transform
      scaleOrientation="0.0 0.0 1.0 0.0"
      center="0.0 0.0 0.0"
      scale="1.0 1.0 1.0"
      translation="0.0 0.0 0.0"
      rotation="0.0 0.0 1.0 0.0"
      bboxSize="-1.0 -1.0 -1.0"
      bboxCenter="0.0 0.0 0.0">
      <NavigationInfo
        isBound="false"
        bindTime="-1.0"
        headlight="false"
        visibilityLimit="0.0"
        bind="true"
        avatarSize=" 0.25 1.6 0.75"
```

```

    type="&quot;WALK&quot;"
    speed="1.0"/>
<DirectionalLight
  ambientIntensity="0.0"
  on="true"
  direction="0.0 0.0 -1.0"
  color="1.0 1.0 1.0"
  intensity="1.0"/>
<Transform
  scaleOrientation="0.0 0.0 1.0 0.0"
  center="0.0 0.0 0.0"
  scale="1.0 1.0 1.0"
  translation="3.0 0.0 1.0"
  rotation="0.0 0.0 1.0 0.0"
  bboxSize="-1.0 -1.0 -1.0"
  bboxCenter="0.0 0.0 0.0">
  <Shape >
    <Sphere
      radius="2.3"/>
    <Appearance >
      <Material
        specularColor="0.0 0.0 0.0"
        shininess="0.2"
        ambientIntensity="0.2"
        transparency="0.0"
        emissiveColor="0.0 0.0 0.0"
        diffuseColor="1.0 0.0 0.0"/>
      </Appearance>
    </Shape>
  </Transform>
<Transform
  scaleOrientation="0.0 0.0 1.0 0.0"
  center="0.0 0.0 0.0"
  scale="1.0 1.0 1.0"
  translation="-2.4 0.2 1.0"
  rotation="0.0 0.70710677 0.70710677 0.9"
  bboxSize="-1.0 -1.0 -1.0"
  bboxCenter="0.0 0.0 0.0">
  <Shape >
    <Box
      size="2.0 2.0 2.0"/>
    <Appearance >
      <Material
        specularColor="0.0 0.0 0.0"
        shininess="0.2"
        ambientIntensity="0.2"
        transparency="0.0"
        emissiveColor="0.0 0.0 0.0"
        diffuseColor="0.0 0.0 1.0"/>
      </Appearance>
    </Shape>
  </Transform>
</Transform>
</Scene>
</X3D>

```

### 8.3 Apêndice C – Net Interfaces

#### Documentação do suporte a objetos com interfaces sincronizadas.

##### Interface de controle de acesso aos Objetos.

(Deve ser definida uma interface para cada objeto que se deseja controlar via rede)

Proto:

```
EXTERNPROTO NetworkControl [
  field      SFBool      multiuser
  eventIn    SFFloat     control_in
  eventIn    SFFloat     control_fromnet
  eventOut   SFFloat     control_out
  eventOut   SFFloat     control_tonet
] "net.wrl"
```

Uso exclusive do programa (Não devem ser utilizados pelos scripts):

**multiuser** => Indica o modo de operação, não deve ser alterada pelos scripts.

**control\_fromnet** => Usado internamente para enviar dados para o servidor.

**control\_tonet** => Usado internamente para receber dados do servidor.

Eventos utilizados na personalização de scripts:

**control\_out** => Envia comandos de controle para o servidor:

0 => Requisita controle sobre o objeto.

1 => Liberação do controle sobre o objeto.

**control\_out** => Recebe comandos de controle do servidor:

0 => Requisição de controle sobre o objeto aceita.

2 => Objeto foi bloqueado para uso de outro cliente.

3 => Objeto foi liberado para novas requisições.

##### Interfaces de sincronismo de dados de Objetos.

(É definida uma interface para cada propriedade ou evento do Objeto que se deseja sincronizar via rede)

Protos atualmente suportados (outros podem ser adicionados caso seja necessário):

**NetworkSFBool** => Para interface do tipo SFBool (Booleano)

**NetworkSFFloat** => Para interface do tipo SFFloat (Ponto flutuante)

**NetworkSFVec3f** => Para interface do tipo SFVec3f (Vetor de 3 dimensões)

Proto-exemplo (todos os protos desta classe possuem a mesma estrutura mudando somente o tipo de dado):

```
EXTERNPROTO NetworkSFVec3f [
  field      SFBool      multiuser
  eventIn    SFVec3f     set_value
  eventIn    SFVec3f     value_fromnet
  eventOut   SFVec3f     value_changed
  eventOut   SFVec3f     value_tonet
] "net.wrl"
```

Uso exclusivo do programa (Não devem ser utilizados pelos scripts):

**multiuser** => Indica o modo de operação, não deve ser alterada pelos scripts.

**value\_fromnet** => Usado internamente para enviar dados para o servidor.

**value\_tonet** => usado internamente para receber dados do servidor.

Eventos utilizados na personalização de scripts:

**set\_value** => Envia um novo valor de sincronismo para o servidor.

**value\_changed** => Recebe um novo valor de sincronismo do servidor.

Exemplo:

Suponha que se tenha um painel que mostra um vídeo, e é requerido que ele seja controlado por qualquer cliente conectado no ambiente. Este painel deve possuir uma interface de proto do tipo:

```
EXTERNPROTO VIDEO [
  field      SFVec3f    translation
  field      SFRotation rotation
  field      SFBool    enabled
  field      MFString  url
  eventIn    SFFloat  control_in
  eventIn    SFFloat  in
  eventOut   SFFloat  control_out
  eventOut   SFFloat  out
] "video.wrl"
```

onde:

**control\_in** => recebe um comando de controle:

0 => Habilita o painel para controle do cliente local.

2 => Desabilita o controle do painel pelo cliente local.

3 => Indica que o painel esta disponível para controle local.

**control\_out** => envia um comando de controle:

0 => Solicita controle local sobre o painel.

1 => Libera o controle do painel para uso de qualquer cliente.

**in** => recebe sinal de controle do vídeo:

0 => tocar

1 => parar

**out** => envia sinal de controle do vídeo:

0=> tocar

1 => parar

Para controlar este painel via rede de forma sincronizada, deve-se usar estas interfaces de rede :

```
DEF NET_CONTROL_VIDEO NetworkControl {}
DEF NET_VIDEO1 NetworkSFFloat {}
ROUTE VIDEO.control_out TO NET_CONTROL_VIDEO.control_in
ROUTE NET_CONTROL_VIDEO.control_out TO VIDEO.control_in
ROUTE VIDEO.out TO NET_VIDEO1.set_value
ROUTE NET_VIDEO1.value_changed TO VIDEO.in
```

**Observação:** A implementação atual do controle deixa o desenvolvedor de objetos com a responsabilidade de utilizar os sinais de forma adequada e manter o sincronismo da aplicação.

Arquivo de definição dos protos utilizados (net.wrl):

```
#VRML V2.0 utf8
PROTO NetworkSFVec3f [
  field    SFBool  multiuser FALSE
  eventIn  SFVec3f set_value
  eventIn  SFVec3f value_fromnet
  eventOut SFVec3f value_changed
  eventOut SFVec3f value_tonet
]
{
  DEF S Script {
    directOutput TRUE
    mustEvaluate TRUE
    field    SFBool  multiuser IS multiuser
    eventIn  SFVec3f set_value IS set_value
    eventIn  SFVec3f value_fromnet IS value_fromnet
    eventOut SFVec3f value_changed IS value_changed
    eventOut SFVec3f value_tonet IS value_tonet
    url "vrmlscript:
    function set_value(value) {
      if (multiuser == true) value_tonet = value; else value_changed = value;
    }
    function value_fromnet(value) {
      value_changed = value;
    }
    "
  }
}
PROTO NetworkSFBool [
  field    SFBool  multiuser FALSE
  eventIn  SFBool  set_value
  eventIn  SFBool  value_fromnet
  eventOut SFBool  value_changed
  eventOut SFBool  value_tonet
]
{
  DEF S Script {
    directOutput TRUE
    mustEvaluate TRUE
    field    SFBool  multiuser IS multiuser
    eventIn  SFBool  set_value IS set_value
    eventIn  SFBool  value_fromnet IS value_fromnet
    eventOut SFBool  value_changed IS value_changed
    eventOut SFBool  value_tonet IS value_tonet
    url "vrmlscript:
    function set_value(value) {
      if (multiuser == true) value_tonet = value; else value_changed = value;
    }
    function value_fromnet(value) {
      value_changed = value;
    }
    "
  }
}
PROTO NetworkSFFloat [
  field    SFBool  multiuser FALSE
  eventIn  SFFloat set_value
  eventIn  SFFloat value_fromnet
```

```

eventOut  SFFloat  value_changed
eventOut  SFFloat  value_tonet
]
{
  DEF S Script {
    directOutput TRUE
    mustEvaluate TRUE
    field    SFFloat  multiuser IS multiuser
    eventIn  SFFloat  set_value IS set_value
    eventIn  SFFloat  value_fromnet IS value_fromnet
    eventOut SFFloat  value_changed IS value_changed
    eventOut SFFloat  value_tonet IS value_tonet
    url "vrmlscript:
    function set_value(value) {
      if (multiuser == true) value_tonet = value; else value_changed = value;
    }
    function value_fromnet(value) {
      value_changed = value;
    }
    "
  }
}
PROTO NetworkControl [
  field    SFFloat  multiuser FALSE
  eventIn  SFFloat  control_in
  eventIn  SFFloat  control_fromnet
  eventOut SFFloat  control_out
  eventOut SFFloat  control_tonet
]
{
  DEF S Script {
    directOutput TRUE
    mustEvaluate TRUE
    field    SFFloat  multiuser IS multiuser
    eventIn  SFFloat  control_in IS control_in
    eventIn  SFFloat  control_fromnet IS control_fromnet
    eventOut SFFloat  control_out IS control_out
    eventOut SFFloat  control_tonet IS control_tonet
    url "vrmlscript:
    function control_in(value) {
      if (multiuser == false) {
        if (value == 0) control_out = 0;
        if (value == 1) control_out = 3;
      }
      else {
        control_tonet = value;
      }
    }
    function control_fromnet(value) {
      control_out = value;
    }
    "
  }
}
}

```