

Renato Cividini Matthiesen

Faculdade Anhanguera de Limeira

renato.matthiesen@unianhanguera.edu.br

AGENTES MÓVEIS PARA CONSTRUÇÃO DE AMBIENTES VIRTUAIS DISTRIBUÍDOS COM REALIDADE AUMENTADA

RESUMO

Realidade Aumentada é uma área da Realidade Virtual que utiliza técnicas de interface computacional para realizar sobreposição de objetos virtuais no mundo real. Agentes móveis é uma tecnologia que representa um modelo de programação distribuída, onde código executável e seu estado são transportados entre *hosts* em uma rede de computadores. Este artigo apresenta um modelo de interação das tecnologias de Agentes Móveis com Realidade Aumentada utilizando metodologia e programação orientados à objetos, linguagem *Java*, biblioteca *Aglets* (programação de agentes móveis) e a biblioteca *ARToolkit* (interação na cena de realidade aumentada). Apresenta uma estrutura chamada *VWBuilder*, composta por três agentes. Um agente para buscar objetos virtuais distribuídos em uma rede, outro para fazer *download* dos objetos e outro para carregá-los em uma cena de realidade aumentada. O objetivo do trabalho é demonstrar a possibilidade de se utilizar agentes móveis para construção de mundos virtuais em sistemas distribuídos de realidade aumentada.

Palavras-Chave: *aglets*; realidade virtual; realidade aumentada; sistemas distribuídos; agentes móveis em *Java*.

ABSTRACT

Augmented Reality is a Virtual Reality area that uses computational interfaces techniques to do a superposition of virtual objects in the real world. Mobile Agents is a technology that represents a distributed programming model, where the executable code and its state are transported among hosts in a computer network. This work presents a study of interaction of Mobile Agents and Augmented Reality using object oriented programming methodology, *Java* as a programming language, *Aglets* (library to implement the mobile agents) and *ARToolkit* (to interact in augmented reality scene). As result, a structure *VWBuilder* was obtained, composed by three agents. One agent searches distributed virtual objects in a network, another to download this objects and a third does the load objects in a scene of augmented reality. The objective of this work is to demonstrate the possibility of use of mobile agents to build virtual worlds in distributed systems of augmented reality.

Keywords: *aglets*; virtual reality; augmented reality; distributed system; *Java* mobile agents.

Anhanguera Educacional

Correspondência/Contato

Alameda Maria Tereza, 2000

Valinhos, São Paulo

CEP 13.278-181

rc.ipade@unianhanguera.edu.br

Coordenação

Instituto de Pesquisas Aplicadas e

Desenvolvimento Educacional - IPADE

Informe Técnico

Recebido em: 22/10/2009

Avaliado em: 20/12/2010

Publicação: 21 de dezembro de 2010

1. INTRODUÇÃO

A tecnologia da informação é parte do processo evolutivo dos meios de comunicação que no princípio da civilização usava desenhos nas paredes de cavernas, em placas de barros ou casca de árvores para transmitir informações. Ainda em um passado remoto, surgiu o papel, primeiro grande avanço na forma de apresentar informações. Com o advento da eletrônica, dos computadores e das redes, a informação passou a ser armazenada eletronicamente e apresentada em equipamentos como a televisão e computadores. O computador eletrônico trouxe um novo processo sofisticado de interação com as aplicações, exigindo conhecimento simbólico, uma vez que o conhecimento do mundo real já não era mais suficiente para representar informações (KIRNER; SISCOOTTO, 2007). As formas de captura, tratamento e transmissão de informações sofreram modificações significativas devido ao desenvolvimento dessas tecnologias. O final do século XX vivenciou a explosão do desenvolvimento das telecomunicações, que possibilitou a proximidade virtual entre pessoas.

Imitar a realidade utilizando gráficos 3D para apresentar dados, tornou-se uma das maneiras mais eficientes de transportar informações (LEA; MATSUDA; MIYASHITA, 1996). O forte apelo das imagens seduz o ser humano mais do que as palavras, e quanto mais próximo da realidade for o apelo visual, mais próximo a realidade o homem se sente (COSTA et al., 2008). Ambientes virtuais 3D baseados na *Web* e multiusuários foram pensados como a interface do futuro (JUNG; MILDE, 1999) e podem ter representações de usuários através de um avatar (agente), que compõem mundos virtuais, comunicam-se e interagem com outros objetos. O uso combinado desses elementos fez com que fosse cunhado o termo Realidade Virtual (RV) que se apresenta como uma maneira contemporânea de se apresentar informações. A RV é tratada como um *continuum* que abrange da Virtualidade Aumentada até a Realidade Aumentada (RA). É composta de elementos reais inseridos em ambientes virtuais (onde a interface é virtual), até o seu oposto, que é o ambiente virtual povoado com elementos reais (onde a interface é real). Quando a percepção não permite a distinção entre a realidade aumentada e a virtualidade aumentada, tem-se a Realidade Misturada (Figura 1). Nela o virtual e o real se misturam em mundos virtuais e possibilitam a construção de ambientes, antes, apenas imagináveis (MILGRAN et al., 1994).

A Realidade Virtual e a Realidade Aumentada apresentam-se com um papel importante, pois são consideradas uma nova geração de interface (KIRNER; SISCOOTTO, 2007). Na medida em que usa representações 3D mais próximas da realidade do usuário, permite romper a barreira da tela e possibilitar interações naturais entre o sistema e o

usuário. A RV e a RA trazem uma importante vantagem sobre outras formas de interação humano-computador, pois o ambiente pode ser visualizado a partir de qualquer ponto de vista à medida que vão sendo feitas alterações em tempo real (SALVADOR; KIRNER, 2007).



Figura 1 – Realidade Misturada (adaptado de MILGRAN et al., 1994).

Nos sistemas de RV e RA, os objetos que compõem o mundo virtual são carregados via leitura do arquivo de grafo de cena (arquivo VRML), armazenado localmente no sistema.

Aplicações simples de Realidade Virtual e Aumentada costumam ser voltadas a um único usuário. No entanto, a demanda por trabalho colaborativo envolvendo vários usuários remotos, vem crescendo e exigindo o desenvolvimento de aplicações sobre redes de computadores ou constituindo sistemas distribuídos. (KIRNER; SISCOOTTO, 2007, p. 16)

Neste contexto, foi observado o problema de que, para a construção de mundos virtuais, o sistema fica limitado ao carregamento dos objetos disponíveis localmente, restritos a poucos objetos virtuais. O compartilhamento ou a possibilidade de carregar objetos virtuais distribuídos em uma rede de computadores, desenvolvidos por outros usuários em diferentes sistemas ou ainda em outros ambientes virtuais é inexistente.

Para buscar objetos virtuais distribuídos, dentre as tecnologias de exploração e busca de objetos para o tratamento da informação, destacam-se os agentes móveis (BAX, 2001). Eles representam um passo na utilização de códigos em execução na Internet, que introduzidos na rede, podem ser transportados em tempo de execução juntamente com suas informações (IBM, 2005). Alinhado com a busca de objetos virtuais distribuídos, testa-se neste trabalho a utilização da tecnologia de agentes móveis (*Aglets*) no domínio de sistemas distribuídos de Realidade Aumentada.

Como solução para o problema da composição de mundos virtuais por objetos virtuais distribuídos e compartilhados, é apresentado um protótipo para mapeamento, *download* e carregamento de objetos virtuais distribuídos em uma cena virtual em uma rede de computadores explorando um novo modelo de programação. O resultado é a

discussão relativa à construção de ambientes de RA com o uso de agentes móveis como uma alternativa aos modelos distribuídos de mundos virtuais.

O projeto e desenvolvimento é orientado a objetos, usando o modelo de programação baseado em agentes móveis, na linguagem *Java* para a implementação dos agentes, e o modelo cliente-servidor com programação em C++ para a implementação da conexão entre o agente móvel e a aplicação de RA.

O trabalho está organizado em 5 seções. Esta seção faz uma introdução deste artigo. A seção 2 aborda os domínios de: Realidade Aumentada e de Agentes Móveis. A seção 3 apresenta a descrição do protótipo, a seguir a seção 4 apresenta os resultados obtidos e para finalizar, a seção 5 faz as considerações finais deste artigo.

2. REALIDADE AUMENTADA COM AGENTES MÓVEIS

2.1. Realidade Virtual

Morton Heilig sintetiza o porquê do uso da RV em:

Se o computador deve alcançar todo o seu potencial como professor, curandeiro, ou inspirador, ele precisa dialogar com o homem da mesma maneira que esse dialogou com a natureza, durante milhões de anos. Isto é, o computador deve falar com o homem, estimulando todos os seus sentidos e permitindo que ele reaja com todas as suas respostas motoras. (HAMIT, 1993, p 18).

A RV objetiva aproveitar o potencial dos computadores, permitindo o crescimento pessoal do próprio homem. Considerando que 70% dos receptores do sentido humano encontram-se nos olhos, uma abordagem especial deve ser empregada em relação ao estímulo à visão humana. Considerando o uso de aplicações comuns baseados na tecnologia WIMP (*Window, Icon, Menu and Pointer*) amplamente utilizada na atualidade em sistemas computacionais, e aspectos dos sistemas baseados em realidade virtual, alguns aspectos podem ser observados: (i) O 3D é um superconjunto do 2D; (ii) A sobreposição de janelas esconde parte da informação; (iii) A distância não é importante em um mundo virtual.

A partir destas observações, nota-se que a utilização de mundos virtuais apresenta vantagens em relação às GUIs (*Graphical User Interfaces*). A imersão em um ambiente virtual considera mais do que os estímulos à visão humana. O aplicativo de RV é uma simulação animada que permite definir e exibir objetos 3D, alterar seu ponto de referência, campo de visão, manipular e interagir com os objetos.

Para que usuários ou programas tenham interação com cenas, fazendo uso de operações não suportadas apenas com o VRML, utilizam-se outras linguagens em conjunto na programação de cenas como: *Java, Java3D, C++, Tcl/Tk, Prolog, X3D*.

2.2. Realidade Aumentada

A Realidade Aumentada (RA) é vista como uma variação da RV. Em RV, o usuário é imerso em um ambiente sintético e não participa do mundo real a sua volta. A RA permite que o usuário veja o mundo real com objetos virtuais superpostos ou combinados com ele. A RA suplementa a realidade, ao invés de substituí-la (RAPOSO et al., 2005). Para o usuário, os objetos reais e os virtuais coexistem no mesmo espaço. A RA é considerada uma tecnologia revolucionária, pois possibilita a simulação de mundos reais e imaginários na tela do computador ou em outros dispositivos (SILVA et al., 2009).

Entende-se que a Realidade Aumentada enriquece o ambiente físico com objetos virtuais, assim, a RA foi beneficiada pelo progresso da tecnologia da computação, tornando viáveis aplicações desta tecnologia, tanto em plataformas sofisticadas quanto em plataformas mais populares. (SILVA et al., 2009, p. 109)

A possibilidade de combinar representações virtuais com o mundo real permite dar ao usuário informações adicionais sobre o mundo real que não podem ser obtidas pelos sentidos humanos. As aplicações possíveis para a RA envolvem, por exemplo, o conserto de componentes de um sistema mecânico, cujas informações de manutenção são virtualmente mostradas sobre as peças a serem reparadas. Na medicina, a RA também pode ser útil, ao indicar, por exemplo, onde o cirurgião deve fazer a incisão em um paciente. Os militares também têm empregado a RA em ambientes virtuais distribuídos (NVEs - *Network Virtual Environments*) no intuito de prover à tropa informações vitais sobre seus arredores (SINGHAL; ZYDA, 1999).

A realidade misturada (formada pela realidade aumentada e a virtualidade aumentada) pode ser classificada em 2 grupos, subdividida em outros 6 grupos principais segundo o tipo de *display* utilizado (MILGRAN et al., 1994). Eles são: (i) Realidade Aumentada com monitor (não imersiva) que sobrepõe objetos virtuais no mundo real; (ii) Realidade Aumentada com capacete (HMD - *Helmet Mounted Display*) com visão ótica direta (*see-through*); (iii) Realidade Aumentada com capacete (HMD) com visão de câmera de vídeo montada no capacete; (iv) Virtualidade Aumentada com monitor, sobrepondo objetos reais obtidos por vídeo ou textura no mundo virtual; (v) Virtualidade Aumentada imersiva ou parcialmente imersiva, baseada em capacete (HMD) ou telas grandes, sobrepondo objetos reais obtidos por vídeo ou textura no mundo virtual; e (vi) Virtualidade Aumentada parcialmente imersiva com interação de objetos reais, como a

mão, no mundo virtual. Uma definição mais precisa de realidade misturada envolve: a combinação do real com o virtual; a interação em tempo real; o posicionamento tridimensional do real e virtual (AZUMA et al., 2001).

Os dispositivos de RA com tecnologia de vídeo funcionam através da combinação de um HMD ou aparelho similar e câmeras de vídeo montadas na cabeça do usuário. As câmeras de vídeo fornecem ao usuário a visão do mundo real. A imagem da câmera é combinada com a imagem virtual e o resultado é enviado para um monitor.

Para o alinhamento dos objetos virtuais com os objetos reais nos sistemas ópticos uma única informação de posição é passada por um dispositivo rastreamento. No sistema de vídeo, técnicas de Visão Computacional permitem recuperar a posição e orientação da câmera (KIRNER; SISCOOTTO, 2007). A seguir, é feita uma descrição de um sistema de rastreamento.

2.3. Rastreamento Baseado em Marcas

O rastreamento é útil para sistemas de Realidade Virtual e Aumentada, pois objetos rastreados servem como dispositivos de interação em cenas 3D (CONSULARO; COELHO; CALONEGO JR., 2007).

O *ARToolkit* é uma biblioteca de rastreamento de marcas (*Patterns*). Uma de suas aplicações, o *simpleVRML.cpp* realiza a apresentação do mundo virtual combinando imagens capturadas por uma câmera (*WebCan*) misturado com um objeto virtual (arquivo “.wrl”). Para isto usa técnicas de Visão Computacional para calcular o ponto de vista real da câmera em relação a um marcador no mundo real. O *ARToolkit* usa, o tamanho conhecido do quadrado de uma marca (Figura 2) e a orientação do padrão encontrado para calcular a posição real da câmera em relação a posição real do marcador. Uma matriz 3x4 conterá as coordenadas reais da câmera em relação ao marcador. Se as coordenadas virtuais e reais da câmera forem iguais, o modelo de computação gráfica pode ser desenhado precisamente sobre o marcador real.



Figura 2 – Marca (*Pattern*).

Ao executar o *ARToolkit* aparece uma janela de visualização da cena com a imagem real capturada pela câmera e a imagem do objeto virtual carregado sobre a marca posicionada em frente a câmera, conforme mostra a Figura 3.



Figura 3 - Imagem do mundo virtual renderizada sobre imagem do mundo real.

Para mostrar detalhes da seqüência de funcionamento do *ARToolkit*, é apresentado abaixo, na Figura 4 um diagrama com a seqüência de processos do software.



Figura 4 – Diagrama dos passos da detecção dos marcadores e posicionamento de objetos virtuais sobre os marcadores na cena do mundo virtual (CALONEJO JR., 2004).

2.4. Agentes Móveis em Sistemas Distribuídos

Segundo Tanenbaum e Van Steen (2007), um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente. Uma variação dos sistemas distribuídos, os fracamente acoplados, representa atualmente a plataforma base para os sistemas de redes de computadores baseadas em simples aplicações cliente-servidor, sistemas baseados em agentes, até os novos sistemas baseados em computação em nuvem¹, que formam a grande estrutura dos atuais sistemas computacionais.

¹ Computação em nuvem é um espaço de processamento e armazenamento de dados que não depende de nenhuma máquina específica para existir. Revista Veja, ed. 2125, ano 42, n. 32, p. 64, 12 ago. 2009.

Em virtude da expansão e do desenvolvimento tecnológico das redes de comunicação e concepção e implementação de sistemas que têm o aspecto distribuído e/ou compartilhado como principal agente motivador tem crescido. Atualmente é possível utilizar-se da rede para diversas atividades, incluindo negócios, entretenimento e disseminação do conhecimento, entre outros. (RODELLO, 2007, p. 130).

Agentes móveis constituem um modelo de programação distribuída que oferece uma maneira alternativa para o projeto de programação de aplicações distribuídas. Este modelo se fundamenta na idéia de que, em vez de transmitir somente dados entre computadores de uma rede, como em sistemas distribuídos tradicionais, o código executável também pode ser transmitido (NAGAMUTA, 1999; LANGE; OSHIMA, 1998; NUNES; LABIDI, 2002). Eles são entidades de *software* autônomas, capazes de viajar através de uma rede e executar tarefas em nome do usuário (EID et al., 2005).

Os agentes, quanto a sua mobilidade, podem ser estáticos ou móveis. Agentes estáticos são aqueles que executam operações somente em um mesmo endereço (*host*). Agentes móveis são aqueles que não possuem limitação de local de execução, e podem migrar entre endereços (*hosts*) dentro de uma rede (LANGE; OSHIMA, 1998). O termo código se refere ao código da classe (no contexto de orientação a objetos) necessário para que o agente possa ser executado. O estado corresponde aos valores dos atributos do agente, que caracteriza o seu estado de execução.

Apresenta-se a seguir sete razões para utilização de agentes móveis (LANGE; OSHIMA, 1999): (i) Redução do tráfego na rede; (ii) Redução da latência da rede; (iii) Encapsulamento de protocolos; (iv) Execução assíncrona e autônoma; (v) Adaptação dinâmica; (vi) Heterogeneidade; (vii) Robustez e tolerância à falhas.

Existem várias plataformas para desenvolvimento de agentes móveis que podem ser utilizadas atualmente. Entre os artigos analisados (MATTHIESEN, 2006), apresentou-se a utilização da plataforma *Aglets* (que será utilizada neste trabalho) em diversos domínios. Eles são: (i) Monitoramento e Gerenciamento de Redes (Gerenciamento de Recursos e Detecção de Falhas); (ii) Busca e Filtragem de Informações (Gerenciamento de Sistemas de Banco de Dados Distribuídos, Recuperação e Reunião de Informações); (iii) Internet para integração com servidores *Web*; (iv) Detecção de intrusões utilizando métodos de detecção.

Verificou-se, através da análise da pesquisa citada que 13% dos artigos fazem uso de agentes em busca e filtragem de informações e observa-se que, dentro desta área, 57% dos artigos correspondem ao domínio específico deste trabalho, pois estão relacionados à busca e recuperação de informações em uma rede. Na categoria de busca, foram encontrados dois trabalhos que fizeram uso da plataforma *Aglets*.

Além da plataforma, para o desenvolvimento de agentes móveis, é necessária também a utilização de linguagens de programação. A linguagem VRML (*Virtual Reality Modeling Language*) e X3D são tecnologias que podem ser utilizadas para construção de ambientes virtuais distribuídos (NVEs). A VRML estabeleceu o padrão para descrição de cenas 3D em WWW, mas não suporta mundos virtuais multiusuários. A utilização de outras linguagens de programação como *Java* e de tecnologias como a de agentes passa a ser uma extensão das ferramentas para construção de NVEs.

2.5. Agentes Baseados em Applets

A tecnologia *Aglets* representa o próximo passo na evolução de programas executáveis na Internet, introduzem programas que podem ser transportados em uma rede de computadores com seu estado de informação (LANGE; OSHIMA, 1998; VELLOSO et al., 2002).

O *Aglets Software Development Kit (ASDK)* introduz o conceito de *Aglets* (Agente + *Applet*) que é um programa em execução que pode mover-se de uma máquina (*host*) para outra em uma rede de computadores. *Aglets* estende o modelo de código móvel de *applets* o seu estado de execução.

Um *aglet* utiliza uma interface para obter informações do ambiente e enviar mensagens a este ambiente e a outros *aglets* ativos. Esta interface oferece gerenciamento dos *aglets* no ambiente com segurança contra *aglets*, ou códigos maliciosos. O contexto *Aglets* é criado por um servidor que tem o objetivo de monitorá-los na rede. Este contexto cuida dos *aglets* que estão sendo enviados e recebidos no *host* e oferecem um recurso gráfico para visualização. No *ASDK* este servidor é chamado de *Tahiti*. Os *aglets* comunicam-se através de troca de mensagens e a biblioteca *Aglets* de classes *Java* é chamada de *JAAPI (Java Aglets Application Program Interface)*.

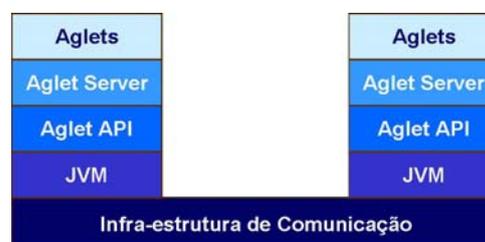


Figura 5 – Modelo conceitual de agentes móveis com *Java*.

O modelo da Figura 5 ilustra uma plataforma *Aglets* que fornece uma *API* com a funcionalidade básica para os processos de gerenciamento, migração e comunicação de agentes. O servidor de *Aglets* é *multithread* (executa múltiplas linhas de processamento) e

funciona no topo da *JVM (Java Virtual Machine)*. Os elementos dos três níveis superiores da Figura 5 são: *Aglets*, *Tahiti (Aglet Server – Ambiente de Execução, Figura 6)* e *Aglet API*. Estes componentes determinam as características dos *Aglets*, com seu ambiente de execução.

Uma das principais funções do *Aglet Server* é controlar o acesso aos recursos dos *hosts* onde os *aglets* são executados. Em cada *host*, múltiplos usuários podem ter acesso à plataforma, e, cada usuário, pode corresponder a várias instâncias do servidor para a execução dos *aglets*. A utilização dos servidores é feita pela configuração de portas lógicas.



Figura 6 – *Tahiti*: Ambiente de execução de um *aglet*

O *Aglets Software Development Kit (ASDK)* é uma plataforma para desenvolvimento de agentes móveis (*Aglets*). Os principais conceitos da plataforma *ASDK* são: (i) *Aglet*: objeto *Java*, autônomo, possui a sua própria *thread* de execução e é capaz de migrar de uma máquina para outra na rede; (ii) Contexto: lugar de execução de um *aglet* (trata-se de um objeto estacionário que gerencia os *aglets* em execução, sendo que um nó na rede pode conter mais de que um contexto); (iii) *Proxy*: é utilizado para a interação com outros agentes (representa um *aglet* e protege seus métodos públicos do acesso direto por outros *aglets*, oferece transparência de localização escondendo a localização verdadeira do *aglet*); (iv) Mensagem: objeto de comunicação síncrono ou assíncrono dos *aglets*; (v) Itinerário: plano de migração de um *aglet* e (vi) Identificador: identificação do *aglet*, única e imutável durante sua vida.

A *API Aglet*, *AAPI* ou *JAAPI* é um pacote *Java* que define as classes e interfaces que implementa as funcionalidades para a construção e o funcionamento dos agentes móveis. As principais classes e interfaces são: *Aglet*, *AgletProxy*, *AgletContext*, *Message*, *FutureReplay* e *AgletID*.

3. PROTÓTIPO DO *VIRTUAL WORLD BUILDER*

Virtual World Builder (*VWBuilder*) é a denominação do protótipo desenvolvido para o uso de agentes móveis aplicados na composição de mundos virtuais com realidade aumentada.

Apresenta-se nesta seção uma descrição do protótipo *VWBuilder* e a descrição do *NetARToolkit*, por se tratar do mecanismo utilizado para a produção da realidade aumentada, conforme ilustra a Figura 7.

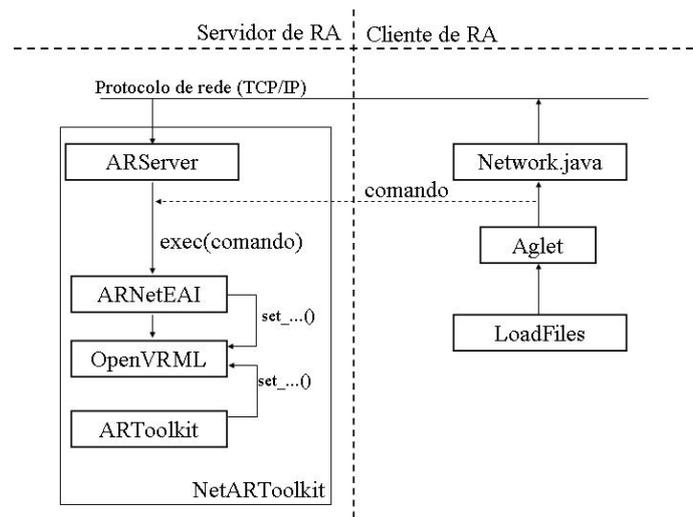


Figura 7 – Estrutura modular do *VWBuilder* e interação com a biblioteca *ARToolkit*.

A aplicação *NetARToolkit* implementa um serviço de RA, em que *ARServer* é um processo baseado em *thread* que obtém comandos a partir da rede e os transmite ao objeto *ARNetEAI*. Os agentes móveis baseados em *Aglets* usam a classe *Network* para atuar na cena, enviando mensagens ao servidor. O ambiente *NetARToolkit* permite que os agentes móveis interajam com os usuários, utilizando o espaço virtual.

As principais funcionalidades do *VWBuilder* são: a busca de identificadores de arquivos; a busca de arquivos (*download*); a carga de cenas em uma aplicação de RA. Essa estrutura é constituída de agentes móveis independentes entre si e ocorrem em três etapas distintas, implementadas através de três *aglets*: (i) O primeiro agente móvel *SearchFileNames* consulta um arquivo local contendo os URLs de *hosts* a serem visitados, criando o seu itinerário. Percorre o itinerário montando uma lista com todos os identificadores de arquivos com as respectivas URLs, armazenando localmente o resultado da busca; (ii) O segundo agente móvel *DownloadFile* que usa o resultado da pesquisa para buscar os arquivos e armazená-los localmente; (iii) O terceiro agente móvel *LoadFiles* realiza a inserção dos objetos virtuais no ambiente de RA.

3.1. O Projeto VWBuilder

O *VWBuilder* é constituído pelas classes: *ContextConfig*, *SearchFileNames*, *DownloadFile* e *LoadFiles*. As classes *SearchFileNames*, *DownloadFile* e *LoadFiles* herdam a classe *Aglet* do pacote `com.ibm.aglets`. O *VWBuilder* utiliza também a classe *Network* como meio de acesso ao servidor de mundos virtuais *ARToolkit*. A Figura 8 ilustra o diagrama de classes do *VWBuilder*.

A classe *ContextConfig* carrega as URLs que o agente deve utilizar para criar seu itinerário. As URLs estão definidas no arquivo `c:/aglets/listURLs.txt` da máquina local. Esta é uma classe abstrata utilizada pela classe *SearchFileNames* para montar o itinerário que o agente usará para a referência de URLs e arquivos “.wrl” (exemplo: `atp://micro-r02:4435`).

O método `getURL()` é utilizado para ler as linhas (endereços de URLs) do arquivo `c:/aglets/listURLs.txt` da máquina local. Este método irá ler cada linha do arquivo e armazená-las na variável “url” formando uma lista de endereços (URLs) armazenada em `list`. Um objeto “l”, instanciado como *ContextConfig* é utilizado para imprimir os valores lidos no console.

A classe *SearchFileNames* realiza duas funções. Primeiro ela verifica o itinerário do agente (lista de URLs que o agente deverá visitar) definida no arquivo `c:/aglets/listURLs.txt` da máquina local e estabelecido pela classe *ContextConfig*. Depois o agente busca nos URLs o nome dos arquivos (.wrl) encontrados nos diretórios pesquisados (`c:/aglets/vrml` das máquinas remotas) e monta uma nova lista com o endereço (url) com o arquivo tipo “.wrl” (o formato é `atp://micro-r02:4435/box.wrl`). Estas informações ficam gravadas no arquivo `c:/aglets/listDir.txt` na máquina local. O método `onCreation()` é responsável por criar o agente de busca. O conteúdo do arquivo “`c:/aglets/listURLs.txt`”, informados pelo objeto *ContextConfig* é utilizado para criar o itinerário que o agente vai percorrer.

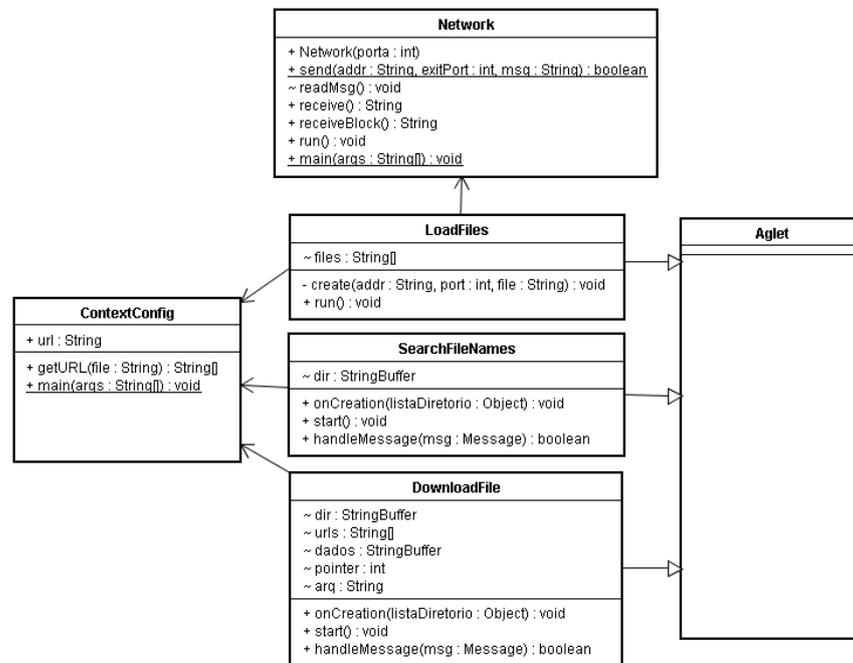


Figura 8 – Diagrama de classes UML do protótipo do *VWBuilder*.

O método *itinerary.startTrip()* faz com que o agente dê início ao processo de busca pelas informações sobre locais e nomes dos objetos virtuais distribuídos na rede, visitando os URLs informados em *ContextConfig*. Em seguida, o agente percorre as URLs informadas enquanto o agente de busca recebe a mensagem *getDir*. Desta forma, o agente estará copiando as informações de “URL + nome do arquivo .wrl” do diretório *c:/aglets/vrml* da máquina remota pesquisada. O agente continua a busca por informações até que o URL pesquisado seja comparado com o seu próprio endereço (URL). Isto é realizado pelo método *getAgletContext().getHostingURL()* herdado da classe *Aglet*. Quando é passado a mensagem *saveList* ao agente, ele encerra o trabalho de busca e finaliza a inserção das informações no arquivo *c:/aglets/listDir.txt*.

A classe *DownloadFile* estende a classe *Aglet*. Ela realiza o *download* dos arquivos “.wrl” dos diretórios das máquinas pesquisadas na rede, mapeadas através da classe *SearchFileNames* com as informações de URL e arquivo “.wrl” gravadas no arquivo local *c:/aglets/listDir.txt* gravando os arquivos baixados no diretório local *c:/aglets/tempWRL*. Sua funcionalidade é análoga à classe *SearchFileNames*. A diferença fundamental é que ela faz a cópia dos arquivos “.wrl” mapeados no arquivo *c:/aglets/listDir.txt*, para o diretório local *c:/aglets/tempWRL*. O método *onCreation* cria um novo agente. Este agente verifica o diretório *c:/aglets/listDir.txt*, monta o itinerário do agente e parte para os hosts em busca dos arquivos. Enquanto a mensagem passada a ele for *getFile*, o agente busca no próximo URL novos arquivos. Esta tarefa é repetida até que uma mensagem com conteúdo *saveFile* seja passada para o agente de *download*. Quando isto ocorrer, ele encerra a tarefa de

download. Esta mensagem é transmitida pelo próprio agente, fazendo-se a comparação do próximo URL a ser pesquisado, que retorna o URL da máquina local.

O agente lê os arquivos dos diretórios remotos *c:/aglets/vrml* e os copia para o diretório local *c:/aglets/tempWRL*. No final do trabalho, o diretório *c:/aglets/tempWRL* local contém uma cópia dos arquivos distribuídos na rede.

A classe *LoadFiles* também estende a classe *Aglet*. Ela realiza o carregamento dos arquivos do diretório *c:/aglets/tempWRL* da máquina local, que foram descritos no arquivo *c:/aglets/listDir.txt* e baixados para o diretório local *c:/tempWRL* para a cena do mundo virtual do aplicativo *NetARToolkit*. Por se tratar de um *aglet*, é necessário realizar a importação de alguns pacotes *Aglets*. É definida a criação de um novo *aglet*. Em seguida é realizada a verificação das informações da URL e da porta em que o arquivo VRML será carregado.

A classe *Network* realiza a comunicação com o mundo virtual no renderizador do *ARToolkit*. Esta classe implementa a interface de comunicação via *socket* (porta 1201) com um servidor que lê os arquivos “.wrl” e os insere no mundo virtual. O método *send* permite que se faça o envio do arquivo “.wrl” ao servidor *ARServer* da biblioteca *ARToolkit*.

Este objeto abre uma conexão via *socket* (porta 1201) e envia a *stream* do conteúdo do arquivo lido. Ao finalizar a leitura de todos os arquivos da *string* “saída”, fecha-se o *socket* com e finaliza-se o carregamento do objeto virtual na cena.

3.2. Diagrama de Caso de Uso do Projeto

A Figura 9 apresenta o diagrama de caso de uso do projeto *VWBuilder* apresentando as tarefas e as relações entre o usuário (ator) e as tarefas realizadas pelas classes. A tarefa “Buscar nomes de arquivos” é executada pela classe *SearchFileNames*; a tarefa “Baixar arquivos relacionados” é executada pela classe *DownloadFile* e a tarefa “Carregar arquivos na cena” é realizada pela classe *LoadFiles*. Esta última depende do renderizador do *ARToolkit* estar ativado e com um a cena carregada.



Figura 9 – Diagrama de Casos de Uso UML do projeto *VWBuilder*.

4. RESULTADOS

O *VWBuilder* irá pesquisar e buscar objetos virtuais (arquivos com extensão *.wrl*) em *hosts* da rede. O primeiro passo para se fazer o programa *VWBuilder* ser executado em uma rede de computadores é realizar a instalação da plataforma *Aglets* nas máquinas que farão parte do sistema. Após esta etapa e configuradas as variáveis de ambiente, deve-se executar o servidor *Tahiti* (Figura 10).

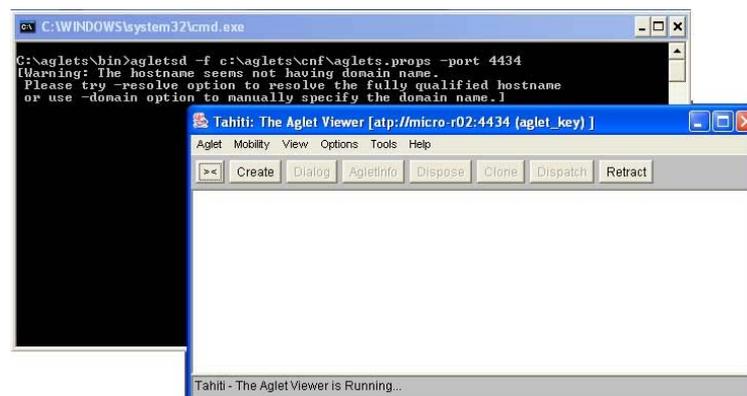


Figura 10 – Ambiente de execução *Tahiti* carregado.

Com os repositórios criados e com uma das máquinas contendo o *VWBuilder*, a próxima etapa é configurar o arquivo que terá os endereços para o *Tahiti* disparar os agentes na rede. As Figuras 11a e 11b mostram exemplos de arquivos *listURLs.txt*.

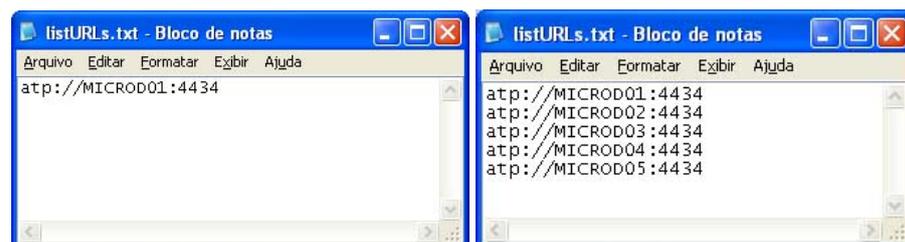


Figura 11 – Arquivos *c:/aglets/listURLs.txt* do *host* *micro-r02*:
a) Usado no primeiro teste e b) usado no segundo teste.

O próximo passo é executar o *aglet SearchFileNames* através do *Tahiti*. Para executar o *SearchFileNames*, deve ser escrito seu caminho completo em *Aglet name* no *Tahiti* a partir do diretório *c:/aglet/public*. As barras devem ser substituídas por pontos. A Figura 12 apresenta esta tarefa.

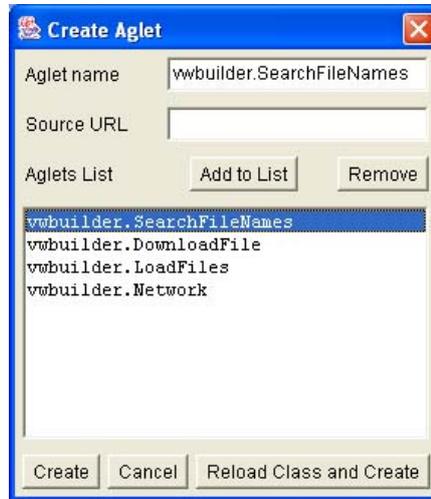


Figura 12 – Execução (criação) do *aglet SearchFileNames*.

Após a execução do *aglet SearchFileNames*, o arquivo *c:/aglets/listDir.txt* deverá ter sido completado com informações a respeito das URLs juntamente com os arquivos “.wrl” encontrados na rede. As Figuras 13a e 13b apresentam exemplos de arquivos *listDir.txt* com as informações geradas nos testes.



Figura 13 – Visualização do arquivo *listDir.txt* após uso do *SearchFileNames*.
a) Teste com arquivos em uma máquina e b) Teste com arquivos em diferentes máquinas.

Após criar o arquivo *listDir.txt* com as informações a respeito dos arquivos “.wrl” distribuídos nas máquinas na rede, a próxima etapa é criar um agente para fazer o download destes arquivos. O *aglet DownloadFile* é o responsável por esta tarefa. A Figura 14 apresenta o *Tahiti* com os agentes criados e já retornados ao contexto da máquina local e o diretório com os arquivos copiados.

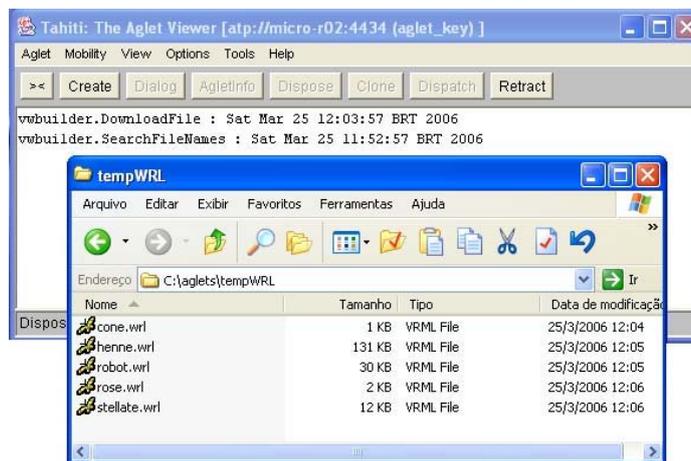


Figura 14 – Agentes no *Tahiti* e diretório *c:/aglets/tempWRL* com os arquivos baixados pelo agente *DownloadFile* no segundo teste realizado.

Antes de criar o agente *LoadFiles* que carrega os objetos virtuais na cena é necessário executar o aplicativo do *ARToolkit*. A Figura 15 apresenta os arquivos localizados na rede carregados no ambiente virtual.

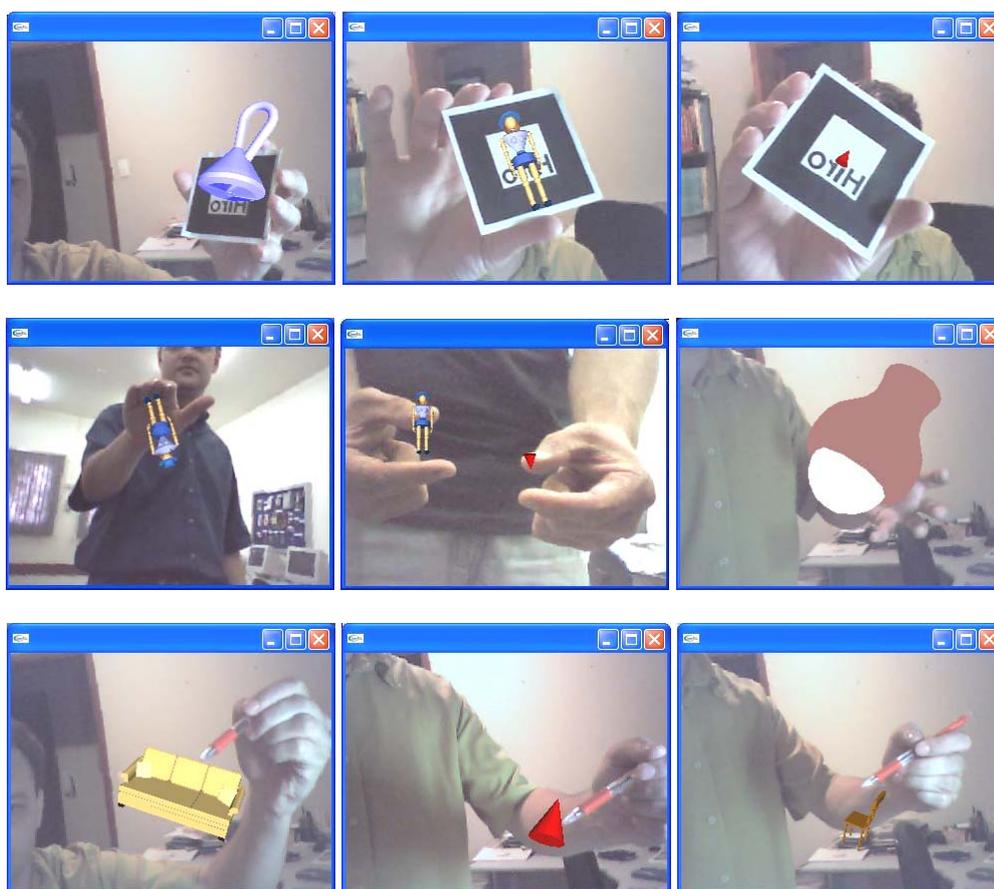


Figura 15 – Cenas com objetos virtuais carregados pelo *LoadFiles* na cena de realidade aumentada.

5. CONSIDERAÇÕES FINAIS

Este artigo trata do estudo e apresentação de uso de uma estrutura de busca de objetos virtuais (arquivos com extensão “.wrl”) distribuídos em uma rede de computadores e seu carregamento em uma cena de realidade aumentada utilizando-se a tecnologia de agentes móveis (utilizando a biblioteca *Aglet*) para a busca, *download* e carga na cena de realidade aumentada.

Agentes móveis vêm tendo crescimento em interesse da comunidade de pesquisas. Não se espera que as aplicações de agentes móveis substituam as tecnologias utilizadas nas aplicações tradicionais no futuro, mas os agentes têm um grande potencial de dinamizar a realocação de código em execução.

Realizou-se testes com os agentes, onde se percebeu que, ao se criar um agente móvel, ele migra para o *host* de destino, permanece neste *host* um determinado tempo, realizando sua tarefa, e retorna com para o *host* original com a informação já processada. Desta maneira, é possível observar uma das características principais dos agentes móveis que é distribuir o processamento de informações, utilizando-se da rede para enviar e receber informações processadas, através de códigos em execução.

Como resultado, foi apresentado evidências da possibilidade de desenvolvimento de aplicações distribuídas de realidade aumentada com agentes móveis.

Uma vantagem, comparada a outros modelos de programação, é que os agentes móveis se transportem em uma rede que possua o servidor de agentes (*Tahiti*) ativado, e possibilitam que códigos em execução migrem entre *hosts* executando suas tarefas. Finalizando a operação com o retorno para o *host* original que o criou. Neste modelo, todo *host* do sistema pode enviar e/ou receber agentes.

Como contribuição, foi desenvolvido um estudo que fez uso das tecnologias baseadas em agentes móveis, juntamente com a área de realidade aumentada. O resultado deste estudo deu origem um protótipo formado por agentes que podem ser utilizados em trabalhos futuros como parte de projetos que busquem explorar a mobilidade de códigos em execução em sistemas distribuídos de realidade virtual e realidade aumentada.

REFERÊNCIAS

AZUMA, R. et al. Recent Advances in Augmented Reality. **IEEE Computer Graphics and Applications**, v. 21, n. 06, p.34-37, nov./dez. 2001.

BAX, M.P.; SOUZA, R.R. Uma proposta de uso de agentes e mapas conceituais para representação de conhecimentos altamente contextualizados. **Simpósio Internacional de Gestão do**

- Conhecimento**, Curitiba, PR. 2001. Disponível em: <<http://www.emack.com.br/info/apostilas/nestor/agentes.pdf>>. Acesso em: fev. 2006.
- CALONEGO JR., N., CONSULARO, L.A. **Manual do ARToolkit**. Piracicaba, SP: Faculdade de Ciências Matemática, da Natureza, UNIMEP, 2004.
- CONSULARO, L.A.; COELHO, R.C.; CALONEGO JR., N. Rastreamento Óptico para Sistemas de Realidade Virtual e Aumentada. **IX Symposium on Virtual and Augmented Reality**, Petrópolis, RJ, 2007.
- COSTA, R.G. et al. Integração de Técnicas de Modelagem com VRML e Java para Criação de Ambientes Virtuais em Realidade Aumentada. **Workshop de Realidade Virtual e Aumentada**, Bauru, SP, 2008.
- EID, M. et al. Trends in Mobile Agent Applications. **Journal of Research and Practice in Information Technology**, v. 37, n. 4, p.323-351, 2005. Disponível em: <<http://www.acs.org.au>>. Acesso em: jan. 2006.
- HAMIT, F. **Realidade Virtual e a Exploração do Espaço Cibernético**. Rio de Janeiro: Berkeley Brasil Editora, Original Sams Publishing, 1993.
- IBM. IBM Aglets Home Page. **IBM - International Business Machine**. Disponível em: <<http://www.trl.ibm.com/aglets>>. Acesso em: jun. 2005.
- JUNG, B.; MILDE, J.T. An Open Virtual Environment for Autonomous Agents Using VRML and Java. **Proceedings for 4th Symposium on Virtual Reality Modeling Language**, Paderbon, Germany, 1999. 5p.
- KIRNER, C.; SISCOUTTO, R.A. Fundamentos de Realidade Virtual e Aumentada. **IX Symposium on Virtual and Augmented Reality**, Petrópolis-RJ, 2007.
- LANGE, D.B. Java Aglet Application Programming Interface (J-AAPI). **White Paper, IBM Tóquio Research Laboratory**, 1997. Disponível em: <<http://www.research.ibm.com/trl/aglets/JAAPI-whitepaper.htm>>. Acesso em: jun. 2003.
- LANGE, D.B.; OSHIMA, M. **Programming and Deploying Java Mobile gents with Aglets**. Boston, MA: Addison-Wesley Longman Co. 1998. 225p.
- _____. Seven good reasons for mobile agents. **Communications of ACM**, v. 42. Issue 3, p. 88-89, 1999.
- LEA, R.; MATSUDA, K.; MIYASHITA, K. **Java for 3D VRML Worlds**. Ed. New Riders, 1996. 399p.
- MATTHIESEN, R.C. **VWBUILDER: Agentes Móveis com Realidade Aumentada**. 2006. Dissertação (Mestrado) - Universidade Metodista de Piracicaba, São Paulo.
- MILGRAM, P. et al. Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. **Telem manipulator and Telepresence Technologies. SPIE**, v.2351, p.282-292, 1994.
- NAGAMUTA, V. **Coordenação de Agentes Móveis Através do Canal de Broadcast**. 1999. Dissertação (Mestrado) - Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/45/45134/tde-11102005-123304>>. Acesso em: jun. 2004.
- NUNES, H.; LABIDI, S. Mobile agents for information extraction in Mathnet System. In: **32nd ASEE/IEEE Frontiers in Education Conference**, Session F2E, Boston, 2002. Disponível em: <<http://citeseer.ist.psu.edu/nunes02mobile.html>>. Acesso em: jan. 2006.
- RAPOSO, A.B. et al. **Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração. Tecgraf** - Grupo de Tecnologias em Computação Gráfica, Departamento de Informática, PUC - Rio de Janeiro, RJ. 2005, Disponível em: <<http://www.tecgraf.puc-rio.br/publications>>. Acesso em: fev. 2006.
- RODELLO, I.A. et al. Sistemas Distribuídos de Realidade Virtual e Aumentada. **IX Symposium on Virtual and Augmented Reality**, Petrópolis, RJ, 2007.
- SALVADOR, V.F.M.; KIRNER, T.G. Desenvolvimento de Ambientes Virtuais. **IX Symposium on Virtual and Augmented Reality**, Petrópolis, RJ, 2007.

SILVA, W.A. et al. Aplicações de Realidade Aumentada na Criação de Interfaces Distribuídas. **XI Simpósio de Realidade Virtual e Aumentada**, Porto Alegre, RS, 2009.

SINGHAL, S.; ZYDA, M. **Network Virtual Environment**. Designs and Implementations. Addison-Wesley, 1999, 331p.

TANENBAUM, A.S.; VAN STEEN, Maarten. **Sistemas Distribuídos: Princípios de Paradigmas**. 2. ed. São Paulo: Pearson Prentice Hall, 2007.

VELLOSO, P.B. et al. Uma Ferramenta de gerenciamento de redes baseada em agentes móveis. In: **Proceedings XIV Congresso Brasileiro de Automática**, Natal, RN, p.679-684, 2002.

WEB3D, **VRML97 SPECIFICATIONS**, 2004. Disponível em: <<http://www.web3d.org/x3d/specifications/vrml>>. Acesso em: mar. 2006.