

O PADRÃO MPEG-4 MU NO SUPORTE A AMBIENTES VIRTUAIS MULTIUSUÁRIOS

Matheus Pessote Verdi – Faculdade Anhanguera de Rio Claro

RESUMO: Os Ambientes Virtuais Multiusuários (AVM) se caracterizam pelo compartilhamento de informações e propagação de mensagens entre múltiplos usuários ali presentes. Para tal, a propagação e sincronização de mensagens são fatores que influenciam e limitam a preservação da consistência do ambiente. Muitas linguagens são utilizadas para desenvolver um AVM, mas, a maioria delas possui limitações ou não garantem o suporte a AVM. Nesse contexto, o padrão MPEG 4 e sua extensão o MPEG 4 MU, apresentam condições favoráveis e satisfatórias para que um Ambiente Virtual Multiusuário possa suportar atualizações em tempo real e assim permitir que esses ambientes troquem e compartilhem informações de forma que a preservação da consistência do ambiente seja garantida

ABSTRACT: The Multiuser Virtual Environments (MVE) are characterized by information sharing and dissemination of messages between multiple users are present there. To this end, the dissemination and synchronization messages are factors that influence and limit the preservation of a consistent environment. Many languages are used to develop an MVE, but most have limitations or do not provide support for MVE. In this context, the MPEG 4 and its extension to MPEG 4 MU, they are favorable and satisfactory for a Multiuser Virtual Environment can support real time updates and thus enabling these environments to exchange and share information so that the preservation of consistency environment is guaranteed.

PALAVRAS-CHAVE:

Mundo Virtual, Ambiente Virtual Multiusuário, MPEG 4 e MPEG 4 MU

KEYWORDS:

Virtual World, Multiuser Virtual Environment, MPEG 4 and MPEG 4 MUe strategies

Informe Técnico

Recebido em: 31/10/2009

Avaliado em: 11/07/2013

Publicado em: 22/04/2014

Publicação

Anhanguera Educacional Ltda.

Coordenação

Instituto de Pesquisas Aplicadas e Desenvolvimento Educacional - IPADE

Correspondência

Sistema Anhanguera de de Revistas Eletrônicas -SARE

rc.ipade@anhanguera.com

1. INTRODUÇÃO

Os ambientes virtuais multiusuários para a Web são construídos por meio de várias linguagens e padrões que dão suporte a esses ambientes. Entretanto, devido à necessidade de tecnologias mais completas de suporte à codificação, transmissão e apresentação de múltiplas mídias, alternativas como o padrão MPEG-4 MU, surgiram para o suporte à implementação de ambientes virtuais multiusuários na Web. O padrão MPEG-4 e a sua extensão, o MPEG-4 MU, permitem a criação de ambientes virtuais multiusuários e ainda suprem limitações como, a falta de suporte para a geração de um formato de arquivo binário e a inexistência de um suporte à integração multimídia nos ambientes virtuais de forma satisfatória.

2. AMBIENTES VIRTUAIS MULTIUSUÁRIOS

Ambientes virtuais multiusuários (AVM) ou ambientes virtuais compartilhados, como o próprio nome sugere, consiste em um ambiente virtual onde existe mais que um usuário interagindo com o ambiente e com outros usuários em tempo real. Os usuários são representados por qualquer forma geométrica tridimensional e são denominados avatares.

Nos AVMs, é necessário tratar algumas ações, uma vez que neles, a interação entre múltiplos participantes se torna evidente. Essas ações são definidas como (LIVING, 1997):

As modificações na cena podem ser iniciadas por qualquer usuário;

Qualquer modificação deve, potencialmente, ser sincronizada para todos os usuários participantes;

Para todas as modificações, cada objeto, ou é a sua origem, com a responsabilidade de espalhar a sua notícia de modificação, ou é um receptor de notícias de modificação, e deve realizar a ação apropriada;

Nem todas as modificações são realizadas de uma só vez; a otimização do desempenho depende de saber o que é mais provável mudar em seguida;

Nem todas as notícias de modificação precisam ser comunicadas; a otimização depende também de ter conhecimento de quem necessita saber o quê e como.

As figuras 1 e 2 exemplificam, respectivamente, um mundo virtual não compartilhado, onde apenas a navegação por ele é possível e um mundo virtual compartilhado.



Figura 1 - Mundo virtual não compartilhado



Figura 2 - Mundo virtual compartilhado

Um dos maiores desafios dos ambientes virtuais multiusuário é a preservação da consistência do ambiente. Cada aplicação da simulação tem que ser capaz de trocar dados de estado da simulação de modo consistente. Para isso, modelos de comunicação eficientes têm que ser utilizados para suportar a comunicação entre os terminais cliente de um ambiente virtual multiusuário.

Um modelo de comunicação amplamente utilizado na web é o modelo cliente-servidor, em que o terminal cliente se conecta apenas a um servidor, e o servidor se conecta a cada um dos seus terminais cliente, no paradigma do “um-para-um”. Este modelo, entretanto, apresenta limitações quanto ao crescimento do sistema, uma vez que o servidor pode se tornar um ponto de gargalo quando o número de terminais cliente aumenta muito. Outra limitação deste modelo é o atraso adicional causado pelo servidor como um agente

intermediário na comunicação entre dois terminais cliente (terminal cliente 1 – servidor – terminal cliente 2) que poderiam, de outra maneira, comunicar-se diretamente um com o outro, gerando um atraso menor (terminal cliente 1 – terminal cliente 2).

Um modelo de comunicação que permite que terminais cliente comuniquem-se diretamente, sem qualquer elemento intermediário, é denominado peer-to-peer. Esta solução, entretanto acarreta um maior processamento por parte de cada terminal cliente, uma vez que cada um terá que ter um subconjunto das funcionalidades que um servidor assumiria quando num modelo cliente-servidor.

Uma solução híbrida é também possível, em que servidores são utilizados, por exemplo, apenas para controle, deixando a atualização de dados para ser feita diretamente entre os terminais cliente.

2.1. Requisitos de Ambientes Virtuais Multiusuário

De acordo com (VANDAN, 1993) (ARAUJO, 1996) os requisitos para AVMs são:

- Taxas rápidas de atualização do ambiente virtual (dependendo da aplicação, o ambiente tem que ser atualizado a taxas que variam de 10 a 60 quadros por segundo - qps);
- Latência mínima (entre 100ms a 300ms);
- Tratamento simultâneo de múltiplos dispositivos de Entrada/Saída (capacetes, óculos, dispositivos de rastreamento de movimentos, dispositivos de tato e força, etc.);
- Simulação de um número grande de objetos com comportamento complexo;
- Representação realista do mundo virtual;
- Distribuição dos dados em redes de longa distância;
- Suporte a tipos diferentes de unidades de dados, com requisitos diferentes de confiabilidade na entrega (dados sobre o estado da simulação, por exemplo, não exigem confiabilidade de entrega, ao contrário dos dados de colisão);
- Suporte à comunicação multiponto de várias fontes para vários destinos (paradigma do muitos-para-muitos);
- Suporte à integração e sincronização de informação multimídia nos ambientes virtuais;
- Suporte a recursos computacionais e de rede heterogêneos;

Em uma arquitetura de suporte a AVM, baseada no modelo cliente-servidor, vários requisitos devem ser observados do lado do cliente, do servidor, e do meio de transmissão. A tabela 1 apresenta alguns desses requisitos (MUTH, 2000).

Tabela 1 - Requisitos de um ambiente virtual (MUTH, 2000)

	Cliente	Comunicação	Servidor
Padrões	<ul style="list-style-type: none"> WWW browser - API-Plugin - Visualizador, Plugins- Graphics/ GUI-APIs 	<ul style="list-style-type: none"> - Infra-estrutura da Rede - Protocolos de Transporte - Serviços (HTTP, URL) 	<ul style="list-style-type: none"> - Servidor WWW - Proxy CGI - Armazenamento - Formatos de arquivos e fluxos MM e RV
Desempenho	<ul style="list-style-type: none"> - Baixa Latência (Carregar e Navegar) - Decodificação - Descompressão - Parse - Construção da cena gráfica - Alta taxa de Quadros/s e de Renderização 	<ul style="list-style-type: none"> - Alta taxa de transmissão de bits - Baixa Latência - Baixo atraso entre elementos da rede (jitter) - Implementação Otimizada 	<ul style="list-style-type: none"> - Capacidade de produção de volume de dados na rede
Qualidade	<ul style="list-style-type: none"> - Resolução - Antialiasing - Manipulação de cores 	<ul style="list-style-type: none"> - Qualidade do Serviço 	<ul style="list-style-type: none"> - Resolução - Método de compressão - Manipulação de cores
Funcionalidade	<ul style="list-style-type: none"> - Apresentação ao vivo - Sistemas de projeção de RV - Conforto na navegação (sensores 3D) 	<ul style="list-style-type: none"> - Streaming - Sincronização 	<ul style="list-style-type: none"> - Atributos para apresentação de RVs - Seqüências de objetos 3D (Filme Virtual)

A compressão está relacionada ao formato de mídia utilizado, e tem grande importância para a transmissão de objetos 3D, pois minimiza a largura de banda necessária para a transmissão.

O padrão MPEG-4 suporta compressão através dos melhores algoritmos de compactação existentes atualmente.

Quanto à taxa de quadros por segundo renderizada no terminal cliente, que é um dos requisitos de maior importância na representação de mundos virtuais, ela deve proporcionar ao usuário a sensação de animação. A taxa mínima na qual um cérebro humano relaciona

um fluxo de imagens congeladas para percepção de uma animação contínua é de vinte imagens por segundo. A taxa mínima aceitável em ambientes virtuais é de 10 qps (CAPIN, 1998). Entretanto, esse número é questionável e são encontradas na literatura taxas mínimas que variam de 5 a 10 qps.

3. O PADRÃO MPEG-4

O MPEG-4 (N4030, 2007) é um padrão da ISO/IEC 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 televisão digital). Este padrão advém de um esforço internacional que envolve centenas de pesquisadores e engenheiros de todo mundo, movidos pelo desafio da provisão de serviços de televisão digital, aplicações gráficas interativas e multimídia interativa, em especial na Web. Trata dos aspectos de composição, sintetização, compressão, sincronização e distribuição de objetos audiovisuais, além de tratar da descrição das cenas.

Uma cena no padrão MPEG-4 é composta por objetos audiovisuais, que são referidos como fluxos elementares. Estes são dados encadeados recebidos da saída do *buffer* de um codificador, independentemente de seu conteúdo. A integridade de um fluxo elementar é assumida a ser preservada de maneira fim-a-fim entre dois sistemas. Ele contém uma representação codificada do conteúdo de dado, que pode ser: objetos de áudio ou de vídeo, ou informações de descrição da cena (BIFS), ou informações de envio para identificar fluxos ou para descrever as dependências lógicas entre os fluxos (descritores), ou informações relacionadas ao conteúdo do objeto (Fluxos OCI – *Object Content Information*).

Os fluxos elementares são produzidos e consumidos pelas entidades: camada de compressão, codificador e decodificador.

A representação de uma cena MPEG-4 se faz através da especificação BIFS, que fornece um *framework* completo para a máquina de apresentação nos terminais MPEG-4. BIFS suporta uma mistura de várias mídias MPEG-4, como gráficos 2D e 3D, tratamento de interatividade e negociação com mudanças locais ou remotas da cena. Ele foi projetado como uma extensão da linguagem VRML 2.0 num formato binário. Todos os objetos audiovisuais contidos numa cena MPEG-4 são representados na forma de uma árvore hierárquica. Nenhum outro formato de cena abrange todos os requisitos da máquina de apresentação MPEG-4. Os conceitos principais dirigidos ao projeto da especificação BIFS são os seguintes (FAQBIFS, 2007):

- Integração de mídia sintética 2D e 3D num único formato: O desenvolvedor de conteúdo possui uma maneira de projetar conteúdo multimídia completo, sem o desagrado de negociar com muitos formatos diferentes, para evitar a carga do

misto de múltiplos formatos de mídia;

- **Ambiente de Streaming:** Todos os formatos de descrição de cena são projetados de tal forma que a cena completa tem de ser descarregada antes que qualquer coisa possa ser visualizada no terminal. No MPEG-4, os terminais são ligados a um ou vários servidores MPEG-4. A descrição da cena, como qualquer outra mídia, é encadeada para o terminal do usuário, permitindo que a cena seja “quebrada em pedaços”, que são encadeados ao terminal, bem como seus parâmetros de animação, fornecendo um modelo mais eficiente de transmissão, que emparelha com os requisitos usuais do MPEG-4. Quando se lida com aplicações de comunicação, tais características de streaming também são necessárias para enviar novos dados ao usuário durante a comunicação;
- **Compressão:** Muitas representações de cena existentes aparecem no formato textual, tornando-as editáveis, mas muito ineficientes em termos de tamanho de dado. O tamanho das cenas é, freqüentemente, muito menor que outras mídias. Entretanto, cenas complexas podem ter um conjunto de dados muito grande (vários megabytes). Até para cenas pequenas, ser capaz de reduzir o tamanho do dado pode trazer melhoras significantes no momento da transmissão, especialmente para ambientes de taxas de bit baixo ou broadcast, nos quais a cena tem que ser transmitida repetidamente. Além disso, dados de animação também podem ser encadeados em MPEG-4. A compressão eficiente de tais dados reduz significativamente as taxas de bit. Um exemplo típico: Um fluxo BIFS-Anim que consome 10 kbit/s, sem a compressão de dados deveria consumir mais que 120 kbit/s.

O Sistema MPEG-4

O MPEG-4 descreve um sistema para comunicação de cenas audiovisuais interativas compostas: pela representação codificada de objetos 2D/3D, naturais ou sintéticos, que podem ser manifestados áudio e/ou visualmente, denominados objetos de mídia; pelo posicionamento espaço/temporal de objetos de mídia, bem como seus comportamentos em resposta à interação (descrição da cena); e pelas informações relacionadas ao gerenciamento de fluxos (sincronização, identificação, descrição e associação de conteúdo).

A completa operação de um sistema de comunicação com as cenas audiovisuais é descrita assim: no lado do transmissor, as informações das cenas audiovisuais são comprimidas, suplementadas com informações de sincronização e passadas para a camada de entrega que multiplexa as cenas audiovisuais em um ou mais fluxos binários codificados, denominados Fluxos Elementares, os quais são transmitidos ou armazenados. No receptor, esses fluxos são desmultiplexados e descomprimidos. Os objetos de mídia são compostos

de acordo com a descrição da cena e as informações de sincronização, e apresentados ao usuário final, o qual pode interagir com a apresentação. Informações de interação podem ser processadas localmente ou enviadas ao transmissor.

O Sistema MPEG-4 define os seguintes elementos:

- Um modelo de terminal para o gerenciamento de buffer e tempo;
- Uma representação codificada de informações de descrição da cena audiovisual interativa (BIFS);
- Uma representação codificada da identificação e descrição de fluxos audiovisuais, 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 audiovisual descritiva (informação de conteúdo do objeto - OCI);
- Um motor de aplicação.

Arquitetura do Padrão MPEG-4

A representação de informações descreve uma cena audiovisual interativa, contendo informações audiovisuais e informações associadas à descrição da cena. A entidade que recebe e mostra a representação codificada de uma cena audiovisual interativa é genericamente referida como um “terminal áudio-visual” ou apenas “terminal”, que pode corresponder a uma aplicação *standalone* ou ser parte de um sistema de aplicação.

As operações básicas realizadas no terminal seguem: informações que permitem acesso ao conteúdo são providas como informações de estabelecimento de sessão inicial para o terminal. A parte 6 da especificação MPEG-4 define os procedimentos para estabelecer tais contextos de sessão, bem como a interface para a camada de entrega que genericamente abstrai o meio de armazenamento e transporte. A informação de estabelecimento inicial permite, num processo recursivo, localizar um ou mais Fluxos Elementares, que são parte da representação do conteúdo codificado. Alguns deles podem ser agrupados, usando-se uma ferramenta de multiplexação (FlexMux).

Cada Fluxo Elementar contém somente um tipo de dado. Os fluxos elementares são decodificados usando-se seus decodificadores de fluxo específicos. Os objetos audiovisuais são compostos de acordo com as informações da descrição da cena e mostrados à camada de apresentação do terminal. Todos esses processos são sincronizados de acordo com o Modelo de Decodificação do Sistema (SDM), usando-se a informação de sincronização provida na camada de sincronização.

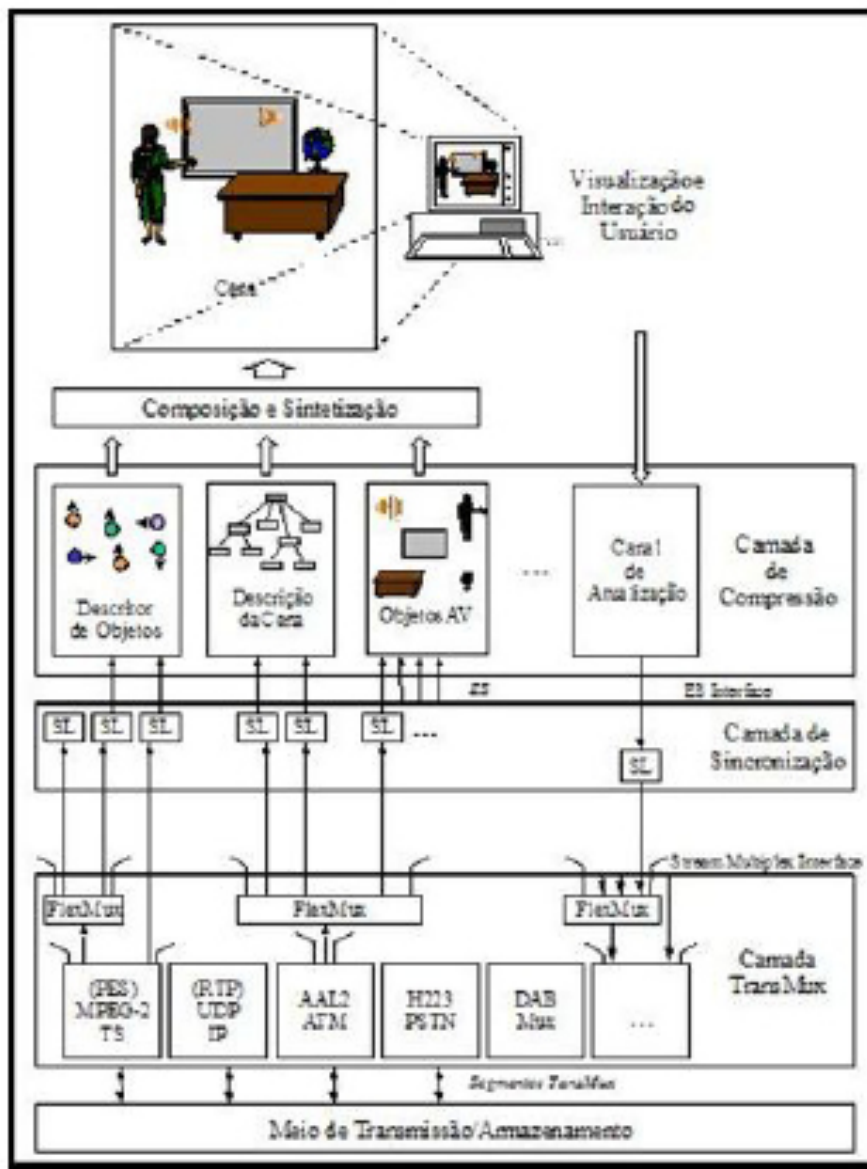


Figura 3 - Arquitetura do Padrão MPEG-4 (N4030, 2007)

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

O SDM fornece uma visão abstrata do comportamento de um terminal conformante com 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 da reconstrução da informação audiovisual que engloba a apresentação. O SDM inclui um modelo de tempo e um modelo de *buffer*.

- Modelo de Tempo: Define os mecanismos pelos quais um receptor estabelece a noção de tempo que o capacita a processar eventos de dependência de tempo. Ele permite ao receptor estabelecer mecanismos para manter a sincronização em ambos os lados e de acordo com os tipos de mídia particulares, bem como nos eventos de interação do usuário. O modelo de tempo requer que os fluxos de dados transmitidos contêm informações de tempo explícitas ou implícitas.

Dois conjuntos de informação de tempo são definidos: referências de clock e time stamps (marcas de tempo). As referências de clock são usadas para transportar os tempos bases do transmissor para o receptor. Os time stamps são usados para transportar o tempo (em unidades de tempos bases do transmissor) para eventos específicos, tais como a decodificação desejada ou tempo de composição de “pedaços” de informação audiovisual decodificados.

- Modelo de Buffer: Capacita o transmissor a controlar e monitorar os recursos de buffers que são necessários para decodificar cada Fluxo Elementar individual na sessão. Os recursos de buffers requeridos são transportados para o receptor por meio de descritores de Fluxos Elementares no início da sessão, de modo que ele possa decidir se é ou não capaz de tratar esta particular sessão. Tal modelo permite ao transmissor especificar quando as informações são removidas desses buffers e agendar a transmissão do dado, de modo que não ocorra transbordamento (overflow).

Camada de Entrega - Multiplexação de Fluxos

O termo camada de entrega é usado como uma abstração genérica das pilhas de protocolos de transporte das camadas de entrega existentes que podem ser usadas para transmitir e armazenar conteúdo. Somente a interface dessa camada, denominada Interface de Aplicação DMIF (DAI), é especificada 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. Existe uma ampla variedade de mecanismos de entrega abaixo dessa interface. Eles servem tanto para a transmissão como para o armazenamento de fluxos de dado, ou seja, um arquivo é considerado uma instância particular da camada de entrega.

Para aplicações em que a facilidade de transporte desejada não endereça completamente as necessidades de um serviço de acordo com a especificação do MPEG-4, uma simples ferramenta de multiplexação (FlexMux) com baixo atraso (*delay*) e baixa sobrecarga (*overhead*) é definida.

Camada de Sincronização de Fluxos – (Sync Layer - SL)

Os Fluxos Elementares são a abstração básica de qualquer origem de dado. Fluxos Elementares são transportados como fluxos SL empacotados (SyncLayer-packetized – SL-Packet) na Interface de Aplicação DMIF. Esta representação empacotada, adicionalmente, fornece 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 camada de compressão divide-se em:

- **Framework de descrição de objetos:** A proposta do framework de descrição de objetos é identificar e descrever os fluxos elementares e associá-los apropriadamente, a uma descrição da cena audiovisual. Os descritores de objetos servem para acessar um conteúdo, o qual contém a informação e a interface para o gerenciamento e proteção de propriedades intelectuais.
- **Gerenciamento e proteção de propriedades intelectuais (IPMP):** O framework IPMP para conteúdo MPEG-4 consiste em interface normativa que permite um terminal MPEG-4 unir a um ou mais sistemas IPMP. A interface IPMP consiste em um fluxo elementar IPMP e em descritores IPMP. Os descritores IPMP são levados como parte de um fluxo de descritor de objeto. Fluxos elementares IPMP levam informações de variação de tempo que pode ser acessada por múltiplos descritores de objeto.
- **Informação de conteúdo de objetos (OCI):** Os descritores OCI transportam informações descritivas sobre os objetos audiovisuais. Os principais descritores de conteúdo são: descritores de classificação de conteúdo, descritores de palavras-chaves, descritores de taxa, descritores de linguagem, descritores textuais e descritores de criação de conteúdo.
- **Fluxos de descrição da cena:** A descrição da cena endereça a organização dos objetos audiovisuais em uma cena, em termos de atributos espaço/temporal. Esta informação permite a composição e renderização de objetos audiovisuais individuais, os quais são reconstruídos por decodificadores específicos em forma de fluxo de dados.
- **Fluxos Audiovisuais:** As representações codificadas de informação audiovisual são descritas na especificação do padrão, e os dados audiovisuais reconstruídos são disponibilizados para o processo de composição para uso potencial durante a renderização da cena.
- **Fluxos Upchannel:** Fluxos elementares downchannel podem requerer informação upchannel transmitidas de um terminal receptor para um terminal emissor (por exemplo, permitir a interação cliente-servidor).

Motor de Aplicação

O MPEG-J é um sistema programático que especifica API para interoperação dos players de mídias MPEG-4 com código Java. Pela combinação de mídias MPEG-4 e código executável seguro, criadores de conteúdo podem embutir controle complexo e mecanismos de processamento de dados com seus dados de mídias para gerenciamento inteligente de operações da sessão audiovisual.

A aplicação Java é entregue como um fluxo elementar separado para o terminal MPEG-4. A aplicação será direcionada para um ambiente de tempo de execução MPEG-J, de onde o programa MPEG-J terá acesso a vários componentes e dados requeridos do player MPEG-4 para seu controle.

Comandos BIFS para Atualização de Objetos

Os BIFS-Commands, que são chamados também de BIFS-Updates, são usados para modificar um conjunto de propriedades da cena gráfica em um determinado instante de tempo, podendo tratar-se da sincronização, atualização e modificação de uma cena, dos nós e campos (atributos) de nós através de pacotes BIFS em forma de unidades de acesso, que especificam a forma como a cena será atualizada ou modificada

- Inserção: Usado para inserir nós, valores (que podem ser multivalorados) e rotas.
- Deleção: Pode ser usado para remover nós, valores e rotas.
- Substituição: É usado para substituir um nó (por outro qualquer que seja do mesmo tipo), valores e rotas.
- Substituição da Cena: Ele remove, por completo, toda a cena gráfica, juntamente com todos os objetos audiovisuais e rotas previamente criadas para a adição de um novo conteúdo (uma nova cena).

A figura 4 mostra todos os possíveis comandos *BIFS*.

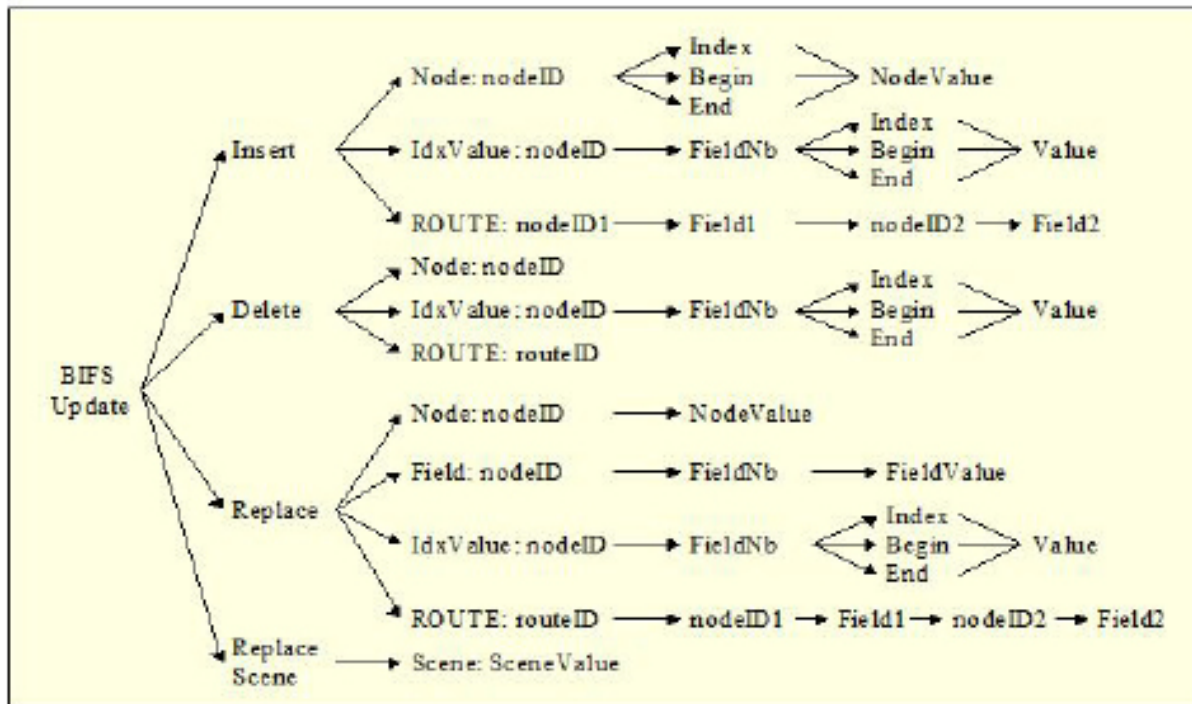


Figura 4 - Estrutura BIFS (N4264, 2007)

4. O PADRÃO MPEG-4 MU NO SUPORTE A AMBIENTES VIRTUAIS MULTIUSUÁRIO

O padrão MPEG-4 possui um formato de arquivo binário, o BIFS (*Binary Format Scenes*), que encapsula os segmentos elementares (*streams* de dados) do conteúdo da cena para apresentá-los no visualizador MPEG-4 (*Player*). Isso faz com que o tamanho final do arquivo contendo o ambiente virtual seja pequeno e, portanto, mais rápido,

Nós Multiusuários do padrão MPEG-4 MU

Os principais nós multiusuários são *MUSession* e o *MUZone*. O *MUSession* é definido como o nó raiz dos componentes compartilhados em uma cena MPEG-4; é ele também que define a localização do computador que controla a respectiva seção multiusuário que é definida por este nó. O nó *MUZone* divide a cena compartilhada em várias pequenas subcenas que podem ter diferentes permissões de acesso e diferentes servidores controlando a sincronização e atualização dos nós delas (VASSILIADIS, 2001).

Os nós *MUAvatar* e *MUAvatar2D* são a representação do usuário no AVM; eles são diretamente relacionados com o nó *MUSession*.

Arquitetura Do Padrão MPEG-4 MU

A arquitetura do padrão multiusuário define um *framework* no qual ambos distribuidores de *softwares* e provedores de conteúdo possam implementar suas próprias soluções multiusuários. Este *framework* define os níveis de interoperabilidade, que implicam no formato do conteúdo multiusuário, e o protocolo de mensagens para carregar dados multiusuário (N4030, 2007)

A interoperabilidade deve ser definida nos seguintes níveis:

- **Objetos compartilhados:** Objetos arbitrários que podem ser declarados como compartilhados, enviados para o servidor e incorporados em um ambiente multiusuário.
- **Mundo:** Ambientes criados por diferentes provedores de conteúdo, que podem ser visualizados em diferentes browsers.
- **Browser:** Qualquer browser que pode ser usado para visualizar conteúdos multiusuários provido por qualquer servidor.

O protocolo de mensagens, juntamente com uma infra-estrutura para compartilhamento de subárvores e grafos de cena entre múltiplos usuários, define o sistema de comunicação do padrão MPEG-4 MU. Este sistema estipula como os nós podem ser individualmente compartilhados e como as subárvores, que são compartilhadas, podem ser protegidas contra modificações não desejadas de fontes externas.

Ainda não foi definido no padrão como os nós devem reagir em ambientes distribuídos. Regras para cada tipo de nó devem ser estabelecidas no padrão para que os desenvolvedores saibam como implementar *softwares* que suportam a tecnologia multiusuário.

Segundo o documento “*a Proposal for an MPEG-4 multiusers worlds architecture and tools*” (M3874, 2001), aprovado e adotado pelo padrão MPEG-4, a arquitetura multiusuário do padrão emergente MPEG-4 MU, está amplamente baseada no mecanismo **Pilot/Drone** proposta pela especificação *Living Worlds* (LIVING, 2001). De acordo com a especificação citada, os *Pilots* são cópias mestre de nós em terminais 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*.

Todo 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. Entretanto, esse conceito pode ser aplicado a qualquer objeto compartilhado. A figura 5 representa a sincronização entre o mecanismo *Pilot/Drone*. Para melhor compreensão, os avatares serão representados por **An**, que são controlados por **Pn** através das instruções dos usuários diante de cada Terminal (*Host*), onde cada terminal é capaz de exibir todos os avatares participantes do ambiente.

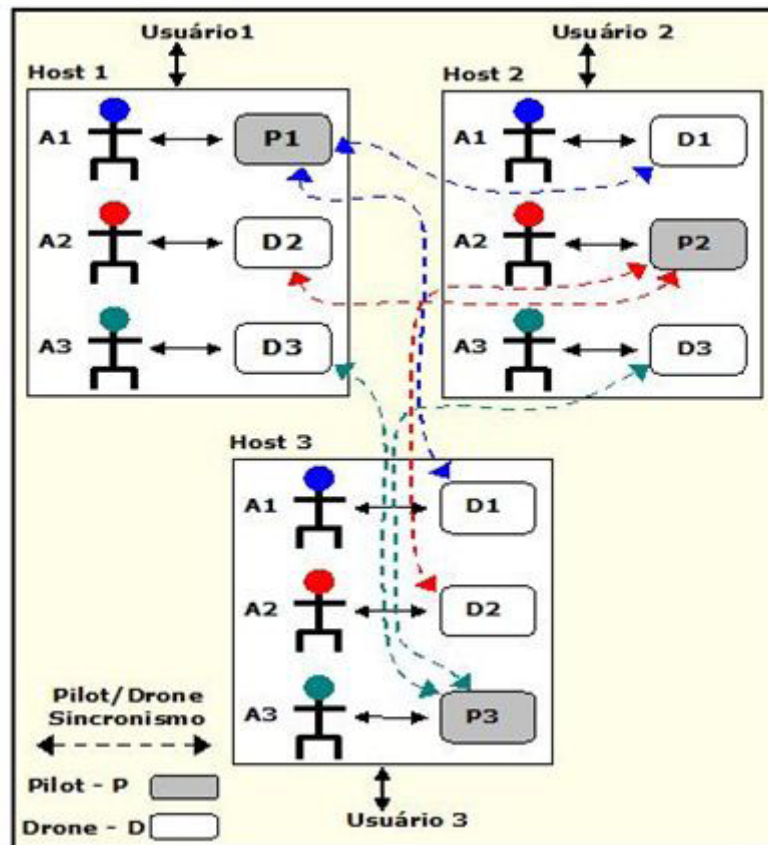


Figura 5 - Mecanismo de Sincronização entre *Pilot* e *Drone*

Para a sincronização da cena gráfica, há 2 cenários possíveis que são tratados para atualizar as cenas quando determinadas mudanças ocorrem nos terminais, ou seja, a forma Peer-to-peer (ponto-a-ponto) ou centralizado usando um Bookkeeper.

1. *Peer-to-peer*: Neste cenário, a sincronização é determinada pelo terminal que está pilotando um objeto compartilhado.

Um *pilot* recebe uma mensagem requisitando mudanças (*RequestChangeMessages*) de *drones*, aceitando ou não a atualização requisitada. Em caso positivo, distribui comandos de atualização para todos os *drones* e para si mesmo. Um *pilot* pode também receber requisições locais, por exemplo, por meio da API MPEG-J, e tratá-las da mesma forma.

Nesse cenário, os mecanismos devem ser definidos tal que um novo usuário possa pesquisar a existência e estados dos objetos pilotados por outros terminais, e caso haja a saída anormal de um usuário que esteja pilotando um objeto, o sistema deve detectar o fato e tratar apropriadamente a adequação da cena.

Centralizado usando um Bookkeeper: Uma ou várias instâncias *bookkeeper* são responsáveis por manter a consistência dos estados dos objetos compartilhados. Se o *bookkeeper* está pilotando o objeto, ele recebe mensagens *RequestChanges*, e distribui comandos de atualização. Se o *bookkeeper* não está pilotando o objeto, ele roteia a mensagem de requisição para o usuário que possui o *pilot*, este envia os comandos de atualização que são distribuídos pelo *bookkeeper* para os respectivos *drones*.

Componentes Da Arquitetura MPEG-4 MU

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

A figura 6 mostra os componentes da arquitetura MPEG-4 MU, destacando (envolvidos por uma linha tracejada) os elementos que são deixados para os fabricantes de navegadores e de sistemas para a *web*, e os elementos especificados pelo padrão.

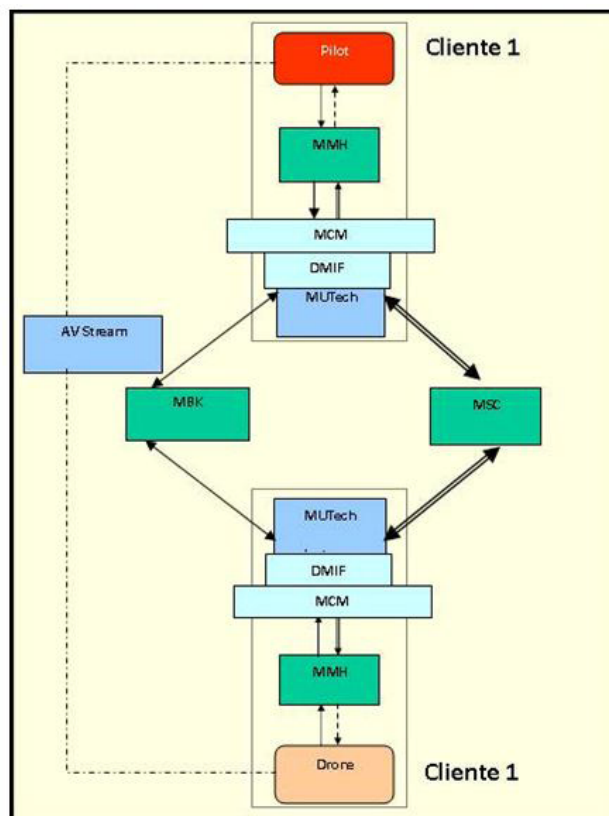


Figura 6 - Arquitetura do Padrão MPEG-4 MU (N4415, 2007)

MUTEch Session Controler (MSC)

O MSC realiza a configuração do canal de comunicação entre clientes ou entre cliente e *bookkeeper*, atribui os *ClientsIDs*, fornece suporte a *multicast/unicast* quando disponível, mantém uma lista de quais clientes entraram em uma determinada zona em um determinado instante de tempo (esta informação pode ser requisitada por clientes ou ser usada pelo *bookkeeper*), e mantém o endereço de rede dos clientes. Essas informações são críticas para o roteamento de mensagens. Cada cliente mantém um canal de controle permanente com o MSC. Este fornece aos clientes uma lista com todas as zonas existentes, notificações de entrada e saída de usuários e criação e remoção de zonas. Também controla o acesso a zonas em sessões, manipula requisições de clientes para entrar e sair de zonas, aceitando ou não a requisição. E ainda inicia o *MUTEch Bookkeeper*.

MUTech Bookkeeper (MBK)

A responsabilidade do MBK é sincronizar o estado dos objetos compartilhados. Sua principal responsabilidade é rotear mensagens pilot-drone. Ele pode tomar a responsabilidade da pilotagem de um objeto compartilhado dentro de uma zona, ou pode transferi-la para um cliente sob requisição. O MBK sabe qual é o pilot de um determinado objeto compartilhado em qualquer instante de tempo, e também armazena um completo gafo de cena contendo todas as zonas e nós dentro de sessões.

MUTech Message Handler (MMH)

O MMH tem como propósito:

- Prover uma interface com o terminal do cliente para envio e recepção de mensagens multiusuário;
- Agrupar várias mensagens multiusuário para formar pacotes SL;
- Inserir nas mensagens o correto ClientID antes que sejam enviadas por meio do canal Request/Change aberto no DMIF.

O MMH disponibiliza uma interface (MAI, Multi-User Application Interface) para ser usada por componentes da aplicação, como o EAI e o MPEG-J, que irão enviar uma requisição relacionada aos nós que serão modificados. A requisição deve ser roteada via MMH para a validação pelo pilot, conforme ilustra a figura 7.

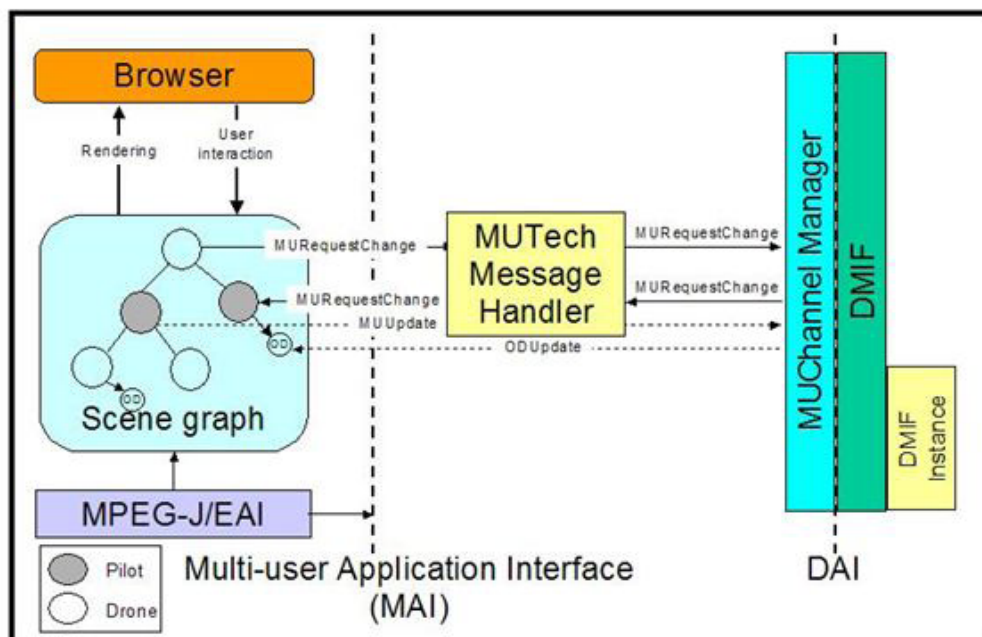


Figura 7 – Funcionalidades do MMH (N4415, 2007)

MUChannelManager (MCM)

MCM é um componente auxiliar para configurar *streams* de dados e controle entre cliente e MSC, por meio da interface DAI do DMIF, uma vez que é necessário um canal constante entre cliente e MSC. Os atributos desse canal serão descritos usando-se a sintaxe de *Object Descriptor* e *Elementary Stream Descriptor*, que permite a inclusão de parâmetros, tais como direção, informação de QoS, direitos autorais e informações de conteúdo dos objetos .

Mensagens multiusuário são enviadas via MCM e DMIF por meio de canais abertos no servidor DMIF e módulos de cliente. Elas estarão em forma de *streams*, que serão associadas com um *Object Descriptor* para formar o tipo *MUCommand Stream*. Cada Sessão, definida pelo nó *MUSession*, tem um *Object Descriptor* contendo uma URL para o *MUSessionController* associado.

- Uma ESD para descrever a stream que carrega os MU-commands no sentido servidor-cliente;
- Uma segunda ESD descrevendo a stream que carrega os comandos multiusuário no sentido cliente-servidor;

O DMIF *streaming server* provê suporte em tempo real para mensagens entre cliente e servidor por meio da API(dll) LIVE and SINK, que são usadas para comunicação servidor-cliente e cliente-servidor respectivamente.

A figura 8 ilustra as chamadas ao DMIF.

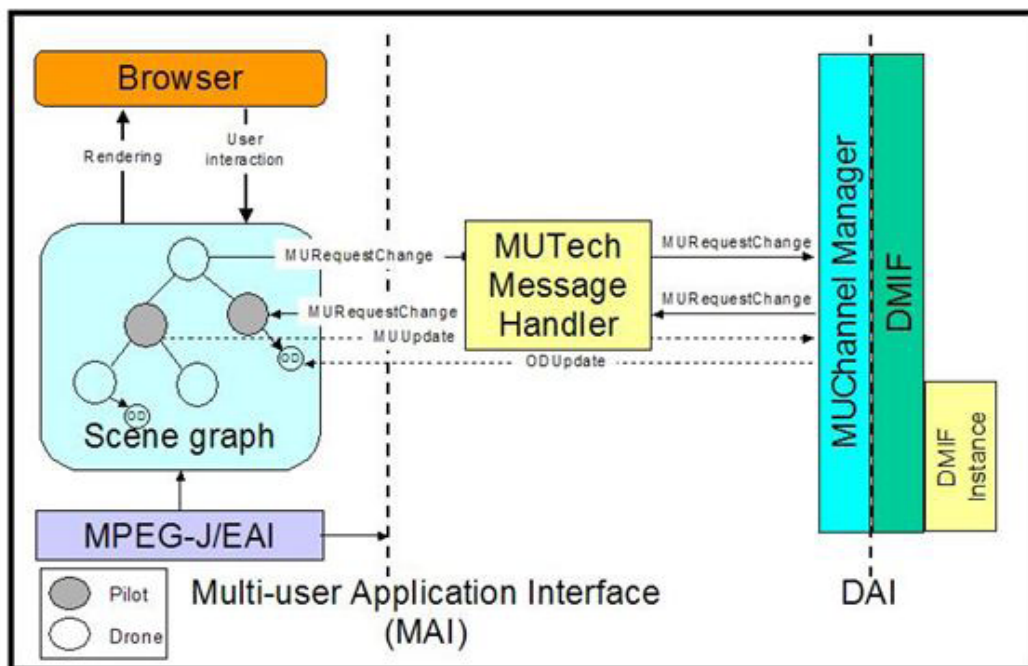


Figura 8 – Chamadas ao DMIF (N4415, 2007)

MUTech Instance

Esta é uma instância do DMIF que permite uma aplicação abrir canais de mensagens e transferi-las de e para o MSC. Essa instância segue a especificação do DPI (*DMIF Plug-in Interface*), e faz a comunicação com o MSC transparente para a aplicação. Em adição, ele pode suportar comunicação ponto a ponto entre clientes, já que, neste caso, o MSC e o MBK não são usados para roteamento de mensagens.

O *Mutech Instance* pode ser implementado por uma instância DMIF; outra opção é entregar as *elementary stream* requisitadas pelo MMH multiplexadas junto com a cena áudiovisual principal.

AV Stream Server

Este módulo é uma abertura na arquitetura para a inserção de um servidor que forneça os nós das cenas como *streams* AV MPEG-4 referenciadas pelos *Object Descriptors*. Essa abordagem permite o uso de *unicast* ou *multicast*, arquivos locais, distribuição e compartilhamento de *streams* entre terminais de clientes

5. CONSIDERAÇÕES FINAIS

Na criação de Ambientes Virtuais Multiusuários um dos principais aspectos que impactam seu desempenho é propagação de mensagens de atualizações entre os objetos ali compartilhados.

O padrão emergente MPEG-4 MU utiliza comandos de atualização que são transmitidos pela rede de forma diferente dos sistemas tradicionais de suporte a AVMs. Estes comandos são transmitidos na forma de 0's e 1's compactados o que contribui para diminuir a latência na rede na sincronização do AVM entre todos os participantes. Nesse sentido, a propagação de mensagens de atualização no ambiente ocorre de maneira satisfatória e transparente ao usuário.

REFERÊNCIAS

ARAÚJO, R. B., **Especificação e Análise de um Sistema Distribuído de Realidade Virtual**, Tese de Doutorado, EPUSP, Julho, 1996.

Tolga CAPIN, Maja JOVOVIC, Joaquim ESMERADO Amaury AUBEL, Daniel THALMANN - "Efficient Network Transmission of Virtual Human Bodies" Computer Graphics Laboratory, **Swiss Federal Institute of Technology (EPFL)** CH-1015 Lausanne, Switzerland, JULHO 1998.

FAQ - BIFS - <http://www.cselt.it/mpeg/faq.htm>. Último acesso em 2007

Living Worlds Proposal Draft 2, http://www.vrml.org/WorkingGroups/living-worlds/draft_2/index.htm.

(M3874, 2001) J. Arthur et al. **Proposal for an MPEG-4 multi-user worlds architecture and tools**, January 2001.

Thomas A. MUTH, "The Virtual Reality Benchmark in Telecommunications:"
International Telecommunications Society **XIII BIENNIAL CONFERENCE** March 15,
2000. (N4030, 2007) Overview of MPEG-4 Standard, (Sydney Version), August 2007.

Overview of MPEG-4 Standard, (V.18 - Singapore Version), March 2001.

GRINI, I et al. **Proposal for an MPEG-4 multi-user worlds architecture and tools**,
December 2001.

(VANDAN, 1993) VAN DAM, A., VR as a Forcing Function: Software Implications of a
new Paradigm, **IEEE Symposium on Research Frontiers in Virtual Reality**, October,
25-26, CA, 1993.