

MODELO PARA GERENCIAMENTO REMOTO DE FALHAS EM
ARQUITETURAS DE AMBIENTES VIRTUAIS EM REDES

por

Luiz David S. de P. Campêlo

Dissertação Apresentada ao
Mestrado em Ciência da Computação
Universidade Federal do Ceará

Orientadores:
Mauro Cavalcante Pequeno, Dr.
Miguel Franklin de Castro, Dr.

Setembro, 2006

*“Até mesmo uma jornada de mil quilômetros
começa com um primeiro passo.”*

Lao Tse

Agradecimentos

Agradeço, primeiramente, a Deus, o primeiro e maior de todos os seres virtuais. A meus pais, José Maria e Marta, e familiares, pelo enorme incentivo recebido; à minha amiga e namorada, Lediane, por todo o carinho, atenção e todos os momentos preciosos que tivemos juntos mesmo em épocas tão difíceis.

Agradeço também ao professor Mauro Pequeno, que me confiou a difícil missão de concluir este curso, e aos professores Miguel Franklin e Fernando Carvalho, por todo o conhecimento transmitido nos últimos meses. Certamente, este trabalho não seria possível sem o apoio de tantos outros companheiros de laboratório, como Camilo, Melo Jr., W@nder, George, Glaudivney, Humberto, Gilvan, Palácio, Ricardo e tantos outros, que sempre indicaram boas soluções para os mais variados problemas.

Há também a quota de participação dos amigos distantes de Natal, João Pessoa, Recife e, especialmente, São Paulo, que sempre me abrigaram durante as mais divertidas viagens. Amigos de verdade só estes mesmo...

Aos amigos alvinegros, que nos momentos mais difíceis me mostraram que é preciso sempre seguir em frente e acreditar.

Por fim, então, agradeço aos que sempre acreditaram neste trabalho, que mesmo a distância estiveram virtualmente sempre ao meu lado com bons pensamentos.

Sumário

O uso das técnicas de Realidade Virtual para interação e colaboração entre usuários através das redes de computadores induz a criação dos Ambientes Virtuais em Redes (AVR). A crescente utilização e expansão das arquiteturas de AVR torna indispensável o gerenciamento dos recursos distribuídos utilizados por esses ambientes de modo que seja garantida uma boa qualidade nos serviços providos aos seus usuários. Esse gerenciamento envolve o monitoramento (supervisão) e o controle de aplicações distribuídas em máquinas conectadas à rede.

Especificamente para sistemas de ambientes virtuais voltados para trabalhos colaborativos, falhas tem conseqüências maiores. Nesses sistemas, usuários conduzem, em tempo real, tarefas pré-estabelecidas e direcionadas a um objetivo comum. Uma pane de sistema representaria a perda parcial ou total do trabalho colaborativo realizado (Gomes, 2005). Logo, é necessário um contínuo monitoramento e controle do sistema de ambientes virtuais de modo que as conseqüências de quaisquer falhas ocorridas sejam minimizadas. Os altos níveis de confiabilidade e disponibilidade exigidos podem ser alcançados com o emprego de técnicas de tolerância a falhas.

O gerenciamento do sistema AVR consiste no gerenciamento do ciclo de vida dos mundos virtuais instanciados neste sistema. Os principais requisitos para gerenciamento das instâncias dos mundos virtuais no sistema são mecanismos para criação, configuração, migração e exclusão dos mundos.

O principal objetivo dessa dissertação é a definição de um modelo para gerenciamento de uma arquitetura de ambientes virtuais em rede onde são usadas técnicas previstas na área de Gerência de Redes para fins de análise e resolução de falhas em componentes críticos do sistema. Por fim, é descrito um estudo de caso onde o modelo de gerenciamento de falhas é integrado a uma arquitetura de AVR.

Índice

Lista de Apêndices.....	7
Lista de Figuras	8
Lista de Tabelas	9
Capítulo 1	10
Introdução.....	10
1.1 Contexto e Motivação.....	10
1.2 Objetivos.....	11
1.3 Estrutura do trabalho	12
Capítulo 2	13
Gerenciamento de Redes	13
2.1 Introdução.....	13
2.2 Motivação	14
2.3 Arquitetura de gerenciamento de redes	16
2.4 Áreas funcionais	19
2.5 Padrões de gerenciamento	22
2.6 Considerações Finais	25
Capítulo 3	27
Ambientes Virtuais em Redes	27
3.1 Introdução.....	27
3.2 Características e Requisitos	29
3.3 Aplicações educacionais.....	33
3.5 Outras Aplicações.....	39
3.6 Considerações finais.....	42
Capítulo 4	43
Aspectos de desenvolvimento de Ambientes Virtuais em Redes.....	43
4.1 Visão geral.....	43
4.2 Escalabilidade.....	44
4.3 Comunicação em redes.....	46
4.4 Técnicas compensatórias	52
4.5 Gerência de falhas em AVR.....	55
4.6 Considerações finais.....	58

Capítulo 5	60
NuVEM – Um modelo para gerenciamento de AVR.....	60
5.1 Visão geral.....	60
5.2 Requisitos para gerência remota de AVR	60
5.3 Funcionamento do NuVEM	63
5.4 Trabalhos relacionados	73
5.5 Considerações Finais	74
Capítulo 6	75
Gerenciamento Remoto de Falhas em uma arquitetura de AVR.....	75
6.1 Introdução.....	75
6.2 Arquitetura CRAbCVE	75
6.3 Integração do NuVEM à arquitetura CRAbCVE	79
6.4 Testes de funcionalidade	86
Capítulo 7	91
Conclusões e Trabalhos Futuros.....	91
7.1 Principais contribuições.....	92
7.2 Trabalhos futuros.....	92
Referências Bibliográficas.....	95
Apêndices	102

Lista de Apêndices

Apêndice A – Código da interface JNI.....	103
Apêndice B – Diagrama de seqüência do teste de validação	105

Lista de Figuras

Figura 2.1. Camadas de uma arquitetura de gerenciamento (Stallings, 1999).....	16
Figura 2.2. Arquitetura básica para gerenciamento de redes.....	18
Figura 2.3. Modelo de gerenciamento distribuído.....	18
Figura 3.1. Simulação de batalha aérea no NPSNET-V.....	35
Figura 3.2. Centro de Turismo Virtual	36
Figura 3.3. Interface inicial do Laboratório de Química.....	37
Figura 3.4. Módulo de visualização de Moléculas.....	37
Figura 3.5 Interface da ferramenta de autoria.....	38
Figura 3.6. Alguns ângulos de visão de um treinamento.....	39
Figura 3.7. Visão do ambiente virtual AlphaWorlds.....	40
Figura 3.8. Interface do ambiente Blaxxun.....	41
Figura 3.9. Mundos virtuais disponíveis no Papo3D.....	41
Figura 4.1. Modelo centralizado.....	46
Figura 4.2. Modelo distribuído.....	47
Figura 4.3. Unicast (três mensagens enviadas, uma para cada usuário).....	49
Figura 4.4. Broadcast (apenas uma mensagem enviada, todos recebem).....	49
Figura 4.5. Multicast (uma mensagem enviada somente para o Grupo 3).....	50
Figura 5.1. Iniciação do Gerente.....	66
Figura 5.2. Iniciação do Agente.....	67
Figura 5.3. Instanciação remota.....	68
Figura 5.4. Resolução de falha em Mundo Virtual.....	70
Figura 5.5. Resolução de falha em Agente.....	71
Figura 5.6. Resolução de falha no Gerente.....	72
Figura 6.1. Componentes da arquitetura CRAbCVE.....	76
Figura 6.2. Integração do Gerente ao componente GPS.....	80
Figura 6.3. Integração do Agente aos componentes do sistema CRAbCVE.....	81
Figura 6.4. Integração dos componentes do NuVEM ao sistema CRAbCVE.....	83
Figura 6.5. Interfaces <i>web</i> e Java do Gerente.....	84
Figura 6.6. MBeans implementados pelo Agente.....	85
Figura 6.7. Interface AgentConnectionsMBean.....	85
Figura 6.8. Métrica simples para Instanciação remota.....	86
Figura 6.9 Aplicação emuladora de mundo virtual.....	87
Figura 6.10. Gerente iniciado em modo passivo (espera).....	88
Figura 6.11. Agente conectando ao Gerente BONO depois de receber notificação de falha no Gerente ENGHAW. Em seguida, o Agente recebe uma instância de Mundo Virtual 5.....	88
Figura 6.12. Agentes conectando ao Gerente reserva.....	89
Figura 6.13. Gerente recebendo notificação de falha em mundo virtual e repassando para nova instanciação.....	89
Figura 6.14. Gerente recuperando mundos virtuais após falha em Agente.....	90

Lista de Tabelas

Tabela 3.1. Requisitos de fidelidade com o mundo real em ambientes virtuais.	31
Tabela 4.1. Vantagens e desvantagens dos principais protocolos e técnicas.	Erro!

Indicador não definido.

Capítulo 1

Introdução

1.1 Contexto e Motivação

Nos últimos anos, a evolução das redes de computadores, representadas tanto pelas *intranets* quanto pela Internet, aumentou sensivelmente as possibilidades de interação entre indivíduos. A variedade de informações e as possibilidades de interação encontradas especialmente na *World-Wide Web* levaram pesquisadores a procurar novos meios para auxiliar a interação entre indivíduos de forma distribuída. Nesse contexto, houve um grande aumento de ambientes computacionais que suportam a realização de atividades cooperativas bem estruturadas entre pessoas geograficamente dispersas.

Em conseqüência dessa crescente utilização e expansão das redes de computadores e sistemas distribuídos, torna-se indispensável o gerenciamento do sistema para que seja garantida uma boa qualidade nos serviços providos aos seus usuários. Esse gerenciamento envolve o monitoramento (supervisão) e o controle de recursos distribuídos das máquinas conectadas à rede.

Nos últimos anos, a significativa evolução do poder de processamento gráfico dos computadores e o grande aumento da capacidade de transmissão de dados em redes de computadores têm viabilizado o uso de aplicações de Realidade Virtual (RV), tornando-as cada vez mais atrativas (Leite Jr, 2000).

Os Ambientes Virtuais em Rede (AVR) representam uso de tecnologias de RV para interação e colaboração entre usuários através das redes de computadores, possibilitando que múltiplos usuários localizados em diferentes partes do planeta interajam em tempo real, buscando, de modo geral, a criação de experiências imersivas por meio de imagens tridimensionais realistas e som estéreo (Singhal & Zyda, 1999; Çapin et. al, 1999).

No entanto, é comum haver simultaneamente um grande número de usuários inseridos no ambiente virtual, o que pode causar situações de comprometimento de desempenho do sistema, tendo em vista a crescente utilização dos recursos disponibilizados (Gomes, 2005). Além disso, devido à instabilidade das redes de comunicação e até mesmo dos próprios dispositivos físicos (discos rígidos, memórias,

processadores, placas-mãe, etc.), é bastante comum a ocorrência de perdas de dados momentâneas ou até mesmo partes gerais de sistema (Tanenbaum, 2002; Jalote, 1994).

Os altos níveis de confiabilidade e disponibilidade exigidos em sistemas de tempo real podem ser alcançados com o emprego de técnicas de tolerância a falhas. No entanto, a aplicação de tais técnicas apresenta vários desafios, ficando sua aplicação quase sempre restrita a sistemas de ultra confiabilidade, como computadores de bordo e controle de plantas nucleares, e a sistemas de alta disponibilidade, como centrais telefônicas e servidores *web*. Empregando sempre algum tipo de redundância, geralmente de componente de hardware, aplicações de tolerância a falhas aumentam tanto o custo de desenvolvimento como de fabricação do sistema (Tanenbaum, 2002).

Especificamente para sistemas de ambientes virtuais voltados para trabalhos colaborativos, falhas têm conseqüências maiores. Nesses sistemas, usuários conduzem, em tempo real, tarefas pré-estabelecidas e direcionadas a um objetivo comum. Uma pane de sistema representaria a perda parcial ou total do trabalho colaborativo realizado (Gomes, 2005). Assim, é necessário um contínuo monitoramento e controle do sistema de ambientes virtuais de modo que as conseqüências de quaisquer falhas ocorridas sejam minimizadas.

O gerenciamento do sistema de AVR consiste no gerenciamento do ciclo de vida dos mundos virtuais instanciados neste sistema. Os principais requisitos para gerenciamento das instâncias dos mundos virtuais no sistema são mecanismos para criação, configuração, migração e exclusão dos mundos. Para este fim, é importante que o administrador tenha ferramentas especializadas de gerenciamento que dêem uma visão ampla e transparente da localização das instâncias de mundos virtuais na rede (Almendra et al., 2003).

1.2 Objetivos

O presente trabalho objetiva explorar o uso de tecnologias de gerenciamento de redes para o desenvolvimento de um modelo que possibilita administrar de modo mais otimizado uma arquitetura de AVR, provendo mecanismos para detecção e resolução de falhas no sistema. Além disso, o modelo proverá ao administrador do sistema ferramentas simples e portáteis para gerenciamento de maneira unificada.

Para isso, serão observados os impactos da utilização dos componentes da arquitetura em tempo de execução, verificando possíveis falhas a serem resolvidas. Por conseguinte, são pesquisados meios para adicionar à arquitetura mecanismos para

automatizar e simplificar o processo de disponibilização e recuperação (em caso de falha) de ambientes virtuais.

Com base nessas observações, será definido um modelo para gerência de falhas em mundos virtuais de sistemas AVR, o NuVEM¹. De modo geral, este modelo consiste na utilização de mecanismos de gerência (e, em especial, mecanismos de tolerância a falhas) em uma arquitetura de AVR. Desde já, é prevista a implementação de aplicações responsáveis pelo controle e monitoramento do sistema.

Como estudo de caso, o NuVEM será incorporado à arquitetura CRAbCVE (Gomes, 2005), que é uma arquitetura de AVR com propósitos educacionais.

1.3 Estrutura do trabalho

O presente trabalho encontra-se organizado em sete capítulos, cujos conteúdos estão resumidos a seguir.

No Capítulo 2, apresentam-se os conceitos pertinentes à área de Gerenciamento de Redes, discutindo-se seus aspectos motivacionais, funções de gerenciamento, padrões, etc. No Capítulo 3, a tecnologia de AVR é apresentada, verificando-se seus conceitos e requisitos necessários para disponibilizá-los através de redes de computadores. Ainda no Capítulo 3 são detalhados alguns exemplos de AVR. No Capítulo 4, discutem-se os aspectos de desenvolvimento e gerência de AVR, relatando-se principalmente aspectos de comunicação em redes, como modelos de transmissão e protocolos. Além disso, no Capítulo 4, expõem-se conceitos de tolerância a falhas voltados para sistemas distribuídos. O Capítulo 5 apresenta o NuVEM, um modelo para gerenciamento de falhas para uma arquitetura de AVR, detalhando cenários de utilização e componentes envolvidos. O Capítulo 6 apresenta os detalhes do estudo de caso realizado utilizando a arquitetura de ambientes virtuais CRAbCVE (Gomes, 2005). Por fim, no Capítulo 7, são apresentadas as conclusões deste trabalho e indicadas algumas sugestões de trabalhos futuros.

¹ Do inglês, “Networked Virtual Environment Management model”.

Capítulo 2

Gerenciamento de Redes

2.1 Introdução

Devido à popularização do computador nas últimas décadas, tornou-se muito comum o uso de redes de computadores em residências, instituições acadêmicas e estabelecimentos comerciais e governamentais. Tais redes são muitas vezes conectadas à Internet e oferecem vários tipos de recursos interativos e serviços aos seus usuários. No entanto, apesar de quase duas décadas de pesquisas nas áreas de processamento distribuído e balanceamento de cargas, ainda é comum a implementação de redes de computadores de forma simplista e ineficiente (Kon & Marques, 2002).

As facilidades alcançadas com o uso das redes de computadores tendem a estimular seu crescimento, o que exige um maior planejamento e manutenção. Ao se considerar uma rede distribuída por vários lugares fisicamente dispersos, a tarefa de gerenciamento torna-se complexa e indispensável. Além disso, com a grande utilização de dispositivos heterogêneos ligados à rede, é necessário haver um esquema que ofereça soluções de gerenciamento adaptáveis e estruturadas, permitindo a monitoração de equipamentos compatíveis ou não (Pereira, 2001).

Portanto, com o crescimento das redes, devido ao número tanto de usuários quanto de dispositivos conectados, motivaram o desenvolvimento de ferramentas de gerência de redes. A área de gerência de redes foi inicialmente impulsionada pela necessidade de monitoração e controle do universo de dispositivos que compõem as redes de comunicação. Contudo, não foi preciso muito tempo até que os mecanismos de gerência de redes fossem aplicados também à gerência de aplicações distribuídas ao longo das redes.

De acordo com Oliveira e seus co-autores (1998), Gerenciamento de Redes é uma área de estudo conceituada como a coordenação tanto de recursos materiais (dispositivos físicos) como de recursos lógicos (aplicações, protocolos e serviços), ambos distribuídos pela rede. Assim, gerenciar uma rede significa adicionar ferramentas específicas para inicialização, monitoração e controle de recursos de modo a permitir o funcionamento da rede dentro de um patamar de Qualidade de Serviço (QoS) aceitável para os usuários e garantido, na medida do possível, confiabilidade e segurança na transação de informações.

Com o intuito de resolver os problemas de gerenciamento de redes de grande heterogeneidade foram desenvolvidos os padrões OSI e SNMP, que fornecem soluções coerentes e estruturadas, permitindo a administração de recursos independente de suas compatibilidades (Carvalho, 1993). No entanto, na prática estes padrões se mostraram ineficientes para o gerenciamento de grandes redes, o que proporcionou a criação de padrões de gerência baseados em tecnologias *web*.

Ademais, o desenvolvimento de um sistema de gerência deve ser guiado pelo fluxo de serviços de cada organização, levando sempre em consideração suas peculiaridades. Se as facilidades de gerenciamento não resolvem direta ou indiretamente os problemas de uma rede, o sistema de gerência torna-se inútil.

São atividades comuns do gerenciamento de redes registrar a ocorrência de eventos específicos, detectar e solucionar falhas na rede, acompanhar o desempenho da rede, contabilizar e garantir a segurança física e lógica dos recursos.

São vários os aspectos motivacionais para o uso de mecanismo de gerência de redes. Alguns dos principais aspectos são discutidos na seção 2.2.

Basicamente, a tarefa de gerência consiste em três etapas, a de coleta de dados (oriundos dos recursos gerenciados), diagnóstico (tratamento e análise a partir dos dados coletados) e ação (a ser realizada sobre os recursos). Para isso foram desenvolvidos os modelos centralizado e distribuído de arquitetura de gerenciamento, vistos com mais detalhes na seção 2.3.

Com o objetivo de modelar os requisitos de gerência, a área de Gerenciamento de Redes foi dividida em 5 (cinco) áreas funcionais, as quais serão abordadas na seção 2.4.

Em seguida, na seção 2.5, são mostrados alguns dos padrões de gerenciamento mais utilizados, como o SNMP e o OSI, além de outros novos padrões que estão se firmando nos últimos anos. Por fim, na seção 2.6, tecem-se algumas considerações finais.

2.2 Motivação

Usualmente, numa grande organização são estipuladas metas com o intuito de torná-la cada vez maior. Assim, à medida que essa organização cresce, é comum haver uma dependência funcional em seu sistema de comunicação. O crescimento da organização implica no crescimento do sistema de comunicação, aumentando o número de usuários e recursos inseridos. Como consequência disso, surge a importância do uso de mecanismos de gerência eficientes (Carvalho, 1993).

Assim, as redes de computadores e os seus recursos associados têm tido tanta importância para uma organização que elas basicamente "não podem falhar", fazendo com que os níveis de ocorrência de falhas e de degradação de desempenho considerados aceitáveis diminuam cada vez mais. Em alguns casos, estes níveis podem tender a quase zero, dependendo da organização e do tipo de aplicações utilizadas (Stallings, 1999).

No entanto, a tarefa de gerenciamento de redes é por si só complexa. Na proporção em que as redes tornam-se complexas (em tecnologias), maiores (em extensão) e heterogêneas (em elementos utilizados, tanto *hardware* como *software*), mais complicado torna-se o gerenciamento. Logo, o gerenciamento não deve ser realizado somente por recursos humanos especializados, é preciso ter ferramentas automatizadas, pois estas permitem simplificar o processo de gerência. Além disso, possibilitam a concentração das funções de monitoramento e controle do funcionamento da rede em um ou mais pontos, viabilizando que um único operador exerça efetivo controle sobre uma infra-estrutura distribuída, independente das distâncias entre os recursos envolvidos.

São vários os benefícios provenientes do uso de mecanismos para gerenciamento de redes. Como exemplo, a seguir são expostos alguns destes benefícios. Esta lista de benefícios é baseada nos estudos feitos por Pras (1995), Sharda e seus co-autores (1997) e Gupta e seus co-autores (1999):

- ◆ **Redução de custos** - através do uso de mecanismos de gerência é possível reduzir os custos totais de propriedade do sistema de comunicação. Dentre os custos estão desde recursos humanos especializados para manutenção do sistema até recursos físicos e/ou lógicos utilizados.
- ◆ **Melhoria no desempenho e aumento de disponibilidade do sistema** – através do acompanhamento dos níveis de utilização da rede é possível reduzir gargalos no sistema de forma pró-ativa. Além disso, é possível detectar e solucionar falhas nas redes rapidamente por meio do uso de ferramentas de monitoração, aumentando o grau de disponibilidade dos serviços providos pelo sistema.
- ◆ **Maior controle no acesso a recursos** – dada a alta demanda de recursos durante a fase de execução do sistema, torna-se complicada a contabilização da utilização dos mesmos. Logo, através de mecanismos de bilhetagem é possível cobrar por utilização de recursos. Além disso, simplifica um possível planejamento de expansão do sistema através da verificação dos pontos de grande utilização ou gargalos na rede.

- ◆ **Aumento dos níveis de segurança** – dependendo do tipo de aplicação, pode ser requisito de alta prioridade a segurança dos recursos ligados ao sistema. Logo, através do uso de mecanismos de segurança torna-se possível um maior controle sobre os recursos gerenciados.
- ◆ **Aumento da flexibilidade dos procedimentos operacionais** – através da simplificação da execução de procedimentos e tarefas de manutenção usuais como automação de tarefas, realização de inventários, distribuição de *softwares* entre outras.
- ◆ **Maior adaptação mediante mudanças de requisitos do sistema** – é bastante comum haver mudanças nos requisitos de usuários durante a fase operacional do sistema. Logo, através do uso de ferramentas de gerenciamento é possível adaptar o sistema, de maneira efetiva, às mudanças decorrentes, reaproveitando ao máximo a estrutura do sistema, bem como seus recursos gerenciados.

Como consequência dos benefícios supracitados, tem-se a melhoria na qualidade de serviços providos aos usuários do sistema, bem como o aumento na facilidade de administração do sistema como um todo.

2.3 Arquitetura de gerenciamento de redes

Uma arquitetura para um sistema de gerência consiste numa coleção de ferramentas para monitorar e controlar recursos e serviços contidos na rede. Esta arquitetura é composta por quatro camadas de funcionalidades e uma camada de dados, além do protocolo de gerenciamento (Figura 2.1).

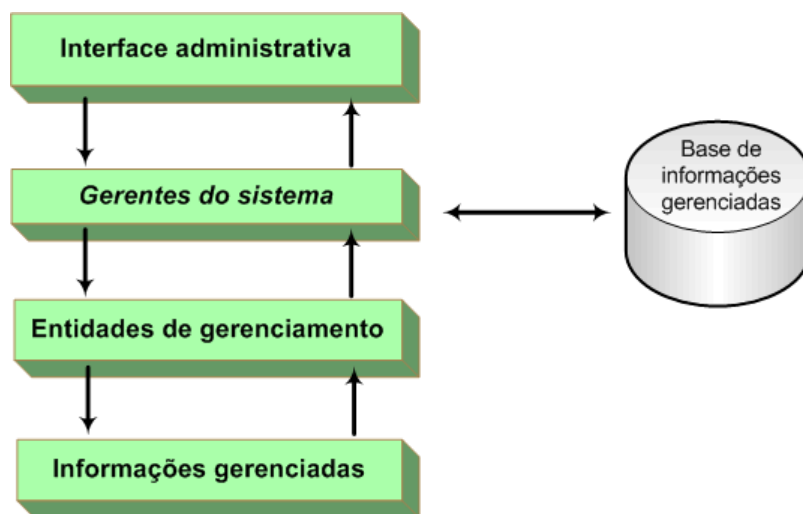


Figura 2.1. Camadas de uma arquitetura de gerenciamento (Stallings, 1999).

- ◆ Base de informações gerenciadas – compreende um conjunto de variáveis usadas para representar informações estáticas ou dinâmicas vinculadas a um determinado recurso ou serviços gerenciado. Grande parte das funcionalidades da arquitetura destina-se à troca de dados existente na base de informações gerenciadas. É mais conhecida na literatura como MIB (*Management Information Base*).
- ◆ Objetos gerenciados – são dispositivos, aplicações ou quaisquer outros recursos e serviços a serem monitorados/controlados pela arquitetura.
- ◆ Entidades de gerenciamento – aplicações contidas nas estações do sistema com o objetivo de coletar e armazenar dados acerca dos objetos locais gerenciados, além de responder às requisições feitas pelos gerentes do sistema. Geralmente, são mais conhecidas como agentes.
- ◆ Gerentes do sistema – uma ou mais entidades de gerenciamento designadas para reunir informações provenientes de outras entidades, formando uma base de dados para análise e geração de relatórios gerais. Adicionalmente, podem programar outras entidades de gerenciamento (agentes) para enviarem notificações de modo automático mediante regras pré-estabelecidas.
- ◆ Interface administrativa – interface (gráfica) pela qual o administrador controla e monitora o sistema, configurando, requisitando e recebendo informações ou parâmetros específicos.

O modelo mais utilizado para desenvolvimento de uma arquitetura de gerenciamento é o centralizado, onde são implementados um gerente e vários agentes. A comunicação do gerente com os agentes se dá por meio de um protocolo de gerenciamento e todas as informações relacionadas ao sistema são armazenadas em uma base de dados central (Figura 2.2) (Stallings, 1999). Como consequência da centralização, neste modelo é alta a probabilidade de sobrecarga do gerente, o que poderia causar uma parada geral do sistema.

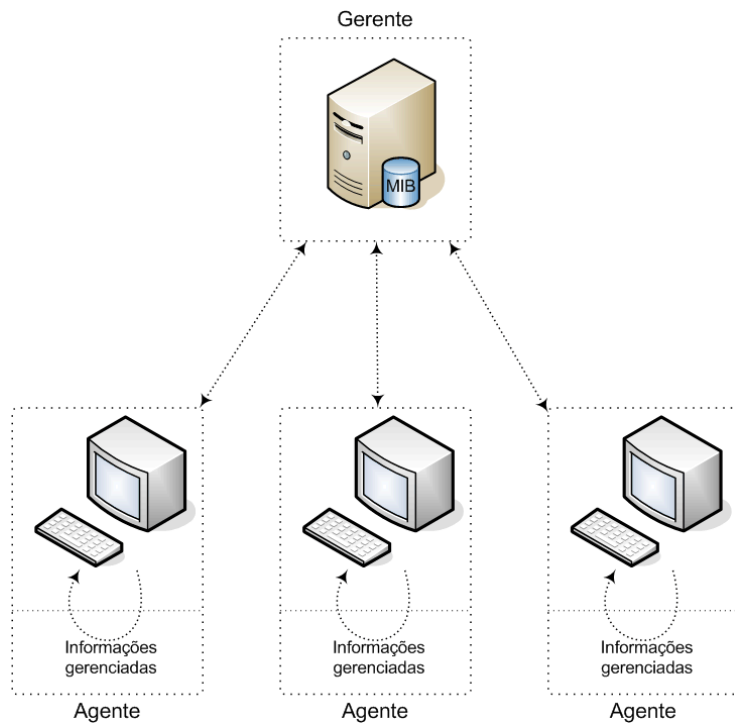


Figura 2.2. Arquitetura básica para gerenciamento de redes.

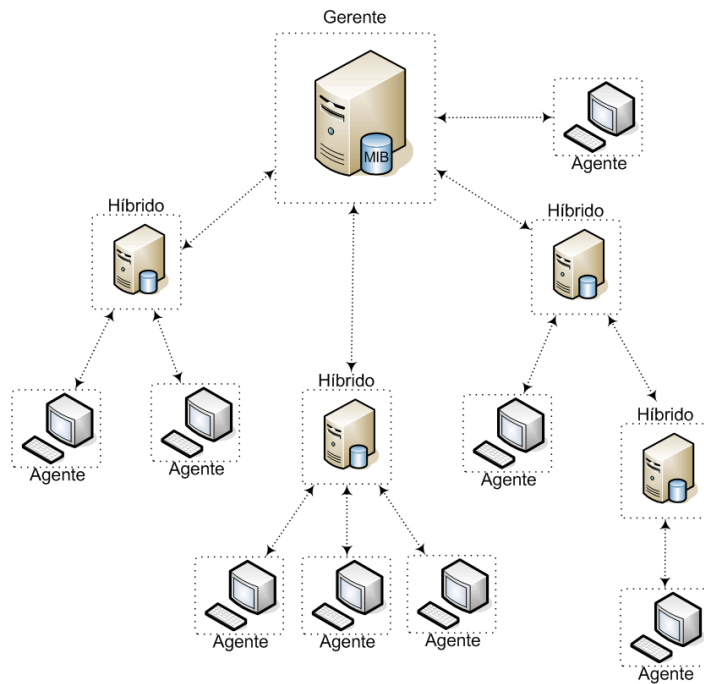


Figura 2.3. Modelo de gerenciamento distribuído.

Alternativamente, para grandes redes pode ser implementado um modelo distribuído de gerenciamento, onde se tem vários subsistemas de gerenciamento independentes entre si.

Há ainda a possibilidade de distribuir o processamento do gerente do sistema dividindo a arquitetura em uma hierarquia de gerentes e agentes. Como consequência

disso, são criadas entidades de gerenciamento híbridas que ora atuam como gerente e ora atuam como agente (Figura 2.3) (Leinwand & Fang, 1993).

2.4 Áreas funcionais

De acordo com o modelo de gerenciamento OSI (*Open Systems Interconnect*), a tarefa de gerenciamento de redes foi definida em 5 (cinco) áreas funcionais. Embora essa classificação funcional tenha sido desenvolvida para um ambiente específico, é grande sua aceitação por vários fabricantes de equipamentos e desenvolvedores de padrões de gerência. Através dessa divisão é possível definir requisitos de usuários e de sistema. As áreas funcionais consistem em (Stallings, 1999; Stallings, 1993; Oliveira, 1998; Pras, 1995; Carvalho, 1993):

- ◆ Gerenciamento de Falhas
- ◆ Gerenciamento de Contabilização
- ◆ Gerenciamento de Desempenho
- ◆ Gerenciamento de Configuração
- ◆ Gerenciamento de Segurança

Muitas vezes, as áreas funcionais possuem funções de gerenciamento que se sobrepõem, isto é, são utilizadas não somente em uma, mas até mesmo em várias áreas funcionais, apesar de terem finalidades diferentes em cada uma. Há também casos em que uma função serve de suporte para uma outra função em outra área.

2.4.1 Gerenciamento de Falhas

Consiste num conjunto de instrumentos para detecção, localização, isolamento e correção de falhas, bem como para geração de relatórios acerca de anormalidades na operação da rede. É de responsabilidade desta área coletar continuamente informações sobre o estado dos recursos gerenciados e assegurar o bom funcionamento dos componentes do sistema. Em caso de ocorrência de uma falha no sistema, é necessário que o mecanismo de gerência de falhas opere de modo rápido e eficaz, tomando decisões de acordo com regras pré-estabelecidas.

Alternativamente, pode-se gerar um registro completo das falhas ocorridas, diagnósticos e suas respectivas soluções, resolvendo a falha com maior rapidez em caso de reincidência.

Apesar de exigir processamento adicional, é recomendado que o sistema detecte problemas na rede de modo que as falhas sejam solucionadas antes que seus efeitos sejam

percebidos. Para este fim, podem ser usados mecanismos de monitoração de taxas de erros e notificações de eventos indesejados, gerando alarmes para o gerente do sistema, que, então, poderá definir ações necessárias para corrigir o problema e evitar danos críticos.

Dentre as funções do Gerenciamento de Falhas, estão:

- ◆ Procurar e identificar problemas;
- ◆ Receber e resolver notificações de detecção de erros;
- ◆ Desencadear seqüências de testes de diagnóstico;
- ◆ Isolar os pontos de falhas;
- ◆ Corrigir os problemas; e
- ◆ Manter e examinar o histórico de erros.

Maiores detalhes acerca do desenvolvimento de sistemas voltados para tolerância a falhas serão expostos no capítulo 4.

2.4.2 Gerenciamento de Contabilização

Tem como objetivo medir os níveis de utilização dos recursos do sistema por parte de um usuário ou de um grupo de usuários para posterior tarifação. Alternativamente, podem ser registrados no sistema, dados acerca dos limites de utilização específicos de cada usuário ou grupo de usuários, bem como tarifas personalizadas.

A partir dos registros de utilização do sistema é possível desenvolver um planejamento para um crescimento mais adequado do sistema, minimizando custos de expansão e maximizando o grau de confiabilidade do sistema.

São funções do Gerenciamento de Contabilização:

- ◆ Estipular tarifas específicas para utilização os recursos do sistema;
- ◆ Registrar níveis de utilização por usuários ou grupos de usuários;
- ◆ Gerar relatórios de utilização para planejamento de possíveis expansões do sistema.

2.4.3 Gerenciamento de Desempenho

É de responsabilidade desta área do gerenciamento de redes coletar e analisar dados sobre o funcionamento e a eficácia dos serviços dispostos no sistema, verificando o desempenho corrente da rede através de parâmetros estatísticos, como atraso, vazão, disponibilidade e número de conexões e retransmissões realizadas.

Com o objetivo de manter o sistema com desempenho em nível adequado, o sistema de gerência determina taxas mínimas de desempenho aceitável. Caso seja percebida uma situação em que o sistema tenha um desempenho inferior às taxas calculadas, será

disparada uma notificação para o gerente do sistema, que, de posse desta notificação, poderá tomar decisões cabíveis para solucionar o problema, possivelmente em conjunto com as ferramentas de Gerenciamento de Falhas.

De modo adicional, podem ser registrados os eventos relativos ao desempenho dos serviços disponíveis no sistema de modo a detectar pontos de grande utilização ou excesso de carga computacional. Logo, de modo similar ao Gerenciamento de Contabilização, estes registros podem ser usados para planejar adequadamente o crescimento do sistema.

A seguir, as funções do Gerenciamento de Desempenho:

- ◆ Monitorar os recursos do sistema;
- ◆ Determinar níveis de desempenho aceitável;
- ◆ Detectar pontos de desempenho abaixo do esperado;
- ◆ Garantir o desempenho adequado;
- ◆ Registrar eventos relativos ao desempenho dos recursos.

2.4.4 Gerenciamento de Configuração

É formado por um conjunto de ferramentas responsáveis pela preparação, inicialização e manutenção dos diversos serviços e recursos contidos no sistema. Para isso, o sistema deve registrar informações concernentes a estes serviços e recursos, como localização, estado, tempo de vida, carga, dentre outros. Para fins de manutenção do sistema em tempo de execução, deve haver meios para alterar atributos dos serviços e recursos gerenciados.

Assim, a área de Gerenciamento de Configuração tem basicamente o dever de controlar os serviços e recursos dispostos na rede, identificando, coletando e alterando suas informações. São suas as seguintes funções:

- ◆ Gerenciar os parâmetros relativos a serviços e recursos contidos na rede;
- ◆ Preparar, iniciar e encerrar a execução de serviços e recursos gerenciados; e
- ◆ Prover meios de modificar configurações em tempo de execução.

2.4.5 Gerenciamento de Segurança

Consiste em um conjunto de mecanismos para controlar o acesso aos recursos gerenciados de acordo com regras pré-estabelecidas, garantindo que somente usuários autenticados e autorizados farão uso destes recursos. Para isso, o sistema de gerência deve manter controle sobre o cadastramento de usuários e suas respectivas permissões (Stallings, 2000).

São requisitos básicos de um sistema de segurança assegurar a integridade, confidencialidade e autenticidade das informações enviadas e recebidas. Para isso, o sistema de gerência deve utilizar protocolos seguros de comunicação usados com a finalidade de prover comunicação segura, bem como utilizar *firewalls* (Cheswick & Bellovin, 1994; Carl-Mitchell & Quarterman, 1992).

Assim como em outras áreas do gerenciamento, é recomendável que sejam gravados registros de eventos ocorridos no sistema para posterior verificação.

Dentre as funções do Gerenciamento de Segurança, estão:

- ◆ Controlar cadastros de usuários;
- ◆ Autenticar usuários;
- ◆ Autorizar (ou não) o acesso a recursos do sistema;
- ◆ Assegurar a integridade, a confidencialidade e a autenticidade das informações;
- ◆ Utilizar protocolos de comunicação segura e *firewalls*; e
- ◆ Registrar eventos do sistema.

2.5 Padrões de gerenciamento

O desenvolvimento de arquiteturas distribuídas e heterogêneas bem como sua integração têm tornado cada vez mais complexa a tarefa de gerência de redes. Uma solução para a questão da heterogeneidade é a adoção de padrões de gerenciamento.

Nas décadas passadas, vários padrões foram desenvolvidos, dentre eles os principais foram o CMIP/CMIS – *Common Management Information Protocol/Common Management Information Service* – (OSI, 1991) e o SNMP – *Simple Network Management Protocol* (McCloghrie & Rose, 1991; Davin & McCloghrie, 1993; Case et al., 1993). Além disso, nos últimos anos tem-se destacado uma iniciativa para padronização do gerenciamento com enfoque em tecnologias *Web*, como o WBEM.

Em geral, cada padrão especifica uma estrutura de informações gerenciadas e um protocolo para comunicação entre as entidades de gerenciamento. A seguir, são descritos em detalhes cada um desses padrões de gerenciamento.

2.5.1 Padrão SNMP

O padrão SNMP foi desenvolvido pela IETF (*Internet Engineering Task Force*) com o propósito de gerenciar dispositivos e serviços em redes TCP/IP.

Este padrão é composto por um protocolo de gerenciamento, o SNMP, e uma especificação de uma estrutura de informações gerenciadas, a SMI (*Structure of*

Management Information), que comporá a MIB do sistema de gerência. Atualmente, três versões foram especificadas, SNMPv1 e SNMPv2 e SNMPv3.

O protocolo SNMP é baseado em poucos tipos mensagens (PDUs – *Protocol Data Units*) trocadas entre elementos gerenciadores (gerentes) e gerenciados (agentes). Ao definir poucas PDUs, os criadores do SNMP procuraram simplificar ao máximo o protocolo SNMP, limitando as restrições impostas às antigas ferramentas de gerenciamento, pois, na medida em que se aumenta a complexidade do protocolo e as possibilidades de suas mensagens, dificulta-se a criação das ferramentas de gerenciamento devido ao uso de operações complicadas (Stallings, 1999).

A flexibilidade oferecida por todas estas características possibilitou a criação de um maior número de funções de gerenciamento suportadas remotamente nos elementos de rede, tornando a arquitetura de gerência SNMP extensível para acomodar funcionalidades adicionais, de forma independente da plataforma onde está sendo executada.

Projetado inicialmente para aplicações de gerência de redes simples, o padrão SNMP firmou-se no mercado e é atualmente o mais utilizado em redes de comunicação de dados devido a sua interoperabilidade e simplicidade.

2.5.2 Padrão CMIP/CMIS

Em 1979, a ISO (*International Organization for Standardization*) em atividade com a atual ITU-T (*International Telecommunication Union*) desenvolveu um modelo OSI (*Open Systems Interconnection*) para comunicação entre sistemas abertos. Esses sistemas apresentam intercâmbio com o ambiente através de entradas e saídas, além de ter uma grande capacidade de adaptação e crescimento escalar (Rose, 1991). O padrão OSI fornece um modelo mais complexo e poderoso (abrangente) de gerência que o SNMP. Por ser um padrão internacional definido pela ISO, permite que os fabricantes testem suas implementações através de testes de conformidade e de interoperabilidade.

O padrão OSI é composto por um conjunto de aplicações de propósito geral para gerência de redes, um serviço de gerenciamento e protocolo (CMIP - *Common Management Information Protocol*), uma especificação de estrutura de dados, e um conjunto de objetos de dados (OSI, 1991). Pela sua complexidade, e pela lentidão do processo de padronização, este sistema de gerenciamento não é muito popular.

2.5.3 Gerenciamento baseado em tecnologias *web*

Desde os anos 80 até o início da década atual, o padrão SNMP tem dominado o mercado de gerência. No entanto, nos últimos anos as redes têm contado com recursos e serviços cada vez mais complexos e heterogêneos, o que tornou a simplicidade do SNMP extremamente limitante no processo de gerência de grandes redes. Para resolver estas limitações do SNMP surgiram várias pesquisas na área de gerenciamento de redes através de tecnologias *Web* (Ju et al., 2002; Martin-Flatin, 2000).

No entanto, devido à grande aceitação do SNMP pelo mercado, há muitos sistemas de comunicação que já utilizam o padrão SNMP. Logo, com o intuito de reaproveitar ao máximo os dispositivos que implementam o SNMP, estão sendo desenvolvidas alternativas onde é prevista uma integração das tecnologias *Web* de gerência com este ambiente (Choi et al., 2003). Especificamente, esta integração consiste no desenvolvimento de um protocolo e uma base de informações com o poder de traduzir informações de um padrão para o outro.

Basicamente, as plataformas de gerenciamento baseadas em tecnologias *web* fazem uso do protocolo HTTP (onde o conteúdo das mensagens é XML²) para comunicação entre as entidades do sistema. Além disso, a base de informações também é construída de acordo com padrões de descrição de dados em XML e é utilizado como interface administrativa um simples navegador (*browser*) Internet.

Ao permitirem a monitoração e o controle da rede a partir de navegadores, estas plataformas têm-se mostrado bastante atrativas. Dentre suas principais vantagens destacam-se a grande portabilidade do sistema de gerência, pois torna-se possível monitorar e controlar os elementos da rede usando qualquer navegador em qualquer nó da rede. Antes, nas plataformas mais comuns, os administradores da rede só podiam usar as ferramentas de gerência em máquinas onde estivessem previamente instaladas e configuradas (Hyde, 1997).

Além disso, não é preciso usar ferramentas diferentes para gerenciar os diversos elementos da infra-estrutura de TI, inclusive serviços e aplicações em execução. Alguns padrões para gerência baseada em *web* já estão se firmando. Dentre eles, destaca-se o WBEM – *Web-Based Enterprise Management* – (WBEM, 2005), que é um padrão

² XML (eXtensible Markup Language) é uma linguagem de marcação capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações através da Internet (Graham & Quim, 2005).

totalmente orientado a objetos que propõe a definição de um modelo de dados padrão que possa ser utilizado como repositório de dados de gerência, e que seja possível acessar estes dados gerenciados via *web*, com HTTP, XML e um gerenciador de objetos, o CIMOM, que é responsável por fornecer e gerenciar o acesso a esta base de objetos denominada de CIM e fornecer a comunicação com os provedores (*providers*) de instâncias das classes (Job & Simões, 2002).

Além do WBEM, existe uma alternativa da *Sun Microsystems*, que é o JMX (Kreger, 2001; Sun Microsystems, 1999) – *Java Management Extensions*. O JMX consiste em um *framework* de componentes para construir um ambiente de gerenciamento completo e flexível. JMX contém componentes especializados para gerenciar aplicações, dispositivos ou recursos específicos, provendo acesso à instrumentação e ao controle dos sistemas gerenciados. Para isso, o JMX define um conjunto de componentes acessíveis por meio de múltiplos protocolos, como SNMP, HTTP ou SOAP. Com JMX, uma mesma aplicação pode ser implementada um só vez e acessada por vários protocolos diferentes, o que diminui bastante o tempo de manutenção do sistema de gerência em caso de mudança no protocolo de comunicação. Cada um dos protocolos utilizados pelo JMX é chamado de conector (*JMXConnector*).

Em JMX, cada recurso gerenciado é chamado de *MBean*, que é uma interface Java com métodos para acesso aos dados gerenciados. Estas interfaces devem ser implementadas por classes de objetos destinados ao gerenciamento da aplicação. Estes objetos são agrupados em um *MBeanServer*, que é um repositório de objetos gerenciados acessado por meio dos conectores.

2.6 Considerações Finais

A grande demanda por qualidade de serviço tem impulsionado o desenvolvimento da área de gerenciamento de redes. Esta área da Ciência da Computação consiste em mecanismos para monitoração e controle de recursos e serviços distribuídos ao longo dos sistemas de redes de comunicação. Com base no grande crescimento e heterogeneidade destes sistemas é que os mecanismos de gerência devem ser implementados.

Este capítulo apresentou uma fundamentação teórica sobre sistemas de gerenciamento, arquiteturas e padrões mais conhecidos, bem como requisitos de suas áreas funcionais. Com bases nas análises apresentadas neste capítulo foi possível a identificação de estratégias a serem usadas durante o desenvolvimento do modelo de gerenciamento de Ambientes Virtuais em Redes, o NuVEM.

Assim, tanto as estratégias de gerenciamento de dispositivos e aplicações previstas nos padrões de gerenciamento SNMP, OSI e *Web*, quanto os modelo de arquitetura gerente-agentes podem ser adequados para o gerenciamento de aplicações de Realidade Virtual distribuídas. Estas estratégias consistem em mecanismos para de detecção e recuperação de falhas, configuração automática do sistema, monitoramento de desempenho e segurança de dados.

Capítulo 3

Ambientes Virtuais em Redes

3.1 Introdução

De acordo com Çapin e seus co-autores (1999), Realidade Virtual (RV) pode ser definida como uma tecnologia capaz de transportar um indivíduo para um ambiente diferente do real, sem movê-lo fisicamente, manipulando as informações destinadas aos órgãos sensoriais humanos de tal maneira que o ambiente percebido seja associado ao ambiente virtual desejado e não ao ambiente real.

Por meio de dispositivos de interação, as aplicações de RV proporcionam ao usuário sensações de imersão, no ambiente virtual, compatíveis com aquelas que ele experimenta no mundo real. Um ambiente imersivo ideal é aquele onde os órgãos sensoriais do usuário são estimulados de maneira contínua pelo sistema computacional (Slater et al., 1994).

Os sistemas de RV mais especializados demandam uso de dispositivos de interação e visualização de custo ainda muito elevado, como luvas, capacetes, etc. Já os sistemas chamados semi-imersivos (*Desktop Virtual Reality*³) utilizam dispositivos comuns, como teclado, mouse e fones de ouvidos, ligados a computadores domésticos e não necessitam de *hardware* especial. No entanto, proporcionam ao usuário um nível de imersão mais limitado.

Para prover uma maior interatividade entre usuários dispersos fisicamente, as aplicações de RV convencionais foram dotadas de ferramentas de comunicação por meio das redes de computadores, resultando na criação dos Ambientes Virtuais em Redes (AVR ou net-VEs⁴) ou Ambientes Virtuais Distribuídos (AVDs).

Ambientes Virtuais em Redes possibilitam a múltiplos usuários localizados em diferentes partes do planeta interagirem em tempo real, buscando, de modo geral, a criação de experiências imersivas por meio de imagens tridimensionais realísticas e som estéreo (Singhal & Zyda, 1999). Nos últimos anos, a significativa evolução do poder de

³ Modalidade de realidade virtual semi-imersiva que faz uso de equipamentos comuns, como monitores de vídeo, mouse, teclado e *joysticks* (Machado, 1995).

⁴ Do inglês *Networked Virtual Environments*.

processamento gráfico dos computadores e o grande aumento da capacidade de transmissão de dados em redes de computadores têm viabilizado estas aplicações, tornando-as cada vez mais atrativas (Leite Jr, 2000).

Trabalhos em torno de AVR foram iniciados na década de 80 com o desenvolvimento de ambientes virtuais para uso em treinamentos militares do Departamento de Defesa dos EUA (Matijasevic, 1997). Durante esses trabalhos pioneiros foi criado o SIMNET, que foi posteriormente estendido e rebatizado de protocolo DIS. Em seguida, veio o NPSNET, outra continuação dos trabalhos do SIMNET, mas dessa vez por parte da *Naval Postgraduate School* (Macedonia et al., 1995). Desde então, o uso de ambientes virtuais tem formado uma nova base para a criação de aplicações interativas em diversas áreas, tais como entretenimento (Leite Junior et al., 2002; Cardoso et al., 2003), medicina (Freitas et al., 2003), arquitetura (Silva et al., 2003), teleconferência, comércio, entre outras.

Particularmente, para propósitos educacionais, estes sistemas AVR têm tido grande destaque, propiciando interação e colaboração entre seus usuários em tempo real. Em (Vidal et al., 2003; Wazlawick et al., 1999; Leite Junior et al., 2001; Gomes, 2005) são encontrados alguns trabalhos em torno do desenvolvimento e uso de AVR voltados para a área de educação. A principal motivação para o uso em aplicações com fins educacionais advém da possibilidade de apresentar a complexidade de situações do mundo real de tal maneira que usuários possam observar diretamente os resultados de suas próprias decisões, além de melhorar o desempenho coletivo através da troca de informações (Leite Junior et al., 2000).

Os AVR são considerados os sistemas de software mais complexos já desenvolvidos (Stytz, 1996), pois apresentam características de vários outros tipos de sistemas computacionais. Logo, exigem conhecimento em diversas áreas durante seu desenvolvimento, como protocolos de rede, processamento paralelo e distribuído, modelagem de ambientes gráficos (tridimensionais), bancos de dados, interface homem-computador, etc.

Segundo Singhal & Zyda (1999), um sistema AVR é composto basicamente de quatro componentes:

- ◆ **motor gráfico e dispositivos de apresentação**, para geração e visualização de imagens tridimensionais;
- ◆ **dispositivos de interação e comunicação**, para possibilitar o relacionamento dos usuários com outros usuários e com o próprio ambiente virtual;

- ◆ **sistema de processamento**, que recebe e trata eventos oriundos dos dispositivos de interação e da rede de comunicação;
- ◆ **rede de comunicação**, usada para troca de informações entre os usuários e o sistema, proporcionando a interação no sistema.

Na Seção 3.2, são discutidos alguns requisitos funcionais e não-funcionais, bem como características que definem os AVR. Alguns exemplos de aplicações de ambientes virtuais com propósitos educacionais e com propósitos gerais são expostos nas Seções 3.4 e 3.5, respectivamente; e, por fim, na Seção 3.6, são feitas algumas considerações finais.

3.2 Características e Requisitos

Imersão é o principal requisito de uma simulação em ambientes virtuais. Quanto maior o grau de imersão, maior a sensação de realidade percebida pelo usuário. Para isso, é preciso que os usuários tenham uma representação física no ambiente virtual de forma que sejam percebidos por outros usuários dentro de um espaço compartilhado. Ademais, os usuários devem ser livres para se movimentarem dentro desse espaço, encontrando-se e interagindo com outros usuários, além de manipular entidades e informações de interesse comum.

Assim, sempre que um evento provocar alteração no estado do ambiente virtual compartilhado, o mesmo deve ser transmitido automaticamente a todos os usuários inseridos no sistema. Dessa forma, a consistência do ambiente virtual é garantida, mantendo nos participantes a impressão de estarem compartilhando um único ambiente virtual (Çapin et al., 1999).

3.2.1 Requisitos funcionais

Singhal e Zyda (1999) definiram em seus estudos cinco requisitos comuns em sistemas AVR. Tais requisitos equivalem a características percebidas pelos usuários do sistema:

Sensação de espaço compartilhado

É imprescindível que seja criada nos participantes a ilusão de estarem situados no mesmo lugar, mesmo que estejam fisicamente dispersos. Nesses espaços, as interações que os usuários realizam com o mundo virtual e seus objetos, bem como as mudanças resultantes nos seus estados, devem ser percebidas pelos outros participantes.

Logo, um sistema AVR mantém a coerência do espaço compartilhado, apresentando as mesmas características a todos os seus participantes em tempo real. Por exemplo, todos os participantes devem perceber as condições climáticas, assim como a mesma acústica.

Sensação de presença compartilhada

O conceito de presença compartilhada foi introduzido por Slater e seus co-autores (1996) e consiste em dois aspectos: o senso de presença de outros indivíduos no mesmo ambiente (perceber) e o senso de ser parte de um grupo (ser percebido).

Ao entrar no ambiente virtual, cada participante assume o controle de um personagem virtual (avatar), e é através dele que o participante estabelece relações interativas com o próprio ambiente e com outros participantes, proporcionando funcionalidades importantes além daquelas encontradas nos ambientes mono-usuários, como percepção, senso de localização e orientação, identificação de usuários, comunicação gestual, dentre outras (Çapin et al., 1999).

Sensação de tempo

Com o objetivo de proporcionar um alto nível de realismo e interatividade, os usuários do ambiente virtual devem perceber as ações, interações, mútuas ao mesmo tempo em que elas ocorrem, ou seja, em tempo real.

Para isso, o sistema deve informar quaisquer eventos relativos ao ambiente virtual de modo síncrono, isto é, no momento em que estes ocorrem.

Meio de comunicação

Os canais de comunicação entre os usuários são recursos primordiais em um sistema AVR, pois através desses canais os usuários podem estabelecer diálogos a fim de interagir com outros usuários. A maioria dos AVR possibilita a comunicação entre seus participantes através de gestos, texto ou voz. A comunicação adiciona um sentimento necessário de realismo a qualquer ambiente simulado.

Apesar de não ser uma forma usual de comunicação em situações reais, o uso de mensagens de texto é bastante natural para os usuários domésticos, dado que o teclado é um periférico bastante familiar. Por isso, esta é a forma de comunicação encontrada em praticamente todos os AVR semi-imersivos (Almendra, 2003). É possível também utilizar personagens com animações pré-programadas ou mesmo com corpos articulados para representar gestos. Já o canal de comunicação mais natural no mundo real, o canal de

comunicação via voz, apresenta problemas com relação à disponibilidade de recursos de rede, pois os canais de áudio são grandes consumidores de largura de banda.

Um modo de compartilhar

Segundo Singhal e Zyda (1999), o grande crescimento dos sistemas AVR advém da habilidade dos usuários interagirem realisticamente tanto entre si quanto com o próprio ambiente virtual. Na simulação de uma batalha ou de um jogo, por exemplo, os usuários podem atirar uns nos outros, ou podem colidir uns com os outros. Para isso, os usuários devem ser capazes de manipular objetos que existem no ambiente virtual, assim como repassá-los a outros participantes.

Em suma, ambientes virtuais imersivos proporcionam aos seus usuários a habilidade de interagirem uns com os outros, compartilhando informações e manipulando objetos através de imagens gráficas imersivas. A presença de múltiplos usuários independentes diferencia esses sistemas de jogos de realidade virtual mono-usuário. Além disso, a possibilidade de partilhar objetos diferencia esses ambientes das tradicionais salas de bate-papo e a interatividade em tempo real é o diferencial desses ambientes em relação aos navegadores ou correio eletrônico (Singhal & Zyda, 1999). Assim, os ambientes virtuais interativos são mais apropriados para aplicações que demandam a criação de telepresença⁵, onde é criada a ilusão de estar fisicamente em um espaço ou tempo tanto real como imaginário através da transmissão de imagens e sons, criando um espaço de reciprocidade ausente dos meios de comunicação de massas comuns (Goldberg, 1998).

Stytz (1996) também compilou um conjunto de requisitos que, a seu ver, deveriam ser atendidos para a manutenção de um nível aceitável de fidelidade entre o mundo virtual e o mundo real. Cada requisito diz respeito a uma característica do mundo real que deve ser respeitada durante a simulação no ambiente virtual. Os requisitos estão descritos na Tabela 3.1:

Tabela 3.1. Requisitos de fidelidade com o mundo real em ambientes virtuais.

Requisito	Descrição
Modelagem	Os objetos presentes no ambiente virtual devem apresentar proporções corretas entre si, além de serem movimentados em velocidades compatíveis com as do mundo real.

⁵ Segundo Goldberg (1998), telepresença pode ser definida como a “apresentação de uma informação perceptiva que reivindica corresponder a uma realidade física distante”.

Sensorial	Informações relativas aos sentidos humanos devem ser apresentadas aos usuários da forma mais realista possível.
Tempo	Deve haver um atraso mínimo entre a ação de uma determinada entidade virtual (artefatos, objetos, personagens, etc.) e a notificação desta ação aos participantes.
Informação	A quantidade e consistência das informações contidas no ambiente virtual devem ser suficientes para o desenvolvimento de todas as situações previstas e para o suporte à tomada de decisões.
Comportamento das personagens	O comportamento das personagens controladas por computadores deve ser semelhante ao desempenhado pelas personagens controladas por humanos face às mesmas circunstâncias.
Leis físicas	O ambiente virtual deve levar em consideração eventos físicos existentes no mundo real, como gravidade, eletromagnetismo, propagação de sons, etc.
Dispositivos de entrada	Os dispositivos utilizados para interação devem ser condizentes com as simulações previstas para o ambiente virtual. Por exemplo, numa simulação de corrida de carros é desejável que o usuário possa interagir através de um volante, pedais e caixa de marcha, etc.
Sistema	As atividades e respostas dos diversos participantes do ambiente virtual (humanos ou simuladores) devem, quando consideradas como um todo, apresentar um comportamento geral próximo àquele do mundo real.

Em resumo, os requisitos propostos por Stytz (1996) indicam que ambiente deve ser o mais realista possível, induzindo usuários humanos a aceitarem imagens e informações como se fossem reais.

Dependendo dos objetivos da simulação, alguns dos requisitos supracitados podem ser relaxados ou até mesmo desconsiderados. Geralmente, é relaxado um requisito em benefício de um outro.

3.2.2 Requisitos não funcionais

Requisitos não-funcionais relacionados a aspectos de software, hardware e fatores externos são tão importantes para a construção de sistemas de ambientes virtuais quanto para a construção de qualquer software. Os tópicos a seguir relacionam os principais

requisitos não-funcionais ligados ao projeto e desenvolvimento de um sistema AVR (Kirner & Davis, 1996; Sommerville, 1997):

- ◆ **Desempenho** – trata-se do atendimento às exigências de processamento, tempo de resposta, consumo de recursos, eficiência e processos contidos no escopo do sistema;
- ◆ **Usabilidade** – compreende o atendimento de critérios relacionados ao entendimento e facilidade de uso do sistema e nível e satisfação dos usuários em suas interações com o sistema;
- ◆ **Segurança** – trata-se da habilidade do sistema em defender-se ou mesmo de reabilitar-se de ataques acidentais ou intencionais que possam ocorrer por conta de acesso ou uso indevido de recursos contidos no sistema;
- ◆ **Portabilidade** – refere-se à capacidade do sistema ser executado através de diferentes plataformas e/ou sistemas operacionais;
- ◆ **Confiabilidade** – está relacionado à habilidade do sistema em comportar-se de acordo com o que foi especificado na definição prévia do ambiente e com a utilização dos recursos previstos. É necessário garantir que alguns dados críticos, como modelos, comportamentos, controles, etc., serão percebidos pelos usuários sem perda ou dano.

Assim como pode ocorrer com os requisitos de fidelidades apresentados no tópico anterior, é possível que haja um relaxamento em um dos requisitos não-funcionais em prol do fortalecimento de um outro requisito. Por exemplo, é possível que sejam usados métodos de codificação bastante avançados nas mensagens trocadas entre os componentes do sistema, o que resultará na degradação de desempenho do sistema, pois a codificação de mensagens demanda um maior processamento no sistema. No entanto, será aumentado o nível de segurança do sistema.

3.3 Aplicações educacionais

O uso de tecnologias de RV na educação consiste na construção de ambientes tridimensionais que permitem a interação e o envolvimento de alunos com objetos disponíveis no ambiente virtual (Kubo et al., 1999). Por utilizar-se de alguns recursos não disponíveis em outras mídias educacionais, o uso de sistemas AVR para treinamento de usuários pode atingir objetivos de aprendizagem mais facilmente e de um modo mais eficiente e estimulante para o aluno. Assim, os alunos podem controlar ativamente o

ambiente e observar diretamente o comportamento de objetos no interior do mundo virtual, interagindo e colaborando com outros alunos em tempo real. Além disso, a RV não limita o aluno a apenas alguns ângulos de visualização como acontece na observação de fotos ou vídeos. Dentro do ambiente virtual, o aprendiz pode explorar os objetos disponíveis sob diferentes perspectivas, de acordo com sua vontade (Rodello et al., 2002).

Algumas vantagens da utilização de RV para propósitos educacionais foram observadas por Pantelides (1995) e Kalawsky (1996) em seus estudos, as quais seguem:

- ◆ Possibilita a visualização e exploração de lugares inexistentes ou de difícil acesso no mundo real;
- ◆ Promove a motivação do estudante, pois possibilita novas formas de visualização de informações, ilustrando também características, processos, etc.;
- ◆ Estimula estudantes a serem aprendizes mais ativos, pois requer e promove maior interatividade;
- ◆ Não restringe o aprendizado ao período regular de aula, possibilitando a realização de atividades educacionais através da Internet;
- ◆ Permite que o estudante tenha seu próprio ritmo de aprendizado, o que também irá determinar o seu rendimento;
- ◆ Oferece muitas possibilidades para aprendizes não tradicionais, incluindo os alunos com problemas físicos;
- ◆ Provê múltiplas visões de objetos dentro do ambiente (sob diferentes ângulos), fornecendo oportunidade de melhor compreensão do assunto de estudo;
- ◆ Pode ser usada como uma poderosa ferramenta de treinamento e de simulação.

Atualmente, com o crescente desenvolvimento da Internet, a RV também tem auxiliado muito no processo de ensino, sendo aplicada em Educação a Distância (EaD). Entre os principais motivos para a utilização de interfaces de RV em sistemas de EaD, destaca-se o alto grau de interação que tais interfaces proporcionam, ou seja, possibilitam que alunos e professores, mesmo que distantes fisicamente, se inter-relacionem dentro de uma aplicação educacional (Rodello et al., 2002).

A seguir, são apresentados exemplos de sistemas com fins educacionais:

NPSNET

O NPSNET (*Naval Postgraduate School Network Vehicle Simulator*) foi um projeto iniciado em meados da década de 1980 e é um dos sistemas AVR mais antigos encontrados

na literatura. Foi desenvolvido pelo Departamento de Computação da *U.S. Naval Postgraduate School* para o treinamento militar em larga escala com simulação de exercícios de guerra (Macedonia et al., 1994; Macedonia et al., 1995; Macedonia et al., 1995b).

Este projeto é uma continuação dos trabalhos em torno do SIMNET (Calvin et al., 1993). Um dos focos desse sistema é estender o protocolo de comunicação do SIMNET de forma a dar suporte à representação dos mais variados tipos de “jogadores” (soldados, carros, aviões e tanques) presentes nas simulações militares. Além disso, o sistema permite que cada participante controle um veículo militar ou um soldado da infantaria, e dá suporte para que milhares dessas entidades interajam durante a simulação.

Segundo Macedônia e seus co-autores (Macedonia et al., 1995), o desenvolvimento do NPSNET envolve temas de pesquisa em diversas áreas, tais como:

- ◆ O suporte à simulação distribuída em larga escala através da Internet;
- ◆ O treinamento médico com personagens humanos;
- ◆ O desenvolvimento de protocolo de comunicação (DIS) para integração entre diversos simuladores (aviões, veículos, etc.);
- ◆ O uso de hipermídia (áudio e vídeo) distribuída em tempo real dentro de ambientes tridimensionais;
- ◆ O uso de entidades virtuais (entidades ou usuários autônomos) como elementos de povoamento de mundos virtuais; e
- ◆ O uso de som tridimensional de baixo custo.



Figura 3.1. Simulação de batalha aérea no NPSNET-V.

Atualmente, este projeto segue em sua quinta versão (NPSNET-V), onde foi desenvolvida uma plataforma extensível para desenvolvimento do sistema AVR (Capps et

al., 2000). Desse modo, o NPSNET-V implementa a adição dinâmica de novos comportamentos de entidades contidas no ambiente virtual através do carregamento de novas definições recebidas em tempo de execução. Além disso, é provida também uma infra-estrutura de componentes onde são usadas definições de acordo com padrões de codificação XML. Por fim, o sistema possibilita um crescimento escalar por meio de métodos de particionamento de responsabilidades.

AVAL

O Projeto AVAL – Ambientes Virtuais para o Aprendizado de Línguas – (Borges, 2001; Vidal et al., 2001), visou oferecer a guias de turismo capacitação na língua inglesa, especialmente em situações do cotidiano de suas profissões, como visitas a centros de turismo e recepção de turistas em aeroportos. O treinamento dos guias baseou-se em tarefas a serem realizadas dentro dos ambientes virtuais, tais como ajudar turistas a comprar produtos de artesanato ou orientá-los sobre a localização de serviços.

Para a simulação, foram utilizados ambientes virtuais semelhantes aos pontos turísticos reais. O treinamento era feito em grupo, onde os alunos alternavam-se nas condições de guias ou de turistas, tendo que realizar as tarefas específicas de seus papéis dentro do ambiente a cada turno. A Figura 3.2 mostra a parte externa do Centro de Turismo (à direita) e dois usuários interagindo na parte interna do Centro de Turismo (à esquerda).



Figura 3.2. Centro de Turismo Virtual

Laboratório Virtual de Química

O Laboratório Virtual de Química (Rodello et al., 2002) é um projeto direcionado aos alunos do ensino médio onde são usadas técnicas de RV para aprimorar conhecimentos de química. Para isso, o aluno tem a capacidade de explorar um laboratório virtual tridimensionalmente modelado, interagindo com seus objetos. A Figura 3.3 mostra a tela de abertura do Laboratório.

Este sistema foi desenvolvido em HTML (*Hiper Text Markup Language*) e VRML (*Virtual Reality Modeling Language*), o que permite seu fácil acesso via Internet por meio de um navegador *web*. Através dos dispositivos de interação (teclado e mouse), o aluno pode mover-se no ambiente e explorar várias funcionalidades.

O sistema é dividido em módulos, onde cada módulo diz respeito a uma atividade específica. Um dos módulos deste sistema é o “Moléculas”, onde o aluno pode visualizar algumas das moléculas mais comuns com todos os seus átomos e ligações químicas (Figura 3.4).

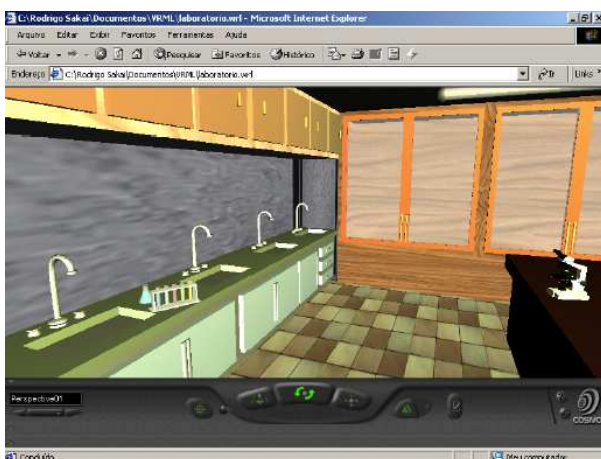


Figura 3.3. Interface inicial do Laboratório de Química

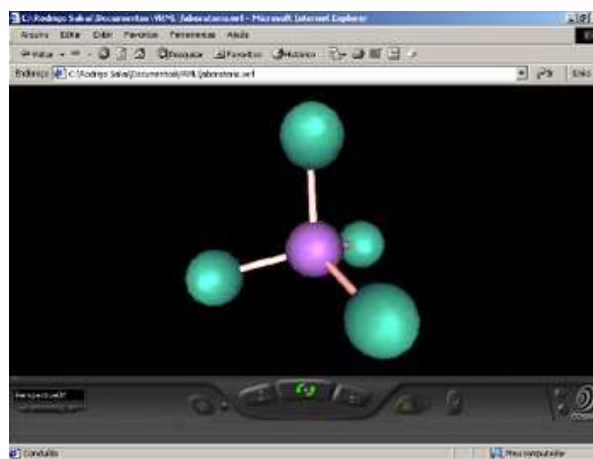


Figura 3.4. Módulo de visualização de Moléculas.

Projeto Museu Virtual

O sistema AVC-MV (Ambiente Virtual Colaborativo – Museu Virtual) foi desenvolvido como parte do Projeto Museu Virtual, por pesquisadores do Centro de Pesquisas de São Carlos, da Fundação Eurípedes de Marília e da Universidade Federal de Santa Catarina (Kawamoto et al., 2001).

Dentre os objetivos do Projeto Museu Virtual (Wazlawick et al., 2001; Kirner et al., 2001), está incluído o desenvolvimento de uma ferramenta de autoria, que possibilite a criação de museus virtuais por estudantes, de forma colaborativa e de acordo com os princípios do construtivismo. Os museus virtuais criados por essa ferramenta de autoria podem ser organizados em salas com diferentes exposições onde os objetos tridimensionais em exibição estão disponíveis para serem diretamente explorados.

Essa ferramenta de autoria é colaborativa e possibilita a criação de museus, nos quais os usuários podem caminhar livremente dentro do espaço tridimensional e explorá-lo segundo as características de navegação, imersão e interação propiciadas pela realidade virtual. Cada usuário (estudante ou visitante) pode escolher o seu avatar, passear pelo

mundo virtual, observar as atividades e ações realizadas e participar da criação ou modificação do museu virtual. Os usuários podem interagir entre si ou com guias virtuais através de texto e também podem visualizar informações relevantes para o contexto do museu apresentadas em forma tridimensional.

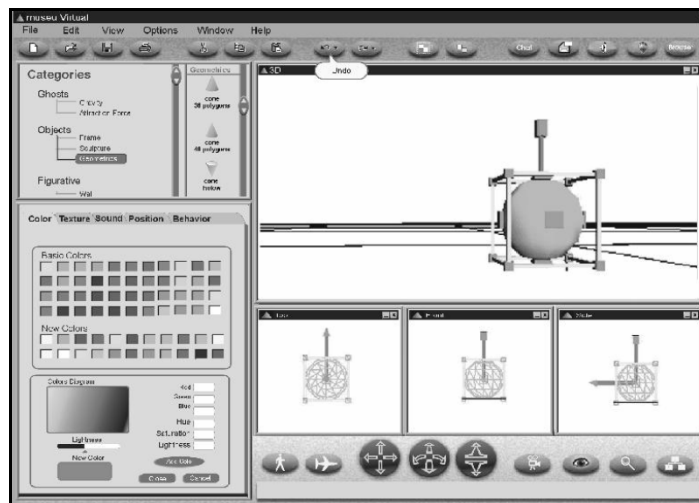


Figura 3.5 Interface da ferramenta de autoria.

É possível também a criação de simulóides⁶ ou guias virtuais, os quais possuem conhecimento sobre os objetos que são exibidos e sobre os visitantes do museu. Os simulóides guiam os usuários nas suas interações com objetos e avatares. Esses guias estão aptos a responder perguntas, apresentar sugestões para os visitantes do museu, realizar ações tais como caminhar de uma sala para outra e mover objetos. Eles também aprendem com os visitantes, isto é, descobrem seus nomes, seus interesses, suas localizações correntes no mundo virtual, etc.

COVET

COVET foi um projeto desenvolvido no *Multimedia Communications Research Laboratory*, Universidade de Ottawa, Canadá, e trata-se de um ambiente virtual colaborativo para o treinamento industrial (Hosseini & Georganas, 2001).

É uma aplicação de tele-treinamento multi-usuário onde os usuários, representados no ambiente por seus respectivos avatares, podem treinar procedimentos de reparo de um *switch ATM*.

Este ambiente virtual foi desenvolvido com o intuito de diminuir os custos dos treinamentos nas indústrias, principalmente em casos onde operários e clientes de algumas indústrias precisam se familiarizar com equipamentos e aparelhos de difícil manutenção.

⁶ Entidade virtual controlada por computador.

Os usuários podem interagir com o ambiente tridimensional clicando em vários objetos, visualizando funções relacionadas ao cenário do treinamento. As ações dos usuários são comunicadas a outros participantes de modo que eles tenham a impressão de estarem envolvidos no treinamento.

O sistema virtual dispõe de vários ângulos de visão. Uma simples tarefa onde um usuário ensina outros como efetuar a retirada segura de uma placa defeituosa de um *switch* ATM é demonstrada na Figura 3.6.

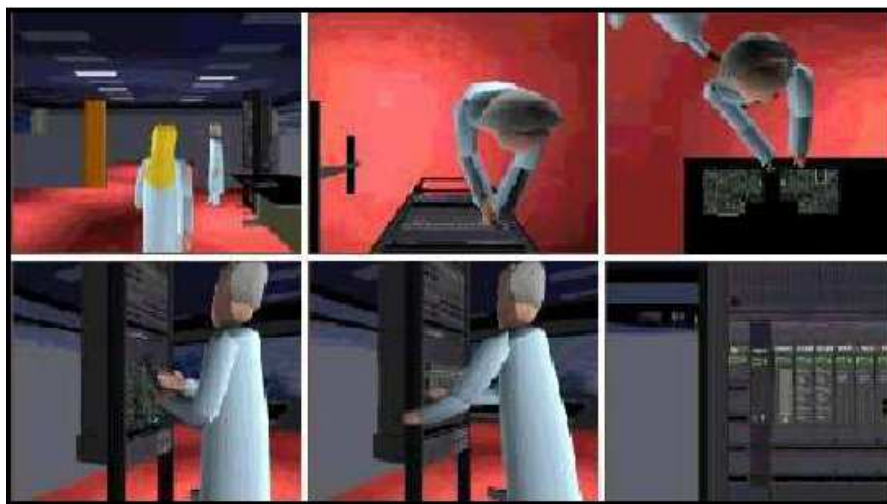


Figura 3.6. Alguns ângulos de visão de um treinamento.

3.5 Outras Aplicações

Por oferecerem um sistema natural de compartilhamento – baseado em modelos de interação da própria natureza humana –, os AVR podem ser aplicados aos mais diversos fins. Além de aplicações voltadas para fins educacionais, há outras áreas onde são encontradas várias aplicações, por exemplo (Çapin et al., 1999):

- ◆ telemedicina (experimentos a distância);
- ◆ terapia de fobias (Rothbaum et al., 1995);
- ◆ jogos multiusuários;
- ◆ escoamento de fluidos (Bryson & Levit, 1992);
- ◆ entretenimento (Pausch et al., 1996);
- ◆ teleconferência;
- ◆ telecompras;
- ◆ turismo virtual.

A seguir, são apresentados em detalhes alguns exemplos de outras aplicações de sistemas AVR:

AlphaWorld

Ao contrário de outras comunidades virtuais, os usuários do AlphaWorld (Dodge, 1999; Ryan, 2004) podem, além de comunicar-se com outros membros da comunidade, estabelecer residências, decorar jardins e realizar diversas outras atividades relacionadas ao convívio em comunidade (Figura 3.7). No ambiente virtual do AlphaWorld, os participantes são motivados a partilharem da urbanização da cidade virtual. No entanto, a urbanização nessa cidade difere substancialmente das cidades do mundo real, onde há restrições populacionais, de custos ou mesmo de transportes. Todas as modificações feitas pelos usuários são acumulativas, de modo que os usuários que entram no ambiente virtual percebem todas as mudanças feitas pelos usuários que ali entraram anteriormente.

O AlphaWorlds faz parte de uma das maiores comunidades virtuais presentes na *web*, a comunidade virtual ActiveWorlds⁷, que possui também vários outros ambientes virtuais disponíveis para edição e navegação.

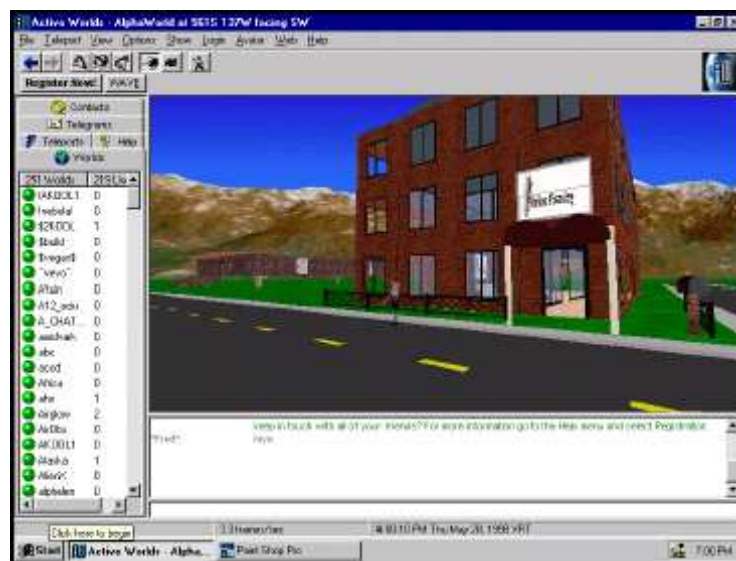


Figura 3.7. Visão do ambiente virtual AlphaWorlds.

Blaxxun

O Blaxxun Community Platform (Blaxxun, 2005) é um sistema modular que fornece um amplo conjunto de funções para desenvolvimento de comunidades virtuais para *web* (Figura 3.8).

É usado para a criação on-line de comunidade virtuais onde os participantes podem se conhecer, conversar e trocar experiências. Além dos serviços de gerência de usuários e

⁷ <http://www.activeworlds.com>

mundos virtuais e comunicação entre os participantes, o sistema disponibiliza também uma variedade de módulos que oferecem interfaces para banco de dados externos e para outros sistemas, como por exemplo, os de comércio eletrônico.

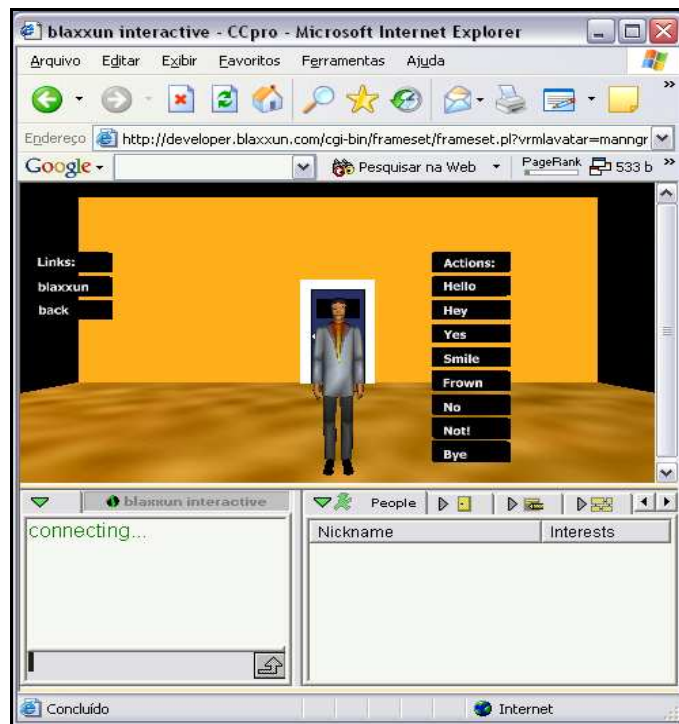


Figura 3.8. Interface do ambiente Blaxxun.

Papo3D

O Papo3D (Leite Junior et al., 2002) é um ambiente virtual multi-usuário voltado para o entretenimento que pode ser usado através da Internet. O projeto desse ambiente visou tornar disponível, através do uso de microcomputadores comuns e de redes de baixa velocidade, ambientes virtuais interativos voltados para o entretenimento e que podem ser facilmente estendidos para outras áreas.



Figura 3.9. Mundos virtuais disponíveis no Papo3D.

Nesse sistema, o usuário pode interagir com outros usuários em tempo real, bem como com entidades contidas no mundo virtual. Há vários ambientes e avatares diferentes que podem ser escolhidos pelos usuários (Figura 3.9).

3.6 Considerações finais

Os sistemas distribuídos de RV estão se tornando cada vez mais utilizados e têm comprovado o potencial deste tipo de aplicação. Um ambiente virtual distribuído é um sistema que possibilita que usuários interajam entre si através do mesmo ambiente virtual, mesmo que estejam geograficamente dispersos.

As primeiras aplicações de AVR foram desenvolvidas nos Estados Unidos para fins de treinamento militar. Contudo, atualmente, as aplicações distribuídas de RV podem ser utilizadas com foco em outras áreas, tais como educação, visualização científica, jogos, turismo virtual, entretenimento, etc.

Apesar da crescente popularidade, os sistemas AVR ainda apresentam uma série de desafios em seu desenvolvimento tais como os relacionados aos requisitos não-funcionais listados neste capítulo, onde pode-se destacar o requisito de desempenho (vide 3.2.2). Estes requisitos deverão ser retomados no próximo capítulo, quando serão discutidos alguns aspectos de desenvolvimento de sistemas AVR. Posteriormente, serão apresentadas estratégias para atender a estes requisitos.

Capítulo 4

Aspectos de desenvolvimento de Ambientes Virtuais em Redes

4.1 Visão geral

Ao mesmo tempo em que os ambientes virtuais podem ser aplicados nas mais variadas áreas (educação, entretenimento, visualização científica, etc.), também a sua forma de desenvolvimento pode variar bastante. Logo, dependendo dos objetivos da simulação, pode-se exigir uma implementação mais fiel ou relaxada de alguns dos requisitos funcionais ou não-funcionais descritos anteriormente.

Geralmente, os sistemas AVR se distinguem pelas suas capacidades de agregação de novos recursos, como usuários e entidades virtuais, onde é possível encontrar desde ambientes virtuais que suportam apenas um único usuário até ambientes que, potencialmente, suportariam um número infinito de usuários e entidades interagindo entre si.

Um dos aspectos fundamentais de um sistema AVR é a rede de comunicação, pois é através desta rede que serão trocadas informações entre os usuários do sistema e o próprio ambiente virtual. Na medida em que o sistema vai crescendo, em número de participantes e de entidades virtuais, mais recursos de redes são necessários. Portanto, o uso dessas redes traz uma série de implicações para o desenvolvimento e crescimento do sistema como um todo.

Conforme mencionado anteriormente, o desenvolvimento de sistemas AVR é uma tarefa complexa e exige a consideração de uma série de requisitos de *software* e *hardware*, sendo alguns dos quais essenciais. Neste capítulo, é feito um levantamento dos principais aspectos relativos ao desenvolvimento e uso otimizado do sistema AVR a partir da análise de trabalhos encontrados na literatura. Os vários aspectos pesquisados são concernentes ao atendimento dos requisitos funcionais e não-funcionais descritos no capítulo anterior. Com base nos tópicos apresentados neste capítulo será justificado o Modelo NuVEM proposto no presente trabalho.

Na Seção 4.2, serão discutidos alguns aspectos relativos à escalabilidade dos sistemas AVR; na Seção 4.3 são apresentadas algumas considerações sobre a utilização de redes de comunicação, onde foram observados métodos de transmissão, modelos de

distribuição, protocolos e peculiaridades da própria rede de comunicação. Ainda na Seção 4.3 são previstos impactos da utilização da rede de comunicação na qualidade de serviço provida aos usuários do ambiente virtual. Em seguida, na Seção 4.4, são descritas algumas técnicas compensatórias criadas com o objetivo de amenizar os impactos da utilização das redes de comunicação; na Seção 2.5, são considerados a aplicação de cada uma das áreas funcionais de gerência de redes (descritas em detalhes no Capítulo 2) em ambientes virtuais. Por fim, na Seção 4.6, tecem-se algumas considerações finais.

4.2 Escalabilidade

A escalabilidade é um dos requisitos mais importantes para o desenvolvimento de um sistema AVR. Isto se deve ao fato de que ele possivelmente contará com recursos limitados, como quantidade de memória de cada máquina, poder de processamento numérico/gráfico e velocidade de transmissão da rede.

Em geral, a escalabilidade de um sistema pode ser medida por sua capacidade de aceitação de novas entidades participantes – como usuários, simulóides e objetos interativos – e depende de fatores tais como: a capacidade da rede de dados, o desempenho e o poder de processamento e renderização⁸ das máquinas dos usuários.

De acordo com Singhal e Zyda (1999), a complexidade do sistema aumenta exponencialmente com o número das entidades participantes, em virtude do número de possíveis interações entre essas entidades, ou seja, $2^{(\text{número de entidades})}$ possíveis interações. Entretanto, na prática, uma entidade não interage com todas as outras entidades, o que reduz significativamente essa complexidade. Outro fator que deve ser considerado na complexidade dos AVR é o comportamento das entidades. Entidades que podem apenas ser observadas pelo usuário implicam em um nível de complexidade menor do que aquelas com os quais o usuário pode interagir, modificando-as, ajustando-as, etc. (Rodello et al., 2001).

A escalabilidade do sistema também pode ser medida com base na quantidade de ambientes compartilhados distintos ativos simultaneamente. A partir dessa segunda forma de medir escalabilidade, é possível analisar a primeira sob dois aspectos: a capacidade de aceitação de entidades em um único ambiente compartilhado do sistema e a capacidade total de aceitação de entidades no sistema.

⁸ Geração e processamento de imagens, possivelmente tridimensionais, do ambiente virtual.

Limitações na escalabilidade são provocadas por uma variedade de pontos críticos (gargalos). Quando o número de entidades ativas é elevado, ocorre um grande volume de tráfego na rede, especialmente pela movimentação de pacotes de atualizações (estados das entidades virtuais) e áudio. Assim, o computador local de um usuário deve ser capaz de processar as informações que chegam a ele e renderizar o mundo virtual compartilhado com qualidade satisfatória enquanto mantém uma resposta suficientemente rápida aos movimentos e ações dos outros participantes. A seguir, são listadas algumas situações que podem originar limitações na escalabilidade de um sistema AVR, sejam elas de processamento ou de utilização da rede (Ferreira, 1999):

- ◆ A presença de muitas entidades no ambiente virtual pode implicar em uma elevada utilização de tempo de processamento dedicado à atualização e à predição dos comportamentos;
- ◆ O uso de geometrias altamente detalhadas também pode implicar em uma elevada utilização de tempo de processamento gráfico necessário para renderizar uma entidade dentro de uma cena;
- ◆ O uso excessivo de texturas pode comprometer o desempenho quando a memória destinada a elas for insuficiente, acarretando acessos indesejados à memória secundária;
- ◆ A existência de entidades com comportamentos não-determinísticos ou pouco comportados pode implicar na necessidade de atualizações excessivas, gerando tráfego excessivo na rede;
- ◆ O compartilhamento da rede com outras máquinas que não estejam participando da simulação pode degradar o sistema;
- ◆ A presença de roteadores e *switches* na interligação entre as redes conectadas às máquinas pode exigir a utilização de algoritmos mais complexos para o sincronismo e implicar em maiores retardos na rede.

Segundo Macedonia (1995), não bastam computadores e sistemas de comunicação mais rápidos para resolver problemas de escalabilidade de um sistema. É necessário que sejam previstos vários outros fatores durante o desenvolvimento, como algoritmos de distribuição de serviços, geração de gráficos tridimensionais e colisão de entidades virtuais, serialização de tarefas, protocolos de comunicação, mecanismos de transmissão, dentre outros.

Nas próximas sessões, serão vistos alguns destes fatores que influenciam a escalabilidade dos sistemas AVR, bem como possíveis soluções para uma utilização mais otimizada dos recursos computacionais do sistema.

4.3 Comunicação em redes

4.3.1 Modelos de comunicação

De acordo com Gossweiler e seus co-autores (1994), existem dois modelos de comunicação que podem ser implementados por um sistema de AVR: o modelo centralizado e o modelo distribuído.

No modelo centralizado, um computador principal recebe todos os dados provenientes dos usuários conectados ao mundo virtual, aplica essas informações ao ambiente virtual armazenado no computador central e devolve, para cada um dos usuários conectados o resultado das alterações feitas sobre o mundo virtual, atualizando-os (Figura 4.1).

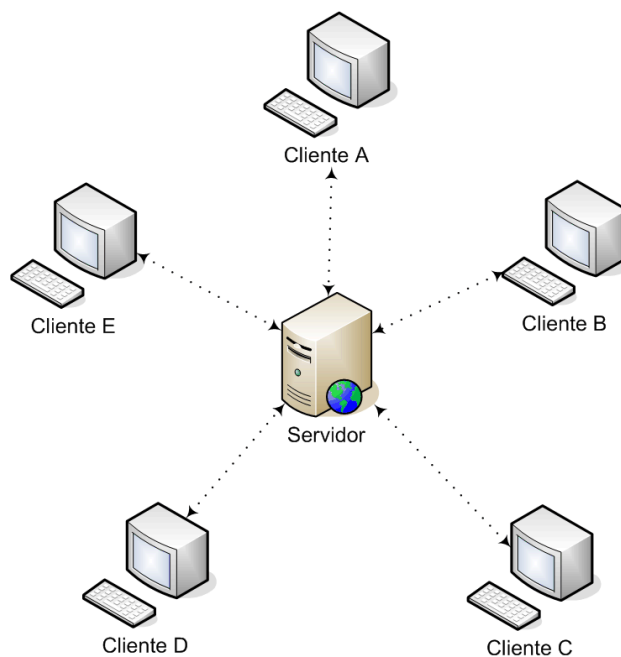


Figura 4.1. Modelo centralizado

Apesar de apresentar facilidades com relação à implementação dos mecanismos de comunicação e controle de usuários e recursos computacionais dispostos no sistema, este modelo traz sérias desvantagens com relação à escalabilidade, pois na medida em que os números de usuários e de recursos crescem, maior o tráfego de mensagens com o computador central. Além disso, há também um demasiado processamento no computador central, o que poderá prejudicar o desempenho do sistema como um todo, fazendo com que

a velocidade de acesso caía devido ao excesso de mensagens as quais o computador central precisa receber, processar e enviar.

Como forma de resolver essa limitação de escalabilidade do modelo de comunicação centralizado, foi desenvolvido o modelo distribuído. Nesse modelo, cada usuário do sistema mantém em seu *host* uma cópia própria completa do ambiente virtual, realizando, além das tarefas de renderização, computação e animação dos objetos do ambiente virtual, a distribuição de mensagens para os outros usuários conectados ao sistema. Dessa forma, quando um usuário realiza alguma modificação sobre o ambiente virtual, ele se encarrega de comunicar a todos os outros usuários sobre as alterações realizadas (Figura 4.2).

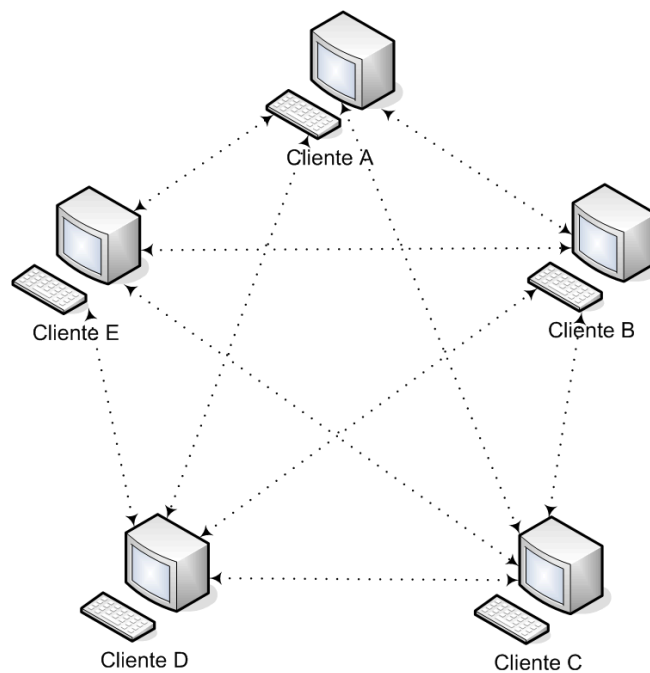


Figura 4.2. Modelo distribuído.

No entanto, nessa solução o número de conexões cresce muito rapidamente, pois cada usuário precisa estabelecer um canal de comunicação com cada um dos demais usuários. Além disso, percebe-se que é gerado um número maior de mensagens trocadas no sistema.

Com o intuito de reduzir o número de conexões e mensagens do modelo distribuído foram desenvolvidos métodos de transmissão de mensagens, como o *broadcast* e o *multicast*, os quais serão vistos a seguir.

4.3.2 Métodos de transmissão

De acordo com Stytz (1996) e FunkHouse (1996), o controle da comunicação em redes corresponde à criação de canais de comunicação e ao modo como as mensagens

serão transmitidas pela rede. A seguir, serão descritos alguns dos métodos de transmissão mais utilizados em sistemas AVR. Como já é de se esperar, cada um dos métodos traz vantagens e desvantagens, as quais também serão descritas.

Unicast

Neste método, quando um usuário do ambiente virtual faz alguma modificação, ele precisa mandar uma mensagem correspondente a esta modificação para cada um dos outros usuários do ambiente virtual. Pelo fato de ser necessário que cada usuário estabeleça uma conexão com os demais usuários, a sincronia das atualizações entre os usuários será alta. No entanto, um número elevado de usuários demandaria um número muito alto de conexões, tornando difícil o gerenciamento das mesmas. Além disso, é gerado um número elevado de mensagens, pois cada atualização no ambiente virtual resulta no envio de cópias de uma mesma mensagem para cada um dos outros usuários (Figura 4.3).

Este método de transmissão é usado na implementação do modelo de comunicação distribuída simples visto no tópico anterior.

Broadcast

Foi desenvolvido com o objetivo de solucionar os problemas decorrentes do uso do *Unicast*. No caso do *Broadcast*, não é necessário que haja uma conexão entre cada um dos usuários do ambiente virtual. Basta que todos os usuários comuniquem-se por meio de um canal comum. Com isso, o problema do alto número de mensagens geradas é resolvido, pois a cada atualização no ambiente virtual o usuário envia somente uma mensagem para a rede, que será recebida por todos os demais usuários (Figura 4.4).

Nesse método de comunicação geralmente é utilizado *UDP Broadcast*, onde as mensagens são enviadas para um endereço de *broadcast* da rede local através do protocolo UDP (Gossweiler et al., 1994; Singhal & Zyda, 1999), o que torna bastante útil o seu uso principalmente em redes locais (LAN). No entanto, por enviar a mensagem original a todos os endereços da rede, o *UDP Broadcast* pode gerar uma quantidade alta de mensagens desnecessárias, o que torna inviável sua utilização em redes maiores. Logo, para AVR sobre grandes redes (WAN) a utilização de tal método não seria possível. Nesse caso, tem-se como opção o *Multicast*, que é visto a seguir.

Multicast

Neste método, um conceito de canal de comunicação é adotado. Cada usuário do ambiente virtual recebe mensagens de um ou mais canais do ambiente virtual, onde cada canal corresponde a um grupo *multicast* representado por dois ou mais usuários. Dessa forma, quando um usuário envia uma mensagem, ele a envia para um canal específico do qual faz parte e somente os usuários que estiverem participando do mesmo canal (mesmo grupo) receberão a mensagem (Figura 4.5).

Conforme Brutzmann e seus co-autores (1995), no *IP Multicast* a transmissão de datagramas IP pode ser feita para um número ilimitado de *hosts* conectados à Internet através de roteadores compatíveis com *Multicast*.

Os usuários de um sistema AVR podem ser distribuídos através de vários canais com a utilização de um endereço IP classe D correspondente aos endereços de 224.0.0.0 até 239.255.255.255. Dessa forma, um *host* pode escolher entrar ou sair de um grupo *Multicast* apenas informando ao roteador (ao qual este *host* se conecta à Internet) em quais grupos *Multicast* ele tem interesse. A partir daí, esse roteador somente repassaria as mensagens dos grupos aos quais há algum *host* cadastrado (Comer, 1995; Singhal & Zyda, 1999).

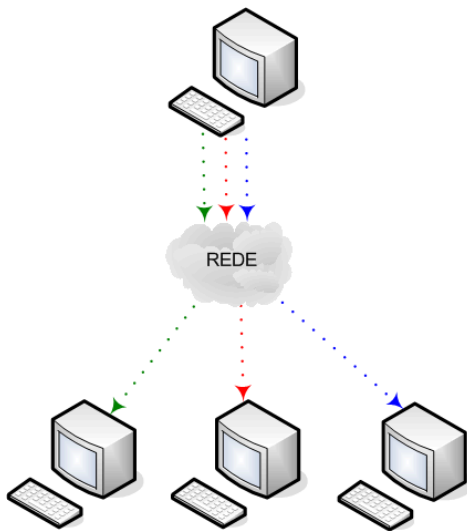


Figura 4.3. Unicast (três mensagens enviadas, uma para cada usuário)

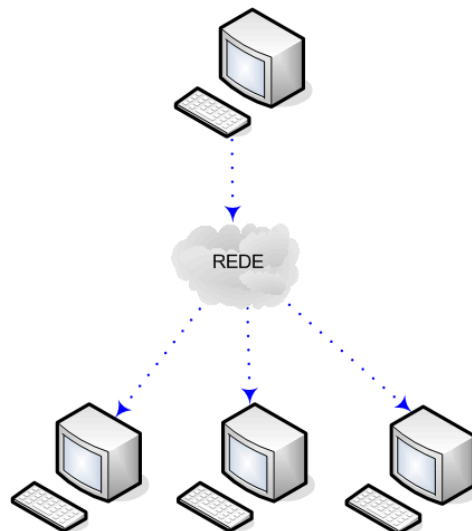


Figura 4.4. Broadcast (apenas uma mensagem enviada, todos recebem)

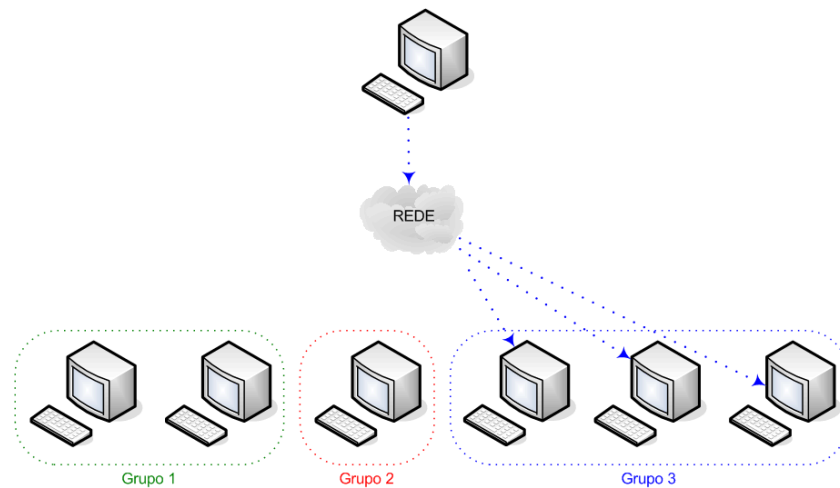


Figura 4.5. Multicast (uma mensagem enviada somente para o Grupo 3)

Um caminho natural de ambientes virtuais distribuídos é, portanto, buscar modos de se livrar de atualizações desnecessárias. A extensão natural para sistemas utilizando *Broadcast* é tentar remediar o excesso de mensagens por meio de comunicação *Multicast*.

A idéia de se utilizar grupos de IP *Multicast* torna-se ainda mais interessante caso sejam usadas técnicas compensatórias para se dividir o ambiente virtual, distribuindo seu processamento. Estas técnicas serão vistas mais à frente com detalhes.

4.3.3 Características da rede de comunicação

É importante perceber que, independente do modelo de comunicação ou método de transmissão utilizados, é preciso levar em consideração também fatores inerentes à própria rede de dados. Infelizmente, esses fatores geralmente representam limitações físicas das quais o projetista do sistema AVR deve ter conhecimento prévio de modo a evitar situações inesperadas em tempo de execução.

Largura de banda

A largura de banda de uma rede pode ser definida como a capacidade máxima de se transferir uma quantidade determinada de informações num determinado período, geralmente expressa em bits por segundo – bps – (Szwarcman, et al., 1999).

Num sistema AVR é a largura de banda da rede utilizada para interconectar os usuários que determinará o tamanho e a riqueza de detalhes do ambiente virtual. À medida que cresce o número de entidades virtuais, aumenta a necessidade de transferência dessas informações entre os diversos componentes do sistema. A largura de banda disponível para o sistema é, portanto, fator determinante no projeto da arquitetura de comunicação e das

funcionalidades que estarão disponíveis no sistema. Portanto, o número de entidades é que irá exercer maior influência com relação à utilização da largura de banda.

De acordo com Macedônia e Zyda (1997), em redes locais (LANs), que geralmente dispõem de uma banda de 100 Mbps, o número de usuários é relativamente limitado e a largura de banda disponível é geralmente suficiente. No entanto, a partir do momento em que é estendido o uso do sistema sobre uma rede de longa distância (WAN) para interconectar os usuários do ambiente virtual, geralmente larguras de banda bem mais modestas estão disponíveis. Levando-se em consideração que numa WAN o número de usuários interconectados pode ser bem maior, nem sempre essa largura de banda é suficiente para garantir um desempenho satisfatório do ambiente virtual. Num modelo de comunicação distribuído não é difícil perceber como o aumento do número de usuários aumenta consideravelmente o uso da largura de banda.

Assim, largura de banda e escalabilidade são fatores intimamente relacionados, pois quanto maior for a escalabilidade do sistema, maior deverá ser a largura de banda disponível. Logo, a adoção de soluções para minimizar a carga de dados a ser transmitida na rede sem que os usuários percam detalhes importantes da simulação é um requisito fundamental para o desenvolvimento de um sistema escalável.

Latência

Em sistemas de comunicação, a latência é definida pelo tempo decorrido entre o envio de um pacote de dados e o recebimento do mesmo no endereço de destino.

Se o objetivo do ambiente virtual é simular um mundo real, este precisa operar em tempo real, considerando-se a percepção humana. Conforme Macedônia e Zyda (1997), é o controle da latência que determinará se a interação e a dinâmica do ambiente virtual são satisfatórias para os usuários do ambiente virtual. É preciso garantir que a entrega de pacotes tenha uma latência mínima para permitir que as imagens do ambiente virtual sejam geradas, para cada um dos usuários, numa frequência suficiente para garantir a ilusão de realidade. Os valores ideais dessa frequência variam numa faixa de 30 a 60 quadros por segundo (30-60 Hz).

De modo semelhante à largura de banda, a latência não apresenta maiores problemas em redes locais (LANs). No entanto, em redes maiores (WANs), o longo caminho que um pacote precisa percorrer até chegar ao seu destino, passando por vários equipamentos que interconectam redes, contribui significativamente para o aumento da latência.

Além disso, é importante perceber que não só a latência pode comprometer a ilusão de realidade de um ambiente virtual. Para garantir uma frequência satisfatória de quadros por segundo, possibilitando uma boa ilusão de realidade, itens como a interface de rede, memória e capacidade de processamento gráfico de cada um dos computadores conectados ao ambiente virtual também são relevantes.

Confiabilidade

Em sistemas de comunicação, a confiabilidade está relacionada à garantia de que uma mensagem enviada por um usuário chegará de modo integral no destino desejado.

Assim, o conceito de confiabilidade aplicado a AVR consiste na garantia de que as alterações feitas no ambiente virtual por algum usuário sejam transmitidas aos outros usuários do sistema de modo que todos mantenham a ilusão do espaço compartilhado.

Segundo Macedonia & Zyda (1997), confiabilidade em comunicações de AVR pode ser alcançada com o uso de protocolos de comunicação orientados a conexão, que utilizam mecanismos de confirmação de recebimento e recuperação quando da ocorrência de erros ou não recebimento de algum pacote. No entanto, o uso desse tipo de protocolo pode resultar em uma maior latência. Logo, o uso ou não destes protocolos depende diretamente do nível de confiabilidade exigido para uma simulação e dos níveis de latência aceitáveis.

A seguir, serão descritos alguns aspectos relativos ao projeto e uso de protocolos em sistemas AVR.

4.4 Técnicas compensatórias

São métodos para diminuir os efeitos negativos produzidos pelas limitações da rede de comunicação citadas anteriormente, como latência, largura de banda e confiabilidade, reduzindo de forma coerente o tráfego de informações entre os pontos do sistema.

Os processadores têm evoluído mais rapidamente que as redes de comunicação. Logo, algumas das técnicas descritas nesta sessão consistem em poupar largura de banda em troca de um processamento computacional extra (Smed et al., 2001; Singhal & Zyda, 1999).

4.4.1 Compactação de mensagens

Por meio da compactação de mensagens é possível economizar parte da largura de banda disponível em um sistema de comunicação. No entanto, o processo de compactação

e descompactação demanda um custo computacional adicional no momento de envio e recebimento de mensagens.

A escolha do algoritmo de compactação (ou combinação de algoritmos) depende da arquitetura do sistema. Vários fatores devem ser analisados no momento de escolha deste algoritmo, como a frequência com que os clientes enviam e recebem mensagens, os tipos de mensagens e os métodos de transmissão implementados pelo sistema. O algoritmo de compactação pode ser implementado diretamente ao protocolo usado pelo sistema.

4.4.2 Agregação de mensagens

Consiste em reduzir a frequência de transmissão de mensagens por meio da junção de várias mensagens em uma só. Assim, em vez de enviar várias mensagens pequenas é enviada apenas uma mensagem maior por vez. Assim como a técnica de compressão, esta técnica demanda um processamento adicional durante o envio e o recebimento de mensagens. Segundo Singhal e Zyda (1999), esta técnica pode reduzir em até 50% o uso de largura de banda em um sistema distribuído.

Infelizmente, por ter que acumular um número mínimo de mensagens para transmitir, esta técnica pode artificialmente aumentar a latência da rede, forçando os destinatários a esperarem mais tempo pelas atualizações do ambiente virtual. Para resolver esse problema, foram desenvolvidas políticas de atualização baseadas em *timestamps*, onde há um tempo máximo para descarregar as mensagens para a rede.

4.4.3 Gerenciamento de interesses

São técnicas que utilizam mecanismos de filtragem de mensagens baseadas em Expressões de Interesses (IE – *Interest Expressions*) para enviar aos processos contidos na simulação do ambiente virtual apenas as informações as quais lhes são pertinentes (baseado em atributos relativos ao ambiente virtual). Os dados referentes a cada processo constituem o Domínio de Interesses (DoI – *Domain of Interest*) de cada processo. Entidades especiais conhecidas como Gerentes de Interesses (IM - *Interest Managers*) são responsáveis por filtrar os dados gerados e encaminhá-los aos demais processos com base em suas IE (Morse, 1996).

Dentre os mais conhecidos esquemas de Gerenciamento de Interesses está o gerenciamento baseado em IP *Multicasting*, onde as mensagens são enviadas apenas para um subgrupo de participantes de ambiente virtual através de estratégias de *Multicast* (Zyda, 1995). No caso da divisão do ambiente virtual, pode-se atribuir um endereço IP

Multicast para cada uma das áreas divididas. Nesse caso, ao entrar em uma determinada área, um usuário passaria a fazer parte de um grupo *Multicast* específico, enviando e recebendo somente as mensagens referentes àquele grupo.

Logo, por meio de técnicas de gerenciamento de interesses pode-se especificar quais informações serão recebidas por cada participante do ambiente virtual. Com isso, é possível reduzir o número de mensagens na rede ao custo de um processamento computacional usado na filtragem das mensagens.

4.4.4 Predição de movimento

O uso de técnicas de *dead-reckoning*⁹ promove uma grande redução no tráfego de mensagens necessárias para a correta atualização do posicionamento das entidades contidas nos ambientes virtuais (Gutmann et al., 1998). Consistem na previsão da trajetória da entidade virtual com base na sua posição inicial e nos vetores velocidade e aceleração.

Na prática, algoritmos de predição de movimento funcionam da seguinte maneira: quando um usuário se movimenta no ambiente virtual, ao invés de serem enviadas várias mensagens informando as posições que o usuário vai ocupando, apenas, por exemplo, a direção, a velocidade e a aceleração do mesmo são informadas. Estando de posse dessa informação, cada computador conectado ao ambiente virtual é capaz de calcular a nova posição que o usuário vai ocupar.

4.4.5 Partição de ambientes virtuais

A partição de um AVR consiste na divisão de todo o ambiente compartilhado em determinadas áreas, que podem ser processadas independentemente em paralelo. Os participantes que se encontrarem nas mesmas partes do mundo virtual podem interagir entre si, sendo, dessa forma, necessária a troca de mensagens somente entre um subconjunto de participantes remotos. Esse tipo de distribuição auxilia tanto na diminuição da utilização da rede de comunicação quanto na carga computacional necessária ao processamento de todo o AVR, incluindo até mesmo o processo de geração gráfica e as simulações envolvidas (Macedonia et al., 1995; Singhal & Zyda, 1999).

⁹ O termo *dead-reckoning* (abreviação de *deduced reckoning*) tem suas origens em uma técnica de navegação utilizada até mesmo por Colombo, que consiste em marcar a posição corrente da embarcação com base nas medidas de direção e tempo gasto no último trecho conhecido percorrido. Ainda hoje, esta técnica pode ser utilizada em navegação, e existem programas feitos para isto, caso haja problemas com equipamentos de GPS.

Logo, por meio desta técnica é possível dividir o processamento de um ambiente virtual em diversos pontos da rede. Conforme os trabalhos de Çapın e seus co-autores (1999), as partições devem ser ligadas por meio de portais e a transição entre as mesmas deve ser transparente ao usuário do ambiente virtual.

4.4.6 Níveis de detalhes

Em seu trabalho, Clark (1976) discute a necessidade da adequação do nível de detalhe (LoD – *Level of Detail*) em relação à distância entre o objeto sendo desenhado e o observador. Isso decorre do fato de que, por exemplo, desenhar uma entidade dotada de 10.000 polígonos a uma grande distância do observador não acrescenta nenhum ganho efetivo de qualidade visual. Uma versão simplificada da entidade, contendo cerca de 500 polígonos, poderia ser desenhada caso o resultado final ocupasse apenas uma pequena porção da tela, alcançando o mesmo efeito visual.

Assim, é possível processar de modo diferenciado as várias entidades contidas no ambiente virtual, economizando poder computacional e simplificando o processo de geração de gráficos tridimensionais.

4.5 Gerência de falhas em AVR

Apesar do escopo deste trabalho se restringir à aplicação de uma área específica de Gerenciamento de Redes em AVR, foi necessária a compreensão acerca de todas as outras áreas para que o modelo especificado no capítulo 5 fosse viabilizado. Para isso, são descritas a seguir algumas considerações relativas à implantação de estratégias para Gerenciamento de Redes em sistemas AVR em geral e, em seguida, são definidos alguns aspectos relativos ao gerenciamento de falhas em sistemas distribuídos.

Infelizmente, na literatura não são encontrados muitos exemplos de aplicações diretas de estratégias de gerência de redes em ambientes virtuais. No entanto, para algumas áreas, como gerenciamento de desempenho ou segurança, podem ser encontrados trabalhos correlatos, principalmente no desenvolvimento de sistemas de realidade virtual voltados para o entretenimento (jogos multi-usuário).

4.5.1 Gerência em sistemas AVR

O gerenciamento de entidades dos ambientes virtuais está diretamente ligado ao gerenciamento de desempenho do sistema como um todo. As interações em tempo real definem um conjunto de processos (*threads*) que compõe a aplicação de realidade virtual.

Cada processo pode ser representado por uma entidade contida no ambiente virtual (usuários, objetos, simulóides, etc). Assim, dentro do escopo da aplicação interativa, vários processos concorrem por processamento (Singhal & Zyda, 1999).

Logo, o gerenciamento do desempenho do sistema consiste no escalonamento da execução de cada um dos processos contidos na simulação em tempo real. O projetista do sistema deve aplicar técnicas para gerenciamento provendo a execução de cada um dos processos que demandam tempo de processamento. Em (Lee et. al, 2002; Rodrigues et. al, 2004) são descritas arquitetura escaláveis de AVR por meio do gerenciamento de interesses e particionamento de ambientes. Além disso, há também trabalhos envolvendo balanceamento de carga processada no sistema com o objetivo de melhorar o desempenho (Carlsson & Hangsand, 1993; Capps et al., 2000).

Já o gerenciamento de configuração consiste na habilidade de configurar os ambientes virtuais em tempo de execução, adequando seus requisitos de acordo com a capacidade física (*hardware*) dos componentes do sistema. Dentre os principais parâmetros que devem ser configurados estão as características das entidades que denotam o crescimento do sistema, como usuários, objetos dinâmicos, etc., contidos nos ambientes virtuais (Singhal & Zyda, 1999). Para cada recurso a ser gerenciado, deve-se armazenar suas peculiaridades de acordo com os fatores de escalabilidade do sistema.

Além destas características, é preciso armazenar informações acerca de todos os componentes físicos do sistema, como capacidade de processamento e memória dos servidores, capacidade de transmissão de dados, etc. Estes atributos serão necessários durante a avaliação de recursos físicos (*hardware*) livres no sistema.

Nos últimos anos, têm sido cada vez mais comuns os ataques a servidores nas grandes redes, o que tem feito com que o gerenciamento de segurança tenha um papel fundamental durante o desenvolvimento do sistema AVR (Cecin et. al, 2003). Assim, são tópicos importantes para o gerenciamento de segurança (Smed et al., 2001): proteger informações sensíveis (como números de cartões de crédito) e assegurar os requisitos de consistência do mundo virtual, evitando trapaças e situações inesperadas pelos usuários. Em (Cecin et. al, 2003), é prevista um modelo centralizado para suporte de AVR massivos voltados para o entretenimento e imunes às trapaças dos usuários.

4.5.2 Tolerância a falhas em sistemas distribuídos

Tolerância a falhas é a habilidade de um sistema de funcionar corretamente ainda que ocorra uma falha em algum de seus componentes (Jalote, 1994). O uso de técnicas de tolerância a falhas visa principalmente o aumento do grau de dependabilidade¹⁰ do sistema.

Em geral, o processo de tratar falhas em sistemas de computação envolve quatro fases (Stallings, 1999): detecção, isolamento, restauração. Para detectar falhas, pode-se instaurar mecanismos de replicação de componentes, checagens periódicas, diagnósticos ou mecanismos de exceção. Após a detecção, deve-se estipular a abrangência das conseqüências da falha, isolando a falha.

Na fase de restauração o sistema precisa retornar a um estado válido. Há duas abordagens para viabilizar a restauração: recuperação de estado válido anterior¹¹ e recuperação de estado válido posterior¹². A abordagem mais utilizada em sistemas de tempo real é a de estado válido anterior e consiste na gravação periódica de estados válidos do sistema para posterior restauração. Evidentemente, esta abordagem envolve o uso de um serviço de persistência para manutenção dos estados válidos, bem como exige a capacidade de restauração com base em um destes estados. Analogamente, a abordagem de recuperação de estado válido posterior consiste na restauração do sistema com base em estimativas de processamento onde é estimado um novo estado válido.

Por fim, há casos onde há falha permanente em componentes do sistema de modo que a restauração não é possível. Assim, devem ser utilizados componentes redundantes para substituição de componentes falhos em tempo de execução.

Sistemas distribuídos são constituídos de vários processadores independentes e diferenciam-se de computadores paralelos pelo acoplamento fraco entre seus processadores, ou seja, os elementos de um sistema distribuído não têm acesso a uma memória comum (Tanenbaum, 2002). Desse modo, toda interação é feita por troca de mensagens através de uma rede de comunicação. Por isso, estes sistemas apresentam um problema bastante peculiar, que é garantir a integridade e a consistência dos dados distribuídos pelos vários processadores. Esse problema força a aplicação de técnicas de

¹⁰ Dependabilidade significa que um sistema é confiável na execução de uma tarefa à qual foi designado. Dependabilidade pode ser decomposta em aspectos específicos: confiabilidade (habilidade de o sistema de executar corretamente a tarefa), disponibilidade (habilidade de o sistema estar disponível quando necessário) e segurança (habilidade de o sistema prevenir acesso não-autorizado ao sistema).

¹¹ Do inglês, *backward error recovery*.

¹² Do inglês, *forward error recovery*.

tolerância a falhas, mesmo quando não existe qualquer outra exigência mais forte quanto à confiabilidade imposta pela aplicação.

Em geral, sistemas distribuídos devem operar sabendo que, a qualquer momento, um dos componentes do sistema pode falhar (Blanke et. al, 2001). Assim, o projetista do sistema deve prever, em tempo de desenvolvimento, potenciais falhas do sistema, bem como desenvolver soluções para eliminar (ou pelo menos amenizar) suas conseqüências.

Assim como em arquiteturas paralelas, sistemas distribuídos apresentam uma redundância natural, extremamente proveitosa para o emprego de técnicas de tolerância a falhas. Defeito em um nó da rede não precisa provocar necessariamente a queda de todo o sistema, e o sistema pode ser facilmente reconfigurado usando apenas os nós disponíveis (Mullender, 1993).

Geralmente, a maioria das falhas é decorrente da interrupção do funcionamento de algum dos recursos físicos do sistema, como servidores de mídias, servidores de autenticação, inativação do enlace de comunicação, dentre outros. Logo, uma alternativa para resolver estas falhas seria por meio do uso de redundância de componentes físicos.

Assim, devido à instabilidade das redes de comunicação e até mesmo dos próprios dispositivos físicos é bastante comum a ocorrência de perdas de dados momentâneos ou até mesmo panes gerais de sistema (Tanenbaum, 2002; Jalote, 1994).

Caso haja uma falha em um servidor, podem ser aplicadas técnicas de migração de processos entre os servidores do sistema.

4.6 Considerações finais

Neste capítulo, foram apresentados os principais aspectos relativos ao projeto e à implementação de sistemas AVR, bem como alguns tópicos relativos à gerência de falhas em sistemas distribuídos. Os vários aspectos pesquisados são referentes ao atendimento dos requisitos funcionais e não-funcionais descritos no capítulo anterior.

Com o objetivo de prover uma melhor qualidade de serviço aos usuários dos AVR, os aspectos relativos à escalabilidade deverão ser sempre previstos. Para isso, é de suma importância o gerenciamento de entidades que podem afetar a escalabilidade do sistema, como conexões com usuários, processamento de entidades virtuais, capacidade de processamento dos servidores, largura de banda utilizada e total, dentre outros.

Durante a fase projeto do sistema, é preciso ainda prever os impactos decorrentes da utilização do sistema de comunicação, como utilização da largura de banda, latência e

confiabilidade. Com o objetivo de amenizar esses impactos podem ser implementadas algumas técnicas compensatórias.

Como resultado dos fatores analisados neste capítulo, foi desenvolvido o NuVEM, um modelo de gerenciamento do sistema AVR contendo algumas estratégias para otimizar o uso dos recursos contidos no sistema. Em especial, o modelo descrito a seguir adiciona à arquitetura de AVR mecanismos para detecção e resolução de falhas em componentes críticos do sistema.

Capítulo 5

NuVEM – Um modelo para gerenciamento de AVR

5.1 Visão geral

Neste capítulo, é apresentado o NuVEM – Modelo para gerenciamento de ambientes virtuais em rede –, definido para prover mecanismos de controle e monitoramento de sistemas AVR. Este modelo consiste na aplicação de tecnologias de gerenciamento de redes voltadas para a área de AVR e, especificamente, gerenciamento de falhas.

O objetivo principal da criação do modelo descrito neste trabalho consiste na adoção de métodos para detecção e resolução de falhas em sistemas AVR. Apesar de focado em uma área particular do Gerenciamento de Redes (descrito no capítulo 2), o NuVEM fornece um arcabouço que poderá ser usado em trabalhos futuros para a gerência completa do sistema, abrangendo todas as áreas previstas no Gerenciamento de Redes.

O restante deste capítulo está dividido da seguinte forma: na Seção 5.2, são discutidos requisitos para o modelo de gerência; na Seção 5.3, é apresentado o funcionamento do NuVEM, onde são destacados seus componentes, suas interfaces com o sistema AVR e os cenários de utilização previstos. Por fim, nas seções 5.4 e 5.5, são apresentados alguns trabalhos relacionados e algumas considerações finais, respectivamente.

5.2 Requisitos para gerência remota de AVR

O gerenciamento de um sistema AVR fundamenta-se no gerenciamento do ciclo de vida dos mundos virtuais instanciados neste sistema, que consiste em manipular as entidades ligadas a estes mundos (conexões com usuários, processamento de entidades virtuais, capacidade de processamento no servidor hospedeiro, largura de banda utilizada, etc.). Os principais requisitos para gerenciamento dessas instâncias são mecanismos para criação, configuração, migração e exclusão dos mundos. Para este fim, é importante que o administrador tenha ferramentas de gerenciamento que dêem uma visão ampla e transparente da localização das instâncias de mundos virtuais na rede (Almendra et al., 2003).

Um dos aspectos da gerência remota, que também pode ser considerado um dos principais aspectos de sistemas distribuídos, é a instanciação e localização de serviços (Almendra, 2003). Assim, uma estratégia de instanciação remota tem que ser definida.

Outro detalhe importante consiste na disponibilidade dos mundos virtuais contidos nos sistema em um determinado momento. Com o propósito de fornecer uma maior disponibilidade do sistema aos usuários é que devem ser adotadas estratégias para resoluções de falhas assim como políticas de segurança para garantir uma maior confiabilidade ao sistema.

A seguir, são discutidas estratégias para gerenciamento dos servidores dispostos na rede, para instanciação remota e para resolução em caso de falhas.

5.2.1 Gerenciamento de servidores e mundos virtuais

Uma arquitetura de AVR geralmente possui vários servidores distribuídos fisicamente para processamento dos mundos virtuais. Ações para instanciação, configuração e desinstanciação de mundos virtuais remotos são suportadas através de mensagens de controle trocadas entre os servidores da arquitetura.

Para fins de gerenciamento dos servidores de mundos virtuais, é necessária a instalação de um agente em cada servidor de modo que este monitore localmente seus recursos e envie suas estatísticas para o gerente do sistema. O agente deve se autenticar com o gerente no momento de sua entrada no sistema. No instante seguinte, o agente já estará habilitado a responder qualquer requisição proveniente do gerente. Estas requisições vão desde comandos para instanciação de mundos até monitoração de características do servidor, como tráfego de rede, taxas de uso de memória e processamento, disponibilidade, etc.

Com o propósito de gerenciar aspectos relativos aos mundos virtuais instanciados, são utilizadas peculiaridades do mundo virtual. Desse modo, algumas características específicas dos ambientes virtuais devem ser compartilhadas com o sistema de gerenciamento. Dentre estas características destacam-se: quantidades de conexões de entidades virtuais (usuários, simulóides e objetos interativos), estrutura arquitetônica, cargas de processamento, uso de memória, etc.

5.2.2 Instanciação remota

A instanciação de mundos virtuais em servidores da rede significa a inicialização de um processo na máquina servidora. Este processo será responsável pela manutenção da consistência e das peculiaridades do mundo virtual.

No entanto, a instanciação de processos em estações remotas envolve o uso de estruturas mais complexas que as tradicionais, como métodos construtores e carregamento dinâmico, disponíveis na maioria das linguagens. Estas estruturas tradicionais de criação podem instanciar novos processos somente no espaço de memória nas quais estão alocadas. No caso específico de instanciação remota é necessário que seja possível implementar um processo em um outro espaço de memória ou até mesmo em máquinas diferentes (Emmerich, 2000).

Com o propósito de viabilizar a instanciação remota, deve ser implementado nos agentes um módulo que possua a capacidade de instanciar mundos virtuais. Este módulo, chamado de Fábrica, encapsula os detalhes de instanciação e execução de mundos a partir de requisições feitas pelo administrador do sistema por meio do agente. Esta estratégia de fabricação de processos é uma extensão do padrão de projeto "Abstract Factory" (Gamma et al., 1995), onde se tem uma interface padrão para criação de famílias de objetos relacionados ou não.

No momento de instanciação de um novo mundo virtual no sistema, é altamente recomendável que o administrador não precise especificar exatamente qual agente seria responsável pela instanciação. O gerente deve decidir, através de consultas aos agentes, qual servidor (equipado de um agente) estaria com maior quantidade de recursos computacionais livres para alocação de uma nova instância de mundo virtual. Logo, a fim de evitar a instanciação de um mundo virtual em um servidor com recursos computacionais insuficientes para seu processamento, o gerente deve decidir através de estatísticas colhidas pelos agentes qual servidor estaria com maior aptidão. Apesar de não fazer parte do escopo deste trabalho, é previsto que seja feita uma análise de carga nos servidores de modo a viabilizar a instanciação remota.

5.2.3 Recuperação de estado de falha

Em sistemas de computação, as aplicações podem terminar sua execução de forma abrupta por vários motivos: erro na execução de uma transação, insuficiência de recursos (memória, CPU), falha interna (ou no próprio sistema operacional), queda da rede de comunicação, etc (Mullender, 1993). Nesses casos, os sistemas de gerência devem ser

capazes de se recuperar dessas falhas. Isso pode ser realizado, por exemplo, através da definição de estados de execução e da criação de *logs* para a realização de operações de recuperação de falhas (Jalote, 1994).

São requisitos de um sistema de gerenciamento de falhas detectar (ou perceber), reportar e solucionar falhas (Stallings, 1999). O sistema deve ter habilidades para manter registros de eventos significantes e erros. Estes registros serão necessários no momento de restauração do estado funcional do componente que provocou a falha. No sistema de gerenciamento da arquitetura de AVR, cada servidor de recursos deve estar em contato com um outro servidor, o servidor de persistência compartilhado, que guardará os estados intermediários dos mundos virtuais.

O gerente deve efetuar a migração das instâncias de mundos presentes no servidor falho para outro servidor. Ao mesmo tempo, o gerente entra em contato com o servidor de persistência compartilhado para saber quais instâncias de mundos foram afetadas pela falha ocorrida. A partir daí, o gerente poderá requisitar a re-instanciação destes mundos virtuais.

Existe também o caso do gerente do sistema falhar. Para resolver este problema pode-se criar um outro servidor gerente, chamado de reserva, que, em intervalos de tempo pré-definidos, verifica a disponibilidade do gerente. Assim, se o gerente falhar, o reserva toma posse de sua posição, evitando que o sistema de gerenciamento pare.

Para os casos acima, a tolerância a falhas pode ser viabilizada por meio de tratamentos de exceções (Cristian, 1995). Um dos conceitos mais utilizados acerca de tolerância a falhas por meio de tratamento de exceções é o *Idealized Fault Tolerance Components* (IFTC) (Lee & Anderson, 1991). Este conceito isola os comportamentos normal e anormal de cada componente do sistema de modo que quando uma falha ocorre o componente dispara uma exceção.

Em conjunto com as estratégias previstas pelo IFTC, o gerente do sistema deve utilizar a abordagem de recuperação de estado válido anterior – *backward error recovery* – (Capítulo 4) para resolução de falhas. Nessa abordagem é utilizado o último estado válido do mundo virtual – gravado de acordo com regras pré-estabelecidas entre o servidor de persistência e os agentes – como ponto de restauração.

5.3 Funcionamento do NuVEM

O NuVEM define um modelo de comunicação centralizado, onde deve haver apenas um único gerente ativo por vez em contato com vários agentes subordinados (Capítulo 2; Capítulo 4). Este modelo simplifica a implementação do sistema, facilitando o

controle por reunir as estratégias de gerência em um só componente. Além de centralizado, o NuVEM utiliza o método *Unicast* de transmissão de dados, pois este método possibilita que os componentes do sistema mantenham uma alta sincronia de seus estados (Capítulo 4).

A seguir, são descritos detalhes referentes ao funcionamento do NuVEM: seus componentes, os cenários de utilização previstos e suas interfaces de comunicação com o sistema AVR.

5.3.1 Componentes

Os componentes decorrentes dos requisitos expostos na seção anterior são utilizados para disponibilizar, simultaneamente, um ou mais mundos virtuais. Além disso, estes componentes representam um arcabouço para a implementação de técnicas de gerenciamento de redes e, especificamente, gerenciamento de falhas. Para isso, foram definidos 3 (três) componentes (Figura 5.1):

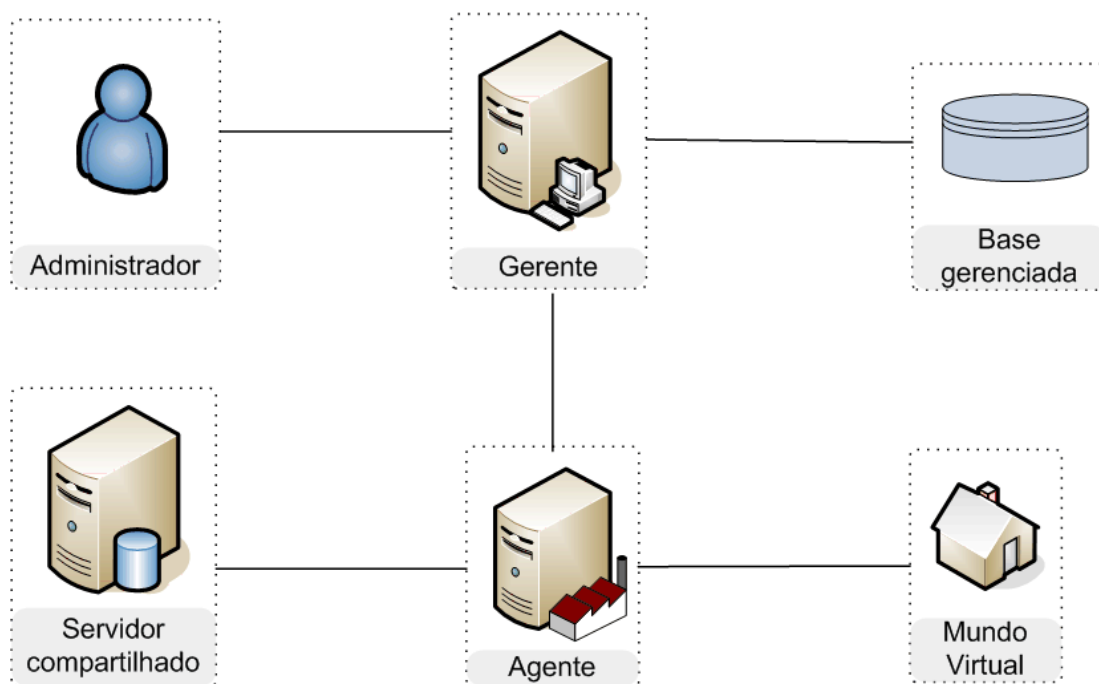


Figura 5.1. Componentes do NuVEM.

- ◆ Servidor compartilhado – é o responsável pela manutenção dos estados intermediários dos mundos virtuais cadastrados no sistema AVR. É um ponto de interseção entre o sistema de gerência e o sistema AVR e será utilizado para registrar dados provenientes das interações nos mundos virtuais para uso durante as rotinas de recuperação de falhas. Basicamente, consiste em uma aplicação servidora

de mídias – possivelmente um banco de dados – com uma interface (de comunicação) padrão para consultas por parte dos Agentes implantados no sistema.

- ◆ Agente – é a aplicação contida nos servidores de recursos com o objetivo de coletar e armazenar dados do sistema local e dos mundos instanciados, além de responder às requisições feitas pelo Gerente do sistema. Faz parte da aplicação Agente o módulo para instanciação de mundos virtuais (Fábrica).
- ◆ Gerente – é a aplicação centralizadora responsável pela manutenção de registros gerais de todo o sistema de gerência, como conexões com Agentes, dados de usuários e mundos virtuais do sistema AVR, métodos para resolução de falhas, etc. Estes registros devem persistir em uma base de informações gerenciadas de uso exclusivo do Gerente.

Além dos componentes supracitados, existe um outro componente que é intrínseco aos propósitos do NuVEM: o mundo virtual (Figura 5.1). O mundo virtual é o componente do sistema AVR mais importante para o sistema de gerenciamento. Além dos componentes já descritos é previsto que o NuVEM seja integrado aos componentes do pro AVR. Essa integração é constituída com base nas interfaces de comunicação presentes nos componentes Gerente e Agente (ver seção 5.3.2).

Esses componentes podem ser executados em um número ilimitado de computadores de modo a prover um número também ilimitado de mundos virtuais. Para isso, é previsto que sejam ativados apenas um Servidor compartilhado e dois Gerentes (um titular e um reserva). O número de Agentes contidos no sistema fica a cargo do seu administrador. No entanto, não há necessidade de iniciar mais de um Agente por servidor de recursos.

Como usuário do sistema é destacado o Administrador, que é o responsável pela manipulação do sistema. Dependendo da flexibilidade das interfaces gráficas dos componentes, o Administrador pode ser desde um especialista em informática até um usuário leigo ou operador de sistemas.

A seguir, são apresentados os principais cenários de utilização do NuVEM para ilustrar seu funcionamento.

5.3.2 Cenários de Utilização

Nessa seção, são apresentados os cenários encontrados durante a utilização do NuVEM. Para cada cenário, será detalhado o conjunto de ações realizadas por cada componente do sistema envolvido.

Cenário 1: Iniciação do Gerente

Este cenário, por sinal o mais simples, consiste basicamente da ativação de uma aplicação Gerente em um servidor da rede. Desse modo, o Gerente inicia sua base de objetos gerenciados e fica em modo passivo (Figura 5.2 a-b). A partir daí, espera por conexões de Agentes para mudar para o modo ativo (Figura 5.2 c). A distinção entre os modos passivo e ativo é utilizada para dissociar os Gerentes titular e reserva, onde o reserva fica em modo passivo até o momento em que for necessária sua intervenção diante de uma falha do Gerente titular. Após iniciação do Gerente o administrador do sistema deve iniciar o Servidor compartilhado.

Conforme dito anteriormente, de início devem ser instanciados no sistema apenas 2 (dois) Gerentes. A partir daí, os Agentes deverão fazer conexão com um dos Gerentes, que passa para o modo ativo e mantém uma referência para o seu reserva (que deve ser passada aos Agentes no momento do pedido de conexão). Para que o sistema funcione em condições mínimas, é necessária a ativação de apenas um Gerente. No entanto, em caso de falha do Gerente, é possível que o sistema inteiro pare.

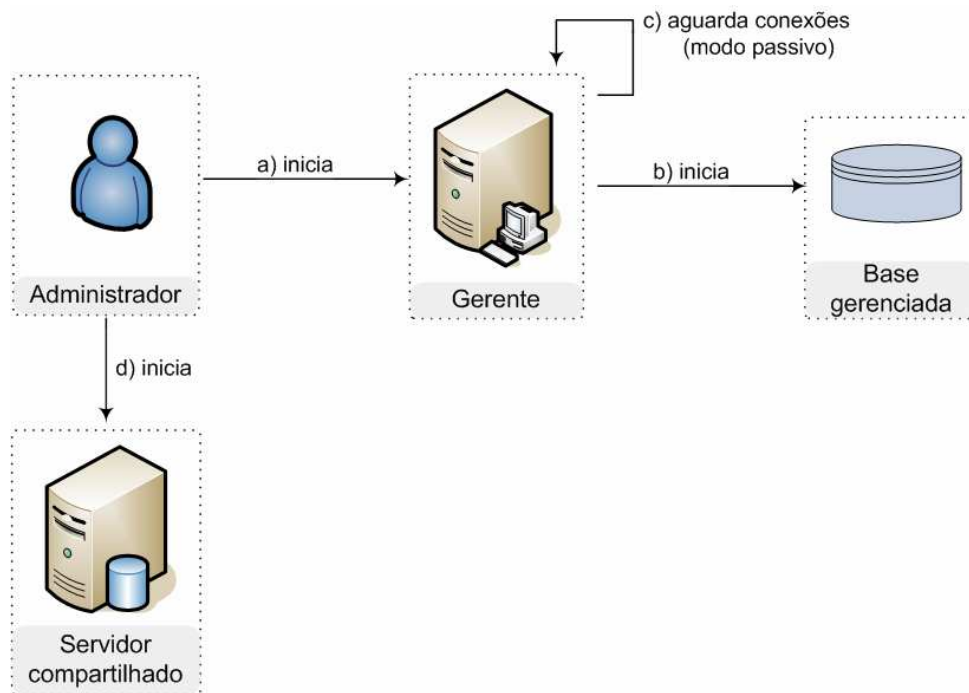


Figura 5.2. Iniciação do Gerente.

Juntamente com o Gerente do sistema deve ser iniciado o componente Servidor compartilhado (Figura 5.2 d).

O Gerente deve ter uma interface (GUI) padrão para monitoração e controle do sistema por parte do Administrador. Esta interface será acessada durante o processo de instanciação remota.

Cenário 2: Instalação e iniciação do Agente

Antes de um iniciar qualquer agente no sistema, ambos os Gerentes, titular e reserva, devem ser iniciados, assim como o Servidor compartilhado. Por incluir um módulo específico para instanciação de mundos virtuais (Fábrica), é necessário que a aplicação Agente tenha permissões de sistema suficientes para iniciar aplicativos em tempo de execução. Além disso, a Fábrica precisa ser equipada com dados específicos dos mundos virtuais os quais ela pode instanciar. Estes dados são provenientes do próprio sistema AVR e não devem ser alterados pelo sistema de gerência.

A instalação de Agentes envolve uma simples carga do programa executável ou ainda, dependendo dos recursos disponíveis, como um serviço persistente no sistema operacional (Figura 5.3 a). Sempre que um Agente é iniciado, é feita sua autenticação com o Gerente (Figura 5.3 b) e, a partir de então, o Gerente contabiliza o novo Agente, passando a contar com os recursos de seu servidor hospedeiro para instanciar mundos virtuais no servidor hospedeiro (Figura 5.3 d). Durante a fase de autenticação do Agente, o Gerente envia dados relativos ao Gerente reserva e requisita dados de sistema do Agente. Caso o Gerente esteja em modo passivo antes da autenticação, isto é, este é o primeiro Agente a fazer autenticação, o mesmo deverá mudar para o modo ativo (Figura 5.3 c). Nesse novo modo, o Gerente deve monitorar periodicamente os Agentes, bem como responder às requisições do administrador do sistema e receber notificações de eventos previamente configurados (como eventos de notificação de falhas).

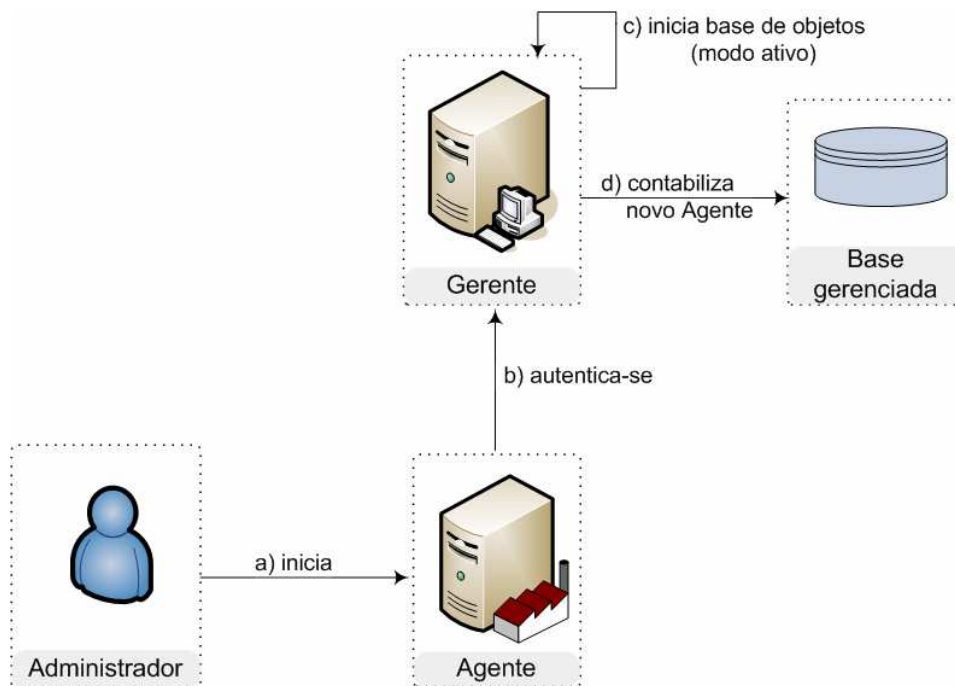


Figura 5.3. Iniciação do Agente.

Cenário 3: Utilizando a Instanciação remota

Logo após a iniciação dos componentes de gerência (Gerente, Servidor compartilhado e Agente), já é possível instanciar mundos virtuais nos servidores da rede. Por meio de uma interface disponibilizada pelo Gerente, o administrador do sistema pode instanciar mundos virtuais.

Para isso, o administrador faz uma requisição diretamente ao Gerente ativo no sistema para que este encaminhe um comando de instanciação (Figura 5.4 a). Apesar de não fazer parte do escopo deste trabalho, é prevista uma análise de carga dos Agentes para que o Gerente escolha qual servidor deverá processar a nova instância (Figura 5.4 b). De posse do servidor escolhido, o Gerente envia uma requisição ao Agente respectivo (Figura 5.4 c), que contata o Servidor compartilhado para pegar informações do mundo virtual (Figura 5.4 d) e inicia seu processamento (Figura 5.4 e).

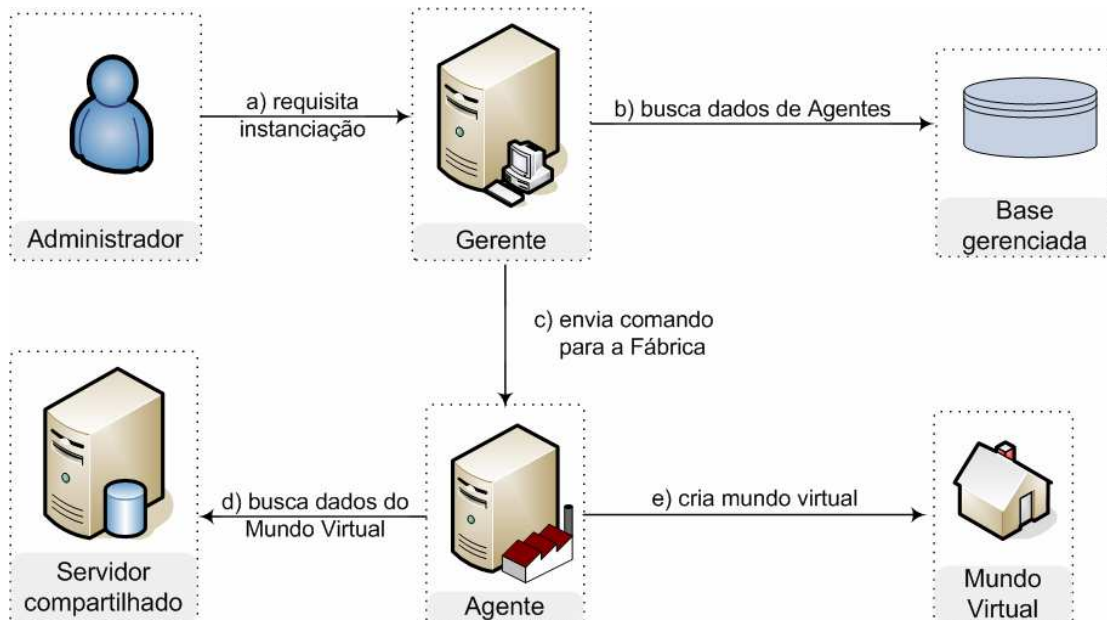


Figura 5.4. Instanciação remota.

Cenário 4: Gravação no Servidor compartilhado

A partir da instanciação do mundo virtual, o Agente passa a monitorar seus estados, que consistem em dados relativos às interações dos participantes do ambiente virtual entre si e dos participantes com o próprio ambiente virtual. O estado de um mundo virtual consiste no estado de seus componentes estáticos e dinâmicos. Componentes estáticos são componentes que não apresentam mudança em suas características ao longo do tempo. Em geral, esses componentes são relativos à estrutura arquitetônica do mundo virtual.

Ao contrário dos componentes estáticos, os componentes dinâmicos podem sofrer mudanças em suas características. Essas mudanças podem ser provocadas, ou não, por

usuários através de interações no próprio mundo virtual, isto é, as mudanças nos estados dos componentes dinâmicos ocorrem de modo pró-ativo ou reativo. Em geral, são componentes dinâmicos do mundo virtual os simulóides, usuários, objetos interativos, etc.

Por questões de otimização, dados estáticos do mundo virtual não são gravados no Servidor Compartilhado, apenas a parte dinâmica (mutável) do mundo virtual deve ser gravada. Desse modo, é configurada pelo administrador do sistema a frequência com que o Agente colherá (Figura 5.5 a) e gravará no Servidor Compartilhado (Figura 5.5 b) os estados dos mundos virtuais que estão sob sua responsabilidade. Esses estados serão utilizados durante os processos de recuperação de falhas no sistema.

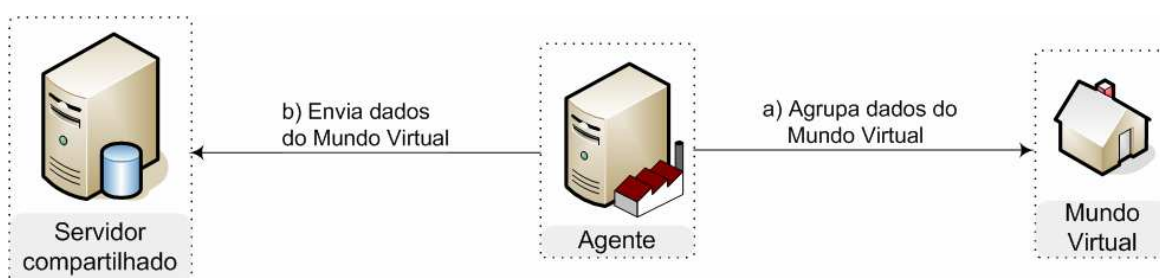


Figura 5.5. Gravação no Servidor compartilhado.

Ainda por questões de otimização, é possível que sejam estabelecidas frequências de gravação variáveis para que mundos virtuais com maiores modificações em seus estados tenham maiores frequências de gravação que mundos virtuais com menores modificações em seus estados ao longo do tempo. A operação de gravação no Servidor Compartilhado deve ser atômica e, evidentemente, quanto maior a quantidade de dados a serem enviados, maior o tempo necessário para conclusão da mesma.

Cenário 5: Resolução de falhas – Caso 1: Falha em um dos mundos virtuais

Mesmo após o processo de autenticação do Agente, é mantida a sua conexão com o Gerente. Esta conexão permite que o Gerente receba notificações oriundas do Agente, e vice-versa. Desse modo, é possível que o Agente envie notificações (*traps*) acerca do funcionamento de seus mundos virtuais instanciados localmente. Caso haja alguma situação que caracterize falha em alguma instância, o Agente dispara uma notificação para o Gerente. As situações que caracterizam falhas são previstas em tempo de desenvolvimento do sistema e não podem ser alteradas em tempo de execução. Em versões posteriores do NuVEM é possível que sejam utilizadas técnicas de *plugins* para adicionar ou remover tipos de falhas a serem monitoradas em tempo de execução.

O processo de resolução de falhas em Mundos Virtuais consiste basicamente de três etapas: a detecção da falha por parte do Agente (Figura 5.6 a), o envio da notificação ao

Gerente (Figura 5.6 b) e a re-instanciação do mundo virtual (Figura 5.6 c-f), se necessário. Dependendo do estado do Agente que enviou a notificação, o Mundo Virtual pode ser re-instanciado ou instanciado por outro Agente.



Figura 5.6. Resolução de falha em Mundo Virtual.

Cenário 6: Resolução de falhas – Caso 2: Um Agente falha

Em sistemas de comunicação, falhas ocasionalmente provocam a parada total (*fail-stop*) da estação servidora (Mullender, 1993). Falhas desse tipo consistem em perda de comunicação com quaisquer outras estações do sistema.

Em períodos de tempo fixos e pré-estabelecidos, o Gerente deve fazer comunicação com o Agente de modo uma falha de conexão seja rapidamente percebida. Caso o Gerente constate a falha, segue-se um período de testes de conexão para que a falha seja confirmada e então são tomadas as devidas providências para recuperação. Os períodos de tempo utilizados para confirmar a falha nos Agentes devem ser definidos em tempo de implantação e integração do NuVEM ao sistema de ambientes virtuais.

Diante de uma falha na conexão (Figura 5.7 a), é disparada uma exceção no Gerente (Figura 5.7 b) indicando os dados do Agente falho. São feitos alguns testes de conexão com o Agente de modo a confirmar a queda de comunicação (Figura 5.7 c). Caso seja confirmada a falha o cenário prossegue, senão o sistema volta ao seu estado anterior, sem falhas.

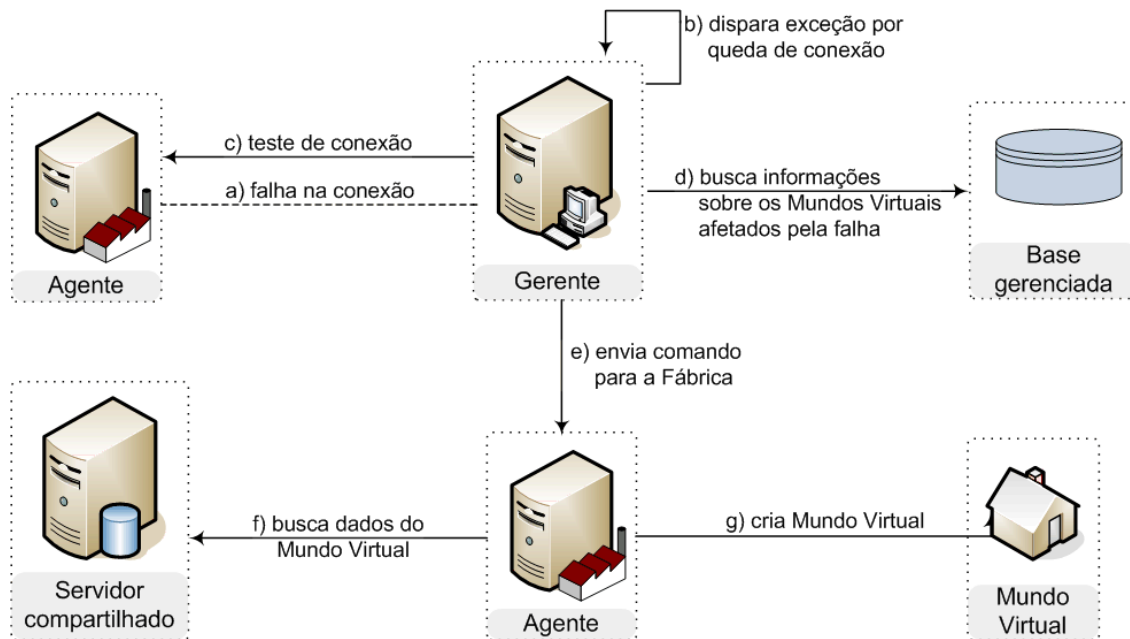


Figura 5.7. Resolução de falha em Agente.

Desse modo, ao ratificar a falha no Agente o Gerente busca informações detalhadas acerca dos mundos virtuais afetados (Figura 5.7 d), instanciando-os novamente em outros servidores remotos (Figura 5.7 e-g). Logo, a última etapa deste cenário consiste na execução de vários comandos de instanciação remota.

Cenário 7: Resolução de falhas – Caso 3: O Gerente do sistema falha

De forma análoga ao cenário anterior, é possível que ocorra uma falha no Gerente. Com o intuito de solucionar esta falha, é utilizado um mecanismo de redundância de componentes onde é utilizado um Gerente reserva, que fica em estado passivo até o momento em que é chamado por algum Agente. O processo de recuperação neste cenário consiste em fazer com que os Agentes conectem-se ao Gerente reserva.

Ainda de forma semelhante ao cenário anterior, neste cenário é previsto que haja um período de testes onde os agentes deveriam confirmar a falha no Gerente. Este passo é muito importante e evita, principalmente, que existam dois gerentes ativos no sistema ao mesmo tempo, o que contraria o fato de o Gerente se um componente centralizador no NuVEM. Assim, caso a falha no Gerente seja intermitente, o Gerente reserva não é necessariamente ativado.

Diante de uma falha na conexão (Figura 5.8 a), é disparada uma exceção em cada Agente autenticado (Figura 5.8 b) indicando que o Gerente está inacessível. A partir daí, cada Agente efetua alguns testes de conexão com o Gerente durante um período de tempo pré-estabelecido de modo a confirmar a queda de comunicação (Figura 5.8 c). Caso seja

confirmada a falha no Gerente o cenário prossegue, senão o Agente volta ao seu estado anterior, sem falhas de comunicação com o Gerente.

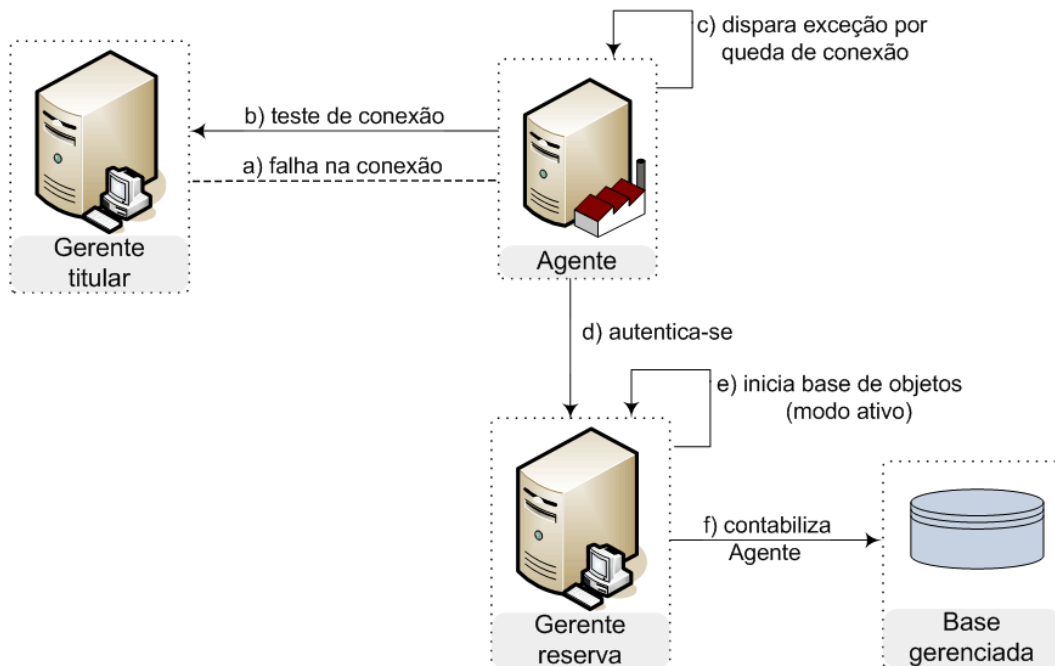


Figura 5.8. Resolução de falha no Gerente.

Ao comprovar a falha cada Agente deve utilizar as informações sobre o Gerente reserva concedidas pelo Gerente falho durante a fase de autenticação. Com base nessas informações o Agente se autentica junto ao Gerente reserva (Figura 5.8 d), que, por sua vez, o muda seu estado para o modo ativo (Figura 5.8 e), contabilizando o Agente recém autenticado (Figura 5.8 f) e assumindo o papel de Gerente único do sistema.

Opcionalmente, o administrador do sistema poderia ser notificado por e-mail ou alguma outra ferramenta de comunicação no momento da falha no Gerente. Desse modo, o administrador poderia ser notificado no momento em que o Gerente reserva mudasse para o estado ativo.

5.3.3 Interfaces com o sistema AVR

Ao gerenciar o ciclo de vida dos mundos virtuais contidos no sistema AVR, algumas características particulares aos ambientes virtuais devem ser compartilhadas com o NuVEM. Atributos como quantidade de entidades virtuais conectadas, estados das entidades, cargas de processamento, uso de memória, taxas de transmissão, etc, devem ser acessíveis a ambos os sistemas.

Logo, para que o NuVEM funcione corretamente quando aplicado a um sistema AVR previamente construído, é preciso que sejam implementadas algumas interfaces de comunicação entre os dois sistemas.

Primeiramente, é preciso que o componente Gerente tenha acesso às informações de permissões dos usuários e mundos virtuais do sistema AVR. Estas informações serão necessárias durante processos de instanciação automática (Cenário 3) e autenticação de usuários. Em versões posteriores do NuVEM é possível que sejam utilizados mecanismos de gerência de segurança para fornecer ao sistema AVR maior confiabilidade.

Para que os Agentes tenham acesso aos estados de cada mundo virtual é necessário que haja uma interface de comunicação entre os Agentes e o componente do sistema AVR onde ficam armazenadas informações relativas aos estados das estruturas estáticas e dinâmicas de cada mundo virtual. Essas informações serão utilizadas tanto durante o processo de gravação no Servidor Compartilhado (Cenário 4) como durante o processo de Instanciação remota (Cenário 3).

Existe ainda a possibilidade de o administrador do sistema requisitar informações dos mundos virtuais instanciados. Para isso, é preciso que o Agente tenha uma interface de comunicação com o componente mundo virtual do sistema AVR. Essas informações vão desde características dos mundos virtuais até características do próprio servidor de recursos onde foi alocado o Agente.

5.4 Trabalhos relacionados

Alguns trabalhos na área de desenvolvimento e implementação de arquiteturas de AVR podem ser encontrados na literatura. A arquitetura DIVE (Carlsson & Hangsand, 1993) tenta aumentar sua escalabilidade por meio do uso de protocolos *multicast* para distribuir mensagens entre seus participantes. Já a arquitetura NPSNET-IV (Capps et al., 2000) especifica mecanismos de componentes dinamicamente extensíveis para alcançar maior escalabilidade. A arquitetura VELVET (Oliveira & Georganas, 2002) também usa técnicas de *multicast*, mas aliadas a técnicas de gerenciamento de interesses relativos aos espaços geográficos do ambiente virtual para aumentar a escalabilidade do sistema. Todos estes sistemas definem estruturas para criar e distribuir mundos virtuais pela rede, no entanto, nenhum deles especifica métodos para tolerância ou recuperação em caso de falhas.

Alguns outros sistemas como o AlphaWorlds (Dodge, 1999; Ryan, 2004) ou COVET (Hosseini & Georganas, 2001) nem ao menos citam aspectos de escalabilidade ou gerenciamento de seus ambientes virtuais.

5.5 Considerações Finais

As estratégias definidas pelo NuVEM neste capítulo fornecem meios para a administração e o uso otimizado de um sistema AVR, pois viabiliza a monitoração e o controle das instâncias de mundos virtuais contidas no sistema. Funcionalidades como criação, configuração, migração e exclusão de mundos são suportadas. Ademais, o NuVEM fornece um arcabouço para gerenciamento de toda arquitetura de ambientes virtuais.

Dentre os detalhes de funcionamento do NuVEM descritos anteriormente, destacam-se os cenários onde são previstas falhas em componentes críticos do sistema. Desse modo, estão previstas falhas nos componentes: Mundo Virtual, Agente e Gerente. Como o componente Servidor Compartilhado consiste em uma aplicação servidora de mídias, sua implementação pode ser completamente reutilizada de terceiros. Caso durante a implementação seja escolhido utilizar um sistema de gerenciamento de banco de dados (SGBD), poder-se-ia utilizar o próprio mecanismo de resolução de falhas desse sistema.

Caso necessário, pode-se usar uma interface de acesso ao Gerente que seja independente do componente Gerente. Assim, para o cenário onde há falha do Gerente (Figura 5.8 a-f), o administrador do sistema poderia acessar o Gerente de modo transparente com relação à sua localização na rede ou mesmo com relação a qual dos gerentes ele esteja acessando, se é o titular ou o reserva.

A seguir, será descrito um estudo de caso da integração do NuVEM a um sistema AVR, o CRAbCVE (Gomes, 2005). Esta integração consiste na implementação dos componentes descritos anteriormente, bem como de suas funcionalidades de resolução de falhas e interfaces com o próprio sistema AVR.

Capítulo 6

Gerenciamento Remoto de Falhas em uma arquitetura de AVR

6.1 Introdução

Este capítulo descreve a implementação das estratégias previstas no NuVEM descrito no capítulo anterior aplicadas a uma arquitetura de ambientes virtuais, a arquitetura CRAbCVE (Gomes, 2005). As implementações estão relacionadas ao projeto Sintegra – Sistema integrado de Educação a Distância com suporte a Avaliação de Competências – (Sintegra, 2006), que é um projeto conseqüente de uma parceria entre SENAC/CE e a Universidade Federal do Ceará (UFC).

Na Seção 6.2 é apresentada a arquitetura CRAbCVE. Na Seção 6.3, apresentam-se as implementações do Gerente e do Agente do sistema de gerência, bem como sua integração aos componentes da arquitetura CRAbCVE. Na seção 6.4 são descritos alguns testes de funcionalidade e alguns resultados decorrentes da integração, além de algumas considerações finais.

6.2 Arquitetura CRAbCVE

A arquitetura CRAbCVE (Gomes, 2005) foi projetada para viabilizar CVEs de baixo custo (utilizando computadores comuns) através da Internet. A CRAbCVE é uma evolução da arquitetura Ataxia (Leite-Junior, 2000) que inclui elementos para permitir o trabalho colaborativo.

A concepção de ambientes virtuais de baixo custo requer a adoção de uma estratégia que divida as funções de sustentação do sistema AVR entre diversos componentes, que podem então ser alocados em máquinas separadas. Essa abordagem, baseada no processamento distribuído em sistemas de baixo custo, apresenta uma relação custo/benefício superior ao uso de poderosos e caros sistemas para processamento centralizado (Tanenbaum, 2002). A CRAbCVE define, então, um conjunto de componentes que podem ser executados em diversos microcomputadores comuns, de maneira simultânea, sendo cada um responsável por prover serviços específicos para o sistema.

Durante o desenvolvimento da arquitetura CRAbCVE, foram definidos componentes especializados (Figura 6.1) de modo a prover suporte aos processos de colaboração (Borges et al., 1995) e permitir a publicação de mundos virtuais, garantindo uma maior acessibilidade e tornando a CRAbCVE uma arquitetura de uso genérico e extensível.

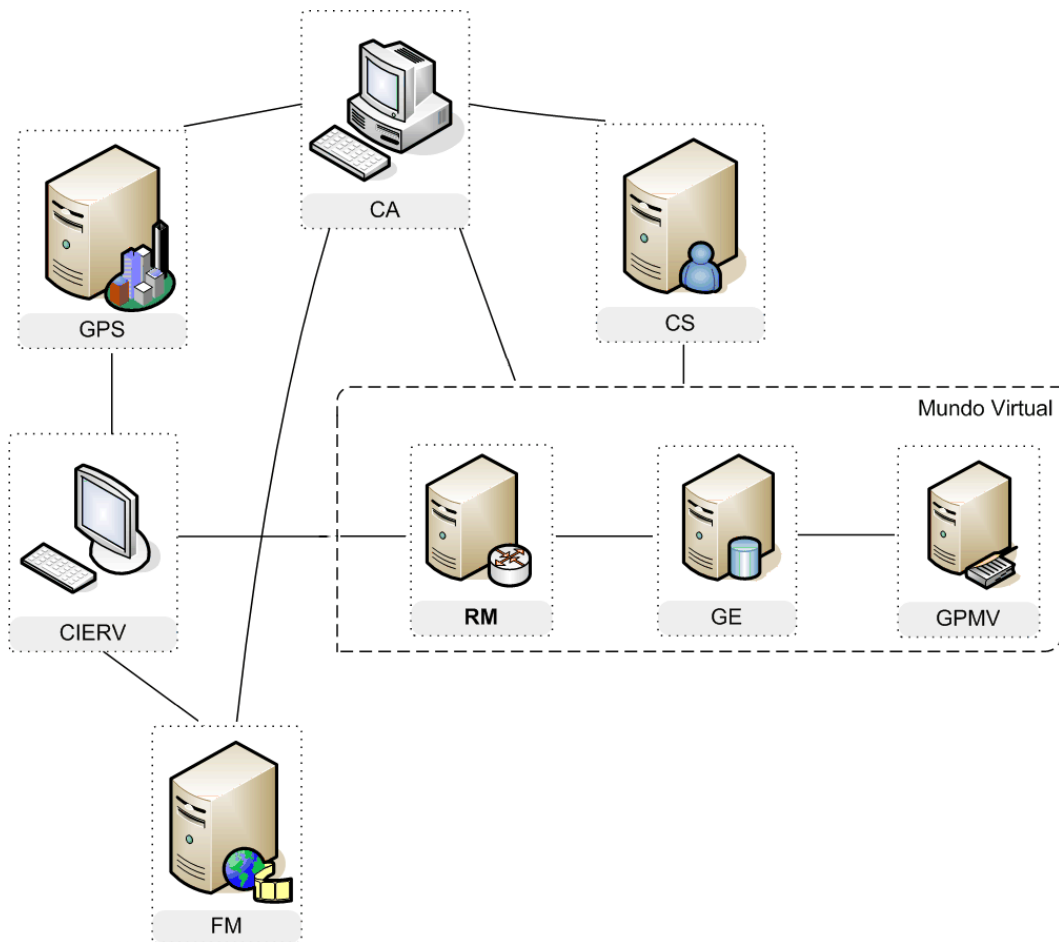


Figura 6.1. Componentes da arquitetura CRAbCVE.

A publicação de mundos virtuais é uma funcionalidade muito importante, pois permite que mundos virtuais construídos pelos participantes a partir de bibliotecas de modelos pré-fabricados fiquem automaticamente disponíveis para a exploração por outros usuários do sistema. Assim, a criação de mundos virtuais deixa de ser uma tarefa exclusiva de especialistas para se tornar uma atividade de autoria, ao alcance de usuários das mais diversas áreas. Isso possibilita um aumento no número de mundos virtuais disponíveis, despertando um maior interesse nos usuários do sistema. Para viabilizar a publicação de novos mundos virtuais, é preciso que o sistema inclua mecanismos para persistência dos estados dos ambientes virtuais, bem como instanciação dos mesmos na rede.

A seguir, são apresentados os componentes da arquitetura CRAbCVE e discutido como esses componentes são empregados na disponibilização de mundos virtuais

compartilhados. Esta arquitetura, ilustrada na Figura 6.1, é composta de oito componentes cujas funcionalidades estão descritas a seguir.

6.21 Gerenciador de Políticas do Sistema

O Gerenciador de Políticas do Sistema (GPS) é o componente responsável pelo cadastro de usuários e mundos virtuais, além das permissões de uso e edição nestes ambientes virtuais. Para isso, o GPS mantém uma base de dados com conjuntos de direitos de acesso detalhados por usuários e mundos virtuais.

Assim, para ter acesso ao sistema, todos os participantes precisam, antes de tudo, autenticar-se no GPS. Por agrupar funções de localização e autenticação, este componente é o ponto de convergência do sistema. Assim, deve existir uma única instância do GPS ativa no sistema, ao contrário dos outros componentes que podem ter várias instâncias.

6.2.2 Fornecedor de Mídias

O Fornecedor de Mídias (FM) torna disponível aos participantes do CVE o conjunto de arquivos utilizados em interações no interior do ambiente. O FM é bastante simples, funcionando basicamente como um sistema de armazenamento de arquivos de mídias específicas. Entre essas mídias, estão presentes todos os elementos tridimensionais do ambiente virtual, incluindo sua própria estrutura arquitetônica, imagens, sons, vídeos, etc. O FM armazena ainda arquivos que descrevem a formação dos mundos virtuais e artefatos, e listas de permissões dos elementos RV em cada mundo virtual.

6.2.3 Gerenciador de Eventos

O Gerenciador de Eventos (GE) é o componente responsável por iniciar o funcionamento do ambiente virtual e por tratar todos os eventos oriundos das interações de seus participantes, gerando, sempre que necessário, mensagens de atualização do estado do mundo virtual que são distribuídas aos participantes através do Roteador de Mensagens.

Para o tratamento desses eventos, o Gerenciador de Eventos armazena o estado do mundo virtual e de todas as suas entidades. Isso permite que o estado atual seja facilmente informado aos usuários recém-conectados ao ambiente virtual. Para otimizar o acesso ao estado do mundo virtual, adotou-se uma estrutura de banco de dados que concentra todas as informações do mundo virtual.

6.2.4 Gerenciador de Políticas do Mundo Virtual

O Gerenciador de Políticas do Mundo Virtual (GPMV) é o componente especializado no armazenamento das restrições de ações de participantes do CVE. O conjunto de direitos de acesso permitirá ou não que um usuário interaja com determinados artefatos do CVE, ou tenha acesso a determinados sub-ambientes. As restrições podem ser atribuídas diretamente pela equipe de administração, ou pelos próprios participantes durante a autoria do ambiente virtual. Junto ao Roteador de Mensagens e ao Gerenciador de Eventos, o GMPV é um dos três componentes essenciais para o funcionamento de um CVE.

6.2.5 Roteador de Mensagens

O Roteador de Mensagens (RM) é o componente responsável por tornar as conexões de um determinado AVR disponíveis aos seus diversos participantes (humanos ou simuladores). Esse componente é empregado no repasse de informações a outros elementos da arquitetura, a fim de manter o estado compartilhado do mundo virtual (posicionamento e velocidade de deslocamento das entidades, por exemplo) e viabilizar a comunicação entre participantes (mensagens de texto, requisição de mídias, etc.).

Juntos, os componentes RM, GE e GPMV fazem o processamento de um mundo virtual. É possível ainda o uso destes componentes para processamento de duas ou mais instâncias de mundos virtuais.

6.2.6 Cliente de Interação e Edição em Realidade Virtual

Os recursos do ambiente virtual estão disponíveis para os usuários humanos através Cliente de Interação e Edição em Realidade Virtual (CIERV), que se conecta diretamente ao RM. Essa aplicação cliente faz uso de diferentes ferramentas necessárias à apresentação dos diversos tipos de mídias disponíveis no ambiente virtual e uma interface gráfica tridimensional para apresentação desse mundo virtual. O CIERV também torna os recursos de comunicação disponíveis aos usuários.

O CIERV mantém uma *cache* local de mídias para o armazenamento persistente de todas as mídias utilizadas, inclusive a própria arquitetura tridimensional do ambiente virtual. Essas mídias são replicadas sob demanda através de uma conexão direta com o componente FM.

6.2.7 Controlador de Simulóides

Personagens virtuais controlados pelo computador podem desempenhar diversos papéis dentro do ambiente virtual, inclusive o de interface de acesso a recursos originalmente externos ao AVR. O componente Controlador de Simulóide (CS) deve conectar-se diretamente ao Roteador de Mensagens, da mesma maneira como um CIERV se conecta. Um modelo de construção detalhado desse componente encontra-se em (Santos, 2001; Vidal et al., 2000).

6.2.8 Cliente de Administração

O Cliente de Administração (CA) é a uma ferramenta utilizada para realização de diversas tarefas relacionadas à manutenção simplificada da arquitetura. Esse cliente atua conectando-se diretamente a cada componente definido na arquitetura ou a controladores de simulóides. Uma vez conectado, o Cliente de Administração tanto pode realizar diversas alterações nas configurações do componente, quanto acessar as informações armazenadas por esse, a fim de realizar análise de funcionamento do sistema.

6.3 Integração do NuVEM à arquitetura CRAbCVE

Atualmente, um sistema baseado na arquitetura CRAbCVE está implementado (Gomes, 2005). Nesse sistema, os componentes CS e CA não estão presentes e o componente GPMV foi implementado de modo rudimentar como um módulo do componente GE. No entanto, nesta versão os componentes GE e RM não foram devidamente e completamente implementados, sendo utilizados apenas com parâmetros fixos, o que impossibilita a alocação dinâmica de mundos virtuais. Para contornar esse problema foi implementado um emulador de mundo virtual que será utilizado durante a fase de testes (seção 6.5).

A fim de possibilitar a gerência de falhas dos mundos virtuais disponibilizados através da arquitetura CRAbCVE, foram-lhe incorporados os componentes do modelo NuVEM discutidos no capítulo anterior.

Para implementação dos componentes de gerência foi utilizada a tecnologia JMX descrita no capítulo 2. Esta escolha é justificada pelo fato de que JMX é uma tecnologia aberta e expansível. Apesar de também ser uma ótima alternativa, o WBEM foi descartado por conta da dificuldade na utilização dos provedores de acesso a dados do sistema operacional (Job & Simões, 2002).

A seguir, são apresentados os passos seguidos para integração dos componentes de gerência aos componentes do sistema AVR escolhido para o estudo de caso.

6.3.1 Integração do Gerente do sistema

Para ter acesso aos ambientes virtuais do sistema CRAbCVE, todos os participantes precisam se autenticar antes junto ao componente GPS (descrito na seção 6.3.1), que pode ser considerado um componente centralizador. Logo, só deve existir uma única instância do GPS no sistema, ao contrário dos outros componentes, que podem ter várias instâncias (Gomes, 2005). Além disso, o GPS é responsável pelo controle e administração de permissões dos usuários e mundos virtuais, sendo o ponto de convergência de todo o sistema. Por outro lado, o Gerente do sistema é o responsável pela catalogação dos ambientes virtuais instanciados.

Foi então criada uma interface de comunicação entre o GPS e o Gerente (ver seção 5.3.3), possibilitando a troca de comandos relativos aos mundos virtuais instanciados. Dentre esses comandos destacam-se também requisições de instanciação de mundos virtuais, monitoramento e controle. Portanto, o GPS repassará as requisições de instanciação para o Gerente, que aplicará o mecanismo de instanciação remota introduzido na seção 5.2.2 (Figura 6.2).

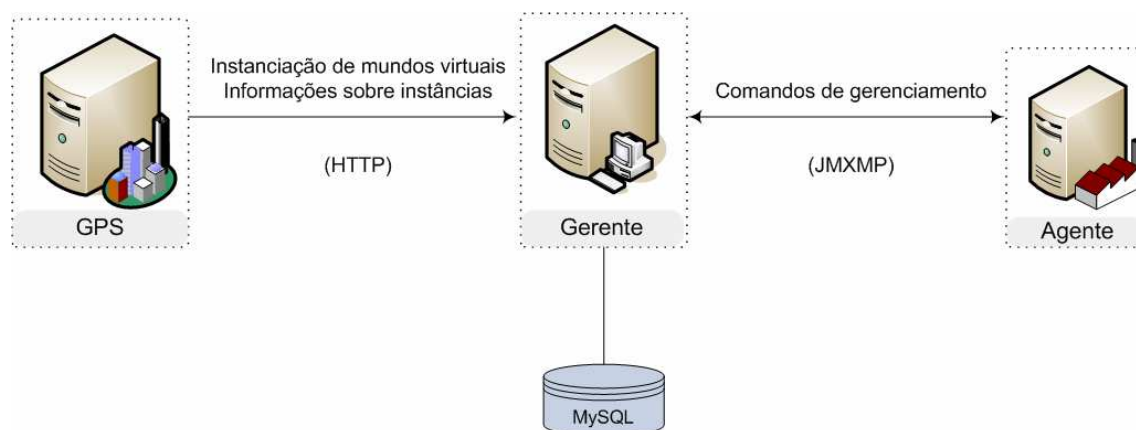


Figura 6.2. Integração do Gerente ao componente GPS.

Durante a implementação do Gerente foram utilizados os protocolos HTTP com suporte a autenticação e JMXMP (Sun Microsystems, 1999). Para comunicação entre o Gerente e o GPS foi usado o HTTP, que também fornece uma interface amigável através de navegadores *web*. Já o JMXMP foi usado para comunicação do Gerente com os Agentes do sistema (Figura 6.2).

A aplicação Gerente demanda a implantação de uma base de dados para armazenamento de informações relativas ao sistema de gerência como um todo. Logo, foi

utilizado o banco de dados MySQL (Widenius & Axmark, 2002) devido à sua simplicidade, gratuidade e velocidade (Figura 6.2). Maiores detalhes sobre a implementação do Gerente serão vistos na seção 6.4.3.

6.3.2 Integração do Agente do sistema

Conforme dito anteriormente, o gerenciamento do sistema de ambientes virtuais consiste no gerenciamento do ciclo de vida dos mundos virtuais instanciados. Com base nisso, os componentes RM, GE e GPMV (descritos na seção 6.3) devem ter seus processamentos controlados pelo sistema de gerência.

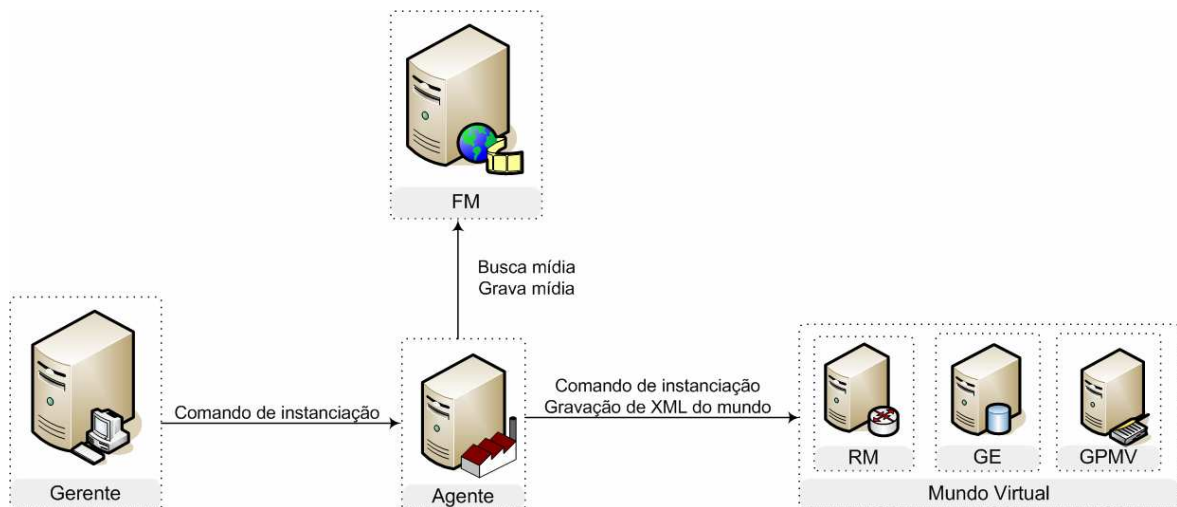


Figura 6.3. Integração do Agente aos componentes do sistema CRAbCVE.

Em (Almendra et. al, 2003) é descrito um mecanismo de fábricas para mundos virtuais usando a tecnologia CORBA (OMG, 1995) - no entanto, o uso desta tecnologia dificulta bastante a integração a outras tecnologias ou serviços, como portais *web*, por exemplo. Desse modo, para o desenvolvimento do Agente foi utilizado o conceito de Fábrica de Mundos descrito em (Almendra et. al, 2003). A Fábrica foi implementada como um módulo do Agente, que expõe ao Gerente uma interface para instanciação de mundos virtuais. Além disso, cada Agente deve manter um registro local de seus recursos computacionais utilizados. Estes dados serão necessários durante o processo de instanciação remota (descrito na seção 5.2.2), onde será escolhido o servidor com maior aptidão para processar uma nova instância de mundo virtual. Apesar de não fazer parte do escopo deste trabalho, foi idealizada uma métrica simples para esta escolha utilizando apenas dados como memória e processamento utilizados pelo sistema hospedeiro.

De acordo com especificação da arquitetura CRAbCVE, o componente FM (descrito em 6.3.2) é responsável pela manutenção de todas as mídias utilizadas nos ambientes virtuais do CRAbCVE, inclusive mídias de estado dos mundos virtuais, que

consistem em arquivos XML contendo informações relativas à estrutura e às permissões de usuários específicas para cada mundo virtual. O XML do mundo virtual é processado pelo componente GE (descrito na seção 6.3.3), que recebe eventos decorrentes das interações no mundo virtual, armazenando o estado do mundo virtual e de todas as suas entidades neste XML.

Para que as Fábricas tenha acesso às informações de mídias dos mundos virtuais é preciso que haja uma interface de comunicação entre o FM e os Agentes (ver seção 5.3.3). Cada Agente mantém uma referência para o FM para uso sempre que o Gerente enviar um comando de instanciação remota (Figura 6.3). Assim, de posse do XML do mundo virtual, a Fábrica poderá criar localmente a nova instância.

Ademais, por ser um componente para persistência de mídias, o FM foi escolhido para incorporar as funções do Servidor compartilhado (descrito na seção 5.3). Logo, periodicamente os Agentes poderão usar o componente FM como servidor para compartilhar os estados intermediários de suas instâncias de mundos virtuais. Para que o Agente tenha acesso aos estados intermediários dos mundos virtuais, faz-se necessária a gravação periódica do arquivo XML pelo componente GE.

6.3.4 Detalhes de implementação

A implementação mais recente da arquitetura CRAbCVE usa a tecnologia C#.NET *web services*¹³ para o codificação do GPS e do FM, enquanto CIERV, RM e GE estão codificados em C/C++. Para comunicação entre o GPS e o Gerente foi usado o *JMXConnector* HTTP com suporte a autenticação Basic. Para comunicação entre o Gerente e o Agente ambos utilizaram o *JMXConnector* JMXMP, que é um protocolo simples e *firewall-friendly*. Por fim, para comunicação entre o Agente e o FM foram usados os pacotes Java para SOAP¹⁴ (Figura 6.4).

Conforme citado anteriormente, foi utilizada também uma base de dados MySQL para persistência de dados do sistema de gerenciamento. A Figura 6.4 ilustra a nova disposição dos componentes do sistema CRAbCVE depois da integração dos componentes de gerência, bem como os protocolos de comunicação utilizados.

¹³ A tecnologia de *web services* consiste em uma coleção de protocolos e padrões usados para a troca de informações entre aplicações ou sistemas. Aplicações desenvolvidas em várias linguagens de programação e rodando em diferentes plataformas podem trocar dados através de uma rede de computadores como a Internet de maneira similar a comunicação entre processos de um computador comum.

¹⁴ SOAP (*Simple Object Access Protocol*) é um padrão para troca de mensagens baseadas em XML de um sistema distribuído em uma rede de computadores, normalmente utilizando o protocolo HTTP (Irani & Basha, 2001; Box et. al, 2000).

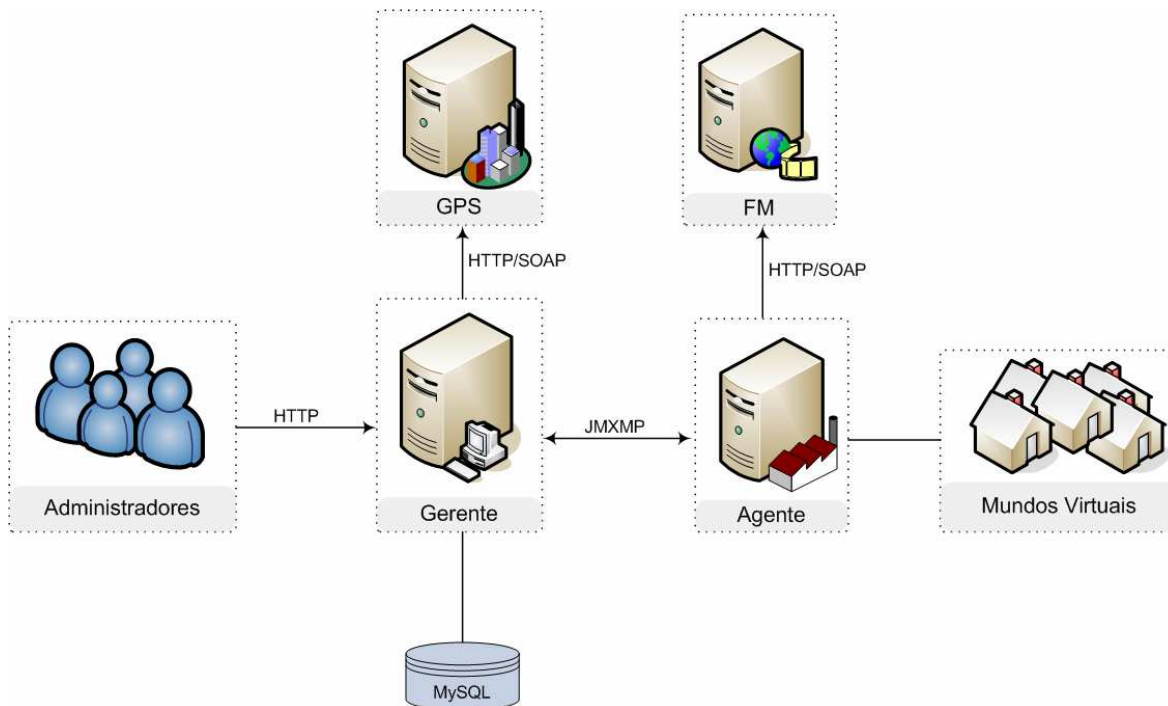


Figura 6.4. Integração dos componentes do NuVEM ao sistema CRAbCVE.

Visando uma maior portabilidade e maior facilidade no momento de integração com outros sistemas, a aplicação Gerente deve possuir também uma interface multi-plataforma e de fácil acesso. É preciso que esta interface contemple vários níveis de abstração de detalhes técnicos, não exigindo conhecimentos específicos de informática. Com isso, detalhes como a localização física dos espaços virtuais, portas de comunicação, dados estatísticos, etc., usualmente devem ser ocultados.

Para implementação desta interface multi-plataforma foi utilizado o componente *JMX HtmlAdaptorServer*, que é um conector para protocolo HTTP (Figura 6.4). Através deste conector é possível criar manipuladores (*parsers*) HTML para exibição de páginas *web* (Figura 6.5). Além da interface *web*, os administradores do sistema de gerência dispõem também da interface da própria aplicação Java (Figura 6.5).

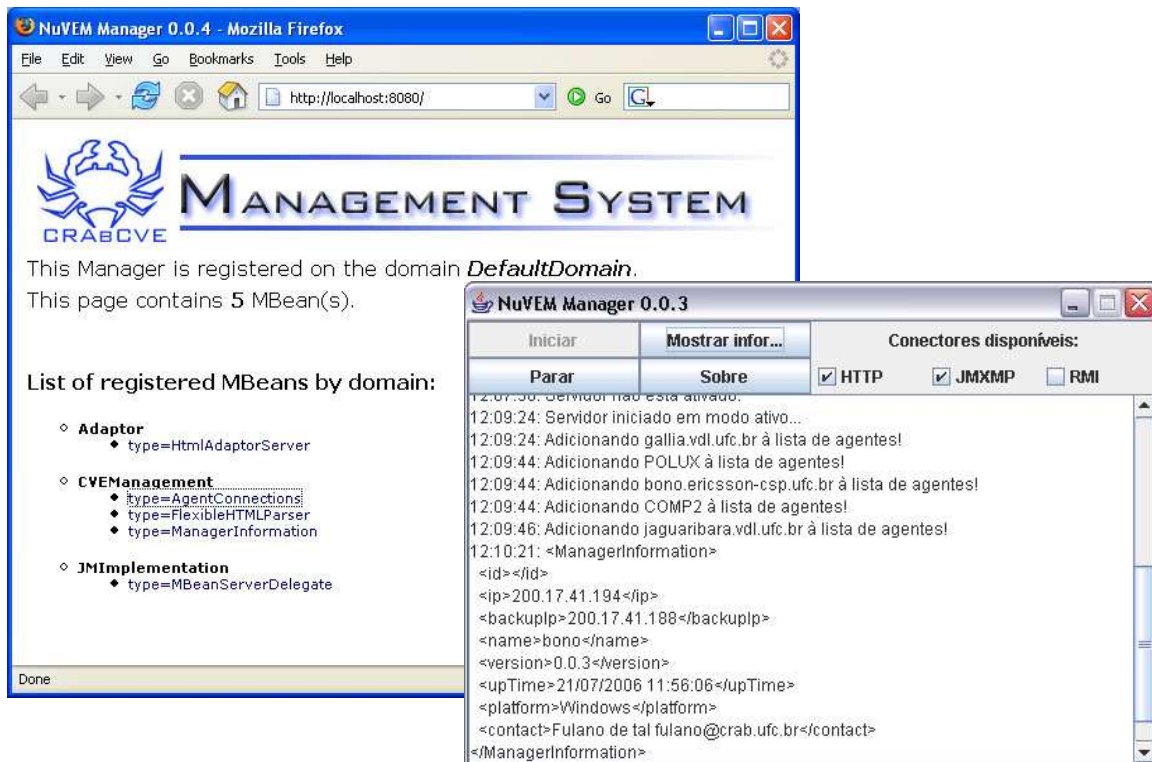


Figura 6.5. Interfaces *web* e Java do Gerente.

Para implementar cada um dos aspectos gerenciados pelo sistema foi utilizado o próprio paradigma do JMX, o *MBean* (descrito na seção 2.5.3). Dentre os objetos gerenciados, destacam-se os mundos virtuais, as fábricas e as informações de sistema de cada Agente.

Para representar os mundos virtuais em cada Agente, é implementada a interface *WorldInstanceMBean*, que define um conjunto de métodos para iniciar e parar o processamento do mundo virtual, bem como modificar parâmetros de conexões (Figura 6.6a). Cada instância de mundo virtual tem um agendamento correspondente, onde são definidas as datas inicial e final nas quais a instância estará disponível.

Além do *WorldInstanceMBean*, o Agente implementa também um *MBean* para monitoramento de informações do sistema hospedeiro, o *SystemInformationMBean*. Este *MBean* expõe vários parâmetros relativos à máquina servidora, como memória em uso, endereço IP, uso médio de CPU, dentre outros (Figura 6.6b). Estas informações de sistema são coletadas pelo Agente por meio de JNI¹⁵ (código da interface JNI no Apêndice A).

¹⁵ O JNI é uma interface de comunicação (API) desenvolvida pela Sun que permitem aos códigos executados pela VM (Java Virtual Machine) interoperarem de forma bidirecional com aplicações e bibliotecas desenvolvidas de outras linguagens, como C/C++, assembly ou Delphi (Gordon, 1998).

<pre>public interface WorldInstanceMBean { public int getNumUsers(); public String getUpTime(); public String getScheduleId(); public String getWorldId(); public String getGEAddress(); public String getGEPort(); public String getRMAddress(); public String getRMPort(); public void stop(); public void start(); public boolean isRunning(); }</pre> <p>a) WorldInstanceMBean interface.</p>	<pre>public interface SystemInformationMBean { public String getVersion(); public String getContact(); public String getName(); public String getPlatform(); public float getMeanCPUUse(); public float getPhysicalTotalMemory(); public float getNotUsingPhysicalMemory(); public float getUsingPhysicalMemory(); public float getUsingPhysicalMemoryRatio(); public String getIp(); public String getUpTime(); }</pre> <p>b) SystemInformationMBean interface.</p>
---	--

Figura 6.6. MBeans implementados pelo Agente.

Para administrar os mundos virtuais contidos na arquitetura, o Gerente implementa o *AgentConnectionsMBean* (Figura 6.7), onde são catalogadas todas as informações relativas aos Agentes e mundos virtuais instanciados no sistema. Este *MBean* agrupa funções relativas às Fábricas contidas em cada Agente. Portanto, o *AgentConnectionsMBean* define métodos para criação, destruição e agendamento de instâncias.

<pre>public interface AgentConnectionsMBean { public String[] getAgentIPList(); public String getWorldInstancesListByAgent(String ipAddr); public String getSystemInformationByAgent(String ipAddr); public String addWorldInstance(String idWorld); public String addScheduledWorldInstance(String idWorld, String dtEnd); public String addScheduledWorldInstance(String idWorld, String dtStart, String dtEnd); public String updScheduledWorldInstance(String idSched, String dtStart, String dtEnd); public String delScheduledWorldInstance(String idSched); public String delWorldInstance(String idWorld, String idSched); }</pre>

Figura 6.7. Interface AgentConnectionsMBean.

O processo de instanciação remota utiliza as informações contidas nos objetos *SystemInformationMBean* de cada Agente. Cada vez que o for requisitada uma instanciação, o Gerente busca o Agente com mais recursos livres para processar a nova instância de mundo. Foi especificada uma métrica simples para calcular qual dos agentes deve processar a nova instância. Assim, esta escolha leva em consideração somente o uso médio de CPU e memória RAM utilizados no servidor hospedeiro (Figura 6.8). Apesar de rudimentar, essa métrica é suficiente para testar a funcionalidade de instanciação remota de modo a garantir o balanceamento de cargas nos servidores.

$\text{Agente} = \min(\text{CPU} + \text{MEM})$ <p> CPU = Percentual médio de uso de CPU. MEM = Percentual médio de uso de memória RAM. </p>

Figura 6.8. Métrica simples para Instanciação remota.

O *framework* JMX não define em sua API mecanismos para implementação de gerentes remotos. Isto é, não existem classes específicas para criação de aplicações gerentes. Foi necessária a implementação de conexões cliente e servidor tanto no Gerente como no Agente.

Inicialmente, o Gerente é carregado e inicia um *socket* servidor para esperar conexões. A partir daí, o Agente abre um *socket* cliente e requisita sua entrada no sistema de gerência. A autenticação é feita com base em um ticket estabelecido previamente pelo administrador do sistema. Esta conexão permanece mesmo depois do término da autenticação (esta conexão constitui uma ligação necessária para que seja detectada pelo Agente uma possível falha no Gerente). Caso a autenticação seja bem sucedida, o Gerente estabelece uma conexão com o Agente, catalogando suas características de sistema na base MySQL. É por meio desta segunda conexão que o Gerente recebe notificações do Agente. Desse modo, existe um par de conexões para cada Agente do sistema. Desse modo, é notável que para cada Agente no sistema é estabelecido um par de conexões JMX.

A funcionalidade de tolerância a falhas é garantida pelo uso da base de dados para guardar informações a respeito das instâncias e agentes ativos em um dado momento. É usado o componente FM como servidor compartilhado entre os Agentes para guardar os estados intermediários das instâncias. Caso haja falha em algum Agente ou mesmo em alguma instância, uma notificação de conexão - *JMXConnectionNotification* - é enviada automaticamente para o Gerente. A partir daí, o Gerente deverá re-instanciar cada um dos mundos afetados pela falha.

Alternativamente, caso o Gerente falhe, é disparada (em forma de exceção) em todos os Agentes uma notificação de conexão (*JMXConnectionNotification*). Desse modo, cada Agente deverá autenticar-se novamente com o Gerente reserva previamente carregado no sistema. As informações necessárias para conexão com o Gerente reserva foram obtidas durante a fase de autenticação com o Gerente titular. Ainda, estes dados de conexão podem ser modificados a qualquer momento. Logo, é preciso que o Gerente envie estas modificações de modo automático a todos os Agentes os quais lhe são subordinados.

6.4 Testes de funcionalidade

Nesta seção são demonstrados alguns dos resultados obtidos através de testes com o sistema resultante da integração (Figura 6.4). Para propósitos de testes, foram usadas seis estações servidoras, onde duas continham Gerentes (titular e reserva) e cinco continham

Agentes. Além disso, os webservices GPS e FM permaneceram em outra rede executando normalmente.

No entanto, na versão mais recente da arquitetura CRAbCVE os componentes RM e GE não estão completamente e devidamente implementados. Nesta versão, os componentes RM e GE não recebem parâmetros relativos a novas instâncias em tempo de execução, o que impossibilita a instanciação remota de mundos virtuais. Porém, esta carência foi suprida por meio de uma aplicação emuladora (Figura 6.9) que faz a função de carregar o arquivo descritor do mundo virtual (arquivo XML) e, em períodos de tempo fixo, descarregar modificações estruturais ocorridas no mundo virtual de volta para o arquivo descritor. Ainda, também em períodos fixos de tempo, o arquivo é gravado de volta no servidor compartilhado (FM) pelo Agente. A implementação desta aplicação serviu para validar os conceitos referentes às aplicações GE e RM: instanciação remota e gravação periódica do arquivo descritor do mundo virtual. As funcionalidades implementadas nesta aplicação emuladora consistem apenas em funcionalidades necessárias a qualquer arquitetura de ambientes virtuais que deseje utilizar o NuVEM.

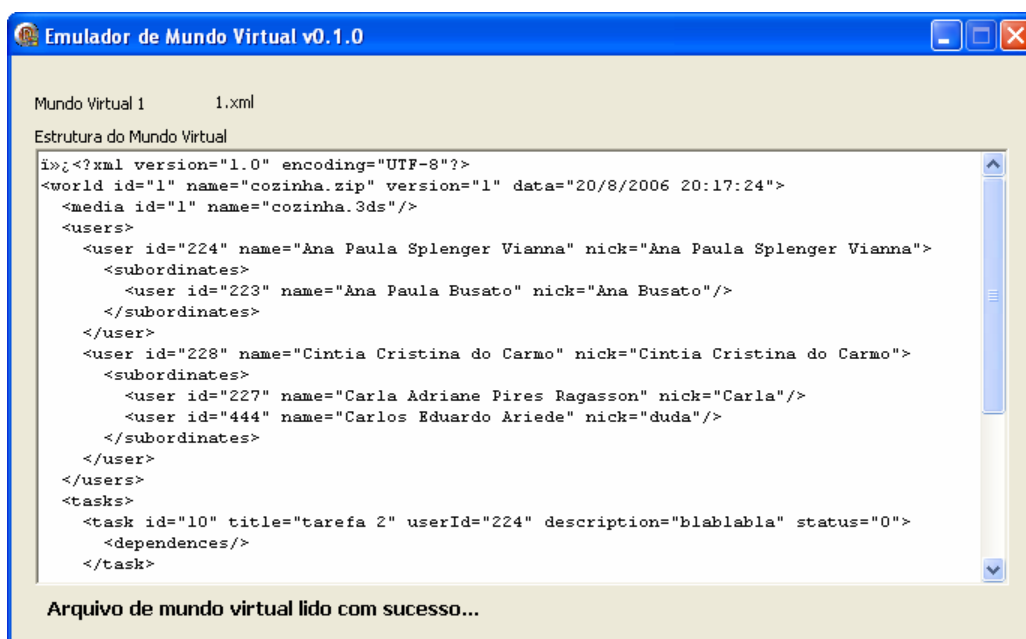


Figura 6.9 Aplicação emuladora de mundo virtual.

Foram utilizadas estratégias de injeção falhas em pontos do sistema de modo a validar os cenários expostos na seção 5.3.1. O diagrama de seqüência correspondente aos testes que se sucedem está, na íntegra, no Apêndice B.

Inicialmente, foram instanciados os 2 Gerentes (cenário 1). Sempre que é iniciado, o Gerente fica em modo passivo, i.e., à espera de conexões (Figura 6.10). A partir daí, são iniciados os 4 Agentes (cenário 2) em estações distintas de uma rede. Ao ser iniciado, cada

Agente tratou de se conectar ao Gerente titular previamente configurado (Figura 6.11). Ao longo de alguns minutos, mundos virtuais foram instanciados (cenário 3).

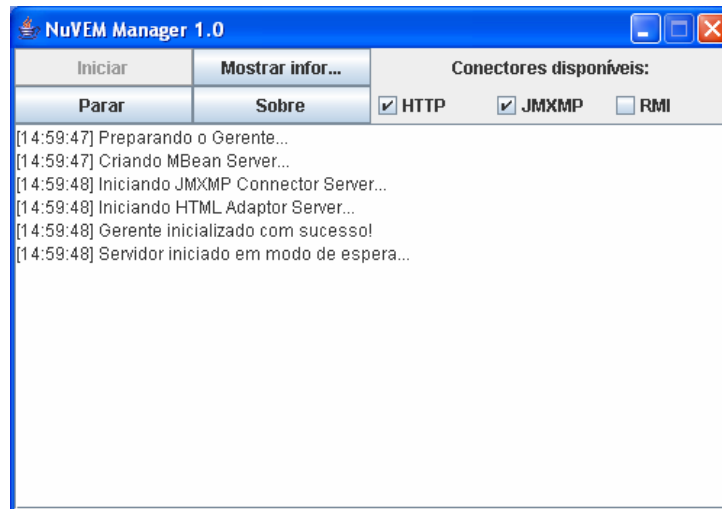


Figura 6.10. Gerente iniciado em modo passivo (espera).

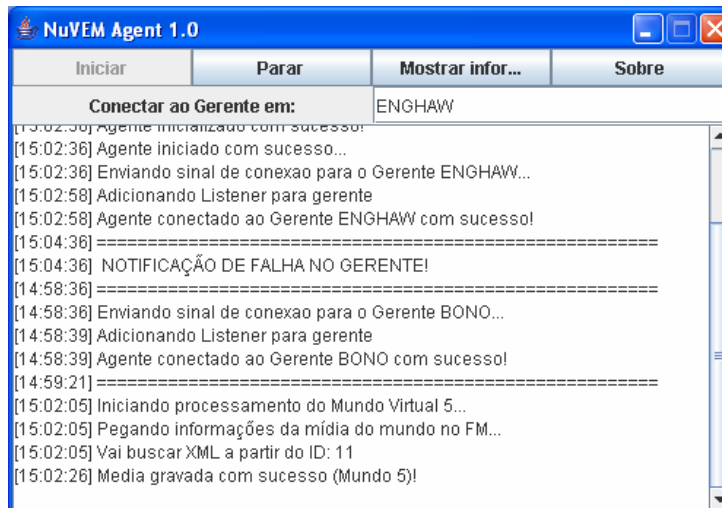


Figura 6.11. Agente conectando ao Gerente BONO depois de receber notificação de falha no Gerente ENGHAW. Em seguida, o Agente recebe uma instância de Mundo Virtual 5.

Foi injetada uma falha no Gerente titular de modo que ele parasse imediatamente. Durante um período arbitrário de 30 segundos os Agentes tentaram re-estabelecer conexão com o Gerente titular. Passados os 30 segundos, os Agentes confirmaram a falha na conexão e conectaram imediatamente ao Gerente reserva (cenário 7). Apesar de tantos pedidos simultâneos de conexão, o Gerente reserva conseguiu autenticar todos os Agentes e todos os seus dados foram coletados corretamente (Figura 6.12). Foi percebido que, no momento em que o Gerente recebia muitas conexões simultaneamente a aplicação travava por alguns instantes e, possivelmente para um número de maior de conexões, o Gerente poderia travar indefinidamente. Foi testado então o uso de uma diretiva do Java, a *synchronized* (Gosling et al., 1996), para os métodos nas classes envolvidas no processo de

autenticação do Agente. Esta diretiva evita que dois ou mais métodos sejam executados de maneira concorrente, enfileirando requisições, fazendo com que as chamadas sejam executadas de modo completamente seqüencial.

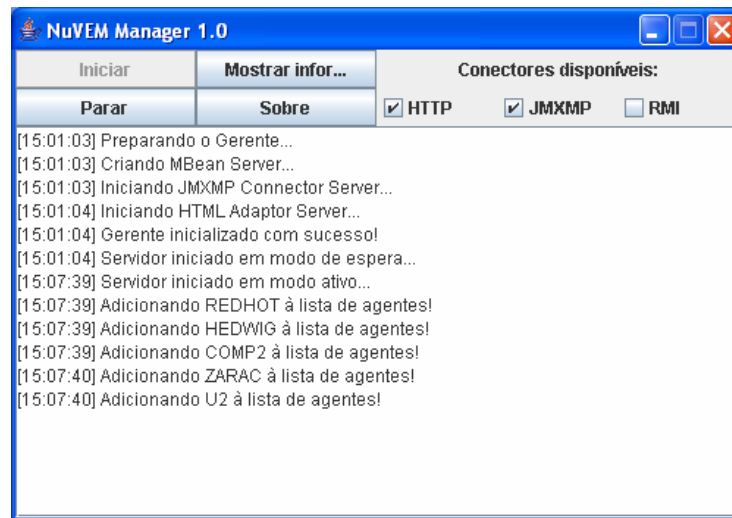


Figura 6.12. Agentes conectando ao Gerente reserva.

Foi injetada uma falha na aplicação emuladora, que parava diante de um comando pré-estabelecido. Esta falha serviu para testar a tolerância a falhas das instâncias de mundos virtuais (Figura 2.1). Desse modo, cada vez que um mundo falhava, o Agente enviava uma notificação ao Gerente, que re-instanciava o mundo virtual, descartando a instância com problemas (cenário 5).

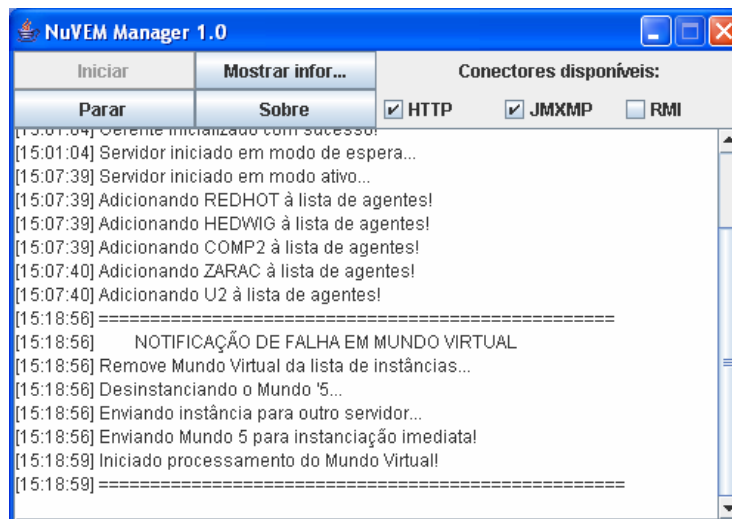


Figura 6.13. Gerente recebendo notificação de falha em mundo virtual e repassando para nova instanciação.

Foi configurado para que os agentes gravassem no Servidor compartilhado os estados dos mundos virtuais (arquivo XML) em períodos fixos arbitrários de 20 segundos (cenário 4). Com isso, as modificações resultantes das interações nos mundos virtuais só seriam perdidas se ocorridas, no máximo, nos últimos 20 segundos. Para que o Agente dispusesse sempre da versão mais recente do arquivo descritor, a aplicação emuladora

também foi configurada para persistir o arquivo XML sempre em períodos fixos arbitrários de 20 segundos. Vale ressaltar que o valor de 20 segundos foi estipulado apenas para efeitos de testes e pode ser alterado a qualquer momento nas configurações de sistema da aplicação Agente.

Foi injetada uma falha na estação servidora de um Agente de modo que a mesma se parou (cenário 6). Com isso, o Gerente recebia imediatamente uma notificação de falha de comunicação. Durante um período arbitrário de 30 segundos o Gerente tentou re-estabelecer conexão com o Agente. Após os 30 segundos o Gerente confirmou a falha e buscou em sua base local a lista de instâncias alocadas pelo Agente. De posse dessa lista, o Gerente re-instanciou cada mundo virtual de acordo com o procedimento de Instanciação remota (Figura 6.14).

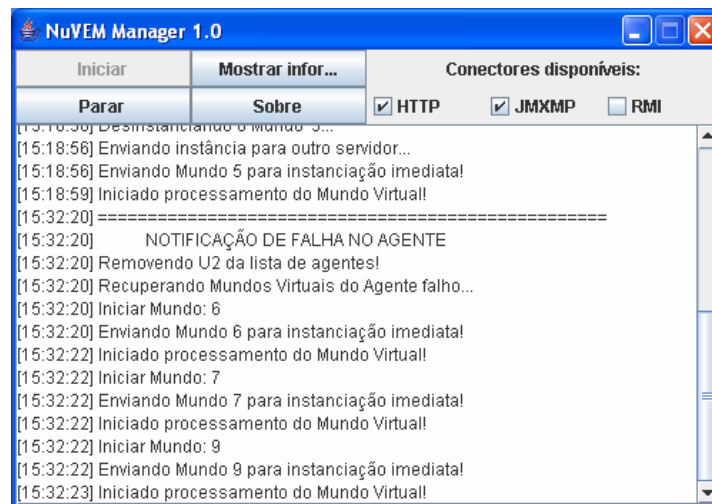


Figura 6.14. Gerente recuperando mundos virtuais após falha em Agente.

Por fim, é preciso salientar que quanto maior a frequência em que o arquivo XML do mundo virtual é atualizado, menor serão as perdas em termos de modificações realizadas no mesmo devido a falhas. Por outro lado, com uma alta frequência de atualizações pode gerar situações de sobrecarga de mensagem na rede (ainda mais para o caso de os pacotes conterem arquivos grandes). Assim, para testes com quantidades maiores de Agentes e mundos virtuais instanciados é altamente recomendável que sejam utilizados períodos maiores que os utilizados nos testes supracitados.

Capítulo 7

Conclusões e Trabalhos Futuros

Gerenciamento de Redes é uma área já bem estudada e bastante desenvolvida. Porém, nos últimos anos, a aplicação de técnicas de gerenciamento de aplicações distribuídas e, em especial, tolerância a falhas em sistemas distribuídos ainda têm se restringido a sistemas de ultra confiabilidade, como computadores de bordo e controle de plantas nucleares, e a sistemas de alta disponibilidade, como centrais telefônicas e servidores *web* (*webfarms*). No entanto, o crescimento das redes de computadores, fornecendo os mais variados serviços, aumentou a dependência tecnológica de uma grande parcela da população nos serviços oferecidos. Falhas nesses serviços podem ser catastróficas para a segurança da população ou para a imagem e reputação de empresas.

Aplicações distribuídas em redes de computadores devem ser mais confiáveis na medida em que se tornam mais populares e que mais pessoas utilizam seus recursos em tarefas do dia-a-dia. Logo, é necessária a utilização de mecanismos para lidar com problemas que possam afetar o funcionamento destes sistemas.

Sistemas AVR fornecem condições adequadas para que os usuários experimentem sentimentos de imersão e presença dentro de ambientes virtuais e possam interagir com outros usuários e com elementos do próprio ambiente.

Aplicar mecanismos de gerenciamento de falhas a sistemas de ambientes virtuais em redes mostra-se como uma tarefa bastante complexa, mas que pode tornar possível o aumento da disponibilidade dos recursos do sistema. O desenvolvimento de um modelo que viabilize o controle sobre estas falhas constituiu o objetivo principal dessa dissertação.

Para atingir esse objetivo, foram executadas análises de algumas aplicações de Ambientes Virtuais em Redes, levando em consideração detalhes sobre suas implementações e requisitos. Também foi realizado um levantamento sobre estratégias para gerenciamento em redes e tolerância a falhas.

A partir de todas as informações obtidas, foi proposto o modelo NuVEM – modelo para gerenciamento de uma arquitetura de ambientes virtuais. Para comprovar a viabilidade do modelo foi desenvolvido um estudo de caso onde o NuVEM foi integrado a uma arquitetura de AVR, a CRAbCVE.

O estudo de caso do NuVEM integrado à arquitetura CRAbCVE possibilitou que vários mecanismos de gerência de sistemas AVR fossem testados. Após uma série de testes, ficou evidenciado que ainda que ocorram falhas em ambientes virtuais é possível recuperar de modo efetivo os dados referentes às interações já realizadas. Além disso, durante os testes foi constatado que os mecanismos de resolução de falha implementados poderiam recuperar mundos virtuais em cerca de 10 segundos, dependendo da velocidade de processamento do servidor que aloca o Agente.

7.1 Principais contribuições

A principal contribuição desse trabalho é a definição de um modelo genérico, o NuVEM, projetado para viabilizar o gerenciamento de falhas em uma arquitetura de AVR. Baseado em conceitos de Gerenciamento de Redes, o NuVEM fornece um arcabouço para gerência remota de uma arquitetura de AVR.

Para a concepção do NuVEM foram consideradas as mais variadas funções de gerência de redes, como configuração, desempenho, segurança, falhas e contabilização de acessos (Capítulo 2). No entanto, para delimitar melhor o escopo do trabalho foi definido que o modelo contemplaria, inicialmente, apenas mecanismos para gerência remota de falhas em tempo real. Assim, foram definidas estratégias para detecção e resolução de falhas em determinados pontos do sistema, bem como administração remota de recursos. Utilizando-se deste modelo de gerenciamento, inúmeras arquiteturas de AVR podem ser gerenciadas de modo a adquirirem maior disponibilidade e confiabilidade.

Por fornecer estratégias que levem em consideração aspectos que interferem na escalabilidade, disponibilidade e confiabilidade do sistema, o NuVEM representa uma ótima solução para otimizar o uso de recursos computacionais contidos em um sistema AVR, além de prover meios de controle e monitoração dos mesmos.

Por último, o protótipo desenvolvido como estudo de caso do NuVEM pôde ser utilizado com o objetivo de aumentar a qualidade de serviço de um sistema de ambientes virtuais com fins educativos, o CRAbCVE.

7.2 Trabalhos futuros

O gerenciamento de entidades dos ambientes virtuais está diretamente ligado ao gerenciamento de desempenho do sistema como um todo. As interações em tempo real definem um conjunto de processos (*threads*) que compõe a aplicação de realidade virtual. Cada processo pode ser representado por uma entidade contida no ambiente virtual –

usuários, objetos, simulóides, etc. (Singhal & Zyda, 1999). Logo, o gerenciamento do desempenho do sistema consiste no escalonamento da execução de cada um dos componentes envolvidos na interação. O projetista do sistema deve aplicar técnicas para balanceamento de cargas no sistema de modo a evitar pontos de consumo excessivo de recursos computacionais, distribuindo processamento no sistema.

Alguns sistemas AVR, como o VELVET (Oliveira & Georganas, 2002), utilizam mecanismos de particionamento dinâmico dos ambientes virtuais de modo a distribuir o processamento na rede. Aliado às técnicas compensatórias vistas no Capítulo 4, o particionamento dinâmico dos ambientes virtuais também poderia diminuir a quantidade de mensagens na rede, aumentando a escalabilidade do sistema como um todo.

Para que o sistema possa contabilizar e configurar recursos em tempo de execução, é preciso que seja refinada a base de informações gerenciadas. Dentre os principais parâmetros que devem ser configurados estão as características das entidades que denotam o crescimento do sistema, como usuários, objetos dinâmicos, etc., contidos nos ambientes virtuais. Para cada recurso a ser gerenciado, deve-se armazenar suas peculiaridades de acordo com os fatores de escalabilidade do sistema. Ademais, é preciso armazenar informações acerca de todos os componentes físicos do sistema, como capacidade de processamento e memória dos servidores, capacidade de transmissão de dados, etc. Estes atributos serão necessários durante a avaliação de recursos físicos (*hardware*) livres no sistema e até mesmo durante o planejamento de expansão física do sistema (adição de novos componentes de *hardware*).

Nos últimos anos, têm sido cada vez mais comuns os ataques a servidores nas grandes redes, o que tem feito com que o gerenciamento de segurança tenha um papel fundamental durante o projeto do sistema AVR. Em particular, é preciso levar em consideração os seguintes tópicos de segurança (Smed et al., 2001): proteger informações sensíveis (como números de cartões de crédito e configurações de sistema) e assegurar os requisitos de consistência do mundo virtual, evitando trapaças e situações inesperadas pelos usuários. É preciso adicionar ao modelo de gerência mecanismos que viabilizem a comunicação segura entre os componentes do sistema, possibilitando a confidencialidade, integridade e autenticidade de informações sensíveis. Assim, podem ser usadas técnicas de codificação das mensagens aliadas a uma estratégia de autenticação distribuída, onde são compartilhadas credenciais para autenticação de requisições de serviços providos pelo sistema AVR.

Por último, é previsto como trabalho futuro uma extensão do estudo de caso realizado no capítulo anterior. A extensão consiste na criação de uma interface web personalizada para integração do sistema AVR (já completamente integrado ao NuVEM) a um Ambiente Virtual de Aprendizado na *web* (AVA).

Referências Bibliográficas

- ALISON, D.; WILLS, B.; HODGES, F. & WINEMAN, J. Gorillas in the Bits. *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 69-76. 1997.
- ALMENDRA, C. C.; VIDAL, C. A.; LEITE-JUNIOR, A. J. M.; SANTOS, E.M. & CAMPÊLO, L. D. S. P. A Web System for Managing and Accessing NVEs Using CORBA. *Proceedings of the 6th Symposium on Virtual Reality*, Ribeirão Preto-SP, Brasil, p. 297-308. 2003.
- ALMENDRA, C. C. *Administração Remota de Ambientes Virtuais em Rede para Integração com Sistemas de Gerenciamento de Aprendizado*. Dissertação de Mestrado, Universidade Federal do Ceará, Brasil. 2003.
- BENFORD, S. D. & GREENHALGH, C. Collaborative Virtual Environments. *Proceedings of Communications of the ACM*, Vol. 44, No. 7 p. 79. ACM Press, 2001.
- BLANKE, M.; STAROSWIECKI, M. & WU, N. E. Concepts and Methods in Fault-tolerant Control. *Proceedings of the Invited tutorial lecture at American Control Conference*. Arlington, VA, Washington, USA, 22-24. 2001.
- BLAXXUN. *Technology and content solutions for Web 3D visualization, collaboration and community*. 2005.
- CAMPOS, M. L. & CAVALCANTI, M. C. R. Suporte por computador ao trabalho cooperativo. *Anais do XV Congresso Brasileiro da Sociedade Brasileira de Computação. XIV Jornada de Atualização em Informática*. Canela, RS. Agosto, 1995.
- BOX, D., EHNEBUSKE, D.; KAKIVAYA, G.; LAYMAN, A.; MENDELSON, N.; NIELSEN, H. F.; THATTE, S. & WINER, D. *Simple Object Access Protocol (SOAP) 1.1*. World Wide Web Consortium. 2000.
- BRUTZMAN, D. P.; MACEDONIA, M. R., & ZYDA, M. J. Internetwork Infrastructure Requirements for Virtual Environments. *Proceedings of the VRML'95VRML'95 Symposium*. San Diego, USA. 1995.
- BRYSON, S. & LEVIT, C. The Virtual Wind Tunnel. *IEEE Computer Graphics and Applications*, Vol. 2, No. 4. 1992.
- CALVIN, J.; DICKENS, A.; GAINES, B.; METZGER, P.; MILLER, D. & OWEN, D. The SIMNET Virtual World Architecture. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, p. 450-455. 1993
- ÇAPIN, T. K.; PANDZIC, I. S.; MAGNENAT-THALMAN, N. e THALMAN, D. *Avatars in Networked Virtual Environment*. John Wiley & Sons. 1999.
- CAPPS, M.; D. MCGREGOR; D. BRUTZMAN; & M. ZYDA. NPSNET-V: A New Beginning for Virtual Environments. *IEEE Computer Graphics & Applications*. 2000.
- CARDOSO, F.; ROCHA, F.; LIMA, L.; COSTA, M.; SOUZA T.; FERREIRA, L.; CARDOSO, A. & LAMOUNIER JR., E. Jogo: Fogo Amigo - Uso de Técnicas de Realidade Virtual no Desenvolvimento de Jogos Interativos. *Proceedings of the 6th Symposium on Virtual Reality*, p. 431, Ribeirão Preto, SP. 2003.
- CARL-MITCHELL, S. & QUARTERMAN, J. *Building Internet Firewalls - UnixWorld*, p. 93-102, Fevereiro de 1992. ISSN 0739-5922. 1992.

- CARLSSON, C. & HAGSAND, O. DIVE - a platform for multi-user virtual environments. *Computers & Graphics (Pergamon)*. Vol. 17, no. 6, pp. 663-669. 1993.
- CARVALHO, T. C. M. *Gerenciamento de redes - Uma Abordagem de Sistemas Abertos*. São Paulo, Makron Books, 1993.
- CASE, J.; MCCLOGHRIE, K.; ROSE, M., & WALDBUSSER, S. *Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1448. Internet Engineering Task Force, Network Working Group, 1993.
- CECIN, F. R.; BARBOSA, J. L. V.; GEYER, C. F. R. (2003) FreeMMG: An hybrid peer-to-peer, client-server, and distributed massively multiplayer game simulation model. *Proceedings of the 2nd Brazilian Workshop on Games and Digital Entertainment (WJogos'03/WJogos'03)*, 2003.
- CHESWICK, W.R. & BELLOVIN, S. *Firewalls and Internet Security*. Addison-Wesley. 1994.
- CHOI, M. J.; HONG, J. W. & JU, H. T. *XML-based Network Management for IP Networks*. ETRI Journal, Vol. 25, No. 6, pp. 445-263. 2003.
- CLARK, J. H. Hierarchical geometric models for visible surface algorithms. *The California Association of Community Managers (CACM)*, October, 1976.
- COMER, E. D. *Internetworking with TCPIP, Volume 1: Principles, Protocols, and Architecture*. PrenticeHall, Engelwood Clis, New Jersey, 1995.
- CRISTIAN, F. *Exception Handling and Tolerance of Software Faults*. John Wiley & Sons, p. 81-108, 1995.
- DAVIN, J. & McCloghie, K. *Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1445. Internet Engineering Task Force, Network Working Group, 1993.
- DODGE, M. Exploration in AlphaWorld: the geography of 3-D virtual worlds on the Internet. *iVirtual Reality in Geography*, RGS-IBG Conference, Leicester, 1999.
- EMMERICH, W. *Engineering Distributed Objects*. John Wiley & Sons, USA, 2000.
- FERREIRA, A. G. *Uma arquitetura para visualização distribuída de ambientes virtuais*. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Brasil, 1999.
- FREITAS, C. M. D. S.; MANSSOUR, I. H.; NEDEL, L. P.; GAVIÃO, J. K.; PAIM, T. C. & MACIEL, A. Framework para Construção de Pacientes Virtuais: Uma Aplicação em Laparoscopia. *Proceedings of the 6th Symposium on Virtual Reality*, p. 117-128, Ribeirão Preto, SP, October 15-18, 2003.
- GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J. *Design Patterns: Elements of Reusable Software*. Addison Wesley, USA, 1995.
- GOLDBERG, K. Virtual Reality in the Age of Telepresence. *Convergence: The Journal of Research into New Media Technologies*, vol.4, n.1, University of Luton Press, 1998.
- GOMES, G. A. M. *CRAbCVE -- Uma Arquitetura para Viabilização de caves através da Internet*. Dissertação de Mestrado, Universidade Federal do Ceará, 2005.
- GORDON. H. *Essential JNI: Java Native Interface*. Prentice Hall, USA, 1998.

- GOSLING, J.; BILL, J. G. & STEELE, L. *The Java™ Language Specification*. Addison-Wesley, 1996.
- GOSSWEILER, R.; LAFERRIERE, R. J.; KELLER, M. L.; PAUSCH, R. An Introductory Tutorial for Developing Multiuser Virtual Environments. *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4, 1994.
- GRAHAM, I. & QUIN, L. *XML Specification Guide*, John Wiley & Sons, USA, 1999.
- GUPTA, A.; STAHL, D. O. & WHINSTON, A. B., The Economics of Network Management, *Communications of the ACM*, p. 57-63, 1999.
- GUTMANN, S.; BURGARD, W.; FOX, D. & KONOLIGE, K. An experimental comparison of localization methods. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, EUA, 1998.
- HOSSEINI, M. & GEORGANAS, N. D. Collaborative virtual environments for training. *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, p. 621-622, New York, NY, USA, 2001.
- HYDE, D. *Web-Based Management: The New Paradigm for Network Management*. 3COM Technical Papers. Disponível em: <http://www.3com.de/pdf/50062701.pdf>. Acesso em: 12/08/2005.
- IRANI, R. & BASHA, S. J. *AXIS: The Next Generation of Java Soap, 1st edition*. Wrox Press Ltd., Birmingham, UK, 2002.
- JALOTE, P. (1994) *Fault tolerance in distributed systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- JOB, J. C. C. & SIMÕES, M. A. C. Gerenciamento de Sistemas Baseado no Padrão WBEM. *Revista Eletrônica de Iniciação Científica (SBC)*. Ano 2, volume 2, número 4. Dezembro de 2002.
- JOHNSON, A.; ROUSSOS, M.; LEIGH, J.; BARNES, C.; VASILAKIS, C. and MOHER, T. The NICE Project: Learning Together in a Virtual World. *Proceedings of IEEE Virtual Reality Annual International Symposium*, Atlanta, Georgia, 1998, pp. 176-183.
- JU, H. T.; CHOI, M. J.; HAN, S. H.; OH, Y.J.; YOON, J.H.; LEE, H. J. & HONG, J. W. K. An Embedded Web Server Architecture for XML-based Network Management. *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2002)*, p.1-14. Florence, Italy, April 2002.
- KALAWSKY, R. S. *Exploiting Virtual Reality Techniques in Education and Training: Technological Issues*. SIMA (Support Initiative for Multimedia Applications) Report Series. Disponível em <http://www.agocg.ac.uk/reports/virtual/vrtech/toc.htm>. Acesso em: 02/09/2005.
- KAWAMOTO, A. L. S.; Cantão, J.; Pinto, A.; Aquino, P. T.; Álvares, J. V.; Kirner, T. G., Kirner, C. & Wazlawick, R. S. AVC-MV: Um Ambiente Colaborativo para Aplicações Educacionais. *Proceedings of the 4o SBC Symposium on Virtual Reality*, p. 226-237. Florianópolis, 2001.
- KIRNER, T. G. & DAVIS, A. M. *Nonfunctional Requirements of Real-Time Systems. Advances in Computers (Zelkowitz, M., Ed.)*, Volume 42, pp. 1-37. Academic Press, 1996.
- KIRNER, T. G.; KIRNER, C.; KAWAMOTO, A. L. S.; CANTAO, J.; PINTO, A. & WAZLAWICK, R. S. Development of a Collaborative Virtual Environment for

- Educational Applications. *Proceedings of WEB3D 2001*, pp. 61-68. Paderborn, Germany, 2001.
- KON, F. & MARQUES, J. R. Gerenciamento de recursos distribuídos em sistemas de grande escala. *XX Simpósio Brasileiro de Redes de Computadores (SBRC2002)*, Búzios, RJ, Brasil, 2002.
- KREGER, H. *Java Management Extensions for application management*. IBM Systems Journal. Volume 40, No. 1, pp. 104-129. Disponível em: <http://www.research.ibm.com/journal/sj/401/kregeaut.html>. Acessado em: 17/08/2005.
- KUBO, M. M.; VICENTIN, V. J.; DERIGGI Jr., F. V. & KIRNER, C. Educação e Treinamento a Distância Baseada na Tecnologia de Realidade Virtual, *V Workshop de Informática na Educação - WIE'99/SBC'99*, p. 669-680. Rio de Janeiro/Brasil, 1999.
- LAU, F. C. M. & XU, C. *Load balancing in Parallel Computers: Theory and Practice*. Kluwer Academic Publishers, p. 232, 1997.
- LEE, P.A. LEE & ANDERSON, T. *Fault Tolerance: Principles and Practice*. Springer-Verlag. 1991.
- LEINWAND, A. & FANG, K. *Network Management A Practical Perspective*, Addison-Wesley Publishing Company, USA, 1993.
- LEITE-JUNIOR, A. J. M., VIDAL, C. A., ALMENDRA, C. C. & SANTOS, E. M., (2001). Construção de Sistemas de Realidade Virtual Semi-Imersivos voltados para Educação a Distância. *Anais do IV Simpósio de Realidade Virtual. Sociedade Brasileira de Computação*. Florianópolis/SC, 2001.
- LEITE-JUNIOR, A. J. M.; VIDAL, C. A.; ALMENDRA, C. C.; SANTOS, E. M.; GOMES, H. O. O. & MENDONÇA-JUNIOR, G. M. (2002). Um ambiente virtual compartilhado voltado para entretenimento. *Proceedings of the 5th Symposium on Virtual Reality*, p. 138-149, Fortaleza, CE, 2002.
- LEITE JUNIOR, A. J. M. *Ataxia: Uma Arquitetura para a Viabilização de CVEs voltados para a educação a distância através da Internet*. Dissertação de Mestrado, Universidade Federal do Ceará, Brasil, 2000.
- MACEDONIA, M. R., ZYDA, M. J. A Taxonomy for Networked Virtual Environments, *IEEE Multimedia*, Vol. 4, No. 1, p. 48-56. 2000.
- MACEDONIA, M. R. & ZYDA, M. J. NPSNET: A Network Software Architecture for Large-Scale Virtual Environments. *Presence: Teleoperators and Virtual Environments*, p. 265-287, 1994.
- MACEDONIA, M. R. & ZYDA, M. J. Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, p. 2-9. 1995.
- MACEDONIA, M. R.; BRUTZMANN, D. P.; Zyda, M. J.; Pratt, D. R. ; Barham, P. T.; Falby, J. & Locke, J. NPSNET: A multi-player 3D virtual environment over the Internet. *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, p. 93--94. ACM SIGGRAPH, Monterey, CA.1995.
- MACEDONIA, M. R. & ZYDA, M. J. Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pp: 2-9. 1995.

- MACHADO, L. S. *Conceitos Básicos da Realidade Virtual*. Instituto Nacional de Pesquisas Espaciais, 1995.
- MARTIN-FLATIN, J.P. *Web-Based Management of IP Networks and Systems*. Ph.D. thesis, Swiss Fed. Inst. of Technology, Lausanne (EPFL), 2000.
- MATIJASEVIC, M. *A review of networked multi-user virtual environments*. Technical Report TR97-8-1, Center for Advanced Computer Studies, Virtual Reality and Multimedia Laboratory, University of Southwestern Louisiana, USA. 1997.
- MCCLOGHRIE, K. & ROSE, M. *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*. RFC 1213, Internet Engineering Task Force, Network Working Group, 1991.
- MORSE, K. L. *Interest management in large scale distributed simulations*. Technical Report pp. 96-27, Department of Information and Computer Science, University of California, Irvine. 1996.
- MULLENDER, S. *Distributed Systems (2nd edition)*. Addison Wesley, ACM Press, New York. 1993.
- OLIVEIRA, J. C. & GEORGANAS, N. D. VELVET: An adaptive hybrid architecture for very large virtual environments. *Proceedings of IEEE International Conference on Communications*. New York, USA, 2002.
- OLIVEIRA, M., CASTRO, M. F., NASCIMENTO, A. & VIDAL, M. Introdução à Gerência de Redes ATM. *XVI Simpósio Brasileiro de Redes de Computadores (SBRC1998)*. Rio de Janeiro, RJ, 1998.
- OMG. *CORBA services: Common Object Services Specification*, OMG Group Document 95-3-31, Revised ed. 2.0, Framingham, Massachusetts, 1995.
- OSI. *Information Technology Open Systems Interconnection, Common Management Information Protocol Specification*. Technical Report IS 9596, International Organization for Standardization, 1991.
- PANDZIC, I. S.; ÇAPIN, T. K.; THALMANN, N. & MAGNENAT-THALMANN, D. VLNET: A Networked Multimedia 3D Environment with Virtual Humans. *Proceedings of Multi-Media Modeling MMM '95*, 1995.
- PANDZIC, I. S.; ÇAPIN, T. K.; THALMANN, N. M., THALMANN, D. VLNET: A Networked Multimedia 3D Environment with Virtual Humans. *Proceedings of Multi-Media Modeling MMM '95*. p.21-32. Singapore, 1995.
- PANDZIC, I.S.; CAPIN, T.K.; THALMANN, N. M. & THALMANN, D. Motor functions in the VLNET Body-Centered Networked Virtual Environment. *Proceedings of 3rd Eurographics Workshop on Virtual Environments*, Monte Carlo, 1996.
- PANTELIDES, V. *Reasons to Use Virtual Reality in Education*. VR in schools. Vol 1, Issue 1. p. 9, 1995.
- PAUSCH, R.; SNOODY, J.; TAYLOR, R.; WATSON, S. & HASELTINE, E. Disney's Alladin: First Steps Toward Storytelling in Virtual Reality. *Proceedings of SIGGRAPH'96*. 1996.
- PEREIRA, M. C. *Administração e Gerência de Redes de Computadores*. Pós-graduação em Ciência da Computação - CPGCC. Universidade Federal de Santa Catarina - UFSC, Florianópolis - SC.

- PRADO, G. Desertesejo: um projeto de ambiente virtual de multi-usuário na Web. Catálogo Eletronic Art Exhibition. 13th SIBGRAPI 2000. *Brazilian Symposium on Computer Graphics and Image Processing*. Caxias do Sul, 2000.
- PRAS, A. *Network management architectures*. PhD thesis, University of Twente, Enschede, The Netherlands, 1995.
- RODELLO, I. A.; KUBO, M. M.; KIRNER, C. & TORI, R. Análise dos Principais Fatores para o Desenvolvimento de Sistemas Distribuídos de Realidade Virtual. *Proceedings of the 4th SBC Symposium on Virtual Reality*, Florianópolis, 314- 323. 2001.
- RODELLO, I. A., SAKAI, R. K. & MANOEL, E. F. Um Ambiente Virtual para Auxiliar o Ensino de Química em Escolas de Ensino Fundamental. *XXII Congresso da Sociedade Brasileira de Computação. Anais do VIII WIE*. Florianópolis, SC. 2002.
- RODRIGUES, L. C. R.; SEMENTILLE, A. C.; BREGA, J. R. F. & RODELLO, I. A. Gerenciamento de Ambientes Virtuais de Grande Escala Utilizando Portais. *I Workshop de Computação. Florianópolis (WORKCOMP-SUL)*. UNISUL, 2004.
- ROSE, M.T. *The Simple Book: an introduction to management of TCP/IP- based Internets*. Englewood Cliffs, USA. Prentice-Hall, 1991
- ROTHBAUM, B.; HODGES, L.; KOOPER, R.; OPDYKE, D.; WILLIFORD, J. & NORTH, M. Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia. *American Journal of Psychiatry*, Vol. 152, No. 4. 1995.
- RYAN, B. D. Alphaworld: the urban design of a digital city. *Journal of Urban Design*, Volume 9, Number 3, pp. 287-309, 2004.
- SANTOS, E.M. *VEGA - Um modelo de integração de aplicações externas a ambientes virtuais através do uso de aplicações mediadoras e de simulóides*. Dissertação de Mestrado, Mestrado em Ciência da Computação, UFC, Brasil. 2001.
- SHARDA, N.; FATRI, R. & ABID, M. Applying deductive database technology to network management. *ACM SIGCOMM Computer Communication Review*, 1997.
- SILVA, R. J. M.; WAGNER, G. N.; RAPOSO, A. B. & GATTAS, M. Experiência de Portais em Ambientes Arquitetônicos Virtuais. *Proceedings of the 6th Symposium on Virtual Reality*, p. 117-128, Ribeirão Preto, SP. 2003.
- SILVA, M. C. O. *Projeto e Implementação de uma ferramenta para detecção de hardware em ambientes heterogêneos*. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação). Universidade Católica Dom Bosco, Campo Grande - MS. 2003.
- SINGHAL, S. & ZYDA, M. *Networked Virtual Environments: Design and Implementation*. ACM Press/Addison-Wesley Publishing Co., New York/NY, USA. 1999.
- SINTEGRA. *Sistema integrado de Educação a Distância com suporte a Avaliação de Competências*. Disponível em: <http://www.vdl.ufc.br/sintegra>. Acesso em: 01/06/2006.
- SLATER, M.; USH, M. & STEED, A. Depth of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 2. 1994.
- SLATER, M.; USOH, M.; BENFORD, S.; BROWN, C.; RODDEN, T.; SMITH, G. & WILBUR, S. *Distributed extensible virtual reality laboratory, Virtual Environments and Scientific Visualization*. Nova York: Springer. 1996.

- SMED, J.; Kaukoranta, T. & HAKONEN, H. Aspects of Networking in Multiplayer Computer Games. *Proceedings of International Conference on Application and Development of Computer Games in the 21st Century*, p.74-81, Hong Kong, China, 2001.
- SOMMERVILLE, I. & SAWER, P. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, England, 1997.
- SOUZA, P.C., WAZLAWICK, R. S., & HOFFMANN, A. Uma ferramenta de autoria para o desenvolvimento de aventuras educacionais. *Congreso Argentino de Ciencia de la Computacion*, p. 959-969.
- STALLINGS, W. *SNMP, SNMPv2, and CMIP The Practical Guide to Network-Management Standards*, Addison Wesley. 1993.
- STALLINGS, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Prentice Hall. 1999.
- STYTZ, M.R. Distributed Virtual Environments. *IEEE Computer Graphics and Applications*, Vol. 16, No. 33, pp. 19-31. 1996.
- SUN MICROSYSTEMS. *JMX: Java Management Extensions, First Public Specification*, v1.9, USA, 1999.
- SZWARCMAN, D.; Feijó, B. & Costa, M. *A framework for networked emotional characters*, Monografia de Ciência da Computação, MCC 23/99, Dept. Informática, PUC-Rio, 1999.
- TANENBAUM, A. S. (2002). *Distributed Systems: Principles and Paradigms*. Prentice Hall, Upper Saddle River, New Jersey 07458, 2002.
- VALLE FILHO, A. M.; SOUZA, P. C.; ALVES, J. B. M.; WAZLAWICK, R. S., & LUZ, R. P. Ferramentas de autoria de realidade virtual - um estudo comparativo. *Anais do Congresso Internacional de Engenharia Informática*. 2000.
- VIDAL, C. A.; LEITE JÚNIOR, A. J. M.; ALMENDRA, C. C.; SANTOS, E. M.; OLIVEIRA, J. P. C. & COSTA FILHO, J. C. S. A Virtual Environment for Distance Learning. *Proceedings of the 7th World Conference on Computers in Education - Networking the Learner*, p. 95-103. Copenhagen: Kluwer Academic Publishers and IFIP. 2001.
- VIDAL, C. A.; SANTOS, E. M.; LEITE-JÚNIOR, A. J. M.; ALMENDRA, C. C. & BORGES, V. M. C. Collaborative Virtual Environments for Language Learning. *Proceedings of the 6th Symposium on Virtual Reality*, p.309-320, Ribeirão Preto-SP. 2001.
- WAZLAWICK, R. S.; Fagundes, L. C. & Kirner, T. G. *Museu Virtual - Ferramenta de Autoria para Criação de Museus de Realidade Virtual para Apoio à Aprendizagem Colaborativa via Internet*. Projeto ProTeM PTI/PEDU, Brasil. 1999.
- WAZLAWICK, R. S.; ROSATELLI, M. C.; RAMOS, E. M. F., Cybis, W. A., Storb, B. H., Schuhmacher, V. R. N., Mariani, A. C., Kirner, T., Kirner, C. & Fagundes, L. C. Providing More Interactivity to Virtual Museums: A Proposal for a VR Authoring Tool. *Presence: Teleoperators and Virtual Environments*, Vol.10, Issue 6.
- WBEM. *Web-Based Enterprise Management Initiative*. Disponível em: <http://www.dmtf.org/standards/wbem/>. Acesso em: 12/08/2005.
- WIDENIUS M. & AXMARK, D. *MySQL Reference Manual*, O'Reilly & Associates, 2002.

Apêndices

Apêndice A – Código da interface JNI

```

#include <jni.h>
/* Header for class br_crabcve_utils_SystemCall */

#ifndef _Included_br_crabcve_utils_SystemCall
#define _Included_br_crabcve_utils_SystemCall
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class:      br_crabcve_utils_SystemCall
 * Method:     getPhysicalTotalMemory
 * Signature:  ()F
 */
JNIEXPORT jfloat JNICALL Java_br_crabcve_utils_SystemCall_getPhysicalTotalMemory
    (JNIEnv *, jclass);

/*
 * Class:      br_crabcve_utils_SystemCall
 * Method:     getNotUsingPhysicalMemory
 * Signature:  ()F
 */
JNIEXPORT jfloat JNICALL Java_br_crabcve_utils_SystemCall_getNotUsingPhysicalMemory
    (JNIEnv *, jclass);

/*
 * Class:      br_crabcve_utils_SystemCall
 * Method:     getUsingPhysicalMemory
 * Signature:  ()F
 */
JNIEXPORT jfloat JNICALL Java_br_crabcve_utils_SystemCall_getUsingPhysicalMemory
    (JNIEnv *, jclass);

/*
 * Class:      br_crabcve_utils_SystemCall
 * Method:     getUsingPhysicalMemoryRatio
 * Signature:  ()F
 */
JNIEXPORT jfloat JNICALL Java_br_crabcve_utils_SystemCall_getUsingPhysicalMemoryRatio
    (JNIEnv *, jclass);

/*
 * Class:      br_crabcve_utils_SystemCall
 * Method:     getMeanCPUUsing
 * Signature:  ()F
 */
JNIEXPORT jfloat JNICALL Java_br_crabcve_utils_SystemCall_getMeanCPUUsing
    (JNIEnv *, jclass);

#ifdef __cplusplus
}
#endif
#endif

```


Apêndice B – Diagrama de seqüência do teste de validação

