

**Universidade Federal de São Carlos**  
*Centro de Ciências Exatas e de Tecnologia*  
**Departamento de Computação**  
Programa de Pós-Graduação em Ciência da Computação

**“Projeto de um Modelador 3D Colaborativo  
Baseado no Padrão Emergente MPEG-4 MU”**

**Fernando Vieira Duarte**

São Carlos - SP  
Julho – 2003

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

D812pm

Duarte, Fernando Vieira.

Projeto de um modelador 3D colaborativo baseado no padrão emergente MPEG-4 MU / Fernando Vieira Duarte. -- São Carlos: UFSCar, 2003.

118 p.

Dissertação (Mestrado). -- Universidade Federal de São Carlos, 2003.

1. Realidade virtual. 2. Ambientes virtuais colaborativos. 3. Modelagem 3D. 4. MPEG-4. I. Título.

CDD: 006 (20<sup>a</sup>)

“Todos os homens buscam a felicidade. E não há exceção.

Independentemente dos diversos meios que empregam, o fim é o mesmo.

O que leva um homem a lançar-se à guerra e outros a evitá-la é o mesmo desejo,

embora revestido de visões diferentes.

O desejo só dá o último passo com este fim.

É isto que motiva as ações de todos os homens...”

---

**Blaise Pascal**

**Dedico este trabalho:**

---

**A Deus, Pai presente em todos os momentos de  
nossas vidas.**

**Aos meus pais Alcides e Iracema, pelos esforços  
contínuos para minha educação.**

**A toda minha família que sempre me incentivou.**

## Agradecimentos

À minha orientadora Profa. Regina Borges de Araujo pela orientação, paciência e dedicação dispensada durante o desenvolvimento deste trabalho.

Aos meus grandes amigos Takeda, Will e Marcelo que sempre estiveram comigo nos momentos de alegrias e dificuldades.

A todos os meus colegas de laboratório Goiano, Matheus, Taciana, Richard, Gislaine, alunos de iniciação científica, Raphael Cortez e Raphael Rabelo, e o estagiário Thadeu Junqueira, que sempre apoiaram na realização deste trabalho.

A todos os professores e funcionários do Departamento de Computação da UFSCar.

E mais uma vez a Deus.

# SUMÁRIO

LISTA DE FIGURAS .....	III
LISTA DE TABELAS.....	IV
LISTA DE ABREVIACÕES E SIGLAS .....	V
RESUMO .....	VI
ABSTRACT .....	VII
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 OBJETIVO DO PROJETO .....	3
1.2 JUSTIFICATIVA E RELEVÂNCIA DO PROJETO.....	3
1.3 ESTRUTURA DA DISSERTAÇÃO .....	5
<b>2 AMBIENTES VIRTUAIS COLABORATIVOS .....</b>	<b>6</b>
2.1 COLABORAÇÃO EM AMBIENTES VIRTUAIS .....	7
2.2 REQUISITOS DOS AMBIENTES VIRTUAIS COLABORATIVOS .....	8
2.2.1 Gerenciamento de Percepção da Presença e Ações dos Usuários Participantes .....	9
2.2.2 Manutenção da Consistência.....	10
2.2.3 Gerenciamento de Sessão .....	14
2.2.4 Segurança.....	15
2.2.5 Gerenciamento de Recursos .....	17
2.2.6 Persistência .....	18
2.2.7 Controle de Acesso.....	19
2.2.8 Controle de Concorrência .....	21
2.3.8.1 Controle de Concorrência por Serialização .....	21
2.3.8.2 Controle de Concorrência com Acesso Privilegiado através de Técnica de Tracamento ( <i>Locking</i> ).....	22
2.3.8.3 Controle de Concorrência baseado em Predição .....	23
2.2.9 Comunicação.....	25
2.3 TIPOS DE DADOS TRANSMITIDOS EM AMBIENTES VIRTUAIS COLABORATIVOS .....	25
2.4 AMBIENTES VIRTUAIS COLABORATIVOS EXISTENTES .....	28
2.5.1 DIVE.....	28
2.5.2 NPSNET-IV.....	29
2.5.3 MASSIVE .....	29
2.5.4 SPLINE.....	30
2.5.5 COVEN.....	31
2.5.6 VLNET.....	32
2.5.7 COSMOS .....	32
2.5 MODELAGEM 3D EM AMBIENTES VIRTUAIS COLABORATIVOS .....	33
2.5.1 M3D Editor.....	34
2.5.2 Escultura Virtual Distribuída Colaborativa.....	36
2.5.3 SpinCAS.....	37
2.6 CONSIDERAÇÕES FINAIS.....	38
<b>3 O PADRÃO EMERGENTE MPEG-4 MULTIUSUÁRIO NO SUPORTE AOS AVCS.....</b>	<b>40</b>
3.1 O PADRÃO MPEG-4 .....	40
3.1.1 Arquitetura do Padrão MPEG-4 .....	42
3.1.2 Protocolo BIFS-Command .....	47
3.2 O PADRÃO EMERGENTE MPEG-4 MU .....	51
3.2.1 Componentes da Arquitetura MPEG-4 MU .....	53
3.2.2 Transferência de Pilotagem.....	57
3.2.3 Propagação das mensagens .....	58
3.2.4 MUCommandStream .....	61
3.3 CONSIDERAÇÕES FINAIS.....	64
<b>4 PROJETO E IMPLEMENTAÇÃO DE UM MODELADOR 3D COLABORATIVO.....</b>	<b>67</b>
4.1 DESCRIÇÃO DO MODELADOR 3D COLABORATIVO .....	68

4.2	GERENCIAMENTO DE SESSÃO .....	81
4.3	MANUTENÇÃO DA CONSISTÊNCIA .....	83
4.4	CONTROLE DE CONCORRÊNCIA.....	86
4.5	CONTROLE DE ACESSO.....	88
4.6	CONSCIÊNCIA DO USUÁRIO.....	89
4.7	PERSISTÊNCIA.....	91
4.8	CONSIDERAÇÕES FINAIS.....	91
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>92</b>
5.1	CONTRIBUIÇÕES GERADAS.....	93
5.2	TRABALHOS FUTUROS .....	94
5.3	CONSIDERAÇÕES FINAIS.....	96
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>97</b>
	<b>ANEXO A – INTERFACES DO MODELADOR 3D COLABORATIVO.....</b>	<b>102</b>
	<b>ANEXO B – IMPLEMENTAÇÃO DOS COMPONENTES MSC E MBK.....</b>	<b>110</b>
	<b>ANEXO C – PUBLICAÇÃO DE ARTIGOS .....</b>	<b>115</b>

\_Toc59246212

## Lista de Figuras

FIGURA 1 – (A) AVC PARA DESENVOLVIMENTO DE PROTÓTIPOS [PAND97].	7
FIGURA 2 – AMBIENTE VIRTUAL COLABORATIVO DE CONFERÊNCIA [DIVE].	28
FIGURA 3 – AMBIENTE DE TREINAMENTO VIRTUAL [NPSNET].	29
FIGURA 4 – IMAGENS DE TV HABITADA CRIADAS USANDO MASSIVE-2 [BENF01].	30
FIGURA 5 – DIAMOND PARK [SPLINE].	31
FIGURA 6 – AMBIENTES VIRTUAIS DE CONFERÊNCIA [COVEN].	31
FIGURA 7 – AVATARES EM VLNET [VLNET].	32
FIGURA 8 – AMBIENTE DE MONTAGEM COLABORATIVA DE UMA BICICLETA [COSMOS].	33
FIGURA 9 – M3D EDITOR [CRES00].	35
FIGURA 10 - ESCULTURA VIRTUAL DISTRIBUÍDA COLABORATIVA [LI01].	37
FIGURA 11 - SPINCAS [GRISO02].	38
FIGURA 12 - ARQUITETURA DO TERMINAL MPEG-4 [EXTRAÍDA DE N4264].	43
FIGURA 13 - TIPOS DE BIFS-COMMANDS [EXTRAÍDA DE N2201].	50
FIGURA 14 – MECANISMO DE SINCRONIZAÇÃO ENTRE PILOT E DRONE [ADAPTADA DE N4415].	52
FIGURA 15 - ARQUITETURA DE REQUISIÇÃO/ATUALIZAÇÃO [ADAPTADA DE N4415].	54
FIGURA 16 – PILOT NO SERVIDOR [ADAPTADA DE N4415].	59
FIGURA 17 - PILOT NO TERMINAL-CLIENTE 1 [ADAPTADA DE N4415].	60
FIGURA 18 - PILOT NO TERMINAL-CLIENTE 2 [ADAPTADA DE N4415].	61
FIGURA 19 - HIERARQUIA DE MENSAGENS [W4852].	62
FIGURA 20 - MODELADOR 3D COLABORATIVO.	69
FIGURA 21 - MENU WORLD.	69
FIGURA 22 - MENU EDIT.	72
FIGURA 23 - MENU VIEW.	73
FIGURA 24 - MENU CREATE.	74
FIGURA 25 - MENU MODIFIERS.	75
FIGURA 26 - MENU TOOLS.	76
FIGURA 27 - MENU PROJECT.	77
FIGURA 28 - MENU SESSION.	78
FIGURA 29 - MENU ZONE.	79
FIGURA 30 – DIAGRAMA DA ENTRADA DE USUÁRIOS NA SESSÃO COLABORATIVA.	82
FIGURA 31 - DIAGRAMA DE SAÍDA DE USUÁRIO DA SESSÃO.	83
FIGURA 32 - DIAGRAMA DA MODIFICAÇÃO DA CENA GRÁFICA.	84
FIGURA 33 - MECANISMO DE ATUALIZAÇÃO DA CENA.	85
FIGURA 34 - DIAGRAMA DA TRANSFERÊNCIA DE PILOTAGEM.	87
FIGURA 35 - ARQUITETURA DE COMUNICAÇÃO ENTRE O MODELADOR 3D COLABORATIVO E OS COMPONENTES MSC E MBK.	88
FIGURA 36 - INTERFACE DO COMPONENTE MSC.	90
FIGURA 37 - INTERFACE PRINCIPAL DO MODELADOR 3D COLABORATIVO COM UM UNIVERSO 3D.	102
FIGURA 38 - PROPRIEDADES DO MUNDO VIRTUAL.	103
FIGURA 39 - CRIAÇÃO DE UM PROJETO.	103
FIGURA 40 - CRIAÇÃO DE UMA SESSÃO.	104
FIGURA 41 - ATRIBUIÇÃO DE PAPÉIS SOCIAIS AOS PARTICIPANTES DE UMA SESSÃO.	104
FIGURA 42 - GERENCIAMENTO DE GRUPOS.	105
FIGURA 43 - CRIAÇÃO DE GRUPOS.	105
FIGURA 44 - PROPRIEDADES DE GRUPO.	106
FIGURA 45 - GERENCIAMENTO DE USUÁRIOS.	106
FIGURA 46 - CRIAÇÃO DE USUÁRIOS.	107
FIGURA 47 - PROPRIEDADES DE USUÁRIO.	107
FIGURA 48 - GERENCIAMENTO DE PAPÉIS SOCIAIS.	108
FIGURA 49 - CRIAÇÃO DE PAPÉIS SOCIAIS.	108
FIGURA 50 - PROPRIEDADES DE PAPEL SOCIAL.	109
FIGURA 51 - ENTRADA E SAÍDA DE ZONAS COMPARTILHADAS.	109

## Lista de Tabelas

TABELA 1. QUESTÕES FUNDAMENTAIS DE SEGURANÇA. ....	16
TABELA 2 – NÍVEIS DE OTIMISMO DA POLÍTICA DE TRANCAMENTO. ....	23
TABELA 3 - EXEMPLO DE ATRIBUIÇÃO DE DIREITOS DE ACESSO AOS PAPÉIS SOCIAIS DOS USUÁRIOS. ....	89

## Lista de Abreviações e Siglas

<b>2D</b>	<i>Bidimensional</i>
<b>3D</b>	<i>Tridimensional</i>
<b>AFX</b>	<i>Animation Framework eXtension</i>
<b>API</b>	<i>Application Programming Interface – Interface de Aplicação de Programação</i>
<b>AVC</b>	<i>Ambientes Virtuais Colaborativos</i>
<b>BIFS</b>	<i>Binary Format for Scene – Formato Binário de Descrição da Cena</i>
<b>CAD</b>	<i>Computer Aided Design – Projeto Auxiliado por Computador</i>
<b>CSCW</b>	<i>Computer-Supported Cooperative Work – Trabalho Cooperativo Assistido por Computador</i>
<b>DAI</b>	<i>DMIF Application Interface – Interface de Aplicação DMIF</i>
<b>DMIF</b>	<i>Delivery Multimedia Integration Framework</i>
<b>DPI</b>	<i>DMIF Plugin Interface</i>
<b>EAI</b>	<i>External Authoring Interface – Interface de Autoria Externa</i>
<b>ES</b>	<i>Elementary Stream – Fluxos Elementares</i>
<b>ISO</b>	<i>International Standard Organization – Organização Internacional para Padronização</i>
<b>MAI</b>	<i>Multiuser Application Interface – Interface de Aplicação Multiusuário</i>
<b>MBK</b>	<i>Mutech Bookkeeper</i>
<b>MCM</b>	<i>MuChannel Manager</i>
<b>MMH</b>	<i>Mutech Message Handler</i>
<b>MPEG</b>	<i>Moving Picture Experts Group</i>
<b>MSC</b>	<i>Mutech Session Controller</i>
<b>MU/MUTech</b>	<i>MultiUser – MultiUsuário</i>
<b>NURBS</b>	<i>Non-Uniform Rational B-Splines</i>
<b>OCI</b>	<i>Object Content Information – Informação de Conteúdo do Objeto</i>
<b>PDA</b>	<i>Personal Digital Assitant – Assistente Digital Pessoal</i>
<b>QoS</b>	<i>Quality of Service – Qualidade de Serviço</i>
<b>SDM</b>	<i>Systems Decoder Model – Modelo de Decodificação de Sistema</i>
<b>SL</b>	<i>Synchronization Layer – Camada de Sincronização</i>
<b>VRML</b>	<i>Virtual Reality Modeling Language – Linguagem de Modelagem de Realidade Virtual</i>

## Resumo

Nos últimos anos temos experimentado avanços nas tecnologias de rede como também na computação gráfica e tecnologia de exibição. Como uma consequência, muitos Ambientes Virtuais Colaborativos emergiram graças às capacidades crescentes da computação dos computadores de mesa (*desktop*), como também o crescimento enorme na largura de banda da rede e a onipresença da Internet. Ambientes virtuais colaborativos para modelagem 3D podem ser caracterizados pela interação entre múltiplos usuários para a modificação e/ou criação de objetos 3D compartilhados. Por exemplo, estes ambientes podem ser usados na modelagem e visualização de protótipos virtuais para reduzir custos no processo de projeto de produtos. Os desafios da construção de ambientes de modelagem 3D colaborativos estão relacionados principalmente com a renderização em tempo-real dos objetos modificados, como também para a interação de usuários com o ambiente virtual e a manutenção da consistência do ambiente virtual compartilhado. São encontrados na literatura somente alguns ambientes de modelagem 3D colaborativos e normalmente os projetos resultam em soluções não padronizadas e complexas para a colaboração entre os usuários. Este trabalho apresenta a implementação de um ambiente colaborativo para modelagem 3D baseado no padrão emergente MPEG-4 MU (*MultiUser*). Com este ambiente, cenas gráficas 3D podem ser criadas em tempo-real, por participantes múltiplos, em sessões colaborativas síncronas. Estas cenas podem ser visualizadas em qualquer terminal MPEG-4, inclusive celulares e PDAs (*Personal Digital Assistants*). Controle de Sessão, manutenção de consistência, controle da concorrência e trancamento (*locking*) dos objetos 3D são realizados pelos componentes MSC (*MUTech Session Controller*) e MBK (*MUTech Bookkeeper*), pelo mecanismo de *Pilot/Drone* e o protocolo de *BIFS-Command*. Todos estes componentes são definidos pelo padrão emergente MPEG-4 MU e implementados pelo Laboratório de Realidade Virtual em Rede (LRVNet) da Universidade Federal de São Carlos.

## Abstract

In the last years we have been experiencing advances in networking technologies as well as computer graphics and display technology. As a consequence, a lot of Collaborative Virtual Environments have emerged thanks to the increasing computation capabilities of desktop computers as well as the enormous growth in network bandwidth and the ubiquity of the Internet. Collaborative virtual environments for 3D modeling can be characterized by the interaction among multiple users for the creation and/or modification of shared 3D objects. These environments can be used, for instance, in the modeling and visualization of virtual prototypes in order to reduce costs in the process of products design. The challenges of building collaborative 3D modeling environments are mainly related to real-time rendering of the modified objects, as well as to users' interaction with the virtual environment, and to consistency maintenance of the shared virtual environment. Only a few collaborative 3D modeling environments are found in the literature and usually their projects result in complex non standardized solutions for the collaboration among users. This work presents the implementation of a collaborative environment for 3D modeling based on the emerging MPEG-4MU (multi-user) standard. With this environment, 3D graphic scenes can be created in real-time, by multiple participants, in synchronous collaborative sessions. These scenes can be visualized in any MPEG-4 terminal, including cellulators and personal digital assistants (PDAs). Sessions control, consistency maintenance, concurrency control and 3D object locking are realized by the MSC (MUTech Session Controller) and MBK (MUTech Bookkeeper) components, through the Pilot/drone mechanism and the BIFS-Command protocol. All these components were defined by the MPEG-4 MU emerging standard and implemented by the Networked Virtual Reality Lab (LRVNet) at Federal University of São Carlos.

# 1 Introdução

A Realidade Virtual pode ser definida como uma interface natural e poderosa de interação homem-máquina, por permitir ao usuário navegação, interação e imersão em um ambiente virtual gerado por computador [ARAU96] [BURD94].

Os Ambientes Virtuais Colaborativos – AVCs (*Collaborative Virtual Environments*), que podem ser vistos como uma convergência de interesses das áreas de Realidade Virtual e de Trabalho Cooperativo Assistido por Computador (*Computer-Supported Cooperative Work – CSCW*), são caracterizados por utilizar de espaços sintéticos tridimensionais que são compartilhados entre múltiplos usuários, remotamente localizados, para suportar trabalho colaborativo, projeto colaborativo, visualização, simulação, treinamento e educação, como também entretenimento [BENF01] [OLIV99]. Questões importantes no desenvolvimento dos AVCs incluem gerenciamento de sessão, percepção de usuários ou de grupo de usuários, manutenção da consistência do ambiente compartilhado, segurança, gerenciamento de recursos, persistência, comunicação, controle de acesso e controle de concorrência.

O uso de AVCs para modelagem colaborativa é caracterizado pela interação entre múltiplos usuários na criação e/ou modificação da forma geométrica de objetos 3D compartilhados, e possuem várias aplicações, entre elas a modelagem e visualização de

protótipos virtuais, que reduz os custos no processo de projeto de produtos [KAN01]. Os desafios na modelagem colaborativa estão relacionados principalmente com a renderização em tempo-real dos objetos modificados, com a interação entre os usuários participantes e o ambiente virtual, e a manutenção da consistência do ambiente virtual compartilhado.

O Padrão MPEG-4, solução padronizada da ISO/IEC para a codificação e entrega de diferentes formatos de mídia sobre uma ampla variedade de redes e plataformas computacionais, apresenta vantagens em relação a outras tecnologias 3D existentes, tais como o X3D, Java 3D e VRML. Algumas das principais vantagens do padrão MPEG-4 incluem o formato de entrega de dados por meio de Fluxos Elementares (*Elementar Streams* - ES), que possibilita o carregamento da cena na forma de *streaming* ou encadeamento, o formato binário de descrição da cena - BIFS (*Binary Format Scene*), que codifica a cena de forma compacta e eficiente, o que torna cenas em formato BIFS mais rápidas de carregar e de transportar pela rede [ABRE01] [HOSS01], e por ser um padrão já conhecido do Laboratório de RV em Rede do DC/UFSCar onde este trabalho de mestrado foi desenvolvido.

Uma extensão do padrão MPEG-4 para o suporte a ambientes 3D virtuais colaborativos é o padrão emergente MPEG-4 multiusuário, que define e especifica os componentes de uma arquitetura de suporte multiusuário. Dois dos principais componentes desta arquitetura, responsáveis pelo controle de sessão e sincronização de cenas, MSC (*MUTech Session Controller*) e MBK (*MUTech Bookkeeper Controller*), respectivamente, foram implementados como parte deste trabalho. Estes componentes dão suporte ao gerenciamento de sessão, percepção de usuários ou de grupo de usuários, manutenção da consistência do ambiente compartilhado, persistência, comunicação, controle de acesso e controle de concorrência. Estes componentes foram integrados ao código do modelador 3D de tal forma que sessões colaborativas podem ser estabelecidas, mantidas e encerradas. Conceitos definidos pelo padrão emergente MPEG-4 MU, como o da pilotagem de objetos

compartilhados em zonas de sessão, foram estendidos e implementados, de tal forma que objetos de uma zona podem ser compartilhados e alterados por múltiplos usuários, através de um controle da "pilotagem" do objeto. Um mecanismo de trancamento, que permite o controle da concorrência, foi implementado como parte do modelador colaborativo. O modelador foi implementado na linguagem Java e usa a biblioteca Java 3D para renderização das cenas gráficas.

## **1.1 Objetivo do Projeto**

De acordo com o cenário apresentado anteriormente, o objetivo principal deste trabalho é a implementação de um ambiente colaborativo para modelagem 3D baseado no Padrão Emergente MPEG-4 MU. O modelador 3D deve suportar a interação de múltiplos usuários na criação ou modificação da cena gráfica compartilhada em sessões colaborativas síncronas.

Este trabalho tem também como objetivo de analisar o potencial do padrão MPEG-4 MU para a construção de ambientes virtuais colaborativos que suportem a modelagem 3D em tempo-real, utilizando os componentes especificados pelo padrão e desenvolvidos pelo *LRVNet*.

## **1.2 Justificativa e Relevância do Projeto**

Apesar do grande número de projetos de pesquisas nesta área, poucos destes suportam a modelagem colaborativa devido a complexidade que envolve a construção de um AVC. Num AVC, as seguintes funcionalidades são importantes: gerenciamento de sessão, percepção dos usuários, manutenção da consistência do ambiente virtual, controle da concorrência e controle de acesso devido às interações dos múltiplos usuários, persistência, segurança, gerenciamento de recursos e comunicação. Mais ainda os sistemas encontrados na

literatura, apresentam, normalmente, soluções proprietárias o que dificulta a interoperabilidade e a disponibilização do sistema numa escala maior em termos, por exemplo, de dispositivos com capacidades diferentes. Este padrão foi escolhido por pelo menos três razões:

1. O padrão MPEG-4, assim como os outros padrões do MPEG (MPEG-1 para criação de vídeo interativo em CD-ROM e camada 3 para música - MP3, MPEG-2 para DVD e TV digital, MPEG-7 para descrição de conteúdos) são padrões que foram amplamente adotados por fabricantes e usuários, principalmente para a codificação e representação de áudio e vídeo;
2. O MPEG-4 já desponta como padrão de sucesso na codificação e entrega de multimídia para dispositivos que vão desde o celular até *set-top-boxes*, PCs e servidores de alta capacidade de processamento e de rede;
3. O Laboratório de RV em rede do DC da UFSCar, onde este trabalho foi realizado, tem trabalhado com o suporte multiusuário no padrão MPEG-4, tendo inclusive enviado uma proposta *softwired* como solução de suporte à colaboração, e que teve pelo menos duas de suas idéias incorporadas no padrão emergente MPEG-4 MU [M6692].

Desta forma, este trabalho explora a implementação um modelador 3D em que, cenas gráficas, em formato MPEG-4, podem ser geradas em tempo-real, por múltiplos participantes em sessões colaborativas.

### 1.3 Estrutura da Dissertação

Este trabalho está organizado da seguinte forma: o capítulo 2 caracteriza os ambientes virtuais colaborativos, sua utilização, seus principais requisitos e apresenta exemplos de ambientes virtuais colaborativos existentes, bem como alguns exemplos de AVCs que suportam a modelagem colaborativa. O capítulo 3 descreve o padrão MPEG-4 desenvolvido para a codificação e entrega de diferentes formatos de mídias em uma ampla variedade de redes e plataformas computacionais, detalhando a parte 1 (Sistema) e o Protocolo *BIFS-Command*, amplamente utilizado na implementação deste trabalho. O capítulo 3 também apresenta a especificação do padrão MPEG-4 MU no suporte a múltiplos usuários, detalhando os principais componentes da arquitetura de suporte que foram implementadas como parte deste trabalho. A descrição do modelador 3D e a apresentação da implementação de algumas de suas funções básicas, que incluem: gerenciamento de sessão e de zonas, controle de concorrência e manutenção da consistência, são apresentadas no capítulo 4, seguidas de contribuições geradas, trabalhos futuros, conclusões e referências bibliográficas.

## 2 Ambientes Virtuais Colaborativos

Os Ambientes Virtuais Colaborativos – AVCs (*Collaborative Virtual Environments*) podem ser vistos como o resultado de uma convergência de interesses de pesquisa dentro da Realidade Virtual (RV) e do Trabalho Cooperativo Assistido por Computador (*Computer-Supported Cooperative Work – CSCW*) e são caracterizados por utilizar de espaços sintéticos tridimensionais que são compartilhados entre múltiplos usuários, remotamente localizados, para suportar trabalho colaborativo, projeto colaborativo, visualização, simulação, treinamento e educação, como também entretenimento [BENF01] [BRAS00].

Os AVCs representam uma extensão natural da tecnologia de Realidade Virtual de suporte monousuário para RV de suporte multiusuário, em que múltiplos participantes comunicam-se entre si, partilhando uma visão comum de um mundo virtual tridimensional. Os AVCs podem ser utilizados em diversas aplicações, tais como: educação, treinamento, visualização compartilhada (científica e/ou de aplicações de projeto industrial), construção e validação digital de protótipos, entretenimento etc.

Na indústria os AVCs podem ser uma solução atraente por reduzir despesas de treinamento e desenvolvimento de novos produtos, pois em vez de trabalhar com objetos físicos, estes são representados por objetos virtuais que podem ser tratados em um ambiente virtual acessível a múltiplos usuários. Os usuários, representados (ou não) por avatares, podem manipular e interagir com os objetos como no mundo real, experimentando e treinando antes de usar e/ou produzir um equipamento real [OLIV00] [CONS01].

A Figura 1(a) mostra a aplicação de colaboração em ambientes virtuais, no projeto de desenvolvimento de protótipos, em que os engenheiros participantes estão localizados em diferentes lugares. A Figura 1(b) ilustra o uso de ambiente virtual colaborativo para treinamento militar de combate e controle de atividades.



Figura 1 – (a) AVC para desenvolvimento de protótipos [PAND97] (b) AVC de Treinamento [SENSE8].

## 2.1 Colaboração em Ambientes Virtuais

A colaboração em ambientes virtuais ocorre com a introdução de objetos no mundo virtual 3D, em tempo-real, e o uso do ambiente virtual como uma interface gráfica 3D por múltiplos usuários, para a realização de uma tarefa específica, em que os usuários

observam e respondem às interações ocorridas no ambiente virtual. Os eventos que alteram o AVC são transmitidos para todos os usuários, deste modo, o AVC pode ser atualizado e mantido em um estado consistente, dando a sensação de que os usuários estão em um único AVC [PAND97].

A colaboração em ambientes virtuais é normalmente interativa, porém existem também formas de colaboração não-interativa, sem a comunicação direta entre os usuários. A colaboração interativa envolve diferentes tipos de atividades, correspondentes a diferentes contextos de tarefas e graus de colaboração. Um grau mínimo de colaboração interativa é a troca de informações. Já um grau máximo de colaboração interativa é a noção de colaboração baseada em tarefas, onde os usuários compartilham uma meta comum e colaboram para a realização desta meta. A colaboração em ambientes virtuais envolve diferentes atividades paralelas que podem gerar conflitos, requerendo uma coordenação das atividades com alguma forma de organização e meios de comunicação [KUIJ97].

O enfoque dominante da colaboração em AVCs citado na literatura assume que cada participante vê o mesmo conteúdo, embora de uma perspectiva diferente. Porém, a experiência da comunidade CSCW no projeto de interfaces gráficas 2D colaborativas, baseadas no princípio de “*What You See Is What I See*” (WYSIWIS), sugere que este enfoque pode impedir a habilidade das pessoas na colaboração de fato. Assim, novos princípios de compartilhamento, necessidade de interfaces e visões diferentes dos dados compartilhados foram avaliados para refletir os diferentes interesses e papéis que usuários desempenham no mundo compartilhado [BENF01].

## **2.2 Requisitos dos Ambientes Virtuais Colaborativos**

O objetivo principal no desenvolvimento de AVCs é a construção de um ambiente seguro, persistente e integrado que ofereça a um grupo de usuários dispersos

geograficamente, as condições necessárias para comunicação, interação e colaboração para que estes possam atingir um objetivo comum. Questões chave no projeto desses ambientes incluem: percepção da presença e das ações dos usuários, manutenção da consistência do ambiente compartilhado (sincronização dos dados compartilhados), gerenciamento de sessão, segurança, gerenciamento de recursos utilizados para a colaboração, persistência, isto é, armazenamento e recuperação do ambiente virtual, comunicação, controle de acesso e controle de concorrência. Estas questões são descritas a seguir.

### **2.2.1 Gerenciamento de Percepção da Presença e Ações dos Usuários Participantes**

Os usuários participantes de um ambiente virtual compartilhado devem receber informações sobre a percepção da presença e das ações dos outros usuários participantes do ambiente, tais como entrada/saída de usuários no/do ambiente e as alterações que cada um promove no AVC. Assim, a função principal do gerenciamento de percepção é determinar o que deve ser replicado, recebido, renderizado, etc [GREE00].

Existem diferentes formas ou níveis de percepção, uma vez que as informações que são transmitidas sobre percepção podem ser selecionadas sem que haja invasão de privacidade dos usuários emissores ou criação de perturbação nos receptores. Algumas formas de percepção relativas à privacidade e acessibilidade podem ser configuradas pelos participantes, para determinar o recebimento ou não de tais informações [PINT01].

O gerenciamento da percepção é um assunto muito importante em um AVC, pois é usado como um mecanismo para regular a quantidade de informação que cada usuário tem que processar. O gerenciamento da percepção suporta a colaboração entre usuários, pois supre toda informação de consciência, sobre os usuários e objetos, que não é importante para a tarefa colaborativa atual dos usuários. O gerenciamento de percepção também tem um papel

importante na escalabilidade deste tipo de sistema, pois a limitação na quantidade de informação que deve ser processada por cada usuário, pode ser um mecanismo muito efetivo para reduzir uso de recursos do sistema, como largura de banda de rede e poder de processamento do computador. Diferentes políticas de percepção podem ser utilizadas, dependendo dos requisitos e objetivos do sistema [ANTU01].

Muitos sistemas de AVC atuais não provêm informação de percepção sobre outros participantes de uma sessão colaborativa. Quando as pessoas trabalham juntas em uma interação face a face, várias informações podem ser percebidas, que ajudam a compreender o que os outros estão fazendo. Esta percepção em um AVC tem que ser obtida a partir de informações geradas pela interação dos outros participantes com o espaço compartilhado.

Um nível simples de percepção envolve o conhecimento de quem está presente, onde está trabalhando, e o que está fazendo no ambiente virtual. Esta percepção geralmente é visual com a utilização de avatares realísticos (ou não) ou ponteiros 3D, que identificam os usuários presentes e as suas ações no ambiente virtual [GRISO02] [LI01].

A informação de percepção é usada na colaboração para coordenar as atividades, simplificar a comunicação verbal, prover ajuda apropriada, e administrar o que é gerado como trabalho individual e como trabalho compartilhado [GUTW99].

### **2.2.2 Manutenção da Consistência**

Os eventos que alteram o AVC, tais como ações dos usuários e atualizações de estado dos objetos compartilhados do ambiente, devem ser transmitidos para todos os usuários, deste modo, o AVC pode ser atualizado mantendo um estado consistente, e dando a sensação de que os usuários estão em um único AVC.

A manutenção da percepção em um espaço compartilhado é dividida em dois níveis: o nível do objeto e o nível espacial. No nível do objeto, a manutenção da percepção busca garantir que as mudanças de estado do objeto sejam propagadas para todos os usuários do espaço compartilhado, de acordo com certos critérios de consistência. Os critérios de consistência devem ser garantidos através de protocolos de consistência. No nível espacial, a manutenção de percepção procura proporcionar para os usuários uma visão consistente do espaço de interação. Isto significa que todos os usuários estão cientes do mesmo conjunto de objetos compartilhados, ou que o conjunto de objetos ou réplicas que cada usuário está ciente, obedece a alguns requisitos para replicação. Esta manutenção de percepção é suportada por uma política de replicação que administra a criação e destruição de réplicas no espaço compartilhado dos usuários - como os objetos compartilhados são inseridos ou removidos e a entrada e saída dos usuários do espaço comum. A associação de usuários no espaço compartilhado e a criação e destruição de réplicas são propagadas para os usuários do espaço compartilhado por um protocolo de consistência espacial [ANTU01a].

Para que haja a manutenção da percepção, alguns requisitos devem ser cumpridos:

- Tempo de resposta: o tempo de resposta para as ações de usuários locais deve ser curto, mesmo que os usuários em colaboração estejam em máquinas diferentes, conectadas pela Internet com alta latência de comunicação. O tempo de resposta é importante porque muitos usuários baseiam, subconscientemente, sua percepção da qualidade de serviço do sistema mais no tempo de resposta do que na funcionalidade do sistema, e o tempo de resposta alto diminui a efetividade do sistema para suportar trabalho colaborativo. A latência da rede de comunicação, que depende da

largura da banda, representa o desafio principal para atingir-se um tempo de resposta aceitável para muitos sistemas colaborativos na Internet, menor que 100ms [HOSS01].

- Alto grau de concorrência: múltiplos usuários podem editar qualquer parte do objeto compartilhado simultaneamente a qualquer momento, para facilitar o fluxo livre e natural de informações entre usuários em colaboração. O desafio principal é a administração dos fluxos múltiplos de atividades simultâneas geradas pelos usuários participantes, de forma que seja mantida a consistência de sistema.

Segundo Sun & Chen [SUN02] para satisfazer a exigência de tempo de resposta baixo para os AVCs no ambiente da Internet, o único modo é adotar uma arquitetura replicada para o armazenamento de objetos compartilhados: os objetos compartilhados são replicados no armazenamento local de cada usuário participante, de modo que as operações de edição podem ser executados nos terminais locais imediatamente e então propagadas para os terminais remotos. O suporte a edição simultânea de objetos compartilhados em uma arquitetura replicada pode ocasionar vários problemas de inconsistência, a saber:

- Divergência – as operações podem chegar e executar em terminais locais diferentes, em diferentes ordens de chegada, resultando em estados de objetos finais divergentes em terminais locais diferentes;
- Violação de causalidade – as operações podem chegar e executar fora de sua ordem de causa-efeito natural, causando confusão ao sistema e ao usuário;

- Violação de intenção – o efeito da execução atual de uma operação pode ser diferente da intenção desta operação.

A intenção de uma operação está definida como o efeito de execução que pode ser alcançado aplicando esta operação ao estado do objeto sobre o qual a operação foi gerada. O estado compartilhado de um objeto está explicitamente definido como um subconjunto de seus atributos e ações (métodos). A replicação é um mecanismo usado para compartilhar a informação, isto é, os objetos de um particular espaço compartilhado. Um espaço compartilhado é uma entidade abstrata que representa um conjunto de objetos compartilhados. Vários espaços compartilhados podem co-existir, mas cada objeto é compartilhado em só um espaço. É necessário se tornar um usuário (participante) de um espaço compartilhado para poder acessar, inserir ou remover seus objetos. Os objetos inseridos por um usuário são chamados de réplicas locais e para cada réplica inserida são criadas outras réplicas nos outros usuários [ANTU01a].

Na definição de uma solução para compartilhamento de informação em sistemas de AVCs devemos identificar qual a informação que deve ser compartilhada e o que deve ser controlado pelo desenvolvedor em relação a informação compartilhada. Por exemplo, deve ser definido se todos os objetos de uma aplicação devem ser compartilhados, ou se todos os atributos ou comportamentos devem ser compartilhados. Os mecanismos para manter a consistência da informação compartilhada é outra questão importante, e devem ser definidos os critérios de consistência para garantir a manutenção de uma percepção consistente da informação compartilhada entre todos os usuários. Por exemplo, definir se é necessário que todos os usuários vejam as mudanças de estado de objetos na mesma ordem, ou se os mesmos critérios de consistência devem ser usados para todos os objetos, ou ainda, se podem ser definidos critérios diferentes para cada objeto.

### 2.2.3 Gerenciamento de Sessão

O gerenciamento de sessão deve prover para os usuários suporte para a entrada/saída de usuários no/do ambiente virtual, além de gerenciamento dos participantes, e definição de regras específicas para serem aplicadas na sessão.

Kausar & Crowcroft [KAUS99] dividiram os serviços de gerenciamento de sessão em grupos funcionais diferentes, que são: configuração da sessão, gerenciamento de participação, controle da ordem de participação (*“floor control”*) e segurança.

- **Configuração da sessão.** A configuração da sessão refere-se essencialmente a definição de um perfil da sessão. Os meios para especificar e cumprir as políticas de sessão também são providos. Um perfil da sessão pode definir as permissões dos participantes, os papéis sociais disponíveis (por exemplo, gerente, técnico, relator ou observador) e as permissões associadas. A atribuição de papéis segue a seguinte ordem: indicação inicial ou solicitação do papel, seguido de concessão, negação ou compartilhamento do mesmo.
- **Gerenciamento de participação.** O gerenciamento de participação inclui serviços para estabelecimento de sessões, convite ao grupo ou indivíduo, e terminação, além de funções para estabelecimento de conexão dos membros de uma sessão. Os participantes podem inscrever-se ou sair de uma sessão. Podem ainda ser convidados ou excluídos de uma sessão. As mudanças de uma sessão para outra também são incluídas como parte das funcionalidades.

- **Controle da ordem de participação (vez).** O controle da vez, em CSCW, é uma metáfora para “atribuir a vez a um orador” que é aplicável para qualquer recurso compartilhado dentro da sessão e ambientes de colaboração. A vez é uma permissão de acesso temporário individual ou uma permissão de manipulação de um recurso compartilhado específico, como, por exemplo, um canal de voz, que permite acesso, simultâneo e livre de conflitos aos recursos, para vários participantes.
- **Segurança.** As funções de segurança consistem da conjunção das opções para controle de sessão e parte do gerenciamento de participação, como também configuração de sessão. A autenticação é executada quando um participante entra na sessão e pode ser repetida arbitrariamente durante o curso da sessão.

Os serviços essenciais do gerenciamento de sessão, comuns a qualquer sessão, são: entrada, convite, saída, controle da vez e controle de erros básicos. Estes serviços provêm as funcionalidades para controlar a sessão e refletem no comportamento dos participantes.

#### **2.2.4 Segurança**

Com a proliferação dos AVCs, muitos destes serão usados em contextos que podem incentivar usuários mal intencionados a fazerem uso indevido de tais sistemas para ganho próprio. Por exemplo, fabricantes podem usar AVCs para prototipagem virtual, o qual permite que engenheiros, em várias localidades, possam tomar decisões de projeto em um ambiente colaborativo, em que todos estão presentes virtualmente e observam informações sensíveis e proprietárias. Neste caso, um competidor que se ocupa de espionagem industrial,

poderia tentar obter a informação proprietária e usá-la para vantagem própria dentro do mercado, ou modificar ou destruir a informação para enganar ou prejudicar os seus competidores [SALL02].

Os AVCs variam no grau de segurança que eles requerem, e isto é definido principalmente pela organização que usa o AVC e o contexto dentro do qual o AVC será usado, devendo sempre ser feita ponderações entre o custo de desenvolvimento e manutenção do sistema, o desempenho do sistema, e os riscos associados a uma violação da política de segurança ou de confiança. Alguns AVCs necessitam de níveis altos de segurança, executando em sistemas confiáveis com políticas de controle de acesso obrigatórias, enquanto outros requerem níveis mínimos ou médios de controle de segurança porque os riscos associados com o uso impróprio destes sistemas é considerado baixo.

As principais questões de segurança – confiança (privacidade), integridade, disponibilidade, não-repúdio, e autenticação –, são todas importantes ao longo do ciclo de vida de um AVC e são mostradas na Tabela 1. É essencial que uma política de segurança seja traçada para o AVC e que esta seja cumprida rigidamente pelo gerente, na configuração do AVC, e pelos usuários no uso do AVC.

**Tabela 1. Questões fundamentais de segurança.**

Áreas	Descrição
Integridade	Previne a modificação dos dados sem autorização
Privacidade	Previne a visualização dos dados sem autorização
Disponibilidade	Assegura que os dados estejam disponíveis para seu uso planejado
Não-repúdio	Assegura que um usuário não pode impedir que a informação que foi colocada no sistema seja acessada por todos os participantes
Autenticação	Assegura que uma entidade (usuário, software) é o que diz ser

No estabelecimento de políticas de segurança e na implementação de mecanismos de cumprimento dessas políticas para um AVC, deve ser dada atenção ao

impacto nos requisitos de QoS (Qualidade de Serviço). Por exemplo, qualquer forma de encriptação pode afetar a latência na transmissão de pacotes de dados. A criptografia de chaves-públicas usada para atualizações de estado, por exemplo, causaria processamento computacional mais intensivo que o convencional, criptografia de chave-única.



### 2.2.5 Gerenciamento de Recursos

O gerenciamento de recursos é responsável pela distribuição dos recursos compartilhados para um grupo de usuários ou usuários individuais, para que possam realizar as suas tarefas, por exemplo, quando dois ou mais usuários querem alterar simultaneamente um documento compartilhado dentro do AVC. Três questões básicas de gerenciamento de recursos são definidas em Raposo [RAPO01] e descritas abaixo:

- **Compartilhamento:** um número limitado de recursos deve ser compartilhado pelos usuários realizando uma tarefa (como a edição de documento compartilhado por múltiplos usuários). Esta dependência inclui noções de divisibilidade (como o recurso é dividido em pequenas partes sem atrapalhar a sua funcionalidade) e concorrência (o uso simultâneo do recurso completo).
- **Simultaneidade:** um recurso estará disponível somente se um certo número de usuários ou tarefas o requisitarem simultaneamente. Por exemplo, uma máquina que deve ser controlada por mais de um operador.
- **Volatilidade:** indica se, após o uso, o recurso estará disponível novamente. Por exemplo, uma impressora é um recurso não-volátil.

No gerenciamento de recursos é possível que mais que dois usuários ou tarefas compartilhem um recurso, requerendo controle sobre o número de recursos a serem compartilhados, o número de usuários ou tarefas que podem requisitar o recurso simultaneamente e o número de vezes que o recurso pode ser usado (volatilidade).

### 2.2.6 Persistência

O AVC tem que prover mecanismos que habilitam o armazenamento e a recuperação dos resultados do trabalho colaborativo. Tal funcionalidade pode ser implementada com o arquivamento do estado do ambiente de colaboração ou o registro do processo todo que conduziu ao resultado final [BECA01].

A persistência em AVCs descreve até que ponto o ambiente virtual existe depois que os participantes deixaram o ambiente. A persistência pode ser dividida em três classes principais [LEIG97]:

- ***Persistência de participação.*** Esta é a persistência na qual o AVC só existe enquanto os participantes estão presentes no ambiente virtual. Quando todos os participantes deixam o ambiente, este é extinto sem registro do seu estado atual e uma nova sessão do ambiente é estabelecida sempre que uma nova colaboração se inicia.
- ***Persistência do estado.*** O estado do ambiente virtual pode ser salvo em determinado momento para ser reproduzido depois ou podem ser registradas as experiências colaborativas inteiras para revisão posterior. Esta forma de persistência pode ser usada para apoiar o controle de versão e anotações feitas no AVC.

- *Persistência contínua.* O estado do ambiente virtual permanece existente mesmo quando nenhum participante está conectado. Conseqüentemente, quando participantes entram novamente no ambiente o estado do mundo pode ter mudado.

### **2.2.7 Controle de Acesso**

O controle de acesso é uma parte indispensável de qualquer sistema de informação compartilhado. Os direitos de acesso controlam como os usuários interagem entre si e com os recursos compartilhados. Os ambientes virtuais colaborativos introduzem exigências novas para controle de acesso, pois os usuários interagem simultaneamente em uma aplicação colaborativa, editando suas estruturas de dados, e necessitam de um conjunto de regras ou padrões de acesso que determinam se um usuário, ou grupo de usuários, ou usuário de um papel particular, executa uma ação específica. Os direitos de acesso da colaboração incluem os direitos de leitura e escrita tradicionais e vários direitos novos como direitos para mudar a visão compartilhada e os direitos de integração com outros usuários [SHEN92] [PETT01].

Os AVCs devem atender os seguintes requisitos relacionados ao controle de acesso:

- Incorporação de papéis múltiplos e dinâmicos: permitir que os direitos de acesso dos usuários sejam deduzidos dos seus papéis, bem como permitir que os usuários possam ter papéis múltiplos simultaneamente e mudar estes papéis dinamicamente durante diferentes fases da colaboração.

- Direitos de colaboração: além das operações tradicionais, tais como leitura e escrita, todas as outras operações, cujos resultados podem afetar múltiplos usuários, devem ser protegidas através de direitos de colaboração.
- Flexibilidade: o sistema deve suportar temas bem definidos, objetos, e direitos de acesso, permitindo uma especificação independente de cada direito de acesso e de cada usuário sobre cada objeto.
- Especificação fácil: os usuários devem poder especificar as permissões de acesso de maneira fácil.
- Armazenamento eficiente e verificação: o armazenamento das definições de acesso e a verificação das regras de acesso devem ser implementadas de forma eficiente.

Em ambientes complexos de computação, tal como os AVCs, onde pode haver muita tensão e conflito em potencial, qualquer mecanismo que se propõe a reduzir ou remover estas preocupações é muito bem-vindo. Bullock & Benford [BULL99] apresentam quatro requisitos básicos para modelos de controle de acesso colaborativos:

- O mecanismo deve ser simples;
- O mecanismo deve ser discreto para os usuários – não deve ter nenhum custo extra (*overhead*) para os usuários para assegurar a segurança em operações inseguras. Assim, o mecanismo deve ser integrado naturalmente na filosofia do sistema e deve utilizar as características existentes do sistema sempre que possível;
- Deve ser fácil a inspeção e alteração de direitos de acesso;

Os efeitos causados pelo controle de acesso devem ser compreendidos e as conseqüências de qualquer mudança devem ser claras.

### 2.2.8 Controle de Concorrência

Os ambientes virtuais distribuídos permitem que usuários distribuídos possam interagir entre si compartilhando um mesmo ambiente virtual. A informação compartilhada é freqüentemente replicada para cada usuário local para prover desempenho interativo aceitável, especialmente quando os usuários estão distribuídos geograficamente em redes extensas como a Internet. A replicação habilita os usuários acessar e atualizar os dados localmente. Porém, o custo da replicação é manter a sincronização entre as réplicas na presença de múltiplas atualizações simultâneas que eventualmente podem conduzir a visões inconsistentes entre os usuários, necessitando assim de um controle de concorrência. Assim, o controle de concorrência tem a função de coordenar as ações que ocorrem simultaneamente [GREE94].

#### 2.3.8.1 Controle de Concorrência por Serialização

Uma abordagem adotada para o controle da concorrência é ordenar os eventos e decidir como executá-los, ou ainda, descobrir e corrigir inconsistências na ordem de execução. A política do controlador de concorrência determina como os eventos podem ser recebidos, a saber:

- Técnicas **não-otimistas** asseguram que os eventos só podem ser recebidos em ordem, garantindo assim a consistência.
- Métodos **otimistas** são baseados na suposição que os eventos conflitantes são raramente recebidos fora de ordem, e que é mais eficiente proceder com a execução e então corrigir os eventuais problemas, do que garantir ordenação correta todas as vezes. Claro que, a correção de problemas de

ordenação pode ser complicado, pois um evento conflitante pode colocar outros eventos válidos fora de ordem, comprometendo todo o encadeamento dos eventos.

Estas políticas de serialização têm problemas. As técnicas não-otimistas podem ser lentas, e as otimistas, devido à necessidade de salvar o estado de uma operação, e eventualmente desfazer uma operação já feita, pode ser complexa e cara.

#### **2.3.8.2 Controle de Concorrência com Acesso Privilegiado através de Técnica de Trancamento (*Locking*)**

Outra abordagem para administração de concorrência é o trancamento, ou *locking*, técnica na qual um usuário participante pode obter acesso privilegiado a algum objeto ou dados compartilhados por um determinado tempo. O trancamento pode ser usado para assegurar a serialização global, controlando a ordem que os usuários obtêm e liberam o poder de trancamento de objetos compartilhados (*locks*).

Tipicamente, um usuário solicita o trancamento de um objeto e, se nenhum usuário estiver com o controle sobre este objeto, a solicitação é aprovada e o usuário obtém a permissão de acesso restrito ao objeto. Assim, pode haver somente um único proprietário com poder de trancamento de um objeto, e se houver outras solicitações, estas serão negadas. Quando o usuário não requer mais acesso restrito ao objeto, este é liberado e torna-se disponível para os demais usuários.

O otimismo de uma política de trancamento de objetos determina se há ou não pausa na execução da colaboração antes de um trancamento ser aprovado e se uma tentativa de trancamento deve ser aprovada antes de ser liberado o objeto compartilhado. A Tabela 2 mostra os níveis de otimismo da política de trancamento.

Políticas específicas de trancamento podem ser usadas para decidir se um trancamento deve ser liberado e concedido a outro usuário. Por exemplo, atribuir o poder de trancamento a outro usuário caso o proprietário atual esteja inativo.

**Tabela 3 – Níveis de otimismo da política de trancamento.**

Nível de Otimismo	Pode manipular o objeto enquanto espera por seu <i>locking</i>	Pode liberar o objeto mudado enquanto espera por seu <i>locking</i>
Não-otimista	Não	Não
Semi-otimista	sim	Não
Completamente-otimista	Sim	Sim

### **2.3.8.3 Controle de Concorrência baseado em Predição**

O controle de concorrência baseado em predição permite interações em tempo real para os usuários, como os esquemas otimistas, e também elimina a necessidade de correções, como os esquemas pessimistas. A questão-chave no controle de concorrência baseado em predição é a precisão em prever, entre os vários usuários que requerem a propriedade de um objeto, qual usuário deve ter a posse e assim conceder a propriedade a um usuário antes que haja algum conflito.

Na abordagem existente, os usuários que desejam obter a propriedade de um objeto devem solicitar a propriedade a todos os donos potenciais, levando em consideração os tempos de colisão. O dono de um objeto coleta somente os pedidos enviados para ele e executa a predição dos tempos de colisão para determinar o próximo dono. Porém, com aumento do número de usuários e objetos em um ambiente virtual, os donos e a rede podem ser sobrecarregados. Assim, o dono do objeto não transfere a propriedade para o próximo dono a tempo, devido ao demorado tempo do processo de pedido de transferência.

Yang & Lee [YANG00] definiram alguns atributos, listados abaixo, que são usados no mecanismo de predição para determinar o próximo proprietário de um determinado objeto.

- ***Raio de atuação da entidade.*** Representa uma área que cerca uma entidade.
- ***Distância.*** É a distância entre o avatar de um usuário e uma entidade alvo.
- ***Velocidade.*** É a velocidade de navegação de um avatar dentro um espaço virtual.
- ***Direção.*** Indica como um usuário está se movendo em relação a uma entidade alvo. A direção tem um de três valores – positivo, zero ou negativo –, que dependem do movimento do usuário e sua orientação. Quando os movimentos de avatar são em direção a uma entidade alvo, o valor da direção fica positivo. Quando se move para longe da entidade, a direção fica negativa, e quando pára, a direção se torna zero.
- ***Tempo de Colisão Predito.*** É o tempo previsto de colisão de um avatar com uma entidade alvo. O tempo é calculado dividindo-se a distância atual pela velocidade.

***Latência.*** É o atraso requerido para trocar uma mensagem de controle de concorrência entre o proprietário de um objeto e um candidato a proprietário. Inclui o tempo de geração da mensagem e de leitura, como também a latência de comunicação entre eles.

### 2.2.9 Comunicação

Os AVCs devem suportar múltiplos usuários geograficamente distribuídos que trocam informações de seus estados entre si. A redução do custo dessa comunicação é uma consideração chave no projeto de AVCs. A arquitetura de comunicação pode ser caracterizada como modelo cliente/servidor, modelo *peer-to-peer* e modelo *peer/servidor* [LEE02].

No modelo cliente/servidor, todas as mensagens são enviadas a um servidor e, então, o servidor as distribui para todos ou alguns usuários conforme as exigências de sincronização. Quando o número de participantes aumenta o servidor pode se tornar um gargalo do sistema.

O modelo *peer-to-peer* permite que os usuários possam trocar mensagens diretamente. Porém, cada usuário tem que assumir toda a responsabilidade de filtragem e sincronização de mensagens.

O modelo *peer/servidor* explora os benefícios dos dois modelos descritos acima, a manutenção da consistência é controlada por um servidor e a comunicação entre os usuários é executada diretamente através de difusão seletiva.

## 2.3 Tipos de Dados Transmítidos em Ambientes Virtuais Colaborativos

Com o crescimento do número de usuários no Ambiente Virtual Colaborativo podem agravar-se os problemas devido à geração de grandes volumes de tráfego na rede, ocasionados por mensagens que tratam da manutenção da consistência dos dados, causalidade (atualização dos dados de forma imediata), comunicação confiável, além da interação em tempo-real.

Vários tipos de dados podem ser trocados em AVCs, cada um com diferentes requisitos e prioridades, conforme classificados por Pandzic [PAND97]:

- **Download.** Quando um usuário novo se associa a uma sessão do AVC, a descrição completa do Ambiente Virtual deve ser carregada na máquina do novo usuário, incluindo os objetos 3D, estruturados em uma hierarquia de cena, texturas, e, possivelmente, comportamentos em forma de *scripts* ou programas. O novo usuário também tem que baixar as descrições de todos os usuários e enviar a sua própria descrição para todos os outros usuários participantes. O *download* não se restringe à fase de estabelecimento de sessão, mas também pode acontecer a qualquer momento durante a sessão devido à introdução de novos objetos na cena.
  
- **Atualizações de estado.** Todas mudanças de estado no ambiente, e as incorporações dos usuários, têm que ser propagado pela rede. Para os objetos no ambiente, as mudanças normalmente consistem em alterações de posição ou orientação desses objetos. A livre deformação de objetos geralmente não é suportada em sistemas de AVC devido à necessidade de maior largura da banda, e requisitos de tempo-real, porém é uma característica muito desejável de ser incluída em sistemas de AVC. Em relação aos usuários, as atualizações de estado envolvem também a movimentação dos usuários no ambiente e a comunicação entre eles, através de expressões faciais e corporais que reflitam as ações de corpos naturais.

- **Eventos.** São tipicamente mensagens curtas sobre eventos que acontecem no ambiente e podem influenciar potencialmente, por muito tempo, o estado de ambiente.
- **Mensagens de sistema.** Estas mensagens são usadas, tipicamente, durante o estabelecimento e encerramento de sessão. A segurança delas é essencial porque erros podem causar mal funcionamento do sistema.
- **Vídeo.** Os vídeos podem ser usados de formas diferentes para várias aplicações no Ambiente Virtual, tais como apresentações de vídeos virtuais, mapeamento de textura facial para comunicação facial. Os requisitos de qualidade de vídeo podem variar de aplicação para aplicação.
- **Áudio.** O uso mais comum de áudio em AVCs é para comunicação via voz. Porém, o som sintético 3D também pode ser importante em um Ambiente Virtual.
- **Texto.** O uso mais comum de texto é em conversas baseadas em texto (*Chat*) entre os usuários.

Normalmente, nem todos os dados precisam ser transmitidos para todos os usuários, por exemplo, se dois usuários estão dentro de zonas diferentes, não há necessidade de haver comunicação, através de expressões faciais entre eles, pois um não vê o outro. Estas transferências de dados podem usar um servidor inteligente para filtrar os dados ou usar difusão seletiva de forma dinâmica.

## 2.4 Ambientes Virtuais Colaborativos Existentes

O número de sistemas de AVCs cresceu muito nos últimos anos devido aos avanços nas áreas de processamento e de comunicação. Estes sistemas variam de protótipos de pesquisa a produtos comerciais. A seguir são descritos alguns desses sistemas.

### 2.5.1 DIVE

O Ambiente Virtual Interativo Distribuído (*Distributed Interactive Virtual Environment* – DIVE), um dos primeiros ambientes virtuais colaborativos, é um sistema de realidade virtual multiusuário desenvolvido pelo *Distributed Systems Laboratory of the Swedish Institute of Computer Science* (SICS). DIVE permite que participantes distribuídos, representados geometricamente na forma de avatares, naveguem livremente dentro de mundos virtuais 3D, e comuniquem-se em tempo-real, através de áudio, vídeo, texto e gestos gráficos simples. Ferramentas básicas de apoio a reuniões são oferecidas, como mostra a Figura 2. Os usuários também podem manipular objetos no mundo virtual (mover e girar objetos ou adicionar objetos novos) [HAGS96].



Figura 2 – Ambiente Virtual Colaborativo de conferência [DIVE].

### 2.5.2 NPSNET-IV

NPSNET-IV é um sistema de ambiente virtual 3D em rede desenvolvido pelo *Computer Science Department of the U. S. Naval Postgraduate School (NPS)*. Foi projetado para apoiar o treinamento de exércitos através de exercícios de combate em ambientes simulados. A máquina de cada participante pode ser configurada para simular um veículo militar ou pessoal de infantaria, e podem ser suportadas milhares destas entidades, interagindo entre si, como mostra a Figura 3. Como resultado, o NPSNET-IV gerou, como uma das contribuições principais, o desenvolvimento de arquiteturas de rede escaláveis e técnicas de gerenciamento de percepção dos usuários e de suas ações [MACE94].



Figura 3 – Ambiente de Treinamento Virtual [NPSNET].

### 2.5.3 MASSIVE

O Modelo, Arquitetura e Sistema para Interação Espacial em Ambientes Virtuais (*Model, Architecture and System for Spatial Interaction in Virtual Environments – MASSIVE*), desenvolvido pela Universidade de Nottingham, é um sistema de realidade virtual que suporta um número muito grande de participantes. Os usuários interagem num mesmo ambiente virtual, através de diferentes equipamentos, mídias e interfaces com o usuário (2D, 3D, texto, áudio). As características principais de MASSIVE incluem a

implementação de um modelo espacial para interação, o uso de negociação espacial para as interações entre os objetos no mundo virtual e recursos para a interação entre os usuários, Figura 4 [GREE95]. MASSIVE-3, a terceira geração do sistema, oferece um *framework* flexível e expressivo baseado em locais, aspectos e políticas para suporte à manutenção da consistência dos dados, da estruturação do mundo virtual e do gerenciamento da percepção dos usuários [GREE00].

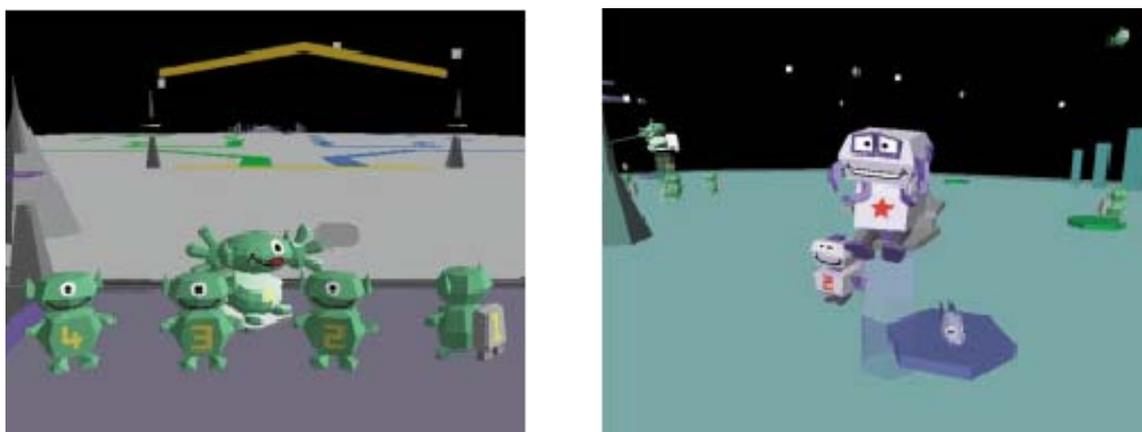


Figura 4 – Imagens de TV habitada criadas usando MASSIVE-2 [BENF01].

#### 2.5.4 SPLINE

A Plataforma Escalável para Extensos Ambientes Interativos em Rede (Scalable Platform for Large Interactive Networked Environments – SPLINE) é uma plataforma multiusuário para Ambientes Virtuais desenvolvido pelo *Mitsubishi Electric Research Laboratories*. SPLINE foi apresentado como um conjunto de mecanismos de suporte à criação de aplicações de ambientes virtuais colaborativos para usuários finais, como mostra a Figura 5. Em particular, o SPLINE é definido como uma API Java usada para criar e conectar mundos virtuais. Um das características mais interessantes do SPLINE é que, em lugar de definir mundos virtuais completos, são definidas regiões menores, chamadas locais, que podem ser unidos para formar mundos completos ou espaços de redes. Os locais podem

ser integrados de forma muito flexível, por exemplo, para a criação de um edifício [BARR96].

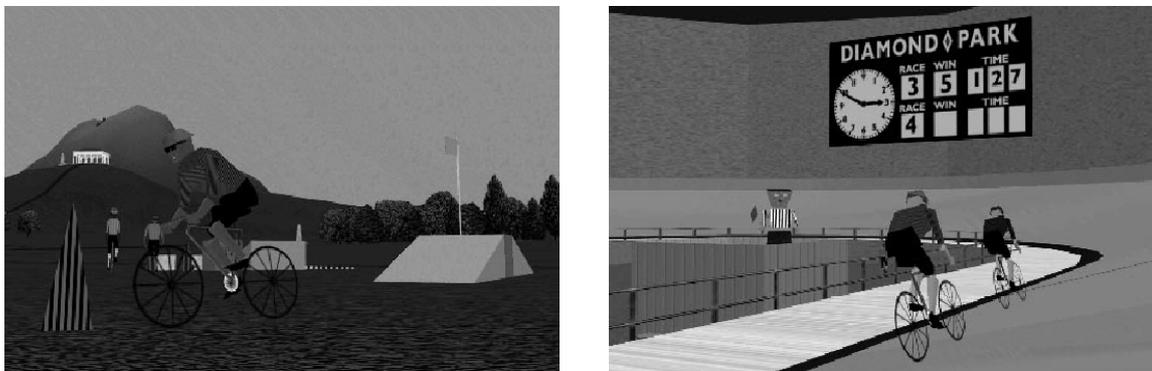


Figura 5 – Diamond Park [SPLINE].

### 2.5.5 COVEN

O Ambiente Virtual Colaborativo (*Collaborative Virtual Environment – COVEN*), projeto europeu que é parte do programa *Advanced Communication Technology and Services (ACTS)*, busca o desenvolvimento de tecnologias para a construção de AVCs. O projeto focaliza principalmente em arquiteturas de redes escaláveis para suportar trabalho colaborativo entre múltiplos usuários geograficamente dispersos que são integrados em um ambiente virtual compartilhado, Figura 6 [BRAS00] [KUIJ97].



Figura 6 – Ambientes Virtuais de Conferência [COVEN].

### 2.5.6 VLNET

A Rede de Vida Virtual (*Virtual Life Network* – VLNET), desenvolvido pelo *MIRALab* da Universidade de Geneve e o *Computer Graphics Lab of the Swiss Federal Institute of Technology*, é um sistema de AVC em rede que usa avatares altamente realistas para a representação dos participantes. Os avatares em VLNET são corpos articulados, com faces articuladas, que reproduzem movimentos naturais, ações, emoções e fala, conforme mostra a Figura 7. VLNET, através de um conjunto de interfaces externas, provê meios flexíveis para controlar a entrada de novos usuários e também de outros objetos na cena [PAND97].



Figura 7 – Avatares em VLNET [VLNET].

### 2.5.7 COSMOS

O Sistema Colaborativo baseado em Objetos e Fluxos MPEG-4 (*Collaborative System based on MPEG-4 Objects and Streams* – COSMOS), desenvolvido pelo *Multimedia Communication Research Laboratory* da Universidade de Ottawa – Canadá, é um *framework* orientado a objeto para o desenvolvimento rápido de aplicações colaborativas. A Figura 8 mostra uma cena no COSMOS. COSMOS oferece a base para a construção de novas aplicações colaborativas, com componentes reutilizáveis, modulares e extensíveis que cumprem os requisitos dos Ambientes Virtuais Colaborativos, tais como interatividade, tempos de resposta baixo, robustez, e manutenção de consistência [DARL00].

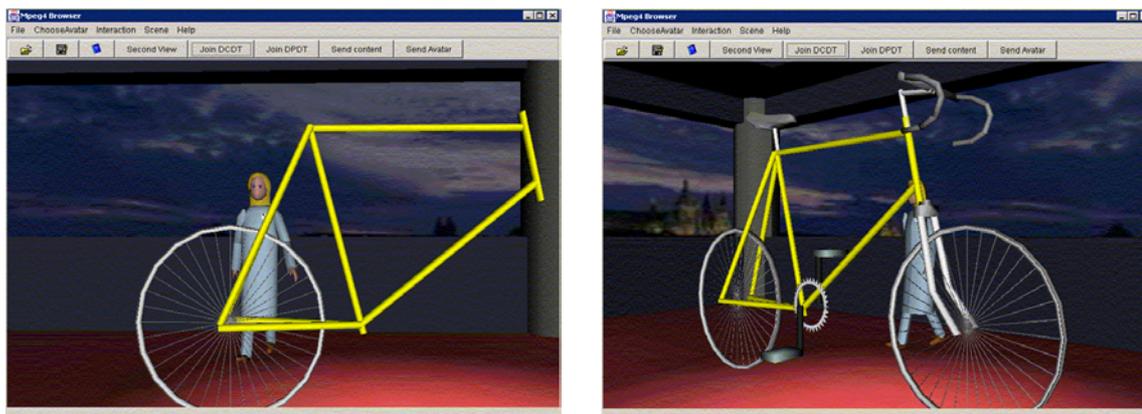


Figura 8 – Ambiente de Montagem Colaborativa de uma bicicleta [COSMOS].

## 2.5 Modelagem 3D em Ambientes Virtuais Colaborativos

Os Ambientes Colaborativos para Modelagem 3D são caracterizados pela interação entre múltiplos usuários na modificação ou deformação da forma geométrica de objetos 3D compartilhados. Nestes ambientes, participantes dispersos geograficamente podem participar na modelagem e visualização de protótipos virtuais e podem, através da validação e simulação digital do protótipo, tomar decisões quanto à viabilidade do projeto [KAN01] [LI01].

Com o uso de protótipos virtuais, o custo com testes pode ser menor do que com o uso de protótipos reais, além do protótipo virtual ter o potencial de prover maior precisão na análise do projeto. A construção e modelagem de um protótipo virtual proporcionam a validação deste protótipo de acordo com as especificações iniciais, podendo ser realizadas algumas mudanças, o detalhamento do modelo e outras otimizações. Desta forma os projetistas e engenheiros são capacitados a investigar a viabilidade de um projeto de maneira efetiva em um estágio preliminar, prevenindo o gasto com mudanças em fases posteriores do projeto do produto [RIED97] [ZACH98].

Num AVC de modelagem 3D, os usuários são imersos no projeto e interagem com o protótipo virtual e outros objetos da cena, tais como áudio, vídeo, texto, imagens, e também com outros usuários, tornando-se um mecanismo eficiente e efetivo na obtenção de soluções ótimas de projetos através da visualização, simulação e avaliação de um número grande de alternativas para o projeto [HIND00] [STUE02].

Muitos esforços de pesquisas foram e ainda estão sendo investidos no desenvolvimento de Ambientes Virtuais Colaborativos, como os AVCs mostrados acima, porém poucos suportam a modelagem virtual colaborativa, no qual as formas dos objetos de interesse são continuamente modificadas. Os desafios desses ambientes estão relacionados principalmente com a renderização em tempo real dos objetos modificados, com a interação entre os usuários participantes e o ambiente virtual, tais como controle de concorrência, controle de acesso, gerenciamento de interesse, e a manutenção da consistência do ambiente virtual compartilhado [LI01]. A seção seguinte descreve alguns AVCs de modelagem 3D.

### **2.5.1 M3D Editor**

O M3D Editor é uma ferramenta de projeto 3D concorrente e de visualização, desenvolvido na Universidade das Ilhas de Balearic – Espanha, utilizado em ambientes de edição 3D colaborativos. Os usuários podem abrir cenas diferentes de arquivos ou bancos de dados e, editá-las em janelas separadas. A ferramenta permite que múltiplos usuários editem a mesma cena ou o mesmo conjunto de cenas simultaneamente - a mudança da cena aparecerá na janela do outro usuário imediatamente, como mostra a Figura 9. Os usuários podem realizar operações de “copiar” e “colar” e também podem desfazer qualquer operação. Portanto, um controle de concorrência é necessário para assegurar a consistência do projeto global [CRES00].

A renderização tridimensional a taxas interativas requer o acesso rápido ao banco de dados geométrico, assim, os dados de cena são replicados em cada terminal local. Toda réplica dos dados pode ser vista como uma cópia local de um banco de dados persistente. O formato de arquivo VRML é usado para a troca de dados. Os dados de projetos tridimensionais são armazenados em uma estrutura de árvore e os elementos da cena

tridimensional são representados por nós da árvore de cena. A ferramenta de projeto 3D colaborativa executa em uma plataforma de comunicação multiusuário distribuída, chamada JESP.

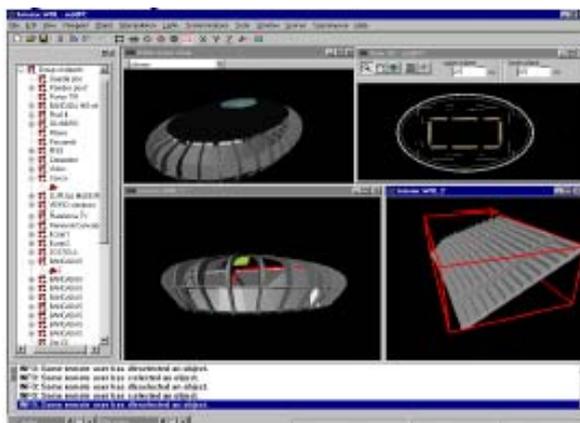


Figura 9 – M3D Editor [CRES00].

### **Manutenção da Consistência no M3D Editor**

A manutenção da consistência do banco de dados persistente é uma tarefa crucial na sessão de trabalho colaborativa. Como os usuários estão trabalhando com réplicas diferentes do mesmo dado, estas réplicas devem estar consistentes. As modificações no banco de dados persistente são enviadas para todos os usuários usando um protocolo de aplicação, chamado Mu3D. Este protocolo é composto de mensagens pequenas que especificam os eventos de sessão e as mudanças de dados. Os usuários locais são responsáveis por enviar as modificações locais encapsuladas em mensagens Mu3D para os outros usuários. Quando um usuário local recebe uma mensagem, o evento remoto é recriado localmente.

### **Controle de Concorrência no M3D Editor**

O controle de concorrência é necessário em trabalho colaborativo para solucionar conflitos entre os participantes. O mecanismo de controle de concorrência utiliza

uma operação de seleção, que é usada pelas operações que requerem controle de concorrência para poderem ser executadas somente em objetos selecionados.

Quando o usuário seleciona um objeto, ele se torna o dono deste objeto até que o objeto seja liberado por ele. Os usuários não podem selecionar um objeto se outro usuário já tiver selecionado, pois só um usuário pode modificar um determinado objeto de cada vez. Esta política de seleção assegura a consistência no banco de dados persistente efetivamente pela implementação do protocolo de Mu3D. Portanto, a seleção e a liberação dos objetos são notificadas para os outros usuários enviando mensagens Mu3D.

As limitações do M3D Editor estão relacionadas principalmente com a atualização da cena gráfica, pois toda estrutura hierárquica dos nós da cena deve ser enviada aos usuários para que a cena seja atualizada. E também por usar uma solução proprietária para suporte à colaboração.

### **2.5.2 Escultura Virtual Distribuída Colaborativa**

A Escultura Virtual Distribuída Colaborativa, desenvolvida na Universidade da Cidade de Hong Kong, é um *framework* que provê um ambiente virtual multiusuário colaborativo para que participantes geograficamente separados possam modificar as formas de objetos 3D e visualizar trabalhos de escultura complexos, através da Internet. O sistema suporta desde prototipação de produtos até esculturas de arte [LI 97] [LI01].

O sistema é baseado em um modelo híbrido de comunicação que combina os modelos cliente-servidor e *peer-to-peer* para suportar a replicação rápida de dados. No *framework* os objetos deformáveis são modelados com superfícies NURBS (*Non-Uniform Rational B-Splines*), para a redução do tempo de transmissão das atualizações no objeto, provendo deformação em tempo-real e aumentando a interatividade do ambiente de escultura virtual colaborativa, pois as modificações são enviadas frequentemente. Isto ocorre devido as

superfícies NURBS serem representadas de uma forma muito mais compacta, e podem ser deformadas apenas mudando as posições dos pontos de controle, conforme ilustra a Figura 10.

Uma estrutura de dados linear é utilizada para a transmissão dos objetos deformáveis, chamada lista de superfície, que diminui o tamanho dos fluxos de dados.

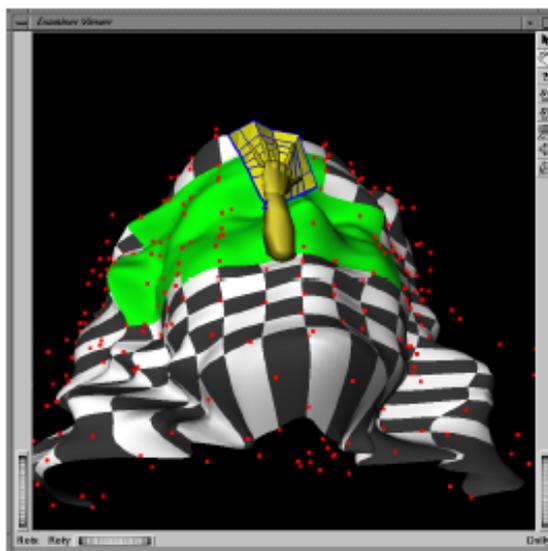


Figura 11 - Escultura Virtual Distribuída Colaborativa [LI01].

O trabalho focaliza mais o uso de NURBS na construção dos objetos deformáveis do que nas questões relacionadas a colaboração, tal como o gerenciamento de sessão.

### 2.5.3 SpinCAS

SpinCAS (*Spin-3D Computer Aided Sculpting*), desenvolvido pelo Laboratório de Informática Fundamental da Universidade de Lille – França, é uma arquitetura de software que provê funcionalidades para esculpir objetos virtualmente, entre múltiplos usuários, na Internet, em tempo-real. O Spin-3D é uma interface tridimensional para trabalho colaborativo síncrono, projetada para reuniões de pequenos grupos, tais como educação a distância e

situações de colaboração em projetos. Os objetos são descritos no formato VRML e múltiplos usuários são suportados através da integração das linguagens VRML e Java. O SpinCAS é mostrado na Figura 11. No Spin-3D os dados compartilhados do grafo da cena se comunicam com suas cópias remotas, através da tecnologia CORBA *multicast* para ações discretas e através de fluxos de comunicação por difusão seletiva, em tempo-real, para as animações [PICA01] [GRISO02].



Figura 13 - SpinCAS [GRISO02].

As limitações do SpinCAS estão relacionadas às limitações da linguagem VRML, tais como: não suporta formato de arquivo binário para a descrição da cena e suporte à multimídia limitado [ABRE01].

## 2.6 Considerações Finais

Este capítulo apresentou uma visão geral dos Ambientes Virtuais Colaborativos, suas aplicações e seus principais requisitos. Vários projetos de Ambientes Virtuais Colaborativos, como DIVE, NPSNET, MASSIVE, SPLINE, COVEN, VLNET E COSMOS foram apresentados que focalizam em questões de interatividade com o usuário, controle de concorrência, gerenciamento de interesse, tráfego de dados, latência do sistema, consistência do ambiente, particionamento do mundo e arquiteturas de redes escaláveis, para

suportar trabalho, treinamento e simulação entre múltiplos usuários geograficamente distribuídos.

Sistemas que tratam da colaboração para a Modelagem 3D, usados em aplicações, por exemplo, de prototipagem de produtos, como o SpinCAs, o M3D e Escultura Virtual Distribuída Colaborativa, também foram apresentados. O problema com estes sistemas é que utilizam a linguagem VRML, a qual possui várias limitações e soluções proprietárias para suporte à colaboração.

Como o objetivo deste projeto é a implementação de um AVC de modelagem 3D com funcionalidades para criação e modificação da cena gráfica e que possa proporcionar a visualização do que está sendo modelado em dispositivos com diferentes capacidades, o padrão MPEG-4 foi escolhido como base para a implementação. Desta forma, no próximo capítulo é apresentado o padrão MPEG-4 no suporte a construção de ambientes virtuais, as suas principais características bem como o seu potencial em relação às demais tecnologias 3D. Também é apresentado o padrão emergente MPEG-4 MU para o suporte aos requisitos dos AVCs, tais como gerenciamento de sessão, manutenção da consistência do ambiente compartilhado, controle de concorrência, entre outros.

## 3 O Padrão Emergente MPEG-4 Multiusuário no Suporte aos AVCs

### 3.1 O Padrão MPEG-4

O MPEG-4 [N4264] é um padrão da (ISO/IEC JTC1/SC29/WG11) desenvolvido pelo MPEG (*Moving Picture Experts Group*), responsável também pelo desenvolvimento dos padrões MPEG-1, usado para criação de vídeo interativo em CD-ROM, e MPEG-2, usado para DVD e televisão digital. O MPEG-4 é um esforço internacional que envolve centenas de pesquisadores e engenheiros do mundo inteiro.

O MPEG-4 suporta com sucesso comprovado três campos: televisão digital, aplicações gráficas interativas (conteúdo sintético) e multimídias interativas (*World Wide Web*), distribuição e acesso para conteúdo. O padrão provê elementos tecnológicos unificados que habilitam a integração da produção, distribuição e paradigmas de acesso a conteúdo destes três campos [N4668]. O MPEG-4 é um padrão promissor para visualizadores em dispositivos com capacidades diferentes, tais como celulares, PDAs, *set-top-boxes* e PCs. O MPEG-4 trata os aspectos de composição, sintetização, compressão, sincronização e distribuição de objetos áudio-visuais, além de tratar da descrição das cenas.

O padrão MPEG-4 possui um formato de arquivo binário, o BIFS (*Binary Format Scenes*), que encapsula os segmentos elementares (fluxos de dados) do conteúdo da cena para apresentá-los no visualizador MPEG-4 (*Player*). Isto faz com que o tamanho final

do arquivo contendo o ambiente virtual seja pequeno e, portanto, mais rápido para ser transportado via rede de comunicação [N4264] [ABRE01].

Atualmente o padrão MPEG-4 [N4264] está dividido em sete frentes de trabalho: *Systems, Visual, Audio, Conformance Testing, Reference Software, Delivery Multimedia Integration Framework (DMIF) e Optimized Software for MPEG-4 Visual Tools.*

O Sistema MPEG-4 descreve um sistema para comunicação de cenas áudio-visuais interativas, incluindo as seguintes ferramentas:

- Um modelo de terminal para o gerenciamento de *buffer* e tempo;
- Uma representação codificada de informações de descrição da cena áudio-visual interativa (BIFS);
- Uma representação codificada da identificação e descrição de fluxos áudio-visuais, bem como as dependências lógicas entre o fluxo de informações (objeto ou outros descritores);
- Uma representação codificada de sincronização de informação (*Sync Layer - SL*);
- Uma representação codificada de informações de conteúdo áudio-visual descritiva (informação de conteúdo do objeto – OCI);
- Uma interface para o gerenciamento de propriedade intelectual e sistemas de proteção (IPMP);
- Uma máquina de aplicação (MPEG-Java – MPEG-J);

- Uma representação multiplexada de fluxos elementares individuais em um único fluxo (*FlexMux*).

Essas ferramentas descrevem a arquitetura do Sistema MPEG-4 melhor explorada abaixo.

### 3.1.1 Arquitetura do Padrão MPEG-4

A representação de informações que descreve uma cena áudio-visual interativa, contendo informações áudio-visuais e informações associadas à descrição da cena, foi especificada no *Final Committee Draft of International Standard*. A entidade que recebe e apresenta a representação codificada de uma cena áudio-visual interativa é genericamente citada como um "terminal áudio-visual", que pode corresponder a uma aplicação *standalone* ou ser parte de um sistema de aplicação [N2201].

As operações básicas realizadas no terminal são:

- **Obter acesso à sessão inicial** - O terminal obtém acesso às informações a partir de uma sessão inicial provida da disponibilização do conteúdo das informações;
- **Estabelecer contextos de sessão e uma interface de entrega** - Utilidade para estabelecer contextos de sessão e uma interface para a camada de entrega o qual separa fluxos como o meio de armazenamento e transporte;
- **Prover a localização dos fluxos elementares** - Provê a localização de um ou mais fluxos elementares os quais são partes da representação do conteúdo previamente codificadas. Muitos desses fluxos podem ser agrupados, usando uma ferramenta de multiplexação como, por exemplo, *FlexMux tool*.

Cada fluxo elementar contém somente um tipo de dado. Como exemplo pode-se citar fluxo de imagens (JPEG) ou áudio (G723). Os fluxos elementares são decodificados usando seus decodificadores de fluxo específicos. Os objetos áudio-visuais são compostos de acordo com as informações da cena e apresentados para o terminal fazer, futuramente, a interação com os usuários. Todos esses processos são sincronizados de acordo com o Modelo de Decodificação do Sistema (SDM), usando a informação de sincronização estabelecida na camada de sincronização (SL) [N4264].

A arquitetura composta por seus componentes, como ilustrado na Figura 12, são detalhados a seguir.

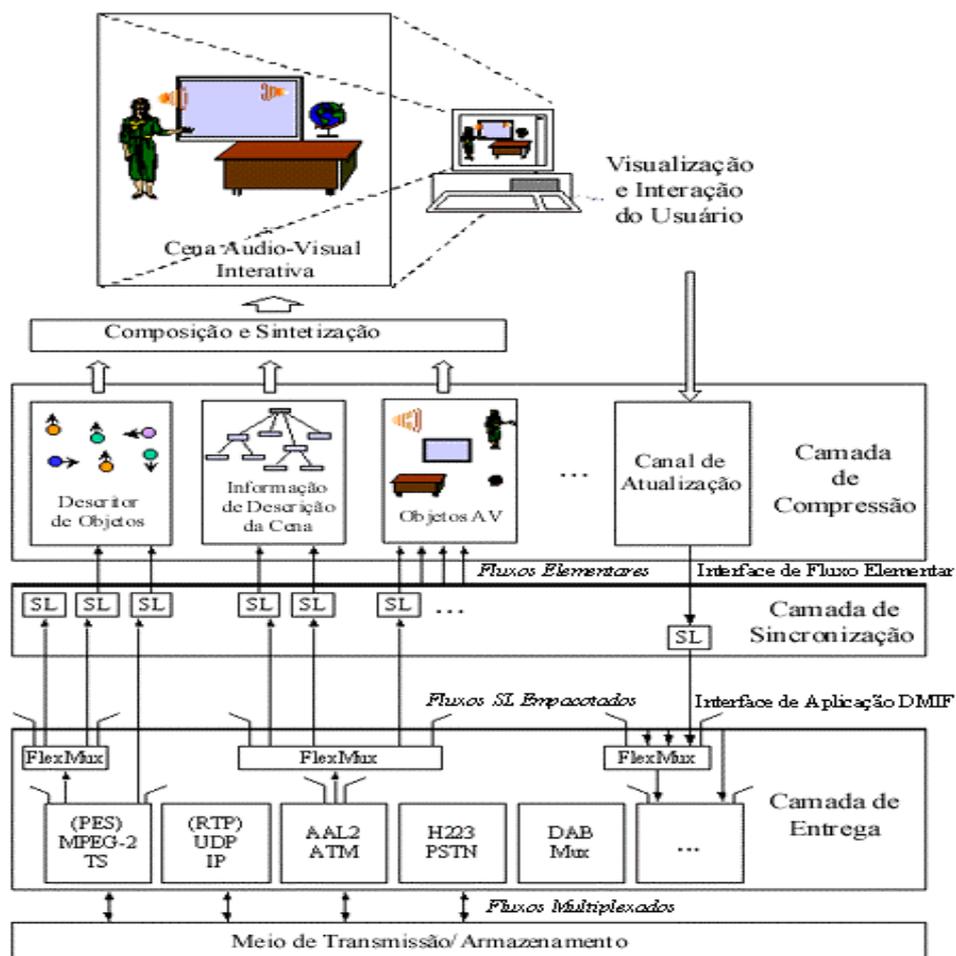


Figura 14 - Arquitetura do Terminal MPEG-4 [extraída de N4264].

### ***Modelo de Terminal – Modelo de Decodificador do Sistema (SDM)***

O SDM fornece uma visão abstrata do comportamento de um terminal, conforme a especificação do padrão MPEG-4. Sua proposta é permitir a um transmissor prever como o receptor se comportará em termos de gerenciamento de *buffer* e sincronização, quando ocorrer a reconstrução da informação áudio-visual que envolve a apresentação [N4264] [N2201].

O SDM especifica:

- A interface por acessar fluxo de dados desmultiplexados (*Stream Multiplex Interface*);
- Decodificação de *buffers* para dados comprimidos para cada fluxo elementar;
- O comportamento de decodificadores de objeto de mídia;
- Memória de composição para dados descomprimidos para cada objetos de mídia e o comportamento de produção para o compositor.

### ***Camada de Entrega - Multiplexação de Fluxos***

A Multiplexação de Fluxos é usada para transmitir e armazenar conteúdo. Somente a interface dessa camada é especificada (denominada Interface de Aplicação DMIF (DAI)), e define não somente uma interface para a entrega de dados encadeados (*streaming data*), mas também informações de sinalização requeridas para o estabelecimento e encerramento de canais e sessões.

## ***Camada de Sincronização de Fluxos - (Sync Layer - SL)***

Os fluxos elementares são quaisquer tipos de dados encadeados.

Um fluxo elementar pode conter:

1. ***Descritores de Objetos:*** O propósito da descrição de objetos é identificar e descrever fluxos elementares e associá-los adequadamente a uma descrição da cena auditivo-visual. Os descritores de objeto são carregados como fluxos elementares. Cada descritor de objeto é nomeado com um identificador (*Object Descriptor ID*) que é único dentro de uma extensão de nome definida. Este identificador é usado para associar objetos áudio-visuais na descrição da cena, como um descritor particular de objeto, assim os fluxos elementares relacionaram, em particular, àquele objeto. Descritores de fluxos elementares também incluem informação sobre o formato de codificação, informação de configuração para o processo de decodificação e a camada de sincronização, como também qualidade de exigências de serviço para a transmissão do fluxo e identificação de propriedade intelectual.
2. ***Informações de conteúdos de Objetos:*** Informações de conteúdos de objetos (OCI) carregam a informação descritiva sobre os objetos áudio-visuais. Os descritores de conteúdos principais são: descritores de classificação do conteúdo, descritores de palavras chaves, descritores de classificação, descritores de linguagens, descritores textuais, e descritores sobre a criação do conteúdo. Podem ainda ser incluídos descritores de OCI diretamente no descritor de objeto relacionado ou a um descritor de fluxo elementar, ou, se for variante de tempo, pode ser

levado por si só em um fluxo elementar. Um fluxo de OCI é organizado em uma sucessão de entidades pequenas, sincronizadas, chamados eventos, os quais contêm um conjunto de descritores OCI. O fluxo OCI pode ser associado a descritores de objetos de múltiplos.

3. ***Fluxos da Descrição da Cena:*** A descrição endereça a organização de objetos audiovisuais numa cena, em ambos atributos espacial e temporal. A descrição consiste em uma hierarquia codificada (árvore) de nós com atributos e outras informações. Os nós da folha nesta árvore correspondem aos dados áudio-visuais elementares, considerando que nós intermediários se agrupam a este material para formar objetos áudio-visuais, e outras operações em objetos áudio-visuais (nós da descrição da cena). A descrição de cena pode evoluir com o passar do tempo usando atualizações onde, mecanismos de interatividade são integrados com a informação da descrição da cena, na forma de recursos de eventos e rotas, bem como sensores (nós especiais que podem ativar eventos baseados em condições específicos). Estes recursos de eventos e rotas fazem parte de nós da descrição da cena, e assim permite a união de comportamento dinâmico e interativo com a cena específica.
4. ***Fluxos Áudio-visuais:*** Os dados audiovisuais reconstruídos são projetados disponibilizando o processo de composição para uso potencial durante a retribuição da cena.

Tais fluxos são transportados como fluxos SL empacotados (*SyncLayer-packetized – SL-Packet*) na Interface de Aplicação DMIF, fornecendo informações de tempo e sincronização, bem como informações de fragmentação e de acesso aleatório. A camada de

sincronização extrai esta informação de tempo para capacitar a decodificação sincronizada e, subseqüentemente, a composição do dado do fluxo elementar.

### ***Camada de Compressão***

A camada de compressão recebe o dado em seu formato codificado e desempenha as operações necessárias para reconstruir a informação original. A informação decodificada é então usada pelo terminal de composição, renderização e subsistemas de apresentação.

A seguir é descrito o protocolo *BIFS-Command* utilizado para a atualização da cena gráfica.

#### **3.1.2 Protocolo BIFS-Command**

O padrão MPEG-4 é projetado para ser usado em aplicações de radiodifusão (*broadcast*), aplicações interativas e aplicações de comunicação *peer-to-peer*. Para suportar estes requisitos, um conceito importante desenvolvido no MPEG-4 é que a própria aplicação pode ser vista como um fluxo temporal, ou seja, a apresentação ou a própria cena tem uma dimensão temporal. Na Web, o modelo usado para apresentações multimídia é que uma descrição de cena (por exemplo, uma página HTML ou uma cena VRML) é carregada de uma vez, e então executada localmente. No MPEG-4, a apresentação BIFS, que descreve a própria cena, é entregue na forma de *streaming* ou encadeamento. Assim, o modelo básico é que uma cena inicial é carregada e pode receber atualizações adicionais, e na realidade, a própria cena inicial é considerada uma atualização [**SIGN02**].

O mecanismo usado para prover as informações BIFS para carregamento na forma de *streaming* para o receptor inclui o protocolo *BIFS-Command* e o Fluxo Elementar que o transporta, chamado de fluxo *BIFS-Command*. O protocolo *BIFS-Command* é usado

para enviar comandos que agem na cena de um servidor para um terminal cliente em que a cena reside [WALS01]. O protocolo BIFS-*Command* transporta comandos para a substituição de uma cena, adição ou remoção de nós, modificação de campos, etc. Um fluxo BIFS-*Command* pode ser lido da Web como qualquer outra cena, potencialmente contendo somente um comando, por exemplo, o comando *ReplaceScene*, mas também pode ser uma radiodifusão, tal como um fluxo "push", ou mesmo troca de informações em uma aplicação de comunicação ou colaborativa.

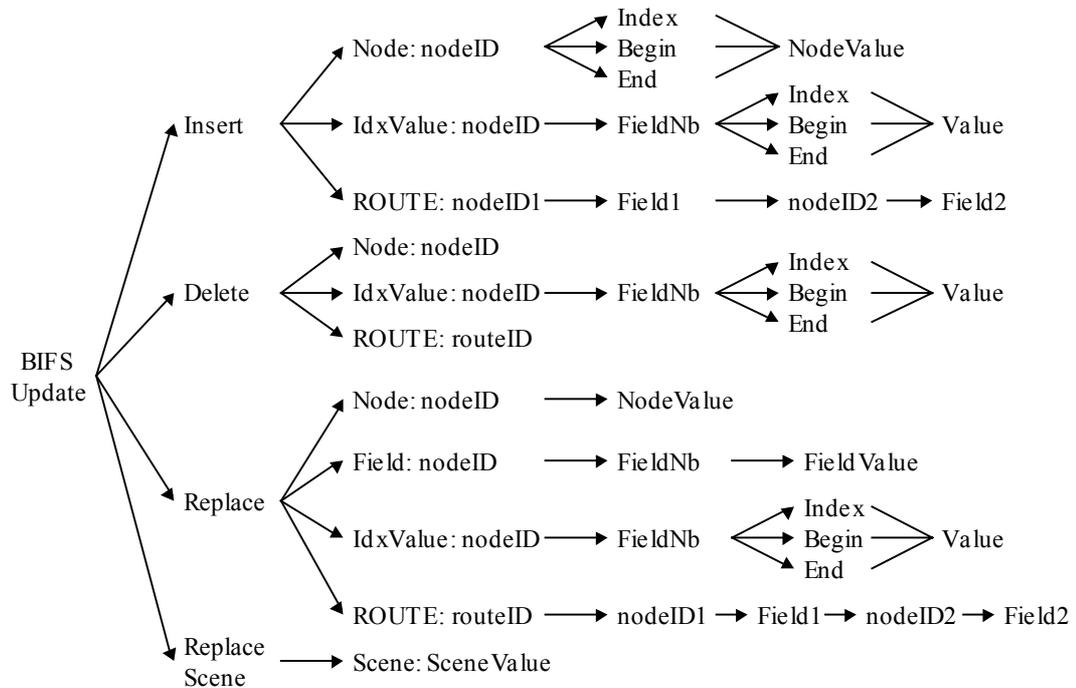
Os BIFS-*Commands* se enquadram em quatro funcionalidades principais: substituição de cena, inserção de nó/campo/rota, remoção de nó/valor/rota, e substituição de nó/campo/valor/rota, como mostra a Figura 13. Os comandos habilitam as seguintes operações:

- ***Substituição da cena atual inteira por uma nova cena.*** Quando um comando *ReplaceScene* é recebido, o contexto inteiro é reajustado, e um grafo de cena novo corresponde à cena BIFS nova é construído.
- ***Comando de Inserção.*** Este comando se divide em três subtipos:
  - ***Inserção de Nó.*** Podem ser inseridos nós no campo filho de um nó de agrupamento. Este comando permite a inserção de um nó no início, final, ou uma posição indexada na lista de nós filhos de um nó já existente.
  - ***Inserção de Campo Indexada:*** com este comando de atualização, um campo genérico é inserido em uma posição especificada em um campo de valor múltiplo.

- ***Inserção de Rota.*** Este comando pode ser usado para habilitar a interação do usuário ou outra funcionalidade dinâmica na cena unindo fonte de evento e campos alvos na cena.
- ***Comando de Remoção.*** Este comando também se divide em três subtipos, análogo ao comando de inserção:
  - ***Remoção de nó.*** Simplesmente remove o nó identificado. Entretanto, é possível remover um nó que não tem um ID usando o comando de remoção *IndexedValue*.
  - ***Remoção de IndexedValue.*** Este comando permite a remoção de uma entrada especificada em um campo de valor múltiplo.
  - ***Remoção de Rota.*** Simplesmente remove uma Rota.
- ***Comando de substituição.*** Este comando também é dividido em três subtipos, mais um para substituir valores de campo de valores únicos:
  - ***Substituição de nó.*** Substitui um nó existente na cena com um novo.
  - ***Substituição de campo.*** Modificação do valor de um campo dentro de um nó. Por exemplo, este comando pode ser usado para mudar uma cor, uma posição, o vértices de uma malha, ou mudar um objeto de tempo em tempo.
  - ***Substituição de IndexedValue.*** Este comando é bem parecido ao comando de substituição de campo, a não ser que o campo referido é um campo com múltiplos valores, e conseqüentemente o comando

também inclui informação de indexação para identificar um valor particular que deve ser substituído.

- **Substituição de Rota.** Substitui uma Rota existente com um novo.



**Figura 15 - Tipos de BIFS-Commands [extraída de N2201].**

A capacidade do BIFS-Command pode ser usada para modificar uma cena para atualização de conteúdo em tempo-real, por exemplo, em manchetes de notícias. Também pode ser usado para modificar a experiência do usuário, para interativamente mudar a posição do usuário ou ponto de visão, ou enviar mensagens de estados ou de conversa entre usuários diferentes em um servidor.

Em julho de 2001 foi aprovada a proposta do padrão emergente MPEG-4MU. Esse padrão descreve a implementação de novos nós no padrão MPEG-4 explorando questões relacionadas a: Ambientes Virtuais multiusuário, compartilhamento de objetos, entre outros. Esses novos conceitos bem como toda a arquitetura desse padrão emergente serão detalhados na subseção 3.2.

### 3.2 O Padrão Emergente MPEG-4 MU

O suporte para Ambientes Virtuais multiusuário no padrão MPEG-4 está em fase de padronização, com propostas para uma arquitetura de alto-nível e extensões exigidas ao padrão MPEG-4 existente para suportar mundos multiusuário (MU) interoperáveis. O principal objetivo é definir um *framework*, dentro o qual vendedores de software e provedores de conteúdo poderão implementar as próprias soluções multiusuário. A interoperabilidade deve ser definida nos seguintes níveis [M3874] [N4272]:

- **Objetos compartilhados:** objetos arbitrários podem ser declarados como compartilhados e são *uploaded* e incorporados no ambientes MU.
- **Mundo:** os ambientes MU criados por provedores de conteúdo diferentes podem ser vistos em *browsers* MU diferentes.
- **Browser:** qualquer *browser* pode ser usado para ver e interagir com o conteúdo MU de qualquer servidor.

A arquitetura multiusuário do padrão emergente MPEG-4 MU, está amplamente baseada no mecanismo *Pilot/Drone* proposta pela especificação *Living Worlds* [LIVING]. De acordo com a especificação citada, os *Pilots* são cópias mestras de nós em clientes ou em servidores, onde mudanças de estados ou comportamentos são replicadas para outras instâncias, isto é, os *Drones* (réplicas de *Pilots*) que correspondem diretamente aos seus *Pilots*.

Tecnicamente, a forma de identificar objetos compartilhados é colocá-los em uma zona (ou uma região contígua), não necessariamente fixa, de uma sessão multiusuário. Zonas diferentes podem ser usadas para agrupar objetos que podem ser manipulados. Os objetos compartilhados normalmente refletem quaisquer mudanças eventualmente sofridas

para as suas instâncias. E para qualquer mudança de estado ou execução de um comportamento, a instância do *Pilot* emite a mudança, enquanto os *Drones* a reproduzem.

Desta forma, cada objeto compartilhado possui seu *Pilot* que é capaz de originar comportamentos e mudanças de estado. A melhor definição para o par *Pilot/Drone* é sem dúvida o próprio avatar, que é a representação tridimensional de um usuário em um ambiente virtual interativo.

Na Figura 14, temos a representação da sincronização entre o mecanismo *Pilot/Drone*. Para melhor compreensão, os avatares serão representados por  $A_n$  que são controlados por  $P_n$  através das instruções dos usuários diante de cada terminal (*host*), onde cada terminal é capaz de exibir todos os avatares participantes do ambiente.

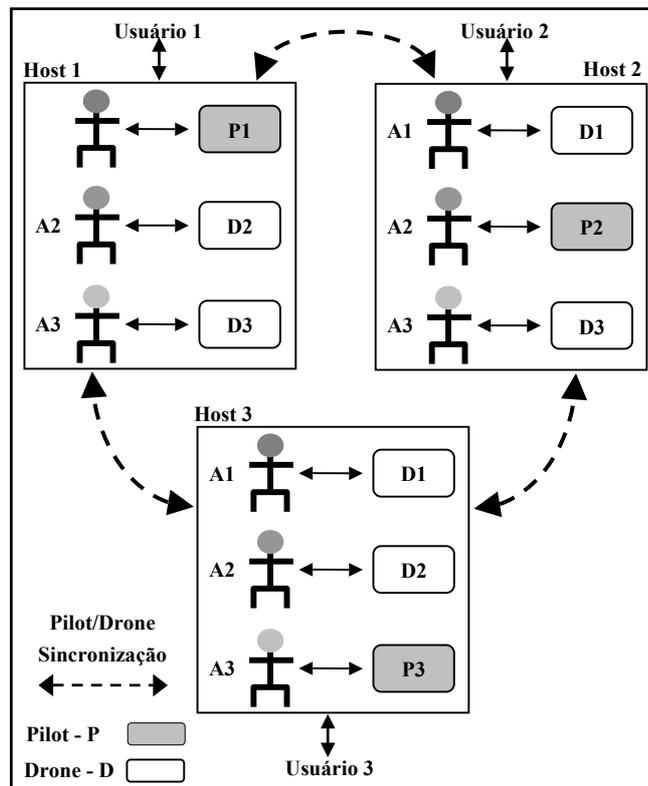


Figura 16 – Mecanismo de Sincronização entre Pilot e Drone [adaptada de N4415].

O retângulo cinza, da Figura 14, representa o *Pilot P* correspondente de cada avatar, o retângulo branco, as réplicas de cada *Pilot*, denominadas *Drones* e as linhas pontilhadas a sincronização entre os correspondentes *Pilots* ( $P_n$ ) e seus *Drones* ( $D_n$ ).

Para a sincronização do mecanismo *Pilot/Drone* há componentes que controlam e gerenciam mensagens de requisição e atualização dos estados/comportamentos dos objetos compartilhados.

### **3.2.1 Componentes da Arquitetura MPEG-4 MU**

Para o gerenciamento e controle do mecanismo *Pilot/Drone*, a arquitetura atual do padrão MPEG-4 foi estendida. A Figura 15 mostra os componentes da arquitetura MPEG-4 MU, especificados pelo padrão e aqueles que são deixados para os fabricantes de navegadores e de sistemas para a web, tais como o Controlador de Sessão *MUTech* (*MUTech Session Controller* - MSC), o *MUTech Bookkeeper* (MBK) e o Manipulador de Mensagem *MUTech* (*MUTech Message Handler* - MMH).

#### ***Controlador de Sessão MUTech - MSC***

O Controlador de Sessão *MUTech* (*MUTech Session Controller* - MSC) faz parte do plano de controle e é responsável pelo gerenciamento de uma sessão multiusuário e sua principal função é proporcionar aos participantes do ambiente a relação das zonas existentes na sessão, a instância do *bookkeeper* associado e o gerenciamento de permissões de entrada/saída de participantes ou adição/remoção de zonas.

#### ***MUTech Bookkeeper - MBK***

O componente *MUTech Bookkeeper* (MBK) faz parte do plano de dados e é o responsável por manter a consistência do ambiente controlando todas as zonas compartilhadas e os objetos compartilhados nela contidos. Seu controle se dá através do gerenciamento das

mensagens de requisições e atualizações BIFS que são distribuídas aos clientes subscritos nas zonas.

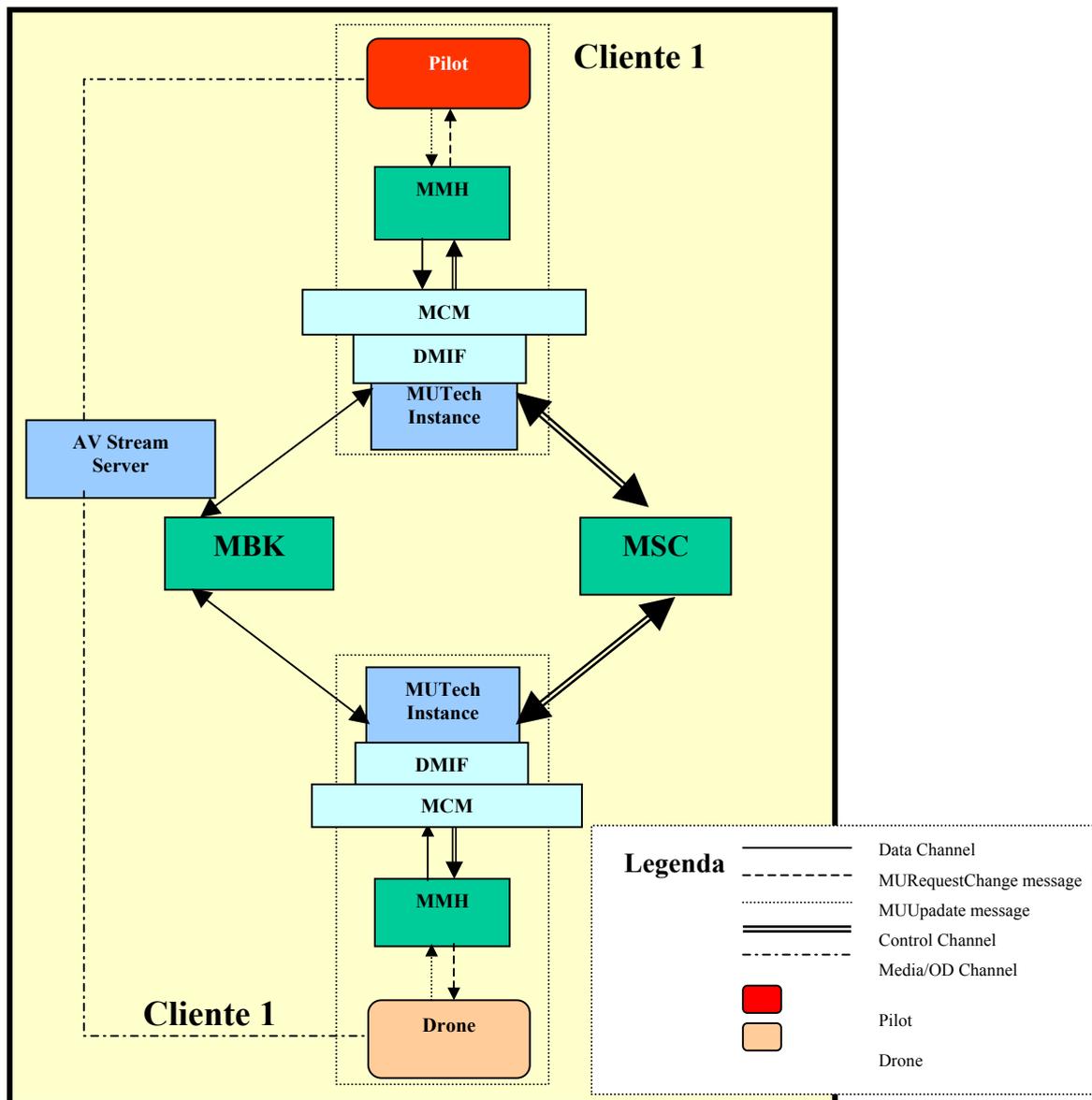


Figura 17 - Arquitetura de Requisição/Atualização [adaptada de N4415].

### *MUTech Message Handler - MMH*

O *MUTech Message Handler* (MMH) responde pelo envio de mensagens de atualização de estado dos objetos compartilhados, provendo uma interface no terminal cliente

para enviar e receber mensagens multiusuário, etiquetando as mensagens com a identificação (*id*) do cliente antes de enviar para o canal de requisição/atualização aberto no DMIF. O MMH expõe uma interface (MAI, Interface de Aplicação de Multiusuário) a qual outros componentes na aplicação poderiam usar, tais como, o EAI (Interface de Autoria Externa) e o MPEG-J, provendo um modo conveniente para tais módulos afetar a cena, e assegurar que as modificações sejam distribuídas.

### ***Gerente de Canal MU - MCM***

O Gerente de Canal MU (*MUChannel Manager* – MCM) é responsável em ajustar os fluxos de controle e dados entre o cliente e o MSC, através da interface DAI do DMIF. O MCM prove uma interface conveniente para o MMH acessar funcionalidades na DAI. Os atributos do canal são descritos usando a sintaxe dos descritores de objetos (*Object Descriptor*) e dos descritores de fluxos elementares (*Elementary Stream Descriptor*), pela inclusão de parâmetros, tais como direção, informação de QoS (Qualidade do Serviço), propriedade intelectual e informação de conteúdo de objetos para ser especificados na configuração do canal.

### ***MUTech Instance***

O componente *MUTech Instance* é um filtro DMIF que permite uma aplicação abrir canais e transferir mensagens respectivas para e do MSC. O filtro concorda com o DPI (*DMIF Plug-in Interface*), e faz a comunicação com o MSC transparente para a aplicação. Em adição o filtro pode suportar a comunicação ponto-a-ponto entre clientes, tal que o MSC e MBK não são usados para rotear o pedido e a atualização de mensagens.

## *AV Stream Server*

O módulo *AV Stream Server* disponibiliza o conteúdo aos clientes, juntamente com os objetos áudio-visuais (AV), apontados através de descritores de objetos. Esta aproximação permite o uso de *unicast* ou *multicast*, arquivos locais etc.

Para melhor exemplificação, quando um participante deseja ingressar no ambiente compartilhado, o MSC é responsável por aceitar ou recusar a entrada do participante. Se aceito, o participante é alocado em uma zona da sessão multiusuário e habilitado para interagir com os demais participantes e com os objetos compartilhados.

Dessa forma, se o participante efetuar alguma modificação de estado do seu avatar (por exemplo, caminhar pelo ambiente) ou de algum objeto compartilhado, esta modificação implicará no disparo de mensagens multiusuário que podem ser de requisições ou atualizações, que são gerenciadas pelo *bookkeeper*.

Assim, quando o participante faz a modificação em um objeto alguns aspectos devem ser tratados:

1. Se o objeto que o participante estiver manipulando for um *Pilot*, então o nó envia uma mensagem de atualização (*UpdateMessage*) para o MBK, o qual passa esta mensagem para os correspondentes nós *Drone* e também atualiza o seu estado.
2. Quando o objeto manipulado pelo participante for um *Drone*, então o nó envia uma mensagem de requisição de mudança (*RequestChange*) para o MBK, o qual passa esta mensagem para o correspondente nó *Pilot*, o qual por sua vez envia uma mensagem de atualização (*UpdateMessage*) para seus *Drones* através do MBK, e o *Pilot* então atualiza o seu estado.

De modo a otimizar a implementação do mecanismo *Pilot/Drone*, nós adicionais foram incluídos na arquitetura do MPEG-4, que geram ações imediatas de atualização do ambiente virtual mediante eventos advindos do usuário e/ou da aplicação.

Estes nós incluem:

- ***MUSession***. Define um nó raiz das partes compartilhadas de uma cena MPEG-4 e a localização do computador controlando a sessão multiusuário definidos por este nó. Ele deve conter uma ou mais zonas compartilhadas obrigatoriamente.
- ***MUZone***. Este nó divide a cena compartilhada em sub-cenas pequenas, que podem ter direitos de acesso diferentes e servidores diferentes controlando a sincronização e atualização dos nós em uma sub-cena de cada zona. Assim, objetos compartilhados ficam relacionados diretamente com suas zonas.
- ***MUAvatar e MUAvatar2D***. São nós de representação de um cliente na aplicação multiusuário e são filhos diretos do nó *MUSession*.

A arquitetura descrita acima apresenta soluções adequadas para estender a atual arquitetura do padrão MPEG-4 para o suporte de ambientes virtuais multiusuário.

### **3.2.2 Transferência de Pilotagem**

A Transferência de Pilotagem é uma forma eficiente de remover temporariamente o atraso de rede que os clientes experimentam quando estão participando em uma sessão multiusuário. Ela pode ser útil e talvez também crucial para alguns tipos de aplicações, por exemplo, desenvolvimento de projetos. Combinando o uso de trancamento (*locking*) e a transferência do *Pilot* podemos também ter alguns benefícios, como o controle

de concorrência. Porém, o conceito de trancamento foi removido do padrão, mas é possível implementar o mecanismo de trancamento como parte da pilotagem de objetos.

O MBK pode ser o responsável pela pilotagem dos objetos compartilhados. Quando isto não acontece, o *Pilot* é um dos terminais-cliente participantes. Os dois casos de pilotagem do objeto são apresentados abaixo.

- **Quando o MBK pilota um objeto**

- Todas as mensagens de requisições para um objeto são processadas pelo MBK;
- O MBK gera mensagens de atualização (*UpdateMessages*) e então as distribui para todos os clientes que compartilham o objeto.

- **Quando o MBK não pilota um objeto**

- O MBK recebe uma mensagem de requisição (*RequestChange*) de um terminal-cliente que não é o *Pilot* e repassa essa requisição para o terminal-cliente que é o *Pilot* do nó objeto.
- O MBK recebe mensagens de atualizações (*UpdateMessages*) do terminal-cliente que pilota o nó e então distribui as mensagens para todos os terminais-cliente que compartilham o objeto.

### **3.2.3 Propagação das mensagens**

A propagação das mensagens entre *Pilots* e *Drones* ocorre de forma diferente dependendo do cenário em que se encontram em um determinado instante de tempo. Os possíveis cenários são descritos e algumas notações são definidas abaixo:

- O círculo amarelo representa um objeto compartilhado em uma *MUZone* com o campo *pilot = true*;
- Os círculos azuis são objetos compartilhados em uma *MUZone* com o campo *pilot = false*, portanto estes objetos são os *Drones* de seus respectivos *Pilots*.

### Cenário 1: *Pilot no Servidor*

1. O cliente 1 interage com o nó (objeto compartilhado) efetuando alguma mudança de estado. Este nó é um *Drone*.
2. O *Drone* então envia uma mensagem de requisição de alteração para o MBK, o qual passa esta mensagem para o correspondente nó *Pilot*.
3. Uma mensagem de atualização então é enviada do *Pilot* para seus Drones através do MBK, e o *Pilot* então se atualiza.

O atraso na rede para esta operação é aproximadamente duas vezes o atraso na rede para ambos os clientes, como ilustra a Figura 16.

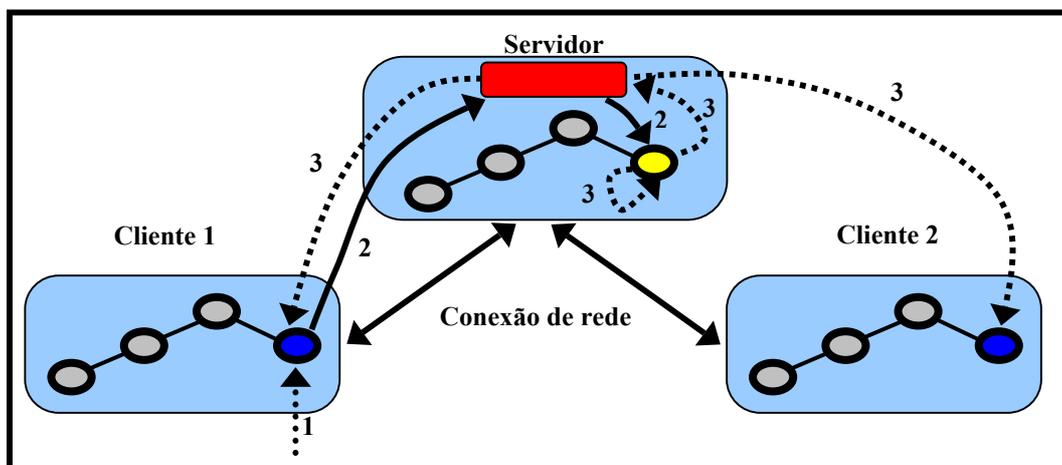


Figura 18 – Pilot no Servidor [adaptada de N4415]

### Cenário 2: *Pilot* no Terminal-Cliente 1

1. O cliente 1 interage com o nó (objeto compartilhado) mudando seu estado. Este nó é um *Pilot*.
2. O *Pilot* então envia uma mensagem de atualização para o MBK, o qual passa esta mensagem para os correspondentes nós *Drone*. Nesse mesmo tempo, o *Pilot* se atualiza também.

O atraso para esta operação é próximo de 0 para o terminal cliente 1, conforme a Figura 17 .

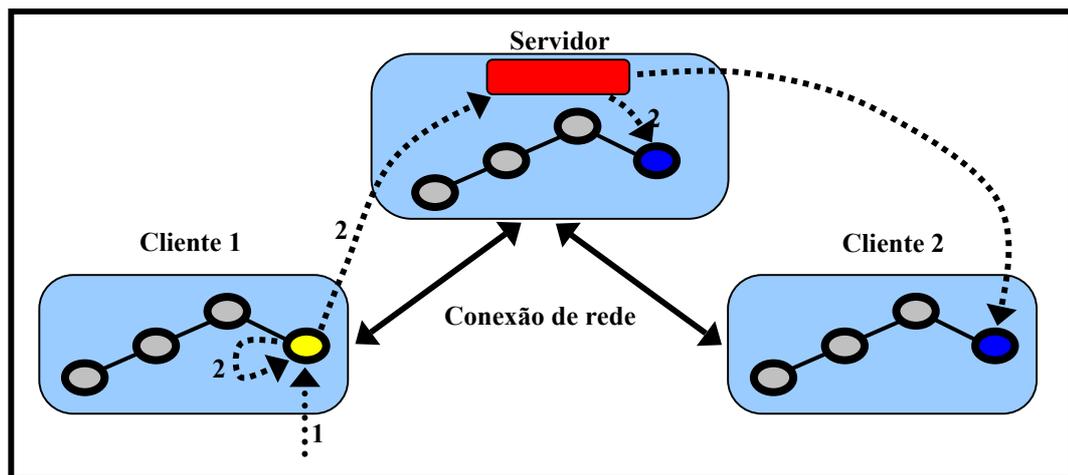


Figura 19 - *Pilot* no Terminal-Cliente 1 [adaptada de N4415].

### Cenário 3: *Pilot* no Terminal Cliente 2

1. O cliente 1 interage com o nó (objeto compartilhado) efetuando uma mudança no estado. Este nó é um *Drone*.
2. O *Drone* então envia uma mensagem de requisição de alteração para o MBK, o qual passa esta mensagem para o correspondente nó *Pilot* (agora no terminal cliente 2).

3. Uma mensagem de atualização então é enviada do *Pilot* para seus *Drones* através do MBK, e o *Pilot* então se atualiza.

No terminal-cliente 1 o atraso para esta operação é aproximadamente quatro vezes o atraso da rede. O terminal-cliente 2 receberá uma atualização com um atraso de cerca de duas vezes o atraso da rede depois de iniciada uma mudança no terminal-cliente 1, ilustrado na Figura 18.

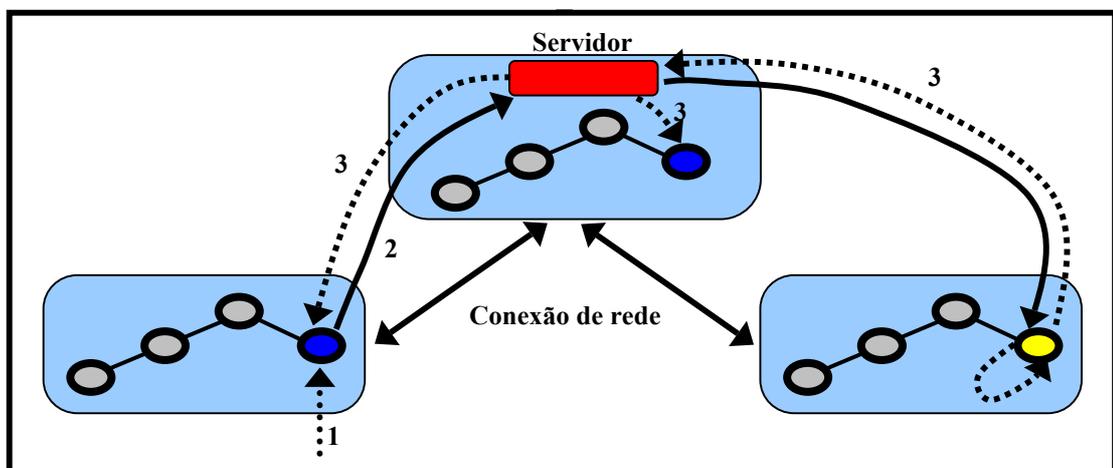


Figura 20 - Pilot no Terminal-Cliente 2 [adaptada de N4415].

As mensagens que são propagadas de acordo com cada cenário discutido acima são fundamentais para a sincronização entre *Pilot* e *Drone* no Ambiente Virtual Multiusuário. No Padrão MPEG-4 MU todas as mensagens são encapsuladas em forma de fluxos que trafegam entre clientes e servidor. Estas mensagens são definidas conforme o fluxo *MUCommandStream* [N4415].

### 3.2.4 MUCommandStream

O *MUCommandStream* é um novo tipo de fluxo definido para transportar mensagens multiusuário (*MU Messages*) entre o servidor e os diferentes terminais-cliente. A conexão entre o servidor e os terminais se faz nos dois sentidos. O *MUCommandStream* consiste dos seguintes elementos: atualizações BIFS (*BIFS-Commands*) encapsuladas,

mensagens de requisições (*Request Messages*), mensagens de confirmação de recebimento (*Acknowledgement Message*) e mensagens de comandos (*Command Messages*).

A Figura 19 apresenta a seqüência de mensagens que são enviadas a partir de um *MUCommandStream*.

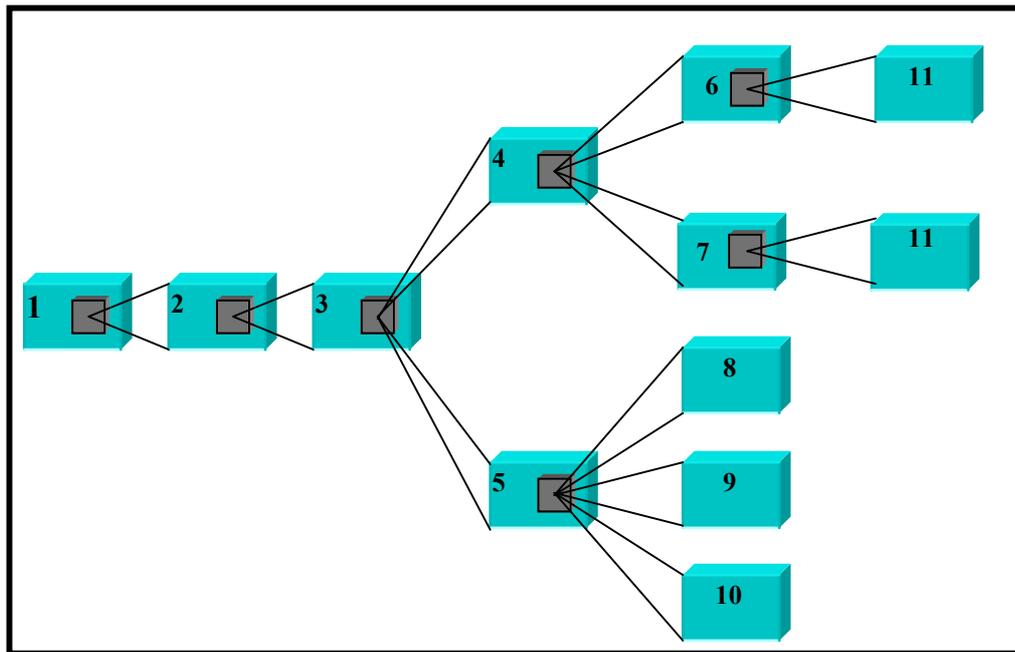


Figura 21 - Hierarquia de mensagens [W4852].

A numeração seguinte corresponde às seguintes mensagens:

1. *MUCommandStream*
2. *MUCommandFrame*
3. *MUMessage*
4. *MUCommandMessage*
5. *MUControlMessage*
6. *MURequestMessages*

7. *MUUpdateMessages*

8. *MUControlRequest*

9. *MUControlAcknowledge*

10. *MUControlCommand*

11. *CommandFrame*

De acordo com a Figura 19, observa-se que há dois tipos principais de mensagens, que são: as mensagens de comando (número 4 da Figura 19) e as mensagens de controle (número 5 da Figura 19). Entretanto, esses dois tipos de mensagens estão relacionados diretamente a um fluxo inicial. Esse fluxo inicial é uma mensagem gerada pelo cliente, podendo ser uma atualização, uma requisição ou uma confirmação de recebimento. Portanto, considera-se o *MUCommandStream* como um fluxo inicial, pois é o primeiro passo a ser encadeado pelas seqüências de mensagens ilustrada na Figura 19.

Para entender como funciona esse fluxo inicial e como ele se relaciona com os demais tipos de mensagens faz-se necessário compreender o funcionamento de cada categoria de mensagem existente. De forma resumida, podemos dizer que uma mensagem multiusuário é gerada da mesma maneira que um comando BIFS. Depois disso, a mensagem é enviada para o *MUCommandFrame* que por sua vez encaminha as mensagens para o *MUMessage*. O *MUMessage* é o responsável por analisar se a mensagem é uma mensagem de atualização de dados (parte do plano de dados, conforme especificação do MPEG-4 MU) ou se é uma mensagem de controle (parte do plano de controle, conforme especificação do MPEG-4 MU). Assim, as mensagens são divididas em *MUCommandMessage* e *MUControlMessage* conforme explicação abaixo:

**Mensagens de comandos (MUCommandMessage):** As mensagens de comando são as responsáveis pela solicitação e atualização de mudanças de estados em objetos compartilhados. Estas mensagens podem ser de requisições (MURequesMessages) ou mensagens de atualizações (MUUpdateMessages). Uma mensagem de requisição é sempre um comando enviado do terminal-cliente para o MBK solicitando mudança de estado de um objeto compartilhado. A mensagem de atualização é um comando que é enviado do MBK para um terminal-cliente mudar o estado de um objeto compartilhado.

**Mensagens de controle (MUControlMessage):** As mensagens de controle são as responsáveis pelo gerenciamento de permissões no Ambiente Virtual Multiusuário. São essas mensagens que gerenciam a entrada de um novo usuário em uma sessão (*Join*), a sua saída (*Leave*), criações de zonas (*CreateZone*) etc no ambiente virtual. Elas estão divididas em mensagens de controle de requisições (*MUControlRequestMessages*), mensagens de respostas (*MUControlAcknowledgeMessages*) e mensagens de controle dos comandos (*MUControlCommandMessages*).

Todas essas mensagens seguem obrigatoriamente uma seqüência que é definida pela especificação multiusuário do padrão MPEG-4 MU [W4852].

### **3.3 Considerações Finais**

Este capítulo descreveu o padrão MPEG-4 no suporte a composição, sintetização, compressão, sincronização e distribuição de objetos áudio-visuais e também o tratamento da descrição das cenas.

O potencial do MPEG-4 consiste na codificação e entrega separada ou integrada de diferentes conteúdos (áudio, vídeo, 2D, 3D, aplicações Java etc) por meio de Fluxos Elementares (*Elementar Streams* – ES) levando à construção de complexas aplicações

multimídia e por definir uma estrutura BIFS (*Binary Format Scene*) que codifica cenas de maneira extremamente compacta e eficiente. O padrão inclui também o protocolo BIFS-*Command* e o fluxo elementar que o transporta para prover as informações BIFS que agem na cena com o passar do tempo.

Este capítulo também apresentou a especificação do MPEG-4 MU como tecnologia emergente no suporte a ambientes virtuais multiusuário. Foram descritos o mecanismo de sincronização, os componentes da arquitetura multiusuário, a transferência de pilotagem, o mecanismo de propagação das mensagens multiusuário e o *MUCommandStream*.

No próximo capítulo apresentamos o projeto de implementação de um modelador 3D colaborativo baseado no Padrão Emergente MPEG-4 MU.



## 4 Projeto e Implementação de um Modelador 3D Colaborativo

Um Modelador 3D Colaborativo, baseado no Padrão Emergente MPEG-4 MU (*MultiUser*), foi especificado e parcialmente implementado como parte deste trabalho de mestrado. O objetivo do projeto é prover um ambiente de colaboração em que múltiplos usuários interagem na edição e/ou modificação simultânea de objetos 3D compartilhados, em aplicações, por exemplo, de prototipagem virtual.

O Modelador 3D Colaborativo em desenvolvimento cria e gerencia um Ambiente Virtual Colaborativo, que permite a criação e edição de cenas gráficas 3D em tempo-real. O modelador garante a efetiva colaboração de todos os membros do grupo para que estes possam criar e/ou modificar os objetos compartilhados, mantendo o ambiente num estado consistente, através de mecanismos de controle de concorrência e de percepção dos usuários e de suas ações.

O Modelador 3D está sendo implementado na linguagem Java e usa a biblioteca Java 3D para a renderização das cenas 3D. Java 3D possui um extenso conjunto de APIs para ser utilizado na criação e manipulação do ambiente virtual.

O modelador possui um modo visual de edição com um universo tridimensional (3D) habilitado a receber objetos áudio-visuais. Com o modo visual de edição, a criação e/ou modificação de uma cena gráfica torna-se muito mais simples, pois os usuários participantes da sessão colaborativa possuem uma simulação do mundo virtual em tempo de edição, e toda a descrição textual do mundo virtual fica transparente para os usuários.

Um conversor de formatos VRML para MPEG-4, desenvolvido pelo *LRVNet*, é utilizado no Modelador 3D com o objetivo de importar projetos existentes no formato VRML para serem editados. Como trabalho futuro pretende-se incorporar um conversor de formatos CAD para MPEG-4, para permitir a modificação de projetos CAD dentro de um ambiente virtual.

Um dos recursos do modelador, sendo ainda desenvolvido, é a possibilidade de um usuário acompanhar a edição colaborativa no papel de observador, em dispositivos que vão do celular a PDAs e PCs, bastando para isso, que o usuário tenha um dispositivo com o *Player* MPEG-4 instalado. Um outro recurso ainda não implementado no modelador, e que será tratado em trabalhos futuros, é a gravação das sessões que poderão ser reproduzidas em diferentes dispositivos, em formato MPEG-4. A reprodução de sessões já decorridas pode ser utilizada como forma de documentação de projetos, bem como de avaliação de pessoal e/ou de metodologia de desenvolvimento de projetos de uma empresa.

A próxima seção descreve, em detalhes, o Modelador 3D Colaborativo e suas funcionalidades.

#### **4.1 Descrição do Modelador 3D Colaborativo**

O Modelador 3D Colaborativo possui uma barra de menus com todas as funcionalidades fornecidas pela ferramenta. A barra é composta pelos menus *World*, *Edit*,

*View, Create, Modifiers, Tools, Project, Session, Zone e Help*. A Figura 20 mostra a interface principal do modelador.

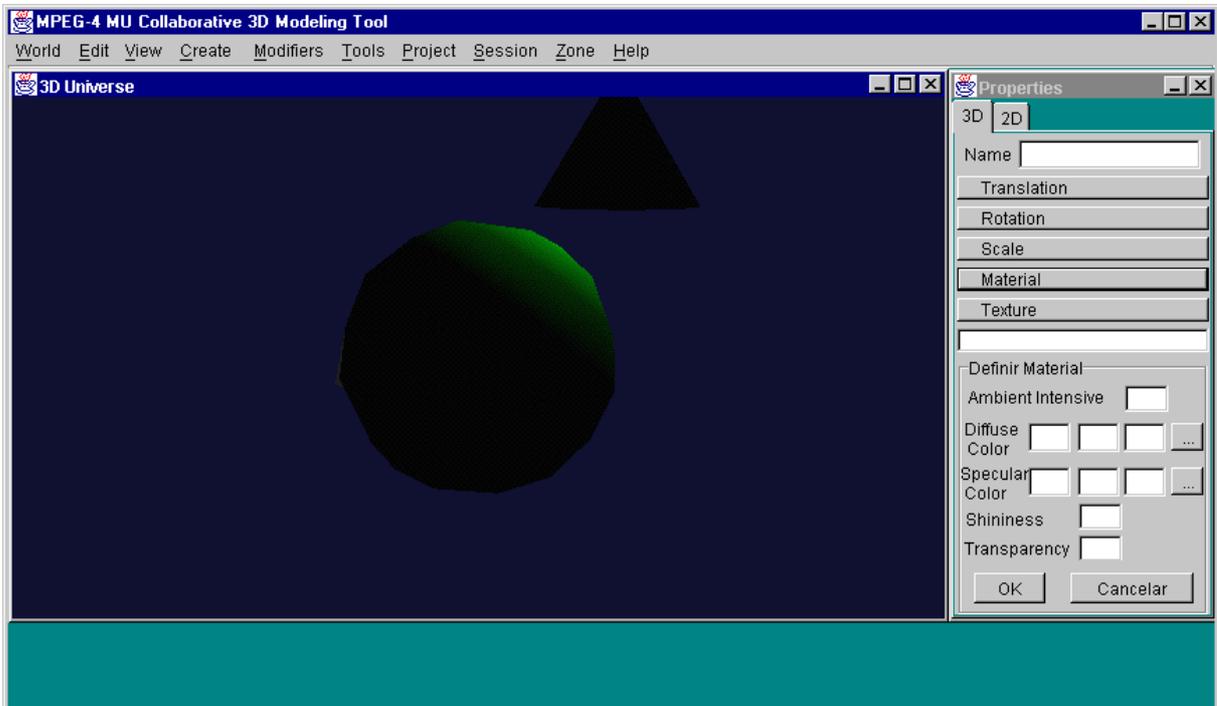


Figura 29 - Modelador 3D Colaborativo.

## Descrição do Menu World

O menu *World* é composto pelos itens *New, Open, Close, Save, Save as, Publish to Server, Import VRML, Export VRML, Properties, File List e Exit*, como mostra a Figura 21.

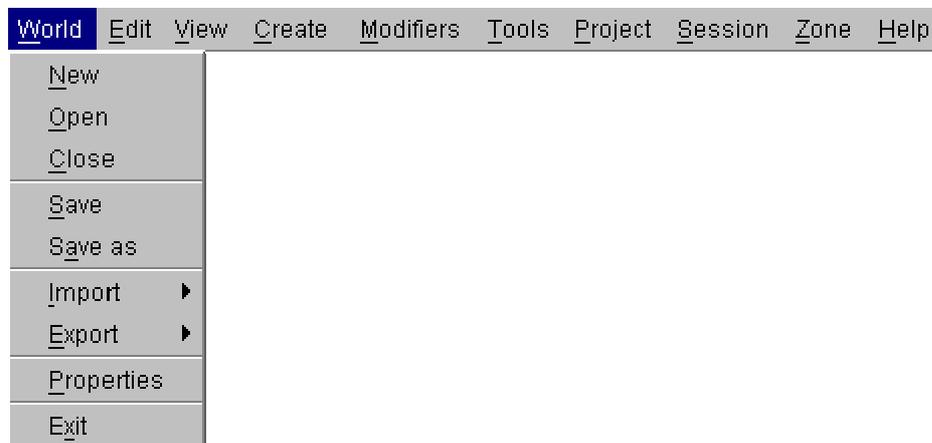


Figura 30 - Menu World.

As funcionalidades dos itens do menu *World* são detalhadas abaixo:

- **New:** cria um novo universo tridimensional em modo visual de edição. Permite também adicionar um novo mundo virtual a um projeto já existente. Além disso, esta opção gera dois arquivos textuais, o primeiro, com extensão SCENE, que contém a descrição textual do mundo virtual, e o segundo, com extensão MUX, que contém informações relacionadas ao tamanho necessário do buffer para decodificar os fluxos de dados, e identificação e controle de fluxos e dados de mídias presentes no mundo virtual. Todas as modificações feitas no modo visual de edição são refletidas nesses arquivos que são utilizados para gerar o arquivo binário do mundo virtual com extensão MP4, o qual pode ser exibido no *Player* MPEG-4;
- **Open:** abre um mundo virtual já existente, buscando a cópia mais recente do mundo virtual em um diretório local ou num servidor de conteúdo. Todo o conteúdo do arquivo SCENE é apresentado no modo visual de edição e pode sofrer alterações por um único usuário ou por múltiplos usuários em uma sessão colaborativa;
- **Close:** fecha um mundo virtual já aberto;
- **Save:** alterações feitas no mundo virtual, em tempo de edição, são salvas em um diretório local. Tais alterações são refletidas nos arquivos SCENE e MUX;
- **Save as:** salva os arquivos abertos com outro nome ou em outro diretório;

- **Publish to Server:** nesta opção os arquivos SCENE e MUX podem ser “publicados”, isto é, disponibilizados em um servidor de conteúdo para que possam ser acessados por todos os usuários em uma sessão colaborativa;
- **Import VRML:** um mundo virtual no formato VRML existente é importado e exibido no modo visual de edição. Este item ativa um conversor de formatos que, a partir de um arquivo VRML, gera os arquivos SCENE e MUX;
- **Export VRML:** exporta um mundo virtual criado no modelador 3D colaborativo no formato MPEG-4 para o formato VRML, realizando o caminho inverso do item anterior;
- **Properties:** apresenta as propriedades gerais de um arquivo SCENE, como nome do arquivo, tipo, localização, tamanho, etc.;
- **File List:** oferece um espaço para exibir uma lista com os nomes dos últimos quatro projetos abertos;
- **Exit:** encerra a execução do modelador 3D colaborativo.

## Descrição do Menu Edit

O menu *Edit* é composto pelas seguintes opções: *Undo*, *Redo*, *Copy*, *Cut*, *Paste*, *Delete* e *Select All* e mostrado na Figura 22.

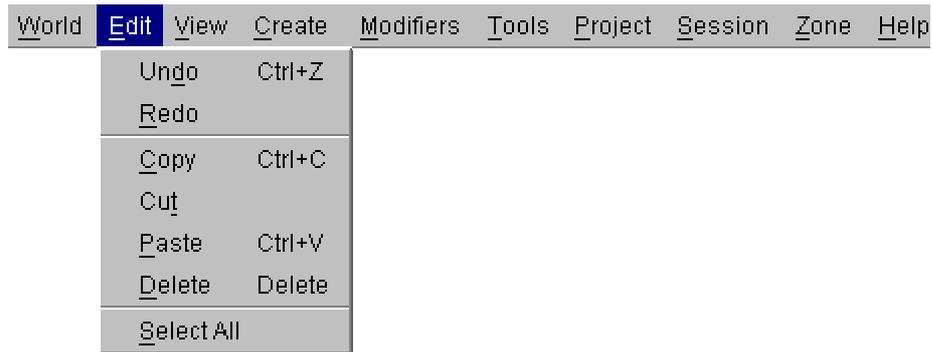


Figura 31 - Menu Edit.

Onde:

- **Undo**: desfaz a última operação realizada;
- **Redo**: refaz a última operação desfeita;
- **Copy**: copia um objeto selecionado;
- **Cut**: recorta um objeto selecionado;
- **Paste**: cola um objeto que foi copiado ou recortado para a área de transferência;
- **Delete**: exclui um objeto selecionado do modo visual de edição e também do grafo da cena;
- **Select All**: seleciona todos os objetos do modo visual de edição.

## Descrição do Menu View

O menu *View* é composto pelos seguintes itens: *Grid*, *Tools Bars*, *Zoom*, *Viewpoint*, *Scene Graph* e *Code Source*. O menu *View* é ilustrado na Figura 23.

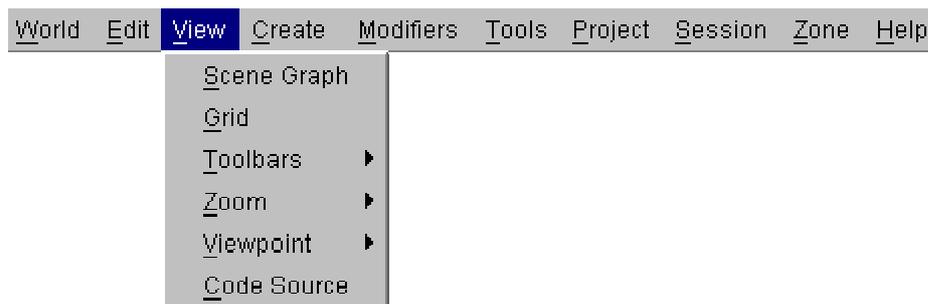


Figura 32 - Menu View.

As funcionalidades do menu *View* são:

- **Grid:** exibe uma malha de quadrantes no modo visual de edição para facilitar a criação e modificação do mundo virtual;
- **Tools Bars:** apresenta todas as barras contendo os botões de acesso rápido disponíveis na ferramenta. Os subitens de **Tools Bars** são: *Standard*, *Edit*, *View*, *Create*, *Session* e *Zone*. O subitem *Standard* exibe os botões de *New World*, *Open World*, *Save*, *Publish to Server*, *Import VRML*, *Export VRML*. O subitem *Edit* exibe os botões *Undo*, *Redo*, *Copy*, *Cut*, *Paste* e *Delete*. O subitem *View* mostra os botões *Grid*, *Zoom*, *Layout*, *Scene Graph* e *Source Code*. O subitem *Create* exibe os botões de criação de objetos mais utilizados, como as primitivas básicas para se criar uma caixa, uma esfera, um cone, etc. O subitem *Session* mostra as opções *Create*, *Star*, *Finish*, *Join*, *Leave*. O subitem *Zone* mostra os botões de *Join*, *Leave Zone* e *Pilotship Request* e *Pilotship Release*.
- **Zoom:** oferece possibilidades de aproximação ou distanciamento da cena gráfica. Os subitens de *Zoom* são: *IN*, *OUT*, *ALL* e *Selected Object*.

- **Viewpoint:** oferece tipos de visualizações do cenário. Os subitens de Viewpoint são Fours Views, Side and Front, Top and Front e Single Front View;
- **Scene Graph:** exibe um menu flutuante com o grafo da cena;
- **Source Code:** apresenta uma janela com o código fonte da descrição da cena.

## Descrição do Menu Create

Uma cena MPEG-4 pode conter objetos 2D e 3D e tais objetos são representados por diferentes nós no grafo da cena. Desta forma, os objetos foram divididos em duas opções para diferenciar a criação desses dois tipos de objetos, embora eles sejam apresentados conjuntamente no modo visual de edição. A Figura 24 mostra o menu *Create*.

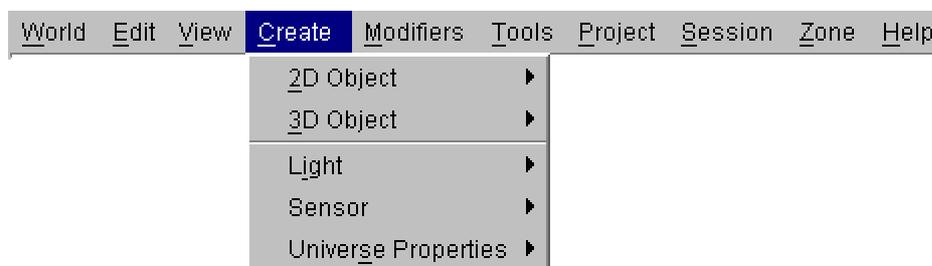


Figura 33 - Menu Create.

O menu *Create* é composto pelos seguintes itens: *Object 2D*, *Object 3D*, *Light*, *Sensor* e *Universe Properties*, cujas funcionalidades são:

- **Object 2D:** permite a criação de objetos 2D no modo visual de edição. Os subitens de *2D Object* são: *Circle*, *Curve2D*, *Rectangle*, *Sound2D*, *PointSet2D*, *Text*, *IndexedFaceSet2D* e *IndexedLineSet2D*;

- **Object 3D:** permite a criação de objetos 3D no modo visual de edição. Os subitens de *3D Object* são: *Box*, *Cone*, *Cylinder*, *Sound*, *Sphere*, *Video*, *IndexedFaceSet* e *IndexedLineSet*;
- **Light:** adiciona luzes ao modo visual de edição. Os subitens de *Light* são *DirectionalLight*, *PointLight* e *SpotLight*;
- **Sensor:** cria sensores no modo visual de edição. Os subitens de *Sensor* são: *Collision*, *Plane*, *Proximity*, *Time* e *Touch*;
- **Universe Properties:** adiciona propriedades no modo visual de edição. Os subitens de *Universe Properties* são: *Anchor*, *Background2D*, *Background*, *Behavior*, *Billboard*, *Fog*, *Group*, *Inline*, *Level of Detail*, *Navigation Info* e *Viewpoint*.

## Descrição do Menu Modifiers

O menu *Modifiers* é composto pelos seguintes itens: *Object Properties* e *Share Object*. O menu *Modifiers* é mostrado na Figura 25.



Figura 34 - Menu Modifiers.

- **Object Properties:** exibe um menu flutuante em que são apresentadas as propriedades dos objetos 2D e 3D e tais propriedades podem ser alteradas. Estas propriedades incluem *Rotation*, *Translation*, *Scale*, *Material* e *Texture*;

- **Share Object:** define se um objeto 3D é compartilhado. Os subitens de *Share Object* são *Define* e *Undefine*.

## Descrição do Menu Tools

O menu *Tools* é composto pelos seguintes itens: *Create MP4*, *Player MPEG-4*, *Chat* e *Manage*. Este menu é ilustrado na figura 26.



Figura 35 - Menu Tools.

As funcionalidades dos itens do menu *Tools* são:

- **Create MP4:** transforma o mundo virtual atual simulado, presente no modo visual de edição, em uma cena BIFS codificada, um arquivo com extensão MP4, a partir dos arquivos com extensão SCENE e MUX. Os arquivos com extensão SCENE e MUX são submetidos ao codificador de BIFS (BifsEnc.exe) e ao multiplexador de cenas (Mux.exe), respectivamente, os quais se encontram no diretório de instalação da ferramenta. Nesses passos, são gerados como saída os arquivos com extensão BIF, LST, OD, TRC e TRIF. Esses arquivos e o arquivo com extensão MUX, são utilizados pelo codificador de MP4 (MP4Enc.exe), que se encontra também no diretório de instalação da ferramenta, gerando como saída uma cena BIFS codificada e representada pelo arquivo com extensão MP4;

- **Player MPEG-4:** ativa o *Player* MPEG-4 para que seja exibida a cena gráfica BIFS codificada;
- **Chat:** ativa uma ferramenta de *Chat* para que os usuários participantes possam trocar informações através de mensagens de texto ou troca de arquivos;
- **Manage:** este item provê funções para que um usuário habilitado possa gerenciar grupos de usuários, usuários e papéis sociais. Desta forma, usuários particulares podem ser adicionados ou removidos de grupos de usuários particulares. A criação de novos usuários ou a remoção de usuários. Também podem ser criados papéis sociais, que os usuários podem desempenhar em uma sessão colaborativa, e atribuição de permissões de acesso.

## Descrição do Menu Project

O menu *Project* é composto pelos itens *New*, *Open*, *Close*, *Save*, *Save as* e *Properties*. O menu *Project* é ilustrado na Figura 27.

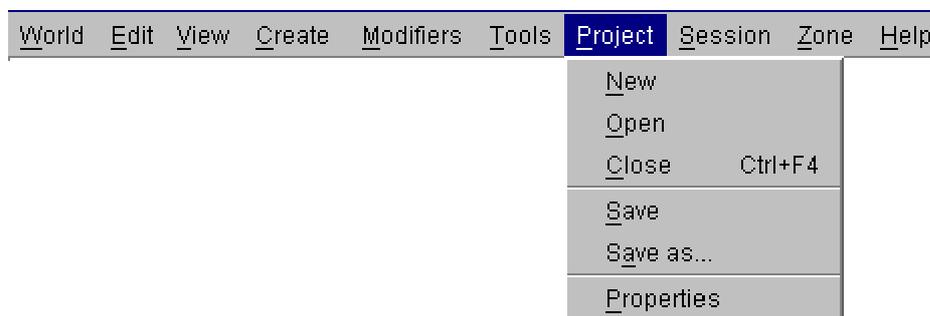


Figura 36 - Menu Project.

As funcionalidades dos itens do menu *Project* são detalhadas abaixo:

- **New:** cria um novo projeto permitindo adicionar/remover grupos de usuários ao/do projeto. Além disso, mundos virtuais já existentes podem ser adicionados ao novo projeto;
- **Open:** abre um projeto já existente e permite efetuar alterações no projeto, tais como, adicionar ou remover grupos de usuários e adicionar ou remover mundos virtuais;
- **Close:** fecha um projeto já aberto;
- **Save:** salva as alterações feitas no projeto;
- **Save as:** salva o projeto aberto com outro nome ou em outro diretório;
- **Properties:** apresenta as propriedades gerais do projeto, como nome do projeto, localização, tamanho, etc.

## Descrição do Menu Session

O menu *Session* é composto pelos seguintes itens: *New*, *Start*, *Finish*, *Join* e *Leave*. Este item está relacionado com a criação e gerenciamento de sessões e é onde se concentra os maiores esforços de implementação feitos neste trabalho de mestrado. O menu *Session* é ilustrado na Figura 28.

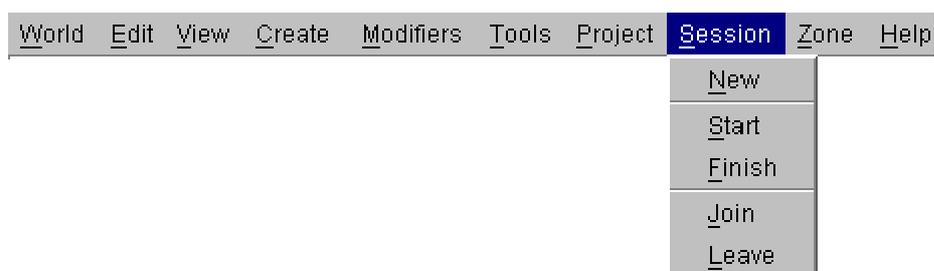


Figura 37 - Menu Session.

- **New:** cria e configura uma nova sessão colaborativa. Grupos de usuários, que pertencem ao projeto, podem ser adicionados para participar da sessão. Papéis sociais e permissões de acesso podem também ser atribuídos aos membros participantes antes do estabelecimento da sessão.
- **Start:** inicializa uma sessão com a execução do componente MSC (MUTech Session Controller), responsável pelo gerenciamento da sessão.
- **Finish:** encerra uma sessão com o término da execução do componente MSC.
- **Join:** solicita a entrada de usuários em uma sessão colaborativa.
- **Leave:** promove a saída de usuários de uma sessão colaborativa.

## Descrição do Menu Zone

O menu *Zone* é composto pelos seguintes itens: *Join*, *Leave*, *Leave All* e *Pilotship*, ilustrados na figura 29.



Figura 38 - Menu Zone.

As funcionalidades do menu *Zone* são descritas abaixo:

- **Join:** solicita a entrada de usuários em uma zona compartilhada específica da sessão colaborativa, apresentando a relação de zonas existentes.

- **Leave:** promove a saída de usuários da zona em que se encontram na sessão colaborativa.
- **Leave All:** promove a saída de usuários de todas as zonas em que se encontram na sessão colaborativa.
- **Pilotship:** atribui permissão para efetuar modificações na zona compartilhada. É composto pelos subitens *Request* e *Release*. O subitem *Request* requisita a “pilotagem” de uma zona compartilhada. O subitem *Release* libera a pilotagem de uma zona compartilhada.

## Descrição do Menu Help

O menu *Help* provê um manual de utilização da ferramenta, bem como informações sobre a ferramenta.

O projeto completo do modelador 3D colaborativo está descrito no Anexo A, onde são apresentados detalhes das interfaces das principais funcionalidades descritas neste capítulo.

Dos itens apresentados nas seções anteriores, as funcionalidades do menu *World*, *Create*, *Project*, *Session* e *Zone* já foram implementados. Este trabalho concentrou-se principalmente na implementação de parte dos componentes de controle de sessão e de atualização de cena (MSC e MBK, respectivamente), descritos no Anexo B, bem como dos mecanismos de suporte a colaboração envolvendo controle de concorrência, trancamento de objetos compartilhados, manutenção de estado do ambiente colaborativo e percepção de presença de usuários no AVC. Nas próximas seções será descrita a implementação da estrutura de suporte aos requisitos dos Ambientes Virtuais Colaborativos para a modelagem 3D.

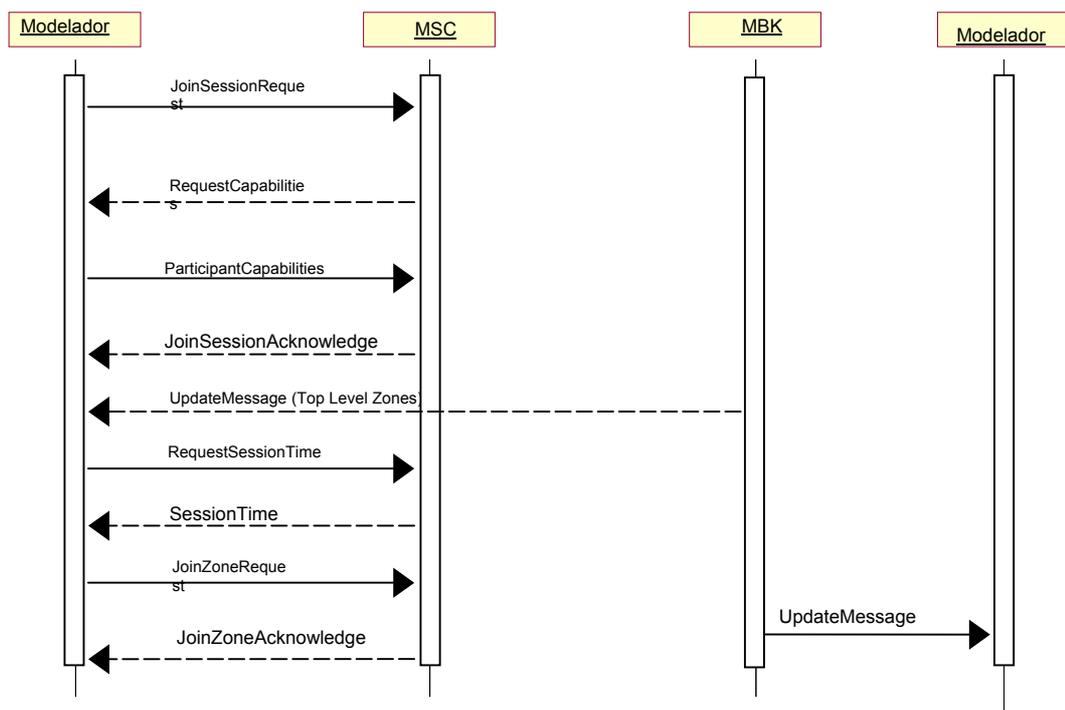
## 4.2 Gerenciamento de Sessão

O gerenciamento de sessão multiusuário no Modelador 3D Colaborativo é realizado pelo componente MSC (*MUTech Session Controller*), implementado em conjunto com outros dois alunos: um de mestrado, que também usa o MSC para controle de sessão em jogos colaborativos, e um aluno de iniciação científica. Os componentes MSC e MBK foram definidos e especificados pelo padrão emergente MPEG-4 MU. Uma sessão colaborativa tem início com o estabelecimento de uma conexão entre o modelador e o componente MSC, que é responsável pelo gerenciamento da sessão, e realiza as seguintes tarefas:

- Provê a relação das zonas existentes na sessão e informa sobre os demais usuários participantes de uma zona particular;
- Inicializa o componente MBK associado (uma cópia do grafo da cena é enviada para o MBK pelo MSC);
- Adiciona ou remove zonas em uma sessão e;
- Gerencia a entrada/saída de usuários participantes da sessão.

Quando um participante deseja ingressar no ambiente compartilhado, o MSC é responsável por aceitar ou recusar a sua entrada, de acordo com critérios estabelecidos pela política de colaboração implementada no MSC. Se aceito, o participante é alocado para uma zona da sessão colaborativa e habilitado para interagir com os demais participantes e com os objetos compartilhados.

Para o participante entrar na sessão, é enviada uma mensagem de solicitação de entrada do cliente em uma sessão (*JoinSessionRequest*). Essa mensagem é disparada pelo terminal-cliente (modelador) quando o participante seleciona a sessão a qual quer associar-se. Assim, após essa mensagem ser interpretada pelo MSC, uma resposta é enviada ao terminal-cliente do usuário. Essa resposta (*JoinSessionAcknowledge*) é confirmada de forma positiva ou negativa. Se a resposta for positiva, uma mensagem de comando, *UpdateMessage* é disparada para que todos os participantes fiquem cientes da entrada do novo participante. Nesse comando, a identificação do terminal-cliente é atribuída a ele e as zonas contidas no mundo virtual são fornecidas pelo MBK para que o participante possa escolher uma determinada zona para se associar. O participante seleciona uma zona e envia uma mensagem de solicitação para entrada na zona (*JoinZoneRequest*). Caso a mensagem seja positiva, o MBK distribui uma mensagem de atualização (*UpdateMessage*) para todos os participantes da zona compartilhada. As ações que compreendem a entrada de um cliente numa sessão do modelador 3D colaborativo MPEG-4 MU, são ilustradas no diagrama da Figura 30.



**Figura 39 – Diagrama da entrada de usuários na sessão colaborativa.**

Quando um participante deseja deixar a sessão colaborativa, inicialmente ele deve sair das zonas as quais pertence. Desta forma, uma mensagem de solicitação de saída de todas as zonas é disparada, através do menu *Zone/LeaveAll* (*LeaveAllZoneRequest*). A mensagem de solicitação é tratada pelo MSC, o qual retorna uma resposta ao terminal-cliente (*LeftAllZones*). Depois que o participante saiu de todas as zonas, ele envia uma mensagem de solicitação de saída da sessão (*LeaveSessionRequest*). A mensagem de solicitação é tratada pelo MSC, o qual retorna uma resposta ao terminal-cliente (*LeftSession*) e uma mensagem de atualização é enviada para o MBK para que este propague a notificação de saída do participante para os demais participantes da sessão colaborativa. O diagrama da Figura 31 ilustra a saída de um cliente de uma sessão.

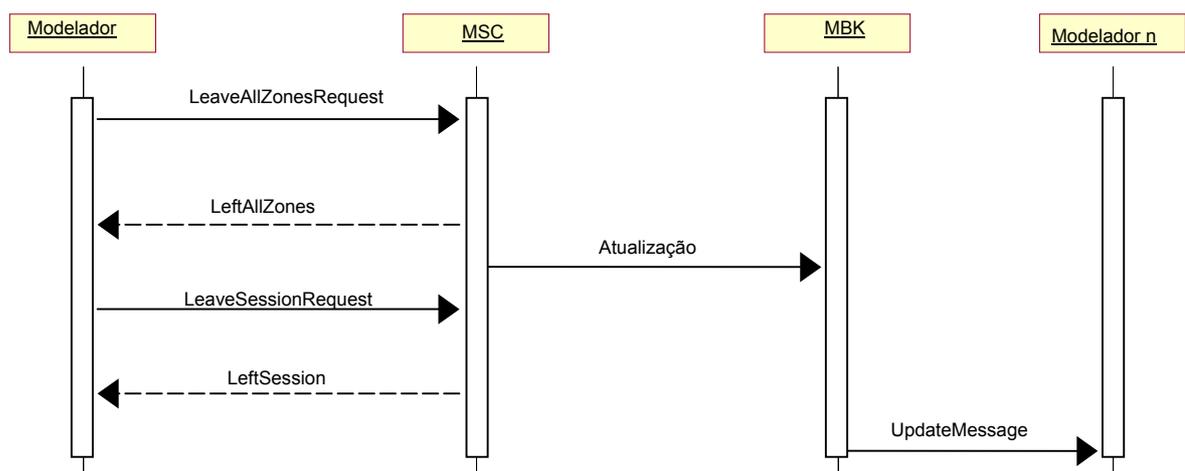


Figura 40 - Diagrama de saída de usuário da sessão.

### 4.3 Manutenção da Consistência

A manutenção da consistência do modo visual de edição do Modelador 3D Colaborativo é suportada pelo mecanismo *Pilot/Drone* e pelo protocolo *BIFS-Command*. No modelador 3D, todo objeto compartilhado possui seu *Pilot* correspondente, que é capaz de originar comportamentos e mudanças de estado.

Quando uma modificação é efetuada em um nó de objeto compartilhado, uma ação é disparada: o *Pilot* do objeto modificado propaga a modificação para os seus

correspondentes *Drones*. As mensagens de atualização (*BIFS-Commands*) são gerenciadas pelo componente MBK (*MUTech Bookkeeper*), também implementado como parte deste trabalho, com a colaboração de um segundo aluno de iniciação científica. O MBK controla todas as zonas e seus objetos compartilhados, fazendo assim a propagação das mensagens para os terminais-cliente subscritos nas zonas. Desta forma, o componente MBK é responsável pelas seguintes tarefas:

- Gerar identificadores (IDs) únicos dos nós da árvore de cena para os objetos compartilhados;
- Propagar as mensagens de *Pilots* para *Drones* subscritos nas zonas;
- Pilotar os objetos compartilhados de uma zona ou atribuir a pilotagem para terminais-cliente e;
- Manter uma cópia atualizada dos dados de um grafo completo da cena contendo todas as zonas e nós de uma sessão.

A Figura 32 mostra o digrama do envio da modificação pelo *Pilot* da zona para o MBK e a propagação para os demais participantes pelo MBK.

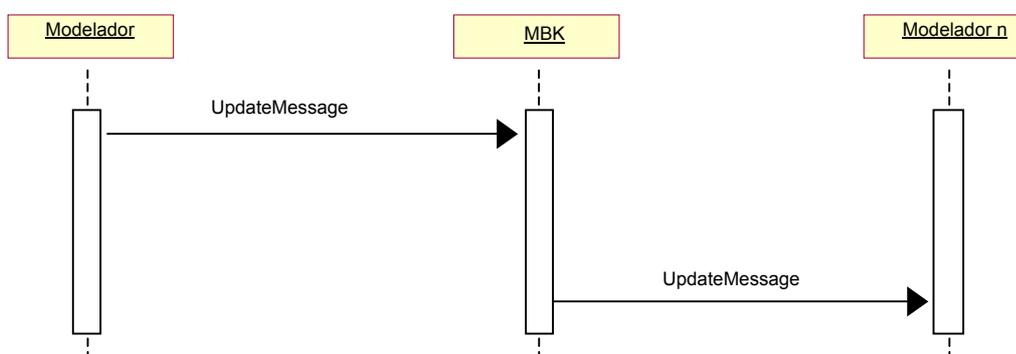


Figura 41 - Diagrama da modificação da cena gráfica.

O mecanismo implementado para a manutenção da consistência da cena gráfica entre os usuários participantes da sessão colaborativa de modelagem 3D é apresentado na Figura 33. Pode ser observado que a modificação do estado do objeto compartilhado é codificada pelo codificador BIFS (BIFS-Encoder), que permite a geração de um BIFS-Command. O BIFS-Command é encapsulado em um MU-Command e transmitido para o componente MBK, que propaga a modificação para os correspondentes terminais-cliente.

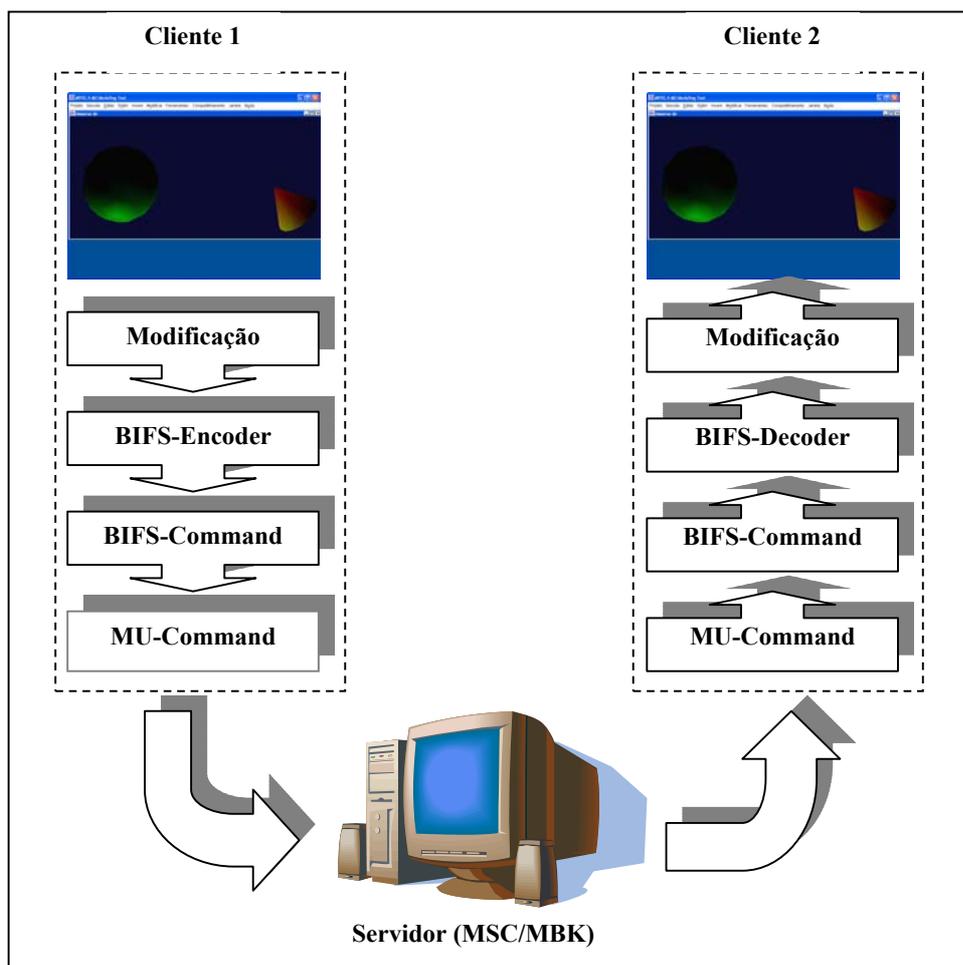


Figura 42 - Mecanismo de atualização da cena.

O uso do protocolo BIFS-Command como parte do mecanismo de manutenção da consistência do ambiente é relatado na literatura como um mecanismo eficiente para reduzir o tráfego de dados na rede, reduzindo assim o atraso na atualização da cena gráfica entre os terminais-cliente [HOSS01] [HOSS02]. Por exemplo, considerando a modificação

do campo *diffuseColor* de um nó *Material* são necessários 120 bits ou 15 bytes, que são usados da seguinte forma:

- 1 bit para identificar o tipo de mensagem multiusuário;
- 1 bit para o Id do comando;
- 4 bits para o Id da zona;
- 4 bits para o tipo e categoria do BIFS-Command;
- 10 bits para o Id do nó;
- 3 bits para o Id do campo;
- 96 bits para os novos valores R, G e B e;
- 1 bit para sinalizar continuidade.

Considerando o tamanho da mensagem de atualização e a frequência com que esta mensagem pode ser transmitida em uma sessão de modelagem colaborativa, o tráfego da rede não é o maior obstáculo para a propagação imediata das modificações da cena entre os terminais-cliente. O maior atraso provavelmente ocorre na troca de informações entre os terminais-cliente e os componentes MSC e MBK, além do processamento da mensagem e atualização da cena nos terminais-cliente remotos.

#### 4.4 Controle de Concorrência

O controle da concorrência é implementado no Modelador 3D Colaborativo, através da integração de dois mecanismos: *Pilot/Drone*, especificado pelo padrão MPEG-4 MU e descrito na seção 3.2, e mecanismo de trancamento (*locking*), removido do padrão

MPEG-4 MU. Entretanto, o mecanismo de trancamento foi implementado como parte deste trabalho para suporte ao controle de concorrência. O mecanismo de trancamento permite que somente o usuário participante que detém a pilotagem do objeto compartilhado da cena pode modificá-lo. Desta forma, para que um usuário possa efetuar uma modificação em um objeto da cena, como descrito na seção anterior, ele deve solicitar a pilotagem do objeto compartilhado enviando uma mensagem de solicitação de transferência de pilotagem ao componente MBK (*TransferPilotshipRequest*). O componente MBK localiza o *Pilot* do objeto e envia a solicitação de transferência de pilotagem para este terminal, que responde com uma mensagem de reconhecimento (*TransferPilotshipAcknowledge*). O *Pilot* do objeto compartilhado atende múltiplas solicitações através do modelo FCFS - *first come first served* e pode responder a esta solicitação aceitando ou negando a transferência de pilotagem (*AcceptZonePilotship*). A Figura 34 mostra o diagrama da troca de mensagens para a transferência de pilotagem.

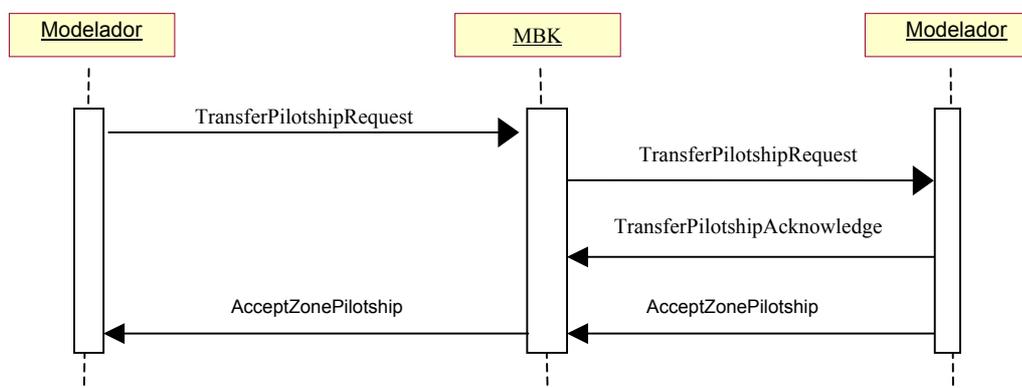


Figura 44 - Diagrama da transferência de Pilotagem.

A aceitação ou negação é baseada nas políticas de direito de acesso estabelecidas na criação da sessão pela atribuição de permissões aos papéis sociais que o usuário participante desempenha durante a sessão. Somente após conseguir a posse do objeto é que o usuário pode efetuar as modificações, desta forma, somente um único usuário pode editar um objeto compartilhado em determinado instante de tempo.

A arquitetura de comunicação entre o modelador e os componentes MSC e MBK é ilustrada na Figura 35.

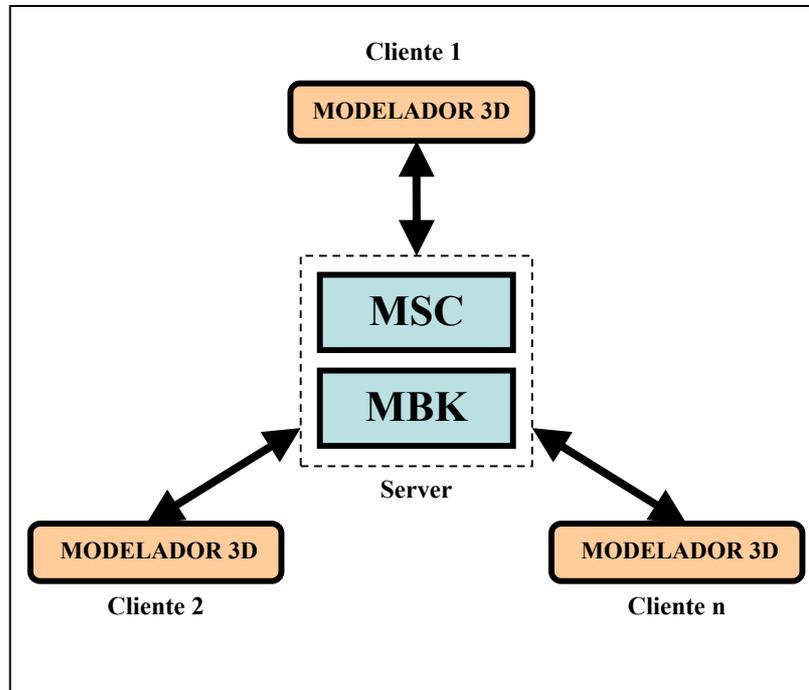


Figura 45 - Arquitetura de comunicação entre o Modelador 3D Colaborativo e os componentes MSC e MBK.

## 4.5 Controle de Acesso

O controle de acesso no modelador 3D colaborativo é realizado pela atribuição de direitos de acesso ou permissões para os papéis sociais que os membros participantes de uma sessão podem exercer. Os direitos de acesso são divididos em:

- **Direitos de Sessão:** incluem as permissões de iniciar ou terminar uma sessão. Desta forma, somente os membros participantes de uma sessão que têm esta permissão podem iniciar ou terminar a sessão de modelagem colaborativa.
- **Direitos de Interação:** estes direitos incluem as permissões de leitura, escrita e modificação dos objetos compartilhados da cena gráfica. Estes

direitos são verificados quando o participante interage com modo visual de edição e é atribuída a pilotagem da zona somente para os participantes que tem a permissão de modificar a cena gráfica.

- **Direitos de Serviços:** incluem as permissões para utilizar ferramentas de áudio, vídeo e chat e também a impressão da cena gráfica. Assim, somente os participantes habilitados podem utilizar estes serviços.

Quando um usuário participante tentar executar uma ação o sistema deve verificar se o usuário pode ou não executar a ação e aceita ou rejeita a sua execução. A Tabela 3 mostra um exemplo de atribuição de papéis sociais aos usuários participantes de uma sessão colaborativa.

Tabela 4 - Exemplo de atribuição de direitos de acesso aos papéis sociais dos usuários.

Direitos \ Papéis	Gerente	Chefe	Colaboradores
Sessão	Iniciar/Terminar	Terminar	—————
Interação	Leitura/Escrita/ Modificação	Leitura/Modificação	Leitura
Serviços	Áudio/Vídeo/Chat	Áudio/Chat	Chat

## 4.6 Consciência do Usuário

O modelador 3D colaborativo provê informações de percepção da presença e das ações dos usuários, através de uma janela de estado. As informações de entrada e saída de usuários são passadas para o modelador, através do componente MSC que mantém as informações de todos os usuários presentes na sessão e em cada zona particular, como ilustra a Figura 36.

As informações das ações dos usuários quem detêm a pilotagem da zona compartilhada são fornecidas para o modelador pelo componente MBK. O componente MBK,

como descrito na seção 4.3, propaga as mensagens de atualização somente para os participantes subscritos em uma determinada zona.

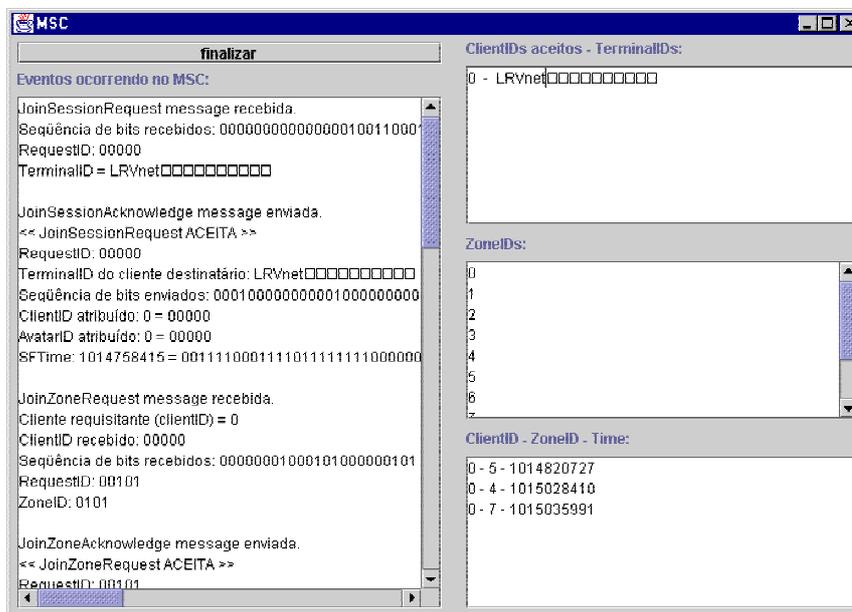


Figura 46 - Interface do componente MSC.

As informações de interação dos usuários com o modo visual de edição do modelador são transmitidas somente para os usuários que realmente necessitam estar cientes dessas alterações. A percepção do usuário se dá através do uso de ponteiros 3D que identificam o usuário na sessão colaborativa e são selecionados quando o usuário entra na sessão. Assim, pode-se prover a percepção do usuário sem ter que enviar alterações na forma geométrica, como é o caso de utilizar avatares realísticos. Como trabalho futuro pretende-se possibilitar que o usuário mude o seu ponteiro 3D durante o decorrer da sessão.

Desta forma, o modelador 3D provê as informações de percepção para todos os usuários participantes da sessão colaborativa para que estes estejam cientes da presença e das ações dos demais usuários, sem sobrecarregar o uso dos recursos do sistema, como largura de banda de rede e poder de processamento.

## 4.7 Persistência

O suporte a persistência no modelador 3D colaborativo é realizado pelo armazenamento da cena gráfica do modo visual de edição nos arquivos SCENE e MUX. Desta forma, a persistência de estado é garantida para recuperação posterior e os usuários podem trabalhar a partir do que já fizeram.

Uma proposta futura é a implementação de um mecanismo que armazena todas as interações que os usuários participantes realizaram na sessão colaborativa através de um mecanismo de rastreamento. Com este mecanismo de persistência contínua as tarefas realizadas em uma sessão poderiam ser recuperadas e visualizadas novamente.

## 4.8 Considerações Finais

Este capítulo descreveu o projeto de um modelador 3D colaborativo baseado no padrão emergente MPEG-4 multiusuário. Com este modelador, cenas gráficas 3D no formato MPEG-4 podem ser criadas ou modificadas simultaneamente por múltiplos usuários. As vantagens deste modelador 3D baseado no padrão de MPEG-4 MU são: a atualização da cena é mais eficaz, pois as modificações são codificadas em um formato binário (*BIFS-Commands*); uma solução padronizada é usada para cumprir os requisitos complexos e desafiadores dos ambientes virtuais colaborativos e; múltiplos usuários podem fazer parte de uma sessão de projeto usando o modelador enquanto outros usuários podem agir como observadores usando um terminal MPEG-4 ubíquo.

## 5 Conclusão

Este trabalho descreve os principais requisitos do desenvolvimento de Ambientes Virtuais Colaborativos, destacando os requisitos do suporte a modelagem 3D colaborativa e apresentou a implementação de um modelador 3D colaborativo baseado no padrão MPEG-4 MU. As vantagens do modelador 3D incluem:

- A atualização da cena gráfica é realizada de forma mais eficiente, pois as modificações são codificadas em um formato binário reduzindo o tráfego da rede (através de *BIFS-Commands*).
- Uma solução padronizada é usada para satisfazer requisitos complexos e desafiadores dos ambientes virtuais colaborativos, tais como manutenção da consistência, controle de concorrência e gerenciamento de sessão multiusuário.
- Múltiplos usuários podem participar da sessão colaborativa usando o modelador para criação e edição de cenas gráficas 3D em tempo-real, enquanto outros usuários podem agir como observadores usando um terminal MPEG-4 em dispositivos ubíquos.

A seguir apresentamos as contribuições geradas e trabalhos futuros deste trabalho.

## 5.1 Contribuições Geradas

O trabalho apresentado procurou contribuir com as pesquisas realizadas pelo *LRVNet* na avaliação do padrão MPEG-4 como um padrão viável no suporte a aplicações multiusuários. Tais contribuições são apresentadas abaixo.

- Especificação e implementação de um modelador 3D colaborativo baseado no padrão emergente MPEG-4 MU para prover um ambiente de colaboração em que múltiplos usuários interagem na edição e/ou modificação simultânea de objetos 3D compartilhados;
- Implementação dos componentes MSC (*MUTech Session Controller*) e MBK (*MUTech Bookkeeper*) em conjunto com outros três alunos: um de mestrado, que também usa o MSC para controle de sessão em jogos colaborativos, e dois de iniciação científica. Os componentes MSC e MBK são utilizados para suporte aos requisitos dos ambientes virtuais colaborativos, tais como gerenciamento de sessão, controle de concorrência e manutenção da consistência;
- Implementação do mecanismo de sincronização da cena gráfica com a integração de dois mecanismos: *Pilot/Drone*, que faz a sincronização das réplicas dos objetos compartilhados, e protocolo *BIFS-Command*, que transporta os comandos BIFS que agem diretamente na cena gráfica. Este mecanismo permite que as modificações realizadas na cena gráfica sejam refletidas para todos os usuários participantes da sessão colaborativa

utilizando o modelador 3D ou visualizadas em terminais MPEG-4, sendo ainda desenvolvido, através do *Player* MPEG-4.

As publicações obtidas no decorrer deste trabalho foram (Anexo C):

- Duarte, F. V., Verdi, M., Araujo, R. B. How to Support Collaborative Modelling with the Emerging Standard MPEG-4 MU. Proceedings of XV Brazilian Symposium on Computer Graphics and Image Processing, Fortaleza, Brazil, October, 2002.
- Duarte, F. V., Araujo, R. B. A 3D Collaborative Modeler Based on the Emerging MPEG-4 MU Standard. Proceedings of XVI Brazilian Symposium on Computer Graphics and Image Processing, São Carlos, Brazil, October, 2003. (aceito)
- Duarte, F. V., Silva, A. R., Araujo, R. B. A Collaborative Virtual for 3D Modelling Based on the MPEG-4 MU Emerging Standard. The 10th International Multimedia Modelling Conference (MMM2004), Brisbane, Australia, January, 2004. (submetido)
- Duarte, F. V., Araujo, R. B. Um Ambiente Colaborativo para Modelagem 3D Baseado no Padrão Emergente MPEG-4 MU. WEBMÍDIA 2003, Salvador, Brasil, Novembro, 2003. (submetido)

## 5.2 Trabalhos Futuros

Como continuidade ao trabalho iniciado, as seguintes atividades ainda deverão ser realizadas:

1. Implementação completa de todas as funcionalidades do modelador 3D colaborativo, tais como as funcionalidades do menu *View* (barras de ferramentas com botões de acesso rápido), as funcionalidades do menu *Edit* (*Undo*, *Redo*, *Copy*, etc.), bem como os menus flutuantes;
2. Implementação de um mecanismo de rastreamento e gravação das tarefas realizadas nas sessões colaborativas, que poderão ser reproduzidas em diferentes dispositivos, em formato MPEG-4. A reprodução de sessões já decorridas pode ser utilizada como forma de documentação de projetos, bem como de avaliação de pessoal e/ou de metodologia de desenvolvimento de projetos de uma empresa;
3. Implementação de modificações no mecanismo de percepção dos usuários, para que estes possam mudar o seu ponteiro 3D durante o decorrer da sessão colaborativa;
4. Medição do tempo de atraso para a sincronização da cena entre os terminais-cliente participantes da sessão colaborativa;
5. Medição do tempo de codificação e decodificação dos comandos de modificação da cena gráfica (*BIFS-Commands*);
6. Inserção dos novos nós da especificação *Animation Framework eXtension* (AFX) do padrão MPEG-4, tais como curvas e superfícies NURBS (*Non-Uniform Rational B-Splines*) no modelador 3D;
7. Integração do modelador 3D com o componente DMIF (*Delivery Multimedia Integration Framework*) do MPEG-4;

8. Construção de um conversor de formato de arquivo CAD para MPEG-4, para permitir que projetos desenvolvidos em CAD possam ser modificados no modelador 3D.

### **5.3 Considerações Finais**

O padrão MPEG-4 é uma tecnologia promissora com um amplo potencial para a construção de ambientes virtuais multiusuário através da padrão emergente MPEG-4 MU. O padrão emergente MPEG-4 MU é uma extensão do padrão MPEG-4 existente para suportar uma ampla gama de aplicações multiusuário, tal como os Ambientes Virtuais Colaborativos. Desta forma, o MPEG-4 MU especifica componentes que dão suporte aos requisitos dos AVCs, tais como gerenciamento de sessão, manutenção da consistência do ambiente compartilhado, percepção dos usuários, comunicação, entre outros.

Este trabalho teve como objetivo principal a implementação de um ambiente de suporte a modelagem colaborativa em tempo-real. Desta forma, o modelador 3D colaborativo baseado no padrão MPEG-4 MU e os componentes MSC e MBK foram implementados como parte deste trabalho para facilitar a criação ou modificação de cenas gráficas, em formato MPEG-4, por múltiplos usuários.

## Referências Bibliográficas

- [ABRE01] Abreu, S. **A Tecnologia MPEG-4 no Suporte a Ambientes Virtuais Multiusuário.** Dissertação de Mestrado, PPG-CC, UFSCar, São Carlos, Agosto, 2001.
- [ANTU01] Antunes, M., Silva, A. R., Martins, J. **An Abstraction for Awareness Management in Collaborative Virtual Environments.** VRST'01, Banff, Alberta, Canada, November, 2001.
- [ANTU01a] Antunes, M., Silva, A. R., Martins, J. **A Flexible Design Solution for Replication in Collaborative Virtual Environments.** IEEE Proceedings of the International Conference on Software, Telecommunications and Computer Networks, October 2001, Split, Dubrovnik, Croatia
- [ARAU96] Araujo, R. B. **Especificação e Análise de um Sistema Distribuído de Realidade Virtual.** Tese de Doutorado, EPUSP, São Paulo, Junho, 1996.
- [BARR96] Barrus, J. W., Waters, R. C., Anderson, D. B. **Locales and Beacons: Efficient and Precise Support For Large Multi-User Virtual Environments** IEEE Computer Graphics and Applications, Vol. 16, No. 6, November 1996, pp. 50-57
- [BECA01] Beca, L. **Building Collaborative Problem-Solving Environments as Shared Places.** Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.
- [BENF01] Benford, S., Greenhalgh, C., Rodden, T., Pycock, J. **Collaborative Virtual Environments.** Communications of the ACM, July 2001, Vol. 44 No. 7, pp. 79-85.
- [BRAS00] Brassé, M., Kuijpers, N. **Realizing a Platform for Collaborative Virtual Environments based on the High Level Architecture.** Proc. 2000 Spring Simulation Interoperability Workshop, March 2000, Orlando, Florida, USA.

- [BULL99] Bullock, A., Benford, S. **An Access Control Framework for Multi-user Collaborative Environments.** Proc. of ACM GROUP '99, Phoenix, Arizona, USA, 1999.
- [BURD94] Burdea, G. & Coiffet, P., **Virtual Reality Technology**, John Wiley & Sons, 1994, 399p.
- [CONS01] Constantini, F., Toinard, C. **Collaborative Learning with the Distributed Building Site Metaphor.** IEEE Multimedia, July-September 2001, pp. 21-29.
- [COSMOS] COSMOS - <http://www.discover.uottawa.ca/~mojtaba/Cosmos.html>
- [COVEN] COVEN - <http://www.crg.cs.nott.ac.uk/research/projects/Coven>
- [CRES00] Crespillo, D. S., Obrador, A. F. B., Granada, R. G., Luo, Y. **Implementation of mechanisms for concurrent 3D design and visualization.** Proceedings of the 2000 IEEE Conference on Information Visualization, July 2000.
- [DARL00] Darlagiannis V., Georganas N.D., **Virtual Collaboration and Media Sharing using COSMOS.** Proc. 4<sup>th</sup> WORLD MULTICONFERENCE on Circuits, Systems, Communications & Computers (CSCC 2000), Greece, July 2000.
- [DIVE] DIVE - <http://www.sics.se/dive/>
- [GREE94] Greenberg, S., Marwood, D. **Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface.** Proceedings of the CSCW 94, Chapel Hill, USA, 1994, 207-217.
- [GREE95] Greenhalgh, C., Benford, S. **MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading.** ICDCS, 1995, 27-34, 1995.
- [GREE97] Greenhalgh, C., Benford, S. **Boundaries, Awareness and Interaction in Collaborative Virtual Environments.** Proceedings of the 6th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE), June 18-20, 1997, Cambridge, Massachusetts, USA.
- [GREE00] Greenhalgh, C., Purbrick, J., Snowdon, D. **Inside MASSIVE-3: Flexible Support for Data Consistency and World Structuring.** Proceedings of the CVE 2000 Symposium, ACM, San Francisco, 2000, 119-127.
- [GRISO02] Grisoni, L., Degrande, S., Chaillou, C., Ferley, E., Cani, M., Gascuel, J. **SpinCAS: a step towards Virtual Collaborative Sculpting.** Virtual Reality. International Conference, Laval, France, 2002.
- [GUTW99] Gutwin, C., and Greenberg, S. **The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware.** ACM Transactions on CHI, 1999.
- [HAGS96] Hagsand, O. **Interactive MultiUser VEs in the DIVE System.** IEEE Multimedia Magazine, Vol 3, Number 1, 1996.

- [HIND00] Hindmarsh, J., Fraser, M., Heath, C., Benford, S. and Greenhalgh, C., **Object-Focused Interaction in Collaborative Virtual Environments**, in ACM Transactions on Computer-Human Interaction (ACM ToCHI), 7 (4), December 2000, ACM Press.
- [HOSS01] Hosseini, M., Georganas, N. D. **Suitability of MPEG4's BIFS for Development of Collaborative Virtual Environments**, Proceedings of 10 th IEEE International Workshop on Enabling Technologies for Collaborative Enterprises 2001 (WET ICE'01), Cambridge, MA, USA, June 2001.
- [HOSS02] Hosseini, M., Georganas, N.D. **MPEG-4 BIFS Streaming of Large Virtual Environments and their Animation on the Web**, Proc. ACM Web3D Symposium, Tempe, Arizona, February 2002.
- [KAN01] Kan, H. Y., Duffy, V. G., Su, C. **An Internet Virtual Reality Collaborative Environment for Effective Product Design**. Computers in Industry, Vol. 45, p. 197-213, 2001.
- [KAUS99] Kausar, N., Crowcroft, J. **An Architecture of Conference Control Functions**. Proceedings of Photonics East, Boston, Massachusetts, September, 1999.
- [KUIJ97] Kuijpers, N., Jense, H. **Collaborative Engineering in Distributed Virtual Environments**, Proc. Spring Simulation Interoperability Workshop, March 3-7, Orlando, Florida, USA, 1997.
- [LEE02] Lee, D., Lim, M., Han, S. **ATLAS – A Scalable Network Framework for Distributed Virtual Environments**. Proceedings of the CVE'02 Symposium, ACM, Bonn, Germany, 2002, 47-54.
- [LEIG97] Leigh, J., Johnson, A. E., DeFanti, T. A. **Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments**. Proceedings of Supercomputing '97 San Jose, California, November, 1997.
- [LI01] Li, F. W. B., Lau, R. W. H., Ng, F. F. C. **Collaborative Distributed Virtual Sculpting**. Proceedings of IEEE VR Conference, March 2001.
- [LI97] Li, F. W. B., Lau, R. W. H., Ng, F. F. C. **Interactive Rendering of Deforming NURBS Surfaces**. EUROGRAPHICS'97, 1997, Vol. 16, N.03
- [LIVING] Living Worlds Proposal Draft 2, [http://www.vrml.org/WorkingGroups/living-worlds/draft\\_2/index.htm](http://www.vrml.org/WorkingGroups/living-worlds/draft_2/index.htm)
- [MACE94] 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, pp.265-287.
- [M3874] ISO/IEC M3874, **Working Draft for MPEG-4 multi-users worlds architecture and tools**. January 2001.

- [M6692] Todesco, G., Rueckert, U., Abreu, S., Bressan, C. M., Araujo, R.B. **MPEG-4 Multi-Users Technology**. Contribuição de Padronização Internacional para o ISO/IEC JTC1/SC29/WG11, number: m6692, January, 2001.
- [N2201] ISO/IEC N2201, **MPEG-4 Systems**. May, 1998.
- [N4264] ISO/IEC N4264, **Information Technology – Coding of audio-visual objects – Part 1: MPEG-4 Systems**. August 2001.
- [N4272] PDAM of ISO/IEC 14496-1 / AMD4, **MPEG-4 Systems**. Sydney, July 2001.
- [N4415] PDAM of ISO/IEC 14496-1 / AMD4, **MPEG-4 Systems**. Pattaya, December 2001.
- [N4668] ISO/IEC N4668, **MPEG-4 Overview – (V.21 – Jeju Version)**, March 2002.
- [NPSNET] NPSNET - <http://www.npsnet.org/>
- [OLIV99] Oliveira, J.C., Shirmohammadi, S., Georganas, N.D. **Collaborative Virtual Environment Standards: A Performance Evaluation**, Proc. IEEE/ACM Third International Workshop on Distributed Interactive Simulation and Real Time Applications (DIS – RT '99), Greenbelt MD, Oct. 1999.
- [OLIV00] Oliveira, J. C., Shirmohammadi, S., Hosseini, M., Cordea, M., Georganas, N. D., Petriu, E., Petriu, D.C., **Virtual Theater for Industrial Trainin<sup>g</sup>: A Collaborative Virtual Environment**. Proceedings of 4th WORLD MULTICONFERENCE on Circuits, Systems, Communications & Computers (CSCC 2000), Greece, July 2000.
- [PAND97] Pandzic, I. S., Capin, T. K., Thalmann, N. M., Thalmann, D. **MPEG-4 for Networked Collaborative Virtual Environments**, Proceedings of the 1997 International Conference on Virtual Systems and MultiMedia (VSMM'97), IEEE, 1997.
- [PAND97a] Pandzic, I. S., Capin, T. K., Thalmann, N. M., Thalmann, D. **Data Exchange in Networked Collaborative Virtual Environments**. Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI'97), Rhodes, Greece, 1997.
- [PETT01] Pettifer S., Marsh J. **A Collaborative Access Model for Shared Virtual Environments**, Proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June 20-22 2001, MIT Cambridge MA, IEEE Press.
- [PICA01] Picard, S. L. D., Degrande, S., Gransart, C., Chaillou, C., Saugis, G. **Communication Platform for Synchronous Collaborative Virtual Environments**. International Conference on Media Futures 2001, Florence, Italy, May, 2001.
- [PINT01] Pinto, M., Amor, M., Fuentes, L. Troya, J. M. **Collaborative Virtual**

- Environment Development: An Aspect-Oriented.** Approach. IDCSW'01, Mesa, Arizona, April, 2001.
- [RAPO01] Raposo, A. B, Magalhães, L. P., Ricarte, I. L. M, Fuks, H. **Coordination of Collaborative Activities: A Framework for the Definition of Tasks Interdependencies.** CRIWG'01, Darmstadt, Germany, September, 2001.
- [RIED97] Riedel, O. H., Breining, R., Scharm, H. R. **How to use Virtual Environments for Engineering Projects,** SIGGRAPH'97, 1997.
- [SALL02] Sallés, E. J., Michael, J. B., Capps, M., McGregor, D., Kapolka, A. **Security of Runtime Extensible Virtual Environments.** In Proc. Fourth Int. Conf. on Collaborative Virtual Environments, ACM, Bonn, Germany, September, 2002.
- [SENSE8] SENSE 8 – <http://www.sense8.com>
- [SHEN92] Shen, H., Dewan, P. **Access Control for Collaborative Environments.** Proceedings of the CSCW'92, ACM, 1992, 51-58.
- [SIGN02] Signès, J., Fisher, Y., Eleftheriadis, A. **MPEG-4's Binary Format for Scene Description.** <http://leonardo.telecomitalialab.com>.
- [SPLINE] SPLINE - <http://www.merl.com/projects/spline>
- [STUE02] Stuerzlinger, W., Smith, G. **Efficient Manipulation of Object Groups in Virtual Environments.** Published in the Proceedings of the VR2002, March 24-28, 2002, Orlando, Florida.
- [SUN02] Sun, C., Chen, D. **Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems.** ACM Transactions on Computer-Human Interaction, Vol. 9, No. 1, March 2002, Pages 1–41.
- [VLNET] VLNET - <http://ligwww.epfl.ch/~thalmann/vlnet.html>
- [YANG00] Yang, J., Lee, D. **Scalable Prediction Based Concurrency Control for Distributed Virtual Environments.** IEEE Virtual Reality 2000 Conference New Brunswick, New Jersey, March, 2000
- [W4852] **MU & AFX Study** of PDAM of ISOIEC 14496-1 AMD4, may 2002.
- [WALS01] Walsh, A. E., Bourges-Sévenier, M. **Core Web3D.** Prentice Hall PTR. 2001.
- [ZACH98] Zachmann, G. **VR-Techniques for Industrial Applications,** Virtual Reality for Industrial Applications. Springer 1998. Chapter 1, pages 13 - 38.

## Anexo A – Interfaces do Modelador 3D Colaborativo

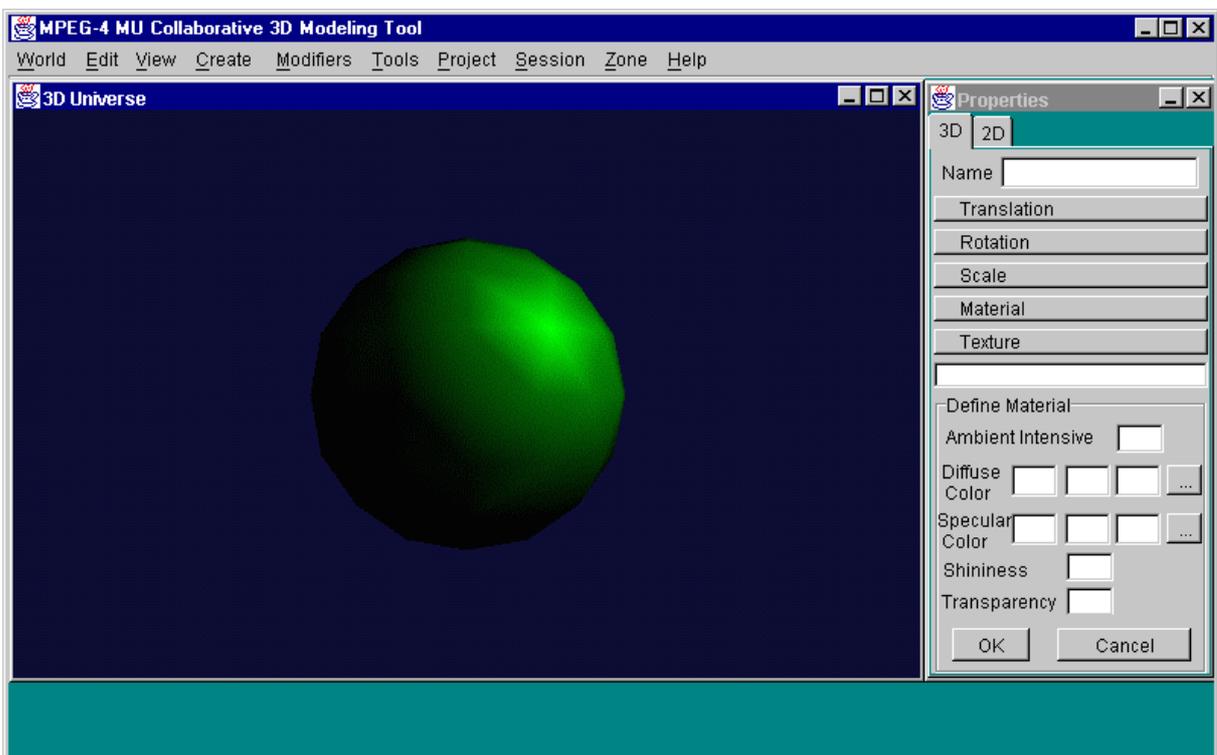


Figura 47 - Interface principal do Modelador 3D Colaborativo com um Universo 3D.

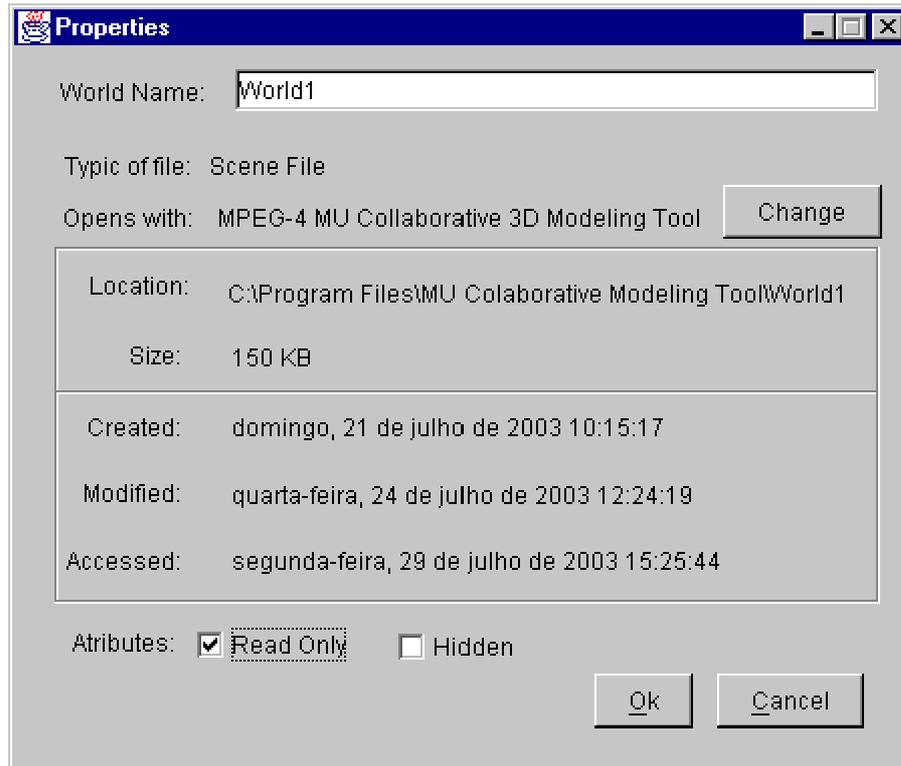


Figura 48 - Propriedades do Mundo Virtual.

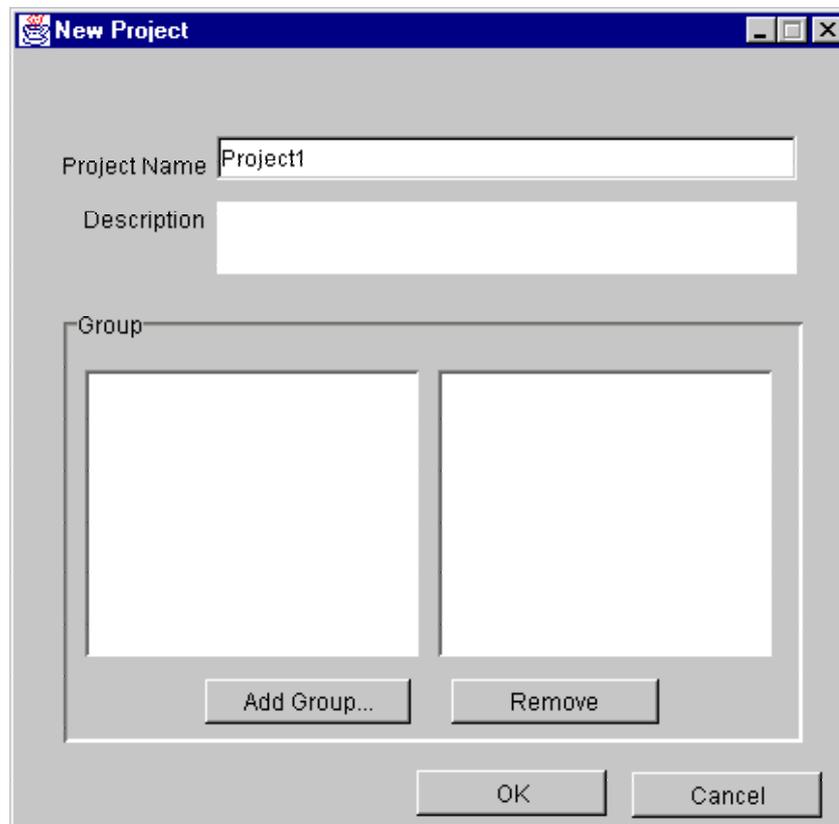


Figura 49 - Criação de um Projeto.

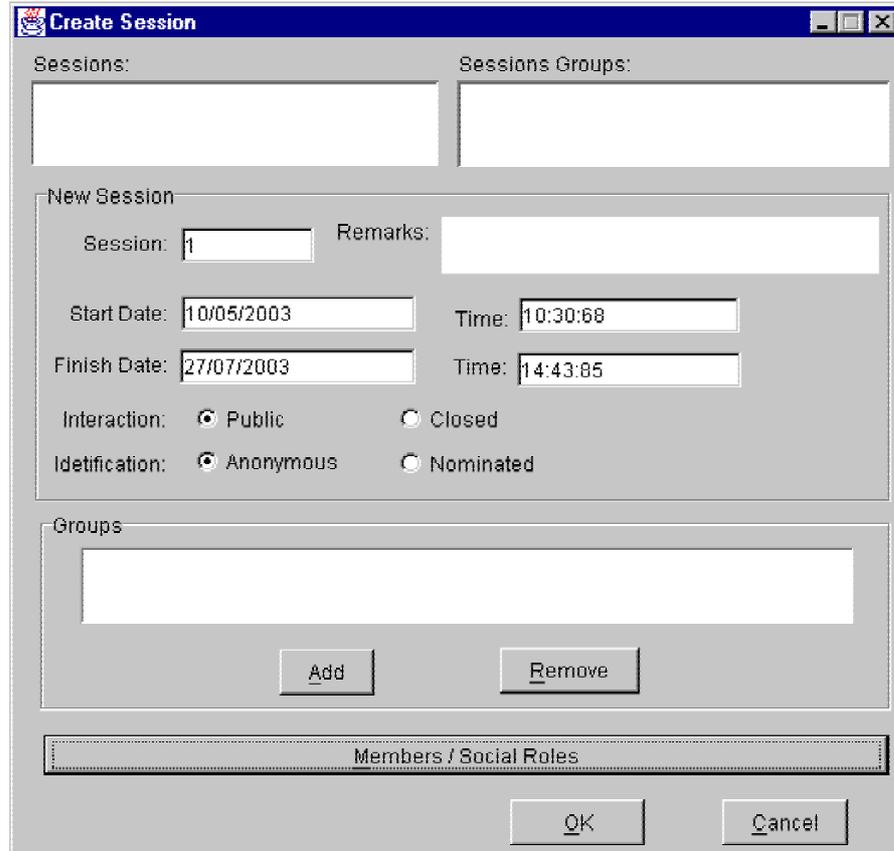


Figura 50 - Criação de uma Sessão.

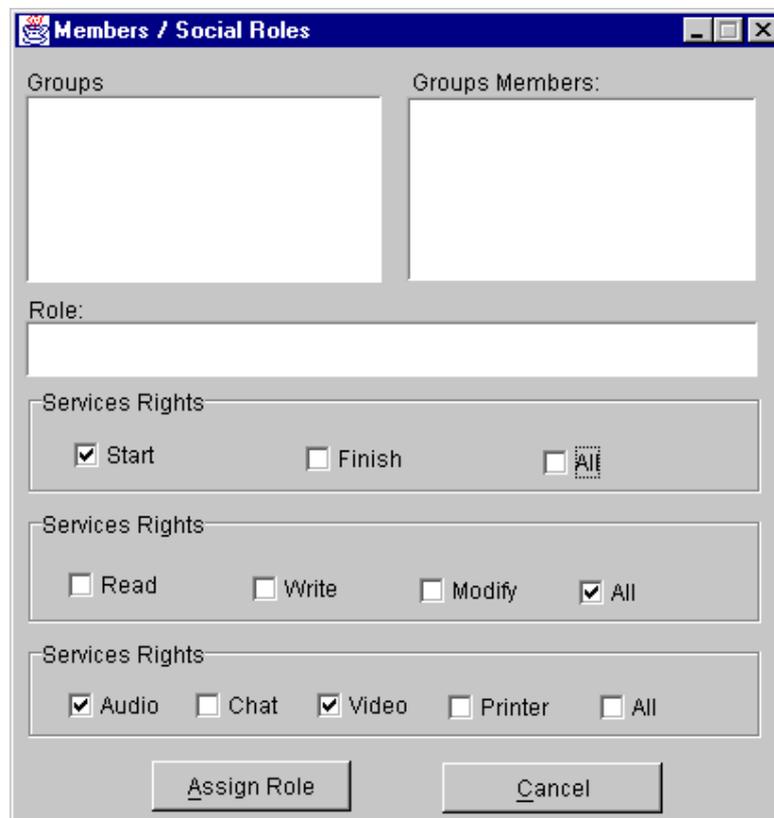


Figura 51 - Atribuição de Papéis Sociais aos Participantes de uma Sessão.

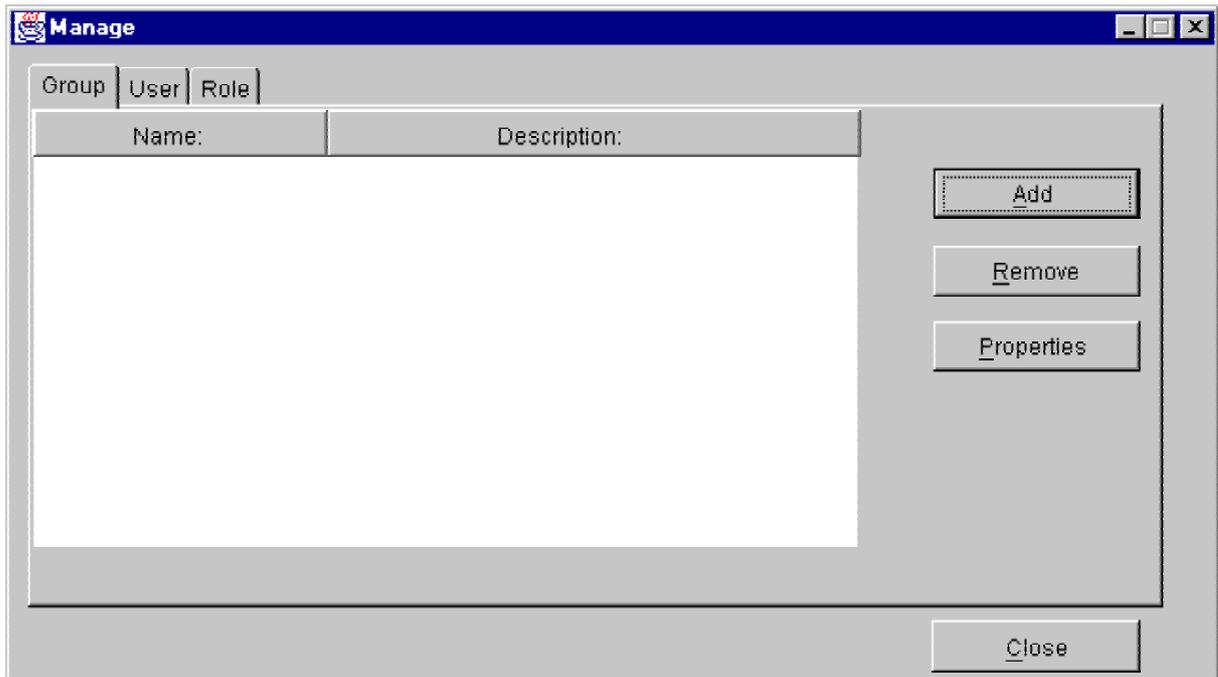


Figura 52 - Gerenciamento de Grupos.



Figura 53 - Criação de Grupos.



Figura 54 - Propriedades de Grupo.

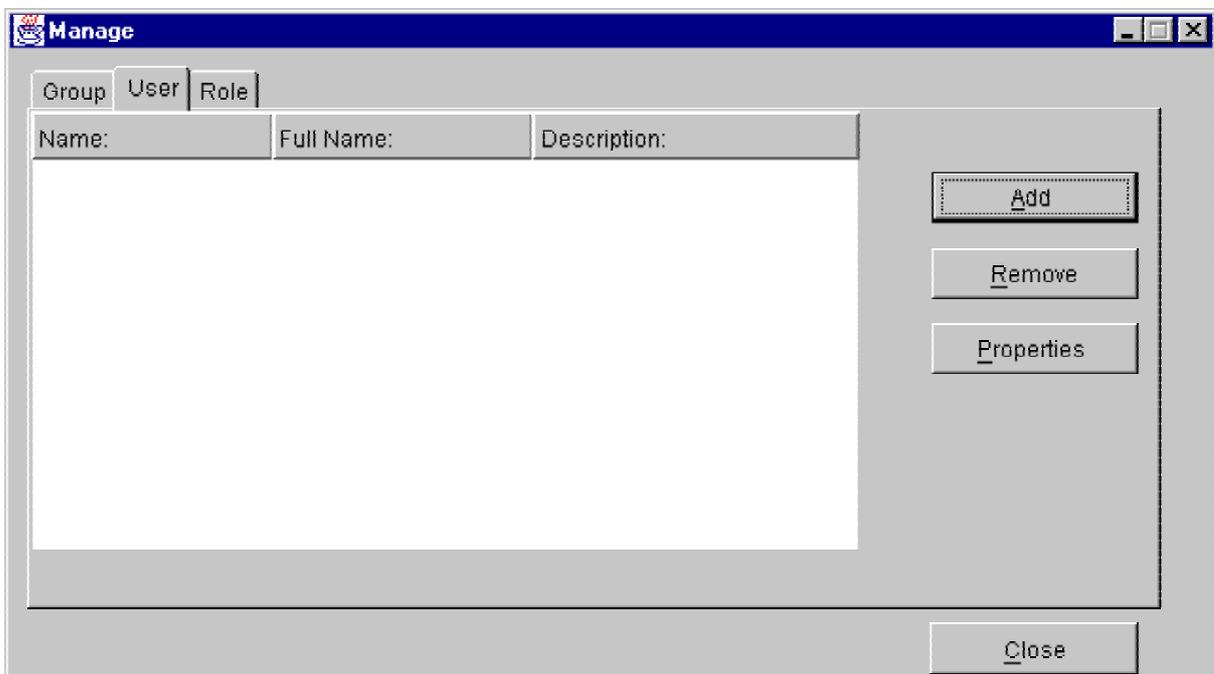


Figura 55 - Gerenciamento de Usuários.

**New User**

User Name:

Full Name:

Description:

Password:

Confirm Password:

User must change password at next logon

User cannot change password

Password never expires

Figura 56 - Criação de Usuários.

**User Properties**

General

**User 1**

Full Name:

Description:

User must change password at next logon

User cannot change password

Password never expires

Figura 57 - Propriedades de Usuário.

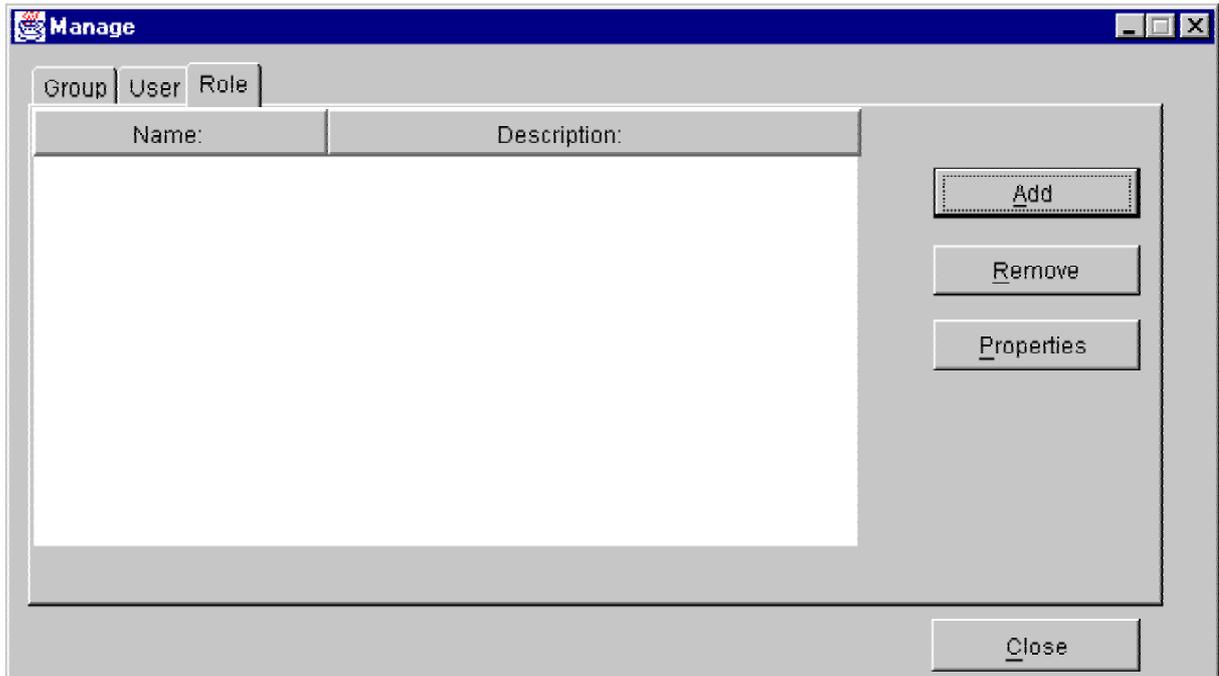


Figura 58 - Gerenciamento de Papéis Sociais.

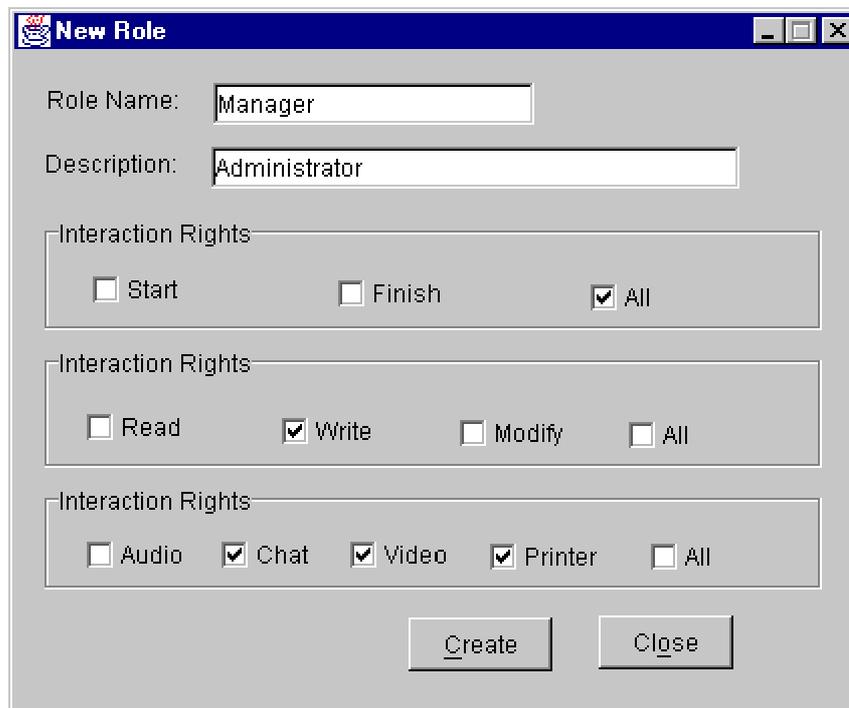


Figura 59 - Criação de Papéis Sociais.



Figura 60 - Propriedades de Papel Social.

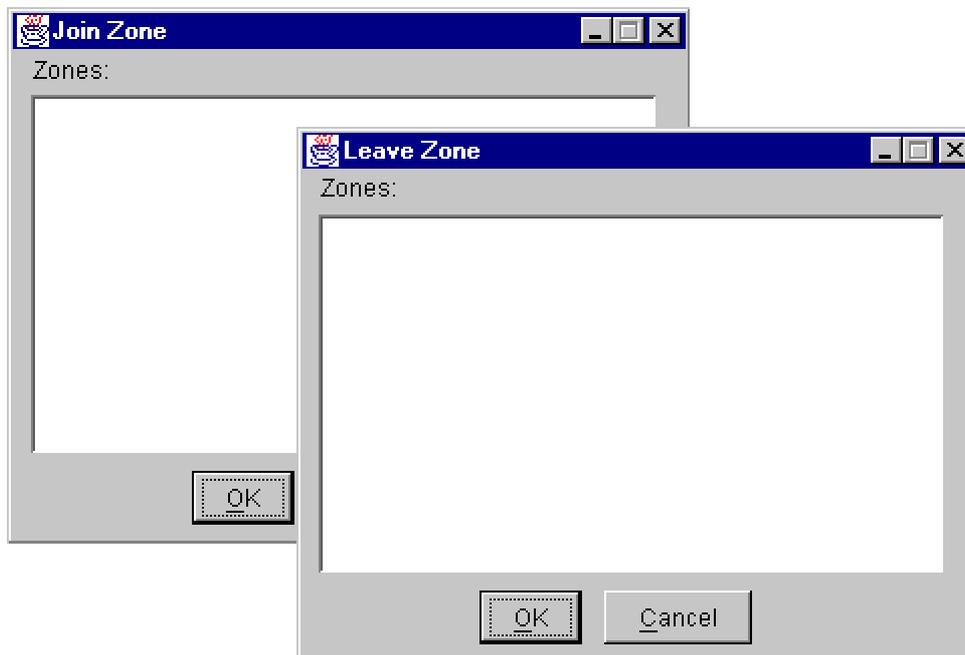


Figura 61 - Entrada e Saída de Zonas Compartilhadas.

## **Anexo B – Implementação dos Componentes MSC e MBK**

Os principais componentes que controlam a arquitetura do padrão MPEG-4 MU, o MSC e o MBK, foram escritos usando a linguagem JAVA.

Os componentes estão divididos em três partes: primeiro o MSC e suas classes, em seguida o MBK e suas classes e por fim as mensagens utilizadas para controle do MSC e MBK e suas respectivas classes. As definições seguem a ordem:

- Nome da classe;
- Funcionalidade da Classe;
- Cabeçalho da classe;

### **MSC – MUTech SessionControl**

#### **Classe MSC Manager**

Classe responsável pelo gerenciamento de todas as operações realizadas pelo MSC.

```
public class MSCManager
```

#### **Classe Client Manager**

Classe responsável pelo gerenciamento dos clientes, conexões, recebimento e envio de mensagens, entre outras tarefas.

```
public class ClientManager extends Thread
```

### **Classe InfoMngr**

A classe InfoMngr é responsável por comandar os vários InfoSupplier. Cada objeto instanciado InfoSupplier administra uma conexão com socket TCP/IP provendo informações sobre o ambiente comandado pelo MSC/MBK.

```
public class InfoMngr extends Thread
```

### **Classe InfoSupplier**

Classe encarregada de estabelecer conexões com clientes ou eventuais futuros clientes do MSC para prover determinadas informações sobre o que ocorre na sessão e zonas que o MSC/MBK está tomando conta atualmente. Cada objeto InfoSupplier instanciado estabelece um socket com algum programa, sendo esse socket TCP/IP e que troca Strings.

```
public class InfoSupplier extends Thread
```

### **Classe MSCInterface**

Classe responsável pela construção da interface do MSC.

```
public class MSCInterface extends JFrame
```

### **Classe MessagesTreatment**

Essa classe é a responsável pelo tratamento das mensagens MPEG-4 e suas possíveis respostas.

```
public class MessagesTreatment
```

## **MBK – MUTech Bookkeeper**

### **Classe MBKManager**

Classe responsável pelo gerenciamento das funções do MBK.

```
public class MBKManager
```

### **Classe MBKInterface**

Essa classe é a responsável pela construção da interface do MBK.

```
public class MBKInterface extends JFrame
```

### **Classe MsgHandler**

Classe responsável pela manipulação das mensagens de atualização do MBK.

```
public class MsgHandler
```

## **Mensagens MPEG-4**

### **Classe JoinSessionRequest**

Classe responsável pela manipulação da mensagem de entrada numa sessão multiusuário.

```
public class JoinSessionRequest
```

### **Classe JoinSessionAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação a entrada de uma sessão multiusuário.

```
public class JoinSessionAcknowledge
```

### **Classe JoinZoneRequest**

Classe responsável pela manipulação da entrada numa zona multiusuário.

```
public class JoinZoneRequest
```

### **Classe JoinZoneAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação a entrada de uma zona multiusuário.

```
public class JoinZoneAcknowledge
```

### **Classe LeaveSessionRequest**

Classe responsável pela manipulação das mensagens de saída de uma sessão.

```
public class LeaveSessionRequest
```

### **Classe LeaveSessionAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação da saída de uma sessão multiusuário.

```
public class LeaveSessionAcknowledge
```

### **Classe LeaveAllZonesRequest**

Classe responsável pela manipulação das mensagens de saída de todas as zonas.

```
public class LeaveAllZonesRequest
```

### **Classe LeaveAllZonesAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação a entrada de todas as zonas.

```
public class LeaveAllZonesAcknowledge
```

### **Classe LeaveZonesRequest**

Classe responsável pela manipulação das mensagens de saída de uma zona.

```
public class LeaveZonesRequest
```

### **Classe LeaveZonesAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação a saída de uma zona.

```
public class LeaveZonesAcknowledge
```

### **Classe MURquest**

Essa classe é a responsável pelas mensagens de solicitação (*Join Session, leave session*, entre outras).

```
public class MURquest
```

### **Classe MUUpdate**

Classe responsável pelas mensagens de propagação das modificações da cena.

```
public class MUUpdate
```

### **Classe ZonePilotShipRequest**

Essa classe é a responsável pela mensagem de solicitação de Pilotagem.

```
public class ZonePilotShipRequest
```

### **Classe ZonePilotShipAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação em resposta ao pedido de pilotagem de um objeto por um usuário.

```
public class ZonePilotShipAcknowledge
```

### **Classe ZoneReleasePilotShipRequest**

Classe responsável pela manipulação das mensagens de liberação de pilotagem.

```
public class ZoneReleasePilotShipRequest
```

### **Classe ZoneReleasePilotShipAcknowledge**

Essa classe é a responsável pela manipulação das mensagens de aceitação/negação da liberação da pilotagem

```
public class ZoneReleasePilotShipAcknowledge
```

## Anexo C – Publicação de Artigos

As publicações obtidas são apresentadas neste anexo.

- Duarte, F. V., Verdi, M., Araujo, R. B. How to Support Collaborative Modelling with the Emerging Standard MPEG-4 MU. Proceedings of XV Brazilian Symposium on Computer Graphics and Image Processing, Fortaleza, Brazil, October, 2002.
- Duarte, F. V., Araujo, R. B. A 3D Collaborative Modeler Based on the Emerging MPEG-4 MU Standard. Proceedings of XVI Brazilian Symposium on Computer Graphics and Image Processing, São Carlos, Brazil, October, 2003. (aceito)
- Duarte, F. V., Silva, A. R., Araujo, R. B. A Collaborative Virtual for 3D Modelling Based on the MPEG-4 MU Emerging Standard. The 10th International Multimedia Modelling Conference (MMM2004), Brisbane, Australia, January, 2004. (submetido)
- Duarte, F. V., Araujo, R. B. Um Ambiente Colaborativo para Modelagem 3D Baseado no Padrão Emergente MPEG-4 MU. WEBMÍDIA 2003, Salvador, Brasil, Novembro, 2003. (submetido)

# How to Support Collaborative Modelling with the emerging standard MPEG-4 MU

DUARTE, F. V.; VERDI, M.; ARAUJO, R.B.

Computer Science Department - UFSCar - Universidade Federal de São Carlos  
Via Washington Luís, Km 235, Caixa Postal 676, CEP 13.565-905, São Carlos, SP, Brasil  
{feduarte,verdi,regina}@dc.ufscar.br

## Introduction

In a Collaborative modelling environment, multiple users may interact in a virtual environment to deform or modify shared 3D objects (e.g., for modelling and visualization of virtual prototypes). The challenges to collaborative modelling are related mainly to four aspects: user interaction with the virtual modelling environment, 3D objects concurrent editing, consistency maintenance among client terminals and real time rendering of the modified 3D objects.

The use of *Non-Uniform Rational B-Splines (NURBS)* surfaces accelerates the rendering process since a small number of control points are used to represent the objects, resulting in a large data compression. Nowadays a number of technologies has either emerged or evolved to support 3D environments, such as: X3D, Java3D and MPEG-4MU. This poster shows how the emerging MPEG-4 MU standard can be used to support real-time modelling through NURBS in a shared virtual environment among multiple client terminals.

## MPEG-4 MU support to consistency maintenance in a shared virtual environment

MPEG-4 is an *ISO/IEC (ISO/IEC JTC1/SC29/WG11)* standard, developed by the *Moving Picture Experts Group (MPEG)*, for the codification and delivery of different media formats over a wide range of networks and computer platforms. Part of the MPEG-4 standard defines a *Binary Format Scene – BIFS*, which codifies scenes in a compact and efficient way (10 to 15 times smaller than VRML files) [1].

The support to multiple users is under standardization, as an extension to the MPEG-4 standard (emerging MPEG-4MultiUser). Consistency maintenance is obtained through a mechanism named *Pilot/Drone* and a *BIFS-Command* protocol, which are used to reflect objects state changes among all participants. It is also under development the *Animation Framework eXtension (AFX)*, which defines a set of interoperable tools to support interactive animated content. AFX specifies new tools to accelerate scene rendering with better visual quality, such as geometry tools (NURBS surfaces and curves, subdivision surfaces etc), modelling tools

(Nonlinear Global Deformation and Free-Form Deformations), among others [2].

## The traffic generated by the BIFS Command Protocol for updating of NURBS objects

In a collaborative modelling environment where objects are modelled with NURBS surfaces and curves, the updating of the control points are fundamental, as they can be modified by one user and the result has to be propagated to the other client terminals.

The propagation of the updates is subject to delays due to messages transmission and processing. With the BIFS Command protocol, 164 bits are necessary to code the updating message of a NURBS object with one control point. (1 bit for the type of message, 1 bit for the command Id, 4 bits for zone Id, 4 bits for command category, 5 bits for the Node Id, 3 bits for the field Id, 2 bits for the modified position identification, 16 bits for specific identification and 128 bits for the new values of X, Y, Z with 32 bits for each coordinate and 32 bits more for the weight represented as W).

Considering the size of the updating message of a NURBS object with one control point, and the frequency with which this message can be transmitted in a collaborative modelling exercise, network traffic will not be a major obstacle for the immediate propagation of the scene modifications among the client terminals. The major delays will come from the exchange of information among client terminals and the MPEG-4MU architecture components (MSC, MBK e MMH), besides message processing and scene updating at the remote client terminals. MPEG-4MU emerging standard however, is a promising technology for the support to collaborative modelling.

## Bibliographic References

- [1] Hosseini, M., Georganas, N. D. "Suitability of MPEG4's BIFS for Development of Collaborative Virtual Environments", Proceedings of 10 th IEEE WET ICE'01, Cambridge, MA, USA, June 2001.
- [2] ISO/MPEG N4415 – MPEG-4 Systems – December 2001.

## A 3D Collaborative Modeler Based on the Emerging MPEG-4 MU Standard

FERNANDO V. DUARTE, REGINA B. ARAUJO

Computer Science Department - UFSCar – Federal University of São Carlos  
Via Washington Luís, Km 235, Caixa Postal 676, CEP 13.565-905, São Carlos, SP, Brazil  
{feduarte,regina}@dc.ufscar.br

**Abstract.** This work presents the implementation of a 3D collaborative modeler based on the emerging MPEG-4 MU (multi-user) standard. With this modeler, 3D graphic scenes in MPEG-4 format can be created in real-time, by multiple participants, in synchronous collaborative sessions.

### 1. Introduction

A 3D collaborative modeler based on the multi-user MPEG standard is being implemented which supports multiple users interaction in the joint task of creating and modifying 3D graphic scenes in MPEG-4 format. The 3D modeler is a Collaborative Virtual Environment – CVE which generates scenes that can be visualized in any MPEG-4 terminal through the MPEG-4 Player. Important issues in the development of CVEs include: session management, scene synchronization and concurrency control. These issues are supported in the 3D modeler through the MUTech Session Controller – MSC and MUTech Bookkeeper – MBK, both components implemented by the Networked Virtual Reality Lab at Federal University of S. Carlos, whose interface and functionalities were defined by the emerging MPEG-4 MU standard. Some details of the implementation are presented below.

### 2. Project Management

A project can be created, open or modified. Groups of users can be added to or removed from a project. The project owner can establish policies, such as who can access what. These policies can be used to configure sessions, before they are set-up. Roles are created which can be used either to enforce concurrency control policies or to establish access rights.

### 3. Support to the Requirements of CVEs

Session management in the 3D modeler is realized by the MSC component. When a participating user wishes to join a shared environment, the MSC is responsible for allowing or denying her entrance, based on the policies set up by the project creator. If the user is allowed to join a session, she is assigned to a zone and enabled to interact with other users and with the shared objects.

The synchronization of the scenes among the terminals of the participating users is supported by the Pilot/Drone mechanism along with the BIFS-Command protocol [1]. In the 3D modeler, every

shared object has a corresponding Pilot. When a modification is made on a shared object node, an action is triggered: the Pilot for the modified object propagates the modification to its corresponding Drones, one per participating user terminal. This means that the updating messages (BIFS-Commands) managed by the MBK component will be sent to all participating users. The modification of the shared object state is encoded by a BIFS-Encoder followed by the generation of a BIFS-Command. The BIFS-Command is encapsulated in an MU-Command and transmitted to the MBK which will broadcast the modification to the corresponding user terminals.

Concurrency control is implemented through the integration of two mechanisms: Pilot/Drone and locking. For a user to be able to make a modification in a scene object, he must request the “piloting” of the shared object by sending a request message to the MBK component. The MBK will then locate the object Pilot. The Pilot of the shared object, which attends multiple requests in a *first come first served* fashion, may answer to that request by accepting or denying the piloting transfer. This acceptance/denial is based on access rights policies set up by the project owner. Once the request is accepted the object is “locked” so that no other user may be piloting the same object – only one user can be editing an object at a particular time.

### 4. Conclusions

The strengths of this 3D modeler based on the MPEG-4 MU standard are threefold: scene updating is made more efficiently since the modifications are encoded in a binary format (through BIFS commands); a standardized solution is used to fulfill complex and challenging requirements of CVEs and; multiple users can take part in a project session by using the modeler while other users can act as observers by using a ubiquitous MPEG-4 terminal.

### References

- [1] FPDAM of ISO/IEC 14496-1/AMD4. MPEG-4 Systems. N5285, Shanghai, October, 2002.

# A Collaborative Environment for 3D Modeling Based on the MPEG-4 MU Emerging Standard

Duarte, F.V., Silva, A. R., and Araujo, R.B.

Computer Science Department - UFSCar – Federal University of São Carlos, Via Washington Luís, Km 235, Caixa Postal 676, CEP 13.565-905, São Carlos, SP, Brasil.

*{feduarte,arsilva,regina}@dc.ufscar.br*

## Abstract

*Collaborative virtual environments for 3D modeling can be characterized by the interaction among multiple users for the creation and/or modification of shared 3D objects. These environments can be used, for instance, in the modeling and visualization of virtual prototypes in order to reduce costs in the process of products design. The challenges of building collaborative 3D modeling environments are mainly related to real-time rendering of the modified objects, as well as to users interaction with the virtual environment, and to consistency maintenance of the shared virtual environment. Only a few collaborative 3D modeling environments are found in the literature and usually their projects result in complex non standardized solutions for the collaboration among users.*

*This paper presents the implementation of a collaborative environment for 3D modeling based on the emerging MPEG-4MU (multi-user) standard. With this environment, 3D graphic scenes can be created in real-time, by multiple participants, in synchronous collaborative sessions. These scenes can be visualized in any MPEG-4 terminal, including cellulators and personal digital assistants. Sessions control, consistency maintenance, concurrency control and 3D object locking are realized by the MSC (MUTech Session Controller) and MBK (MUTech Bookkeeper) components, through the Pilot/drone mechanism and the BIFS-Command protocol. All these components were defined by the MPEG-4MU emerging standard and implemented by the Networked Virtual Reality Lab at S. Carlos Federal University.*

**Keywords:** CSCW, collaborative 3D modeling, collaborative virtual environment, Multi-User MPEG-4

## 1. Introduction

Collaborative Virtual Environments – CVEs can be seen as a convergence of two areas: Virtual Reality – VR and Computer-Supported Cooperative Work – CSCW. CVEs can be characterized by the use of tri-dimensional synthetic spaces which are shared among multiple users, remotely located to support collaborative projects, such as prototypes modeling and visualization, where costs can be reduced in the products project. Other applications

include visualization, simulation, training, education, as well as entertainment [1]. Important issues in the development of CVEs include: session management, user and users group users awareness, consistency maintenance of the shared environment, Os temas centraissegurança, resources management, persistency, access control, concurrency control and communication. a

This paper describes the implementation of a collaborative 3D modeler based on the emerging multi-user MPEG-4MU standard. ISO/IEC JTC1/SC29/WG11 MPEG4 was developed by the MPEG (Moving Picture Experts Group) for coding and delivery of different media formats in a wide variety of networks and computational platforms. The System Part of the MPEG-4 defines a structure named BIFS (Binary Format Scene) which encodes scenes in a compact and efficient way [2]. Multi-user support is specified by the emerging MPEG-4MU through the components MSC and MBK.

With the modeler described in this paper, MPEG-4 graphic scenes can be created, in real-time, by multiple participants in collaborative sessions. The sessions are established by the MUTech Session Controller – MSC. BIFS commands are used to encode modifications in the scene graph. These commands are sent to participating users through the Mutech Bookkeeper component - MBK in order to keep the shared scene synchronized in all users terminal. The use of the BIFS command as the mechanism for the shared scene consistency maintenance has been reported in the literature [3], showing that BIFS commands can be an efficient mechanism since it encodes the data and sends only the necessary information, i.e., what has been modified. The strength of the 3D modeler described in this paper is its close relationship with the player MPEG-4. A group of users could be collaborating in a 3D modeling task while other users could be only visualizing the task, through any device, from a PDA to a desktop computer, so long as these devices have a running MPEG4 player – what is becoming an increasing reality. The paper is organized as follows: section two is a brief overview of CVEs and their main requirements. Section three outlines multi-user MPEG-4 emerging standard. Section four presents the 3D modeler based on the multi-user MPEG4. Related work is described in section five, followed by future work and conclusions.

# Um Ambiente Colaborativo para Modelagem 3D Baseado no Padrão Emergente MPEG-4 MU

Fernando Vieira Duarte, Regina Borges de Araujo

Departamento de Computação – Universidade Federal de São Carlos (UFSCar), Via Washington Luís, Km 235, Caixa Postal 676, CEP 13.565-905, São Carlos, SP, Brasil.

{feduarte,regina}@dc.ufscar.br

***Abstract** Collaborative Virtual Environments for 3D modeling can be characterized by the interaction among multiple users for the creation and/or modification of shared 3D objects. These environments can be used, for instance, in the modeling and visualization of virtual prototypes in order to reduce costs in the process of products design. The challenges of building collaborative 3D modeling environments are mainly related to real-time rendering of the modified objects, as well as to users interaction with the virtual environment, and to consistency maintenance of the shared virtual environment. This paper presents the implementation of a collaborative environment for 3D modeling based on the emerging MPEG-4MU (multi-user) standard.*

***Resumo.** Ambientes Virtuais Colaborativos para modelagem 3D podem ser caracterizados pela interação entre múltiplos usuários para a modificação e/ou criação de objetos 3D compartilhados. Estes ambientes podem ser usados, por exemplo, na modelagem e visualização de protótipos virtuais para reduzir custos no processo do projeto de produtos. Os desafios da construção de ambientes de modelagem 3D colaborativos estão relacionados principalmente com a renderização em tempo-real dos objetos modificados, como também a interação dos usuários com o ambiente virtual, e a manutenção da consistência do ambiente virtual compartilhado. Este artigo apresenta a implementação de um ambiente colaborativo para modelagem 3D baseado no padrão emergente MPEG-4 MU (multi-user).*