Combinando VRML e Java para Criação de Ambientes Virtuais Multi-Usuários

Juliano R. Ipolito
Programa de Pós-Graduação em Ciência da Computação – UFSCar
Caixa Postal 676 CEP 13565-905 – São Carlos – SP – Brasil
(juliano@dc.ufscar.br)

Prof. Dr. Claudio Kirner
Faculdade de Informática – Fundação de Ensino "Eurípides Soares da Rocha"
Caixa Postal 2041 CEP 17525-901 – Marília – SP – Brasil
(ckirner@fundanet.br)

Resumo

A combinação das linguagens Java e VRML (Virtual Reality Modeling Language) possibilita a criação de ambientes virtuais tridimensionais multi-usuários através da união do poder de programação e suporte a rede da linguagem Java e da capacidade de geração de cenas 3D da linguagem VRML. Este artigo apresenta uma breve revisão dos conceitos envolvidos no projeto de ambientes virtuais distribuídos multi-usuários e apresenta um sistema criado a partir da combinação destas duas linguagens.

Palavras-chaves : Realidade Virtual, Ambiente Virtual Tridimensional, Sistemas Distribuídos, Multi-Usuário, VRML, Java.

1 Introdução

Uma das metas que sustentam o desenvolvimento da terceira dimensão na Internet é aquela que busca a criação de ambientes virtuais tridimensionais compartilhados, com comunicação e interação entre seus usuários. Ambiente virtual tridimensional é o cenário gráfico 3D que representa uma situação real ou algo abstrato, imaginário, que só existe no computador. O ambiente virtual tridimensional multi-usuários é composto por objetos gráficos que formam a cena 3D e os usuários que interagem simultaneamente no ambiente.

A Web é essencialmente um lugar formado por apresentações de hipertexto, com páginas que contêm textos, sons e imagens bidimensionais. No entanto, ela é um lugar solitário, porque oferece pouco suporte para interação entre seus usuários, que geralmente podem ter acesso às mesmas informações, mas pouco interagem entre si.

Os ambientes virtuais tridimensionais multi-usuários, ou simplesmente DVEs (Distributed Virtual Environments), têm potencial para fazer a Web evoluir de um espaço só de informação para um lugar social, com informação e com uma comunidade de usuários que compartilham essa informação e interagem entre si. Isso pode alterar drasticamente a forma de se aprender, divertir, comprar ou vender, trabalhar etc., usando a rede Atividades que eram realizadas isoladamente poderão ser realizadas em grupo, de forma colaborativa.

1.1 Objetivo e Importância da Pesquisa

Criar ambientes virtuais compartilhados simultaneamente por mais de um usuário é um dos objetivos da Realidade Virtual. Atualmente, a maioria dos projetos de pesquisa estão sendo desenvolvidos nesta direção. Isto se deve ao grande número de aplicações práticas decorrentes do uso desta tecnologia. Arquitetos e projetistas poderiam discutir detalhes de um projeto ou planta, professores poderiam dar aula para seus alunos, pessoas poderiam se encontrar e trocar informações etc., tudo em ambientes virtuais tridimensionais compartilhados. Para que a Realidade Virtual se torne útil na prática, ela deve possibilitar que vários usuários interajam e compartilhem o mesmo ambiente virtual, local ou remotamente. O objetivo desta pesquisa é explorar os conceitos e a potencialidade da Realidade Virtual Distribuída e analisar o uso das tecnologias Java e VRML para construção de ambientes virtuais multi-usuários na Internet. Existem várias tecnologias que permitem a construção de DVEs. Uma forma de se criar DVEs é através da combinação das linguagens Java e VRML. VRML é uma linguagem de descrição de cenas 3D reconhecida pela International Standards Organization (ISO) e inclui animação e som. Seu grande conjunto de primitivas gráficas possibilita a criação de cenas complexas e objetos 3D estáticos ou com comportamentos. Apesar de seu formato de arquivo descrever cenas 3D com eficiência, a linguagem não permite que os ambientes criados sejam compartilhados por vários usuários ao mesmo tempo. A linguagem Java entra no contexto para cobrir esta deficiência, já que possui uma vasta API para comunicação em rede.

2 Realidade Virtual

Realidade Virtual (RV) é um campo emergente da ciência aplicada, sendo em primeiro lugar uma tecnologia [HEIM98]. Durante toda a história da computação gráfica, engenheiros tentaram desenvolver meios realísticos e imersivos para facilitar a interação homem-máquina. Realidade Virtual é o resultado de todo o esforço para que isso ocorra [LATHROP98]. A RV pode ser definida como sendo a forma mais avançada de interface do usuário de computador até agora disponível [HANCOCK95]. Com aplicação na maioria das áreas do conhecimento, senão em todas, e com um grande investimento da indústria na produção de hardware, software e dispositivos de entrada/saída especiais, a Realidade Virtual vem experimentando um desenvolvimento acelerado nos últimos anos e indicando perspectivas bastante promissoras para os diversos segmentos vinculados com a área [KIRNER96]. Agrupando algumas definições [BURDEA94, JACOBSON91, KIRNER96, KRUGER91], pode-se dizer que realidade virtual é uma técnica avançada de interface, em que o usuário pode realizar imersão, interação e envolvimento em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais em tempo real.

3 Sistemas Distribuídos de Realidade Virtual

As aplicações de Realidade Virtual podem ser vistas sob um aspecto bastante amplo, variando de uma única pessoa, usando um único computador, até muitos usuários, usando um sistema distribuído. Os sistema de RV multi-usuários em ambiente distribuído vêm crescendo e apresentam elevado potencial de aplicação. Esse tipo de sistema permite que os usuários geograficamente dispersos atuem em mundos virtuais compartilhados, usando a rede para melhorar o desempenho coletivo, através da troca de informações [KIRNER96].

3.1 Avatares

O usuário, quando entra num sistema distribuído de realidade virtual, na verdade entra num ambiente virtual tridimensional que pode ou não ser uma representação de um cenário real, ou seja, pode ser algo abstrato, que só existe virtualmente, ou pode ser uma representação de algo real, como um shopping center virtual, por exemplo. Uma vez conectado ao sistema, o indivíduo é "visto" sob a forma de **avatar**, que é uma representação tridimensional de sua pessoa e que está totalmente a seu comando. É possível escolher qualquer objeto 3D para ser o avatar de alguém. É possível associar um avatar que seja uma representação real de uma pessoa, um boneco ou personagem famoso ou algo mais abstrato. O importante é que os outros usuários, quando virem aquela figura 3D, saibam quem está ali representado.

3.2 Escalabilidade

À medida em que o número de pessoas que freqüentam determinado mundo virtual começa a crescer, surge a chamada comunidade virtual distribuída, ou seja, o conjunto de pessoas que interagem em um ambiente virtual distribuído. As aplicações que decorrem do uso desta tecnologia são muitas. As pessoas podem simplesmente se divertir, discutir projetos (ambiente colaborativo) ou até mesmo participar de aulas em laboratórios virtuais coordenadas por professores virtuais. Para permitir a escalabilidade do ambiente virtual, muitos sistemas implementam o particionamento. Normalmente, há grandes regiões do ambiente que não são relevantes para um usuário num determinado instante. A idéia é particionar o ambiente e fazer com que as informações sejam transmitidas apenas para os usuários que estiverem a certa distância de quem está transmitindo ou então, somente para aqueles usuários que estiverem em determinada região do ambiente virtual. Eventos e ações em zonas remotas, fora do interesse do usuário não precisam ser distribuídos e objetos remotos não precisam ser mostrados ou podem ser mostrados com nível de detalhe menor. A Figura 1 demonstra esta técnica : o grupo de usuários que está na sala da esquerda envia e recebe mensagens só deles e da pessoa que está na entrada da sala. A pessoa dentro da sala da direita só manda mensagens para aquela outra que está na entrada. As pessoas que estão em salas diferentes não se comunicam e os que estão na porta podem trocar mensagens entre si.

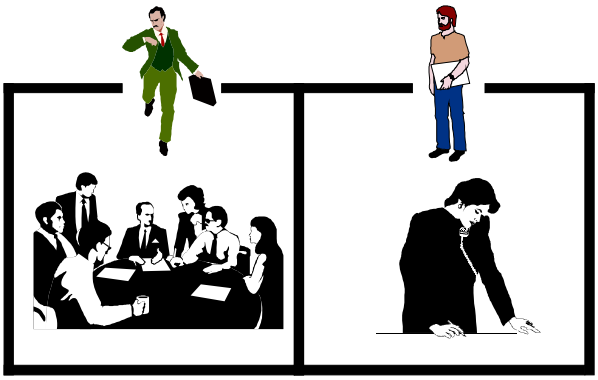

Figura 1 – Particionamento do Mundo Virtual

Pode-se também restringir o alcance de uma mensagem, ou seja, estipular um diâmetro de propagação. Pessoas que estiverem fora deste contexto não a recebem.

3.3 Distribuição de Dados

Num sistema distribuído, as estruturas básicas de comunicação são **unicast**, **broadcast** e **multicast**, conforme mostra a Figura 2. A estrutura **unicast** é o mecanismo de comunicação dos dados conhecido como "um-para-um", realizado pela comunicação de apenas dois hosts. **Broadcast** é o mecanismo de comunicação dos dados conhecido como "um-para-todos", onde um host envia um dado para todos os outros hosts da rede. **Multicast** é o mecanismo de comunicação de dados conhecido como "um-para-vários" ou "vários-para-vários" em uma única operação [TANEMBAUM96].

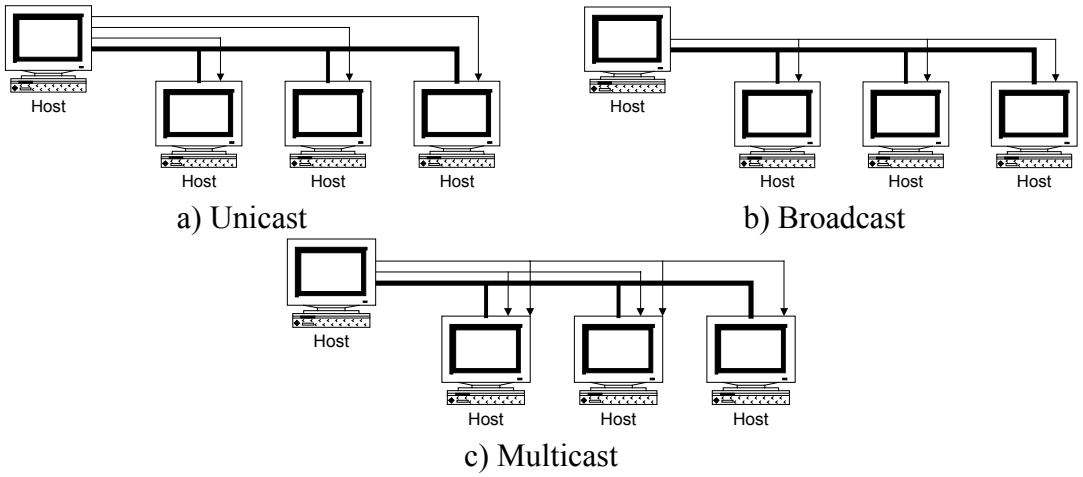

Figura 2 – Distribuição da Dados : unicast, broadcast e multicast

Em um DVE, a informação que circula pela rede pode ser classificada em quatro tipos básicos: arquivos, eventos, mensagens e streaming data. A forma como a mensagem será transmitida e o protocolo usado na transmissão dependerá do tipo da mensagem. A Tabela 1 mostra os tipos de informações e os protocolos a serem usados na transmissão :

Tipo de Informação	Protocolos
Arquivos (descrição da cena e avatares)	HTTP, FTP, TCP
Eventos (atualizações)	UDP (uni/multicast)
Mensagens (dados importantes)	TCP
Streaming Data (áudio, vídeo)	UDP

Tabela 1 – Tipos de informação e protocolos

Dependendo do tipo de informação a ser transmitida no ambiente virtual, um ou outro modelo pode ser usado. Por exemplo: um usuário pode mandar uma mensagem de texto para todos os usuários em um chat público (broadcast). Neste caso, a mensagem sai do host do cliente e vai até o servidor, que a distribui para os demais clientes. Pode ser também que o usuário queira trocar mensagens somente com um outro usuário. Neste caso, pode-se estabelecer uma conexão entre os hosts dos dois usuários e a partir desta conexão eles trocam mensagens entre si. Como este tipo de mensagem, em ambos os casos, tem que ser transmitida com segurança, o protocolo TCP deve ser usado. O protocolo UDP poderia ser usado para transmitir, por exemplo, informações sobre o posicionamento dos avatares. Quando um

usuário se move no DVE seu avatar tem que se mover em todos os outros clientes. Para que isto aconteça, a cada movimento do usuário o processo cliente envia ao servidor uma informação de posicionamento que dever ser distribuída aos demais clientes. Quando o usuário está se movendo muitas informações de posicionamento são geradas, o que significa que não há nenhum problema se o sistema perder uma ou outra informação, porque logo em seguida, uma nova mensagem será gerada com uma nova informação de posicionamento. O UDP é indicado nesse caso. O HTTP pode ser usado para transferir os arquivos VRML.

3.4 Tipos de Sistemas de Realidade Virtual Multi-Usuários

A arquitetura de comunicação em um sistema de Realidade Virtual multi-usuário pode ser centralizada (modelo cliente-servidor) ou distribuída (modelo peer-to-peer), conforme a Figura 3 :

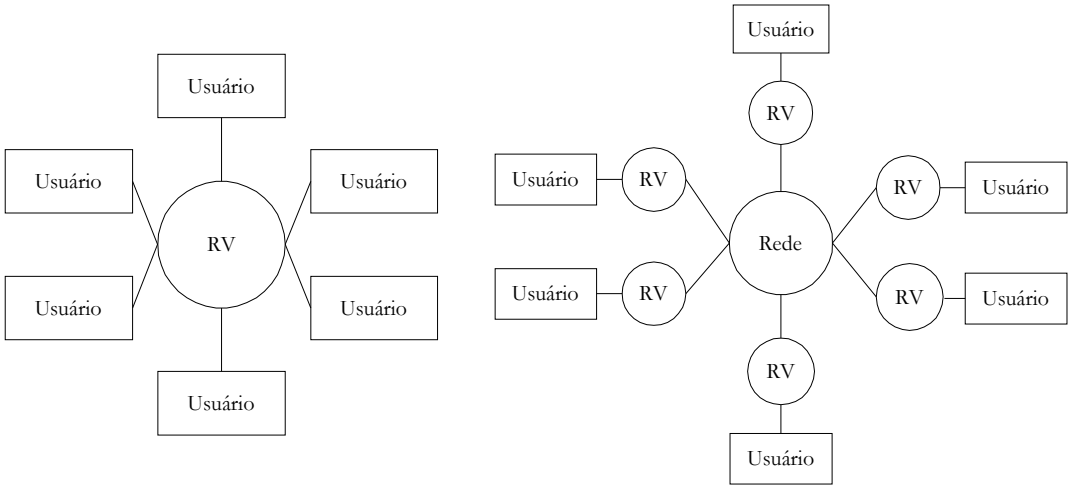

Figura 3 – Modelos de Sistema de RV Multi-Usuário

3.4.1 Modelo Cliente-Servidor

Neste modelo, todos os usuários compartilham o mundo virtual. A maioria dos DVEs de pequena-escala é construída em arquiteturas que seguem o modelo cliente-servidor porque este é o modelo mais simples e de mais fácil implementação. Neste caso, o mundo virtual está centralizado em um único local : o servidor. Qualquer alteração que um usuário cliente faça no ambiente virtual é enviada ao servidor para que este atualize o estado de todos os outros clientes, garantindo a consistência do sistema. Por exemplo, se um usuário *n* se move no ambiente virtual, esta informação é transmitida ao servidor, que diz aos demais (*n-1*) clientes que há um usuário em movimento, e desta forma os (*n-1*) clientes atualizam localmente a posição do avatar que representa o usuário que está se movendo.

Obviamente, o servidor se torna um gargalo, quando o número de usuários aumenta. Evidências empíricas mostram que a escalabilidade de sistemas cliente-servidor sofisticados chegam a ordem de centenas de usuários. Muitos DVEs comerciais usam o modelo cliente-servidor não somente por razões

técnicas, mas também por razões financeiras : clientes são gratuitos, enquanto servidores são pagos [SAAR99].

3.4.2 Modelo Peer-to-Peer

Neste modelo (distribuído), o mundo virtual encontra-se espalhado nos computadores do sistema, podendo ser replicado (para mundos pequenos) ou particionado (para mundos virtuais de grande porte), conforme a Figura 4.

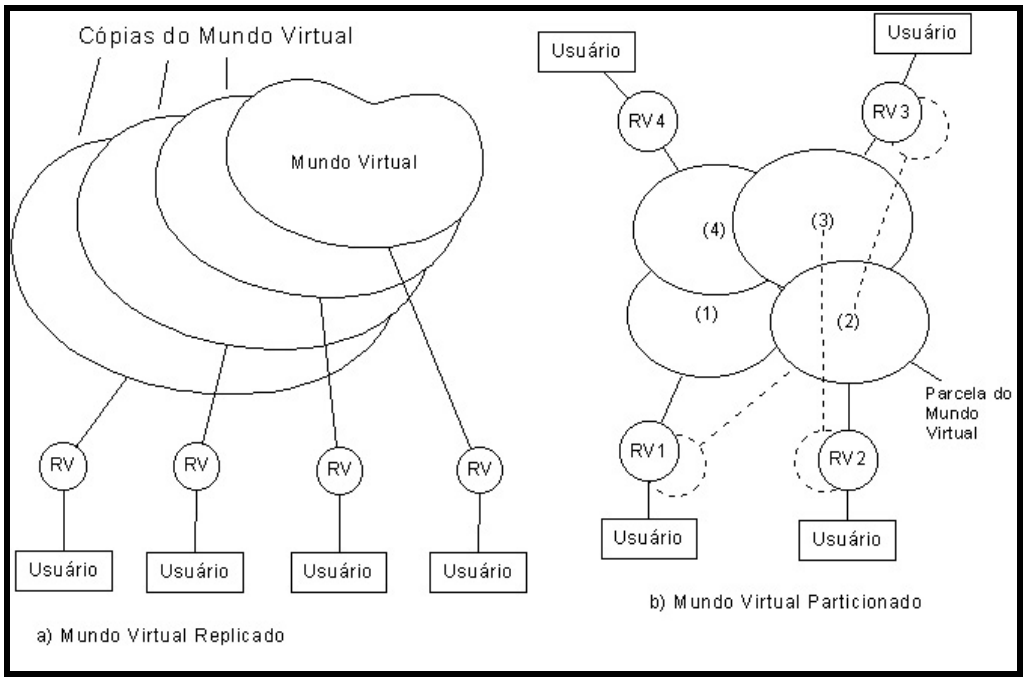

Figura 4 - Acesso ao Mundo Virtual Distribuído

Num sistema replicado com *n* usuários, quando um usuário fizer qualquer alteração no mundo virtual, isto deverá ser comunicado para todas as (*n-1*) versões do mundo virtual, onde estão os outros usuários, constituindo a difusão (broadcast). Num sistema particionado com *n* usuários, a situação é mais complexa, uma vez que o mundo virtual é dividido em várias partes e cada máquina ficará encarregada de uma delas. Como o usuário pode navegar no mundo virtual, ele poderá penetrar em outras regiões, de forma que sua máquina ou servidor deverá receber uma réplica da região, onde ele se encontra. Assim, cada máquina estará cuidando de uma região fora da sua parcela. Se existirem vários usuários em uma mesma região do mundo virtual, esse grupo de usuários receberá uma cópia dessa região. Qualquer alteração no mundo virtual, feita por um membro do grupo, será retransmitida para o restante do grupo, constituindo a retransmissão por grupo (multicast) [KIRNER96].

3.4.3 Modelo Híbrido

Há duas desvantagens em se usar o modelo cliente-servidor: primeiro, o servidor acaba tornando-se um gargalo quando o número de usuários aumenta; segundo, se o servidor falhar, todo o sistema pára. O modelo híbrido é uma combinação do modelo cliente-servidor e do modelo peer-to-peer. No modelo

híbrido, o ambiente virtual é replicado e atualizado em vários servidores (usando o modelo peer-to-peer) e os clientes acessam tais servidores (modelo cliente-servidor), conforme a Figura 5.

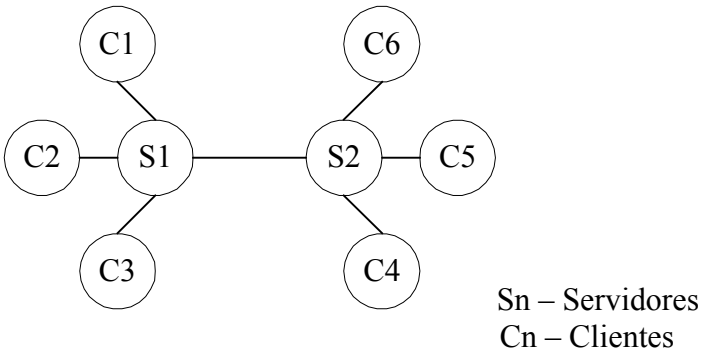

Figura 5 – Modelo Híbrido

4 VRML & Java

A seguir será mostrado porque a combinação Java/VRML é interessante para a criação de ambientes virtuais tridimensionais multi-usuários.

4.1 VRML

VRML é uma linguagem padrão [VRML97] que descreve ambientes virtuais tridimensionais estáticos ou com animação que podem ser vistos localmente ou transferidos pela Internet. Da mesma forma que a linguagem HTML descreve páginas com hipertexto, a linguagem VRML descreve cenas 3D. Normalmente, para se visualizar os ambientes criados com a linguagem, o usuário precisa instalar um plugin (software de apoio) em seu browser, pois a maioria deles não oferece suporte direto à linguagem. Depois de ser configurado, o browser estará apto a mostrar qualquer ambiente criado com VRML. VRML fornece um excelente mecanismo para descrever cenas 3D complexas, mas, infelizmente, não fornece nenhum mecanismo de distribuição/suporte a vários usuários. Esta deficiência na linguagem pode ser coberta com o uso da linguagem Java.

4.2 Java

Java é uma linguagem independente de plataforma, orientada a objetos, muito parecida com C++ em termos de sintaxe, cujo principal objetivo é o desenvolvimento de aplicações que rodem em rede e na Internet. Atualmente, a maioria dos browsers é capaz de executar aplicações Java. Apesar dos programas de rede serem o forte da linguagem, também é possível criar praticamente todo tipo de aplicação de uso geral.

Java é o resultado de todo o processo de refinamento e adaptação que vem ocorrendo com as linguagens de programação no sentido de se conseguir uma linguagem simples, mas com grandes recursos, estando prestes a se tornar um padrão para o desenvolvimento de aplicações orientadas a objetos. Fazendo-se um resumo de suas principais características, tem-se que Java é uma linguagem simples, pequena, segura, orientada a objetos, distribuída, robusta, interpretada (ou compilada

dinamicamente), independente de plataforma, multi-thread, com um mecanismo eficiente de coleta de lixo e manipulação de exceções.

4.3 Java & VRML

Java pode ser usada para acrescentar à linguagem VRML funcionalidades que ela não possui, como por exemplo, acesso à rede e criação de uma interface 2D separada. Além disso, as duas linguagens são independentes de plataforma. Um sistema criado na plataforma Java/VRML terá a sua disposição todo o poder de descrição de cenas 3D da linguagem VRML mais o poder da linguagem Java. Além disso, o sistema rodará em qualquer sistema operacional que tenha um browser com plugin VRML (há implementações de plugins para a maioria das plataformas comerciais atuais). A integração entre as linguagens Java e VRML pode ser feita através do uso da External Authoring Interface (EAI) [VRML-EAI], que conecta o plugin VRML à Java Virtual Machine do browser, conforme mostra a Figura 6.

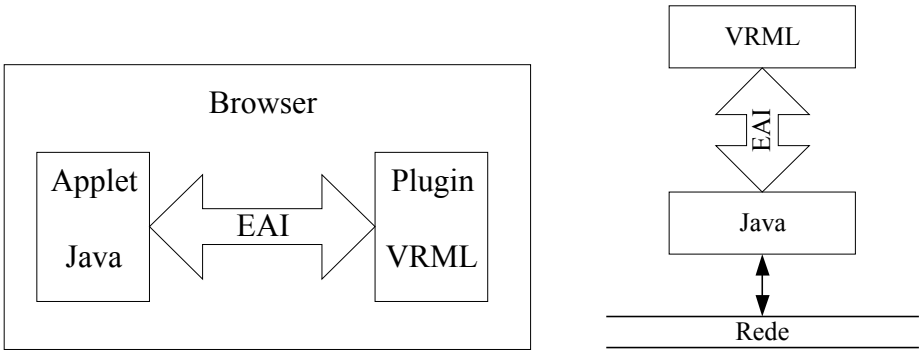

Figura 6 – EAI

A EAI permite uma comunicação bidirecional entre o applet Java e o plugin VRML. Toda a tarefa de mostrar a cena 3D fica por conta do VRML, enquanto que o controle do posicionamento dos avatares e a troca de mensagens entre os usuários é controlada pelo Java, que tem acesso à rede. O Java pode ser usado para tratar eventos e mensagens no mundo virtual. Quando um usuário se move no ambiente virtual, um sensor VRML detecta sua nova posição e, através da EAI, informa o Java que, por sua vez, transmite essa informação pela rede. O mesmo acontece se um usuário altera o estado de algum objeto no ambiente.

5 Ambiente Virtual Interativo Tridimensional - AVIT

O AVIT (Ambiente Virtual Interativo Tridimensional) é um sistema centralizado de realidade virtual multi-usuário (segue o modelo cliente-servidor) implementado, usando a plataforma Java/VRML apresentada neste artigo. O AVIT é composto por dois módulos, conforme mostra a Figura 7.

O módulo **servidor** foi construído 100% em Java e roda em qualquer plataforma de computação que tenha uma máquina virtual Java (JVM – Java Virtual Machine) instalada. O módulo **cliente** é uma combinação Java/VRML e roda em qualquer browser que tenha suporte a Java e VRML. É possível, usando o AVIT, fazer com que qualquer ambiente virtual criado com VRML passe a ter suporte multi-

usuários. Pode-se dizer que o AVIT é, portanto, uma plataforma para criação de ambientes virtuais multi-usuários.

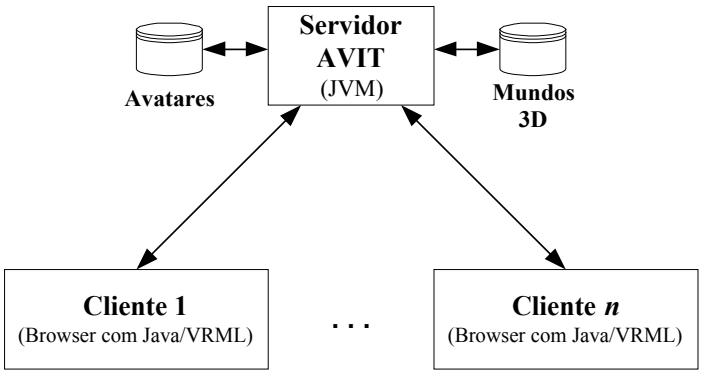

Figura 7 – AVIT

A interface do usuário é mostrada na Figura 8 e é composta por duas partes. No lado superior esquerdo está a parte VRML, que mostra o conteúdo 3D do sistema. O resto é a interface gráfica criada com Java, onde o usuário pode escolher o avatar, entrar com seus dados, e mandar mensagens para outros usuários. A partir do cliente, também é possível ver todos os nomes de usuários que estão conectados ao servidor e obter informações a seu respeito, como nome completo, email etc. Na figura 8 podemos também observar a poderosa combinação entre a interface 3D criada pelo VRML e a interface 2D criada através do Java.

Figura 8 – Interface do Cliente AVIT

5.1 Arquitetura

A arquitetura do AVIT é mostrada na Figura 9, que é um detalhamento da Figura 7. O sistema é multi-thread, ou seja, uma thread nova é criada para cuidar de cada um dos clientes que se conectam ao sistema. Há duas threads principais, que serão explicadas a seguir: a **Dispatcher** e a **Receiver**.

Para entrar no sistema, o usuário precisa escolher, no módulo cliente, em que mundo quer entrar e qual avatar ele quer. Uma vez escolhido o mundo, o arquivo VRML equivalente é transferido via HTTP para o módulo cliente. O próximo passo é escolher o avatar.

Depois que o usuário escolhe o avatar, o módulo cliente requisita uma conexão ao servidor. Se o servidor aceitar a conexão (pode ser que o número máximo de usuários tenha sido atingido), os arquivos VRML correspondentes à descrição de todos os avatares que estão sendo usados pelos outros clientes são transferidos para o novo cliente via TCP/IP. O módulo cliente recebe os arquivos e insere os avatares no mundo virtual via EAI e o servidor informa aos outros clientes que mais um usuário entrou no sistema. Os outros clientes requisitam ao servidor o avatar que representa o novo cliente e inserem esse novo avatar nos seus respectivos ambientes. Se houver mais de um cliente com o mesmo avatar (a geometria 3D pode ser duplicada, desde que tenham nomes de usuários diferentes), apenas uma cópia do avatar é transferida.

A partir do momento em que o servidor aceita uma conexão de um novo usuário, uma nova thread Dispatcher é criada para cuidar exclusivamente deste novo cliente. Depois que o servidor cria a thread Dispatcher, ele entra em estado de espera por uma nova conexão.

As threads Dispatcher são armazenadas em um array e todas têm acesso a todas. A partir daí, toda comunicação será feita por meio das diversas threads Dispatcher, o que deixa o servidor livre para aceitar novas conexões.

No lado do cliente, uma thread Receiver é criada quando a conexão com o servidor é estabelecida. Esta thread está em comunicação direta com sua thread Dispatcher associada, que está em comunicação direta com todas as outras threads Dispatcher e, por conseqüência, com todos os outros clientes.

A troca de mensagens entre os diversos clientes é feita através das threads Dispatcher e Receiver. Na Figura 9, tem-se dois clientes já conectados ao sistema. Se o Cliente-1 se move, ele envia, usando o UDP, uma mensagem de posicionamento ao servidor (no caso, a mensagem chega na thread Dispatcher que cuida deste cliente). A thread Dispatcher que cuida deste cliente envia às demais threads Dispatcher dos outros clientes (no caso, somente para o Cliente-2) uma mensagem contendo a nova posição e orientação do avatar do Cliente-1. Ela faz isso varrendo todo o array de threads Dispatcher e enviando a mensagem para todas as threads que estão no array, exceto ele mesmo. Cada thread Dispatcher que recebe a mensagem de posicionamento repassa a mensagem, via UDP, para a thread Receiver equivalente. Quando a thread Receiver, já no lado do cliente, recebe a mensagem de posicionamento, ela altera a posição do avatar que está se movendo, através da EAI. Em resumo, quando um cliente se move, ele manda uma mensagem de posicionamento à sua thread Dispatcher, que repassa a mensagem à todas as outras threads Dispatcher que, por sua vez, repassam a mensagem aos outros clientes para que eles possam atualizar a posição do avatar do cliente que está se movendo. A

troca de mensagens de texto é dada de forma semelhante. A única diferença é que para este tipo de mensagem é usado o protocolo TCP.

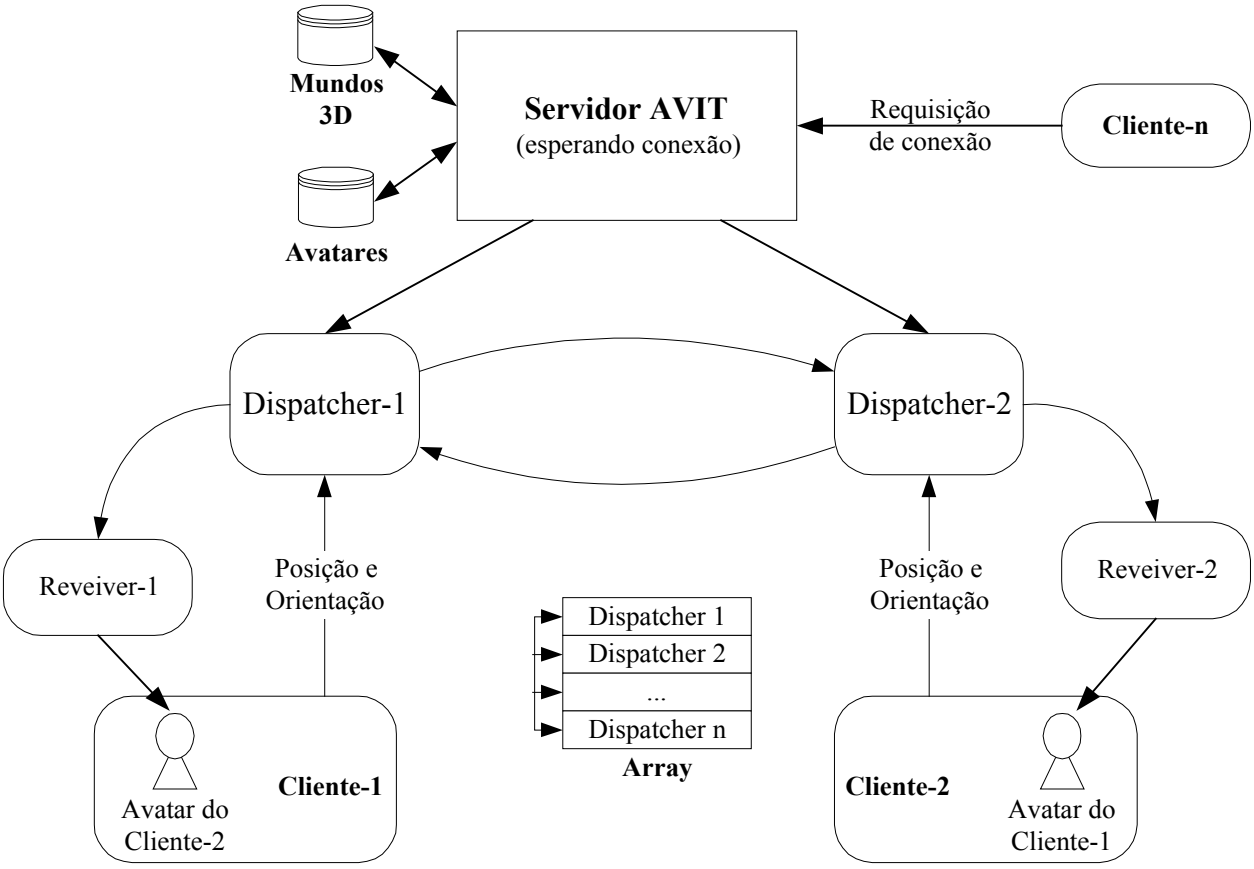

Figura 9 – Arquitetura do Cliente AVIT

6 Resultados e Conclusões

O AVIT é um sistema dinâmico (usuários podem entrar e sair a qualquer momento) de realidade virtual multi-usuário construído em Java e VRML. De um lado, Java possui todos os recursos de programação (principalmente gerenciamento eficiente de threads, concorrência e rede) necessários à programação e implementação de DVEs que seguem a arquitetura apresentada na Figura 9, enquanto de outro, VRML fornece todos os mecanismos necessários para descrição de ambientes 3D.

O sistema foi executado e testado em sistemas com Microsoft Windows NT 4.0, Cosmo Player 2.1 e Microsoft Internet Explorer 4.0 rodando uma máquina virtual Java 1.2, com placa gráfica aceleradora Diamond Fire GL1000Pro e apresentou bom desempenho com até dez usuários. Notou-se que, para um número de usuários maior, a performance do sistema cai, devido à grande quantidade de mensagens que é gerada. Isto aconteceu, principalmente, porque nenhuma técnica de otimização foi implementada. Será implementada, em trabalhos futuros, o particionamento de ambientes, como foi discutido na seção 3.2 e a técnica de dead-reckoning [FISHWICK95, GOSSWEILER94, STYTZ96]. Outras técnicas de otimização de ambientes virtuais criados com VRML também foram utilizadas no sistema [IPOLITO97].

7 Referências Bibliográficas

[BEGAULT94] Begault, D. R. – "***3-D Sound for Virtual Reality and Multimedia"***, Academic Press, Cambridge, MA, 1994.

[BURDEA94] Burdea, G. & Coco, G. – ***"Virtual Reality Technology"***, John Wiley & Sons, New York, NY, 1994.

[CRUZ-NEIRA92] Cruz-Neira, C. et. al. – ***"The CAVE Audio Visual Experience Automatic Virtual Environment"***, Communication of the ACM, vol 35 nº 6, Junho 1992, pp. 64-72.

[FISHWICK95] Fishwick, P.A. – ***"Simulation Model Design and Execution: Building Digital Worlds"***, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[GOSSWEILER94] Gossweiler, R. et al. – ***"An Introductory Tuturial for Developing Multiuser Virtual Environments"***, Presence, vol 3 nº 4, pp 255-264, 1994.

[HANCOCK95] Hancock, D. – ***"Viewpoint : Virtual Reality in Search of Middle Ground"***, IEEE Spectrum, vol. 32, nº 1, Janeiro 1995, pp. 68.

[HEIM98] Heim, M. – ***"Virtual Realism"***, Oxford University Press, 1998.

[IPOLITO97] Ipolito, J. & Kirner, C. – ***"Técnicas de Otimização e Realismo em Aplicações de Realidade Virtual usando VRML",*** Workshop de Realidade Virtual (WRV'97), Universidade Federal de São Carlos (UFSCar), Nov/97.

[JACOBSON91] Jacobson, L. – ***"Virtual Reality: A Status Report"***, Al Expert, Agosto 1991, pp. 26-33.

[KIRNER96] Kirner, C. et al. – ***"Sistemas de Realidade Virtual"***, Apostila do I Ciclo de Palestras de Realidade Virtual, Universidade Federal de São Carlos (UFSCar), disponível a partir de *http://www.dc.ufscar.br/~grv/tutrv.htm*, Outubro de 1996, 54 p.

[KRUGER91] Krueger, M. – ***"Artificial Reality II"***, Addison-Wesley, Reading, MA, 1991.

[LATHROP98] Lathrop, O. – ***"Virtual Reality"***, Notas do Curso "Introduction do Computer Graphics" do SIGGRAPH' 98, Abril 1998.

[SAAR99] Saar, K. – **"VIRTUS : A Collaborative Multi-User Plataform"**, Proceedings of VRML'99, 1999.

[STYTZ96] Stytz, M.R. – ***"Distributed Virtual Environments"***, IEEE Computer Graphics & Applications, vol 16 nº 3, pp. 19-31, Maio 1996.

[TANENBAUM96] Tanenbaum, A. S. - ***"Computer Networks"***, Prentice-Hall, 1996.

[VRML97] ***"VRML Standard Version 2.0"***, ISO/IEC CD 14772, 1996, http://vrml.org/VRML2.0/

[VRML-EAI] ***"VRML External Authoring Interface Specifications"***, 1997, http://vrml.org/WorkingGroups/vrml-eai/