

# Uma comparação entre as principais tecnologias de suporte ao desenvolvimento de ambientes virtuais multiusuário

Matheus Pessote Verdi

Mestrado em Ciência da Computação - Universidade Federal de São Carlos  
Coordenador do curso de Ciência da Computação  
da Faculdade Comunitária de Rio Claro  
matheus.verdi@unianhanguera.edu.br

Marcelo Zani

Mestre em Informática - PUCCAMP  
Coordenador do curso de Ciência da Computação  
do Centro Universitário Anhanguera - Leme  
marcelo.zani@unianhanguera.edu.br

## Resumo

Com o impacto sócio-econômico provocado pela *WWW* (*World Wide Web*), novas tecnologias estão sendo consideradas para tornar ainda mais natural a interação homem-máquina dentro do ambiente da *Web*. A realidade virtual de mesa, que utiliza apenas de um computador convencional, ou seja, teclado, mouse, gabinete (UCP) e monitor, é uma das tecnologias cujas limitações estão sendo superadas, para que múltiplos usuários possam compartilhar um mesmo ambiente virtual tridimensional sintetizado pelo computador, em aplicações que vão de treinamento a entretenimento. Nesse sentido, várias tecnologias têm sido desenvolvidas para implementar ambientes virtuais multiusuário cada vez mais consistentes e mais próximos da realidade.

**Palavra-Chave:** RV, VRML, X3D, JAVA 3D e MPEG-4

## Introdução

Os ambientes virtuais são a representação de um ambiente real ou fictício sintetizado pelo computador.

O desenvolvimento destes pode ser obtido por diversas tecnologias conforme a natureza de aplicação do mesmo. Entretanto, devido a uma necessidade mais completa de suporte à codificação, transmissão e apresentação de múltiplas mídias e compartilhamento, que é a base dos ambientes virtuais multiusuários,

## Abstract

With the large scale reach and the social and economic impact provoked by the *WWW* (*World Wide Web*), new technologies are being considered to make man-machine interaction in the *Web* environment even more natural. Desktop virtual reality, that uses only of a conventional computer, that is, keyboard, mouse, cabinet (CPU) and monitor, is one of the technologies whose limitations are being overcome, so that multiple users can share one same three-dimensional virtual environment. This synthetic environment is rendered by the computer, in applications that stand from training to entertainment. In this direction, some virtual technologies have been developed to implement multiuser virtual environment each time more consistent and next to the reality.

**Key-words:** RV, VRML, X3D, JAVA 3D e MPEG-4.

alternativas, como *Java 3D*, *X3D* e o padrão emergente *MPEG-4*, surgiram para o suporte à implementação de ambientes virtuais multiusuários na *Web*.

O objetivo principal deste artigo é comparar e relatar os diferenciais das principais tecnologias que estão sendo adotadas como padrão ao suporte e desenvolvimento dos ambientes virtuais multiusuários em ambientes distribuídos como a Internet.

## Realidade Virtual

A realidade virtual (RV) pode ser definida como a interface mais natural e poderosa de interação homem-máquina, por permitir ao usuário imersão, interação e navegação em um ambiente tridimensional sintetizado pelo computador, através de canais multissensoriais, tais como: visão, audição, tato, olfato, paladar, etc (ARAUJO, 1996).

A RV pode ser classificada como imersiva e não imersiva: a primeira é baseada no uso de salas de projeção, paredes ou capacetes, onde, o usuário se sente “presente” dentro do ambiente virtual. Entretanto, um ambiente virtual pode ser sintetizado também por meio de monitores e em projetos para uso de telefones celulares e TVs interativas, e nesse caso, a RV é considerada não imersiva. As figuras 1 e 2 exemplificam os conceitos previamente citados.

Na interação, sua estrutura fundamenta-se sob diferentes aspectos, mas em um enfoque geral nota-se que o usuário interage com a RV sob diferentes dispositivos de entradas do usuário e por modificações feitas no ambiente, que são detectadas pela aplicação.

A navegação está relacionada ao grau de motivação para o engajamento de uma pessoa em uma determinada atividade, podendo esta ser passiva, como ler um livro ou assistir televisão, ou ativa, como na participação de um jogo com algum parceiro. A realidade virtual tem potencial para os dois tipos de navegação ao permitir a exploração de um ambiente virtual e ao propiciar a interação do usuário com um ambiente virtual dinâmico (BRANDÃO, 1998).



Figura 1: RV Imersiva



Figura 2: RV não imersiva

A RV tem sido utilizada por profissionais de diversas áreas como: planejamento urbano, prospecção de petróleo, geologia, entre outras. Estas áreas têm exigido um grande esforço no desenvolvimento de aplicações que facilitem a visualização e análise de informações nos mais diversos formatos.

Embora essas tecnologias possam trazer um diferencial para as áreas já citadas, autores como Fernandes (FERNANDES, 2005) acreditam que ainda existem alguns fatores que restringem o uso prático das tecnologias de mundos virtuais 3D na *Web*, tais como:

- **Poder Computacional:** as plataformas capazes de criar interações 3D em tempo-real ainda são dispendiosas e complexas. Por outro lado, experiências focalizadas na solução de problemas mais específicos e restritos produzem efeitos satisfatórios, por exemplo, a criação de shoppings virtuais ou jogos interativos.

- **Fatores Humanos:** a capacidade humana de tratar multidimensionalidade sugere que o tridimensional, embora atraente aos olhos, não traga necessariamente mudanças qualitativas e quantitativas à interação.

- **Ausência de Ferramentas e Ambientes de Desenvolvimento Adequados:** a falta de ferramentas e ambientes adequados para o desenvolvimento contribui para transformar os mundos virtuais 3D em vitrines dispendiosas que oferecem poucas possibilidades de exploração interações através de interações em tempo-real. Já Teichrieb (TEICHRIEB, 1999) observa que crescem o número e o uso de *softwares* especializados que oferecem bibliotecas com modelos 3D e texturas predefinidas, com opções para criação de novos objetos, o que facilita a modelagem de mundos virtuais.

Apesar das limitações mencionadas, o potencial da aplicação de RV na *Web* tem levado pesquisadores do mundo todo a explorar novas tecnologias e/ou promover a evolução das tecnologias existentes.

### Tecnologias para Suporte a Ambientes Virtuais na *Web*

Com a explosão de utilização da *WWW* no início dos anos 90, cujo sucesso se deve principalmente ao uso de hipermídia para a navegação e visualização das informações de forma amigável, a implementação de ambientes virtuais ali compartilhados, tem se tornado viável tanto que tecnologias de suporte à RV na *Web* têm surgido, e neste sentido podem-se citar: *VRML* e

## VRML - The Virtual Reality Modeling Language

O VRML (VRML, 1997) é um formato de arquivo para descrever objetos 3D interativos e mundos tridimensionais.

Projetado para ser usado na *internet*, *intranets*, e sistemas de cliente locais, também é entendido que o VRML é um formato de intercâmbio universal para gráficos 3D integrados e multimídia. O VRML pode ser usado em uma variedade de áreas de aplicação como engenharia e visualização científica, apresentações de multimídia, entretenimento, títulos educacionais, páginas Web, e mundos virtuais compartilhados.

A primeira versão da especificação da linguagem VRML, baseada no *Open Inventor* (biblioteca em C++ desenvolvida pela *Silicon Graphics*), foi completada em 1995 (VRML 1.0) com a liberação de vários *plugins* para *browsers HTML*. O VRML 1.0 suportava apenas recursos para ambientes virtuais estáticos; entretanto, a versão 2.0 do VRML foi liberada no final de 1996 apresentando uma inovação, ou seja, já estendia a linguagem para suportar animação e interação através dos seguintes recursos: modificação da sintaxe, construção de formas, animação, interação, som, efeitos atmosféricos como neblina, plano de fundo e extensões da linguagem. Em 1997, a especificação do VRML 2.0 foi submetida à Organização Internacional de Padronização - ISO. A versão ISO do VRML 2.0 (VRML 97) foi revista e reescrita com pequenas modificações na linguagem.

O VRML foi projetado para cumprir as seguintes exigências:

- **Autoria:** Habilitar o desenvolvimento de programas de computador capazes de criar, editar, e manter arquivos VRML, como também a tradução automática de programas para converter outros formatos de arquivo 3D, comumente usados, em arquivos de VRML.
- **Composição:** Prover a habilidade para usar e combinar objetos dinâmicos 3D dentro de um mundo de VRML e assim permitir reusabilidade.
- **Extensão:** Prover a habilidade de adicionar novos tipos de objetos não definidos explicitamente no VRML e ser capaz de implementação em um largo alcance de sistemas.
- **Desempenho:** Desempenho interativo em uma larga variedade de plataformas computacionais.
- **Escalabilidade:** Habilitar mundos dinâmicos 3D arbitrariamente grandes.

O VRML é capaz de representar objetos estáticos, dinâmicos, multimídia com *hyperlinks* e outras mídias tais como texto, sons, filmes, e imagens. E nesse sentido, os *browsers* de VRML são amplamente disponíveis para muitas plataformas diferentes, como ferramentas de autoria para a criação de arquivos de VRML.

Nos últimos anos, a linguagem VRML tem sido aceita como uma tecnologia padrão da Web para a exibição de conteúdo gráfico 3D, representando um meio rico de expressão de idéias, uma vez que é interativa e pode suportar animação e som. Exemplos de ambientes virtuais na Web utilizando-se VRML são inúmeros. As figuras 3 e 4 mostram alguns deles. Entretanto, a linguagem VRML apresenta limitações e uma delas é o fato de não possuir um *browser* confiável, pois, eles geram uma complexidade e uma instabilidade que influem diretamente no desempenho e na qualidade da apresentação do cenário, bem como, na integração hipermídia e multimídia.

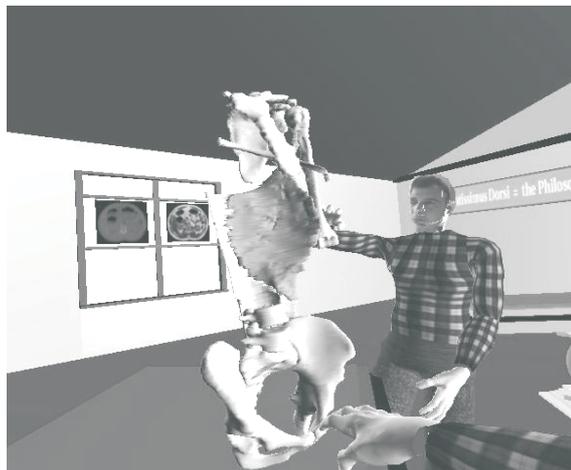


Figura 3: Aula de Anatomia

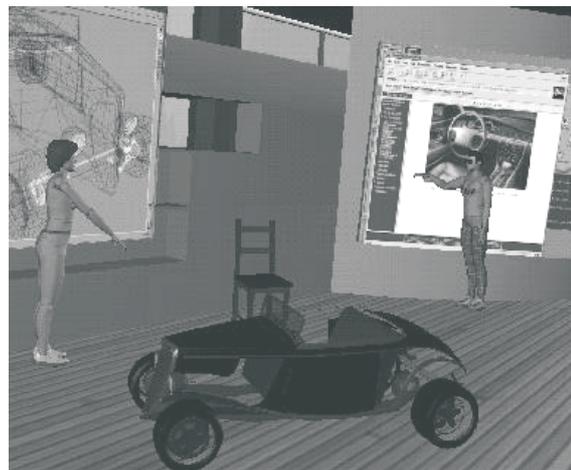


Figura 4: Museu Virtual

As limitações do VRML decorrem do fato de a ferramenta não suportar satisfatoriamente texto, áudio e vídeo. O texto é um objeto 3D de fundo; já o vídeo é

mapeado como textura sobre um objeto 3D e tem que ser completamente carregado para a memória, juntamente com o ambiente virtual, antes de ser apresentado, o que provoca baixa qualidade e sincronismo de áudio quando houver a necessidade da apresentação de mais de um vídeo.

Toda a complexidade da linguagem *VRML* é refletida diretamente na apresentação da cena e conseqüentemente, também na apresentação do dado multimídia. Em ambientes com grande movimentação, pode ser aceitável o compromisso entre qualidade e a taxa de atualização, negociação esta que não é suportada pela linguagem *VRML*. Mais ainda, o *VRML* não possui um formato binário, uma das causas principais de atraso nele, pois, geralmente, arquivos no seu formato são extensos e demoram muito para serem carregados. A figura 5 apresenta um código *VRML* composto por inúmeras linhas responsáveis pela apresentação de um objeto simples que é ilustrado na figura 6.

```
#VRML V2.0 utf8
Shape { # define um objeto
  appearance Appearance { # define a aparência
do objeto
    material Material {
      diffuseColor 1.0 0.0 0.0 # r g b - vermelho
    }
  }
  geometry Box { # define a geometria do objeto
- paralelepípedo (Box)
    size 2.5 2.5 2.5 # largura, altura,
profundidade
  }
} # fim do shape
```

Figura 5: Código VRML (Cubo)

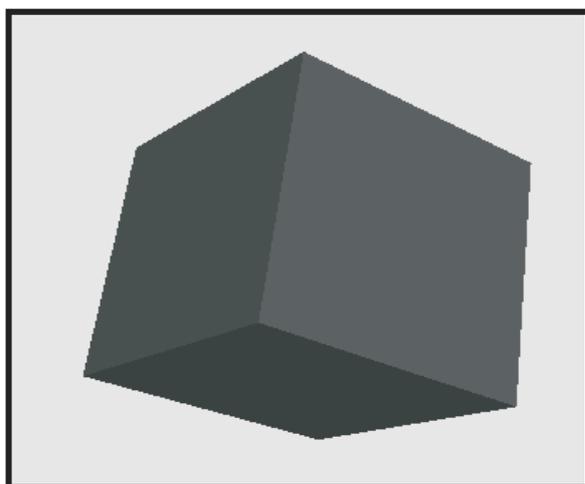


Figura 6: Cubo VRML

Para suprir essas limitações existentes na linguagem *VRML*, outras ferramentas surgem como

alternativa, tais como, *X3D*, a Integração *Java* e *VRML*, *Java 3D* e *MPEG-4*.

### *X3D (Extensible 3D)*

A solução adequada para a distribuição de informações tridimensionais na *Web* deve levar em consideração as limitações de tráfego e uma padronização dos dados que circulam na rede, buscando ainda um equilíbrio no esforço computacional exigido dos clientes e servidores. Para tal, *X3D (eXtensible 3D)* foi desenvolvida pelo *Web 3D Consortium*, com o apoio forte de várias companhias de principais navegadores como *Blaxxun*, *Nexternet*, *OpenWorlds* e *ParallelGraphics*. Considerado um revolucionário passo na trajetória do *VRML*, *X3D* é, para todos os propósitos, a próxima geração do *VRML* (X3D, 2005).

*X3D* carrega na sua essência características intencionadas para serem usadas dentro de ambientes distribuídos tais como a *World Wide Web*. Existem vários objetos e mecanismos construídos dentro da linguagem para o suporte a arquivos múltiplos distribuídos incluindo:

- Arquivos *VRML*;
- *Hyperlinks* para outros arquivos;
- Usar a Internet estabelecida e o padrão *ISO* para outros formatos de arquivos;
- Definir uma codificação compacta para funcionalidades abstratas a fim de satisfazer variedade de usos;
- Acesso a mundos *X3D* para programas externos usando-se a interface de autoria da cena.

O projeto e desenvolvimento do *X3D* é dirigido em grande parte pela demanda da comunidade para a tecnologia 3D, que pode ser usada a fim de construir um extenso percurso para as aplicações básicas na Internet. Desenvolvida para atender uma gama de exigências técnicas, tais como (X3D, 2005):

- Separar a arquitetura em tempo de execução da codificação de dados; suportar uma variedade de formatos codificados, inclusive o *XML (eXtensible Markup Language)*;
- Adicionar novos grafos e objetos com comportamentos interativos; prover uma programação de aplicação alternativa interligada na cena 3D (*APIs*); modularizar a arquitetura em componentes e
- Definir subconjuntos da especificação (“Perfis”); os quais satisfazem as necessidades de diferentes mercados, etc.

As figuras 7 e 8 apresentam respectivamente o Planeta Terra com rotações previamente aplicadas e um

humanóide que possui movimentos alternados representando passos de uma dança.



Figura 7 - Planeta Terra

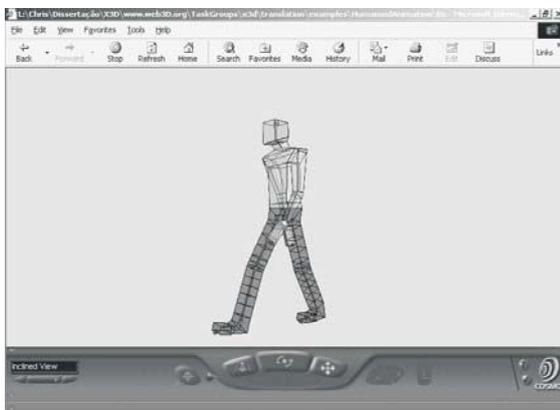


Figura 8 - Humanóide

Em relação às aplicações, X3D pretende estimular produtos de comércio eletrônico e demonstrações de tecnologias, simulação visual e de base de dados, treinamento, multiusuários etc.

A arquitetura de X3D é independente de formatos de dados codificados. X3D suporta múltiplas codificações de dados da cena, incluindo textual (VRML 97 e a sintaxe XML) e binário. Os dados da cena podem ser comprimidos e descomprimidos sem maiores complicações (WALSH, 2001).

Embora X3D seja baseado na VRML 97, ela não é simplesmente um novo caminho de entrega de conteúdo VRML 97. Ao contrário disso, X3D realça a VRML em muitos aspectos, enquanto preserva-se compatível com a padronização. Descreve tecnologias, como superfícies e curvas NURBS (Non Uniform Rational Bsplines), e multitextura - a capacidade de aplicar duas ou mais texturas num único objeto -, características essas que não são mencionadas na VRML 97 (WALSH, 2001).

Não se pode considerar X3D como uma linguagem propriamente de VRML, e sim uma linha de produtos que implementa vários perfis e define um conjunto de objetos, métodos e interfaces (X3D, 2005).

Outras características que serão futuramente adicionadas ao X3D são: gráficos 2D, *streaming* audio e *streaming* vídeo. Essas e outras características, segundo *Web 3D Consortium*, são trabalhos em progresso (WALSH, 2001).

### Integração JAVA e VRML

Nos ambientes virtuais multiusuários alguns controles são necessários. Assim, tanto a movimentação de objetos inanimados, quanto o controle de múltiplos usuários dentro de um ambiente virtual podem ser suportados através de uma integração entre linguagens VRML e Java. Essa integração permite implementar aplicações 3D interativas multiusuário na Internet (ARAÚJO, 1998).

O VRML facilita a interação com a linguagem Java, através da inclusão, no padrão (VRML 2.0), de uma interface de autoria externa - EAI, a qual permite a passagem de eventos que modificam a cena aos *browsers VRML*, através de *applets* escritas em Java.

Essencialmente, a EAI oferece uma forma de interfacear de forma direta e dinâmica as aplicações customizadas num ambiente externo ao mundo VRML, com o *browser*. Uma outra maneira de se comunicar com o *browser* usando-se Java é através de *Scripts (Script Authoring Interface)*, que opera dentro do *browser VRML*.

Ambientes virtuais são construídos, através de modeladores como *3D Studio Max2*, *Cosmo World* e outros, e são convertidos para a linguagem VRML. Para a visualização de arquivos VRML, são necessários *plugins* como: *CosmoPlayer*, *Intervista's World View*, *Community Place (Sony)*, etc. Entretanto, conforme já mencionado na Introdução deste trabalho, o VRML apresenta limitações que estão sendo superadas através de intenso trabalho em desenvolvimento pelo Consórcio W3C. Uma alternativa ao VRML é o MPEG-4, que estende as funcionalidades da linguagem VRML para descrever as cenas audiovisuais, tratando não somente da sua descrição, tal como o VRML faz, mas também se preocupando com os aspectos de composição, sintetização, compressão, sincronização e distribuição delas geradas.

### JAVA 3D

O Java 3D é uma extensão de Java, mais precisamente um padrão de extensão da plataforma Java 2 (JAVA3D, 2005).

Introduzida em meados do ano de 1997 pela Sun

*Microsystems*, *Java 3D* é uma coleção de classes *Java* que define um alto nível de *APIs* (*Application Programming Interface*) para interagir com o crescimento dos ambientes 3D.

Tais versões são compreendidas (WALSH, 2001):

- *Java 3D 1.0*;
- *Java 3D 1.1*;
- *Java 3D 1.2*.

A primeira versão da *API Java 3D*, *Java 3D 1.0*, foi liberada em 1997. Baseada no modelo *VRML*; onde inicialmente obteve suporte à construção interativa gráfica da cena 3D, animações 3D e capacidades básicas de som.

A segunda veio para sanar problemas de performance decorrente da implementação e atualmente a versão disponibilizada a partir do ano 2000 é a *API Java 3D 1.2*, que foi considerada a primeira implementação “madura” de *Java 3D*, pois, além de carregar as modificações das versões anteriores, ainda adicionava estabilidade. Suas características incluem: o modo de renderização na memória sendo posteriormente apresentada na tela, geometria 3D e uma textura melhorada, aumentando assim funcionalidades de aparência e adicionando capacidades de som.

Ainda que a especificação das *APIs* de *Java 3D* dê grandes informações aos programadores, é necessário que sejam escritas aplicações e *applets* em *Java 3D* para que se satisfaça uma determinada operação. Para requerer aplicações em *Java 3D*, é necessário ter uma plataforma de *Java 2* com ambiente em tempo de execução ou programas de desenvolvimento de ambiente *Java 2* e uma implementação de *Java 3D* (SIEBRA, 2000). A figura 9 demonstra onde *Java 3D* está alocado.

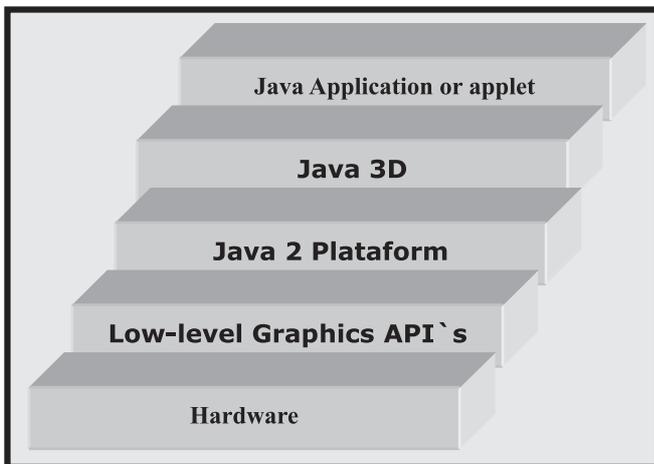


Figura 9 - *Java 3D* como extensão da plataforma *Java* (WALSH, 2001)

*Java 3D* vem de idéias de gráficos existentes de *APIs* e de novas tecnologias. Sendo assim, é

caracterizada pela construção de gráficos sintetizados em baixo nível e encontra as melhores idéias em *APIs* de baixo nível tais como: *Direct 3D*, *OpenGL*, *QuickDraw3D* e *XGL* (figura acima). Paralelamente, o alto nível de construção sintetiza as melhores idéias em diversos sistemas baseados em cenas gráficas. Introduce alguns conceitos não comuns considerados parte de ambientes gráficos, como o som espacial 3D. A capacidade desse som ajuda a prover uma experiência imersiva melhor ao usuário.

Para visualizar os mundos virtuais criados em *Java 3D*, é preciso usar aplicações *Java* tradicionais ou como *applets* onde navegadores conseguem uma visão perfeita da simulação feita, que difere totalmente de *VRML* que são visualizados com *plug-ins* específicos (SIEBRA, 2000).

*VRML* e *Java 3D* são tecnologias que, sob o ponto de vista da autoria de aplicações tridimensionais, podem ser utilizadas em paralelo, e não como excludentes. *VRML* é uma linguagem extremamente interessante para o desenvolvimento de cenários tridimensionais simples e complexos, que requeiram pouca interação. *Java 3D* oferece a possibilidade de construir tais cenários, aproveitando todos os recursos de uma linguagem de programação orientada a objetos. Assim, usando-se mundos virtuais implementados em *VRML* e recursos de linguagem *Java*, é possível obter cenários com características flexíveis, e as demais funções (como interação com o ambiente, acesso à banco de dados) poderão ser desenvolvidas através do *Java 3D* (SIEBRA, 2000). A figura 10 mostra um exemplo de modelagem que utiliza essas duas tecnologias.



Figura 10 - Modelagem 3D usando *Java 3D* com *VRML* (JAVA, 2005)

A especificação de *Java 3D* foi desenvolvida como um esforço colaborativo entre *Sun*, *SGI*, *Intel* e

*Apple*, sendo que a maioria dos objetivos dirigidos para o seu desenvolvimento inclui (WALSH, 2001):

- **Integração com Java:** Permite experientes desenvolvedores de *Java* obter sofisticados programas dentro de ambientes 3D com uma complexidade relativamente mínima, e ainda possibilita acessar características padrões de *Java* como acesso à base de dados, entre outros, e se beneficiar de opções como *multithreading* e coleta automática de lixo.
- **Alta performance:** Permite o suporte à aceleração de hardware válido e um alto nível de otimização e multiprocessamento com renderização.
- **Suporta “loaders” em tempo de execução:** O conceito de “loaders” permite aos programadores de *Java 3D* a habilidade de suportar uma ampla extensão de formatos de arquivos 3D.
- **Suporte para uma generosa série de capacidades críticas:** Foram mantidas intencionalmente as capacidades principais de *Java 3D*, enquanto foram dispensadas e omitidas características que poderiam ser estendidas em camadas acima das APIs de *Java 3D*.

Como um membro da família *Java Media API*, *Java 3D* é o componente 3D de extensão multimídia, com um sistema comprimido que inclui: *Java 2D*, *Java Advanced Imaging (JAI)*, *Java Media Framework (JMF)*, *Java Shared Data Toolkit (JSDT)*, *Java Sound*, *Java Speech* e *Java Telephony* (JAVA, 2005).

*Java 3D* é análogo aos nós *VRML*, pois ambos são usados para a formação de blocos 3D na construção de conteúdos interativos. Porém, enquanto muitos conteúdos compilados em *VRML* podem ser criados usando-se um dispositivo de entrada (como, por exemplo, um simples mouse), o mesmo não ocorre com *Java 3D*, porque uma ferramenta visual de *Java 3D* ainda é inexistente (SIEBRA, 2000).

Usando-se um recurso de editor de código, códigos de arquivos *Java 3D* são executados através de compiladores *Java*, que geralmente correspondem aos *bytecodes* de arquivos de classes.

Diferentemente de *VRML*, as APIs de *Java 3D* permitem aos desenvolvedores comprimir cenas geométricas previamente construídas em formato binário, representando, dessa forma, um pequeno espaço entre muitos formatos de arquivos populares 3D, enquanto que arquivos da linguagem *VRML* são comprimidos com o *Gzip* (WALSH, 2001).

*Java 3D* é muito poderosa, pois seu alto nível de

APIs 3D permite aos programadores focalizar em aspectos mais gerais de desenvolvimento, a entrada de usuário e comportamentos interativos (JAVA, 2005). As figuras 11 e 12 mostram exemplos dessa tecnologia.



Figura 11 - Avatar Java 3D

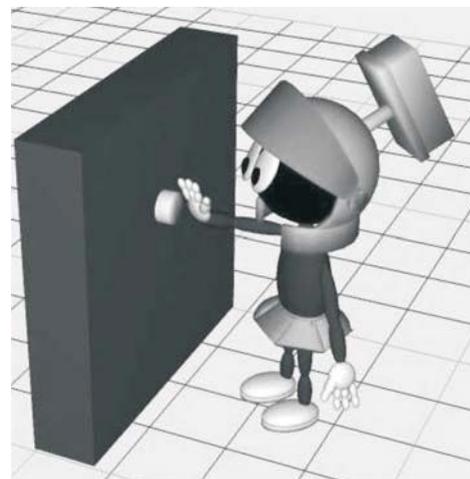


Figura 12 - Exemplo de animação

## MPEG-4

O *MPEG-4*, uma coleção integrada de tecnologias multimídia, desenvolvidas pelo *Moving Picture Experts Group (MPEG)*, é um padrão *ISO/IEC* para a codificação e entrega de diferentes formatos de mídias sobre uma ampla variedade de redes e plataformas de computação. Suporta vários tipos de mídias, como áudio, vídeo, textura, conteúdo 2D/3D. As mídias chegam ao seu destino separadamente (como Fluxos Elementares<sup>1</sup>) e são integradas, produzindo uma rica experiência multimídia no visualizador de conteúdo *MPEG-4* (WALSH, 2001) (N3850, 2003).

O *MPEG-4* utiliza um formato binário para construir a cena, o *BIFS* - que é um completo *framework* para codificação de dados da cena em *MPEG-4*. Esse formato estende a linguagem *VRML* para torná-la mais satisfatória em aplicações *broadcast*.

*BIFS* permite que mídias *MPEG-4* sejam

misturadas com conteúdo 2D e 3D. Ele também trata as interações do usuário com o conteúdo e o gerenciamento de alterações local e remota do dado da cena (Comandos *BIFS*) (WALSH, 2001) (N3850, 2003).

A Internet é somente um dos muitos mecanismos de entrega suportados pelo *MPEG-4*. Sistemas de entrega *broadcast*, tais como aqueles usados na entrega de conteúdo para televisões digitais e *set-top boxes*, também são suportados por esta tecnologia (WALSH, 2001). Sendo assim, o *MPEG-4* pode ser visualizado em uma infinidade de aplicações, por exemplo, jogos de entretenimento 3D para celulares de terceira geração.

*MPEG-4*, através do *BIFS*, estende a linguagem *VRML* em vários aspectos importantes. De acordo com Aaron Walsh (WALSH, 2001), os principais são: compressão binária, onde a cena *BIFS* é armazenada em um formato binário comprimido, que é 10 a 20 vezes menor, em tamanho, que uma cena equivalente em *VRML* comprimida com o *GZip*; mistura de mídias, que integra vários tipos de mídia de forma mais satisfatória que *VRML*; e compressão de áudio, para a mistura de sons naturais, sintéticos e efeitos especiais, característica não suportada pela *VRML*.

### Comparação das Principais Tecnologias 3D para WEB: X3D, JAVA 3D e MPEG-4

Na Seção anterior, foram apresentadas as tecnologias mais satisfatórias para o desenvolvimento de ambientes tridimensionais, tais como, o *X3D*, que é a próxima geração do *VRML*, o *Java 3D* e o padrão *MPEG-4*.

Todas essas tecnologias possuem pontos diferenciados que caracterizam e evidenciam seus principais aspectos. Assim, a tabela apresenta uma análise de algumas características importantes para a disseminação e desenvolvimento de ambientes virtuais elaborados por essas tecnologias 3D.

Comparação: X3D, JAVA 3D e MPEG-4			
	Ferramentas de Autoria	Existência de Plug-ins	Sincronização de Mídia
<i>X3D</i>	Poucas	Poucos	Sim
<i>JAVA 3D</i>	Poucas	Não	Sim
<i>MPEG-4</i>	Uma	Não	Sim

Os itens especificados na tabela são discutidos em detalhe:

- **Ferramentas de autoria:** As ferramentas de autoria facilitam a modelagem de objetos

tridimensionais e, além de facilitarem a criação de mundos geométricos, muitas delas fornecem uma simulação do mundo virtual em tempo de criação, em que o usuário pode incluir, modificar, transladar, rotacionar, entre outras coisas, um determinado objeto com movimentos do *mouse*.

Em relação às tecnologias citadas na tabela, pode-se dizer que o *X3D* não possui muitas ferramentas específicas que tratam seu conteúdo. Atualmente, a *IBM* desenvolveu uma ferramenta de autoria denominada *X3D-Edit Authoring Tool for Extensible 3D (X3D) Graphics (X3DEDIT, 2006)* para oferecer edição e validação de conteúdo *X3D* ou *VRML*.

Em relação a *Java 3D*, poucas ferramentas de autoria são encontradas também. No entanto, vale a pena ressaltar as mais conhecidas, como, o *JICASSO* (JICASSO, 2003), que cria cenas baseadas em *Java 3D* sem qualquer programação e também o *mjbWorld* (MJBWORLD, 2003), o qual pode criar, ler e importar cenas criadas em *VRML 97*.

Como o *MPEG-4* ainda continua em expansão, até o presente momento apenas uma ferramenta preliminar foi disponibilizada, que é o *MPEG4ToolBox1.2* (NIST, 2004) o qual todavia não explora todos os recursos oferecidos pelo padrão.

- **Existência de Plug-ins:** Um *browser* ou *plug-in* é uma aplicação ou um recurso específico incorporado em um *browser* tal como *Netscape*, usado para a visualização de conteúdos *VRML*, ou outros formatos específicos (WEB3D, 2002).

Os *plug-ins* são recursos adicionais de multimídia, que permitem a visualização de um texto, som, vídeo, conteúdo 3D e maior interação com o usuário.

De acordo com a tabela, percebe-se que apenas o *X3D* possui *plug-ins* para a visualização de seu conteúdo. Entretanto, para tais funcionalidades são bem poucos.

Para a visualização do conteúdo criado em *X3D*, vale citar um *plug-in* comumente usado, que é o *OpenWorlds Horizon Browsers* (OPENWORLDS, 2003). Este *plug-in* é capaz de suportar rotas, animação, *script*, *Java*, *JavaScript*, *EAI* e outros.

No caso de *Java 3D*, nenhum tipo de *plug-in* é utilizado, uma vez que ele utiliza-se de *applets* para visualizar o seu conteúdo, onde navegadores conseguem obter uma visão perfeita da simulação realizada.

O *MPEG-4*, por sua vez, também não utiliza nenhum tipo de *plug-in* para visualizar seu conteúdo e

sim de um terminal ou *player*.

• **Sincronização de Mídia:** No contexto de multimídia, o conceito de sincronização é amplo, incluindo o planejamento do acesso a recursos compartilhados, e a especificação e o controle de atividades conjuntas de processos cooperantes. A integração temporal de diversas mídias em uma aplicação requer a utilização de técnicas de sincronização. Mais restritamente, técnicas de sincronização são mecanismos destinados a coordenar a ordenação temporal dos eventos relacionados à aplicação.

Em *X3D*, tal como ocorre em *VRML*, às mídias são mapeadas em objetos geométricos (como um *Box*, *Esfera*) na cena (*X3D*, 2005) e depois sincronizadas de acordo com o tempo desejado.

Em *Java 3D*, as mídias são tratadas através de um pacote de *APIs* denominado *Java Media API* (WALSH, 2001), a qual possui uma coleção de extensões *Java* para permitir que desenvolvedores criem complexos programas multimídias, incluindo 3D.

A sincronização das mídias é feita através da *API JMF - Java Media Framework*. *JMF* permite os desenvolvedores sincronizar e controlar áudio, vídeo e outras mídias baseadas em tempo.

No *MPEG-4*, as mídias são associadas a objetos de mídias onde elas são multiplexadas e sincronizadas pelo terminal no momento da composição da cena audiovisual.

### Considerações Finais

A realidade virtual é a interface mais natural e poderosa de interação homem-máquina, pois, permite ao usuário imersão, interação e navegação em um ambiente tridimensional sintético gerado pelo computador. Entretanto, a técnica de criar ambientes virtuais realistas é ainda um desafio a ser superado, devido à alta demanda computacional necessária para a composição e sintetização de uma cena gráfica.

Um dos aspectos que impactam no desempenho de ambientes virtuais multiusuário é a propagação de mensagens de atualizações entre os objetos ali compartilhados. Diante disso, as técnicas mais promissoras que possibilitam uma melhor integração de mídias, compartilhamento e difusão de mensagens ainda são o *X3D*, *JAVA 3D* e o Padrão *MPEG-4* que todavia está em fase de conclusão.

### Referências Bibliográficas

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

EPUSP, Julho, 1996.

ARAÚJO, R. B., OLIVEIRA, M. A. M. S., *Ambientes 3D Interativos Multiusuário na WEB: Uma Avaliação Realista*, I SICON - Simpósio de Sistemas de Simulação e Controle, Rio de Janeiro-RJ, Setembro, 1998, Anexo I, pp.19-26.

BRANDÃO, E. R. et all. *A Realidade Virtual Como Proposta De Ensino-Aprendizagem, De Matemática Para Deficientes Auditivos - Rv\_Peamda*. IV Congresso RIBIE, Brasília 1998.

FERNANDES, C. J. <http://www.dimap.ufrn.br/~jorge/publicacoes.html>, consultado em 20/11/2005.

JAVA TM DEVELOPMENT KIT DOCUMENTATION, eletronic document in <http://java.sun.com/docs/index.html>, site consultado em 05/11/2005.

JAVA TM DEVELOPMENT KIT DOCUMENTATION, eletronic document in <http://java.sun.com/docs/index.html>, site consultado em 05/03/2005.

JICASSO. 2003 <http://www.cellspark.com/jicasso.html> último acesso em julho de 2002

MJBWORLD. <http://www.martinb.com/> ultimo acesso em julho de 2003.

N3850, Information technology - *Coding of audio-visual objects* - ISO/IEC JTC1/SC29/WG11 N3850, outubro de 2003.

NIST Home Page - <http://mpeg.nist.gov/> ultimo acesso maio de 2004.

OPEN WORLDS. <http://www.openworlds.com/> ultimo acesso maio de 2003.

SIEBRA, A. Claurton - *Uma Arquitetura para Suporte de Atores Sintéticos em Ambientes Virtuais - Uma Aplicação em Jogos de Estratégia*, Dissertação de Mestrado, UFPE, 2000.

TEICHRIEB, V., *Avatares como Guias Interativos para Auxílio na Navegação em Ambientes Virtuais Tridimensionais*, dissertação de mestrado, UFPE, 1999.

VIRTUAL REALITY MODELING LANGUAGE - Especificação version 1997. [http://www.web3d.org/fs\\_specifications.htm](http://www.web3d.org/fs_specifications.htm), ultimo acesso março de 2004.

WALSH, A. & SÉVENIER, M., *Core Web3D*, editora Prentice Hall PTR, ISBN 0-13-085728-9, 2001.

*Extensible 3D (X3D)* - Especificação version 2005, <http://www.web3d.org/x3d/specifications/X3DPublicSpecifications.zip>, consultado em 21/Dezembro/2005.

X3DEDIT. <http://www.web3d.org/x3d/content/REAMDME.X3D-Edit.html>, último acesso em Janeiro de 2006.

### Notas

<sup>1</sup> Fluxos Elementares são dados encadeados recebidos da saída do *buffer* de um codificador, independente de seu conteúdo.

*Recebido em 30 de abril de 2007 e aprovado em 16 de agosto de 2007.*