

CRAbCVE – Uma Arquitetura para Viabilização de CVEs através da Internet

por

George Allan Menezes Gomes

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

Orientadores:

Creto Augusto Vidal, Ph.D.

Joaquim Bento Cavalcante Neto, Dr.

Setembro, 2005

“A luta contra o erro tipográfico tem algo de homérico. Durante a revisão os erros se escondem, fazem-se positivamente invisíveis. Mas assim que o livro sai, tornam-se visibilíssimos...”

(Monteiro Lobato)

A minha irmã Andréa Marla
In memoriam

Agradecimentos

Muitas pessoas colaboraram nesta conquista, tantas que certamente cometerei lapsos ao mencionar apenas algumas. Fica a certeza que só cheguei até aqui porque fui incentivado e apoiado por muitas pessoas importantes.

Em primeiro lugar, agradeço a Deus por mais uma conquista em minha vida.

A minha esposa Lia, pela compreensão, amor e principalmente pelos momentos felizes durante todo este percurso.

Aos meus pais, Raimundo Gomes e Risetete, pelo amor, apoio e incentivo em todos os momentos, amo muito vocês.

A minha irmã Andréa Marla, os meus mais sinceros sentimentos de gratidão por sempre ter incentivado e me apoiado em todos os momentos de minha vida. Apesar de ter partido, sempre estará nos nossos corações. A minha irmã Taise, pelo carinho, apoio e por ter suportado minha ausência.

A todos os meus familiares, em especial aos meus primos Ana Claudya, Auryleda e André; as minhas tias Elisabete e Socorro e, aos meus sogros Elias e Célia pelo apoio durante minha formação acadêmica.

Aos meus grandes amigos Melo Júnior e todos que formam o grupo NOJs (Camilo, Humberto, Ney, Manu, Eduardo, Henrique, Wellington, Patrícia, Gilvan, Cátia, Luiz, David, Paulinho) pelo companheirismo e trabalho em equipe que direta ou indiretamente contribuíram para desenvolvimento deste trabalho.

Ao professor Creto por sua amizade e orientação que serviram de inspiração e apoio no desenvolvimento desta dissertação, no trabalho desenvolvido nos projetos AVAL, Ericsson-UFC, SINTEGRA e, acima de tudo, no meu desenvolvimento e amadurecimento acadêmico, profissional e pessoal.

Ao professor Joaquim Bento por ter aceitado participar da co-orientação deste trabalho e, com dedicação e competência, sugeriu melhoramentos.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico, FUNCAP, pelo apoio financeiro.

Aos colegas, funcionários e professores do Departamento de Computação pelo apoio e aprendizado durante o mestrado.

Sumário

A complexidade das tarefas do mundo atual requer cada vez mais a cooperação das pessoas para sua execução. Entretanto cooperar não é uma tarefa fácil, e, muitas vezes, por falta de coordenação adequada ou por uma definição pobre do contexto em que as atividades dos participantes individualmente ou do grupo serão realizadas, são geradas redundâncias, inconsistências e contradições dentro do trabalho em grupo. Para evitar esses problemas, a área de CSCW (*Computer Supported Cooperative Work*) vem buscando meios de suportar adequadamente o trabalho em equipe. Apesar dos bons resultados alcançados pela área de CSCW, o principal obstáculo enfrentado pelos seus desenvolvedores é fornecer interatividade a suas aplicações. O uso de realidade virtual em sistemas cooperativos permite aos participantes interagirem com o mais alto grau de naturalidade, pois através do ambiente virtual é possível a comunicação por meio de imagem, texto e áudio.

Os CVEs (*Collaborative Virtual Environments*) fazem uso da tecnologia de realidade virtual distribuída e apresentam grande potencial para o suporte ao trabalho colaborativo. Por conseguinte, esses ambientes virtuais colaborativos têm sido desenvolvidos, levando em consideração os resultados obtidos na área de CSCW. Entretanto, desenvolver CVEs é uma tarefa complicada, pois eles são grandes consumidores de recursos, e precisam incorporar conceitos e recomendações de várias áreas de pesquisa, como a de Realidade Virtual (RV), a de Trabalho Colaborativo Assistido por Computador (CSCW) e a de Sistemas Computacionais Distribuídos.

O principal objetivo dessa dissertação é a definição de uma arquitetura de uso genérico, a CRAbCVE, projetada para viabilizar vários ambientes virtuais colaborativos (CVEs) distribuídos em uma rede de servidores na Internet. Essa arquitetura define componentes especializados, capazes de distribuir, através da Internet, o processamento de todo o CVE; obtendo-se, assim, um grande poder computacional a um baixo custo. Outro importante objetivo é a definição do modelo MAC (Modelo de Autoria em CVEs), que visa auxiliar o emprego da arquitetura CRAbCVE no trabalho colaborativo.

Os modelos propostos foram incorporados em um sistema protótipo e um estudo de caso simples foi analisado.

Summary

Although the complexities of many tasks encountered in modern societies require the joint effort of groups of people in order to be accomplished, cooperative work is still a difficult job. Usually the difficulties arise due to lack of appropriate coordination, poor definition of the context in which the activities are to be performed individually or in group; thus, generating redundancies, inconsistencies and contradictions within the workgroup. In order to overcome these problems, a new field of research, called Computer Supported Cooperative Work (CSCW), was created to seek means of properly supporting work groups. Despite the favorable results achieved by CSCW, thus far, tools that provide a high level of interaction among the group members and the leaders of sub-groups are still not satisfactory. The use of virtual reality within cooperative systems allows the interactions among participants to be highly spontaneous, because, in virtual environments, communication by means of image, text and audio is possible.

The Collaborative Virtual Environments (CVEs), which employ shared virtual reality technology, have proved to possess great potential for collaborative work. Therefore, collaborative virtual environments have been developed taking into consideration the results obtained by CSCW research. Nonetheless, developing CVEs is complicated, since they demand a great deal of resources, and need to incorporate concepts and recommendations from several research fields, such as Virtual Reality (VR), Computer Supported Cooperative Work and Distributed Computing.

The main objective of this dissertation is to design a generic architecture (CRAbCVE) for allowing distribution of several collaborative virtual environments within a network of servers on the Internet. This architecture defines specialized components, capable of distributing the processing that takes place in the CVE, thus obtaining a great computational power at low costs. Another important objective is the specification of a Model of Authorship in CVEs (MAC) for helping to integrate the CRAbCVE architecture into the framework of collaborative work.

All models proposed herein have been incorporated into a prototype system and a simple case study has been analyzed.

Índice

Lista de Anexos	xi
Lista de Figuras	xii
Lista de Tabelas	xv
Capítulo 1	1
Introdução	1
1.1 Motivação	1
1.2 Objetivos.....	2
1.3 Estrutura da Dissertação	3
Capítulo 2	5
Colaboração Assistida por Computador	5
2.1 Introdução.....	5
2.2 Processos Envolvidos na Colaboração	6
2.2.1 Comunicação	8
2.2.2 Negociação	9
2.2.3 Coordenação	10
2.2.4 Compartilhamento	12
2.2.5 Colaboração	13
2.2.6 Percepção.....	14
2.3 Taxonomias de Groupware.....	16
2.3.1 Taxonomia Tempo-Espaço.....	16
2.3.2 Taxonomia Baseada em Funcionalidades.....	17
2.4 Considerações Finais	21
Capítulo 3	22
Ambientes Virtuais Colaborativos - CVEs.....	22
3.1 Introdução.....	22
3.2 Características de CVEs	23
3.3 Requisitos para Desenvolvimento de CVEs.....	25
3.3.1 Fidelidade	26
3.3.2 Interação	27
3.3.3 Comunicação não Verbal.....	28
3.3.4 Suporte à Colaboração.....	29

3.3.5 Escalabilidade	30
3.3.6 Adequação à Largura de Banda Disponível	31
3.3.7 Redução do Atraso de Rede	32
3.3.8 Modelo de Comunicação	32
3.3.9 Modelo de Armazenamento.....	35
3.3.10 Protocolo de Comunicação	36
3.4 Arquiteturas de CVEs	38
3.4.1 DIVE.....	39
3.4.2 SIMNET	40
3.4.3 NPSNET	40
3.4.4 MASSIVE.....	41
3.4.5 VELVET.....	42
3.4.6 Blaxxun Community Platform	42
3.4.7 OzGate!.....	43
3.5 Aplicações de CVEs	44
3.5.1 NICE.....	44
3.5.2 HistoryCity	45
3.5.3 AVC–MV	46
3.5.4 AVAL	47
3.5.5 Gorilla World.....	48
3.5.6 Active Worlds.....	48
3.5.7 Ragnarök Online.....	49
3.6 Considerações Finais	50
Capítulo 4	51
A Arquitetura CRAbCVE.....	52
4.1 Introdução	52
4.2 A Arquitetura Ataxia	53
4.3 Características da Arquitetura CRAbCVE	55
4.3.1 Arquitetura Distribuída e Híbrida.....	55
4.3.2 Suporte a Implementações Multiplataforma.....	55
4.3.3 Expansível	55
4.3.4 Arquitetura Escalar	56
4.3.5 Forte Dependência de Administração.....	56

4.3.6	Minimização da Utilização da Rede de Comunicação	57
4.3.7	Persistência de Estado e Autoria de CVEs	58
4.4	Componentes da Arquitetura CRAbCVE.....	59
4.4.1	Gerenciador de Políticas do Sistema	59
4.4.2	Cliente de Interação e Edição em Realidade Virtual.....	63
4.4.3	Roteador de Mensagens.....	65
4.4.4	Gerenciador de Eventos.....	66
4.4.5	Gerenciador de Políticas do Mundo Virtual	69
4.4.6	Fornecedor de Mídias	70
4.4.7	Controlador de Simulóide.....	72
4.4.8	Cliente de Administração	73
4.5	Funcionamento da Arquitetura CRAbCVE.....	74
4.5.1	Cenários de Utilização.....	75
4.5.2	Recomendações de Uso	92
4.6	Protocolo PASSÁRGADa 2.0	95
4.6.1	Conexão de Clientes	95
4.6.2	Instanciação de Elementos.....	97
4.6.3	Manipulação de Elementos.....	98
4.6.4	Comunicação	100
4.6.5	Serviços Específicos da Arquitetura CRAbCVE.....	101
4.6.6	Integração	102
4.7	Considerações Finais	102
Capítulo 5	105
MAC – Modelo de Autoria em CVEs	106
5.1	Introdução.....	106
5.2	Construção de Ambientes Virtuais.....	107
5.3	Ferramenta de Autoria	108
5.3.1	Bibliotecas	109
5.3.2	Comportamentos.....	111
5.3.3	Controle de Objetos	112
5.3.4	Publicação de Arquivos	112
5.4	Políticas de Acesso	113
5.4.1	Controle de Acesso	114

5.4.2 Controle de Concorrência.....	115
5.5 Coordenação	119
5.6 Percepção.....	121
5.7 Comunicação	122
5.8 Considerações Finais	122
Capítulo 6	123
Estudo de Caso da CRAbCVE Utilizando o Modelo MAC.....	124
6.1 Introdução	124
6.2 Síntegra.....	124
6.3 Curso de Boas Práticas	125
6.4 Estudo de Caso	126
6.4.1 Gerenciador de Políticas do Sistema	127
6.4.2 Fornecedor de Mídias	128
6.4.3 Roteador de Mensagens.....	130
6.4.4 Gerenciador de Eventos.....	130
6.4.5 Cliente de Interação e Edição em RV.....	131
6.4.6 Protocolo PASSÁRGADa 2.0	137
6.5 Considerações Finais	138
Capítulo 7	139
Conclusões e Trabalhos Futuros.....	140
7.1 Principais contribuições.....	140
7.2 Trabalhos futuros	141
Referências Bibliográficas.....	143

Lista de Anexos

Anexo A – Exemplo de arquivo descritor de mundo virtual.....	158
Anexo B – Exemplos de arquivo descritor de mídia.....	163
Anexo C – Diagramas de Classes.....	167

Lista de Figuras

Figura 2.1. Modelo de Colaboração 3C (Gerosa et al., 2002).....	7
Figura 2.2. Comunicação direta e indireta (baseada em Xu, 2000).....	9
Figura 2.3. Classificação de sistemas de groupware em termos de comunicação síncrona, assíncrona, direta e indireta (baseada em Xu, 2000).	10
Figura 2.4. Modelando a Coordenação (Fuks et al., 2002).	12
Figura 2.5. Modelando a Colaboração (Fuks et al., 2002).	14
Figura 2.6. Esquema para classificação espaço-temporal das aplicações em trabalho cooperativo auxiliado por computador.	17
Figura 3.1. Modelo de Comunicação centralizado.	33
Figura 3.2. Modelo de Comunicação Distribuído.	33
Figura 3.3. Modelo com conexões unicast.	34
Figura 3.4. Modelo com conexões broadcast.	34
Figura 3.5. Modelo com conexões multicast.	35
Figura 3.6. Participantes compartilhando um quadro branco no DIVE.	39
Figura 3.7. Navegador do NPSNET-V.	41
Figura 3.8. Inhabited TV Combina CVE com TV.	41
Figura 3.9. Interface do ambiente Blaxxun.	42
Figura 3.10. Navegador do OzGate Traveller.	43
Figura 3.11. Uma Criança controlando um avatar dentro do NICE.	45
Figura 3.12. Ambiente do HistoryCity.	45
Figura 3.13. Ferramenta de autoria do Museu Virtual.	46
Figura 3.14. CVE de um centro de turismo virtual.	47
Figura 3.15. Habitante do Gorilla World.....	48
Figura 3.16. Interface do sistema ActiveWorld.....	49
Figura 3.17. Um CVE do sistema Ragnarök Online.	50
Figura 4.1. Componentes da arquitetura Ataxia.	54
Figura 4.2. Distribuição dos componentes definidos na arquitetura CRAbCVE.	60
Figura 4.3. Distribuição de fábricas e CVEs (Baseado em Almendra, 2003).	61
Figura 4.4. Diagrama de implementação do componente GPS.	63
Figura 4.5. Diagrama de implementação do componente CIERV.	65
Figura 4.6. Diagrama de implementação do RM.	66
Figura 4.7. Diagrama de implementação do componente GE.	69
Figura 4.8. Diagrama de implementação do componente GPMV.....	70

Figura 4.9. Diagrama de implementação do componente FM.	72
Figura 4.10. Diagrama de implementação do componente CA.....	74
Figura 4.11. Cenário de instalação de fábrica.	75
Figura 4.12. Cenário de sistema iniciando.	76
Figura 4.13. Cenário de instanciação de CVEs.	77
Figura 4.14. Componentes essenciais de um CVE.....	78
Figura 4.15. Cenário de conexão de clientes a CVEs.....	79
Figura 4.16. Cenário de envio de mensagem de texto.....	81
Figura 4.17. Cenário de animação de avatar.	82
Figura 4.18. Cenário de movimentação de avatar.	83
Figura 4.19. Cenário de interação com objetos.	85
Figura 4.20. Cenário de inserção de objetos.....	86
Figura 4.21. Cenário de remoção de objetos.	87
Figura 4.22. Cenário de alteração de Permissões.....	88
Figura 4.23. Cenário de publicação de mundos virtuais.....	88
Figura 4.24. Cenário de formação de grupo.	90
Figura 4.25. Cenário de desinstanciação de CVEs.....	91
Figura 4.26. Cenário de desinstalação de fábricas.....	92
Figura 4.27. Exemplo de emprego da arquitetura em um único subambiente.	94
Figura 4.28. Exemplo de emprego da arquitetura com dois subambientes simultaneamente.	94
Figura 5.1. Organização dos arquivos do mundo virtual.....	113
Figura 6.1. Ambiente Solar e o curso Boas Práticas.	126
Figura 6.2. Implementação do GPS como um Web Service.	128
Figura 6.3. Implementação do FM como um Web Service.....	129
Figura 6.4. Na figura à esquerda, a lista de mídias cadastradas. Na direita, um formulário para cadastrar novas mídias. Ambas são Web Services.....	129
Figura 6.5. Implementação do Roteador de Mensagens mostrando o registro de ocorrências do sistema.....	130
Figura 6.6. GE mostrando os usuários conectados.....	131
Figura 6.7. GE mostrando registro de PDUs que chegam e que são enviadas.....	131
Figura 6.8. Tela do Cliente de Interação e Edição em RV (CIERV).	132
Figura 6.9. Tela de autenticação do CIERV com o GPS.....	133
Figura 6.10. À esquerda, tela com a lista de mundos do usuário Lia, e a direita, tela com a lista de mundos do usuário João.	134
Figura 6.11. À esquerda, tela de criação de tarefas, à direita, tela de inserção de dependências.....	135

Figura 6.12. À esquerda, a tela com informações sobre o usuário Lia, coordenador geral. À direita, a tela com informações sobre subcoordenador João.	136
Figura 6.13. Biblioteca de objetos para o estudo de caso.	137
Figura C.7.1. Diagrama de Classes do CIERV.	168
Figura C.7.2. Diagrama de Classes do RM.	169
Figura C.7.3. Diagrama de Classes do GE.	169
Figura C.7.4. Diagrama de Classes do GPS.	170
Figura C.7.5. Diagrama de Classes do FM.	170

Lista de Tabelas

Tabela 1. PDU <i>Connect</i> . Conexão de clientes em um determinado subambiente.	95
Tabela 2. PDU <i>Disconnect</i> . Desconexão de clientes em um determinado subambiente. ...	96
Tabela 3. PDU <i>Reconnect</i> . Reconexão de clientes em subambientes.	96
Tabela 4. PDU <i>Accept</i> . Aceitação ou não de conexão de cliente.	96
Tabela 5. PDU <i>In</i> . Criação de elemento virtual.....	97
Tabela 6. PDU <i>Out</i> . Remoção de elemento virtual.	97
Tabela 7. PDU <i>Position</i> . Mudança de posição ou escala, com ou sem deslocamento.	98
Tabela 8. PDU <i>Move</i> . Movimenta ou rotaciona um elemento virtual.....	98
Tabela 9. PDU <i>Group</i> . Agrupa um conjunto de elementos virtuais.....	98
Tabela 10. PDU <i>Action</i> . Executa uma ação pré-definida de um elemento virtual.	98
Tabela 11. PDU <i>Status</i> . Modifica o estado de um elemento virtual.....	98
Tabela 12. PDU <i>Permission</i> . Modifica nível de permissão de um elemento virtual.....	98
Tabela 13. PDU <i>Publish</i> . Modifica a versão de um mundo virtual.....	99
Tabela 14. PDU <i>Show</i> . Apresentação de Mídias.	100
Tabela 15. PDU <i>What</i> . Informações sobre elemento virtual ou componente da arquitetura.	100
Tabela 16. PDU <i>Administrate</i> . Administração de componentes da arquitetura.	101
Tabela 17. PDU <i>Log</i> . Documentação de ocorrência.	101
Tabela 18. PDU <i>Alive</i> . Checagem de estado de componente da arquitetura.....	101
Tabela 19. PDU <i>Generic</i> . Integração com outros sistemas/ferramentas.	102
Tabela 20. Soluções adotadas pela arquitetura CRAbCVE para atender os requisitos identificados no Capítulo 3.....	103
Tabela 21. Lista de tarefas e suas dependências.....	135
Tabela 22. Lista de PDUs implementadas para o protocolo PASSÁRGADa 2.0.	137

Capítulo 1

Introdução

1.1 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 vários 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.

O ambiente computacional que implementa os processos de apoio à cooperação, e assim possibilita o trabalho conjunto, bem como a necessária troca de informações, denomina-se sistema de trabalho cooperativo apoiado por computador (CSCW - *Computer Supported Cooperative Work* (Ellis *et al.*, 1991; Borges *et al.*, 1995)). Nesse sentido, ambientes CSCW representam a união de diversas questões, abordagens e linguagens, voltadas ao estabelecimento de uma estrutura de apoio automatizada a atividades realizadas cooperativamente por pessoas, através do intercâmbio de informações (Reinhard *et al.*, 1994).

Apesar dos bons resultados já obtidos com os ambientes de CSCW desenvolvidos, o principal obstáculo na utilização desses sistemas é fornecer interatividade a eles (Vicentin *et al.*, 1999). O uso de Realidade Virtual (RV) em sistemas cooperativos permite que os participantes interajam com o mais alto grau de naturalidade, pois através do ambiente virtual é possível a comunicação por meio de imagem, texto e áudio.

Segundo Çapin e seus co-autores (1999), realidade virtual pode ser definida como a 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. Com essa tecnologia, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações. De fato, o uso

de ambientes virtuais está sendo difundido pelas diversas áreas do conhecimento. Assim, atualmente, podem-se observar aplicações de RV em áreas tais como educação e cultura (Vidal *et al.*, 2003; Kaufmann *et al.*, 2000), medicina (Freitas *et al.*, 2003), arquitetura (Silva *et al.*, 2003), entretenimento (Leite-Junior *et al.*, 2002; Cardoso *et al.*, 2003), entre outras.

As aplicações de RV são projetadas de forma a proporcionar ao usuário sensações de imersão, no ambiente virtual, compatíveis com aquelas que ele experimenta no mundo real. Segundo Slater e seus co-autores (1994), um ambiente imersivo ideal é aquele onde os órgãos sensoriais do usuário são estimulados de maneira contínua pelo sistema computacional. Os sistemas imersivos atuais fazem uso de hardware especial de custo ainda muito elevado. Os sistemas chamados *Desktop Virtual Reality*, por sua vez, utilizam computadores comuns e não necessitam de hardware especial. No entanto, proporcionam ao usuário um nível de imersão mais modesto.

A principal motivação para o uso de realidade virtual no trabalho colaborativo advém da possibilidade de apresentar a complexidade de situações do mundo real de tal maneira que pessoas possam observar diretamente os resultados de suas próprias decisões. É então neste ponto que os trabalhos das áreas de CSCW e realidade virtual se unem. Através dos estudos para o trabalho em grupo e os métodos de interação e visualização de RV é possível modelar e projetar ambientes aptos a dar apoio a grupos, conhecido como Ambientes Virtuais Colaborativos (AVC ou CVE – *Collaborative Virtual Environments*).

CVEs permitem a interação e a colaboração entre usuários remotos em modelos computacionais tridimensionais de um ambiente real ou imaginário. No desenvolvimento de um CVE, são aglutinadas tecnologias de pelo menos quatro áreas: CSCW, processamento distribuído, computação gráfica e realidade virtual. Segundo Stytz (1996), CVEs são considerados um dos sistemas de software mais complexos já construídos, pois apresentam características de vários outros tipos de sistemas computacionais. Por exemplo, segundo Singhal & Zyda (1999), CVEs apresentam características de sistemas distribuídos, de aplicações gráficas e de aplicações interativas.

1.2 Objetivos

O presente trabalho objetiva explorar o uso das tecnologias de CSCW e de CVEs no desenvolvimento de uma arquitetura que disponibilize ambientes virtuais que suporte

atividades básicas de trabalho cooperativo ao mesmo tempo em que agregue elementos associados a atividades de aprendizagem e tutoria. Para que isso seja possível, a arquitetura proposta, a CRAbCVE, define um conjunto de componentes que podem ser executados em diversos microcomputadores comuns, de maneira simultânea, fornecendo o suporte às necessidades desse tipo de sistema multiusuário.

Para atingir o objetivo principal do trabalho, que é a criação da arquitetura CRAbCVE, é realizada uma análise dos vários conceitos importantes na área de colaboração assistida por computador (CSCW), contribuindo para a identificação dos aspectos cruciais na implementação de aplicações colaborativas. Esse trabalho tem como diferencial agrupar de forma sintética os principais processos considerados importantes, por pesquisadores da área, para o desenvolvimento de sistemas que apoiem o trabalho colaborativo. Também é realizado um estudo acerca de Ambientes Virtuais Colaborativos, incluindo os requisitos mais importantes de software e hardware que devem ser considerados no seu desenvolvimento para tornar disponíveis suas respectivas informações através de redes de computadores comuns.

Outro objetivo importante é a especificação de um Modelo de Autoria em CVEs (MAC), para auxiliar o emprego da arquitetura CRAbCVE no trabalho colaborativo. O MAC consiste de uma especificação de uma ferramenta de autoria e de uma descrição dos mecanismos empregados para dar suporte ao trabalho colaborativo.

O último objetivo deste trabalho é apresentar a especificação e a implementação de um estudo de caso aplicado à educação utilizando a arquitetura CRAbCVE e seguindo o modelo MAC.

1.3 Estrutura da Dissertação

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

No Capítulo 2, apresenta-se a tecnologia de colaboração assistida por computador (CSCW), incluindo conceitos fundamentais. Também nesse capítulo, são identificados os processos envolvidos na colaboração assistida por computador e as taxonomias existentes. No Capítulo 3, a tecnologia de CVEs é apresentada, discutindo-se seus conceitos e identificando-se os aspectos e requisitos de *software* e *hardware* necessários para disponibilizar CVEs através da Internet. No Capítulo 4, discutem-se as características da

arquitetura CRAbCVE, apresentam-se seus componentes e é discutido como esses componentes são empregados na disponibilização de espaços virtuais compartilhados. Ainda nesse capítulo, discutem-se vários cenários de utilização a partir dos quais se define o protocolo de comunicação da arquitetura. No Capítulo 5, expõe-se o modelo MAC de autoria em CVEs voltado para auxiliar o emprego da arquitetura CRAbCVE no trabalho colaborativo. No Capítulo 6, um estudo de caso utilizando a arquitetura CRAbCVE e seguindo modelo MAC é detalhado. Por fim, no Capítulo 7, são apresentadas as conclusões deste trabalho e indicadas algumas sugestões de trabalhos futuros.

Capítulo 2

Colaboração Assistida por Computador

2.1 Introdução

Neste capítulo são abordados os principais conceitos sobre a área de pesquisa de Colaboração Assistida por Computador, a terminologia dessa área e os termos adotados neste trabalho. Por conseguinte, os processos envolvidos na colaboração são identificados e sua importância, bem como suas peculiaridades, são analisadas. Por fim, as taxonomias das aplicações existentes são apresentadas e discutidas.

Os avanços tecnológicos das últimas décadas têm mudado substancialmente a forma como as pessoas interagem e trabalham. Atualmente, a facilidade do acesso à Internet permite que pessoas geograficamente distribuídas colaborem entre si, de forma síncrona ou assíncrona, para solução de problemas das mais diversas naturezas. Entretanto, esse meio ainda não tem sido explorado adequadamente para permitir que pessoas trabalhem em grupo de forma produtiva.

Vários esforços no sentido de resolver esse problema têm sido aglutinados em uma área de pesquisa denominada CSCW, que originalmente foi direcionada ao suporte do trabalho colaborativo em tarefas bem estruturadas. Nesse sentido, CSCW representa a união de diversas questões, abordagens e linguagens, voltadas ao estabelecimento de uma estrutura de apoio automatizada a atividades realizadas cooperativamente por pessoas, através do intercâmbio de informações (Reinhard *et al.*, 1994).

Por ser uma área de pesquisa bastante interdisciplinar, existe uma grande discussão entre os autores sobre a utilização do termo ‘colaboração’ em vez de ‘cooperação’. Alguns autores preferem definir como ‘colaboração’ a realização de tarefas específicas, individualmente, pelos integrantes do grupo em busca de alcançar um objetivo comum, e ‘cooperação’ como a realização dessas tarefas em conjunto (Panitz, 1996). Segundo Lubich (1995), a utilização da diferenciação dos termos não é importante no contexto dos trabalhos desenvolvidos pela comunidade. Nesse sentido, os termos ‘colaboração’ e ‘cooperação’ serão utilizados como sinônimos no restante deste trabalho.

A área de pesquisa que apóia a realização de tarefas para o ensino ou aprendizagem em grupo é denominada de Aprendizado Cooperativo Assistido por Computador (CSCL –

Computer Supported Cooperative Learning) (Borges *et al.*, 1995). Outra discussão entre os autores é a adoção da área de CSCL como uma subdivisão de CSCW dedicada às aplicações educacionais, uma vez que, freqüentemente, suportam atividades básicas do trabalho cooperativo ao mesmo tempo em que agregam elementos associados a atividades de aprendizagem e tutoria (Barros, 1994). No contexto deste trabalho, CSCW refere-se a qualquer esforço de colaboração mediado por computador entre um conjunto de pessoas a fim de alcançar um objetivo comum e, portanto, a área de CSCL está incluída em CSCW.

O termo *groupware* é utilizado por alguns autores como sinônimo de CSCW, porém, na literatura, identifica-se uma tendência diferenciada no emprego desses termos. Enquanto CSCW é utilizado para designar a área de pesquisa de trabalho em grupo com a assistência de computadores, *groupware* tem sido utilizado para indicar a tecnologia (hardware ou software) gerada pela pesquisa nessa área. Segundo Elis e seus co-autores (1991), *groupware* é o sistema computacional que dá suporte a grupos de pessoas engajadas em uma tarefa comum e que disponibiliza uma interface para um ambiente compartilhado. A noção de tarefa comum é fundamental nessa definição, pois exclui os sistemas multiusuários em que não há compartilhamento entre os usuários para atingir um objetivo comum.

O restante deste capítulo está organizado da seguinte forma: na Seção 2.2, são discutidos os processos envolvidos na colaboração assistida por computador; na Seção 2.3, as taxonomias em função das tecnologias utilizadas por *groupware* são apresentadas; e por último, na Seção 2.4, tecem-se algumas considerações finais.

2.2 Processos Envolvidos na Colaboração

Segundo Barros (1994), a colaboração envolve cinco processos: comunicação, negociação, co-realização, compartilhamento e coordenação. A comunicação entre os participantes ocorre durante todo o tempo de vida do grupo. A negociação ocorre sempre que decisões de planejamento e execução das tarefas precisam ser tomadas para a elaboração da solução do problema proposto. Co-realização é o trabalho cooperativo em essência. Compartilhamento é um conceito associado à divisão e à distribuição de informações entre membros do grupo. Comunicar, negociar, compartilhar e trabalhar concomitantemente com muitas pessoas pode gerar situações de perda de controle dos processos. Portanto, coordenação é um processo-chave para colocar ordem nas transações em espaços e tempos virtuais.

Fuks e seus co-autores (2002) propuseram o modelo ilustrado na Figura 2.1, modelo de colaboração 3C, que é um refinamento do modelo 3C apresentado originalmente em (Ellis *et al.*, 1991) e difundido na literatura, como por exemplo, em (Borghoff & Schlichter, 2000). Esse modelo considera os processos de comunicação, cooperação (co-realização) e coordenação como fundamentais para o trabalho colaborativo. Além desses três processos, o modelo 3C ressalta a importância de um quarto processo, a percepção, que é vital para o trabalho colaborativo e que sem o qual esse trabalho fica descoordenado e perde em qualidade e eficiência. Percepção (*Awareness*) é o conhecimento geral sobre o grupo e suas atividades. É o conhecimento sobre o que aconteceu, o que vem acontecendo, o que está se passando no presente e o que poderá acontecer dentro das atividades do grupo, e sobre o próprio grupo, seus objetivos e sua estrutura (Pinheiro *et al.*, 2002).

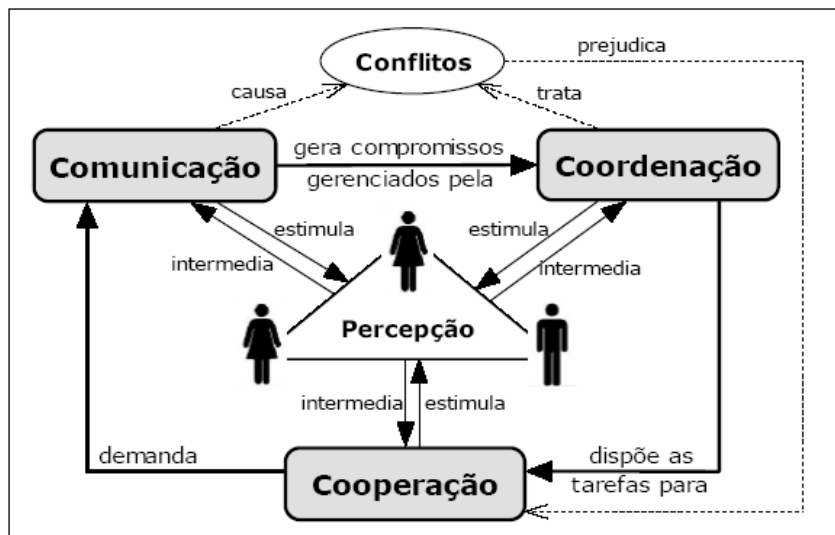


Figura 2.1. Modelo de Colaboração 3C (Fuks *et al.*, 2002).

De acordo com este modelo, para colaborar, os indivíduos têm que dialogar (se comunicar), organizar-se (se coordenar) e operar em conjunto num espaço compartilhado (cooperar). As trocas ocorridas durante a comunicação geram compromissos que são gerenciados pela coordenação, que por sua vez organiza e dispõe as tarefas que são executadas na cooperação. Ao cooperar os indivíduos têm necessidade de se comunicar para renegociar e para tomar decisões sobre situações não previstas inicialmente. Isto mostra o aspecto cíclico da colaboração. Cada evento ocorrido durante a comunicação, a coordenação e a cooperação gera informações que são disponibilizadas ao grupo através de elementos de percepção.

Nas próximas subseções são detalhados os processos de colaboração e suas inter-relações. Vale Salientar que apesar da separação desses conceitos para efeito de análise,

nem sempre é possível considerá-los monoliticamente, uma vez que são intimamente dependentes e inter-relacionados.

2.2.1 Comunicação

Obviamente, a comunicação entre os participantes de um grupo é fundamental para que a colaboração ocorra, sendo, portanto, um elemento essencial para o trabalho colaborativo. O processo de comunicação também é fundamental para a coordenação, pois sem uma comunicação eficiente, os participantes terão dificuldade de harmonizar seus esforços para a conclusão das tarefas. A comunicação também intermedeia o processo de negociação que ocorre durante as discussões sobre a forma como o trabalho está sendo realizado, sobre as dependências entre as tarefas e sobre as responsabilidades dos participantes na realização dessas tarefas.

Para realizar comunicação, os participantes do grupo tipicamente usam duas habilidades humanas fundamentais (Schlichter, 1997): a comunicação direta com os outros participantes, e comunicação indireta. Na comunicação direta, um emissor transfere explicitamente a informação para os receptores. Já na comunicação indireta, o agente da comunicação manipula objetos compartilhados e transmite, sem intenção de fazê-lo, uma mensagem aos outros participantes. A manipulação é observada por todos e a mensagem é decodificada. Um documento é um exemplo típico de objeto em sistemas computacionais.

Geralmente, essas duas formas de comunicação são utilizadas em conjunto. Por exemplo, na comunicação direta, é comum os participantes usarem objetos compartilhados como um modo fácil de estabelecer referência de identidade (Miles, 1993). De forma semelhante, ao trabalharem com objetos compartilhados, os participantes fazem uso complementar da comunicação direta. Na Figura 2.2, estão ilustradas as diferenças entre comunicação direta e indireta.

Quanto ao sincronismo da interação resultante do processo de comunicação, Reinhard e seus co-autores (1994) definem dois tipos de comunicação: a comunicação síncrona e a comunicação assíncrona. Na comunicação síncrona, todas as tarefas realizadas por um participante de uma sessão cooperativa (usuários trabalhando simultaneamente) são disponibilizadas imediatamente para os outros participantes. A edição concorrente de objetos comuns é permitida e resultados intermediários são imediatamente disponibilizados. Na comunicação assíncrona, por sua vez, os participantes trabalham isoladamente e, ao terminarem suas tarefas, disponibilizam os resultados alcançados para que possam ser utilizados nas discussões entre os grupos. Essa forma de comunicação é

caracterizada pelas ferramentas que suportam a argumentação e o armazenamento de decisões de projeto, estabelecendo um ambiente propício para apresentar idéias ou opiniões, propor alterações e levantar questões acerca do produto sendo desenvolvido (Souza *et al.*, 1997).

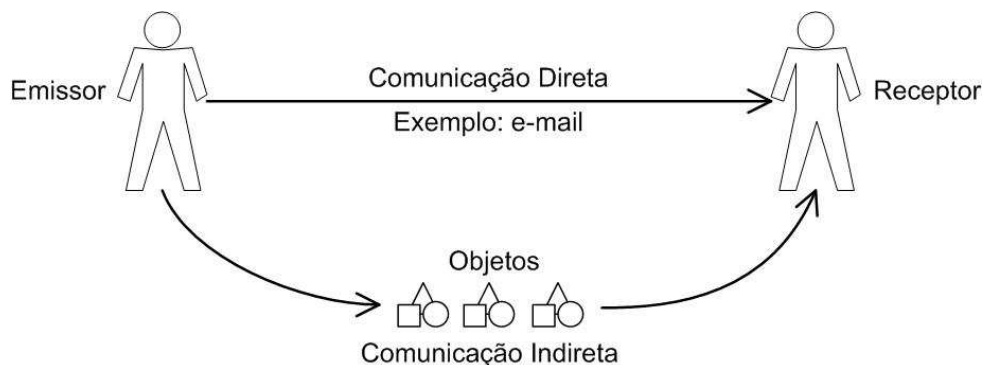


Figura 2.2. Comunicação direta e indireta (baseada em Xu, 2000).

Naturalmente comunicação assíncrona pode ser direta, (exemplo *e-mail*) ou indireta (exemplo documentos compartilhados por um sistema de banco de dados). Porém, comunicação síncrona deve ser principalmente direta (exemplo reunião face-a-face). Assim, sistemas de *groupware* podem ser classificados tanto em termos de sincronismo (síncrono ou assíncrono) quanto em termos da forma como a mensagem enviada pelo emissor é percebida pelo receptor (comunicação direta e ou indireta) (vide Figura 2.3) (Xu, 2000).

Alguns exemplos de ferramentas de comunicação atualmente utilizadas são: *e-mail*, lista de discussão, fórum, ferramentas de CSCA (*Computer Supported Collaborative Argumentation*), ferramentas de votação, mensagem instantânea, *chat*, *eletronic brainstorming*, conferência multimídia e telefone. (Long & Baecker, 1997; Fuks *et al.*, 2002).

2.2.2 Negociação

A negociação é uma forma mais sofisticada de comunicação em que se faz necessária alguma estruturação. Nesse caso, a comunicação não é totalmente livre, sendo submetida ao cumprimento de determinadas regras. Por exemplo, quando o assunto da comunicação é polêmico e gera alguma discussão, é preciso utilizar mecanismos que permitam a dissolução de conflitos, através da tentativa de acordo ou consenso, como por exemplo, a votação, ou elementos (pessoas) facilitadores da discussão. Assim, para que a cooperação seja bem sucedida, deve-se evitar o “mal estar” entre colegas de trabalho, identificando os momentos em que se deve aplicar a negociação (Borges *et al.*, 1995).

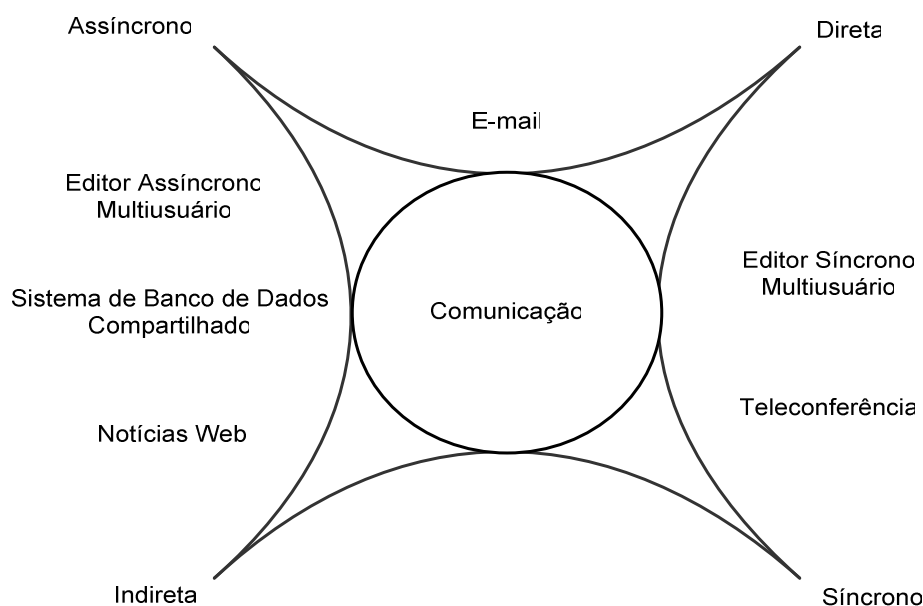


Figura 2.3. Classificação de sistemas de *groupware* em termos de comunicação síncrona, assíncrona, direta e indireta (baseada em Xu, 2000).

O tema, o objetivo, as questões, o "custo" das decisões, o tempo disponível, as pessoas envolvidas, o conhecimento comum, as hierarquias, os locais de reunião e as características culturais são fatores que definem o contexto de uma negociação e devem ser cuidadosamente observados para se fazer uma opção tecnológica de instrumento de suporte a esse processo. Segundo Barros (1994), o desenvolvimento de tecnologias para negociação deve estar estreitamente conectado a pesquisas sócio-afetivo-cognitivas.

Os Sistemas de Suporte à Decisão em Grupo (*Group Decision Support Systems - GDSS*) foram uma das primeiras aplicações no domínio de trabalho cooperativo que dão suporte ao processo de Negociação. Muitos dos sistemas para trabalho cooperativo têm por objetivo auxiliar ou simular reuniões, no pressuposto de que as reuniões são feitas para as pessoas trabalharem e negociarem juntas visando a solução de problemas.

2.2.3 Coordenação

Segundo Borges e seus co-autores (1995), a coordenação é um processo chave para que a colaboração seja bem sucedida. Em alguns casos de trabalho colaborativo, a ausência de um elemento centralizador pode levar à dissidência de alguns membros do grupo. Por conseguinte, dependendo das tarefas e dos participantes do grupo envolvidos no trabalho cooperativo, é necessário identificar o suporte necessário para a coordenação, provendo mecanismos de controle sobre os processos de comunicação e negociação, e de monitoração sobre o produto do trabalho cooperativo.

Ainda segundo Borges e seus co-autores, o trabalho colaborativo em muitos casos envolve o desenvolvimento de objetos como produto ou objetivo final. Durante o desenvolvimento do trabalho, esse objeto é compartilhado e manipulado pelos participantes do grupo, o que caracteriza os dois últimos processos identificados por Barros (1994): a co-realização (colaboração) e o compartilhamento. O processo de coordenação executa um papel importante, na medida em que estabelece regras de acesso ao objeto como, por exemplo, o controle de concorrência e o controle de versão.

O processo de coordenação das atividades do grupo pode ser dividido em três etapas: planejamento, acompanhamento das atividades planejadas e análise do resultado obtido. Na fase de planejamento o grupo necessita discutir sobre a definição de metas, prazos, mapeamento dessas metas em tarefas, seleção dos participantes e distribuição das tarefas entre eles. Geralmente o planejamento é concluído antes do trabalho colaborativo se iniciar (Malone & Crowston, 1990).

A fase de acompanhamento de atividades é a mais importante da coordenação. Por ser muito dinâmica, essa fase precisa ser renegociada de maneira quase contínua ao longo de todo o tempo. A fase da análise ocorre após o término das tarefas, e envolve a avaliação das tarefas realizadas, a verificação da satisfação dos requisitos identificados durante o planejamento e a documentação do processo de colaboração (histórico do processo).

Segundo Rezende (2003), algumas atividades envolvendo mais de um indivíduo não exigem um planejamento formal. Atividades ligadas às relações sociais são bem controladas pelo chamado protocolo social, caracterizado pela ausência de qualquer mecanismo de coordenação explícito entre as atividades e pela confiança nas habilidades dos participantes de mediar interações. Por outro lado, atividades mais diretamente voltadas para o trabalho colaborativo (e não para as relações sociais) exigem sofisticados mecanismos de coordenação para garantir o sucesso da colaboração.

Para concretizar a coordenação é preciso ter uma definição clara de tarefas, atividades colaborativas e interdependências (Raposo & Fuks, 2002). Tarefas são os elementos que compõem as atividades colaborativas e estão ligadas por interdependências (vide Figura 2.4). Elas podem ser atômicas ou compostas de subtarefas. Um grupo de subtarefas pode ser considerado uma tarefa em um nível de abstração mais alto quando elas não apresentam interdependências com tarefas externas a esse grupo. Isso garante a modelagem de atividades colaborativas em vários níveis de abstração.

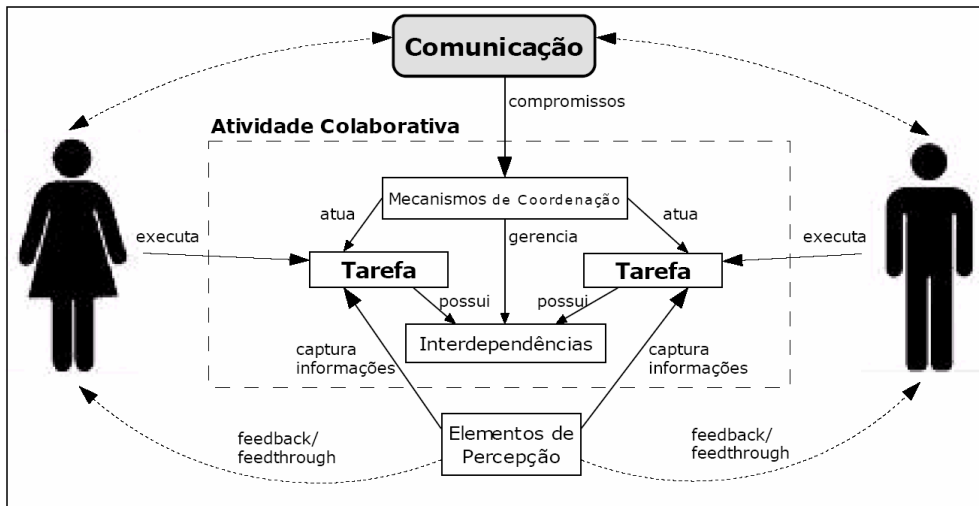


Figura 2.4. Modelando a Coordenação (Fuks *et al.*, 2002).

A coordenação pode ocorrer em dois níveis, o de atividades (temporal) e o de objetos (Ellis & Wainer, 1994). No nível temporal, a coordenação define a seqüência de tarefas que compõem uma atividade. No nível de objetos, a coordenação descreve como lidar com o acesso seqüencial ou simultâneo de múltiplos participantes a um mesmo conjunto de objetos de cooperação.

Com base na separação das tarefas e suas interdependências, é possível caracterizar diferentes tipos de interdependências e identificar os mecanismos de coordenação que os gerenciam, criando um conjunto de interdependências e respectivos mecanismos de coordenação que sejam capazes de atender uma grande gama de aplicações colaborativas (Malone & Crowston, 1990). Em (Raposo *et al.*, 2000), encontra-se um conjunto de mecanismos de coordenação que usa Redes de Petri para modelar as tarefas e o tratamento das interdependências.

Alguns tipos de aplicações que dão suporte ao processo de coordenação são: gerenciamento de fluxo de trabalho (*workflow*), *learningware*, jogos multiusuários e ferramentas de autoria e de desenvolvimento de *software* colaborativo.

2.2.4 Compartilhamento

Comunicação e coordenação, apesar de vitais, não são suficientes. É necessário espaço compartilhado para criar entendimento compartilhado (Schrage, 1995). Compartilhamento é um conceito associado com dividir e distribuir algo com outros. Compartilhar conhecimento, informação, idéias ou propósitos é essencial para atividades colaborativas. Ações colaborativas baseiam-se no compartilhamento dos objetos relacionados com a realização de uma atividade. Quando o computador é usado para dar suporte a essas ações,

surtem problemas envolvendo o acesso, o uso e a alteração indevida dos arquivos desses objetos (Barros, 1994).

Para evitar uso indevido dos arquivos, são oferecidos mecanismos de proteção aos dados e associados privilégios aos participantes, garantindo que apenas as pessoas autorizadas possam fazer acesso aos dados, e apenas àqueles que estão relacionados com as tarefas a elas atribuídas. Outra forma de evitar o uso indevido dos objetos é a atribuição de direitos de acesso aos objetos. O direito de acesso a um objeto é um mecanismo que possibilita preservar a consistência de uma informação quando um ou vários indivíduos atuam sobre um objeto, ou parte dele, ao mesmo tempo ou em tempos distintos. Uma outra forma de proteção é permitir as ações sobre cópias dos objetos. Depois de realizadas operações sobre diversas cópias, elas são utilizadas para gerar uma versão atualizada do objeto que representa o efeito de todas as operações realizadas separadamente. Mecanismos de suporte a versões estão geralmente presentes nos sistemas cooperativos (Barros, 1994).

Em ambientes assíncronos, como não há a obrigatoriedade dos membros estarem trabalhando simultaneamente, o foco maior de atenção está nos objetos compartilhados. É através deles que a comunicação entre os membros se dará, isto é, através da manipulação e do histórico de manipulação desses objetos (Pinheiro *et al.*, 2002). Esse histórico vai fornecer informações sobre o andamento do trabalho em grupo, mostrando o que foi feito sobre os objetos compartilhados, e criando, assim, um contexto para as atividades de cada participante. Segundo Dourish (1997), o objeto compartilhado é essencialmente o único espaço compartilhado disponível aos participantes nesses ambientes e representa também a informação chave na colaboração assíncrona.

O trabalho colaborativo assistido por computador é utilizado em diversas áreas de aplicação – autoria de textos, geração de cursos, projetos participativos em engenharia, reuniões, desenvolvimento de software, legislação, etc. Assim, os objetos comumente compartilhados pelos grupos são do tipo: texto, desenho, som, imagem, vídeo, programas, planilhas e bases de dados.

2.2.5 Colaboração

O processo de colaboração é a operação conjunta dos membros do grupo no espaço compartilhado, visando à realização de tarefas. Eles cooperam produzindo, manipulando e organizando informações e construindo e refinando objetos (Fuks *et al.*, 2002). Para atuar nesses objetos, os membros do grupo contam com elementos de expressão que são

elementos de percepção os quais fornecem informações sobre as alterações realizadas no espaço compartilhado (vide Figura 2.5).

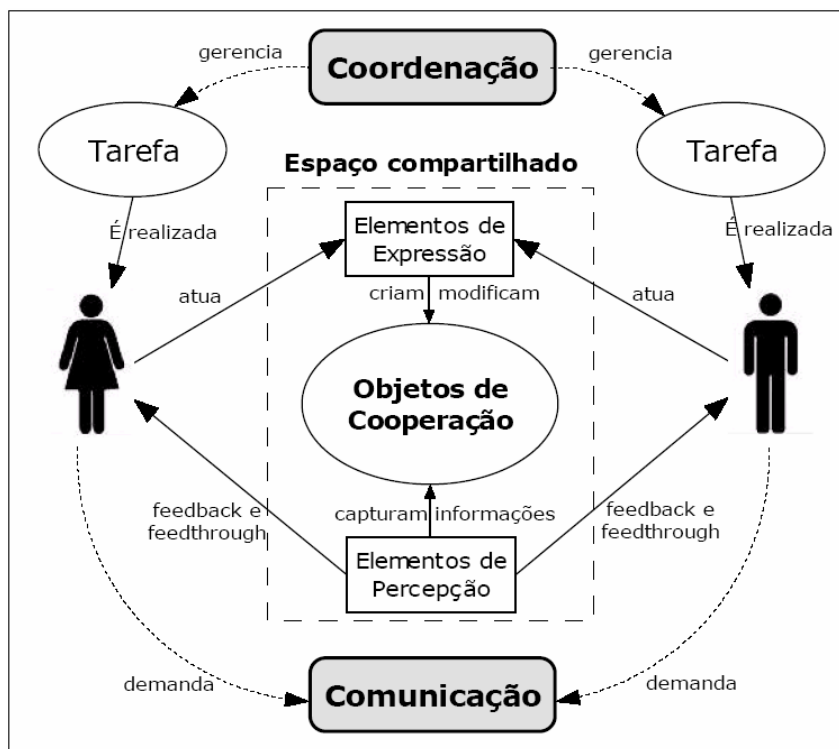


Figura 2.5. Modelando a Colaboração (Fuks *et al.*, 2002).

Os indivíduos buscam nos elementos de percepção as informações necessárias para criar um contexto compartilhado e antecipar ações e necessidades com relação às metas da colaboração. Isso possibilita identificar as intenções dos companheiros do grupo, para que seja possível auxiliá-los quando necessário. Essas interações geram novos acontecimentos e informações no espaço compartilhado, que por sua vez irão refletir-se nos elementos de percepção nos quais os indivíduos buscarão conhecimentos para se comunicar e coordenar interações posteriores.

Devem-se prever onde as informações de percepção são relevantes, como elas podem ser obtidas ou geradas, onde elementos de percepção são necessários, como apresentá-los e como dar aos indivíduos o controle sobre eles. O excesso de informações pode causar sobrecarga e dificultar a colaboração. Assim, para evitar a sobrecarga, é necessário balancear a necessidade de fornecer informações com a de preservar a atenção sobre o trabalho.

2.2.6 Percepção

Percepção pode ser entendida como a contextualização das atividades individuais através da compreensão das atividades realizadas por outras pessoas (Araújo *et al.*, 1997). O

processo de percepção refere-se a ter conhecimento das tarefas do grupo, saber o que aconteceu, o que está acontecendo ou o que poderá vir a acontecer, além do próprio conhecimento do que é esse trabalho e o grupo. Generalizando, "percepção significa uma compreensão do estado total do sistema, incluindo atividades passadas, status atual e opções futuras" (Sohlenkamp, 1998).

O processo de percepção torna-se central para os processos de comunicação, coordenação e cooperação de um grupo de trabalho, pois os indivíduos tomam ciência das mudanças causadas no ambiente pelas ações dos participantes, e redirecionam as suas atitudes (Rezende, 2003).

É importante para a realização eficiente do trabalho colaborativo que cada participante conheça o progresso do trabalho dos companheiros: o que foi feito, como foi feito, o que falta para o término, quais são os resultados preliminares, etc. As informações de percepção são necessárias principalmente durante a fase dinâmica da coordenação, para transmitir mudanças de planos e ajudar a gerar o novo entendimento compartilhado. Elas ajudam a medir a qualidade do trabalho com respeito aos objetivos e progressos do grupo e a evitar duplicação desnecessária de esforços (Dourish & Belloti, 1992; Neisser, 1976).

Elementos de percepção (Figura 2.5) são os elementos do espaço compartilhado por onde são transmitidas as informações destinadas a prover percepção. Perceber as atividades dos outros indivíduos é essencial para garantir o fluxo e a naturalidade do trabalho, assim como para diminuir as sensações de impessoalidade e distância, comuns nos ambientes virtuais (Fuks *et al.*, 2002). Sem percepção, o trabalho cooperativo coordenado é quase impossível (Sohlenkamp, 1998).

Pinheiro e seus co-autores (2002) realizaram uma análise de vários mecanismos de suporte à percepção e identificaram um conjunto de características importantes que estão relacionadas às seguintes questões que identificam, cada uma, aspectos vitais para o fornecimento de percepção dentro de ferramentas de *groupware* síncronas e assíncronas:

- ♦ **O quê**, questão que envolve o conhecimento das tarefas que cada membro está executando, o conhecimento dos objetivos dessas tarefas, o conhecimento das inter-relações entre elas e o conhecimento dos objetos utilizados para concretizá-las;
- ♦ **Quando**, questão que envolve o conhecimento de quando ocorrem os eventos geradores das informações de percepção e de quando se dá a apresentação dessas informações;
- ♦ **Onde**, questão que se refere ao local no qual cada membro está trabalhando para gerar e apresentar as informações;

- ♦ **Como**, questão que indica de que forma as informações são apresentadas aos usuários (interface);
- ♦ **Quem**, questão que envolve o conhecimento da presença de participantes no ambiente, das identidades desses participantes e da identidade do autor de uma determinada ação; e
- ♦ **Quanto**, questão que destaca uma característica essencial que deve ser observada no suporte à percepção – a quantidade ideal de informações que deve ser apresentada ao usuário, a fim de lhe prover percepção sobre o grupo e suas atividades.

2.3 Taxonomias de *Groupware*

Em função das características de tecnologias de comunicação, distribuição de tempo e espaço, número de integrantes do grupo como também tipo de comunicação, aplicações CSCW podem ser classificadas em diferentes taxonomias, dentre essas está o modelo de tempo-espaço (Elis *et al.*, 1991) e a taxonomia baseada em funcionalidades proposto por Teufel & Teufel (1995).

2.3.1 Taxonomia Tempo-Espaço

Nessa taxonomia, os *groupowares* são classificados segundo as dimensões espaço-temporais. Sistemas *Groupware* podem ser projetados para ajudar nos encontros de membros de um grupo. Quando os membros não estão geograficamente dispersos, o *groupware* contribui para os encontros face-a-face, porém os mecanismos de comunicação oferecidos são de uso facultativo. Nos casos em que os membros encontram-se geograficamente dispersos, os mecanismos de comunicação oferecidos pelo *groupware* são essenciais. Esses mecanismos de comunicação podem ser projetados para ampliar a comunicação e a colaboração entre os membros, oferecendo possibilidades de interação em tempo-real ou de forma assíncrona (Elis *et al.*, 1991). Essas considerações de tempo espaço sugerem as quatro categorias de *groupware* representadas pela matriz 2x2 mostrada na Figura 2.6.

	Mesmo Tempo	Tempo Diferente
Mesmo Local	Interação face-face	Interação Assíncrono
Local Diferente	Interação síncrona distribuída	Interação assíncrona distribuída

Figura 2.6. Esquema para classificação espaço-temporal das aplicações em trabalho cooperativo auxiliado por computador.

Nem todas as aplicações se encaixam perfeitamente em uma única célula, por exemplo, um sistema de autoria de textos pode ser projetado para abranger as quatro células. No entanto, nem todas as aplicações podem abranger todas as células, pois uma tecnologia adequada para uma célula pode provocar impactos negativos se adotada em uma atividade de outra célula. Por exemplo, sistemas de correio eletrônico são amplamente adotados e reconhecidos como uma tecnologia de sucesso na área de comunicações entre indivíduos e grupos, mas somente o correio eletrônico não é suficiente como mecanismo de comunicação em reuniões face-a-face, que requerem outras tecnologias tais como as de suporte a decisões (Barros, 1994).

Existem outros tipos de taxonomias baseados no modelo da Figura 2.6, como por exemplo, o modelo de Grundin (Grundin, 1994) que inclui uma categoria intermediária na classificação tempo/espaço. Essa categoria considera que o local ou o momento no tempo podem ser determinados ou não, de modo que uma atividade pode acontecer, por exemplo, em um momento bem definido enquanto que o local pode permanecer indefinido. Outro modelo muito difundido na literatura é o de Nunamaker (Nunamaker *et al.*, 1991) que considera importante o tamanho do grupo, especialmente no contexto de sistemas de suporte a reuniões, e, assim, propõe mais uma dimensão na classificação tempo/espaço, transformando-a em tempo/espaço/tamanho.

2.3.2 Taxonomia Baseada em Funcionalidades

Nessa taxonomia, os sistemas *groupware* são definidos segundo suas características funcionais. Semelhante à anterior, essa classificação não é exclusiva, isto é, suas categorias se sobrepõem, podendo, assim, uma mesma aplicação enquadrar-se em duas categorias

distintas. Existem, na literatura, diferentes classificações de *groupware* segundo as suas funcionalidades (Teufel & Teufel, 1995; Elis *et al.*, 1991; Borges *et al.*, 1995). Entretanto, a mais difundida é a taxonomia proposta por Elis e seus co-autores (Elis *et al.*, 1991) que serviu de inspiração para outras taxonomias, como a proposta por Borges e seus co-autores (Borges *et al.*, 1995). A seguir, será considerada a classificação originalmente proposta por Ellis, que possui pontos em comum com a proposta por Borges e seus co-autores.

Sistemas de Mensagens

Esses sistemas suportam troca assíncrona de mensagem de texto e arquivos entre os usuários do grupo. Com o surgimento da Internet, houve um grande aumento da utilização desse tipo de *groupware*, tornando as aplicações CSCW mais comuns. Os exemplos mais comuns nessa categoria são: os sistemas de e-mail, os sistemas de lista de discussão e os sistemas de quadros de aviso (BBS – *Bulletin Board Systems*). Essas aplicações podem ser utilizadas em qualquer tipo de encontro (face-a-face ou disperso).

Editores de Autoria Colaborativa

São sistemas em que vários usuários podem compor e editar textos e gráficos conjuntamente de forma síncrona ou assíncrona. Em ambas as formas o sistema deve garantir que todos os co-autores possuam conhecimento da participação e das atividades que estão sendo realizadas pelos outros, mesmo que esses estejam dispersos geograficamente (Macedo, 1999).

Em edições síncronas, vários usuários de um grupo podem editar o mesmo objeto simultaneamente. Geralmente o objeto editado é dividido em segmentos, por exemplo, um documento pode ser dividido em seções ou um projeto de software, em módulos. A maioria dos editores multiusuários permite acesso concorrente de leitura a qualquer elemento, mas somente um usuário tem acesso de escrita a cada elemento. Entretanto, existem sistemas de edição que gerenciam aspectos de sincronização e concorrência de acesso de leitura e escrita a todos os elementos simultaneamente de forma transparente ao usuário. Entre os sistemas de autoria colaborativa existem: Grove (Ellis *et al.*, 1991), QUILT (Dourish & Bellotti, 1992), CoMediA (Santos, 1993), SERPIA (Haake & Wilson, 1992) e Shared Books (Lewis & Hodges, 1988).

Sistemas de Suporte à Tomada de Decisão em Grupo e Salas de Reuniões Eletrônicas

Os sistemas de suporte à tomada de decisão em grupo (GDSS) são tipos de *groupware* que implementam facilidades para resolução de problemas em grupo. O objetivo desses sistemas é melhorar a produtividade dos encontros para facilitar a tomada de decisão, tanto acelerando o processo de decisão como melhorando a qualidade dos resultados das decisões tomadas (kraemer & King, 1988). O IBIS (Conklin & Begeman, 1987), QOC (Shum & Hammond, 1994) e o DRL (Lee, 1990) são exemplos de GDSS.

Salas de reuniões eletrônicas (EMS – *Electronics Meeting Systems*) são sistemas que oferecem ambientes com suporte de hardware e software para apoiar reuniões face-a-face, podendo fazer uso de estações de trabalho interligadas em rede, uso de projetores e equipamentos de áudio e vídeo. Essas salas são utilizadas pelos sistemas de GDSS, fornecendo mecanismos para a geração e organização de idéias e para a preparação de pautas de reuniões, a fim de facilitar a tomada de decisão. O sistema PlexCenter (*Planning and Decision Support Laboratory*) da Universidade do Arizona é um exemplo de sala eletrônica (Applegate *et al.*, 1986).

Sistemas de Conferência

Com o grande avanço tecnológico nas áreas de Redes e Comunicação, permitindo transmissão de áudio e vídeo em tempo real, surgiram os sistemas de conferência. Esse tipo de *groupware* provê três novas formas de realização de conferências: sistemas de conferência em tempo real (síncronos), sistemas de conferência assíncrona e sistemas de teleconferência.

Os sistemas de conferência em tempo real permitem que um grupo de usuários interaja sincronamente através de terminais de computadores ou estações de trabalho conectadas em rede, estando ou não no mesmo espaço físico. Nesse tipo de conferência, os participantes utilizam também recursos de áudio para se comunicarem. O MS NetMeeting (NetMeeting, 2005) é um exemplo desse tipo de sistema.

Os sistemas de conferência assíncrona permitem que os integrantes do grupo participem de encontros de acordo com sua própria disponibilidade. Nesse contexto, os participantes apresentam suas contribuições e o sistema as armazena, possibilitando o acesso aos demais integrantes para que eles as analisem. Entretanto, esse *groupware* não oferece condições para extração de resultados dessas discussões, como fazem os GDSS. O USENET (USENET, 2004) é um exemplo desse tipo de sistema.

Os sistemas de teleconferência necessitam recursos de telecomunicação, como transmissão de vídeo e áudio, para a comunicação em tempo real entre os participantes. Conferências multimídia são os maiores representantes deste tipo de conferência (Pequeno *et al.*, 2003).

Agentes Inteligentes

Nesse tipo de *groupware* são utilizados agentes inteligentes, mecanismos automatizados responsáveis por um conjunto de tarefas específicas, como por exemplo, avisos de chegada de mensagens, respostas automáticas a questões ou eventos gerados pelos participantes e sugestões de ações a serem tomadas. Alguns exemplos de sistema que utilizam agentes inteligentes são *Information Lens* e *Object Lens* (Crowston & Malone, 1988).

Sistemas de Coordenação

Os sistemas de coordenação gerenciam e organizam interdependências entre as atividades realizadas pelos usuários para atingir um objetivo comum (Malone & Crowston, 1990). Esses sistemas ainda são responsáveis por garantir que as interdependências das atividades não sejam violadas durante o desenvolvimento do trabalho colaborativo.

Existem quatro abordagens para esse tipo de sistema: formulário, procedimento, conversação e estrutura de comunicação. Os sistemas orientados a formulário são denominados de sistemas de *Workflow* ou de controle de fluxo de trabalho. Os sistemas orientados a procedimentos vêem os procedimentos de uma organização como processos programáveis, isto é, os procedimentos são encadeados em seqüência, de forma que o resultado de um procedimento alimente o seguinte, e assim sucessivamente até que todos tenham sido executados. Os sistemas orientados a conversação controlam os pedidos e os compromissos das pessoas através da monitoração de sua correspondência eletrônica. E finalmente, os sistemas orientados a comunicação estruturada, controlam as atividades de uma organização baseados no relacionamento entre as funções dos indivíduos.

Exemplos de sistemas de coordenação são: os sistemas de *Workflow*, os quais auxiliam grupos de pessoas na execução de procedimentos de trabalho; e as ferramentas CASE (*Computer Aided Software Engineering*), que auxiliam no processo de desenvolvimento de software.

2.4 Considerações Finais

A área de CSCW está em ascensão. A quantidade e a variedade de trabalhos que vêm sendo gerados em tão pouco tempo comprovam a crescente importância da área. A interdisciplinaridade do tema gera o aparecimento de trabalhos relacionados com a área em diversas conferências e publicações. Vertentes como CSCL surgem como consequência natural desse movimento.

Nesse capítulo foram abordados vários conceitos importantes na área de colaboração assistida por computador, contribuindo para a identificação dos aspectos cruciais na implementação de aplicações colaborativas. Esse trabalho tem como diferencial agrupar de forma sintética os principais processos considerados importantes, por pesquisadores da área, para o desenvolvimento de sistemas que apoiem o trabalho colaborativo. Além disso, a identificação das principais taxonomias ajuda o desenvolvedor a situar o campo de atuação de seu projeto.

Apesar dos bons resultados já obtidos com as aplicações de CSCW já desenvolvidas, o principal obstáculo é fornecer interatividade a esses sistemas. O uso de realidade virtual em sistemas colaborativos permite aos participantes interagirem com maior naturalidade, pois através do ambiente virtual é possível a comunicação por meio de imagem, vídeo, texto e áudio. No próximo capítulo, são abordados os ambientes virtuais que apoiam o trabalho colaborativo.

Capítulo 3

Ambientes Virtuais Colaborativos – CVEs

3.1 Introdução

Neste capítulo, são abordados os principais conceitos e características de Ambientes Virtuais Colaborativos. Também são apresentados os requisitos de hardware e software (alguns necessários, outros desejáveis) que devem ser considerados no desenvolvimento de CVEs. E por último, são apresentadas e analisadas plataformas e aplicações de CVEs que são referências para o desenvolvimento de outras aplicações.

Os CVEs utilizam a tecnologia de Realidade Virtual Distribuída para o apoio ao trabalho colaborativo (Benford & Greenhalgh, 2001). Essa tecnologia é a combinação de idéias das áreas de pesquisa em Realidade Virtual, Trabalho Colaborativo Assistido por Computador e Sistemas Computacionais Distribuídos. Da RV, os CVEs utilizaram a idéia de construção de ambientes multidimensionais com alto grau de interatividade, podendo utilizar tecnologias de interface avançadas como capacetes de visualização (HMD's - *Head-Mounted Displays*), luvas de dados (*data-gloves*) e sensores de rastreamento (*body-trackers sensors*). Da área de CSCW, veio a idéia de desenvolvimento de sistemas que provê suporte apropriado para o trabalho colaborativo entre um grupo de pessoas. E por último a área de Sistemas Distribuídos investiu muitos esforços em sistemas computacionais robustos e escaláveis para suportar acesso por múltiplos usuários.

Os CVEs apresentam grande potencial para o suporte ao trabalho colaborativo e, por conseguinte, têm sido desenvolvidos dando grande atenção aos resultados de CSCW. Em particular, eles apresentam características bastante interessantes de comunicação, cooperação e percepção (Fuks *et al.*, 2002). A comunicação aparece na forma de ferramentas de bate-papo, que acompanham os CVEs mais simples, ou na forma de áudio e videoconferência, que acompanham os CVEs mais sofisticados. A capacidade de comunicação é ampliada em CVEs pela noção de espaço oferecida (por exemplo, um avatar se aproximando do outro pode indicar que ele deseja se comunicar). A colaboração faz parte da própria natureza dos CVEs, que são literalmente espaços de trabalho compartilhados. Por imitarem metáforas do mundo real, CVEs trazem possibilidades de percepção que não são triviais em aplicações de *desktop*. Por exemplo, os avatares e suas

localizações fornecem informações diretas sobre os usuários presentes no ambiente e suas atividades. A movimentação dos mesmos antecipa intenções dos usuários e revela a expectativa que os demais usuários têm a seu respeito.

A seguir, na Seção 3.2, são discutidas as características de CVEs; na Seção 3.3, os requisitos para o desenvolvimento de CVEs são abordados; na Seção 3.4, algumas arquiteturas de CVEs são discutidas; na Seção 3.5, algumas aplicações de CVEs são analisadas; e por último, na Seção 3.6, tecem-se algumas considerações finais.

3.2 Características de CVEs

Pode-se observar que os CVEs utilizam os mesmos princípios dos Ambientes Virtuais em Rede (NVEs – *Networked Virtual Environments*), apenas orientando o enfoque e criando características especiais para o trabalho colaborativo (Rosa-Júnior, 2001). No contexto deste trabalho, um CVE é considerado um NVE com suporte à colaboração entre seus usuários. Assim, muitas características de CVEs são herdadas dos NVEs, que são observados por cinco características comuns (Singhal & Zyda, 1999):

- ♦ **Uma sensação de espaço compartilhado**, experimentada por todos os participantes e provocada pela ilusão de estarem situados no mesmo ambiente. Esse espaço compartilhado representa um local comum (real ou fictício) dentro do qual podem acontecer outras interações. O espaço compartilhado deve apresentar as mesmas características a todos os participantes. Por exemplo, todos os participantes devem perceber as mesmas temperaturas e condições climáticas, assim como a mesma acústica (Singhal & Zyda, 1999). Apesar de não serem necessariamente apresentados graficamente, os CVEs mais eficientes provêm uma representação gráfica tridimensional do espaço compartilhado;
- ♦ **Uma sensação de presença**, provocada pela capacidade de ver e de ser visto dentro do espaço compartilhado. Essa capacidade se deve ao fato de que, ao entrar nesse espaço, cada participante é representado por uma personificação digital chamada avatar.¹ Quando o usuário sai do espaço compartilhado, os outros avatares dão conta de sua ausência;
- ♦ **Uma percepção temporal**, decorrente da capacidade dos participantes de observarem os eventos que ocorrem no espaço compartilhado pelas

¹ O termo avatar tem sua origem no hinduísmo, e designa qualquer encarnação de uma divindade na forma de um homem ou de um animal.

movimentações, comportamentos e interações entre os avatares e os objetos do ambiente. Esses eventos são observados no instante em que eles ocorrem, exigindo que o CVE possibilite que interações ocorram em tempo real;

- ♦ **Um modo de comunicação**, que pode ser uma combinação de linguagens corporais (comunicação através de gestos), sonoras (comunicação através da voz) ou textuais (comunicação por meio da escrita), que permitam a troca de informação entre os participantes. Essa comunicação adiciona um sentimento necessário de realismo a qualquer ambiente simulado, sendo um componente fundamental de sistemas de engenharia ou de treinamento; e,
- ♦ **Um modo de compartilhar**, que determina a habilidade dos usuários interagirem realisticamente entre si ou com o próprio ambiente virtual, conferindo o verdadeiro poder dos CVEs. Na simulação de uma batalha ou de um jogo, por exemplo, os usuários precisam atirar uns nos outros, ou podem colidir uns com os outros. Os usuários devem ser capazes de manipular objetos² que existem no ambiente, assim como repassá-los a outros participantes. Um projetista de CVEs deve criar mecanismos que possibilitem aos usuários manipular o próprio ambiente. Por exemplo, construir um abrigo, desenhar em um quadro branco, ou até mesmo destruir o ambiente.

CVEs são considerados um dos sistemas de software mais complexos já construídos (Stytz, 1996), pois apresentam características de vários outros tipos de sistemas computacionais. Por exemplo, segundo Singhal & Zyda (1999), CVEs apresentam características de sistemas distribuídos, de aplicações gráficas e de aplicações interativas. CVEs são sistemas distribuídos por terem que lidar com problemas de gerência de recursos da rede, perda de dados, falha de comunicação e concorrência. CVEs também são aplicações gráficas por terem que manter a apresentação gráfica em taxas aceitáveis, compartilhando o processamento entre a renderização e as outras tarefas do sistema. E finalmente, CVEs são aplicações interativas por terem que processar em tempo real as ações dos usuários, para que os participantes vejam o ambiente virtual como se ele existisse localmente, mesmo que os outros participantes estejam distribuídos remotamente na rede. Ainda segundo Singhal & Zyda, CVEs tornam-se ainda mais complexos por terem que suportar outros serviços, como por exemplo:

² Segundo Xu (2000), objeto representa um produto artificial, ou seja, qualquer elemento que não represente seres vivos.

- ♦ **Persistência de dados**, conseguida através da interligação dos CVEs com sistemas de banco de dados para armazenar de forma persistente as informações do ambiente virtual. Nessa base de dados estão armazenadas, por exemplo, informações detalhadas sobre o relevo do terreno do ambiente, a localização dos objetos no ambiente, e a configuração inicial do CVE;
- ♦ **Autenticação**, para verificar se uma dada transação pode ser completada no CVE. Por exemplo, as conexões de usuários autorizados precisam ser autenticadas. Através do mecanismo de autenticação, CVEs podem interagir com sistemas de comércio eletrônico ou com outros tipos de sistemas de transações; e,
- ♦ **Registro de log**, para permitir que os CVEs façam registro dos eventos em tempo real em uma base de dados persistente. Essa tarefa é complicada, pois o estado do CVE pode estar distribuído na rede.

3.3 Requisitos para Desenvolvimento de CVEs

Segundo Çapin e seus co-autores (1999), a essência de um CVE é a capacidade de representação dos usuários para que sejam identificados explicitamente uns pelos outros dentro de um espaço compartilhado. Além disso, eles devem ter liberdade para mover-se dentro desse espaço, encontrando e interagindo com outros usuários e também manipulando objetos e informações de interesse comum. Assim, sempre que um evento provocar alteração do estado de uma das cópias do ambiente virtual compartilhado, ele é transmitido automaticamente aos computadores utilizados pelos usuários, para que as cópias do ambiente virtual sejam atualizadas. Dessa forma, a consistência geral do CVE é mantida e os participantes têm a impressão de compartilhar um único ambiente virtual. Essa impressão de compartilhamento estimula fortemente a colaboração entre os participantes para a execução das mais variadas tarefas.

O desenvolvimento desses sistemas, que, como já mencionado, são sistemas bastante complexos, é uma tarefa complicada e exige a consideração de uma série de requisitos de software e hardware, alguns dos quais são essenciais para o CVE desenvolvido, enquanto que outros são apenas desejáveis. Neste trabalho, foi feito um levantamento dos principais requisitos, a partir da análise dos trabalhos de diversos autores (Stytz, 1996; Çapin, *et al.*, 1999; Zyda, 1996; Macedonia & Zyda, 1997; Singhal & Zyda, 1999; Greenhalgh, 1997; Leite-Junior, 2000; Rodello *et al.*, 2001; Almendra, 2003), e que estão descritos a seguir.

3.3.1 Fidelidade

A fidelidade entre o mundo virtual e o mundo real, segundo Stytz (1996), é mantida em um nível aceitável em um CVE se forem atendidos os seguintes requisitos:

- ♦ **Fidelidade de realismo visual**, para garantir que a percepção visual do usuário acerca do ambiente virtual seja a mais próxima possível da percepção que ele teria se o ambiente fosse real;
- ♦ **Fidelidade de modelagem**, para assegurar as corretas proporções entre os objetos e o realismo de seus movimentos e velocidades, permitindo que seus comportamentos sejam replicados de forma precisa a todos os participantes;
- ♦ **Fidelidade de tempo**, para garantir que seja mínimo o atraso entre a ação de uma determinada entidade virtual (objetos, personagens etc.) e sua retransmissão às cópias de ambiente de todos os participantes;
- ♦ **Fidelidade de informação**, para assegurar que a quantidade e a consistência das informações distribuídas no NVE sejam suficientes para o desenvolvimento de todas as situações previstas e para o suporte à tomada de decisões, tanto pelos diversos participantes do NVE, como pelos possíveis personagens controlados por computadores;
- ♦ **Fidelidade do comportamento de personagens**, para garantir que os comportamentos de possíveis personagens controlados por computadores simulem corretamente as ações desempenhadas por personagens controlados por humanos face às mesmas circunstâncias;
- ♦ **Fidelidade física**, para que o CVE seja dotado dos efeitos físicos existentes no mundo real;
- ♦ **Fidelidade sensorial**, para que as informações visuais, auditivas e háptico-cinestésicas³ apresentadas aos participantes tentem replicar o mundo real da melhor forma possível;
- ♦ **Fidelidade dos dispositivos de entrada**, para que o hardware utilizado para interagir com o CVE responda da mesma forma que as ferramentas do mundo real por ele simuladas; e,

³ Háptico é relativo ao tato, tátil. Cinestesia é o sentido da percepção de movimento, peso, resistência e posição do corpo, provocado por estímulos do próprio organismo.

- ♦ **Fidelidade do sistema**, para que as atividades e respostas dos diversos participantes do CVE (humanos ou máquinas), quando consideradas como um todo, apresentem um comportamento próximo àquele do mundo real.

De acordo com Çapin e seus co-autores (1999), para minimizar a complexidade do CVE e torná-lo mais eficiente, algumas das condições de fidelidade descritas anteriormente podem ser relaxadas ou desconsideradas. Dessa forma, para que as diversas categorias de fidelidade sejam mais bem exploradas, os seguintes aprimoramentos são sugeridos:

- ♦ **Padronizar os comportamentos** de certos objetos no CVE, a fim de minimizar a complexidade da definição do mundo virtual (Ex.: gravidade);
- ♦ **Tratar Colisões**, para melhorar significativamente a qualidade das simulações nos ambientes virtuais;
- ♦ **Atribuir direitos de acesso individualizado**, para minimizar os problemas de manipulação de objetos no mundo virtual e de entrada em determinados recintos virtuais, que ocorrem quando o CVE é compartilhado por um grande número de participantes;
- ♦ **Utilizar técnicas refinadas de seleção e manipulação de objetos**, a fim de melhorar a interação entre o participante do CVE e os elementos do mundo virtual (Boulic & Thalmann, 1996); e,
- ♦ **Explorar estímulos háptico-cinestésicos**, através de mecanismos *force feedback*⁴, para prover aos usuários um maior grupo de sensações e, conseqüentemente, um maior realismo final.

3.3.2 Interação

A interação com o ambiente e a correlação entre o desempenho de atividades do dia-a-dia no mundo físico e a forma como as mesmas são replicadas no mundo virtual (Ex.: caminhadas, manipulação de objetos, etc.) são fatores que contribuem para a sensação de presença dentro do ambiente virtual (Çapin *et al.*, 1999).

Interação, no contexto de ambientes virtuais colaborativos, tem muitas similaridades com a interação do mundo físico. Robinett (1998) classifica os tipos importantes de interação humana com os objetos e com outras pessoas no mundo físico como:

⁴ Técnica que faz uso de dispositivos interativos que podem gerar forças opostas a determinados movimentos humanos.

- ♦ **Observação**, interação que permite uma pessoa examinar seus arredores e captar as informações sobre o ambiente e sobre as pessoas nele presentes em um dado instante;
- ♦ **Movimentação**, permite que a observação seja feita em diferentes partes do mundo e sob diferentes pontos de vista;
- ♦ **Ação**, permite que uma pessoa interfira no estado do mundo, modificando-o; e,
- ♦ **Conversação**, interação que permite uma pessoa comunicar-se verbalmente com outra.

Considerando o contexto de sistemas distribuídos, interação implica em latência mínima, em um senso de presença e na habilidade de acessar e modificar conteúdo (Brutzman, 1998). A instantaneidade com que os eventos ocorrem e são percebidos pelos participantes fortalecem o sentimento de controle e de presença no mundo virtual. No entanto, esse sentimento é mais exacerbado quando o participante tem a habilidade de agir no ambiente, alterando seu conteúdo. Para ampliar ainda mais o sentimento de presença nos CVEs, é importante melhorar a qualidade das interações físicas e sociais dos participantes, fornecendo uma interface mais intuitiva (Pandzic *et al.*, 1996).

Técnicas de interação em CVEs são necessárias para mapear as ações do usuário, capturada através de dispositivos, em ações e comandos de controle correspondentes. Alguns dos tipos de entrada são: a posição das partes do corpo, os comandos de voz, os gestos das mãos, entre outros. O sistema de ambiente virtual responde modificando o estado do ambiente, isto é, modificando a forma, a posição, a cor, e outras propriedades de várias entidades. Dispositivos de resposta provêm uma resposta sensorial aos usuários, estimulando seus sistemas visuais, auditivos, e outros sistemas de percepção (Poupyrev e Ichikawa, 1999).

3.3.3 Comunicação não Verbal

Como mencionado no Capítulo 2, a comunicação entre os participantes de um grupo é fundamental para que a colaboração ocorra. Com a utilização de RV, novas formas de comunicação não-verbal são adicionadas aos sistemas que apóiam o trabalho colaborativo, sendo, portanto, muito exploradas em CVEs. A comunicação não-verbal preocupa-se com as mensagens transmitidas por meio de movimentações corporais ou expressões faciais a outros indivíduos. Essa comunicação não-verbal é essencial para que a interação entre as pessoas se dê com ou sem contato (Thalmann, 2001). Segundo Mania & Chalmers (1998), existem basicamente seis formas de comunicação não-verbal em CVEs:

- ♦ **Expressões faciais** transmitem mensagens indicativas de aspectos da personalidade ou do estado emocional do indivíduo, sendo utilizadas em CVEs através de um conjunto de expressões predefinidas que são associadas ao avatar do usuário;
- ♦ **Olhar** transmite a mensagem indicativa do nível de atenção ou interesse do indivíduo em algum aspecto específico do ambiente ou na conversação;
- ♦ **Gestos**, movimentos com mãos, pés ou cabeça, que, coordenados com a comunicação verbal, suportam varias funções de comunicação;
- ♦ **Postura** transmite mensagens indicativas de comportamentos sociais associados a estados emocionais através de representações corporais;
- ♦ **Auto-representação** indica vários estados físicos ou emocionais do participante tais como ocupação, personalidade, humor, agressividade, formalidade, entre outros, e, em CVEs, é obtida através da própria representação física do avatar (geometria, cor, textura, etc.) e por representações auxiliares indicando atividade ou condições especiais; e,
- ♦ **Contato Corporal** transmite uma mensagem que, dependendo da cultura, pode exprimir determinado tipo de sentimento ou saudação, mas que, em geral, está associado a um significado primitivo de intimidade que produz um estímulo emocional maior.

Comunicação não-verbal é essencial no processo comunicativo, pois o comportamento não-verbal opera de modo inconsciente. A maioria dos desenvolvedores de CVEs falha em reconhecer o amplo leque de funções que tal comunicação revela (Burgoon & Ruffner, 1978).

3.3.4 Suporte à Colaboração

Um CVE deve fornecer não apenas acesso multiusuário simultâneo a um sistema de realidade virtual, mas também deve apoiar explicitamente os processos envolvidos na colaboração assistida por computador (CVE, 2005), discutidos na Seção 2.2.

A coordenação é um processo chave no controle das atividades colaborativas interdependentes desenvolvidas em CVEs. Portanto, mecanismos de coordenação devem ser oferecidos em um CVE para controlar a execução dessas atividades e garantir que a interdependência entre elas não seja violada (Raposo, 2001). Apesar de a maioria dos CVEs realizar atividades dirigidas pelo protocolo social, sem modelo explícito de coordenação, e ficando a cargo dos participantes a mediação das interações; as atividades

colaborativas fortemente acopladas, como as que ocorrem em jogos ou em um procedimento de autoria, requerem mecanismos explícitos de coordenação para serem realizadas com êxito em ambientes virtuais.

3.3.5 Escalabilidade

A escalabilidade é um dos requisitos mais importantes que deve ser levado em conta no desenvolvimento de CVEs. Em geral, ela pode ser medida por sua capacidade de aceitação de entidades participantes, e depende de fatores tais como: a capacidade da rede de dados, o desempenho e o poder de processamento e renderização dos servidores (Singhal & Zyda, 1999). As entidades participantes são os avatares controlados por humanos, os personagens e objetos controlados por computador e os objetos dinâmicos que podem interagir com os usuários.

A complexidade dos CVEs, em teoria, 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 (Singhal & Zyda, 1999). Outro fator que deve ser considerado na complexidade dos CVEs é o comportamento dos objetos. Objetos que podem apenas ser observados pelo usuário implicam em um nível de complexidade menor do que aqueles com os quais o usuário pode interagir, modificando-os, ajustando-os, etc. (Rodello *et al.*, 2001).

A escalabilidade de um CVE também pode ser medida através da quantidade de ambientes compartilhados distintos que um mesmo sistema pode suportar. A partir dessa segunda forma de medir escalabilidade, é possível analisar a primeira sob dois aspectos: a capacidade de aceitação de participantes em um único ambiente compartilhado do sistema e a capacidade total de aceitação de participantes no sistema.

Limitações na escalabilidade de um CVE são provocadas por uma variedade de pontos críticos (gargalos). Quando o número de participantes ativos é elevado, ocorre um grande volume de tráfego na rede, especialmente pela movimentação de pacotes de atualizações 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.

3.3.6 Adequação à Largura de Banda Disponível

CVEs utilizam-se da rede de dados para a troca de informações sobre as mudanças de estado do ambiente virtual, bem como *streaming* de mídias como áudio. À medida que cresce o número de participantes, 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.

Largura de banda e escalabilidade estão intimamente relacionadas. A preocupação com 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 CVE escalável. Almendra cita em seu trabalho algumas dessas soluções (Almendra, 2003):

- ♦ **Partição de ambientes**, o ambiente virtual é dividido em diversos subambientes ou áreas, para que as informações de atualização geradas dentro de uma partição sejam enviadas apenas para os participantes dessa partição e, possivelmente, para os das partições vizinhas;
- ♦ **Otimização do protocolo de comunicação**, as atualizações de estado de entidades podem ser transmitidas através de mensagens simplificadas que incluam apenas o subconjunto de propriedades que foram modificadas, sem a necessidade de mensagens completas contendo todas as propriedades da entidade;
- ♦ **Restrição ao uso de *streaming***, principalmente em aplicações através da Internet, o uso de *streaming* de áudio ou vídeo em conjunto com CVEs ainda é dispendioso, visto que esse recurso é grande consumidor de largura de banda e de carga de processamento (Moura-Filho & Oliveira, 1998); e,
- ♦ **Predição de movimento**, o uso de técnicas de *dead-reckoning*⁵ promove uma grande redução no tráfego de mensagens necessárias para a correta atualização do posicionamento dos objetos e personagens virtuais (Gutmann *et al.*, 1998).

⁵ A técnica de *dead-reckoning* é um artifício adotado em CVEs que tem como finalidade básica a redução do tráfego de mensagens referentes às atualizações de estado das entidades participantes do CVE.

3.3.7 Redução do Atraso de Rede

A redução do atraso de rede é um requisito muito importante que deve ser considerado no desenvolvimento de CVEs, para que o usuário seja capaz de perceber em tempo real as atividades realizadas por ele próprio e pelos outros participantes, a fim de reagir apropriadamente a essas novas informações. Os CVEs devem apresentar a cada usuário a ilusão de que o ambiente virtual está inteiramente localizado em sua máquina, e que suas ações têm impacto instantâneo.

O atraso de rede em um CVE geralmente é consequência de uma largura de banda incompatível com o nível de atividade no CVE, ou seja, sobrecarga na rede. Assim, o projetista deve analisar a capacidade do CVE projetado e adaptá-la à largura de banda disponível. Para isso, ele deve trabalhar com valores de latência máximos na faixa de 100 a 200 milisegundos (Makrakis *et al.*, 1998).

Na Internet, o atraso causado pelo tempo de viagem dos pacotes de informação pelos canais de comunicação dificulta uma simulação perfeita da interação entre usuários e objetos, principalmente no caso de um ou mais usuários realizando interação simultânea sobre um objeto comum (Shirmohammadi & Georganas, 2001).

Em ambientes muito dinâmicos, é provável que usuários percebam realidades distintas, visto que as atualizações de estado podem chegar em intervalos irregulares, causando um impacto direto no realismo do CVE (Singhal & Zyda, 1999).

3.3.8 Modelo de Comunicação

Segundo Gossweiler e seus co-autores (1994), os modelos de comunicação encontrados em redes de computadores utilizadas em CVEs podem ser basicamente divididos em dois: Modelo Centralizado e Modelo Distribuído.

No modelo centralizado (Figura 3.1), um computador central recebe todas as mensagens provenientes dos participantes conectados ao CVE. Quando uma mensagem chega ao computador central, ele processa essa mensagem armazenando os resultados em uma base de dados, e depois devolve o resultado das alterações feitas a cada um dos participantes. Esse modelo pode apresentar problemas de escalabilidade devido à sobrecarga de mensagens no computador central.

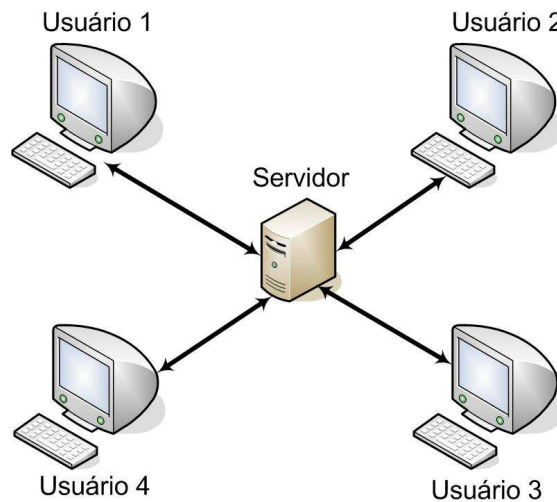


Figura 3.1. Modelo de Comunicação centralizado.

No modelo distribuído (Figura 3.2), cada usuário conectado possui uma cópia do CVE (ou parte dele), de modo que os processos de renderização e atualização dos objetos e cenário ficam a cargo de cada computador conectado ao sistema. Apesar de o modelo distribuído resolver o problema de escalabilidade do Modelo Centralizado, pois nenhum computador fica sobrecarregado quanto ao processamento de mudanças, a quantidade de mensagens e conexões na rede pode ser ainda muito grande.

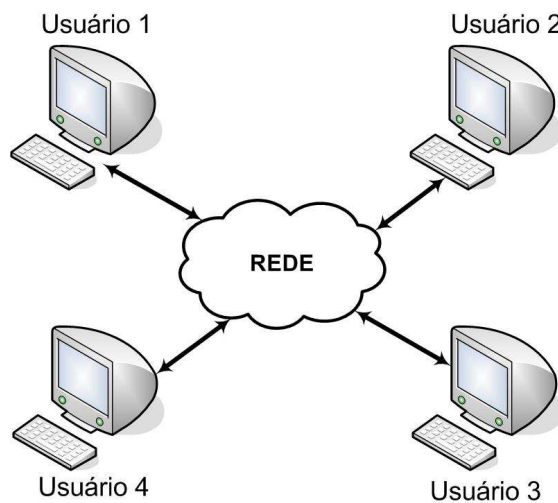


Figura 3.2. Modelo de Comunicação Distribuído.

Além do modelo de distribuição, outro aspecto da comunicação de CVEs refere-se à forma pela qual os usuários transmitem suas mensagens. Segundo Eduardo (2001) e Stytz (1996), há três formas básicas: *unicast*, *broadcast* e *multicast*.

Modelo com Conexões Unicast.

O modelo com conexões *unicast* (Figura 3.3), também conhecido como ponto a ponto (*point-to-point*), é utilizado para o envio de mensagens de uma origem para apenas um

destino. 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 CVE. A principal desvantagem desse modelo é que o número de conexões estabelecidas e mensagens geradas podem ser muito alto quando o número de usuários tende a crescer (Macedonia & Zyda, 1997).

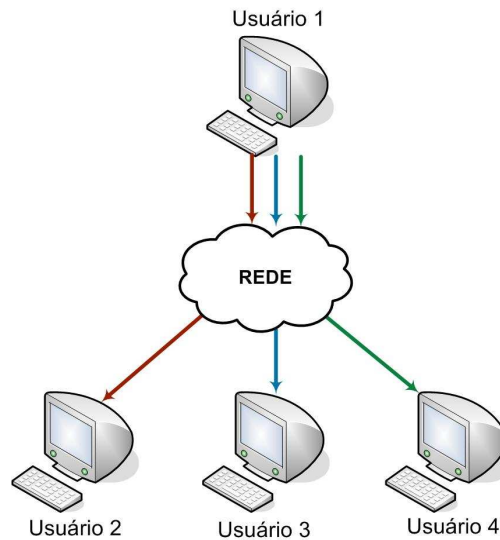


Figura 3.3. Modelo com conexões *unicast*.

Modelo com Conexões Broadcast (difusão).

Para diminuir o número de conexões geradas com a transmissão *unicast*, foi proposto o modelo com conexões *broadcast* (Figura 3.4). Neste modelo, ao invés de ser enviada uma mensagem para cada um dos usuários quando ocorre alguma alteração no CVE, somente uma mensagem é enviada, sendo que todos os outros usuários recebem essa mensagem.

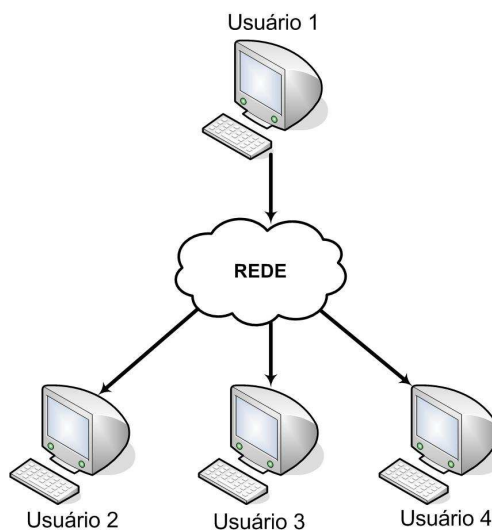


Figura 3.4. Modelo com conexões *broadcast*.

Modelo com Conexões Multicast.

O modelo de comunicação com conexões *multicast* (Figura 3.5) permite que sejam formados grupos de usuários, de modo que é criada uma conexão comum para cada grupo. Quando um usuário envia uma mensagem, ele a envia para um grupo específico do qual faz parte e somente os usuários que estiverem participando do mesmo grupo receberão essa mensagem.

A idéia de se utilizar grupos *multicast* torna-se ainda mais interessante para se dividir o ambiente virtual, ou criar as chamadas áreas de interesse (Zyda, 1995). No caso da divisão do CVE, pode-se atribuir um grupo *multicast* para cada uma das áreas divididas. Nesse caso, ao entrar em uma determinada área, um usuário passaria a fazer parte do grupo *multicast* daquela área, enviando e recebendo somente as atualizações do CVE referentes aquele grupo.

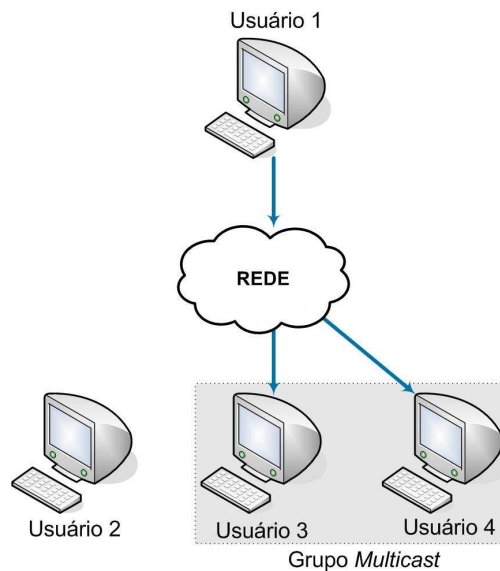


Figura 3.5. Modelo com conexões *multicast*.

3.3.9 Modelo de Armazenamento

Decidir onde as informações serão armazenadas afeta principalmente as escalas de dados do ambiente virtual, os requisitos de comunicação e a confiabilidade do sistema em geral (Macedonia & Zyda, 1997). Baseado em Rodello e seus co-autores (2001), basicamente o armazenamento dos dados pode se dar através de: Banco de Dados Centralizado, Bancos de Dados Replicado ou Bancos de Dados Distribuído.

Banco de Dados Centralizado

Nesse modelo, os dados são centralizados em uma única máquina, que atua como servidor, permitindo uma melhor consistência do CVE, uma vez que os clientes mantêm os dados pertinentes à cena em memória e apenas um usuário pode modificar o banco de dados em um determinado tempo. As desvantagens desse modelo começam com a limitação no número de participantes no ambiente. Quanto maior a quantidade de usuários, maior será a dificuldade em manter a consistência do banco de dados dos objetos em tempo real, tornando o servidor um ponto crítico do sistema.

Banco de Dados Replicado

Nesse modelo, cada elemento da rede conectado ao sistema possui uma réplica completa dos dados do ambiente virtual (modelos geométricos, texturas, sons, objetos etc.). A consistência de todos os ambientes é alcançada com a utilização de mecanismos de atualizações, como por exemplo, *heartbeat*, no qual cada elemento envia periodicamente mensagens de atualização informando seu estado. A vantagem desse modelo é que as mensagens são relativamente pequenas por serem somente mensagens de atualização. As desvantagens estão diretamente relacionadas com a inflexibilidade e o aumento das informações do CVE. Dessa forma, as informações do banco de dados estão replicadas por todos os elementos da rede, aumentando a possibilidade do CVE tornar-se inconsistente à medida que a quantidade de participantes aumenta.

Banco de Dados Distribuídos.

Nesse modelo, cada elemento da rede possui apenas uma parte do ambiente em sua base de dados. O número de mensagens é elevado, no entanto, a técnica de *multicasting* diminui o tráfego na rede. Esse modelo é apropriado para a construção de ambientes distribuídos de larga escala, entretanto é de difícil escalabilidade por causa do custo da comunicação associada com a confiabilidade e a consistência de dados através de rede de grande porte como a Internet.

3.3.10 Protocolo de Comunicação

A fim de se permitir a troca de informações entre diversos elementos de rede, faz-se necessário padronizar as regras de comunicação. Esses padrões são conhecidos como protocolos de comunicação, e são uma prerrogativa para o suporte e gerenciamento de qualquer sistema multiusuário. Existem vários protocolos de comunicação em uso

atualmente, entre os mais conhecidos estão (Rodello *et al.*, 2001; Schulzrinne *et al.*, 2003; Schulzrinne & Casner, 2003):

- ♦ **IP (*Internet Protocol*)** – esse protocolo tem o objetivo de fornecer a melhor forma de transportar datagramas da origem para o seu destino, independentemente das máquinas estarem na mesma rede ou em outras redes intermediárias (Tanenbaum, 1996). Em CVEs, mecanismos do protocolo IP podem ser utilizados para distribuição de dados por meio da comunicação *multicast*. Tal comunicação está relacionada com a classe de endereço D, que identifica um grupo de elementos de rede;
- ♦ **TCP (*Transmission Control Protocol*)** – esse protocolo é utilizado acima da camada de Rede IP e pertence à camada de transporte. O TCP é um protocolo confiável, pois garante a chegada da mensagem por checagem de erros. Entretanto, a confiabilidade do TCP provoca retardos na entrega dos pacotes. Para CVEs que têm a necessidade de entregar informações confiáveis, o protocolo TCP é o mais adequado; e,
- ♦ **UDP (*User Datagram Protocol*)** – esse protocolo não garante a chegada da mensagem. O protocolo UDP é suficiente para aquelas aplicações que não são rígidas na entrega de informações.
- ♦ **RTP (*Real - Time Protocol*)** – esse protocolo é utilizado para transmissão de conteúdo multimídia em aplicações de tempo real como, por exemplo, Voz sobre IP (VoIP). Como a arquitetura do RTP baseia-se no protocolo UDP, os pacotes RTP não têm garantia de entrega.

Existem também protocolos projetados exclusivamente para integrar as diversas aplicações empregadas na disponibilização do ambiente virtual compartilhado. Esses protocolos de integração merecem atenção especial, e suas definições devem levar em consideração todos os recursos disponíveis no CVE, incluindo a troca de dados entre os diversos participantes, a atualização dos estados de todo o conjunto de entidades existentes no interior do mundo virtual e mesmo a restrição de acesso de usuários a determinados objetos ou recintos virtuais. Assim como outros protocolos, o protocolo utilizado no CVE deve preconizar, sempre, o melhor aproveitamento da rede de comunicação utilizada, integrando de forma eficiente os recursos de todas as aplicações que dele fazem parte. Na evolução dos CVEs, dois modelos de protocolos básicos merecem destaque (Leite-Junior *et al.*, 2001):

- ♦ ***DIS-like protocols*** – o protocolo DIS (*Distributed Interactive Simulation*) (IEEE, 1995) consiste em um grupo de padrões desenvolvidos pelo Departamento de Defesa Americano e por entidades da indústria americana voltados para arquiteturas de comunicação, formato e conteúdo de dados, interação e informação sobre entidades virtuais, gerenciamento de simulações, medidas de desempenho, emissões de radiocomunicação, instrumentação de campo, segurança, formatos de bases de dados, fidelidade, controle de exercícios, até o feedback de dispositivos de interação (utilizados ou não em sistemas de realidade virtual). Protocolos de NVEs que seguem a especificação do protocolo DIS são conhecidos como DIS-like protocols (Singhall & Zyda, 1999); e,
- ♦ ***Game-like protocols*** – são protocolos, também conhecidos como SRMPs (Scalable Reliable Multicast Protocols), inspirados em jogos multiusuários de computadores. Nessa abordagem, busca-se somente a existência de cópias aproximadas do mundo virtual nos computadores de cada um dos participantes do CVE. O problema é tratado como uma sincronização de bancos de dados, garantindo o envio de mensagens sem a necessidade do uso de mensagens do tipo *keep-alive* (checagem constante referente ao funcionamento normal de um determinado elemento da rede). Apesar de não se preocupar com a precisão das diversas réplicas do ambiente virtual compartilhado, os *game-like protocols* são bastante eficazes. Por serem desenvolvidos especificamente para o uso em CVEs, os tipos de mensagens que eles englobam são bastante compactos, mas são capazes de transmitir as informações vitais ao correto funcionamento do CVE. Essa transmissão dá-se de uma forma rápida e eficiente, geralmente, com um tempo baixo de atualização global de todas as aplicações envolvidas. Isso acaba por garantir um sincronismo quase ótimo de todas as cópias do ambiente virtual compartilhado, espalhadas pelos vários computadores utilizados para seu acesso;

3.4 Arquiteturas de CVEs

Atualmente, é possível encontrar vários exemplos da utilização de CVEs em diversas áreas do conhecimento. A seguir são apresentados alguns exemplos de arquiteturas de CVEs que mais influenciaram no desenvolvimento deste trabalho.

3.4.1 DIVE

O sistema DIVE (*Distributed Interactive Virtual Environment*), desenvolvido pelo *Swedish Institute of Computer Science* (SICS – Suécia), resultou em um CVE disponível livremente para uso não comercial, utilizado mundialmente como base para outros projetos. O DIVE também provê um ambiente de desenvolvimento geral para CVEs (Frécon & Stenius, 1998; DIVE, 2004).

O DIVE vem sendo desenvolvido continuamente desde o começo da década de 1990. Originalmente ele foi desenvolvido para ser utilizado imersivamente, utilizando monitores acoplados à cabeça (*head-mounted display*) ou projetores, mas recentemente tornou-se acessível por PCs comuns (Hagsand, 1996). Ele permite que participantes distribuídos possam imergir e navegar livremente em um ambiente virtual tridimensional através de uma arquitetura que permite comunicação via áudio, vídeo, bate-papo, gestos simples e ferramentas de apoio à reunião, como por exemplo, quadro-branco (*whiteboard*) compartilhado (Figura 3.6). Os participantes também podem manipular objetos no mundo virtual (movendo e girando objetos ou adicionando novos objetos). O ‘universo’ do DIVE inclui vários objetos e mundos virtuais separados, unidos por portais, que podem ser baixados sob demanda da *Web*.



Figura 3.6. Participantes compartilhando um quadro branco no DIVE.

Uma das contribuições do sistema DIVE foi a utilização de bancos de dados dinâmicos, o que permitia a inclusão e a modificação de objetos de forma consistente. Isso o tornou mais adequado em situações em que a consistência do banco de dados deveria ser

garantida, em tempo real, para todos os participantes. No entanto, essa característica do sistema DIVE o tornava um maior consumidor de largura de banda, dado o volume extra de mensagens gerado pelos mecanismos de controle de concorrência – que garantiam a consistência do banco de dados (Almendra, 2003).

3.4.2 SIMNET

Em 1989, o Departamento de Defesa Norte-Americano lançou SIMNET (*simulator network*) que permitiu ao pessoal militar praticar operações de combate em sistemas de treinamento interativo em tempo real. O SIMNET (2005) é um simulador de batalhas militares onde os integrantes do ambiente virtual assumem o controle de veículos militares como tanques, aviões e soldados para interagirem no ambiente virtual. Utilizando-se de redes de longa distância e *links* de satélite, o SIMNET permite a conexão e interação de centenas de usuários (Shaw & Green, 1993; Locke, 1994).

3.4.3 NPSNET

NPSNET foi desenvolvido pela Naval Postgraduate School (*Computer Science Department of the U.S. Naval Postgraduate School – USA*) para o treinamento militar em larga escala com simulação de exercícios de guerra (Macedonia *et al.*, 1995; NPSNET, 2004).

Um dos focos desse sistema é estender o protocolo de comunicação do SIMNET que tem como objetivo interconectar vários usuários, permitindo a simulação de batalhas militares. 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.

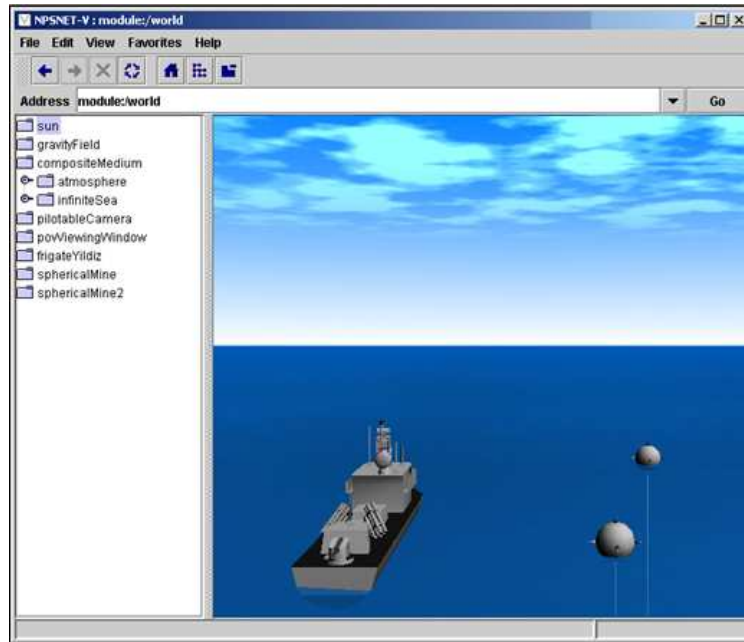


Figura 3.7. Navegador do NPSNET-V.

A principal contribuição do projeto NPSNET foi o desenvolvimento de arquiteturas de rede escaláveis e técnicas de gerenciamento de interesse (Macedonia & Zyda, 1997). O NPSNET encontra-se na quinta versão (vide Figura 3.7) a qual pode ser encontrada no site do projeto (<http://www.movesinstitute.org/~npsnet/v>).

3.4.4 MASSIVE



Figura 3.8. Inhabited TV Combina CVE com TV.

O CVE MASSIVE (*Model, Architecture and System for Spatial Interaction in Virtual Environments*) foi desenvolvido pela Universidade de Nottingham (UK) e vem sendo desenvolvido para suportar teleconferência em grandes redes de computadores (Greenhalgh & Benford, 1995). O MASSIVE foi utilizado para criar uma variedade de

aplicações nas mais diversas áreas. Entre elas, destaca-se o inhabited TV (Greenhalgh *et al.*, 1999), uma combinação de CVE com televisão *broadcast* em que o público participa de espetáculos de TV dentro de um ambiente virtual (Figura 3.8). Atualmente existem três versões do MASSIVE: MASSIVE-1, CVE (também conhecido como MASSIVE-2) e a última versão, o MASSIVE-3.

3.4.5 VELVET

Atualmente, as arquiteturas disponíveis dão suporte a vários usuários em um mesmo ambiente virtual, mas elas não funcionam bem se uma grande quantidade de usuários estiver interagindo em um pequeno "espaço" no mundo virtual. A arquitetura VELVET (V^Ery Large Virtual EnvironmenTs) é uma arquitetura adaptativa híbrida desenvolvida para dar suporte a um grande número de usuários interagirem em um mesmo CVE. Essa arquitetura também suporta pequenos grupos de usuários, mas seu uso em ambientes virtuais de grande porte representa seu maior potencial. Ela utiliza um modelo de filtragem adaptativa baseado em *multicast* para gerenciar áreas de interesse (Oliveira & Georganas, 2003).

3.4.6 Blaxxun Community Platform



Figura 3.9. Interface do ambiente Blaxxun.

Blaxxun (Blaxxun, 2004) foi uma das primeiras empresas a desenvolver um ambiente virtual multiusuário utilizando VRML (*Virtual Reality Modeling Language*), o *Cybergate*.

Como VRML é uma arquitetura aberta, isto significa que é possível construir um mundo ou avatar próprio e incorporá-los ao sistema. Blaxxun Community Platform é um sistema modular cliente/servidor que fornece um amplo conjunto de funções para desenvolvimento de comunidades virtuais para *Web* (Figura 3.9). Além dos serviços que gerenciam usuários, 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.

3.4.7 OzGate!

OzGate International (Ozgate, 2004) é uma empresa que oferece um sistema colaborativo comercial do tipo cliente/servidor que permite grupos de pessoas comunicarem-se por voz pela Internet. Os participantes usam avatares modelados apenas como uma cabeça sem corpo (Wilcox, 1998). Uma vez dentro de um mundo virtual do OzGate, um participante comunica-se com qualquer um no ambiente por um microfone conectado ao seu PC. OzGate Traveller, o navegador cliente (Figura 3.10), utiliza som tridimensional permitindo os participantes que estão mais próximos parecerem falar mais alto do que os avatares que estão distantes. Mecanismos de comunicação não-verbal também são utilizados, tais como piscar olhos, sincronismo labial, expressões faciais (feliz, triste, surpreso, raiva).

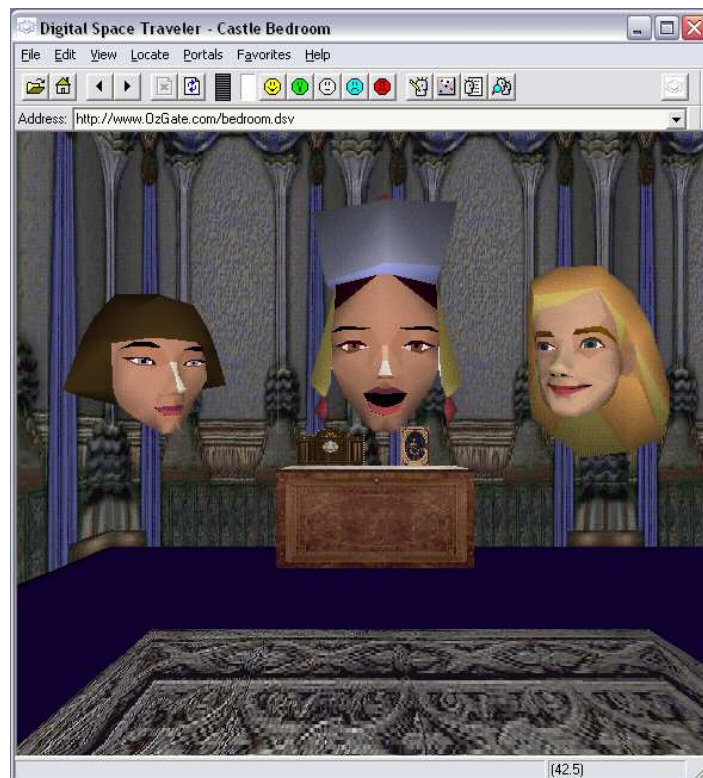


Figura 3.10. Navegador do OzGate Traveller.

3.5 Aplicações de CVEs

Sistemas experimentais do tipo CVE já são usados há décadas, mas apenas recentemente começaram a sair das esferas acadêmicas e militares. Esse aumento de popularidade se deve principalmente ao rápido aumento da capacidade de processamento das máquinas e de seus custos reduzidos (Fuks *et al.*, 2002).

Os CVEs oferecem um sistema natural de compartilhamento baseado em modelos de interação da própria natureza humana, e, por isso, podem ser aplicados para os mais diversos fins, em especial para os de caráter colaborativo. Algumas das áreas de aplicação são, por exemplo (Çapin *et al.*, 1999):

- ♦ Teleconferência virtual;
- ♦ Trabalho cooperativo envolvendo modelos tridimensionais;
- ♦ Jogos multiusuários;
- ♦ Comércio eletrônico;
- ♦ Telemedicina (diagnósticos a distância, treinamento cirúrgico virtual);
- ♦ Treinamento e educação a distância; e
- ♦ Turismo virtual.

A seguir, são apresentadas algumas aplicações de ambientes virtuais colaborativos.

3.5.1 NICE

O projeto NICE (*Narrative-based, Immersive, Constructivist/Collaborative Environment*), desenvolvido pelo *Interactive Computing Environments Laboratory* e pelo *Electronic Visualization Laboratory* da Universidade de Illinois (Chicago – USA), é um ambiente distribuído e imersivo, implementado utilizando tecnologia CAVE⁶ (Johnson *et al.*, 1998). O objetivo do NICE é fornecer um ambiente para que crianças construam e cuidem de ecossistemas virtuais simples, de forma colaborativa, via rede. As crianças, remotamente distribuídas, criam histórias a partir de suas interações nos mundos real e virtual (vide Figura 3.11).

A arquitetura do NICE é baseada no modelo cliente-servidor e utiliza um protocolo desenvolvido para dar suporte às características de dados de realidade virtual e possibilitar aos usuários entrar e abandonar o ambiente facilmente a partir de qualquer lugar na

⁶ CAVE é um dispositivo de exibição de alta tecnologia que consiste de telas de projeção semitransparentes, projetores de vídeo estéreo, espelhos, e um sistema de som ambiente (Cruz-Neira, 1992).

Internet. O servidor NICE mantém a base de dados e assegura a consistência do sistema inteiro (Kawamoto *et al.*, 2001).



Figura 3.11. Uma Criança controlando um avatar dentro do NICE.

3.5.2 HistoryCity

O projeto HistoryCity é um ambiente virtual em rede que representava um período histórico da cidade de Singapura no ano de 1870 (Figura 3.12). Dentro desse ambiente virtual, voltado para crianças de 7 a 11 anos, os usuários eram estimulados a conhecer a história e a cultura antiga da cidade. O ambiente era composto de vinte e quatro comunidades completas, com edificações e objetos históricos.



Figura 3.12. Ambiente do HistoryCity.

HistoryCity foi desenvolvido utilizando o software NetEffect (Das *et al.*, 1997a; 1997b) o qual segue uma arquitetura cliente-servidor que particiona um mundo virtual em comunidades, distribui essas comunidades entre um conjunto de servidores e migra os clientes de um servidor para outro na medida em que os clientes se movimentam através das comunidades. Tal arquitetura visa minimizar o tráfego de rede, particularmente o

tráfego entre diferentes servidores. A arquitetura do HistoryCity consiste de um servidor mestre e vários outros servidores. Cada servidor mantém um grafo completo da topologia, permitindo, assim, se comunicarem sem atravessar um servidor intermediário. Um cliente está conectado a apenas um servidor em qualquer instante, mas pode migrar de um servidor para outro dinamicamente (Das *et al.*, 1997b).

3.5.3 AVC–MV

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).

O projeto Museu Virtual (Wazlawick *et al.*, 1999) é uma proposta de ferramenta de autoria para construção colaborativa de museus em realidade virtual. Baseando-se na teoria construtivista, o projeto pretende oferecer a alunos e professores a oportunidade de serem os curadores de seus próprios museus virtuais, o que os permite atuar, de forma cooperativa, na construção e manutenção de espaços e objetos dentro do museu (Figura 3.13). Um dos destaques do projeto é a possibilidade de construção de objetos interativos programados pelos próprios usuários.

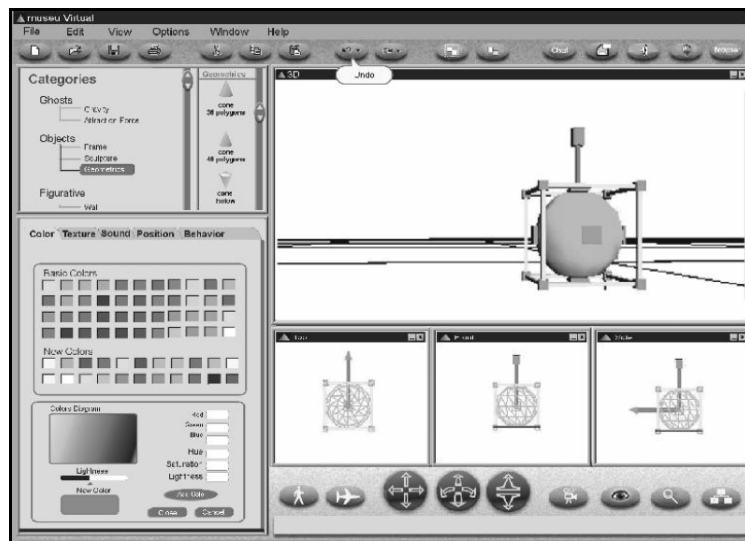


Figura 3.13. Ferramenta de autoria do Museu Virtual.

AVC-MV foi inspirado, em vários aspectos, nas idéias do projeto NICE, principalmente no que diz respeito à criação de um ambiente de ensino construtivista. Assim como os projetos HistoryCity e NICE, o AVC-MV baseia-se numa arquitetura cliente-servidor, adaptada aos objetivos e recursos técnicos disponíveis ao projeto.

Particularmente, as abordagens para organizar os conjuntos de dados distribuídos e para traçar o comportamento dos usuários, utilizadas, respectivamente, pelo AGORA e pelo NICE, foram muito úteis na especificação do AVC-MV (Kawamoto *et al.*, 2001).

3.5.4 AVAL

O Projeto AVAL – Ambientes Virtuais para o Aprendizado de Línguas – (Borges, 2001; Vidal *et al.*, 2001a), 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 CVEs representando 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. Na Figura 3.14, é mostrado o ambiente de um centro de artesanato modelado para um dos módulos do treinamento.



Figura 3.14. CVE de um centro de turismo virtual.

No Desenvolvimento dos CVEs do AVAL, foi utilizada a arquitetura ATAXIA (Leite-Junior, 2000; Leite-Junior *et al.*, 2001) desenvolvida pelo grupo CRAB (www.crab.ufc.br). A ATAXIA foi projetada para funcionar através da Internet, utilizando-se de microcomputadores comuns. No próximo capítulo, essa arquitetura é apresentada com mais detalhes, pois a arquitetura proposta nesse trabalho é uma extensão da arquitetura ATAXIA.

3.5.5 Gorilla World

O Gorilla World foi desenvolvido por uma equipe do *Georgia Institute of Technology* (Alison *et al.*, 1997). Trata-se de um ambiente virtual colaborativo para dar suporte ao aprendizado do comportamento de grupos de gorilas. O Gorilla World baseia-se no comportamento real dos gorilas que vivem no zoológico de Atlanta, e tem sido utilizado por crianças em algumas escolas americanas (Figura 3.15).



Figura 3.15. Habitante do Gorilla World.

3.5.6 Active Worlds

O Active Worlds, desenvolvido por Circle of Fire Studios Inc., é um CVE que utiliza o modelo cliente/servidor (ActiveWorlds, 2004). O navegador do Active Worlds (aplicação cliente), o qual é gratuito, permite que os usuários se comuniquem com outros membros da comunidade, construam sua própria residência e realizem diversas outras atividades relacionadas ao convívio em comunidade. Além disso, o servidor do Active Worlds permite que o usuário possua uma propriedade (grátis por um período de trinta dias) dentro do Universo Active Worlds que é composto por centenas de mundos virtuais. Também existem servidores (não gratuitos) que podem ser usados para construir CVEs independentes do Universo Active Worlds. O software provê uma seleção de avatares de vários sexos e etnias para o usuário escolher. Os avatares comunicam-se através de texto que é exibido no histórico do *chat* e em cima do avatar. O CVE utiliza mecanismos de comunicação não-verbal através de ações pré-definidas, como por exemplo, ‘feliz’, ‘bravo’, ‘acenar’, ‘saltar’, ‘brigar’, ‘dançar’ (vide Figura 3.16).



Figura 3.16. Interface do sistema ActiveWorld.

O sistema Active Worlds possui também uma versão educacional, que permite hospedar sem custos os mundos virtuais com propósitos educacionais, sendo necessário apenas que os curadores construam as edificações e os objetos a serem colocados dentro do mundo virtual. O seu uso permite, assim, avaliar e projetar de forma mais rápida as funcionalidades necessárias para um ambiente virtual voltado para a educação (Almendra, 2003).

3.5.7 Ragnarök Online

Ragnarök Online (Ragnarök, 2004) é um Massively Multiplayer Online Role-Playing Game (MMORPG) desenvolvido pela Level Up Games. Esse CVE é um jogo *online* em que centenas ou milhares de participantes distribuídos pela Internet conectam-se ao mesmo tempo em um ambiente virtual onde todos podem interagir entre si (Figura 3.17). Como se trata de um jogo de RPG, o personagem do participante evolui mudando aparência e características como, por exemplo, roupas, armas, nível, poder ofensivo e defensivo.

Em Rune-Midgard (nome do universo de Ragnarök Online) os usuários podem jogar contra outros participantes ou formar clãs unindo forças para desenvolver tarefas colaborativas no ambiente. Entre essas tarefas estão: derrotar monstros ou outros clãs de

usuários, fazer armas, procurar objetos raros ou simplesmente conversar com outros participantes. Ragnarök Online utiliza o modelo centralizado e possui um banco de dados persistente nos seus servidores, permitindo que a evolução dos participantes dentro do jogo seja mantida. (Ragnarök, 2004).



Figura 3.17. Um CVE do sistema Ragnarök Online.

3.6 Considerações Finais

Apesar da crescente popularidade, CVEs ainda apresentam uma série de desafios em seu desenvolvimento tais como os relacionados aos requisitos listados neste capítulo, quais sejam: gerenciar eficientemente recursos de rede (adequação a largura de banda, perda de dados, escalabilidade, etc.), manter aplicações gráficas em tempo real (fidelidade, por exemplo, alocação de CPU para geração das imagens) e desenvolver aplicações robustas multiusuários (por exemplo, manutenção de consistência entre os usuários).

Também há as dificuldades específicas da área de aplicação do CVE, tais como a integração com grandes bases de dados (por exemplo, para as informações geográficas de simulações militares) e a autenticação de usuários (para aplicações de comércio eletrônico, por exemplo). Quando o campo de aplicação é especificamente a realização de trabalhos colaborativos, somam-se os desafios relacionados à área de CSCW. Para mencionar alguns, há a dificuldade de se trabalhar com os objetos do mundo virtual (Benford *et al.*, 1994) e a necessidade de se criar avatares realistas para ampliar a capacidade de comunicação entre os participantes e seu sentimento de presença (Joslin *et al.*, 2001).

A análise dos CVEs e de suas diferentes áreas de aplicação apresentada neste trabalho serviu para identificar estratégias de desenvolvimentos para atender aos requisitos

mencionados. Entre as áreas de aplicação analisadas, a que mais se destaca é a área de jogos. Sem dúvida, a indústria de jogos tem desenvolvido soluções eficazes para a renderização e a coordenação de cenários virtuais e para o suporte à interação multiusuário.

Os conceitos e características de CVEs apresentados nesse capítulo são essenciais no desenvolvimento do sistema proposto neste trabalho, já que se trata de uma arquitetura para viabilização de CVEs através da Internet denominada de CRAbCVE. A identificação dos requisitos de hardware e software foi muito importante na definição da arquitetura CRAbCVE que será apresentada no próximo capítulo.

Capítulo 4

A Arquitetura CRAbCVE⁷

4.1 Introdução

A arquitetura CRAbCVE, apresentada neste capítulo, 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.

Para o desenvolvimento da arquitetura CRAbCVE, as funcionalidades dos elementos da Ataxia foram redefinidas, um novo componente foi adicionado e um novo protocolo de comunicação foi especificado. Essas modificações visam: prover suporte aos processos de colaboração definidos no Capítulo 2 – uma das contribuições mais significativas deste trabalho; permitir a publicação de mundos virtuais; garantir uma maior acessibilidade e tornar a CRAbCVE uma arquitetura de uso genérico.

A publicação de mundos virtuais é uma funcionalidade muito importante. Ela 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 e desperta um maior interesse nos usuários do sistema. Para que a publicação de novos mundos virtuais seja possível, técnicas de instanciação automática de componentes de CVEs são adotadas. Essas técnicas são baseadas no trabalho de Almendra (2003) que utiliza a arquitetura Ataxia para viabilizar NVEs integrados com SGAs (Sistemas de Gerenciamento de Aprendizagem). Em seu trabalho, Almendra propõe o uso de fábricas para a instanciação e desinstanciação de componentes da Ataxia para viabilizar NVEs de forma automática.

A maior acessibilidade é garantida pela CRAbCVE através de um novo componente que permite ao usuário ter conhecimento de todos os CVEs disponíveis no sistema. Esse

⁷ A parte CRAb do nome da arquitetura refere-se ao grupo de pesquisa do qual o autor participa (<http://www.crab.ufc.br>).

novo componente funciona como um portal de acesso aos CVEs, além de ter o cadastro de todos os usuários do sistema. Essa funcionalidade distingue a arquitetura CRAbCVE da Ataxia para a qual um ambiente virtual é um sistema isolado, ao qual o usuário deve conectar-se diretamente.

Ao contrário da Ataxia, que foi originalmente projetada para a área de educação, a arquitetura CRAbCVE é uma arquitetura genérica que pode ser usada para diversas aplicações, inclusive entretenimento.

O restante deste capítulo está dividido da seguinte forma: na Seção 4.2, a arquitetura Ataxia é apresentada; na Seção 4.3, são discutidas as características da arquitetura CRAbCVE; na Seção 4.4, os componentes da arquitetura são apresentados e é discutido como esses componentes são empregados na disponibilização de espaços virtuais compartilhados; na Seção 4.5, é apresentado o funcionamento da arquitetura, sendo destacados vários cenários de utilização; na Seção 4.6, o protocolo de comunicação da arquitetura é definido (baseado nos cenários da seção anterior); e por último, na Seção 4.7, tecem-se algumas considerações finais.

4.2 A Arquitetura Ataxia

A função básica da arquitetura Ataxia é disponibilizar espaços virtuais compartilhados. A arquitetura foi criada para suprir as necessidades de um ambiente virtual em rede (NVE), de baixo custo, voltado para a educação a distância, segundo um modelo de escola virtual (Leite-Junior, 2000; Leite-Junior *et al.*, 2001).

A motivação para o desenvolvimento da arquitetura foi a possibilidade de sua aplicação no projeto Infovias do Desenvolvimento (Pequeno, 1998), a fim de que as novas propostas de uso de realidade virtual na educação possam ser oferecidas através da Internet pelos programas de educação a distância voltados para o interior do Estado do Ceará (Vidal *et al.*, 2001b).

A idéia principal desse modelo de escola virtual é que os indivíduos podem acessar simultaneamente o AVR, mesmo estando geograficamente distribuídos, e, então, reunirem-se em locais específicos dentro do ambiente, tais como fóruns, laboratórios, salas de aulas ou ambientes exploratórios, a fim de realizar atividades em grupo.

Para atender aos requisitos do modelo de escola virtual a baixo custo, a arquitetura foi projetada para funcionar através da Internet e utilizando-se de microcomputadores comuns. Todavia, a concepção de ambientes virtuais de baixo custo requer a adoção de uma estratégia que divida as funções de gerenciamento do AVR entre diversos

componentes, que, então, podem ser alocados em máquinas separadas. A arquitetura Ataxia 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 ambiente.

Outra característica importante da arquitetura Ataxia é a possibilidade de incorporação de novos serviços e aplicações legadas ao ambiente virtual. Esse caráter aberto da arquitetura possibilita que alguns recursos sejam adicionados ou melhorados à medida que a infra-estrutura disponível evolui. Por exemplo, com a instalação de uma nova e mais rápida rede de dados, é possível incorporar comunicação via voz em um ambiente virtual onde previamente a comunicação restringia-se à troca de mensagens de texto. A arquitetura Ataxia, ilustrada na , é composta de sete componentes.

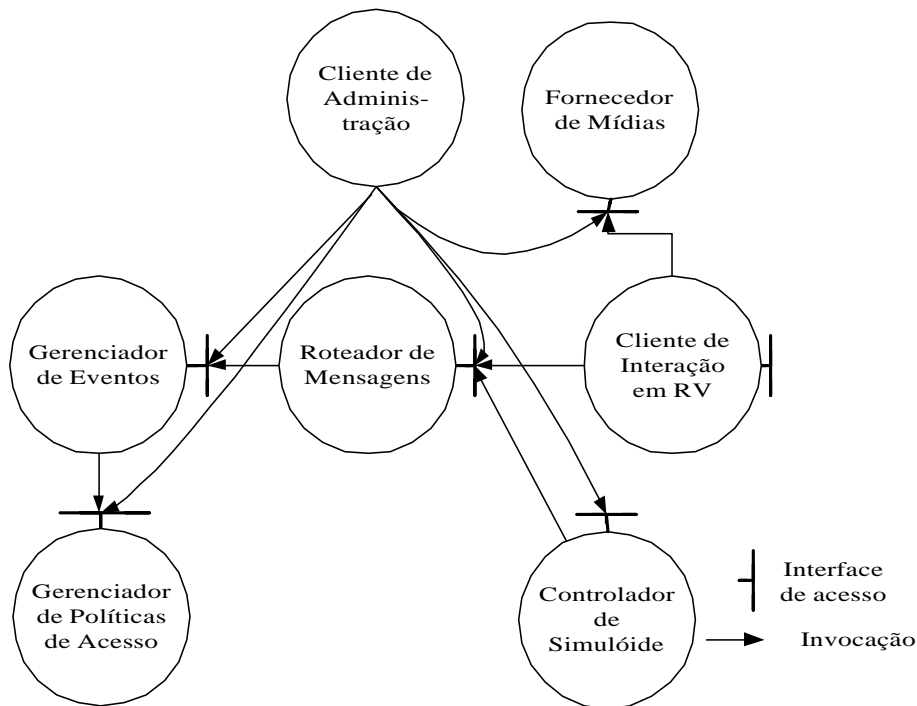


Figura 4.1. Componentes da arquitetura Ataxia.

Como já foi mencionado, a CRAbCVE é uma evolução da arquitetura Ataxia. Essa evolução surgiu da modificação dos componentes existentes e da adição de um novo para permitir o trabalho colaborativo. Por esse motivo, no contexto deste trabalho, os componentes da Ataxia não são discutidos diretamente (para maiores informações, ver (Leite-Junior, 2000)). Na Seção 4.4, são discutidas as modificações realizadas nos componentes da Ataxia e o novo componente é apresentado.

4.3 Características da Arquitetura CRAbCVE

A arquitetura CRAbCVE é uma arquitetura distribuída e híbrida; com suporte a implementações multiplataforma; expansível; escalar; com forte dependência de administração; que minimiza a utilização da rede de comunicação e que oferece persistência de estado e autoria de CVEs. Todas essas características, exceto a persistência de estado e autoria de CVEs, foram herdadas da Ataxia.

4.3.1 Arquitetura Distribuída e Híbrida

A concepção de ambientes virtuais de baixo custo requer a adoção de uma estratégia que divida as funções de gerenciamento do CVE entre diversos componentes, que possam ser alocados em máquinas separadas. A relação custo/benefício dessa abordagem, que se baseia no processamento distribuído em sistemas de baixo custo, é superior àquela obtida com o uso de supercomputadores para processamento centralizado (Tanenbaum, 1995). Então, adotando essa abordagem, a arquitetura CRAbCVE define um conjunto de componentes que podem ser distribuídos e executados em diversos microcomputadores comuns, de maneira simultânea, sendo cada um responsável por prover serviços específicos para o ambiente.

O modelo de comunicação utilizado na arquitetura CRAbCVE é uma combinação do modelo centralizado e do modelo de distribuído, o que caracteriza essa arquitetura como híbrida.

4.3.2 Suporte a Implementações Multiplataforma

O uso de um modelo distribuído também possibilita a utilização de plataformas diferentes para implantação do CVE. Isso permite que os desenvolvedores explorem as melhores características dos diversos padrões existentes e dos sistemas computacionais disponíveis. Assim, soluções de software e hardware de diferentes fabricantes podem ser integradas, viabilizando a construção do CVE como um todo e permitindo o melhor aproveitamento dos recursos computacionais disponíveis. De uma forma ideal, as melhores características de diversos padrões podem ser unidas, garantindo-se a qualidade do sistema final gerado.

4.3.3 Expansível

A arquitetura CRAbCVE é uma arquitetura expansível, pois permite a incorporação de novos serviços e aplicações legadas ao ambiente virtual. Para tanto, a arquitetura pode utilizar-se de três diferentes artifícios:

- ♦ **Uso de Simulóides⁸**, para realizar a interligação de aplicativos externos ao CVE (Vidal *et al.*, 2000), tornando-os disponíveis a todos os participantes do ambiente virtual;
- ♦ **Re-configuração de componentes**, a fim de permitirem a distribuição de informações nos dois sentidos, entre eles e os novos aplicativos; e,
- ♦ **Integração de serviços por PDU⁹**, que faz uso de uma PDU genérica para integrar as funcionalidades de serviços externos ao CVE – o protocolo empregado na arquitetura prevê a possibilidade de uso de aplicativos não pertencentes ao CVE.

Esse caráter aberto da arquitetura possibilita que alguns recursos sejam adicionados ou melhorados à medida que a infra-estrutura disponível evolui. Por exemplo, com a instalação de uma nova e mais rápida rede de dados, é possível incorporar comunicação via voz em um ambiente virtual onde previamente a comunicação restringia-se à troca de mensagens de texto.

4.3.4 Arquitetura Escalar

Os CVEs gerados a partir da arquitetura CRAbCVE são escaláveis, podendo conter um grande número de subambientes, que podem ser facilmente interligados através da técnica de portais (Çapin *et al.*, 1999) prevista na arquitetura. Todos esses subambientes são gerenciados diretamente por instâncias dos diferentes componentes da arquitetura, que podem ser replicados a fim de garantir um correto balanceamento de acessos e tráfego total.

4.3.5 Forte Dependência de Administração

O correto funcionamento dos CVEs gerados a partir da arquitetura CRAbCVE tem uma relação direta com a administração dos vários componentes por ela definidos. Por isso, estruturas específicas de documentação estão presentes na grande maioria desses componentes, possibilitando tanto a verificação de informações importantes (problemas de conexão, tipos de requisições de serviços, etc.) quanto à tomada de decisão posterior. Assim, a administração do CVE assume papel fundamental no provimento das funcionalidades presentes no ambiente.

⁸ Simulóides são personagens controlados por computador (Csordas, 2000).

⁹ PDU (Protocol Data Unit) é uma unidade básica de transmissão de dados de um protocolo de comunicação.

A fim de simplificar o gerenciamento dos CVEs, processos automatizados de monitoramento e controle, que não são tratados neste trabalho, podem ser estabelecidos e implementados nos sistemas que se utilizem da arquitetura CRAbCVE.

4.3.6 Minimização da Utilização da Rede de Comunicação

Para possibilitar seu uso em redes de baixa de velocidade, a arquitetura CRAbCVE apresenta um conjunto de soluções voltadas para a minimização da utilização da rede de comunicação e para um melhor aproveitamento da estrutura distribuída de processamento. As três principais soluções apresentadas para esse fim pela arquitetura são apresentadas a seguir.

Emprego de Dead-Reckoning

A técnica de *dead-reckoning* é um artifício adotado em CVEs que tem como função básica a redução do tráfego de mensagens referentes às atualizações de estado das diversas entidades do CVE na rede de comunicação. Apesar de arquiteturas de CVEs e NVEs geralmente não explicitarem a necessidade de definição de uma técnica de *dead-reckoning*, ficando tal tarefa deixada aos responsáveis pela implementação do ambiente virtual, a arquitetura CRAbCVE recomenda a adoção de um modelo específico, onde as fases de predição e convergência são obrigatórias e não baseadas em controle de tempo (Gutmann *et al.*, 1998). Entretanto, o emprego de *dead-reckoning* é desvantajoso para CVEs que possuam uma grande quantidade de participantes, pois há uma grande carga de processamento das fases de predição e convergência que são executadas na máquina local dos usuários. Assim, a CRAbCVE também recomenda um estabelecimento de um limite da quantidade de usuários em um CVE para utilização da técnica de *dead-reckoning*. Quando esse limite for superado, deverão ser enviadas todas as mensagens de posicionamento.

Minimização de Recursos de Áudio e Vídeo

Uma outra característica dessa arquitetura é o baixo uso de recursos de áudio e vídeo, principalmente por demandarem grande largura de banda e ainda pela significativa carga de processamento de suas estruturas de controle (Moura-Filho & Oliveira, 1998). É exatamente por isso que a arquitetura proposta não faz uso intensivo desses recursos, suportando-os somente como aplicações externas (discutida posteriormente), que podem ser agregadas devido ao caráter expansível da arquitetura.

Minimização de Consultas Referentes a Restrições de Acesso

Os CVEs gerados a partir da arquitetura somente permitem o acesso de usuários previamente cadastrados. Além disso, esses usuários podem ter seus direitos de acesso restringidos. As constantes consultas, necessárias à aplicação desse tipo de restrição, podem consumir bastantes recursos da rede. A fim de evitar a verificação dos direitos de acesso de usuários todas as vezes que os mesmos tentarem fazer uso de algum recurso do CVE, a arquitetura CRAbCVE adota o uso de *caches* locais de políticas de acesso. Assim, ao conectar-se ao sistema, a aplicação do usuário automaticamente armazena na *cache* local de políticas de acesso as possíveis restrições de utilização dos recursos disponíveis no CVE. O uso da *cache* de direitos de acesso evita que sempre que o usuário tente utilizar algum recurso do CVE, sua respectiva aplicação proceda a verificação de seus direitos de acesso junto à rede. A modificação dos direitos de acesso é realizada através do envio de mensagens do protocolo de comunicação para atualizar a *cache* de todos os usuários.

4.3.7 Persistência de Estado e Autoria de CVEs

A interação dos participantes com os objetos disponíveis nos CVEs suportados pela arquitetura CRAbCVE resulta na mudança de estados desses objetos. A mudança de estado de um objeto pode ocorrer, por exemplo, com a mudança de posição dentro do ambiente (movimentação) ou com a ativação de um comportamento pré-definido do objeto (por exemplo, o objeto porta possui um comportamento ‘abrir porta’ e, assim, o estado da porta pode ser ‘aberta’ ou ‘fechada’). Com o objetivo de manter o estado de todos os objetos dos CVEs, permitindo assim uma maior sensação de realismo e identidade com o ambiente, a arquitetura CRAbCVE adota soluções para que os estados dos objetos e do próprio ambiente sejam armazenados de forma persistente. Assim, se por alguma falha ou por opção dos administradores, o CVE ficar indisponível, é possível recuperá-lo exatamente como era antes.

Além da interação entre os participantes e da interação dos participantes com os objetos, a arquitetura CRAbCVE permite que os participantes interajam com o próprio ambiente. A interação dos participantes com o ambiente, inserindo, removendo e manipulando objetos dentro dos CVEs, juntamente com a persistência de estado de objetos possibilita a autoria de CVEs (esse assunto é tratado no próximo capítulo). Assim, torna-se possível a publicação de novos CVEs a partir de modelos pré-elaborados disponíveis pela arquitetura.

4.4 Componentes da Arquitetura CRAbCVE

A arquitetura CRAbCVE é composta por oito componentes, clientes ou servidores, que desempenham funções especializadas e trabalham de forma integrada para tornar disponíveis ambientes virtuais, através da troca de PDUs específicas. É importante ressaltar que um componente servidor não necessariamente corresponde a um servidor na rede, podendo um servidor possuir várias componentes servidores instanciados. Os componentes da arquitetura CRAbCVE estão ilustrados na Figura 4.1.

4.4.1 Gerenciador de Políticas do Sistema

O Gerenciador de Políticas do Sistema (GPS) é um componente novo em relação à arquitetura Ataxia. Para ter acesso aos CVEs, todos os participantes precisam passar antes pelo GPS, que pode ser considerado um componente centralizador. Existe uma única instância do GPS no sistema, ao contrário dos outros componentes que podem ter várias instâncias. O GPS apresenta três importantes funcionalidades: registro de usuários e elementos de RV, controle automático de instâncias de CVEs e localização de CVEs.

Registro de Usuários e Elementos de RV

O GPS é um servidor com uma base de dados que possui informações de usuários e elementos de RV (CVEs e objetos) do sistema. Para um usuário ter acesso a um CVE é preciso ter um cadastro junto ao GPS, e nesse cadastro é definido seu papel no sistema. Por exemplo, um usuário pode exercer o papel de um administrador, de um usuário avançado que pode realizar autoria e publicar CVEs, ou de um usuário comum que pode apenas navegar nos CVEs disponíveis. A definição e os tipos de papéis depende da aplicação em que o modelo da arquitetura foi utilizado. No próximo capítulo é definido um modelo de autoria colaborativa de CVEs que utiliza a CRAbCVE e possui três tipos de papéis: publicador, navegante e administrador.

Os registros de todos os CVEs do sistema e dos objetos disponíveis descrevem também as permissões de utilização por parte dos participantes. Um participante entra em um CVE do sistema conectando-se ao GPS e validando sua entrada no sistema. De acordo com seu cadastro no sistema, o GPS, então, disponibiliza somente o conjunto de CVEs aos quais o participante pode ter acesso. O GPS também possui o cadastro e as permissões de todos os elementos armazenados no Fornecedor de Mídias – um servidor que funciona como um repositório de mídias.

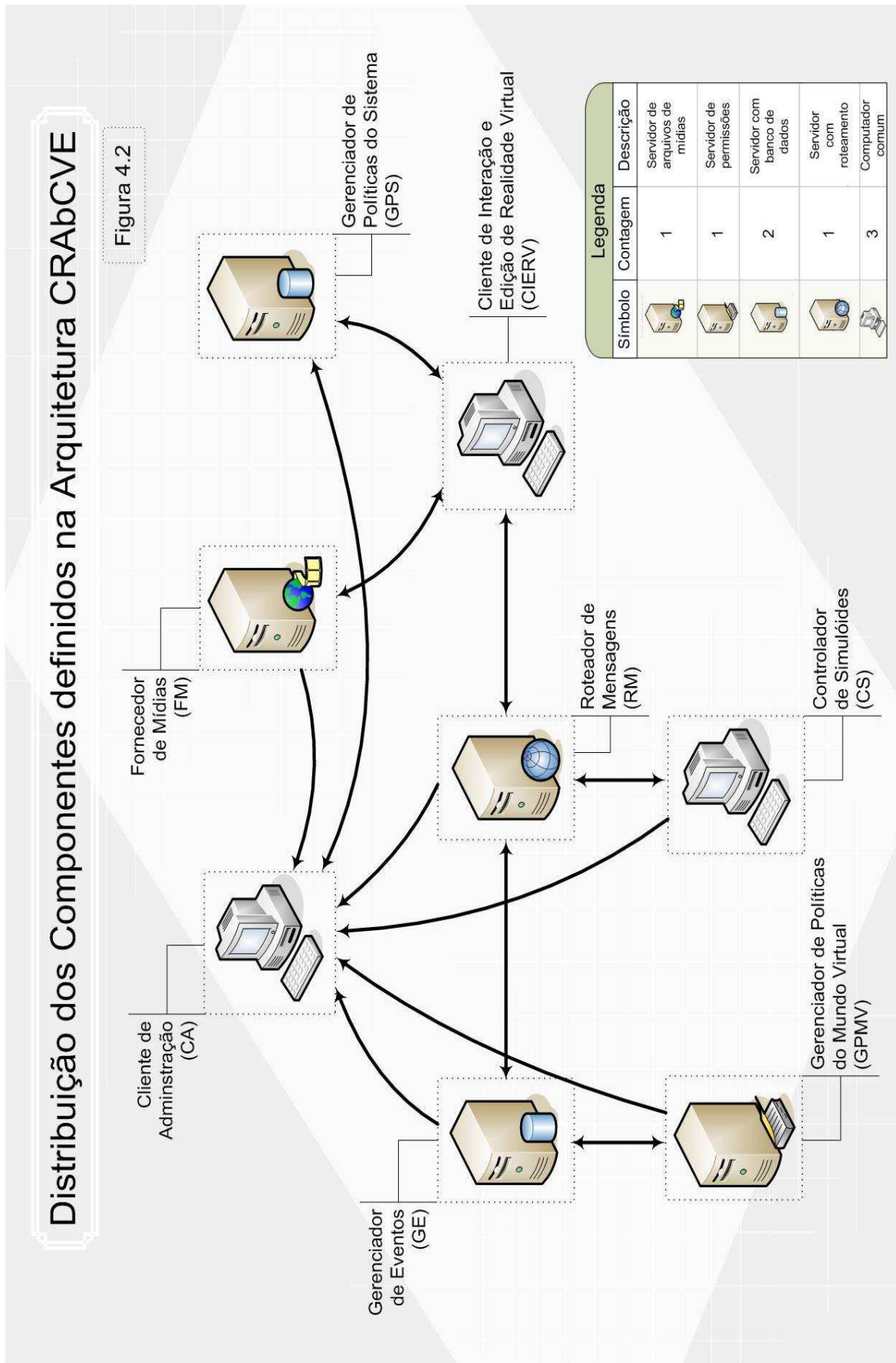


Figura 4.2. Distribuição dos componentes definidos na arquitetura CRAbCVE.

Controle Automático de Instâncias de CVEs

A arquitetura CRAbCVE disponibiliza simultaneamente mais de um subambiente virtual, através da combinação de múltiplas instâncias de seus componentes. Na arquitetura, cada subambiente é tratado como um ambiente virtual separado, com suas instâncias de componentes – a Seção 4.5.2 Recomendações de Uso, mostra os componentes necessários para o funcionamento de um CVE. Esse aspecto pode ser explorado tanto para ofertar diversos subambientes isolados, quanto para disponibilizar uma rede de subambientes interligados que representem um único grande CVE. Combinada com a especialização dos componentes que controlam o funcionamento do CVE, esse aspecto torna a arquitetura CRAbCVE extremamente capaz de sustentar o crescimento de demanda por CVEs.

O GPS possui mecanismos automáticos que instanciam, configuram e desinstanciam CVEs. Assim, quando é necessária a instanciação de um CVE, o GPS automaticamente instancia e configura nos servidores disponíveis na rede os componentes necessários para o funcionamento do CVE. Para que isso seja possível, cada servidor da rede possui uma fábrica de CVEs (Almendra, 2003). Um CVE instanciado fica hospedado no mesmo servidor da fábrica que o instanciou (Figura 4.3).

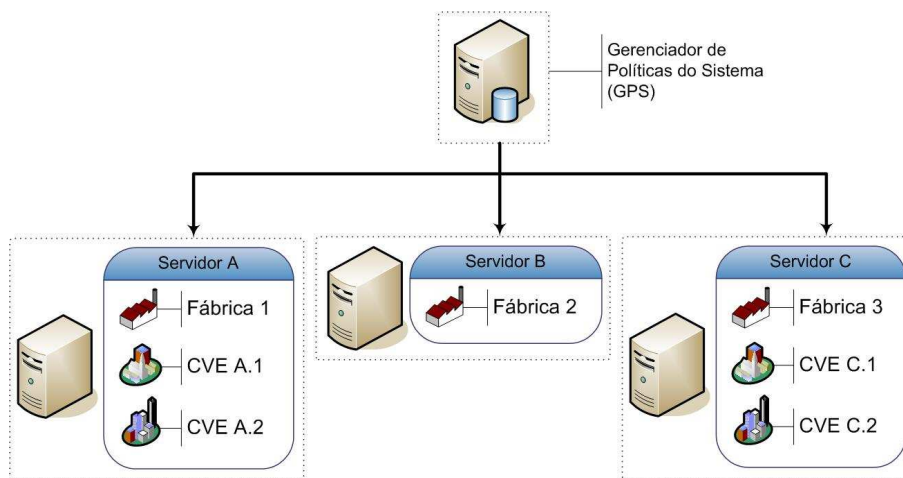


Figura 4.3. Distribuição de fábricas e CVEs (Baseado em Almendra, 2003).

Cada servidor com uma fábrica instalada torna-se um potencial hospedeiro de CVEs. É possível ter servidores na rede com o mesmo número de CVEs, mas com quantidades de usuários conectados bastante diferentes, sobrecarregando alguns servidores. Para evitar sobrecarga de CVEs alocados nos servidores da rede, o GPS decide no momento da instanciação do CVE qual o servidor com maior disponibilidade para alocar uma nova instância de CVE. Um outro problema possível é a falha de algum servidor da rede. Nesse caso, as instâncias de CVEs alocados nesse servidor devem ser migradas para

outro servidor. A fim de automatizar o balanceamento de carga e a recuperação de falhas dos servidores na rede, processos de migração e replicação de CVEs entre os servidores, que não são tratados nesse trabalho, podem ser implementados nos sistemas que se utilizem da arquitetura CRAbCVE.

Localização de CVEs

Para uma efetiva distribuição dos componentes dos CVEs na rede, é necessário tratar o problema da transparência de localização entre os diversos componentes. Transparência de localização refere-se à possibilidade do sistema distribuído de encapsular a localização real dos diversos serviços disponíveis, a fim de minimizar o impacto, no funcionamento do sistema, de uma mudança na localização ou mesmo de uma replicação desses serviços (Tanenbaum, 1995).

O Serviço de localização do GPS apresenta o conceito de “referência de objeto”, que serve para identificar o endereço real de um objeto na rede, de forma que, para se ter acesso a um CVE, não é necessário conhecer sua localização real, mas sim sua identificação ou referência. Esse modelo de localização fica ainda mais flexível quando as referências de objetos são associadas a nomes únicos, de maneira que um objeto possa ser encontrado a partir de seu nome no sistema. Para disponibilizar esse recurso, o GPS utiliza o Serviço de Nomes (*Naming Service*) – um mecanismo para publicação de referências de objetos que possibilita a identificação e a localização de objetos no sistema a partir de seus nomes (Balen, 2000).

Com o serviço de nomes, o GPS fica responsável por armazenar os registros de endereçamento dos CVEs e das fábricas existentes no sistema. No esquema da Figura 4.3, o serviço de nomes guarda tanto as informações de endereçamento de todos os sete objetos (fábricas 1, 2 e 3; CVEs A.1, A.2, C.1 e C.2) existentes, quanto as informações de cada CVE ou fábrica.

A Figura 4.4 apresenta um diagrama de implementação que resume as funcionalidades do componente Gerenciador de Políticas do Sistema.

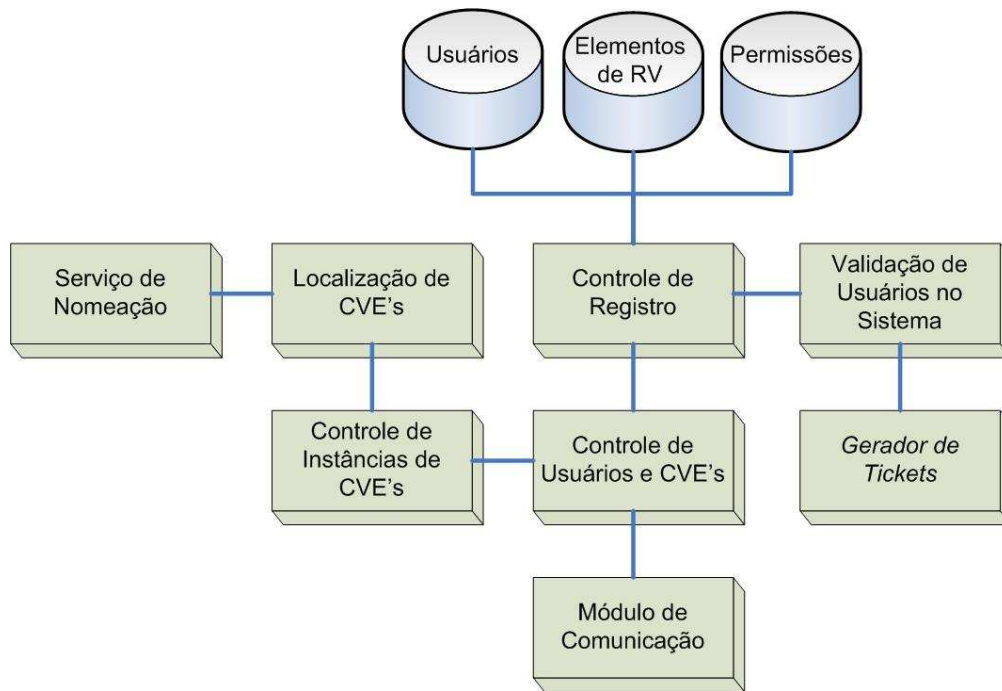


Figura 4.4. Diagrama de implementação do componente GPS.

4.4.2 Cliente de Interação e Edição em Realidade Virtual

Os recursos do ambiente virtual estão disponíveis para os usuários através do Cliente de Interação e Edição em Realidade Virtual (CIERV), que se conecta diretamente ao Roteador de Mensagens – componente descrito na Seção 4.4.3. Essa aplicação cliente faz uso de uma interface gráfica tridimensional (GUI – *Graphic User Interface*) para a realização de todo o processo de geração e apresentação de um mundo virtual.

A principal modificação realizada no cliente de interação da Ataxia (CIRV) foi a inclusão de edição no CIERV, permitindo que os participantes realizem autoria colaborativa de mundos virtuais. A autoria de mundos virtuais no CIERV dá-se pela inserção, remoção e manipulação (escala, rotação, translação) de objetos dentro do ambiente virtual. O processo de autoria provê o suporte apropriado para o trabalho colaborativo entre um grupo de participantes, mas não é suficiente para a realização eficiente de trabalho colaborativo. Assim, mecanismos de coordenação, comunicação e percepção são inseridos no CIERV. Os mecanismos de coordenação baseiam-se na divisão de tarefas e hierarquia de grupos. Os mecanismos de Comunicação fazem uso de recursos de comunicação verbal e não verbal. Por último, os mecanismos de percepção baseiam-se na própria percepção fornecida pela RV e pelo histórico de tarefas. No Capítulo 5, é detalhado um modelo de autoria.

O CIERV pode ser integrado com outras ferramentas não necessariamente voltadas ao CVE, como as ferramentas para a apresentação de mídias específicas, planilhas eletrônicas, ferramentas educativas e muitas outras. Sempre que algum tipo de mídia específica (vídeo, por exemplo) não puder ser apresentado no interior do ambiente virtual, uma janela exibindo a respectiva ferramenta responsável por essa tarefa é aberta na própria GUI, para a apresentação dessa mídia ao participante. Essa integração pode ser realizada a partir da inclusão de rotinas específicas para a troca de informações entre as aplicações externas e a própria aplicação-cliente. Caso essas aplicações integradas necessitem utilizar informações da rede, PDUs especiais do protocolo empregado na arquitetura CRAbCVE podem ser utilizadas para prover tais dados.

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 Fornecedor de Mídias – componente descrito na Seção 4.4.6. O CIERV possibilita aos participantes adicionarem seus próprios objetos durante a autoria de mundos virtuais. No momento em que um novo objeto é inserido em um mundo virtual, ele é transferido automaticamente para o Fornecedor de Mídias a fim de se apresentar nas cópias do mundo virtual dos outros participantes.

Além de suportar hardware para interações simples, baseadas na combinação mouse-teclado-monitor (*Desktop Virtual Reality*), O CIERV pode ser estendido para fazer uso também de equipamentos específicos para RV imersiva (HMD e *data glove*). Isso é possível porque todo o processamento necessário a esse tipo de suporte a hardware se dá na própria aplicação-cliente.

Uma estrutura importante do CIERV é sua *cache* de direitos de acesso, que evita consultas desnecessárias a outros componentes do CVE. Como previamente mencionado, a presença dessa *cache* minimiza o tráfego na rede, pois evita consultas sobre restrições de acesso sempre que um usuário tenta fazer uso de algum objeto ou portal¹⁰.

A Figura 4.5 apresenta um diagrama de implementação que resume as funcionalidades do componente Cliente de Interação e Edição em Realidade Virtual.

¹⁰ Portal é um elemento por meio do qual o usuário é transportado a um outro ambiente virtual.

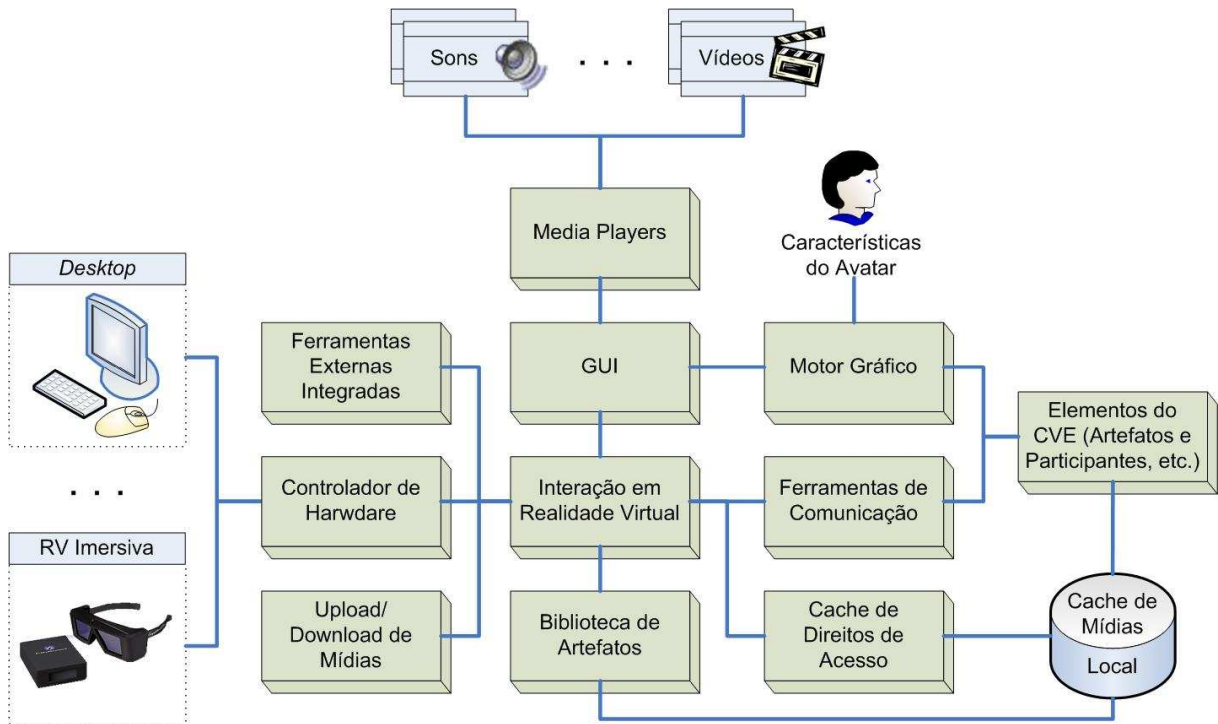


Figura 4.5. Diagrama de implementação do componente CIERV.

4.4.3 Roteador de Mensagens

O Roteador de Mensagens (RM) é o componente responsável por tornar as conexões de um determinado CVE disponíveis aos seus diversos participantes. Esse componente é empregado no repasse de informações a outros elementos da arquitetura CRAbCVE a fim de manter o estado compartilhado do CVE e de viabilizar a colaboração entre participantes.

O critério de roteamento é baseado em uma lista persistente de serviços disponíveis, presente no RM, e na análise das PDUs que trafegam na rede. Ao receber uma determinada PDU, o RM identifica seu tipo e consulta a lista de serviços para identificar os componentes da arquitetura que devem recebê-la. O reenvio direto da PDU é, então, realizado de forma automática ao destino. Assim, devido ao seu caráter de integração de componentes, uma das funções básicas do RM é mediar a comunicação entre os participantes do ambiente.

O RM é um dos componentes essenciais para o funcionamento de um CVE. Assim, quando o GPS instancia um CVE nos servidores da rede através das fábricas, automaticamente um RM é instanciado e configurado. A configuração de um RM dá-se pelo fornecimento da lista persistente de serviços disponíveis e pela indicação do Fornecedor de Mídias – componente descrito na Seção 4.4.6 – que os CIERVs dos participantes devem utilizar para a obtenção de arquivos específicos.

O RM, tal como o Gerenciador de Eventos – componente discutido na Seção 4.4.4 –, é um dos componentes responsáveis por sincronizar as diversas cópias do ambiente virtual presentes nos vários CIERVs a ele conectados. Isso é feito através do repasse, tanto aos participantes quanto ao Gerenciador de Eventos, das PDUs geradas pelas interações realizadas no interior do ambiente virtual. As possíveis PDUs de resposta, oriundas do Gerenciador de Eventos, também fazem uso dos serviços do RM para atingir as aplicações utilizadas por todos os participantes do CVE.

Além disso, o RM pode ser integrado diretamente a ferramentas externas ao CVE. Isso pode ser possível através da re-configuração de sua lista persistente de serviços e da adoção de PDUs especiais do protocolo da CRAbCVE – descrito na Seção 4.5. Assim, quando um novo serviço precisar ser agregado ao CVE, a equipe de administração pode modificar a estrutura da lista de serviços disponíveis, adicionando o novo serviço e, conseqüentemente, permitindo que as PDUs específicas possam ser reenviadas diretamente para esse novo serviço.

A principal modificação realizada no RM da Ataxia foi a inclusão do tratamento de mensagens de edição (PDUs de inclusão, remoção e manipulação de objetos). A Figura 4.6 apresenta um diagrama de implementação que resume as funcionalidades do componente Roteador de Mensagens.

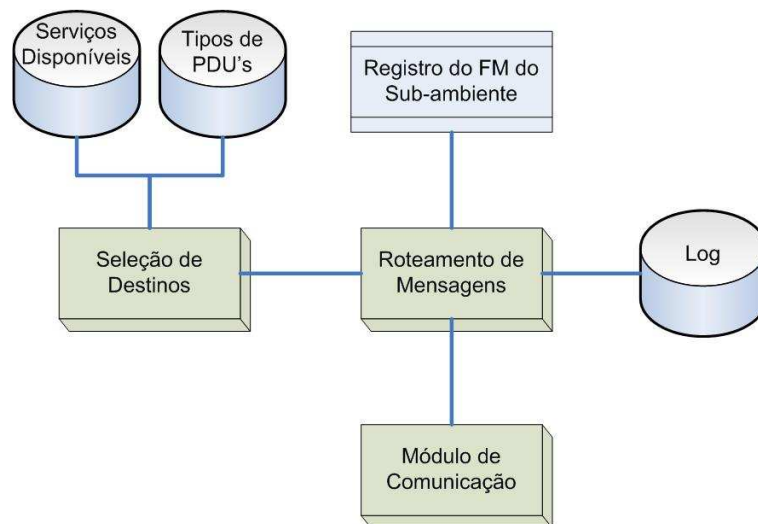


Figura 4.6. Diagrama de implementação do RM.

4.4.4 Gerenciador de Eventos

O Gerenciador de Eventos (GE) é o componente responsável por iniciar o funcionamento do ambiente virtual e por tratar eventos oriundos das interações de seus participantes e da edição de mundos virtuais, gerando PDUs de resposta sempre que necessário. Com esse

fim, o GE armazena: os estados, inclusive a posição, de cada um dos objetos disponíveis no CVE; as definições de portais e nascedouros¹¹ para interconexão de ambientes; e também informações referentes a avatares e simulóides.

Todos os eventos de resposta gerados pelo GE são controlados por estruturas de avaliação de ocorrências e armazenados na forma de *scripts* simples e específicos. Esses *scripts* também são utilizados para a configuração inicial do próprio ambiente virtual quando o gerenciador de eventos é iniciado. O GE também pode controlar a realização de modificações periódicas no estado de objetos (um relógio de parede, por exemplo) através de *scripts* simples. Um mantenedor de tempo global (Universal Time, 2000), também presente no GE, pode controlar as ocorrências de mudança de estado dos objetos de comportamento periódico. O CIERV também apresenta um relógio interno, que é sincronizado com o mantenedor de tempo do GE, no momento em que o CIERV conecta-se a um ambiente virtual.

O GE é ainda utilizado para a interconexão de subambientes, através de portais em um CVE. Quando um determinado participante entra na área definida por um portal, o GE analisa essa ocorrência e envia uma PDU específica para o CIERV do participante. Essa PDU contém informações referentes ao novo ambiente virtual na qual o CIERV do participante deve se conectar.

A fim de otimizar a consulta às informações sobre objetos e participantes, o GE, faz uso de uma base de dados, a Base de Informações de Gerenciamento do Ambiente Virtual (BIGAV). O conceito de BIGAV tem sua origem na idéia de MIB (*Management Information Base*), presente em sistemas voltados ao gerenciamento de redes de computadores (Stallings, 1998). A MIB é utilizada para armazenar o estado atual, capturado através da rede, dos diversos recursos gerenciados (computadores, roteadores, hub's, serviços específicos, etc.). A BIGAV, que assim como a MIB é uma estrutura de banco de dados, concentra todas as informações, incluindo os estados dos vários elementos existentes no ambiente virtual. Entre as funcionalidades da BIGAV, podem-se destacar três principais: simplificação do processo de busca de informações, geração de informações sobre o CVE para novos usuários, e reconstituição automática do CVE em caso de falha.

O uso da BIGAV simplifica o processo de aquisição de informações sobre determinados elementos do CVE por parte dos participantes, evitando que os estados desses elementos sejam pesquisados diretamente nas várias máquinas espalhadas pela rede,

¹¹ Nascedouros são posições no mundo virtual de onde objetos ou personagens virtuais podem aparecer.

o que necessariamente implicaria em atraso. Assim, sempre que um usuário deseja saber algo sobre um determinado elemento, a PDU correspondente é encaminhada ao GE, a respectiva consulta é realizada junto à BIGAV e o resultado é enviado como resposta ao usuário.

Quando um novo participante conecta-se ao sistema, o GE consulta o Gerenciador de Políticas do Mundo Virtual – componente discutido na Seção 4.4.5 – e estabelece sobre quais elementos aquele participante pode realizar ações. Então, a partir de consultas realizadas à BIGAV, o GE gera PDUs que especificam os estados atuais dos diversos objetos e as possibilidades de manipulação dos mesmos por parte do participante que está adentrando o ambiente virtual. Assim, a aplicação do participante é informada sobre as permissões de acesso a objetos (armazenadas na *cache* de direitos de acesso do CIERV) e é garantida a consistência das informações de sua respectiva cópia do CVE.

A BIGAV, que armazena todos os dados referentes aos diversos elementos presentes no ambiente virtual, pode ser utilizada para a reconstituição automática do CVE em casos de quedas do sistema.

Pode-se constatar que o GE é também um dos componentes essenciais para o funcionamento de um CVE. Portanto, quando um CVE é instanciado pelo GPS, um GE também é automaticamente instanciado e configurado. A configuração de um GE dá-se pelo fornecimento da base de dados da BIGAV, que pode ser um arquivo de texto com a descrição completa do mundo virtual e seus elementos, que incluem objetos, portais, nascedouros, etc.

Entre as principais modificações realizadas no GE da Ataxia estão: tratamento de eventos de edição (inclusão, remoção e manipulação de objetos) e a persistência de estado para facilitar a autoria em CVEs. A Figura 4.7 apresenta um diagrama de implementação que resume as funcionalidades do componente Gerenciador de Eventos.

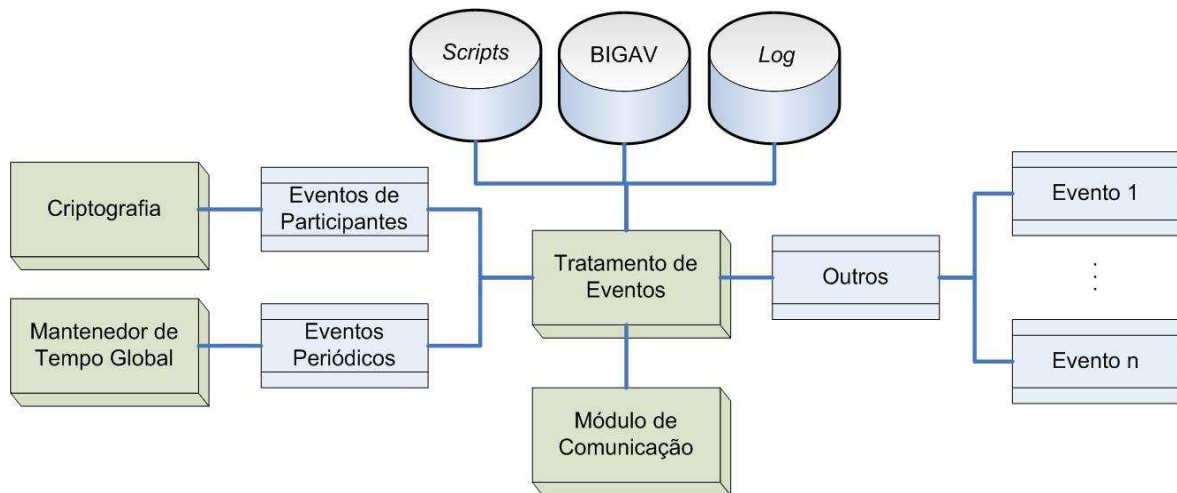


Figura 4.7. Diagrama de implementação do componente GE.

4.4.5 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 objetos do CVE, ou tenha acesso a determinados subambientes. 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.

Durante a edição de um mundo virtual, por exemplo, um participante adiciona um objeto ‘porta’ ao mundo virtual e atribui um nível de permissão que apenas ele, o proprietário, e o grupo a que ele pertence têm direito de interagir com a ‘porta’. Assim, quando esse mundo virtual for publicado, essa restrição de acesso do objeto porta será publicada junto com ele. Conseqüentemente, quando um CVE for instanciado com esse mundo virtual, o GPMV receberá essa restrição em sua lista de restrições de acesso.

Cada elemento do mundo virtual recebe um nível de direitos de acesso que, de acordo com o tipo de participante ou grupo, possibilita diferentes níveis de interação com esse elemento. O estabelecimento de restrições de acesso pode também ser temporal. Por exemplo, um ambiente educacional possui horários escolares dos diversos alunos e professores que participam do CVE. Assim, além de restrições de acesso comuns, podem ser incluídas restrições baseadas em horários específicos. Se, por exemplo, um determinado aluno deve ter aula de química das 14:00 às 16:00 horas, nesse horário, ele não pode entrar em nenhum outro recinto virtual que não aquele onde ocorrerá sua atividade.

Como já foi explicitado anteriormente, o processo de distribuição dos dados de restrições está baseado no uso de *caches* de direitos de acesso. Sempre que um novo usuário conecta-se ao ambiente, o GE solicita ao GPMV a lista de todas as possíveis restrições de acesso e envia essas informações, através do RM, à *cache* de direitos de acesso presente no CIERV. Assim, evita-se que novas consultas sejam realizadas ao GPMV para aquele mesmo usuário, bastando apenas consultar sua *cache* de direitos de acesso. Essa mesma técnica de armazenamento local de permissões é adotada para o recebimento das especificações dos portais presentes no ambiente virtual e, também, para as possíveis permissões de inclusão de mídias no CVE.

Ao contrário da *cache* de mídias local, a *cache* de direitos de acesso não é persistente, e as possíveis restrições são recebidas sempre que o usuário se conecta ao CVE. Toda vez que um elemento tiver seu nível de direitos de acesso modificado (durante a autoria, por exemplo) uma mensagem do protocolo de comunicação é enviada aos outros participantes para atualizarem sua *cache* local.

O GPMV da CRAbCVE equivale ao GPA da Ataxia. A principal modificação realizada no GPA foi o uso de níveis de direitos de acesso e a alteração desses níveis durante a autoria em CVEs. A Figura 4.8 apresenta um diagrama de implementação que resume as funcionalidades do componente Gerenciador de Políticas do Mundo Virtual.

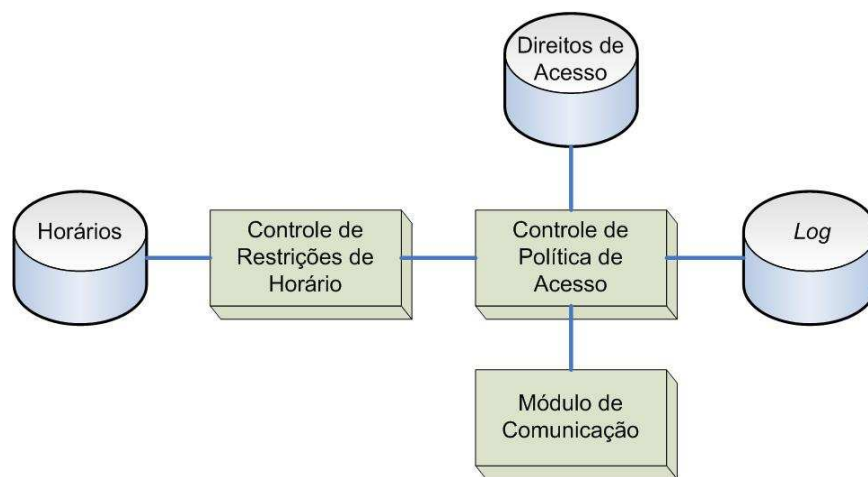


Figura 4.8. Diagrama de implementação do componente GPMV.

4.4.6 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 objetos, e listas de permissões dos elementos RV em cada mundo virtual.

Como já mencionado, o GPS possui o cadastro de todos os elementos armazenados no FM e os direitos de acesso desses elementos. Assim, o GPS e o FM trabalham como se fossem um só componente, complementando um ao outro. Entretanto, a divisão desses componentes é necessária devido à otimização do tráfego de rede, já que a transferência de arquivos de mídias consome muita largura de banda, e o GPS é um componente centralizador de conexões.

Todos os arquivos de mídias são organizados em categorias específicas e indexados tanto no GPS como no FM, a fim de se agilizar o processo de busca dos mesmos. A definição da estratégia de indexação e a seleção dos padrões de mídias suportados no CVE ficam a cargo das equipes de implementação e administração. A inclusão de novos arquivos é realizada tanto pela equipe de administração do CVE quanto pelos próprios participantes do ambiente através da adição de objetos durante a autoria de mundos virtuais.

A distribuição dos elementos do FM para os CIERVs adota como estratégia de *download* a replicação sob demanda. Os dados replicados são, então, armazenados na *cache* de mídias local persistente, como mencionado anteriormente. Assim, sempre que um participante precisar utilizar algum tipo de mídia, o sistema, através de uma aplicação específica executada localmente, inicialmente confere se a mesma não se encontra na *cache* de mídias local. Se o respectivo arquivo já estiver presente nessa *cache*, o mesmo é automaticamente apresentado. Caso contrário, o CIERV apresenta um curinga e realiza, então, o *download* do arquivo necessário a partir do FM.

Os curingas são arquivos definidos de acordo com cada tipo de mídia disponível no ambiente, e podem representar momentaneamente um determinado arquivo, enquanto o mesmo não se encontra disponível na *cache* de mídias local. O conteúdo do curinga deve ser bastante simples, funcionando como uma representação mínima aceitável do arquivo necessário. Assim, por exemplo, para mídias do tipo som, o curinga pode ser um simples bip. É importante notar que, enquanto não for finalizado o *download* do arquivo de mídia necessário, o sistema somente poderá apresentar o curinga correspondente. Logo, na primeira vez que uma determinada mídia é necessária, ela pode não ser apresentada ao participante enquanto o *download* não for finalizado. Esse tipo de abordagem é necessário devido ao emprego da técnica de replicação sob demanda. A questão da ausência das

mídias nas *caches* locais dos CIERVs pode ser minimizada através da distribuição prévia, através da rede ou armazenadas em meio magnético ou óptico, dos respectivos arquivos.

O *upload* e o *download* dos arquivos armazenados no FM são realizados através de um protocolo de comunicação específico para essa transferência. No caso da Internet, o protocolo adotado pode ser o FTP (*File Transfer Protocol*) ou o HTTP (*HyperText Transfer Protocol*), por exemplo.

A principal alteração feita no FM da Ataxia é a sua integração com o GPS. A Figura 4.9 apresenta um diagrama de implementação que resume as funcionalidades do componente Fornecedor de Mídias.

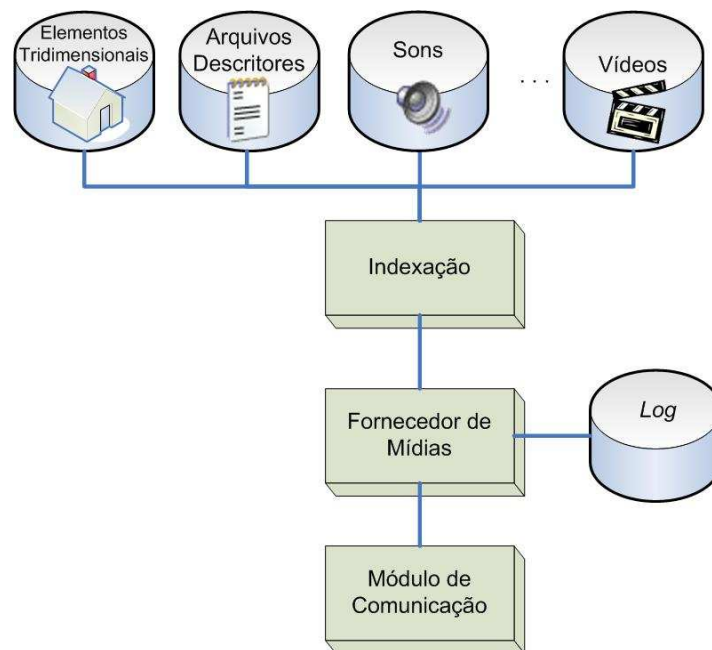


Figura 4.9. Diagrama de implementação do componente FM.

4.4.7 Controlador de Simulóide

Personagens virtuais controlados pelo computador podem estar disponíveis no interior do ambiente virtual. Esses personagens podem desempenhar uma série de tarefas, entre as quais estão a inclusão de recursos originalmente externos ao CVE e o tratamento de eventos interativos de objetos muito complexos. Para que seja possível a existência desses personagens virtuais no ambiente virtual compartilhado, é necessário, para cada um deles, o emprego de um componente Controlador de Simulóide (CS). O CS conecta-se diretamente ao RM e, através dele, pode enviar e receber informações para todos os demais participantes do CVE. Uma explicação detalhada sobre esse componente, incluindo um exemplo prático, encontra-se disponível em (Vidal *et al.*, 2000). Esse componente é igual ao da Ataxia.

4.4.8 Cliente de Administração

Como já mencionado, o funcionamento de todo o CVE depende das tarefas desempenhadas pela equipe de administração. O Cliente de Administração (CA) atua conectando-se diretamente a cada servidor definido na arquitetura ou conectando-se a controladores de simulóides. Uma vez conectado, o CA tanto pode realizar diversas alterações nas configurações do componente, quanto pode acessar as informações nele armazenadas, a fim de realizar análise de funcionamento do sistema. Entre as tarefas desempenhadas pela equipe de administração destacam-se:

- ◆ O cadastro de usuários junto ao GPS, incluindo a adição de informações pessoais, pseudônimo, senha, etc.;
- ◆ A definição de políticas, que tratam tanto dos direitos de acesso através da criação de grupos específicos; quanto do acesso aos CVEs disponíveis, da atribuição de permissões sobre os elementos do FM, e da atribuição de privilégios de autoria de mundos virtuais;
- ◆ A inclusão de elementos de mídia disponíveis no FM, e o registro desses elementos no GPS;
- ◆ A criação de CVEs, com a inserção no FM dos arquivos de um modelo de CVE e seu registro no GPS, servindo como modelo para autoria, e possibilitando a publicação de novos mundos virtuais;
- ◆ A configuração de simulóides em CVEs, definindo as funcionalidades dos componentes controladores de simulóides de um CVE;
- ◆ A configuração da lista de serviços disponíveis do RM de um CVE, englobando a inclusão de serviços dos diversos componentes da arquitetura CRAbCVE e a inclusão de possíveis aplicações externas; e,
- ◆ A análise de *log*, feita através de consultas às informações armazenadas nos diversos componentes da arquitetura CRAbCVE (servidores e controlador de simulóide) para verificar ocorrências específicas e para a auditoria do sistema.

O CA agrega recursos básicos para a realização dessas tarefas, e pode ligar-se diretamente, através de conexões ponto-a-ponto, a qualquer um dos servidores definidos na arquitetura CRAbCVE e dos Controladores de Simulóides utilizados. Ele não exige bastante poder computacional já que todas as informações necessárias à administração do CVE encontram-se distribuídas entre os diversos componentes do CVE, através da rede. Assim, somente tarefas simples, referentes a consultas e envio de PDUs específicas para a

configuração de componentes, precisam ser realizadas. Apesar de não serem originalmente exigidas, operações automáticas de monitoração de ocorrências (o mau funcionamento de um componente, por exemplo) podem estar presentes no CA a fim de facilitar a administração do CVE (Araújo, 2000). Essas tarefas automatizadas, no entanto, podem exigir um maior poder de processamento do CA, fazendo com que o mesmo seja desmembrado em subaplicações específicas, através do paradigma gerente-agente¹².

As principais alterações realizadas no CA da Ataxia foram: cadastro de usuários no GPS, criação de modelos de CVEs e o controle da política de acesso. A Figura 4.10 apresenta um diagrama de implementação que resume as funcionalidades do componente Cliente de Administração.

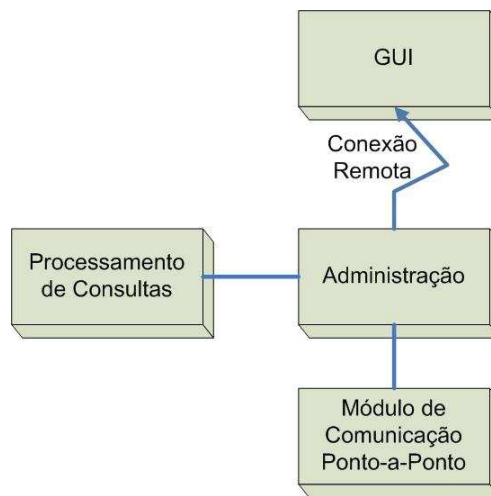


Figura 4.10. Diagrama de implementação do componente CA.

4.5 Funcionamento da Arquitetura CRAbCVE

O conjunto de componentes especificado na arquitetura CRAbCVE pode ser utilizado para disponibilizar, simultaneamente, um ou mais subambientes virtuais que são instanciados e acessados a partir do GPS. Esses componentes podem ser executados em um número ilimitado de computadores, e os subambientes podem ser interligados para formar um único grande CVE. A seguir, são apresentados os principais cenários de utilização da arquitetura para ilustrar seu funcionamento. Posteriormente são apresentadas algumas recomendações básicas que devem ser seguidas para o emprego da arquitetura CRAbCVE.

¹² Estrutura de organização de componentes especializados de softwares (agentes – coletam informações diretamente de dispositivos e gerentes – recebem informações dos agentes) geralmente empregados no gerenciamento de redes.

4.5.1 Cenários de Utilização

Nessa seção, são apresentados os cenários mais importantes que podem ser encontrados durante a utilização da arquitetura. Esses cenários ilustram como os principais recursos de exploração, colaboração e comunicação, encontrados em ambientes de realidade virtual compartilhados, estão disponíveis aos participantes. Para cada cenário, será detalhado o conjunto de ações realizadas por cada componente do sistema envolvido.

Cenário 1: Instalando Fábricas

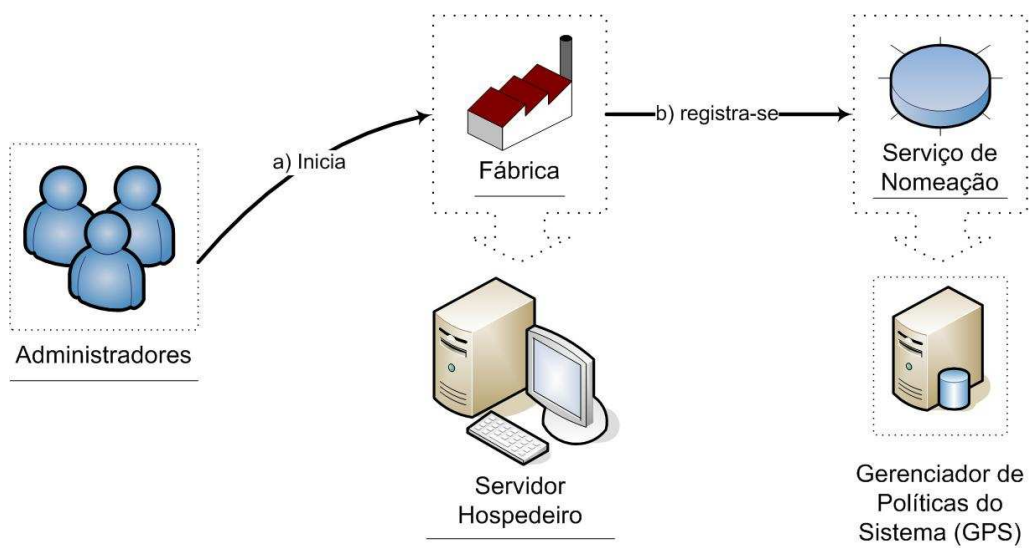


Figura 4.11. Cenário de instalação de fábrica.

Antes de um sistema que faz uso da arquitetura CRAbCVE ser iniciado, as fábricas devem ser instaladas nos servidores hospedeiros de CVEs. A instalação ou criação das fábricas envolve procedimentos de configuração locais nos servidores, o que requer participação da equipe de administradores (Figura 4.11 a). No entanto, a atuação desses administradores ocorre somente no momento da instalação do sistema, ou em períodos de remanejamento de recursos, não sendo necessária sua intervenção no decorrer do funcionamento dos CVEs.

A instalação de fábricas envolve uma simples carga do programa executável ou do processo no sistema operacional, ou ainda, dependendo dos recursos disponíveis, um cadastro da fábrica como um serviço persistente no sistema operacional. Sempre que uma fábrica é instalada, ela cadastra-se no serviço de nomes do GPS (Figura 4.11 b), e a partir de então, pode ser encontrada por ele para instanciar CVEs no servidor hospedeiro da fábrica. Para realizar esse cadastro, a fábrica chama um método do serviço de nomes do GPS, que armazena o nome e a referência da fábrica instanciada passados como parâmetro.

Cenário 2: Iniciando o Sistema

Após as fábricas serem instaladas, o GPS já pode ser iniciado e receber conexões dos participantes. No início do funcionamento do GPS, nenhum CVE é instanciado. A instanciação de CVEs é realizada sob demanda. Sempre que um participante tenta acessar um CVE (Figura 4.12 a), o GPS verifica no seu serviço de nomes se ele já está instanciado (Figura 4.12 b), senão os componentes desse CVE são instanciados a partir das fábricas (Figura 4.12 d) a pedido do GPS (Figura 4.12 c). A Fábrica, ao instanciar um CVE, deve cadastrar o nome e a referência do CVE no serviço de nomes do GPS (Figura 4.12 e), para possibilitar sua localização no sistema.

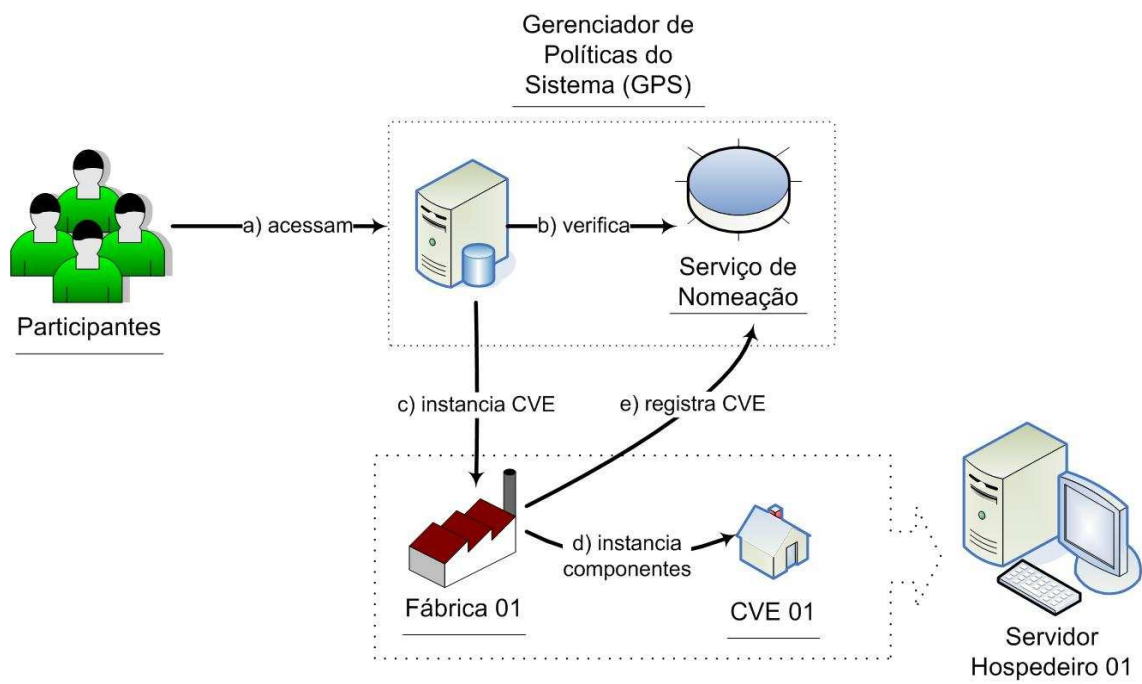


Figura 4.12. Cenário de sistema iniciando.

Assim, para instanciar um CVE, o GPS deve selecionar uma das fábricas disponíveis, consultando, para tanto, a lista de fábricas registradas no serviço de nomes. A decisão de qual fábrica será usada para a instanciação do CVE deve ser feita de acordo com as aplicações envolvidas. Por exemplo, enquanto uma estratégia pode ser orientada para balancear o número de CVEs já instanciados em cada servidor, outra estratégia pode ser orientada para alocar CVEs afins (mesmo público, mesma temática, etc.) em servidores comuns.

O serviço de nomes fornece dois métodos para a identificação dos CVEs e das fábricas disponíveis no GPS: *listar* e *localizar*. O método *listar* retorna todo o conjunto de registros de CVEs e fábricas armazenados no GPS. Através de uma filtragem, é possível

diferenciar as instâncias dos CVEs e das fábricas. O método *listar* não retorna as referências das instâncias, retorna apenas seus nomes. Logo, para acessar efetivamente um CVE ou fábrica, é necessário recuperar sua referência no serviço de nomes através do método *localizar*, o qual recebe como parâmetro o nome da instância. O GPS deve orquestrar esses recursos da forma que for mais adequada para sua aplicação.

Cenário 3: Instanciando CVEs

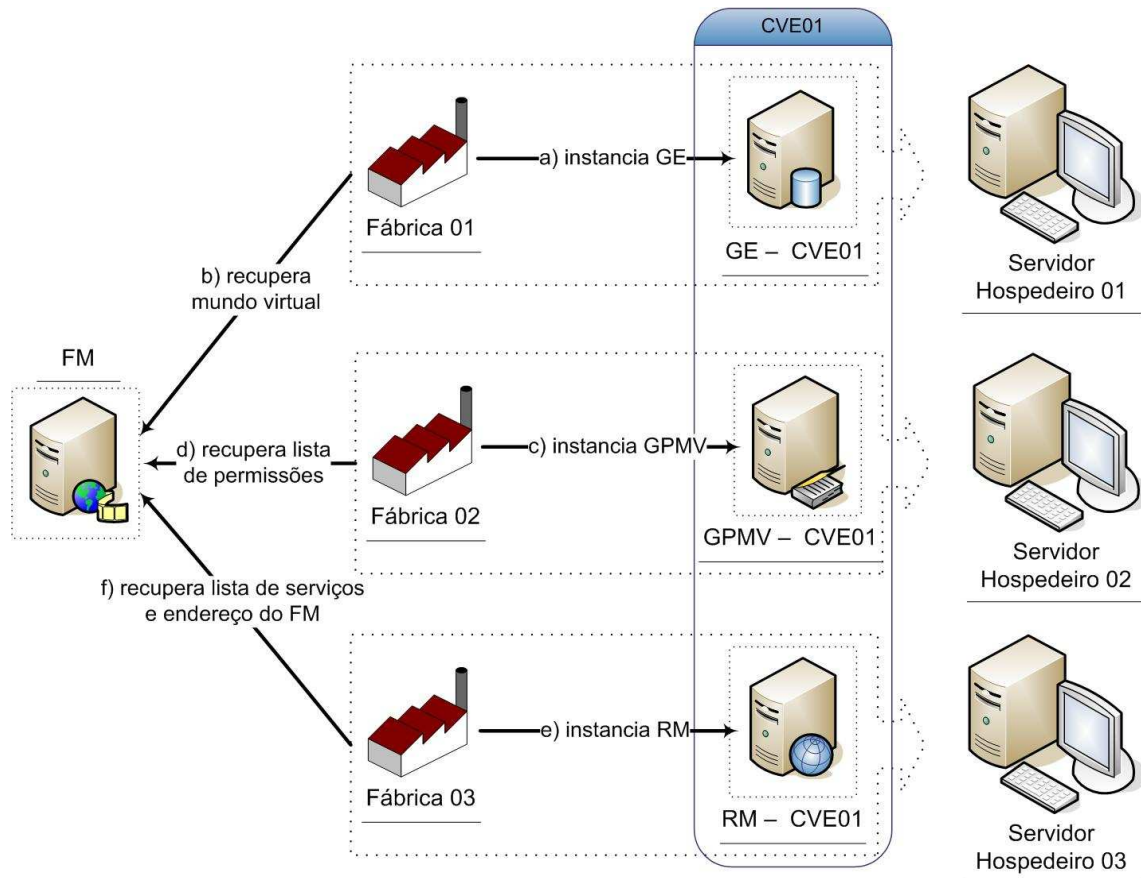


Figura 4.13. Cenário de instanciamento de CVEs.

Depois do GPS ter verificado no serviço de nomes e ter certificado que o CVE a ser instanciado não existe, as fábricas são selecionadas para a instanciamento do CVE. Para instanciar um CVE, o GPS precisa chamar o método *Instanciar* das fábricas selecionadas, que, então, realizarão procedimentos locais de instanciamento do ambiente virtual. O nome do CVE, os componentes a serem instanciados e um identificador do tipo de CVE são passados como parâmetros ao método *Instanciar*. Isso se dá pela necessidade, em alguns casos, de criar réplicas de CVEs. Por exemplo, pode ser necessário criar dois CVEs do mesmo tipo, que terão a mesma estrutura física, mas representarão espaços virtuais distintos.

Como já mencionado, para o funcionamento de um CVE, essencialmente é necessária a utilização conjunta de um Roteador de Mensagens, de um Gerenciador de Eventos e de um Gerenciador de Políticas do Mundo Virtual (Figura 4.14). Esses componentes podem estar instanciados em um mesmo servidor, ou distribuídos em servidores separados, já que podem ser localizados pelo serviço de nomes. Quando uma fábrica instancia um CVE, o primeiro passo para tornar o CVE disponível é configurar o GE, escolhendo o mundo virtual a ser disponibilizado. Todos os modelos de mundos virtuais disponíveis no sistema estão armazenados no FM e possuem um identificador próprio.

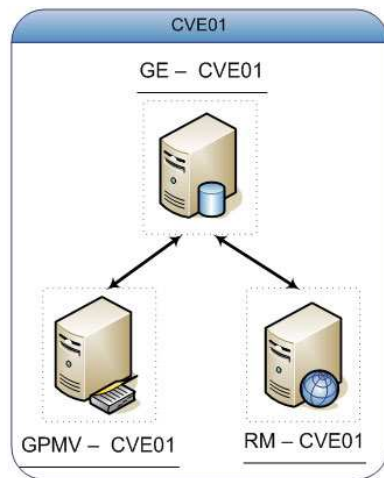


Figura 4.14. Componentes essenciais de um CVE.

Um mundo virtual é definido por um conjunto de arquivos, nos quais estão descritos sua estrutura arquitetônica, os objetos nele incluídos, as permissões de acesso, etc. Assim, Quando o GE é instanciado (Figura 4.13 a), as informações da descrição e composição do mundo virtual selecionado são carregadas para a BIGAV (Figura 4.13 b). É importante ressaltar, que na BIGAV não ficam armazenadas as mídias que compõe o mundo virtual, mas apenas as referências a essas mídias que ficam armazenadas no FM. O segundo passo é iniciar o GPMV (Figura 4.13 c), transferindo para ele as permissões de acesso do mundo virtual selecionado que estão armazenadas no FM (Figura 4.13 d). Finalmente, o RM é instanciado (Figura 4.13 e, f), estabelecendo conexão através de suas portas de comunicação com o GE, e esse com GPMV. Conseqüentemente, ao serem estabelecidas essas conexões, o RM já estará apto a receber pedidos de conexão de clientes. Para que os usuários possam acessar esse novo CVE, as informações de localização do RM são registradas no GPS, cuja localização é conhecida pelas aplicações clientes.

Cenário 4: Cliente Conectando a CVEs

Antes de conectar-se a algum CVE, um participante deve primeiramente conectar-se ao GPS para validar sua entrada no sistema (Figura 4.15 a) e receber a lista de CVEs disponíveis (Figura 4.15 c). Essa lista varia de acordo com os privilégios de cada participante no sistema. Por exemplo, um participante pode ter o privilégio de modificar e publicar mundos virtuais, ou pode apenas explorar os existentes. Cada CVE pode, por exemplo, permitir acesso a apenas um grupo de participantes. Assim, cada participante recebe do GPS uma lista personalizada de CVEs disponíveis de acordo com seus privilégios no sistema (Figura 4.15 b).

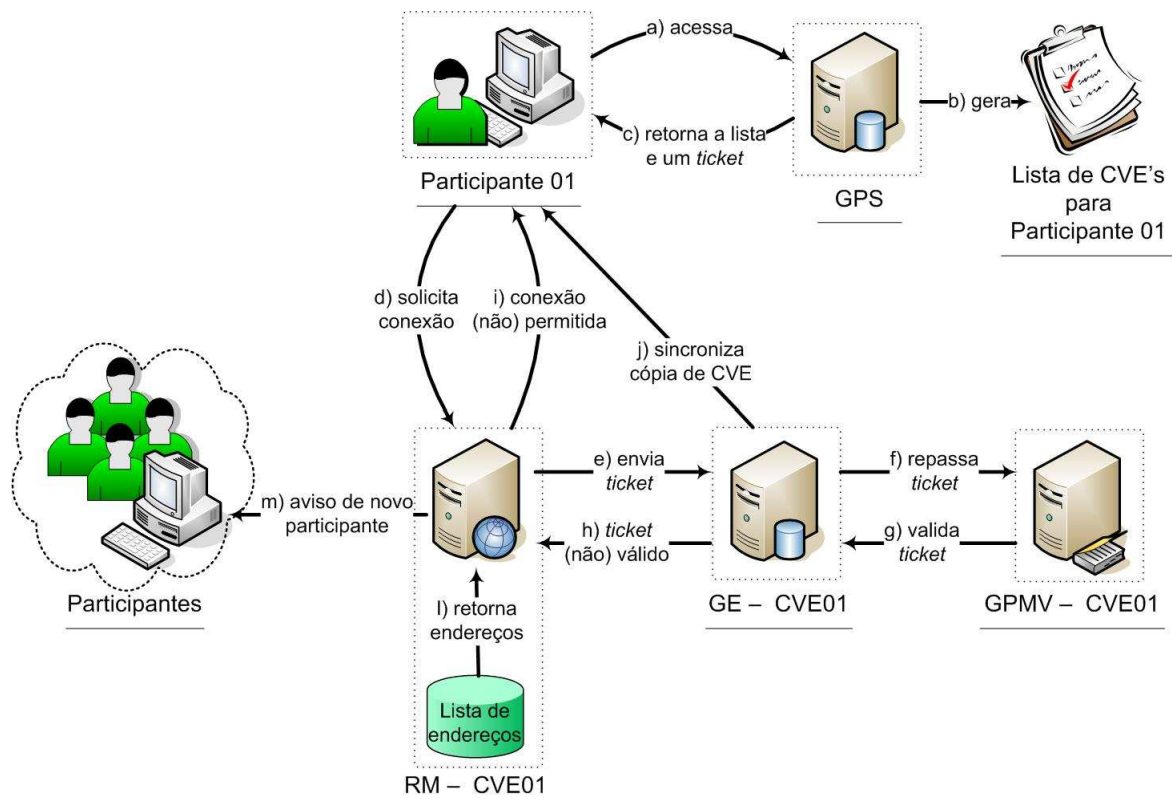


Figura 4.15. Cenário de conexão de clientes a CVEs.

A lista de CVEs fornecida pelo GPS possui referências para CVEs que podem ou não já ter sido instanciados. Ao selecionar um CVE da lista, o participante solicita o endereço de conexão (endereço IP e porta do servidor hospedeiro da rede) ao GPS, o qual, antes de retornar essa informação, verifica se o CVE já está instanciado. Caso o CVE ainda não esteja, ele é instanciado, e somente então seu endereço de conexão é enviado ao participante.

Nesse momento, a aplicação cliente tem conhecimento do RM do CVE escolhido e envia um pedido de conexão (Figura 4.15 d). O RM verifica se o pedido de conexão é

válido, ou seja, originado de um participante autenticado pelo GPS. Essa verificação é realizada através de um mecanismo de *tickets*. O GPS fornece ao CIERV um *ticket* temporário (Figura 4.15 c) que informa o CVE a ser conectado e uma assinatura certificando que esse *ticket* foi emitido pelo GPS. Então, RM envia esse *ticket* para o GE (Figura 4.15 e) que repassa ao GPMV (Figura 4.15 f) para examinar a validade do *ticket*. Se for válido, então a conexão do CIERV com o RM é permitida (Figura 4.15 i), e a entrada do novo participante é informada ao GE. No caso de já existirem outros participantes conectados ao CVE, o RM envia-lhes as informações do novo participante (Figura 4.15 m).

Quando um CIERV tenta conectar-se a um CVE através de um portal de um outro CVE, o GPMV do CVE de origem modifica o campo do *ticket* que informa o CVE a ser conectado para o destino do portal. Logo, o *ticket* continua válido.

Após estar conectado ao RM, o CIERV está apto a sincronizar sua cópia local do mundo virtual com as outras cópias dos participantes (Figura 4.15 l). Inicialmente o CIERV verifica se a versão de sua cópia local é a mesma do CVE instanciado, pois é possível que o mundo virtual tenha sido modificado e, publicado em uma nova versão. Se a versão for diferente, o CIERV deve fazer o *download* do arquivo com a descrição e composição da nova versão do mundo virtual armazenada no FM. No momento em que as versões das cópias são iguais, a sincronização dos estados dos objetos pode ser feita. Essa sincronização se processa quando o GE envia PDUs com a atualização de estado de todos os objetos presentes na lista da BIGAV. Entretanto, se o mundo virtual tiver sido editado (objetos inseridos ou removidos), primeiramente o GE envia as PDUs necessárias para a inserção e para a remoção de objetos, de forma a compatibilizar a cópia local do participante com a cópia do GE (ambas devem conter os mesmos objetos), e só então, atualizar os estados desses objetos. Após a sincronização, o participante está pronto para explorar o CVE.

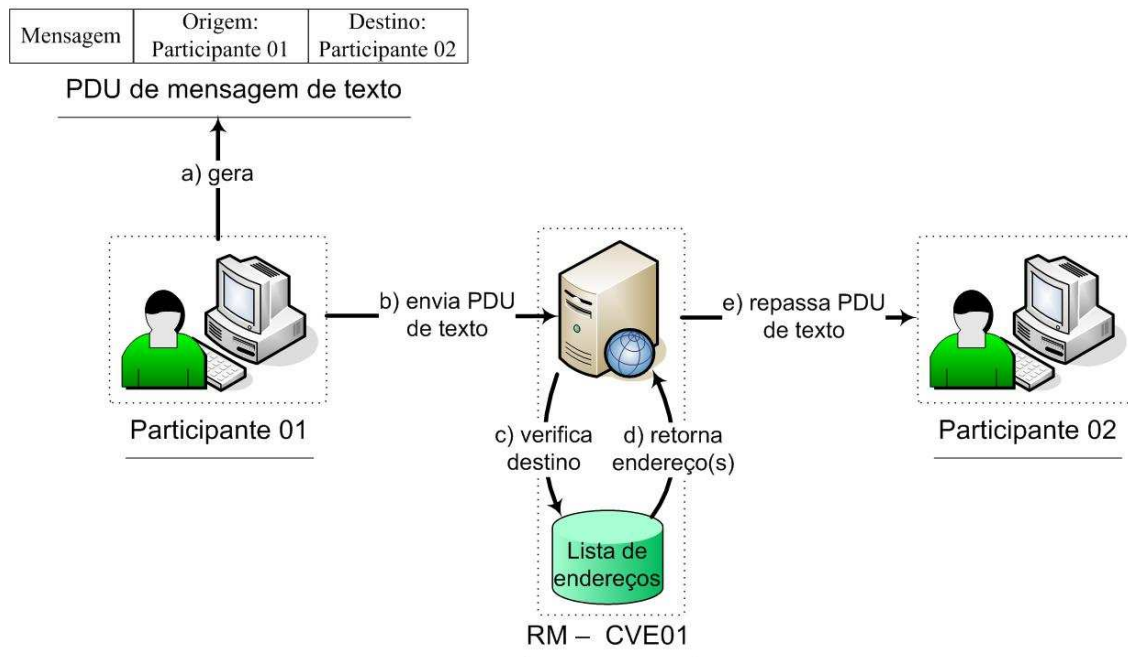
Cenário 5: Cliente Enviando Mensagem de Texto

Figura 4.16. Cenário de envio de mensagem de texto.

Os recursos de comunicação são ferramentas bastante importantes em CVEs, pois, ao permitirem a troca de informações entre os participantes do CVE, possibilitam a composição de esforços para a realização de tarefas em grupo. A forma mais básica de comunicação disponível é a textual. Os participantes digitam a mensagem numa caixa de texto e enviam-na para um determinado participante, para um grupo, ou para todos os participantes do CVE (Figura 4.16 a). Nesse caso, quando a mensagem chega ao RM em forma de PDU, ele verifica o destinatário (ou o conjunto de destinatários – Figura 4.16 c e d) e repassa a mensagem (Figura 4.16 e).

Cenário 6: Cliente Animando o seu Avatar

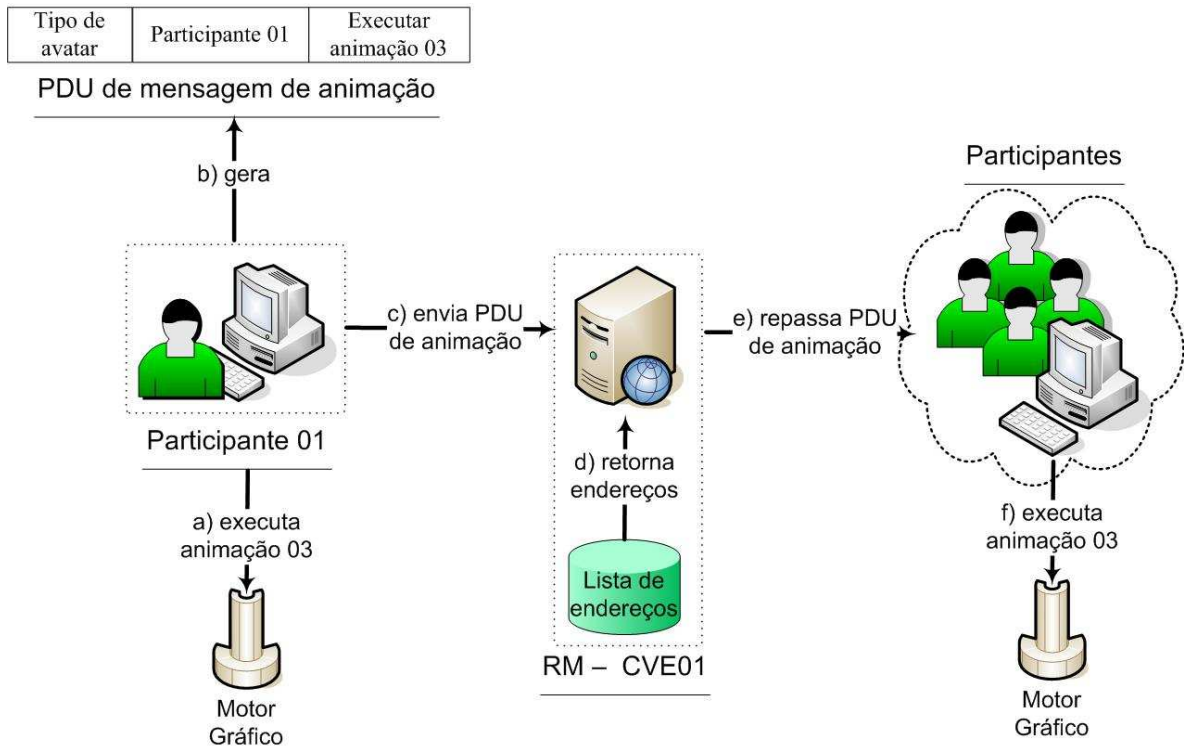


Figura 4.17. Cenário de animação de avatar.

Além de mensagens de texto, os usuários também podem utilizar a comunicação não verbal através da linguagem corporal, inclusive expressões faciais, animando o seu avatar. Assim, é possível acenar, cumprimentar, dançar, sorrir ou demonstrar tristeza ou raiva. Quando um participante escolhe uma animação para seu avatar, é enviada ao RM uma PDU específica para animações, contendo o tipo da entidade (avatar), a identificação do participante e a identificação da animação escolhida (Figura 4.17 b e c). Quando essa PDU chega ao RM, ele a repassa para todas as aplicações clientes do CVE (Figura 4.17 d,e), exceto para a do remetente que executa sua animação localmente (Figura 4.17 a). Finalmente, ao chegar às aplicações clientes, a PDU é processada, para gerar uma chamada de procedimento do motor gráfico a fim de executar a animação do avatar do remetente (Figura 4.17 f).

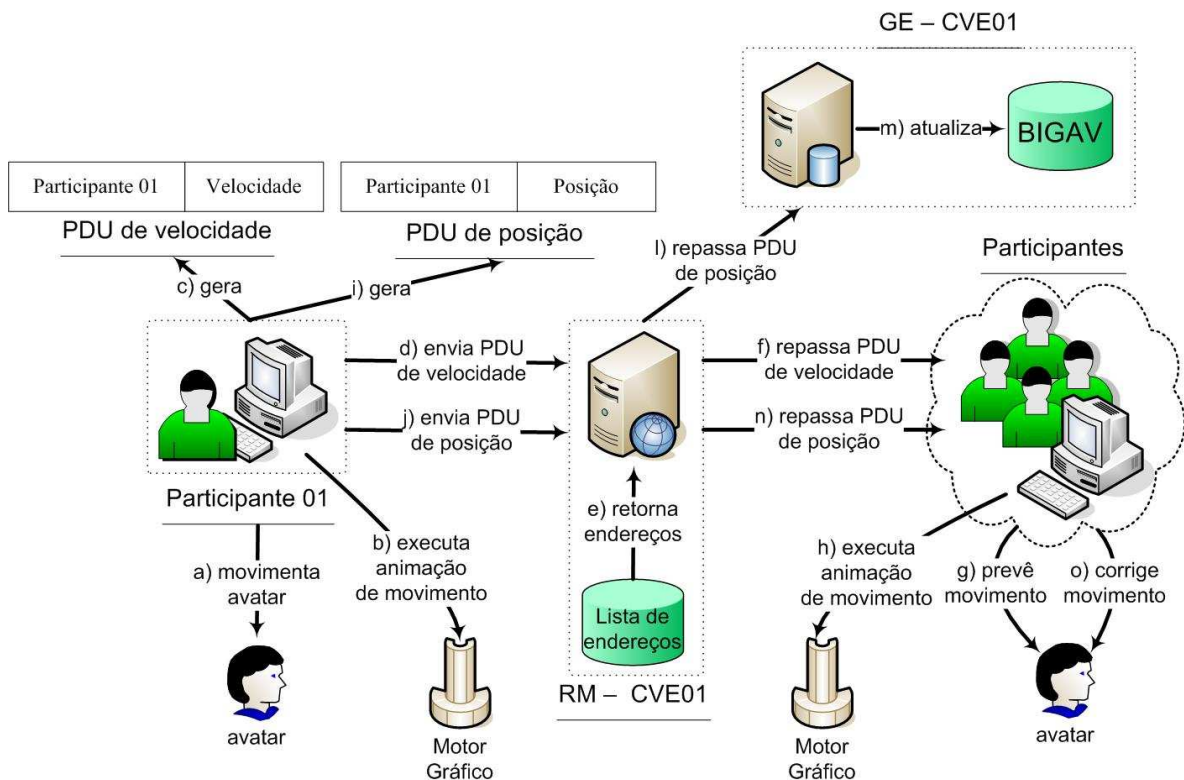
Cenário 7: Cliente Movimentando-se pelo Ambiente Virtual

Figura 4.18. Cenário de movimentação de avatar.

A exploração é uma das prerrogativas para que ocorra a imersão do indivíduo no ambiente virtual e consiste na movimentação de seu avatar por entre os elementos tridimensionais que formam o mundo virtual. Para a otimização do tráfego de rede, foi adotada a técnica de *dead-reckoning* (discutida anteriormente) que necessita do envio à rede de apenas duas PDUs para cada movimentação de um personagem virtual. A primeira PDU, que contém a velocidade linear ou angular inicial do personagem virtual (Figura 4.18 c), é enviada a todos os outros clientes (Figura 4.18 d, e, f) quando o personagem inicia o movimento. A segunda, que contém a posição e a orientação reais do personagem virtual (Figura 4.18 i), é enviada quando ele pára o movimento (Figura 4.18 j) ou quando supera um tempo limite (*timeout*). Até receber essa segunda PDU, cada um dos clientes realiza cálculos relativos sobre o possível caminho percorrido pelo personagem virtual que está se movimentando (Figura 4.18 g). Enquanto o personagem move-se, o motor gráfico da aplicação cliente exibe a animação de movimentar daquele personagem virtual para dar mais naturalidade ao movimento (Figura 4.18 b, h).

Cenário 8: Cliente Interagindo com Objetos

A interação com objetos consiste na possibilidade de avatares e simulóides realizarem processos de manipulação (operações de translação, rotação e escala) e disparo de ações de objetos interativos presentes no CVE. Esses objetos são tratados como entidades dinâmicas do sistema, cujos estados são sincronizados através do RM e gerenciados pelo GE, o qual armazena uma lista desses objetos na BIGAV, controlando seus comportamentos.

No CIERV, a manipulação é feita através de operações com os dispositivos de entrada (mouse e teclado, ou até *data gloves*). Por exemplo, para transladar um objeto, um participante deve ‘agarrá-lo’, movê-lo para a posição desejada e depois ‘liberá-lo’. Nesse momento, a aplicação cliente envia uma PDU com a nova posição do objeto ao RM, o qual a repassa aos outros clientes e ao GE. Se o usuário ficar ‘agarrando’ o objeto superando um tempo limite (*timeout*), uma PDU com a posição atual do objeto é enviada. Os clientes, então, automaticamente transladam o objeto para essa nova posição, e o GE atualiza o estado do respectivo objeto na BIGAV (Cenário semelhante ao de movimentação de avatar, exceto pela predição do movimento – PDU de velocidade). Para um maior realismo na movimentação dos objetos, pode ser utilizado um mecanismo de *Path Finder*¹³ no traslado dos objetos.

Objetos mais especializados podem ter comportamentos mais complexos, permitindo a detecção de aproximação ou reação pré-programada ao toque. Por exemplo, uma porta automática pode ser considerada um objeto interativo. À medida que os personagens movimentam-se pelo mundo, o GE recebe as novas posições dos personagens virtuais e repassa-as a todos os objetos de sua lista. Cada objeto, por sua vez, dependendo do comportamento programado, responde ou não à aproximação do personagem. No caso da porta automática, a resposta é uma mudança de estado (de ‘fechada’ para ‘aberta’) que é enviada a todos os participantes, através do RM, e acarreta a execução de uma animação que mostra a abertura da porta em cada um dos CIERVs conectados ao sistema.

¹³ *Path Finder*, que pode ser traduzido como localizador de caminho, é um mecanismo que encontra o menor caminho entre dois pontos respeitando um conjunto de restrições. No caso de ambientes virtuais, as restrições seriam a colisão com os objetos e com o próprio mundo virtual.

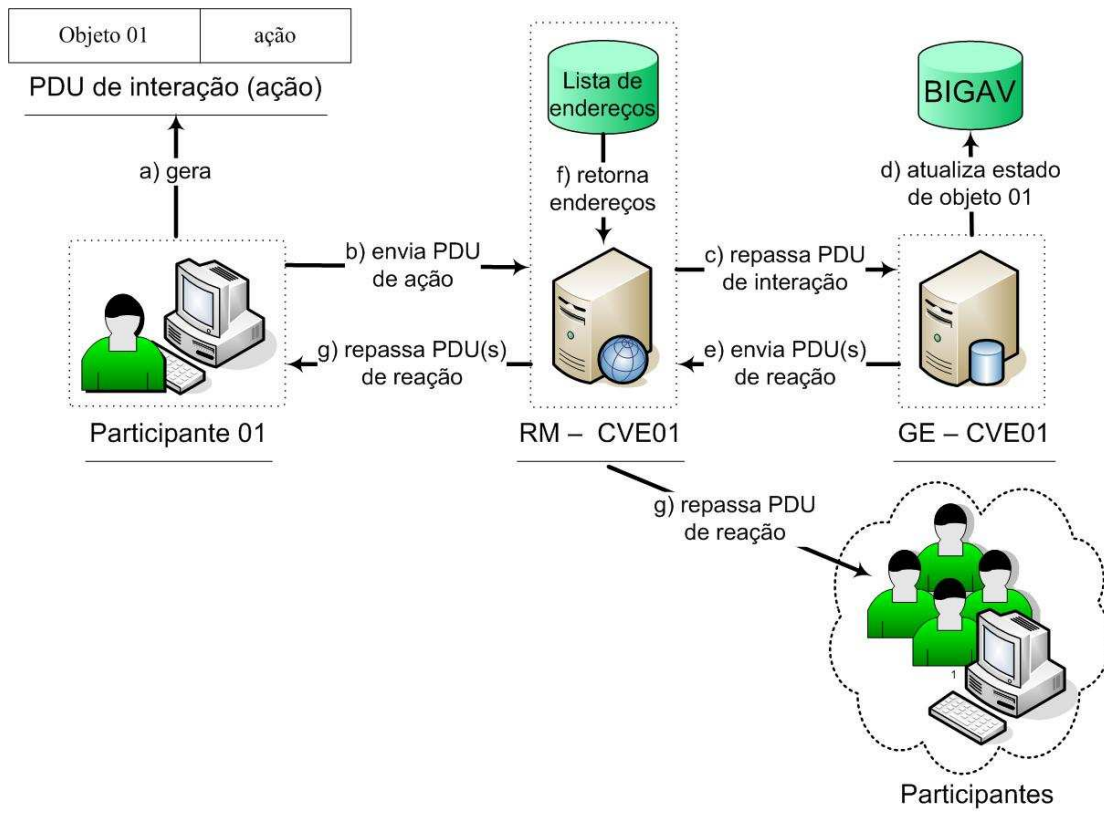


Figura 4.19. Cenário de interação com objetos.

Da mesma forma, a porta poderia ser aberta a partir de uma ação de ‘abrir porta’ proveniente de algum personagem virtual. Tal ação é repassada na forma de PDUs para o GE (Figura 4.19 b, c) que analisa as implicações da ação e modifica o estado do objeto ‘porta’ de ‘fechada’ para ‘aberta’ (Figura 4.19 d). A fim de retransmitir essa modificação a todas as cópias presentes nas aplicações utilizadas pelos participantes do ambiente virtual, o GE cria e envia, através do RM (Figura 4.19 e, f, g), PDUs que determinam a animação de abertura da ‘porta’, a apresentação do som da porta rangendo e, finalmente, a mudança do estado final da porta a todos os participantes.

Cenário 9: Editando Mundos Virtuais

Como já mencionado, o processo de edição é realizado basicamente pela inserção (Figura 4.20), remoção (Figura 4.21) e manipulação (translação, rotação, escala) de objetos no mundo virtual. Quando um participante tem privilégios de edição para inserir ou remover um objeto no mundo virtual, uma PDU é criada e repassada ao GE (Figura 4.20 d, e; Figura 4.21 b, c) que, por sua vez, atualiza sua lista de objetos na BIGAV (Figura 4.20 f; Figura 4.21 d). A PDU de inserção ou remoção de objeto também é repassada, através do RM, para todas as cópias do mundo virtual dos outros participantes (Figura 4.20 g, h; Figura 4.21 e, f). Se por acaso, um participante não tiver a mídia correspondente do objeto

a ser inserido, será então apresentado um curinga (Figura 4.20 i) enquanto o *download* da mídia não for concluído a partir do FM (Figura 4.20 j). O processo de manipulação de objetos é mais simples, ele faz uso de uma PDU de mudança de estado (nesse caso, mudança de posição, orientação ou escala). Logo, quando um participante manipula um objeto, uma PDU é enviada ao RM que a repassa ao GE e aos outros participantes. O cenário de manipulação de objetos tem o mesmo funcionamento do cenário de remoção (Figura 4.21), exceto pelo tipo de PDU e pela funcionalidade.

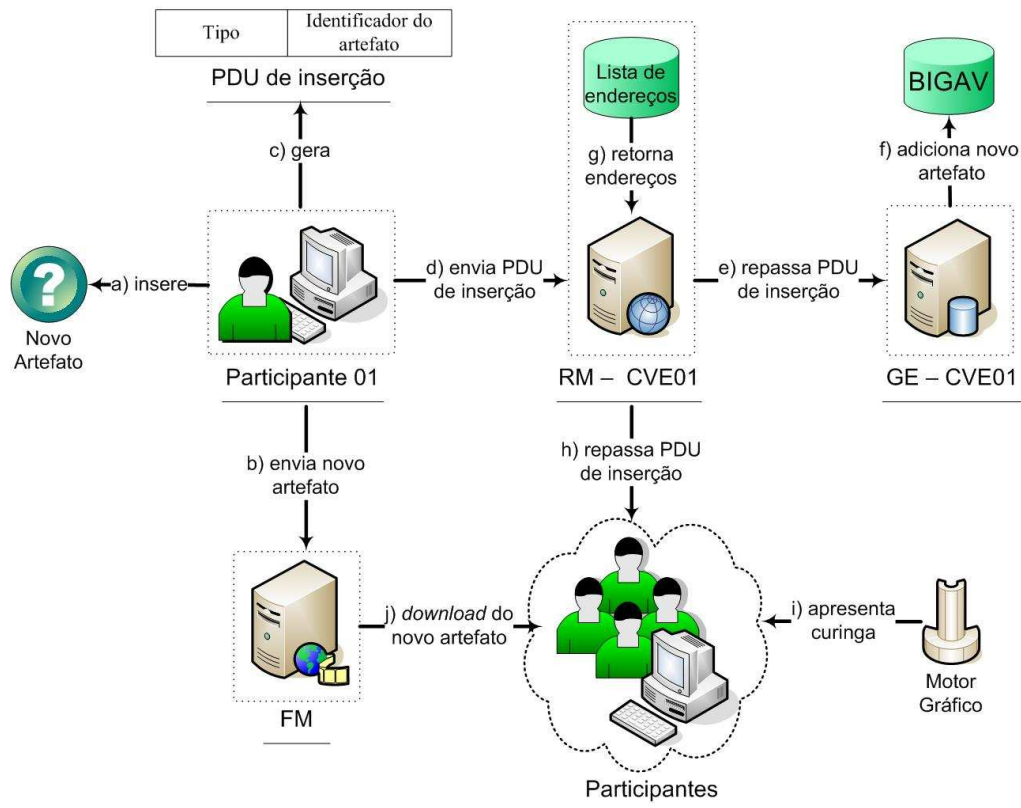


Figura 4.20. Cenário de inserção de objetos.

Como já mencionado, o processo de edição é realizado basicamente pela inserção (Figura 4.20), remoção (Figura 4.21) e manipulação (translação, rotação, escala) de objetos no mundo virtual. Quando um participante tem privilégios de edição para inserir ou remover um objeto no mundo virtual, uma PDU é criada e repassada ao GE (Figura 4.20 d, e; Figura 4.21 b, c) que, por sua vez, atualiza sua lista de objetos na BIGAV (Figura 4.20 f; Figura 4.21 d). A PDU de inserção ou remoção de objeto também é repassada, através do RM, para todas as cópias do mundo virtual dos outros participantes (Figura 4.20 g, h; Figura 4.21 e, f). Se por acaso, um participante não tiver a mídia correspondente do objeto a ser inserido, será então apresentado um curinga (Figura 4.20 i) enquanto o *download* da mídia não for concluído a partir do FM (Figura 4.20 j). O processo de manipulação de objetos é mais simples, ele faz uso de uma PDU de mudança de estado (nesse caso,

mudança de posição, orientação ou escala). Logo, quando um participante manipula um objeto, uma PDU é enviada ao RM que a repassa ao GE e aos outros participantes. O cenário de manipulação de objetos tem o mesmo funcionamento do cenário de remoção (Figura 4.21), exceto pelo tipo de PDU e pela funcionalidade.

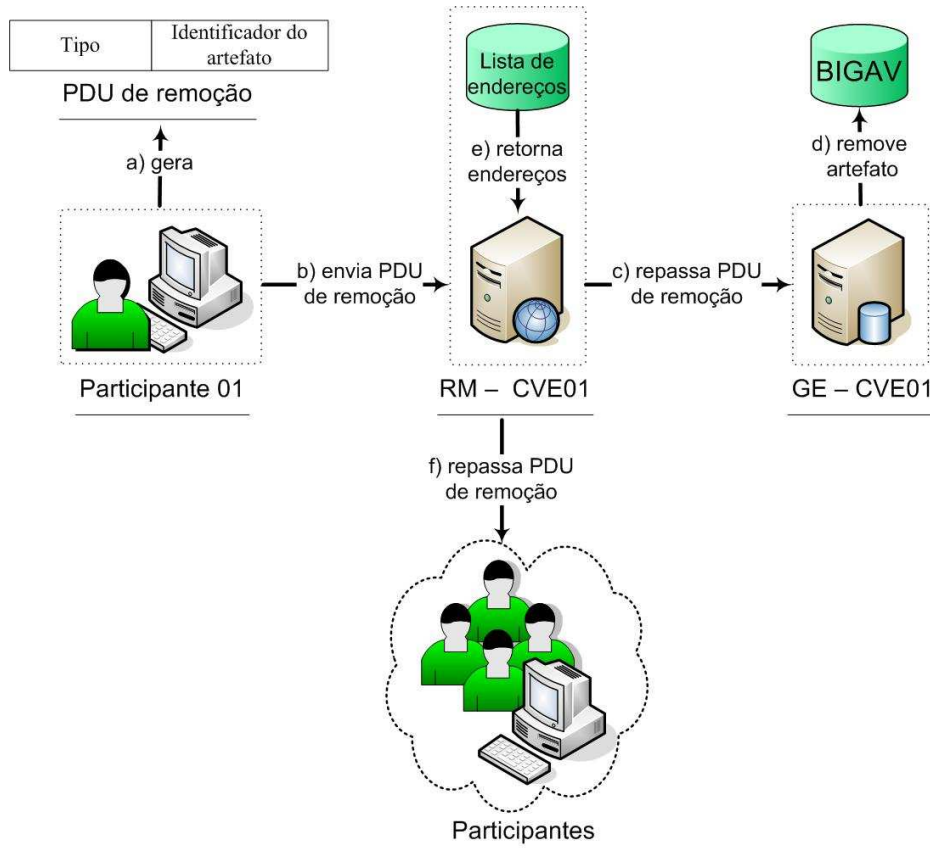


Figura 4.21. Cenário de remoção de objetos.

Cenário 10: Alterando permissões

Permitir que participantes estabeleçam políticas de acesso a objetos inseridos no mundo virtual possibilita um maior realismo às interações dentro do ambiente virtual, pois concretiza um sentimento de propriedade. Ao inserir um objeto no mundo virtual, o participante define o nível de acesso que outros participantes ou grupo terão sobre esse objeto. Para isso, uma PDU é gerada com a identificação do participante e do objeto, e com o nível de permissão e, em seguida, enviada ao RM (Figura 4.22). O RM envia a PDU ao GE que a repassa ao GPMV para atualização de sua lista de permissões. Após o GPMV confirmar a alteração, o GE repassa a PDU a todos os participantes para atualização de suas *cache* locais de permissões. O funcionamento do nível de acesso varia de acordo com a aplicação. Por exemplo, nível de acesso zero dá permissões de interação apenas a seu proprietário; nível de acesso um dá permissões ao proprietário e ao grupo, etc. A PDU de

permissão também se presta para alterar o nível de acesso do próprio mundo virtual, no caso de um proprietário não querer que outros participantes alterem seu mundo virtual.

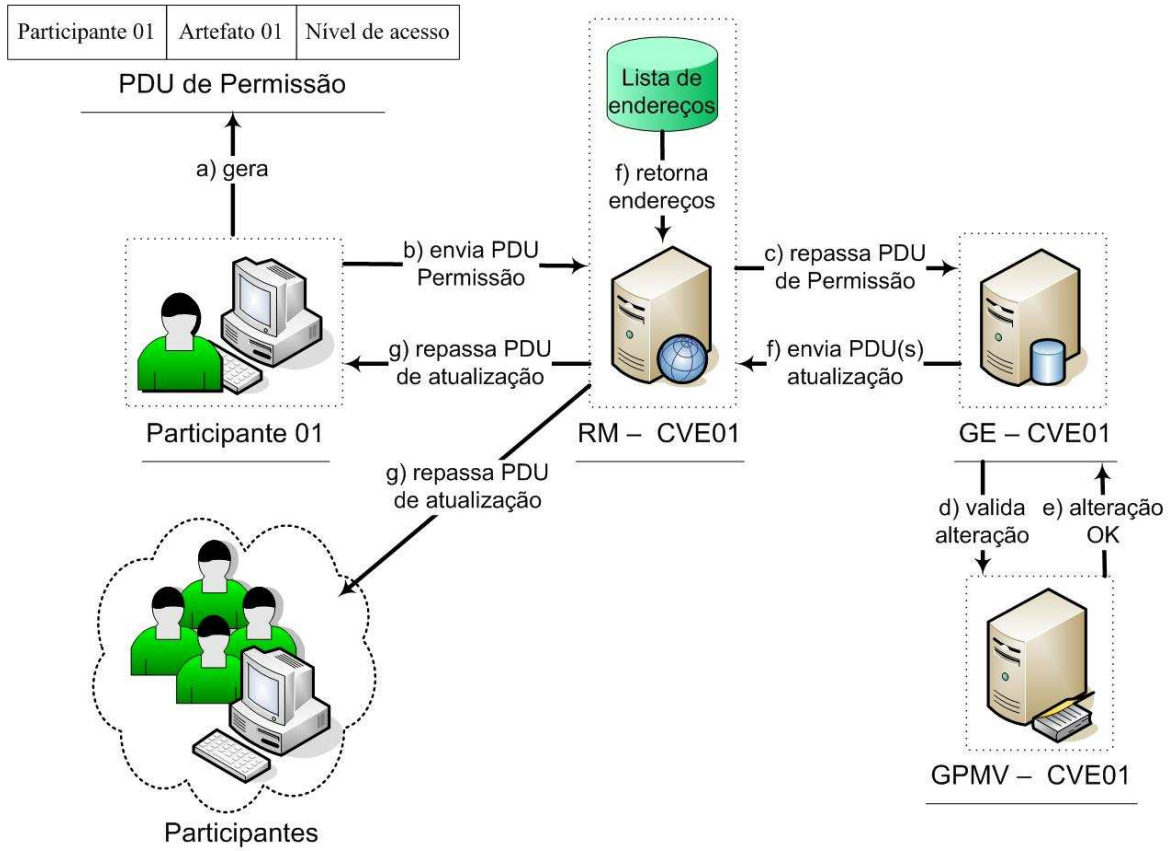


Figura 4.22. Cenário de alteração de Permissões.

Cenário 11: Publicando Mundos Virtuais

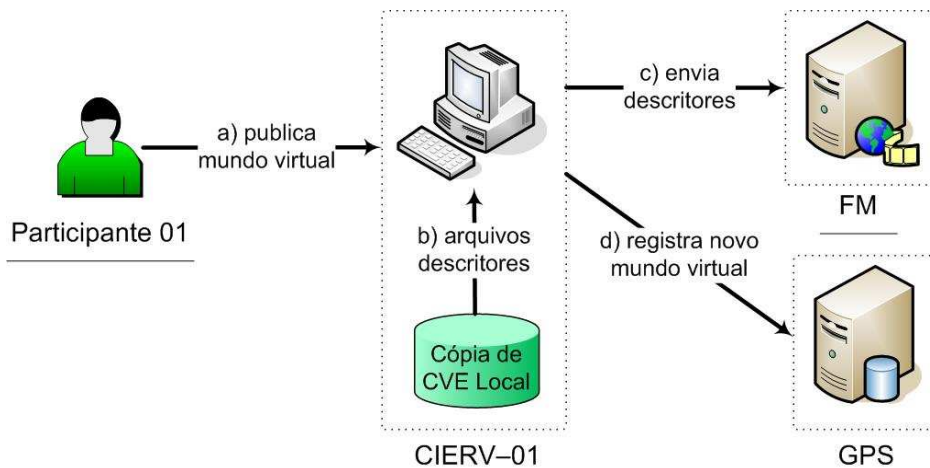


Figura 4.23. Cenário de publicação de mundos virtuais.

O processo de publicação de mundos virtuais consiste basicamente no envio dos arquivos de descrição do novo mundo virtual para o FM (Figura 4.23c). É importante ressaltar que

uma cópia de todas as mídias utilizadas por um CVE encontra-se armazenada no FM, assim, não é necessário que as mídias sejam transferidas para o FM durante a publicação. A publicação de um mundo virtual é realizada por um participante que possui permissões de publicador. Quando esse participante publica uma nova versão de um mundo virtual, ela deve ser registrada no GPS (Figura 4.23 d) para que esteja acessível a outros participantes.

Quando um participante com direitos de publicador decide publicar um mundo virtual, sua aplicação cliente inicialmente envia uma PDU de permissão para impedir alterações durante a publicação. Em seguida, a aplicação envia os arquivos ao FM e avisa ao GPS. Quando a aplicação cliente receber uma resposta positiva de que o mundo foi publicado, uma PDU de publicação é enviada aos componentes RM, GE e GPMV e aos participantes para atualização da versão do mundo virtual. Por fim, uma PDU de permissão de alterações no mundo virtual é enviada.

A publicação de mundos virtuais pode ser bastante complexa dependendo da aplicação do sistema que utiliza a CRAbCVE. Por exemplo, um sistema pode permitir que seus usuários publiquem não somente o resultado da edição de mundos virtuais, mas também o resultado da interação realizada por um grupo de participantes em um CVE (experiências não finalizadas em um laboratório de física, por exemplo), e para isso, o FM necessita de um mecanismo de controle de versões.

Cenário 12: Trabalhando Colaborativamente em CVEs

Como já foi discutido no Capítulo 2, para haver trabalho colaborativo é necessário atender aos seis processos seguintes: Comunicação, Negociação, Compartilhamento, Colaboração (Co-realização), Coordenação e Percepção. O processo de Comunicação é realizado através dos mecanismos de comunicação verbal (vide cenário 5 – “cliente enviando mensagens de texto”) e de comunicação não-verbal (vide cenário 6 – “cliente animando seu avatar”). O processo de Negociação também é realizado por esses mecanismos, já que a comunicação também intermedeia a negociação. O Compartilhamento faz parte da própria natureza dos CVEs, que são literalmente espaços de trabalho compartilhados. A Colaboração é fornecida pela edição e pela publicação de mundos virtuais (vide os cenários 9 e 11). A Coordenação é realizada através de hierarquia de grupos e divisão e atribuição de tarefas. O processo de Percepção baseia-se na própria percepção fornecida pela realidade virtual (Onde e Como), pelo conhecimento da divisão dos grupos de participantes (Quem), pelo conhecimento das tarefas sob a responsabilidade de cada participante ou do grupo (O que), e pelo histórico de tarefas (Quando).

A formação de hierarquia de grupos é possível através de uma PDU de agrupamento. Essa PDU é usada tanto para formação de grupos de participantes como também para formação de objetos compostos. Assim, quando um participante está formando uma hierarquia de grupos, o CIERV está enviando várias PDUs de agrupamento para formar essa hierarquia. Cada PDU possui uma identificação do ‘líder’ do grupo (campo pai – Figura 4.24 a) e um ‘subordinado’ (campo filho – Figura 4.24 a). Desse modo, para formar um grupo de 4 participantes são necessárias 4 PDUs de agrupamento (Figura 4.24).

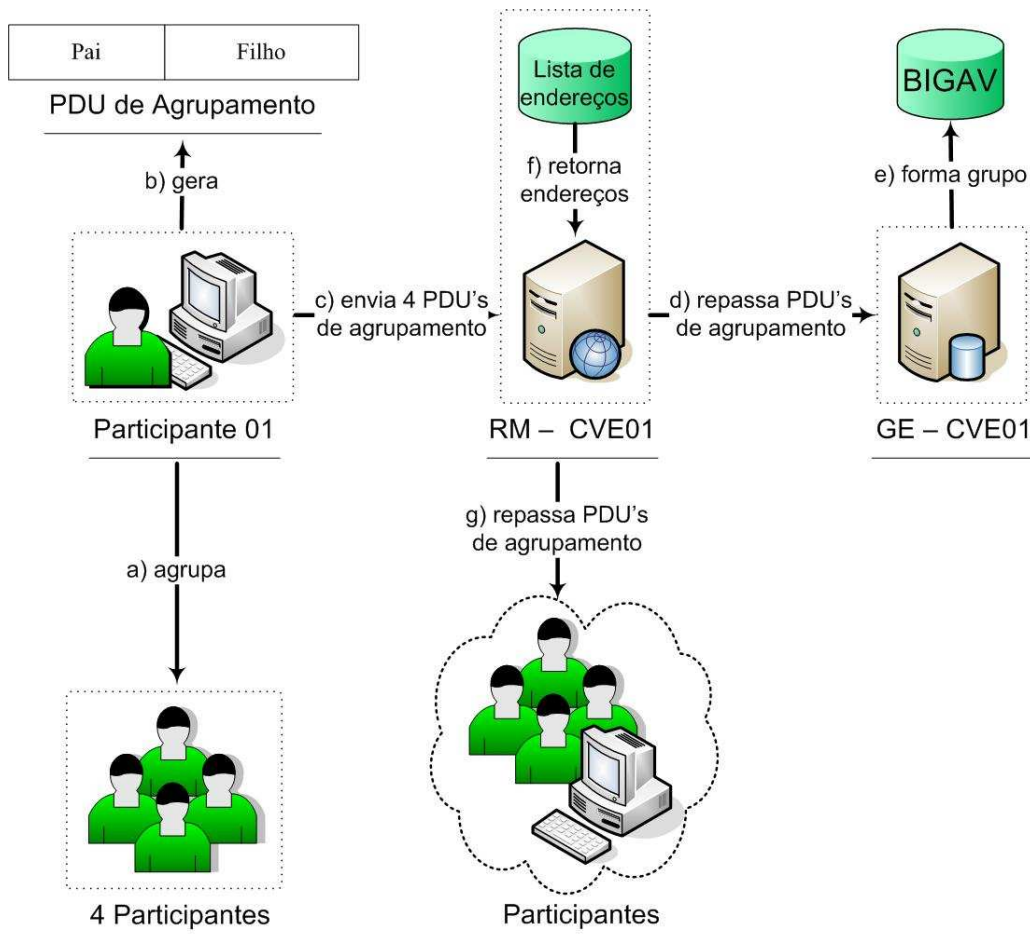


Figura 4.24. Cenário de formação de grupo.

Uma tarefa é tratada na arquitetura CRABCVE como um objeto que não possui mídia vinculada. Esse objeto contém a descrição da tarefa, suas dependências, e o participante ou grupo ao qual a tarefa foi atribuída. Assim, o processo de criação e definição de tarefas pelos participantes é semelhante ao processo de inserção de um novo objeto no ambiente, onde o CIERV envia uma PDU de inserção de objeto aos participantes e o disponibiliza no FM (semelhante ao cenário de inserção de objetos – Figura 4.20). A modificação de tarefas, por sua vez, usa uma PDU de mudança de estado. Isso é

semelhante ao que ocorre no cenário de manipulação de objetos, cujo funcionamento é idêntico ao descrito no cenário de remoção da Figura 4.21.

Cenário 13: Desinstanciando CVEs

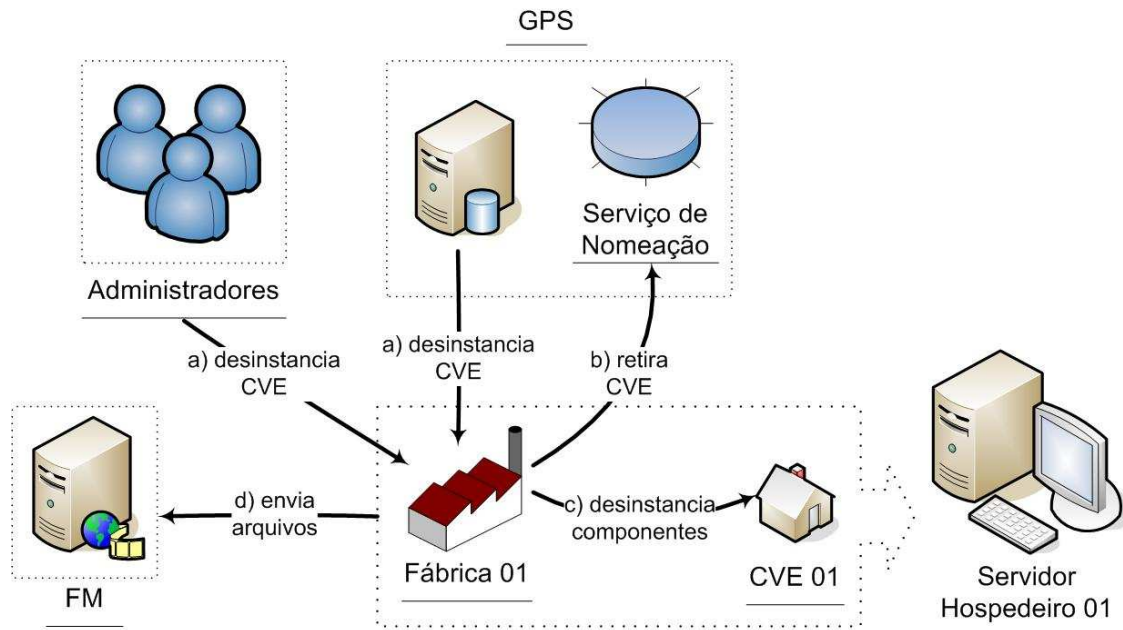


Figura 4.25. Cenário de desinstalação de CVEs.

Um CVE pode ser desinstanciado por dois motivos: tempo de inatividade prolongado ou sobrecarga do servidor hospedeiro. Se um CVE passar um determinado tempo sem receber conexões de participantes, ele pode ser automaticamente desinstanciado pela fábrica. Como já discutido, o GPS pode ter mecanismos que tratem as possíveis sobrecargas de servidores através da migração de CVEs. Assim, um CVE pode ser desinstanciado de um servidor para ser instanciado em outro servidor. Para que isso seja possível, toda vez que um CVE é desinstanciado, suas informações (configuração da BIGAV, do GPMV e do RM) são transferidas para o FM (Figura 4.25 d). Para que o usuário não perceba a migração, ele deverá primeiro conectar-se ao novo servidor e depois desconectar-se do servidor sobrecarregado.

A desinstalação de CVEs pode ser realizada tanto pela administração local do servidor onde o CVE se encontra instanciado, quanto pelo GPS (Figura 4.25 a). Em ambos os casos, o cadastro do CVE deve ser retirado do serviço de nomes do GPS (Figura 4.25 b). Para desfazer seu cadastro, o CVE executa uma chamada de um método do serviço de nomes, passando seu nome como argumento. Além disso, o CVE deverá realizar procedimentos próprios relativos à desativação de um subambiente (Figura 4.25 c).

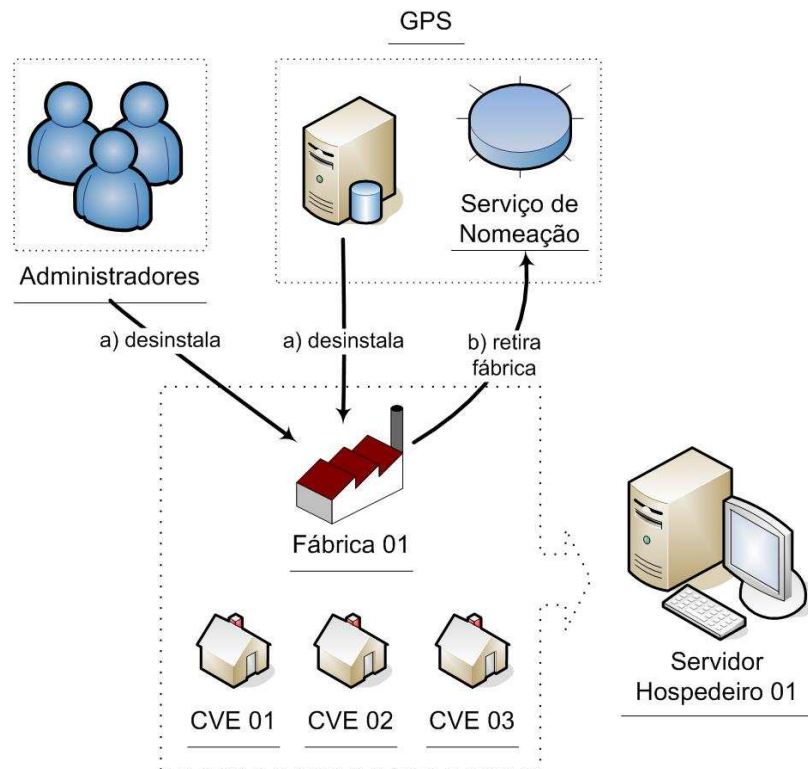
Cenário 14: Desinstalando fábricas

Figura 4.26. Cenário de desinstalação de fábricas

De forma análoga ao processo de desinstanciamento dos CVEs, a desinstalação de fábricas também pode ser realizada tanto pela administração local do servidor, quanto, remotamente, pelo GPS (Figura 4.26 a), através de um método da interface da fábrica. Em ambos os casos, a fábrica deve retirar seu cadastro do serviço de nomes do GPS (Figura 4.26 b). Geralmente, a desinstalação de uma fábrica só ocorre depois que todos os CVEs instanciados por ela tenham sido desinstanciados anteriormente. Para desfazer seu cadastro, a fábrica executa uma chamada de um método do serviço de nomes, passando seu nome como argumento.

4.5.2 Recomendações de Uso

As seguintes recomendações básicas devem ser respeitadas para que os sistemas que fazem uso da CRAbCVE ofereçam aos usuários e administradores uma boa qualidade de serviço:

- ♦ Os componentes da arquitetura devem ter identificadores únicos, a fim de auxiliar o trabalho de administração de todo o sistema;
- ♦ Os elementos presentes no CVE (estruturas tridimensionais, mídias disponíveis, participantes, etc.) devem ter identificadores únicos, para serem armazenados

nas BIGAV's de todos os GEs utilizados (a estratégia de identificação desses elementos fica a cargo da equipe de administração);

- ♦ Um mesmo cliente de administração deve ser capaz de conectar-se a mais de um componente simultaneamente;
- ♦ Um GPS pode gerenciar vários CVEs, dependendo apenas da quantidade de servidores hospedeiros disponíveis;
- ♦ Cada subambiente deve ter seu próprio RM, para possibilitar as conexões de seus vários participantes;
- ♦ Um mesmo FM pode atender a mais de um subambiente;
- ♦ Um GPS pode atender a mais de um FM, significando que os elementos cadastrados na base de dados de um GPS podem ser oriundos de mais de um Fornecedor de Mídias;
- ♦ Um GE pode atender a mais de um subambiente;
- ♦ Um CIERV deve conectar-se a apenas um FM por vez;
- ♦ Todo participante de um subambiente, incluindo simulóides, deve ser registrado no Gerenciador de Eventos utilizado por esse subambiente; e,
- ♦ Um Controlador de Simulóide ou um Cliente de Interação e Edição em Realidade Virtual não pode estar conectado simultaneamente a mais de um subambiente.

Essa última recomendação não decorre de qualquer problema de integração da arquitetura, mas sim da impossibilidade de um indivíduo do mundo real estar presente em dois locais distintos simultaneamente. Caso essa regra não seja respeitada, a sensação de presença compartilhada do CVE gerado poderá simplesmente não existir. Respeitado esse conjunto de recomendações, é possível que diversos tipos de componentes da arquitetura sejam executados em um mesmo computador simultaneamente.

Exemplos de utilização de componentes da arquitetura para tornar CVEs disponíveis são apresentados na Figura 4.27 e na Figura 4.28. Na Figura 4.27, é mostrado um CVE formado por um único subambiente e, na Figura 4.28, é mostrado um CVE formado por dois subambientes.

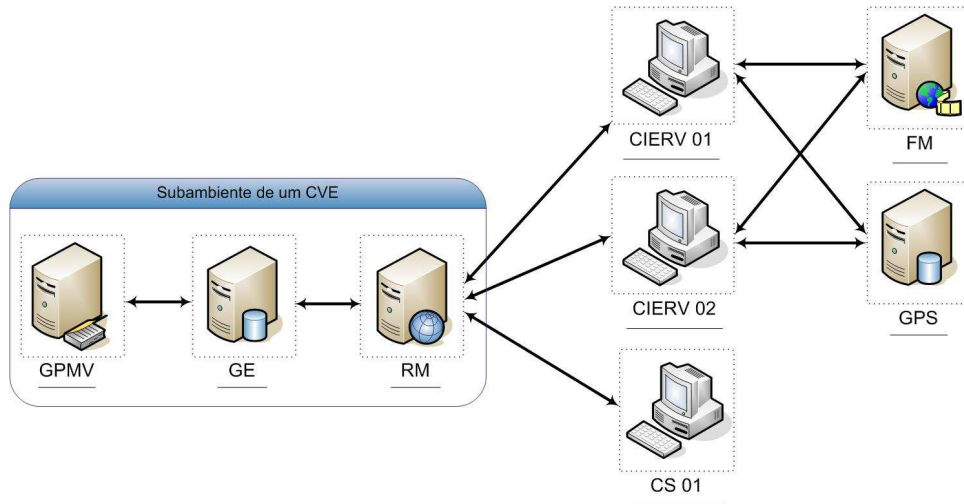


Figura 4.27. Exemplo de emprego da arquitetura em um único subambiente.

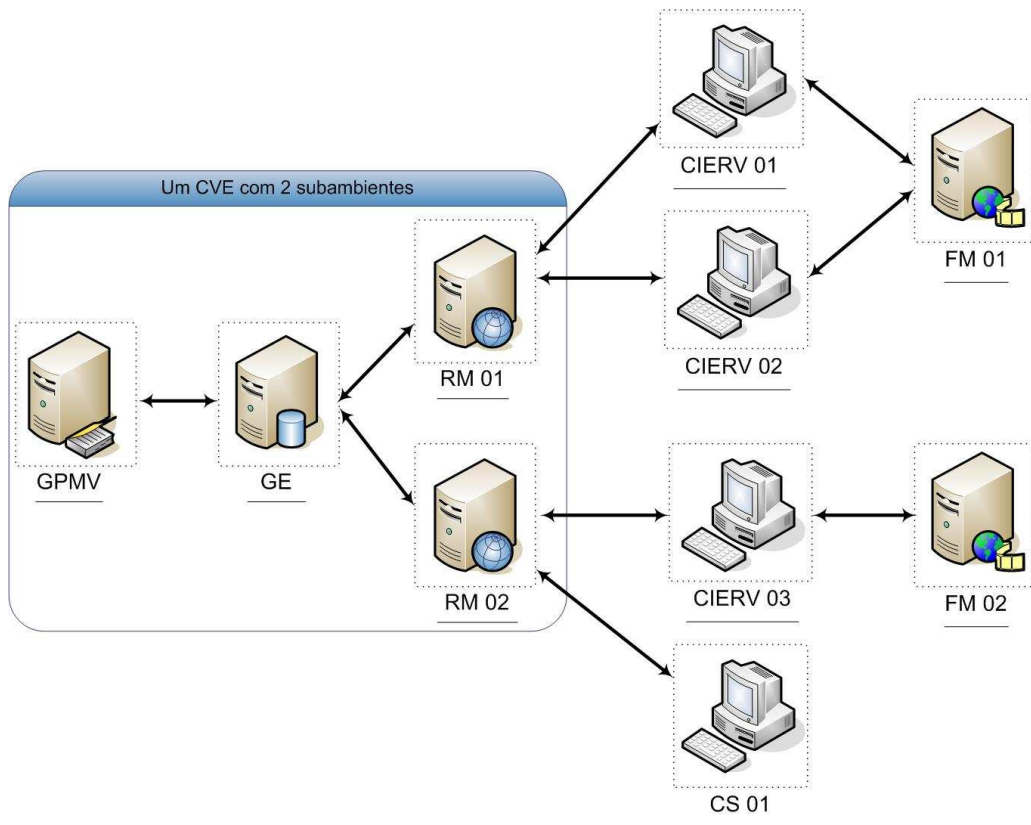


Figura 4.28. Exemplo de emprego da arquitetura com dois subambientes simultaneamente.

4.6 Protocolo PASSÁRGADa¹⁴ 2.0

O protocolo PASSÁRGADa 2.0 (Protocolo de Acesso e Suporte a Ambientes Remotos para Grupos de Ação Distribuídos) é a segunda versão do protocolo que originalmente foi proposto por Leite-Junior (2000). Essa nova reformula a primeira versão para dar suporte ao trabalho colaborativo. Assim como a primeira versão, O PASSÁRGADa 2.0 é um modelo de protocolo do tipo *game-like protocol*, desenvolvido para suportar o conjunto de necessidades de CVEs projetados a partir da arquitetura CRAbCVE. Essas necessidades foram identificadas nos cenários de utilização discutidos anteriormente.

Apesar de inicialmente proposto para sistemas baseados na arquitetura CRAbCVE, o protocolo PASSÁRGADa 2.0, devido à abrangência das PDUs por ele definidas, pode também ser empregado em outros CVEs ou mesmo em outros sistemas multiusuários distribuídos que necessitem fornecer serviços de comunicação, interação e edição a seus participantes.

O primeiro *byte* (o *byte Version*) da PDU de conexão (*Connect*) é reservado para a versão do protocolo, permitindo assim, que a arquitetura diferencie as duas versões do protocolo. O segundo *byte* (o *byte Signal*) é um sinalizador (caractere alfanumérico que identifica a PDU). O restante da PDU contém informações, armazenadas em campos específicos, referentes à ação por ela desempenhada. O protocolo especifica um total de 19 PDUs, divididas em 6 áreas de atuação distintas: conexão de clientes, instanciação de elementos, manipulação de elementos, comunicação, serviços específicos da arquitetura CRAbCVE, e integração.

4.6.1 Conexão de Clientes

As PDUs para conexão de clientes são:

Tabela 1. PDU *Connect*. Conexão de clientes em um determinado subambiente.

PDU: Connect						Tamanho: 46 bytes
Signal: C	Version	WorldID	ClientType	Ticket	BodyID	
1	1	1	1	40	2	

¹⁴ O nome do protocolo é também uma referência a Passárgada, o país maravilhoso onde tudo é possível, descrito no poema “Vou-me embora pra Passárgada”, de autoria de Manuel Bandeira.

Tabela 2. PDU *Disconnect*. Desconexão de clientes em um determinado subambiente.

PDU: Disconnect		Tamanho: 2 bytes
Signal: D	WorldID	
1	1	

Tabela 3. PDU *Reconnect*. Reconexão de clientes em subambientes.

PDU: Reconnect				Tamanho: 21 bytes
Signal: N	WorldID	IP	Port	
1	1	15	4	

Tabela 4. PDU *Accept*. Aceitação ou não de conexão de cliente.

PDU: Accept					Tamanho: 24 bytes
Signal: A	WorldID	UserName	Result	Restriction	
1	1	20	1	1	

Um mesmo computador é capaz de executar simultaneamente diversos componentes da arquitetura CRAbCVE. Assim, é necessário identificar qual componente é responsável por cada subambiente e, para que isso seja possível, emprega-se o campo *WorldID*, que identifica de forma única cada um dos subambientes executados naquele computador. As tarefas de atribuir identificações aos subambientes e de configurar os respectivos componentes ficam a cargo da fábrica no servidor hospedeiro. É importante salientar que devido ao tamanho do campo *WorldID* ser de 1 byte, cada computador pode gerenciar até 256 subambientes, desde que disponha de poder computacional suficiente.

O campo *ClientType* da PDU *Connect* simplesmente identifica o tipo de componente-cliente (Cliente de Interação e Edição em Realidade Virtual, Controlador de Simulóide ou Cliente de Administração) que está realizando a conexão no ambiente. Já o campo *BodyID* da mesma PDU contém o identificador do arquivo que representa o corpo virtual (avatar) escolhido pelo participante. Esse arquivo fica armazenado no FM, ou pode estar armazenado localmente na *cache* de mídias locais. O campo *Ticket*, também da PDU *connect*, garante que o cliente validou sua entrada no GPS, como já foi discutido antes.

Os campos *IP* e *Port* da PDU *Reconnect* informam o endereço do computador onde está sendo executado um subambiente virtual específico. Esses campos são utilizados em conjunto com o campo *WorldID* para a identificação do computador e do subambiente sendo executado no mesmo. Tal função é necessária, por exemplo, para reconexões de usuários, através da passagem pelos portais existentes no CVE.

O campo *Result* da PDU *Accept* refere-se à aceitação de conexões solicitadas pelo cliente (0 – rejeitada, 1 – retorna o *WorldID*). Já o campo *Restriction* da mesma PDU informa ao CIERV do participante se sua exploração possui algum tipo de restrição no subambiente virtual ao qual o mesmo se conectou. Essas restrições específicas podem, por exemplo, referir-se à limitação do participante a apenas navegar pelo ambiente, não podendo editar o ambiente.

4.6.2 Instanciação de Elementos

As PDUs para instanciação e desinstanciação de elementos são respectivamente:

Tabela 5. PDU *In*. Criação de elemento virtual.

PDU: In							Tamanho: 34 bytes	
Signal: I	WorldID	Flow	ElementType	ElementID	Name	BodyID	Access	
1	1	4	1	4	20	2	1	

Tabela 6. PDU *Out*. Remoção de elemento virtual.

PDU: Out					Tamanho: 11 bytes	
Signal: O	WorldID	Flow	ElementType	ElementID		
1	1	4	1	4		

O campo *Flow* identifica o sentido da PDU, ou seja, se ela foi gerada pelo CIERV para o GE, ou vice-versa. O campo *ElementID* contém o identificador do elemento virtual que está sendo instanciado. O campo *ElementType* define o tipo de elemento virtual – usuário ou objeto. Caso esse elemento virtual refira-se a um determinado avatar ou simulóide, o *ElementID* apresenta o identificador do respectivo participante do ambiente.

Já o campo *BodyID* tem seu funcionamento semelhante àquele definido na Seção 4.5.1, porém, referindo-se agora também às estruturas tridimensionais utilizadas pelos objetos do CVE. O campo *Access* indica possíveis restrições à manipulação desse elemento.

Se a PDU *In* for utilizada para informar a entrada de um novo participante no ambiente, o campo *Name* refere-se ao *login* do usuário no sistema. Entretanto, se a PDU for utilizada para inserir um objeto no ambiente, esse campo refere-se ao nome do objeto.

4.6.3 Manipulação de Elementos

Existem cinco PDUs para manipulação de elementos:

Tabela 7. PDU *Position*. Mudança de posição ou escala, com ou sem deslocamento.

PDU: Position							Tamanho 38 bytes					
Signal: P	WorldID	Msg Type	Flow	Element Type	Element ID	Element Part	Pos X	Pos Y	Pos Z	Axis X	Axis Y	Axis Z
1	1	1	4	1	4	2	4	4	4	4	4	4

Tabela 8. PDU *Move*. Movimenta ou rotaciona um elemento virtual.

PDU: Move				Tamanho: 14 bytes			
Signal: M	WorldID	Turn	ElementType	ElementID	ElementPart	Speed	
1	1	1	1	4	2	4	

Tabela 9. PDU *Group*. Agrupa um conjunto de elementos virtuais.

PDU: Group					Tamanho: 16 bytes		
Signal: G	WorldID	Flow	ElementType	ElementID	ChildID	MsgType	
1	1	4	1	4	4	1	

Tabela 10. PDU *Action*. Executa uma ação pré-definida de um elemento virtual.

PDU: Action				Tamanho: 13 bytes		
Signal: T	WorldID	Flow	ElementType	ElementID	Action	
1	1	4	1	4	2	

Tabela 11. PDU *Status*. Modifica o estado de um elemento virtual.

PDU: Status					Tamanho: 15 bytes	
Signal: S	WorldID	Flow	ElementType	ElementID	StatusID	StatusValue
1	1	4	1	4	2	2

Tabela 12. PDU *Permission*. Modifica nível de permissão de um elemento virtual.

PDU: Permission					Tamanho: 15 bytes	
Signal: R	WorldID	Flow	ElementType	ElementID	PermissionType	PermissionValue
1	1	4	1	4	2	2

Tabela 13. PDU *Publish*. Modifica a versão de um mundo virtual.

PDU: Publish			Tamanho: 7 bytes
Signal: B	WorldID	NewWorldID	Flow
1	1	1	4

O campo *ElementPart* possibilita a identificação de uma parte específica do objeto a ser movimentada. Esse campo permite a movimentação de articulações específicas da estrutura tridimensional de um determinado elemento virtual; no entanto, seu uso deve ser comedido, pois a realização de movimentações muito complexas exige o envio de inúmeras PDUs à rede. Geralmente, deve-se optar pela utilização de ações pré-definidas, que são disparadas por uma única PDU. Já os campos *ElementType*, *ElementID* e *Flow* têm seu funcionamento semelhante àquele definido na seção anterior.

A PDU *Group* possibilita o trabalho com elementos compostos, essenciais em sistemas de autoria de mundos virtuais. Um elemento composto é definido por um elemento ‘pai’ (campo *ElementID*) e por seus componentes filhos (campo *ChildID*). Assim, para criar um elemento composto, uma PDU é enviada a cada um de seus componentes. O campo *MsgType* da PDU *Group* indica se a PDU foi criada para agrupar (*MsgType* = 1) ou desagrupar (*MsgType* = 0), nesse caso basta referenciar o pai do elemento.

Os campos *PosX*, *PosY* e *PosZ* da PDU *Position* identificam o centro da nova posição tridimensional a ser ocupada pelo elemento virtual no subambiente, e os campos *AxisX*, *AxisY* e *AxisZ* informam a orientação do elemento nos três eixos. Para economizar tipos de PDUs, a PDU *Position* também é utilizada para aplicar escala a elementos, através dos campos *PosX*, *PosY* e *PosZ* que são utilizados para informar os fatores de escala a serem aplicados nas direções X, Y e Z, respectivamente. A diferenciação do seu funcionamento é feita pelo campo *MsgType* (*MsgType* = 0 indica posicionamento; *MsgType* = 1 indica escala).

A ação de movimentação e rotação de elementos virtuais é realizada pela PDU *Move* que repassa apenas a velocidade (campo *Speed*) com que a ação deve ser executada. Por exemplo, quando um participante inicia um movimento em linha reta, uma PDU com a velocidade do movimento é enviada. Conseqüentemente, quando os outros participantes recebem essa PDU, suas aplicações começam a realizar a movimentação do avatar do participante segundo sua orientação. Assim, quando o participante finaliza o movimento, uma PDU de *Position* contendo informações reais de sua posição e de sua orientação é enviada aos outros participantes, a fim de que eles finalizem a movimentação do avatar e

corrijam sua localização e orientação localmente. A rotação é realizada de forma semelhante, exceto pelo campo *Turn* que informa o sentido da rotação (*Turn* = 0 indica rotação no sentido horário, *Turn* = 1 indica rotação no sentido anti-horário).

O campo *Action* da PDU *Action* indica qual ação deve ser executada, dentro de um conjunto pré-definido de possibilidades, por um determinado elemento virtual. A PDU *Status* atualiza estados de elementos virtuais, informando o estado a ser atualizado (campo *StatusID*) e o valor (campo *StatusValue*). A PDU *Permission* é semelhante à PDU *Status*, informando o tipo de permissão (campo *PermissionType*) e o nível (campo *PermissionValue*).

Por último, a PDU *Publish* é utilizada em conjunto com a PDU *Permission* para a publicação de mundos virtuais, como foi mencionado no Cenário 11: Publicando Mundos Virtuais. Ela informa aos participantes e aos componentes do CVE (RM, GE, GPMV) o novo identificador do mundo (campo *NewWorldId*).

4.6.4 Comunicação

As PDUs para comunicação de clientes são:

Tabela 14. PDU *Show*. Apresentação de Mídias.

PDU: Show						Tamanho: 11 – 251 bytes
Signal: H	WorldID	Origin	Destiny	Type	Msg	
1	1	4	4	1	0 – 240	

Tabela 15. PDU *What*. Informações sobre elemento virtual ou componente da arquitetura.

PDU: What						(Continua)
Signal: W	WorldID	ComponentType	ComponentID	ElementID	Name	BodyID
1	1	1	4	4	20	1

PDU: What (Continuação)						Tamanho: 56 – 246 bytes
PosX	PosY	PosZ	AngleX	AngleY	AngleZ	Description
4	4	4	4	4	4	0 – 190

O campo *Origin* da PDU *Show* contém o identificador do participante que originou a informação transmitida pela PDU. O campo *Type* da mesma PDU identifica o tipo de mídia (por exemplo, texto, som, vídeo, etc.), o que permite a utilização de qualquer mídia armazenada em qualquer tipo de arquivo. A transferência de mensagens de texto (por exemplo, em bate-papo) é realizada pela PDU *Show*.

Se a PDU *What* for usada para um componente da arquitetura, os campos *ComponentType* e *ComponentId* retornaram o tipo e o identificador do componente, respectivamente. Se for usada para um elemento virtual, os campos *ElementID*, *Name*, e *BodyID* retornam, respectivamente, o identificador do elemento, o nome e o identificador do avatar (se for um personagem virtual). O campo *Description* contém informações sobre um determinado elemento virtual do CVE (estrutura tridimensional, participante, etc.). Essas informações são extraídas diretamente da BIGAV do GE utilizado no subambiente que contém o elemento virtual. Já os campos *PosX*, *PosY* e *PosZ* identificam a posição do elemento, e *AngleX*, *AngleY* e *AngleZ* identificam sua orientação dentro do ambiente.

4.6.5 Serviços Específicos da Arquitetura CRAbCVE

As PDUs dos serviços de administração e configuração dos componentes da arquitetura são:

Tabela 16. PDU *Administrate*. Administração de componentes da arquitetura.

PDU: <i>Administrate</i>						Tamanho: 12 – 251 bytes
Signal: E	WorldID	Flow	ComponentType	ComponentID	OptionList	
1	1	4	1	4	1 – 240	

Tabela 17. PDU *Log*. Documentação de ocorrência.

PDU: <i>Log</i>						Tamanho: 9 – 249 bytes
Signal: L	WorldID	ComponentType	ComponentID	Event	Details	
1	1	1	4	2	0 – 240	

Tabela 18. PDU *Alive*. Checagem de estado de componente da arquitetura.

PDU: <i>Alive</i>					Tamanho: 8 bytes
Signal: V	WorldID	ComponentType	ComponentID	ComponentStatus	
1	1	1	4	1	

Os campos *ComponentType* e *ComponentId* são de interesse somente da equipe de administração e informam, respectivamente, o tipo e o identificador de um componente específico da arquitetura CRAbCVE na rede.

O campo *Flow* da PDU *Administrate* identifica se ela é uma ordem para um determinado componente da arquitetura ou um retorno para o Cliente de Administração. O conteúdo do campo *OptionsList* deve ser definido pelos responsáveis pela implementação do subambiente virtual, de acordo com as necessidades específicas da equipe de administração do CVE. O campo *Event* da PDU *Log* identifica o tipo de evento e o campo

Details contém a descrição da ocorrência. Já o campo *ComponentStatus* identifica se o componente está funcionando corretamente.

4.6.6 Integração

Tabela 19. PDU *Generic*. Integração com outros sistemas/ferramentas.

PDU: <i>Generic</i>					Tamanho: 11 – 251 bytes	
Signal: X	WorldID	Origin	Destiny	Type	Content	
1	1	4	4	1	0 – 240	

O campo *Content* deve ser definido de acordo com as necessidades específicas das aplicações a serem integradas ao CVE. Se o valor do campo *Destiny* for nulo, a PDU é encaminhada a todos os participantes.

Como o protocolo PASSÁRGADa 2.0 fornece suporte a todas as funcionalidades da arquitetura CRAbCVE, esse modelo deve ser empregado como base para a geração de protocolos específicos, para serem utilizados em implementações de CVEs que façam uso dessa arquitetura.

4.7 Considerações Finais

A realização de simulações de fenômenos do mundo real, exploradas por grupos de usuários de forma simultânea, mostra-se como uma ferramenta bastante eficiente para a compreensão de fatos e posterior discussão de idéias. Assim, a construção de ambientes virtuais colaborativos, tornados disponíveis através da Internet e atendendo a diferentes categorias de participantes, com diferentes necessidades é uma alternativa aos atuais padrões de trabalho colaborativo.

Neste Capítulo, foi apresentada a arquitetura CRAbCVE, que permite o uso de componentes especializados, os quais podem ser executados simultaneamente em diversas máquinas. Essa arquitetura foi concebida especificamente para viabilizar ambientes de realidade virtual em microcomputadores comuns conectados à Internet. Essa é uma alternativa de baixo custo aos atuais sistemas de suporte a ambientes virtuais colaborativos, e permite a criação de uma estrutura distribuída, com uma grande capacidade de expansão. Ela oferece, assim, novas possibilidades para o trabalho em grupo.

A união de técnicas oriundas de outras áreas, que não a realidade virtual, como o emprego de *caches* locais, da replicação sob demanda, da instanciação automática de CVEs através de fábricas e de estruturas utilizadas originalmente no gerenciamento de redes de computadores, mostra-se como um diferencial importante da arquitetura

CRAbCVE. Isso permite a minimização do uso da rede para a transferência dos dados necessários ao ambiente virtual compartilhado.

A Tabela 20 reúne e resume as soluções adotadas pela arquitetura CRAbCVE para atender os requisitos de software e hardware que devem ser considerados no desenvolvimento de CVEs, conforme identificados no Capítulo 3. Como foi visto no Capítulo 2, muitos desses requisitos são essenciais para CVEs, outros são apenas desejáveis, e a diferenciação de alguns requisitos serem desejáveis ou serem necessários depende da aplicação do CVE.

Tabela 20. Soluções adotadas pela arquitetura CRAbCVE para atender os requisitos identificados no Capítulo 3.

Requisito	Solução adotada pela arquitetura CRAbCVE
Fidelidade.	A fidelidade do realismo visual, sensorial e físico; e a fidelidade da modelagem dos elementos virtuais e dos dispositivos de entrada dependem do motor gráfico utilizado pelo sistema que faz uso da arquitetura CRAbCVE. Entretanto, a fidelidade de tempo (atraso mínimo), de informação (Consistência dos dados) é bem suportada pela arquitetura.
Interação.	A arquitetura permite três tipos de interação: interação entre os participantes, interação dos participantes com os objetos e interação dos participantes com o próprio ambiente virtual (edição de mundos virtuais). A CRAbCVE possui dois servidores responsáveis por tratar as interações ocorridas dentro do CVE, o Gerenciador de Eventos e o Gerenciador de Políticas do Mundo Virtual.
Comunicação verbal.	não Através das animações dos avatares, usuários podem comunicar-se por meio de gestos e posturas. Entretanto, essa funcionalidade depende da forma como essas animações são implementadas no motor gráfico utilizado.
Suporte ao trabalho colaborativo.	Para haver trabalho em colaboração, são necessários: Comunicação, Negociação, Compartilhamento, Colaboração (Co-realização), Coordenação e Percepção. A Comunicação e a Negociação são realizadas através dos mecanismos de comunicação verbal e não verbal. O Compartilhamento faz parte

Requisito	Solução adotada pela arquitetura CRAbCVE
	da própria natureza dos CVEs. A Colaboração é fornecida pela edição e pela publicação de mundos virtuais. A Coordenação baseia-se em hierarquia de grupos, e em divisão e atribuição de tarefas. O processo de Percepção baseia-se na própria percepção fornecida pela RV (Onde e Como), pelos grupos de participantes (Quem), pelas tarefas sob responsabilidade de cada participante ou do grupo (O que), e pelo histórico de tarefas (Quando).
Escalabilidade.	Por ser uma arquitetura distribuída, a CRAbCVE pode disponibilizar uma grande quantidade de CVEs dependendo apenas da quantidade de servidores disponíveis. Além disso, a técnicas de portais torna possível a disponibilidades de CVEs compostos por vários subambientes.
Adequação a largura de banda disponível.	A arquitetura CRAbCVE foi projetada para utilizar o mínimo possível de largura de banda com o objetivo de funcionar através da Internet. Para isso, ela se utiliza de: partição de seus CVEs através de portais; otimização do protocolo de comunicação com a utilização de PDUs simples; restrição ao uso de recursos de áudio e vídeo (a transferência desses recursos consomem muita largura de banda); predição de movimento através do uso da técnica de <i>dead-reckoning</i> ; e técnicas de <i>caches</i> locais para evitar consultas à rede e transferência de arquivos.
Redução do atraso de rede.	Para resolver o problema do atraso de rede a arquitetura faz uso de um protocolo com PDUs bastantes pequenas com tamanho de 500 bytes (tamanho mínimo de um pacote TCP/IP). As PDUs mais utilizadas (Instanciação e manipulação de elementos) usam menos de 40 bytes.
Modelo de comunicação.	A CRAbCVE é considerada uma arquitetura híbrida, pois se utiliza do modelo centralizado e do modelo distribuído. Além disso, a CRAbCVE faz uso do serviço de nomes do modelo de Computação Distribuída Orientada a Objetos.

Requisito	Solução adotada pela arquitetura CRAbCVE
Modelo de armazenamento.	O modelo de armazenamento utilizado pela CRAbCVE é o de Banco de Dados Replicados, no qual cada Cliente conectado ao sistema possui uma réplica completa dos dados do CVE (modelos geométricos, texturas, sons, objetos etc.) sendo que a consistência de todos os ambientes é alcançada com a utilização de um servidor, o Gerenciador de Eventos.
Protocolo	O protocolo utilizado é o PASSÁRGADA, um protocolo do tipo <i>Game-like protocol</i> originalmente desenvolvido para dar suporte às funcionalidades da CRAbCVE.

Capítulo 5

MAC – Modelo de Autoria em CVEs

5.1 Introdução

Neste capítulo, é apresentado o Modelo de Autoria em CVEs (MAC), especificado para auxiliar o emprego da arquitetura CRAbCVE no trabalho colaborativo. O MAC consiste de uma especificação de ferramenta de autoria acoplada ao cliente de interação (CIERV) e de uma descrição dos mecanismos empregados para dar suporte ao trabalho colaborativo.

O MAC pode ser aplicado tanto para interação síncrona quanto assíncrona, e visa auxiliar o usuário na construção de um ambiente virtual, ou seja: na criação do mundo virtual; na disposição e manipulação de elementos estáticos e dinâmicos no mundo; na definição e atribuição de comportamentos a objetos dinâmicos; e na publicação do ambiente virtual.

A ferramenta de autoria especificada pelo MAC é integrada à CRAbCVE para possibilitar a realização do trabalho em grupo, independentemente da localização dos integrantes do grupo no mundo real. Cada integrante possui uma cópia do mundo virtual em sua própria máquina, e quaisquer alterações realizadas localmente através da ferramenta de autoria são distribuídas aos demais membros do grupo, por intermédio dos servidores da arquitetura CRAbCVE, para atualização automática das cópias do mundo residentes em suas máquinas.

Para que atividades colaborativas, especialmente as atividades de autoria de mundos virtuais, sejam realizadas com êxito, o MAC emprega mecanismos de coordenação, percepção e comunicação. Os mecanismos de coordenação, como já mencionado, permitem a organização do grupo para evitar a perda de esforços de comunicação e de cooperação, e para assegurar que as tarefas sejam realizadas na ordem e no tempo corretos e cumpram as restrições e os objetivos pré-especificados (Raposo *et al.*, 2001). Os mecanismos de percepção, segundo Gutwin e Greenberg (1999), permitem que o indivíduo tome ciência de aspectos da atividade cooperativa tais como: o objetivo comum, o papel de cada membro do grupo dentro do contexto, a tarefa que deve realizar e a maneira como deve realizá-la, o impacto das ações executadas, os limites de sua atuação, a identificação

dos membros que se encontram em sua proximidade, etc. Os mecanismos de comunicação, por sua vez, são indispensáveis ao processo de colaboração.

A seguir, discutem-se aspectos relacionados à construção de ambientes virtuais; na Seção 5.3, apresenta-se a ferramenta de autoria e discutem-se seus recursos; na Seção 5.4, descrevem-se as políticas de acesso utilizadas pelo MAC; nas seções 5.5, 5.6 e 5.7, analisam-se os mecanismos de coordenação, percepção e comunicação, respectivamente; e por último, na Seção 5.8, tecem-se algumas considerações finais.

5.2 Construção de Ambientes Virtuais

As etapas de construção de um ambiente virtual – criação do mundo virtual, disposição dos elementos no mundo, configuração dos personagens virtuais e dos objetos interativos, e atribuição de comportamento a esses objetos interativos – não são triviais. Por conseguinte, sistemas robustos que auxiliem o autor nesse processo de criação são indispensáveis. A maioria dos sistemas existentes é voltada para a criação apenas das partes estáticas dos ambientes virtuais, deixando a desejar quanto aos aspectos de manipulação e configuração dos elementos dinâmicos.

A ferramenta de autoria especificada neste capítulo (Seção 5.3) propõe uma solução para alguns dos problemas mais comuns encontrados por um autor de mundos virtuais: o posicionamento dos elementos, nascedouros e portais entre ambientes virtuais; a definição de hierarquias de agrupamento de elementos; a definição e atribuição de comportamento aos elementos dinâmicos; e a compatibilidade de escala entre o mundo virtual construído e os elementos nele incluídos a posteriori.

Posicionamento. O posicionamento dos elementos estáticos e dinâmicos é uma tarefa que também requer o auxílio de ferramentas que facilitem o trabalho de construção de ambientes virtuais. A posição de um elemento pode ser definida de forma fixa, através da simples atribuição de uma localização; ou através de nascedouros, que são posições no mundo virtual onde objetos ou personagens virtuais podem aparecer. Uma vez definido um nascedouro e configurados os tipos de elementos que podem aparecer naquela posição, o elemento só será posicionado quando da utilização do sistema. Isso permite, por exemplo, que se possa definir mais de um nascedouro para um mesmo elemento e deixar a cargo do sistema do CVE decidir em qual deles esse elemento irá aparecer quando o mundo for utilizado. A definição da localização dos portais, elementos por meio dos quais os usuários são transportados a outros ambientes virtuais, também está relacionada ao problema de assistência ao posicionamento.

Hierarquias de agrupamento. Elementos podem estar vinculados entre si, formando agrupamentos hierárquicos. Em geral, alterações feitas ao elemento mestre são propagadas aos elementos vinculados. No entanto, é possível configurar alterações que, mantendo a estrutura do agrupamento, afetem elementos individuais do mesmo.

Definição e atribuição de comportamento. Alguns elementos dinâmicos possuem comportamento ao qual são associadas ações. Portanto, há necessidade de uma ferramenta de auxílio tanto para definir os comportamentos e suas ações, quanto para atribuí-los aos elementos.

Compatibilidade de Escala. Em virtude de alguns elementos (os dinâmicos, por exemplo) serem criados separadamente dos mundos virtuais onde serão inseridos, existe a necessidade de se manter a correta proporção entre eles. Para isso, na fase de edição, deve-se avaliar o tamanho dos elementos em relação a cada mundo virtual no qual serão inseridos.

5.3 Ferramenta de Autoria

Segundo Valle-Filho e seus co-autores (2000), ferramentas de autoria são softwares que devem ser operados com facilidade, sem necessitar de profundos conhecimentos de programação, para autores criarem seus próprios conteúdos. A ferramenta de autoria especificada no MAC desempenha o papel de uma interface entre os autores para realização do trabalho colaborativo. Essa ferramenta acoplada ao CIERV visa auxiliar o autor nas etapas de construção de um ambiente virtual. Para isso, ela dispõe de biblioteca de elementos e recursos de atribuição de comportamento, de controle de objetos e de publicação de arquivos.

Bibliotecas de elementos. O uso de bibliotecas facilita a criação do mundo virtual, por oferecer ao autor um conjunto de elementos pré-elaborados organizados em cinco categorias: modelos de cenários, objetos estáticos e dinâmicos, materiais, sons, e personagens virtuais controlados por computador;

Recurso de atribuição de comportamento. Esse recurso permite que o autor atribua um comportamento a um objeto dinâmico e defina as ações associadas a cada comportamento.

Recurso de controle de objetos. Esse recurso permite que o autor organize os objetos estáticos e dinâmicos no mundo virtual, utilizando-se das funcionalidades de movimentação, escala, alinhamento, rotação, agrupamento de objetos, entre outras. Esse recurso também permite configurar portais e nascedouros.

Recurso de publicação de arquivos. Essencial para publicação do ambiente virtual, arquivos são gerados com a especificação dos elementos componentes do mundo virtual e de seus respectivos estados.

Para que o maior número possível de usuários seja capaz de criar e publicar seus mundos virtuais, a especificação da ferramenta deve levar em conta aspectos de ergonomia e de usabilidade. Assim, a utilização dos recursos disponíveis deve ocorrer de uma maneira intuitiva através de uma interface amigável, requerendo um conhecimento mínimo sobre programação e computação gráfica. Além disso, é interessante que a ferramenta disponibilize documentação de ajuda, auxiliando o autor a utilizar suas principais funcionalidades (Andrade, 1999).

5.3.1 Bibliotecas

A utilização de bibliotecas aumenta a produtividade e evita perda de esforços. Elas disponibilizam um conjunto de elementos já elaborados, que podem, de maneira fácil e rápida, ser utilizados para construir um novo mundo virtual.

Conforme mencionado anteriormente, todos os elementos inseridos no mundo virtual possuem uma cópia no FM, o que os torna acessíveis a todos os outros participantes. O autor pode adicionar elementos construídos com outras ferramentas às bibliotecas existentes, porém eles não estarão disponíveis para os outros participantes enquanto não forem inseridos em algum ambiente em construção (Seção 4.5.1 Cenários de Utilização – Cenário 9: Editando Mundos Virtuais). No momento em que um novo elemento é inserido em um mundo virtual, ele é enviado automaticamente ao FM para que os outros participantes possam acessá-lo, e para ser inserido nas bibliotecas dos outros participantes.

A adição de novos elementos à biblioteca é trivial. Porém, as funcionalidades para construção de elementos são bastante limitadas, principalmente, quando se tratam de personagens virtuais controlados por computador. Os elementos estão organizados nos seguintes cinco tipos de bibliotecas: Biblioteca de cenários, Biblioteca de objetos, Biblioteca de materiais, Biblioteca de sons, e Biblioteca de personagens virtuais controlados por computador.

Biblioteca de cenários. Nessa biblioteca, estão disponíveis os modelos pré-elaborados que podem ser utilizados na construção de um novo ambiente virtual. É possível inserir nessa biblioteca um modelo criado através de outra ferramenta, desde que ele esteja em um formato aceito.

Biblioteca de objetos. Nessa biblioteca, estão disponíveis os objetos estáticos (paredes, por exemplo) e dinâmicos (portas que se abrem e fecham ao comando do usuário, por exemplo) que o autor pode incluir em um mundo virtual construído a priori. Assim como na biblioteca de cenários, o autor também pode adicionar novos objetos criados com o auxílio de outras ferramentas, respeitados os formatos suportados.

Biblioteca de materiais. Nessa biblioteca, encontra-se uma coletânea de materiais que o autor pode aplicar aos objetos inseridos no mundo virtual. Cada objeto possui seu próprio conjunto de materiais, por exemplo, um personagem negro recebe um material de pele negra e uma série de materiais para representar seu estilo de cabelo e sua vestimenta. É importante destacar que os materiais devem ser elaborados previamente para um objeto específico a fim de evitar problemas de mapeamento.

Biblioteca de sons. Nessa biblioteca, estão armazenados os elementos sonoros, dos quais o autor pode dispor para aumentar o realismo dos ambientes virtuais. O autor pode adicionar sons ininterruptos, tais como o emitido pelo compressor de uma geladeira ou por um ventilador; ou associar sons a ações de objetos como, por exemplo, o som de uma porta batendo ou se abrindo.

Biblioteca de personagens virtuais controlados por computador. Nessa biblioteca, estão armazenados os personagens controlados por computador, chamados de simulóides (Csordas, 2000), que devem ser inseridos em um ambiente virtual sempre que o autor quiser aumentar a interação ou disponibilizar o acesso a recursos externos ao ambiente virtual. Por exemplo, ao criar um ambiente virtual de uma escola, o autor poderá inserir um simulóide bibliotecário que interligue serviços de dicionários eletrônicos e enciclopédias; já em uma sala virtual de matemática, poderá inserir um simulóide que se comunique com o software *Mathematica* (software para computação simbólica) e auxilie os alunos da sala (Vidal *et al.*, 2000; Santos, 2001).

A biblioteca de personagens virtuais controlados por computador é a única que não permite que o autor adicione diretamente elementos a ela. Um simulóide é uma aplicação específica que precisa ser desenvolvida a priori. Assim, a adição de simulóides a essa biblioteca só é possível depois que o simulóide for desenvolvido e que um componente específico da arquitetura CRAbCVE – o Controlador de Simulóide (CS) – for instanciado para cada simulóide. A instanciação do CS pode ser realizada tanto de forma automática através de fábricas, quanto por intermédio dos administradores. Quando um simulóide é instanciado, ele fica automaticamente disponível nas bibliotecas dos autores. Por outro lado, quando não estiver conectado ao sistema, ele é retirado da biblioteca.

5.3.2 Comportamentos

O comportamento é definido como um conjunto de ações que um objeto executa quando exposto a um determinado estímulo, podendo, ou não, mudar seu estado. Atribuição de comportamentos a objetos permite uma interatividade maior do usuário com o ambiente virtual. Na ferramenta especificada no MAC, somente os objetos dinâmicos possuem comportamentos. Por exemplo, quando um usuário, através de uma interação com o objeto dinâmico ‘porta’, provocar o estímulo para que ele se abra, o objeto executará a ação ‘abrir’ e seu estado será alterado para ‘porta aberta’.

Segundo o número de reações, os comportamentos podem ser classificados como simples ou complexos. Simples são comportamentos onde só existe uma única reação do objeto a um dado estímulo. Por exemplo, abrir uma porta que está fechada, ou fechar uma que está aberta. Complexos são comportamentos onde, a um dado estímulo, corresponde uma seqüência de reações. Por exemplo, trancar uma porta envolve a ação de fechar e travar a porta.

Segundo o tipo de estímulo, os comportamentos podem ser classificados como: reativo, proativo e simulado. Quando o objeto reage a um estímulo causado por um fator externo, o comportamento é denominado reativo. Por exemplo, uma porta que se abre à aproximação de um avatar. Quando o estímulo é agendado, o comportamento associado é denominado proativo. Por exemplo, o ponteiro de minuto de um relógio de parede pode ser agendado para mover-se a cada minuto. Quando o estímulo resulta de uma simulação, o comportamento associado é denominado simulado. Por exemplo, um simulóide professor de matemática executa a ação de indicar o quadro quando um gráfico é gerado pelo software *Mathematica* (Santos *et al.*, 2001).

Dependendo da complexidade do comportamento de um objeto, um simulóide pode ser instanciado exclusivamente para simular o comportamento desse objeto. Por exemplo, um objeto máquina de refrigerantes apresenta um comportamento complexo, pois o objeto precisa verificar a quantidade de dinheiro inserido na máquina, devolver troco se a quantidade exceder o preço do refrigerante, verificar se o refrigerante escolhido existe no estoque, e por fim, liberar o refrigerante. Nesse caso, como simulóides não são desenvolvidos pelos autores, não é possível inserir comportamentos complexos que envolvam processamento através da ferramenta de autoria.

5.3.3 Controle de Objetos

Os recursos de controle de objetos oferecidos pela ferramenta de autoria do MAC permitem que o autor posicione e manipule os objetos inseridos no cenário de forma eficiente, minimizando assim o tempo de construção de um mundo virtual. Esses recursos também facilitam o trabalho de configuração do ambiente, liberando o autor da necessidade de recorrer a uma linguagem de programação, e estão divididos em quatro categorias: Manipulação de objetos, Agrupamento, Alinhamento e Configuração de portais e nascedouros.

Manipulação de objetos. Nessa categoria, estão, por exemplo, os recursos para mover (translação) e girar (rotação), com os quais o autor pode posicionar os objetos no mundo virtual, e o recurso de escala que permite alterar as proporções dos objetos em relação ao ambiente, entre outros.

Agrupamento. Essa categoria reúne os recursos que permitem o autor agrupar objetos, formando objetos compostos, que posteriormente podem ser adicionados à biblioteca. Os objetos compostos podem ser formados por objetos simples ou por outros objetos compostos, criando-se, assim, uma hierarquia. Dessa forma, quando o autor altera um objeto composto, também altera os objetos a ele agregados.

Alinhamento. Nessa categoria, estão, por exemplo, os recursos que evitam a sobreposição de objetos sólidos (detecção de colisão), e os recursos de atração ('ímã'), entre outros. Esses recursos permitem um posicionamento rápido e simples de objetos no cenário.

Configuração de portais e nascedouros. Nessa categoria, estão os recursos que auxiliam o autor a definir e a posicionar portais e nascedouros no ambiente virtual.

5.3.4 Publicação de Arquivos

O mundo virtual construído através da ferramenta de autoria é armazenado em múltiplos arquivos, cuja organização é apresentada no diagrama mostrado na Figura 5.1. Quando um editor publica um ambiente virtual, os arquivos de descrição da composição do mundo virtual são enviados automaticamente ao FM, e o registro desse novo ambiente virtual é realizado no GPS, para que outros usuários possam explorá-lo.

Quando um CVE é instanciado por um publicador, ele encontra-se no modo de edição e não estará disponível para ser navegado por outros usuários enquanto não for publicada pelo menos uma versão inicial desse CVE para navegação. No ato da publicação, o publicador poderá escolher entre publicar apenas para salvar o atual estado do ambiente ou publicar uma versão para navegação. Adicionalmente, se o CVE tiver sido

instanciado para modificar um ambiente existente, o publicador ainda tem a opção de disponibilizar uma nova versão ou substituir a anterior.

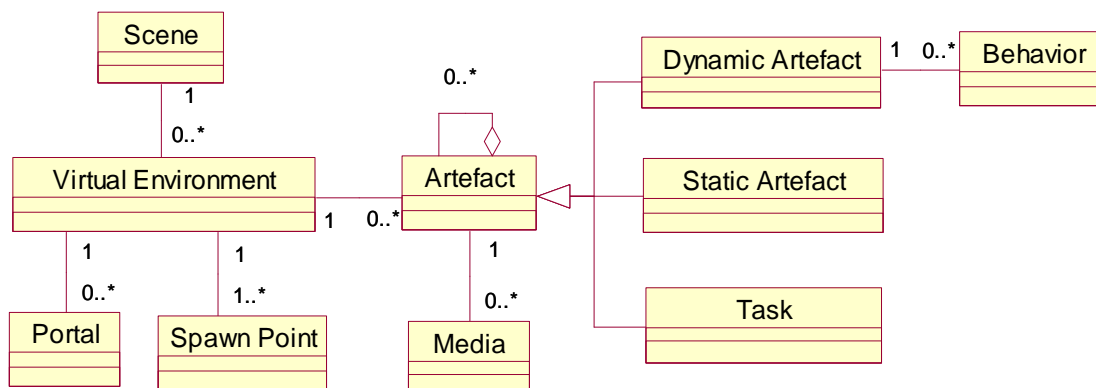


Figura 5.1. Organização dos arquivos do mundo virtual.

O arquivo **Virtual Environment** referencia todos os arquivos que constituem o ambiente virtual: um arquivo **Scene** contendo o cenário; os arquivos **Portal** que contêm a definição dos portais existentes no ambiente; um ou mais arquivos **Spawn Point** especificando os nascimentos; e os arquivos **Artefact** contendo referências aos objetos do ambiente virtual e a seus atributos. Quando um dado objeto for composto, seu arquivo **Artefact** faz referência aos arquivos **Artefact** de seus componentes. Um arquivo **Media** contém as descrições das mídias utilizadas por um objeto. Os arquivos **Static Object** e **Dynamic Object** são especializações dos arquivos **Artefact** que tratam respectivamente dos objetos estáticos e dinâmicos. O arquivo **Dynamic Object** de um objeto dinâmico está relacionado a um arquivo **Behavior** que define seu comportamento. O arquivo **Task** referencia uma tarefa, contendo sua descrição e suas dependências. Uma tarefa também é uma especialização de um objeto (sem mídias relacionadas), e, por isso, pode ser composta por subtarefas.

A utilização da organização de arquivos (Figura 5.1) facilita e reduz o trabalho de programação do CVE, pois a atribuição de comportamentos, a definição dos estados dos objetos, e a configuração de portais e nascimentos que antes eram feitas no CVE, agora são realizadas a priori. Para isso, basta que o CVE reconheça os arquivos publicados pela ferramenta de autoria.

5.4 Políticas de Acesso

Uma das características mais importantes de um sistema de autoria é o conjunto de políticas de acesso disponibilizadas por ele. De maneira geral, políticas de acesso são

responsáveis por coordenar a interação dos autores envolvidos. Normalmente, a política determina quem tem permissão para realizar ações, e quando e onde elas devem acontecer.

No MAC, o conjunto de políticas de acesso caracteriza-se pelo controle de acesso que atribui papéis aos usuários do sistema; e pelo controle de concorrência, que são técnicas de escalonamento para garantir que as ações dos usuários realizadas dentro do ambiente virtual sejam executadas na ordem correta.

5.4.1 Controle de Acesso

O Controle de acesso consiste na atribuição de responsabilidades aos usuários, definindo seu papel no sistema. Os papéis são elementos de destaque em um ambiente colaborativo, pois representam a noção de hierarquia dentro do grupo. Eles indicam os direitos e deveres dos membros sobre o trabalho colaborativo. O MAC define dois tipos de papel que um usuário pode desempenhar. O primeiro tipo refere-se aos seus privilégios no sistema, o segundo refere-se à função a ser realizada pelo usuário no trabalho colaborativo.

Quanto aos privilégios no sistema, o usuário pode desempenhar o papel de: Publicador, Navegante e Administrador.

Publicador. O publicador é o único que tem privilégios de criar novos ambientes virtuais e modificar os existentes no sistema. Quando um publicador deseja criar ou modificar um ambiente, todo o processo de instanciação de um novo CVE é realizado, como demonstrado no cenário 3 na Seção 4.5.1 Cenários de Utilização. Somente o publicador que instanciou o CVE tem o direito de publicá-lo, exceto quando ele compartilhar esse direito com outros usuários.

Navegante. Um navegante é um usuário que pode apenas explorar os ambientes virtuais disponíveis. Ele não possui permissão para criar, modificar ou publicar ambientes virtuais.

Administrador. Esse tipo de usuário é responsável pela manutenção do sistema e pelo cadastro de usuários. É ele quem define os privilégios dos usuários no sistema, e quem adiciona novos modelos de mundos virtuais na biblioteca de mundos virtuais (descrito posteriormente). A partir desses modelos, um publicador poderá criar novos ambientes virtuais. Além disso, o administrador pode excluir qualquer ambiente virtual disponível no sistema. O papel de administrador pode ser concedido a mais de um usuário simultaneamente.

Quanto à função no trabalho colaborativo, um usuário pode desempenhar o papel de: Editor, Colaborador, Espectador e Coordenador.

Editor. O editor é o proprietário do ambiente virtual que está sendo construído. Não pode haver mais de um proprietário para um ambiente virtual em edição. Geralmente, esse papel é atribuído ao publicador que instanciou o CVE. Entretanto, o editor pode escolher um substituto temporário para assumir seu papel quando ele estiver ausente. O editor, como proprietário do ambiente virtual, tem a responsabilidade de coordenar a construção desse ambiente. Para isso, ele define as tarefas que serão realizadas e seus responsáveis.

Colaborador. Um colaborador é responsável por auxiliar o editor na construção de um ambiente virtual. Um usuário recebe esse papel por intermédio de um convite do editor. Quando o usuário convidado é um navegante, ele tem seus privilégios ampliados até que seu papel de colaborador termine. Um outro publicador também pode ser convidado pelo editor do ambiente virtual a colaborar em sua construção.

Espectador. Um espectador é um usuário que pode apenas observar a realização do trabalho colaborativo dos outros usuários. Quando um editor inicia os trabalhos de construção de um ambiente virtual, ele decide se permitirá que espectadores possam entrar no ambiente virtual durante a construção. Permitir que espectadores conectem-se ao ambiente é uma forma de conseguir colaboradores para construção do ambiente virtual, pois o editor pode convidar espectadores para serem colaboradores.

Coordenador. Um coordenador é responsável por um grupo de colaboradores e por realizar uma tarefa atribuída ao seu grupo. O editor, que é o coordenador geral, tem a opção de dividir os colaboradores em grupos, escolher um colaborador de cada grupo para ser o coordenador de seu grupo, e atribuir uma tarefa para cada grupo. Cada coordenador de grupo, por sua vez, pode dividir seu grupo em subgrupos e atribuir subtarefas aos seus subcoordenadores. Dependendo da quantidade de colaboradores, a hierarquia de grupos poderá ter vários níveis de coordenadores.

5.4.2 Controle de Concorrência

Para evitar que um participante sobreponha a ação de outro, devem ser utilizados mecanismos de controle de concorrência de elementos compartilhados dentro de um ambiente virtual. Devido à natureza análoga dos problemas, os mecanismos de controle de concorrência assemelham-se às técnicas de escalonamento utilizadas em Sistemas Operacionais. Nesse último contexto, vários processos precisam utilizar o processador ao mesmo tempo. Esse conflito é solucionado através de técnicas de escalonamento.

Analogamente, as PDUs geradas pelas ações dos participantes devem ser processadas pelo GE e pelos clientes dos outros participantes.

As técnicas de escalonamento de processos utilizam uma estrutura-de-dados auxiliar (geralmente uma fila ou pilha) para definir a ordem dos processos que serão executados pelo processador. No MAC, cada componente responsável por tratar as PDUs dos participantes (os CIERVs e o GE) possui uma estrutura-de-dados auxiliar semelhante ao escalonamento de processos. Como a técnica de escalonamento utilizada é a mesma para cada um desses componentes, a ordem de processamento das PDUs é a mesma em todos os componentes, e assim, a consistência de todas as cópias do ambiente é mantida.

Tanenbaum & Woodhull (1997) destacam cinco técnicas de escalonamento de processos: FIFO, FILO, *Round Robin*, *Priority Policy* e *Random Policy*. A seguir são apresentadas essas técnicas, e para cada uma delas é feita uma analogia com a forma que o MAC emprega. A escolha de qual técnica de escalonamento deve ser utilizada depende da aplicação do MAC e da quantidade prevista de usuários no ambiente virtual.

FIFO (First In First Out)

Os processos prontos são colocados em uma fila organizada por ordem de chegada. Na sua vez, cada processo recebe o uso do processador até que seja completado, ou seja, o processo permanece em execução até que seja finalizado, de forma que os demais processos na fila fiquem esperando por sua oportunidade de processamento, assim sendo, o escalonamento FIFO é um algoritmo não-preemptivo¹⁵.

Dado que o escalonamento FIFO não concebe qualquer mecanismo de distinção entre processos (por exemplo, processos com diferentes níveis de prioridade), dando igual tratamento a todos os processos, processos importantes podem ficar em espera devido à execução de processos menos importantes.

De acordo com as considerações relativas ao escalonamento FIFO, uma forma de tratar as ações realizadas pelos usuários é organizando as PDUs enviadas por eles em filas de acordo com a ordem de chegada (uma para cada um dos CIERVs e para o GE). Quando chegar sua vez, a PDU processada é retirada de cada fila, de forma que as demais PDUs em cada fila fiquem esperando por sua oportunidade de ser processada. O fato do

¹⁵ Preempção: em um ambiente multitarefa, ação ou evento que causa mudança do processamento de uma aplicação para outra

tratamento a todas as PDUs e a todos os participantes ser igual faz com que participantes cujas PDUs deveriam ter maior importância (como coordenadores) não sejam favorecidos.

FILO (First In Last Out)

Os processos prontos são colocados em uma pilha, na qual o primeiro a entrar será o último processo a ser executado. A partir daí, o funcionamento é o mesmo do escalonamento FIFO, pois cada processo recebe o uso do processador até que seja completado, de forma que os demais processos na pilha fiquem esperando por sua oportunidade de processamento.

Do mesmo modo que os processos, as PDUs podem ser colocadas em pilhas (uma para cada um dos CIERVs e para o GE) e a partir daí, o processamento das PDUs é realizado de forma semelhante ao escalonamento FIFO. Esse escalonamento também peca por não fazer distinção entre os participantes.

Round Robin

Nessa técnica, também conhecida como escalonamento circular, os processos são organizados numa fila segundo a ordem de chegada e então são despachados para execução. No entanto, ao invés de serem executados até o fim, a cada processo é concedido apenas um intervalo de tempo (*timeslice* ou *quantum*). Caso o processo não seja finalizado nesse intervalo de tempo, ele é interrompido, enviado para o final da fila, e substituído pelo próximo processo da fila de processos ativos. O escalonamento *Round Robin* baseia-se na utilização de temporizadores. Ele é um algoritmo preemptivo, bastante adequado para ambientes interativos onde coexistem vários usuários, e em que é necessário garantir tempos de resposta razoáveis. O escalonamento *round robin* faz a suposição implícita de que todo processo é igualmente importante.

Analogamente ao tratamento do escalonamento *round robin* com processos, as PDUs geradas pelas ações realizadas pelos participantes são colocadas em filas circulares de acordo com a ordem de chegada (uma para cada um dos CIERVs e para o GE). Geralmente, uma ação de um participante gera mais de uma PDU, como por exemplo, movimentar-se no ambiente virtual gera duas PDUs. Assim, ao invés de utilizar o critério de intervalo de tempo na técnica de *round robin* para processos, é utilizado o critério de processamento de uma PDU de cada ação por vez. Por exemplo, se dez participantes se movimentassem ao mesmo tempo, a primeira PDU de movimento de cada participante seria processada em um primeiro instante (primeiro ciclo na fila), e depois a movimentação

dos dez participantes seria concluída com a execução da segunda PDU de movimento em um segundo instante (segundo ciclo na fila). Com essa técnica, a movimentação seria mais realista, pois os dez participantes iniciariam a movimentação simultaneamente, ao contrário das técnicas anteriores. Essa técnica também não faz distinção entre os participantes.

Priority Policy

Nessa técnica, também conhecida como escalonamento por prioridades, a cada processo é atribuída uma prioridade, e o processo com a maior prioridade recebe permissão para executar. Para evitar que processos de alta prioridade executem indefinidamente, o escalonador pode diminuir a prioridade do processo atualmente em execução a cada intervalo de tempo. Muitas vezes é conveniente agrupar processos em classes de prioridade e utilizar escalonamento por prioridade entre as classes e escalonamento *round robin* dentro de cada classe.

Considerando que cada participante pode ser considerado como um processo pelo escalonamento por prioridades, a cada participante é atribuída uma prioridade. O mesmo raciocínio pode ser usado com as PDUs. Por exemplo, as PDUs geradas pelo coordenador geral (Editor) ou por algum outro coordenador têm maior prioridade que as outras e são executadas primeiro. Assim, a hierarquia de prioridades no MAC é descrita, da maior para a menor, como: coordenador geral, subcoordenador, colaboradores, e espectadores.

Random Policy

Também chamado de escalonamento por sorteio ou escalonamento aleatório, essa técnica tem como idéia básica distribuir bilhetes de loteria para cada processo. Sempre que uma decisão de escalonamento tiver de ser feita, um bilhete de loteria é escolhido aleatoriamente, e o processo que armazena este bilhete recebe o recurso. Os processos mais importantes podem receber bilhetes extras, para aumentar suas chances de ganhar.

De acordo com o conceito do escalonamento aleatório, cada participante recebe um bilhete de loteria. Os participantes com mais privilégio, como os coordenadores, podem receber bilhetes extras para aumentar suas chances de ganhar, como mencionado antes. Sempre que uma decisão de escalonamento tiver de ser tomada, um bilhete de loteria é escolhido aleatoriamente, e as PDUs da ação do participante que detém esse bilhete são processadas.

5.5 Coordenação

O processo de coordenação no MAC é realizado em três etapas: planejamento, execução e análise. O planejamento é realizado logo após o ambiente ser instanciado pelo publicador. A execução, que é realizada após o planejamento, é a etapa mais dinâmica, devido à ocorrência de constantes negociações durante todo o processo. A análise é a etapa final, realizada após a execução.

No Planejamento, o editor identifica os objetivos, mapeia esses objetivos em tarefas e define suas interdependências. As interdependências servem para estabelecer a ordem no processo de execução de tarefas. Em princípio, é responsabilidade do editor, gerenciar as interdependências entre tarefas e garantir que essas dependências não sejam violadas. Entretanto, em alguns tipos de aplicação, mecanismos automáticos podem ser implementados para garantir que as dependências não sejam violadas. Baseado em Raposo e seus co-autores (2000), o MAC especifica dois tipos de dependências de tarefas: dependências temporais e dependências de recursos.

As seis dependências temporais especificadas são descritas a seguir.

Tarefa 1 equals tarefa 2. Essa dependência estabelece que duas tarefas devem ser executadas simultaneamente.

Tarefa 2 after tarefa 1. A execução da tarefa 2 só pode ocorrer após a execução da tarefa 1.

Tarefa 1 before tarefa 2. Do ponto de vista da lógica temporal, essa relação pode ser vista como oposta à anterior, mas gera um mecanismo de coordenação totalmente diferente. Essencialmente, a restrição ocorre na execução da tarefa 1, que só pode ocorrer se a tarefa 2 ainda não tiver iniciado sua execução. A tarefa 2 não espera pela execução da tarefa 1, como ocorre na relação tarefa 2 *after* tarefa 1.

Tarefa 1 meets tarefa 2. A execução da tarefa 2 inicia imediatamente após o término da tarefa 1.

Tarefa 1 overlaps tarefa 2. Essa relação estabelece que a tarefa 2 deve iniciar antes do término da tarefa 1, e terminar após a tarefa 1 ser concluída.

Tarefa 2 during tarefa 1. A execução da tarefa 2 ocorre durante a execução da tarefa 1.

As dependências de recurso lidam com a distribuição dos recursos entre as tarefas. Os três mecanismos básicos são descritos a seguir.

Divisão de recursos. Um número limitado de recursos precisa ser compartilhado entre várias tarefas. É um caso muito comum que ocorre, por exemplo, quando vários computadores compartilham uma impressora, uma área de memória, etc.

Simultaneidade no uso de recursos. O recurso só fica disponível se um determinado número de tarefas precisarem utilizá-lo simultaneamente. Esse é o caso que ocorre, por exemplo, quando uma máquina precisa de mais de um operador.

Volatilidade de recursos. Essa dependência indica se o recurso volta a estar disponível após ser utilizado. A impressora não é um recurso volátil, mas uma folha de papel é.

As dependências de gerenciamento de recursos, diferentemente das temporais, não são relações binárias. É possível, por exemplo, que mais de duas tarefas compartilhem um recurso. Além disso, essas dependências requerem um parâmetro para indicar o número de recursos compartilhados, o número de tarefas que devem solicitar um recurso simultaneamente, ou o número de vezes que um recurso pode ser utilizado (volatilidade).

Na etapa de execução, o editor organiza os colaboradores em grupos e distribui as tarefas entre eles. A execução é uma etapa muito dinâmica, pois há uma constante negociação sobre a redefinição das tarefas e suas interdependências, bem como sobre a reorganização dos participantes. Geralmente, a execução ocorre de forma síncrona, na qual o trabalho colaborativo é realizado com a presença do editor e de todos os colaboradores simultaneamente. Entretanto, a execução pode ser realizada também de forma assíncrona. Para que isso seja possível, o editor deverá definir previamente o papel e as tarefas de cada participante. Porém, dessa forma nenhum espectador poderá tornar-se colaborador, a menos que o editor esteja presente. A presença do editor é muito importante para o desenvolvimento do trabalho colaborativo, por isso, o editor pode escolher substitutos para desempenhar seu papel quando estiver ausente.

A organização da equipe em grupos fica a critério do editor. Se ele preferir coordenar todo o grupo, no caso de a equipe ser pequena, não haverá hierarquia de grupos. Caso contrário, no momento em que o editor cria um grupo e atribui privilégios de coordenador para um colaborador, este poderá criar novos grupos que estarão sob seu controle e a hierarquia poderá crescer sem a intervenção do editor. Cada coordenador de grupo pode dividir sua tarefa em subtarefas e atribuí-las aos seus subordinados, portanto, a hierarquia de tarefas também poderá crescer sem a intervenção do editor.

A etapa de análise envolve a avaliação das tarefas realizadas, a verificação da satisfação dos objetivos identificados durante o planejamento e a documentação do processo de colaboração.

5.6 Percepção

Como visto no Capítulo 2, a percepção em atividades colaborativas é importante, pois evita a falta de contexto sobre as ações dos colegas e do grupo. As ações realizadas fora de contexto podem gerar redundâncias, inconsistências e contradições dentro do trabalho do grupo. Conseqüentemente, o trabalho gerado pode ser de baixa qualidade e não atingir os objetivos propostos. Em geral, o trabalho apresentar-se truncado e sem coesão de idéias.

Segundo Pinheiro e seus co-autores (2002), o ambiente colaborativo deve empregar mecanismos que respondam às seis questões: *o que, quando, onde, como, quem e quanto*; as quais identificam aspectos vitais para o fornecimento de percepção. O MAC propõe mecanismos que visam responder a todas essas questões.

A questão “*o que*” refere-se às informações que devem ser fornecidas aos usuários. Essas informações são adquiridas pelo conhecimento do papel de cada participante e da tarefa sob sua responsabilidade. Por exemplo, se um participante pertence a um grupo, ele precisa conhecer seu papel dentro do grupo (coordenador ou colaborador), e as tarefas que foram atribuídas a ele ao restante de seu grupo.

A questão “*quando*” refere-se ao momento de ocorrência dos eventos geradores das informações e ao momento de apresentação dessas informações. Essas informações são adquiridas no MAC através do conhecimento do histórico das tarefas. Cada tarefa possui um estado indicando se ela foi concluída, se está incompleta e se ainda não foi iniciada. O estado das tarefas é usado no histórico para referenciar: *passado* (tarefas concluídas que ocorreram em um intervalo de tempo no passado); *passado contínuo* (tarefas incompletas que começaram no passado, mas que continuam válidas); *presente* (tarefas incompletas que ainda estão sendo realizadas) e *futuro* (tarefas do grupo ainda não iniciadas).

A questão “*como*” refere-se à forma com informações são apresentadas aos usuários, como é sua interface. No caso do MAC, que faz uso de RV, são utilizadas interfaces com maior acoplamento, como interfaces WYSIWIS (*What You See Is What I See*), e WYSIWIS relaxadas. Nas interfaces WYSIWIS, todos os participantes têm a mesma visão do ambiente virtual e de seus elementos, o que garante o contexto das atividades. Entretanto, essas são interfaces muito restritivas, podendo prejudicar o andamento do trabalho colaborativo. Uma alternativa para essas interfaces são as WYSIWIS relaxadas, que garantem maior liberdade aos indivíduos, como, por exemplo, a livre navegação no ambiente virtual.

A questão “*quem*” refere-se à identificação de quem está trabalhando e presente no momento. Essa questão trata do conhecimento da hierarquia do grupo, ou seja, cada participante precisa saber quem são seus companheiros de grupo, quem são seus subordinados, quem é seu coordenador direto e quem é o coordenador geral. Como o MAC é um modelo para autoria síncrona e assíncrona, ele ainda precisa fornecer ao participante a informação de quem está presente e ausente (*online e offline*) no ambiente virtual.

A questão “*quanto*” refere-se à quantidade ideal de informações que deve ser apresentada ao usuário. Essa questão depende da quantidade de usuários da aplicação que faz uso do MAC. Visando fornecer informações suficientes, sem sobrecarregar o usuário, cabe aos desenvolvedores limitar a quantidade de informações apresentadas.

5.7 Comunicação

Os mecanismos de comunicação são fundamentais para que a colaboração ocorra, e, portanto, essenciais para o trabalho colaborativo. O MAC emprega dois mecanismos de comunicação: o verbal, através de bate-papo textual ou vídeo conferência; e o não-verbal, através de animações do personagem virtual dentro do ambiente virtual (postura, olhar, gestos, expressões faciais, etc.).

Através desses mecanismos, os participantes podem: definir o papel de cada um dos envolvidos (hierarquia de grupo); analisar a divisão das tarefas; discutir a atribuição das dependências entre elas; e definir a responsabilidade por cada uma delas. Para auxiliar a autoria assíncrona, o processo de comunicação textual é registrado. Isso ajuda os participantes a entender o contexto das mudanças que foram realizadas quando eles não estavam presentes.

A comunicação também intermedeia o processo de negociação que ocorre durante as discussões sobre a forma como o trabalho está sendo realizado, para tomar decisões sobre situações não previstas inicialmente.

5.8 Considerações Finais

Os mecanismos definidos pelo MAC neste capítulo fornecem meios para o trabalho colaborativo, pois propiciam a discussão, a troca de informações e idéias, a formulação e a resolução conjunta de problemas, a motivação para a participação, o sentimento de responsabilidade conjunta por algo. Apesar do desempenho do grupo na resolução das tarefas ser extremamente dependente de fatores contextuais (composição do grupo, características e habilidades dos membros e tipo do problema abordado), os mecanismos

do MAC dão suporte à interação entre os participantes de forma a minimizar as dificuldades contextuais.

A existência de modelos como o MAC que auxiliem na construção de ambientes virtuais é importante, não apenas para a equipe especializada em construir novos ambientes virtuais, como também para o usuário final, o qual pode construir seus próprios ambientes virtuais ou editar os ambientes existentes. Isso permite ampliar o número de ambientes virtuais disponíveis, além de aumentar a motivação dos usuários finais do sistema, por torná-los agentes modificadores dos ambientes que exploram.

Capítulo 6

Estudo de Caso da CRAbCVE Utilizando o Modelo MAC

6.1 Introdução

Este capítulo tem por objetivo apresentar o estado atual da implementação do conjunto de componentes definidos pela arquitetura CRAbCVE seguindo as especificações do modelo MAC. As implementações estão relacionadas ao projeto Sintegra – Sistema integrado de Educação a Distância com suporte a Avaliação de Competências – resultado de uma parceria entre SENAC/CE e a Universidade Federal do Ceará, e financiado pelo programa RHAE/CNPq.

O restante deste capítulo está dividido da seguinte forma: na Seção 6.2, é apresentado o projeto Sintegra ao qual o estudo de caso é aplicado; na Seção 6.3, apresenta-se o curso de Boas Práticas que é um dos dois cursos propostos pelo Sintegra e é utilizado como material de apoio no estudo de caso; na Seção 6.4, apresentam-se os componentes da arquitetura CRAbCVE que foram implementados para o estudo de caso; e, por último, na Seção 6.5, apresentam-se algumas considerações finais.

6.2 Sintegra

O projeto Sintegra (Sistema integrado de Educação a Distância com suporte a Avaliação de Competências) tem como objetivo principal o desenvolvimento de um ambiente computacional integrado para aprendizagem a distância através da *Web* (Sintegra, 2005). Um importante diferencial desse sistema é a inclusão de serviços de avaliação de competências de participantes – uma necessidade premente do SENAC/CE. Esses serviços merecem destaque especial, por estabelecerem novas considerações sobre modelos de educação a distância. O sistema proposto também faz uso intensivo de recursos multiusuários de realidade virtual voltados para treinamento. Esses recursos, atuais objetos de estudo da Universidade Federal do Ceará – o outro parceiro do Projeto –, se adequam perfeitamente ao ensino profissionalizante praticado pelo SENAC.

Para atingir o objetivo proposto, respeitando as peculiaridades do sistema, serão criadas metodologias pedagógicas específicas, que contemplem o ensino a distância através da internet e de modelos baseados em realidade virtual. Da mesma forma, também

será necessária a criação de uma metodologia especial para a geração de conteúdos direcionados a esse tipo de sistema. O desenvolvimento de tais metodologias, abordando, inclusive, modelos teóricos para avaliação de competências, também é um dos objetivos do Projeto.

O projeto propõe a geração de três ferramentas voltadas para a simplificação e a otimização do uso do sistema. A primeira delas consiste em um sistema de autoria de mundos virtuais, voltada para a criação de ambientes de realidade virtual por alunos e professores. A segunda ferramenta é um simulador de ambientes, para gerenciar o comportamento dos diversos participantes do modelo de realidade virtual, abrangendo, inclusive, interações com objetos e possíveis reações a determinados estímulos. Esta ferramenta é essencial para o realismo das simulações propostas aos participantes dos cursos. A terceira, um gerenciador de cargas, visa controlar o fluxo de dados transmitidos na rede de comunicação empregada, modificando configurações e, caso seja necessário, incluindo novos computadores para o correto balanceamento do poder computacional envolvido. Para avaliar o sistema proposto, serão especificados e implementados dois cursos, na forma de módulos, que serão aplicados a turmas de alunos do SENAC.

6.3 Curso de Boas Práticas

O curso Boas Práticas é um dos dois cursos propostos pelo Sintegra. Ele é desenvolvido completamente a distância e perfaz um total de 20 horas/aula. Destina-se ao aperfeiçoamento de profissionais ou estudantes de Nutrição, Engenharia de Alimentos e Economia Doméstica que trabalham com a manipulação de alimentos.

O curso de Boas Práticas estabelece um conjunto de princípios e regras para o correto manuseio de alimentos, abrangendo, desde as matérias-primas, até o produto final, de forma a garantir a saúde do consumidor. Ele é realizado através da metodologia de educação a distância, envolvendo as seguintes estratégias com o objetivo de gerar colaboração: leitura de textos on-line, análise de situações-problema, estudo de casos, dinâmicas de grupo, fóruns de discussão, formação de comunidade virtual, pesquisa orientada, e projeto.

A avaliação será feita por meio da observação do desenvolvimento de competências ao longo do curso. Serão observados os seguintes critérios: participação no ambiente virtual, utilização das ferramentas, colaboração, criatividade, clareza, e assiduidade na entrega das tarefas. Às atividades serão atribuídos valores quantitativos e qualitativos expressos em: tarefas corretas; tarefas concluídas no prazo; resultados para competências

D (desenvolvida), N (não desenvolvida) e P (parcialmente desenvolvida); conceitos MB (muito bom), B (bom), R (regular) e I (insuficiente). Para os resultados de competência P e N serão propostas correções e atividades complementares com o objetivo de atingir D buscando dar um caráter formativo à avaliação.

O curso de Boas Práticas é ministrado através do SOLAR (Sistema *Online* de Aprendizagem (SOLAR, 2005)). O SOLAR é um ambiente virtual de aprendizagem desenvolvido pelo Instituto UFC Virtual, da Universidade Federal do Ceará. Ele é orientado ao professor e ao aluno, e possibilita a publicação de cursos e a interação com esses cursos. O SOLAR foi desenvolvido potencializando o aprendizado a partir da relação com a própria interface gráfica do ambiente. Essa interface permite que o usuário tenha rapidez no acesso às páginas e ao conteúdo, além de oferecer fácil navegação e compatibilidade com Navegadores do mercado.



Figura 6.1. Ambiente Solar e o curso Boas Práticas.

6.4 Estudo de Caso

O estudo de caso implementado será utilizado no processo de avaliação de competência do curso de Boas Práticas. Os alunos utilizarão o sistema de autoria implementado, para estruturar uma cozinha que siga os padrões de qualidade para a manipulação de alimentos. Eles adotarão os procedimentos apresentados no curso de Boas Práticas para controlar a qualidade e a segurança dos alimentos produzidos na cozinha e, assim, impedir a contaminação dos alimentos.

O estudo de caso exposto neste trabalho refere-se a um ambiente virtual definido de acordo com as especificações do SENAC/CE para estruturação de uma cozinha de boa qualidade. Nesse ambiente, participarão seis usuários: Lia, João, Pedro, Carlos, Lúcia e Marcos. A principal responsável pelo ambiente é a Lia, e, conseqüentemente, no sistema, possui o privilégio de publicadora e, no trabalho colaborativo, desempenha os papéis de

editora e coordenadora geral. João também é publicador, mas nesse ambiente ele será apenas um colaborador. Pedro e Carlos são Navegantes com papel de colaboradores no ambiente. Lúcia também é uma navegante, mas será apenas uma espectadora. Por último, o Marcos é o administrador do sistema, responsável pela manutenção e pelo cadastro dos seis usuários.

6.4.1 Gerenciador de Políticas do Sistema

O GPS (descrito na Seção 4.4.1) é um servidor que possui uma base de dados com o cadastro de todos os usuários e CVEs do sistema. Para um usuário ter acesso a um CVE, é preciso ter um cadastro no GPS, e nesse cadastro é definido seu privilégio no sistema (Publicador, Navegante ou Administrador). O GPS foi implementado para esse estudo de caso como um *Web Service*¹⁶ (Figura 6.2, diagrama de classes no Anexo C) utilizando um protocolo XML¹⁷, o SOAP¹⁸. As funcionalidades do GPS implementadas foram:

- ♦ **Adição de CVEs** - Insere um cadastro de um novo CVE no sistema;
- ♦ **Atualização de CVEs** - Altera o registro de CVE existente;
- ♦ **Exclusão de CVEs** - Exclui um CVE do sistema;
- ♦ **Consulta de um CVE** - Retorna as informações de um CVE;
- ♦ **Consultas de CVEs** - Retorna uma lista de CVEs disponíveis para um usuário específico de acordo com seus privilégios no sistema;
- ♦ **Adição de usuários** - Adiciona um novo usuário;
- ♦ **Atualização de usuários** - Atualiza o cadastro de um usuário;
- ♦ **Exclusão de usuários** - Remove um usuário do sistema;
- ♦ **Consulta de usuários** - Retorna as informações de um usuário; e,
- ♦ **Checagem de usuário** - Verifica se um usuário está cadastrado no sistema.

No estudo de caso, a base de dados do GPS é interligada com o SOLAR para que os usuários cadastrados no curso de Boas Práticas também estejam cadastrados no GPS.

¹⁶ A tecnologia de *Web Service* 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 (Wikipedia, 2005).

¹⁷ 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 (W3C, 2005).

¹⁸ 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 (Box et al., 2005).

Marcos, que é administrador de ambos os sistemas, cadastra todos os outros usuários definindo o papel deles no ambiente (Lia e João como publicadores; Pedro, Carlos e Lúcia como navegantes).

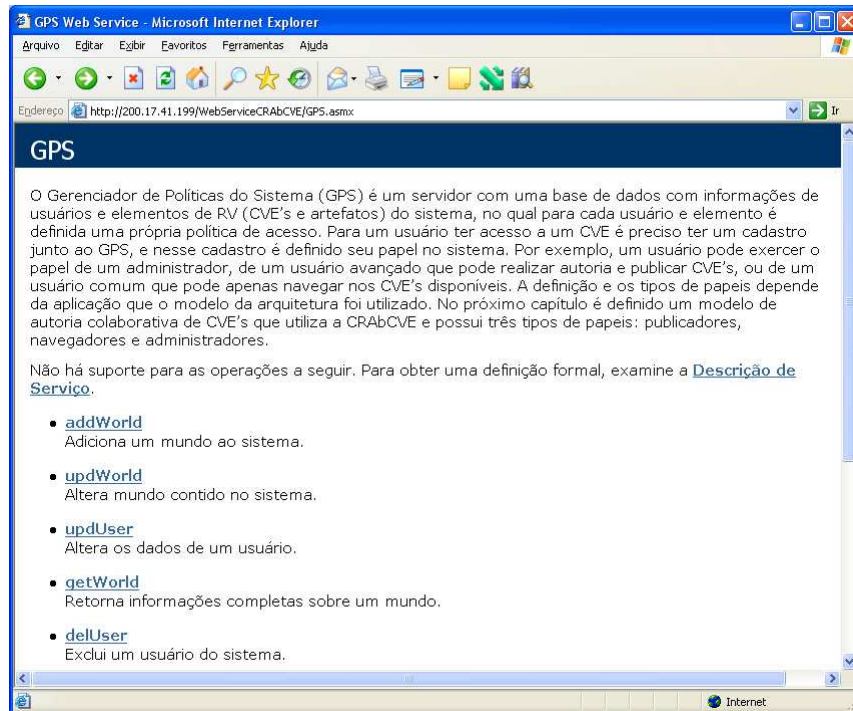


Figura 6.2. Implementação do GPS como um *Web Service*.

6.4.2 Fornecedor de Mídias

O FM (descrito na Seção 4.4.6) é um servidor que atua como um repositório de mídias. Entre essas mídias, estão presentes todos os elementos tridimensionais do ambiente virtual. O FM armazena também os arquivos que descrevem a formação dos mundos virtuais e os objetos. Esses arquivos descritores são formatados utilizando XML e são interpretados pelo GE e pelo CIERV (Exemplos desses arquivos encontram-se no Anexo A e B). Assim como o GPS, o FM foi implementado como um *Web Service* (Figura 6.3, Figura 6.4, diagrama de classes no Anexo C). As funcionalidades implementadas foram:

- ♦ **Adição de arquivos** - Insere uma mídia ou um arquivo descritor;
- ♦ **Exclusão de arquivos** - Remove um arquivo;
- ♦ **Consulta de mídia** - Solicita informações sobre uma mídia armazenada; e,
- ♦ **Atualização de mídia** - Atualiza informações de uma mídia específica.

Para o estudo de caso, Marcos, o administrador, deve adicionar previamente no FM todos os elementos necessários para a estruturação do ambiente virtual da cozinha, mesmo que esses elementos estejam armazenados localmente nos clientes dos usuários. Isso é necessário devido à importância de se manter um *backup* das mídias no FM. Armazenados

no FM, encontram-se: a estrutura arquitetônica do ambiente virtual, os objetos, os arquivos descritores, as imagens, os sons, os vídeos, etc.

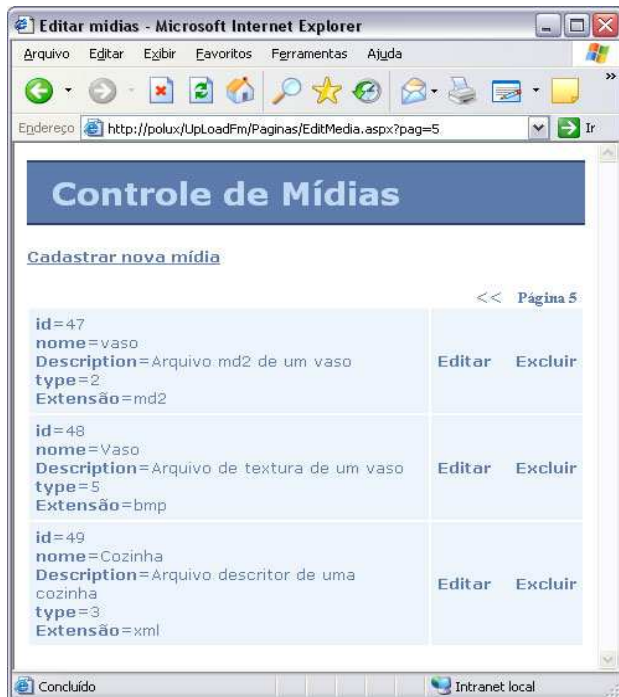
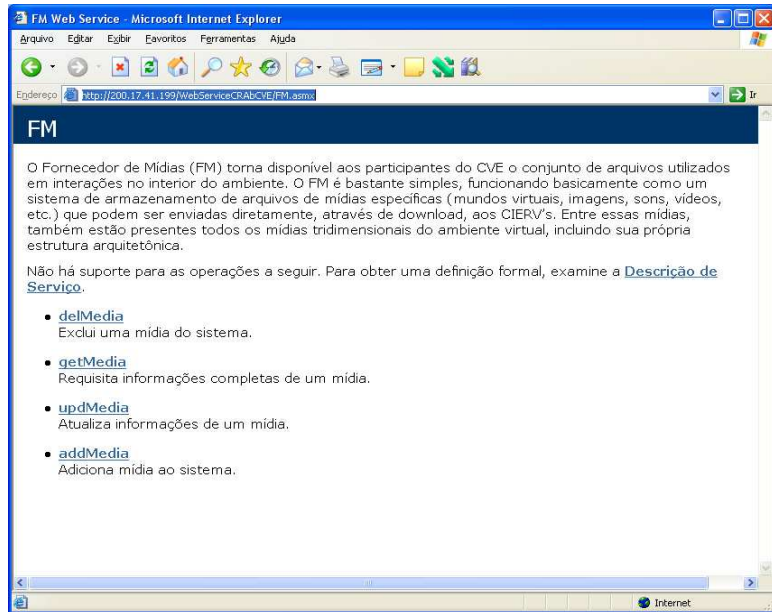


Figura 6.3. Implementação do FM como um Web Service.

Figura 6.4. Na figura à esquerda, a lista de mídias cadastradas. Na direita, um formulário para cadastrar novas mídias. Ambas são Web Services.

6.4.3 Roteador de Mensagens

O RM (descrito na Seção 4.4.3) é responsável por tornar as conexões do ambiente virtual da cozinha disponíveis aos seus participantes. Ele é empregado no repasse de informações a outros elementos da arquitetura CRAbCVE a fim de manter o estado compartilhado desse CVE e de viabilizar a colaboração entre seus participantes. O RM, em associação com o GE, é um dos componentes responsáveis por sincronizar as diversas cópias do ambiente virtual.

Neste estudo de caso, visando facilitar o uso do RM em diferentes tipos de sistemas operacionais, ele foi implementado na linguagem C++ (Figura 6.5, diagrama de classes no Anexo C), que possui compiladores em quase todas as plataformas, e uma biblioteca genérica de *sockets*: a ENet (ENet, 2005).

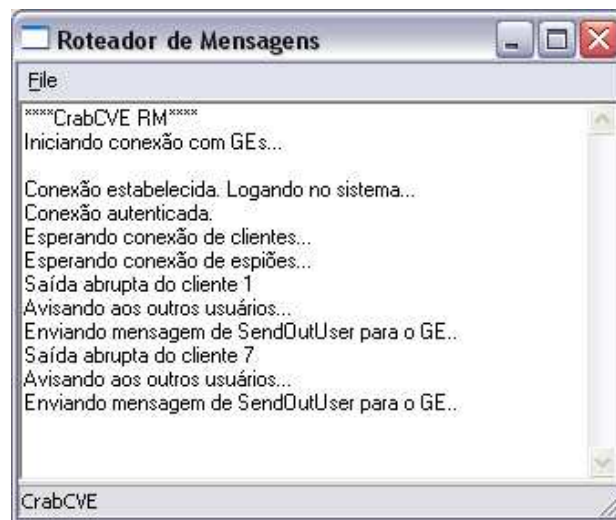


Figura 6.5. Implementação do Roteador de Mensagens mostrando o registro de ocorrências do sistema.

6.4.4 Gerenciador de Eventos

O GE (descrito na Seção 4.4.4) é o componente responsável por iniciar o funcionamento do ambiente virtual e por tratar eventos oriundos das interações de seus participantes, gerando PDUs de resposta sempre que necessário. Assim como o RM, o GE também foi implementado usando a linguagem C++ e a biblioteca ENet, garantindo sua portabilidade (Figura 6.6 e Figura 6.7, diagrama de classes no Anexo C).

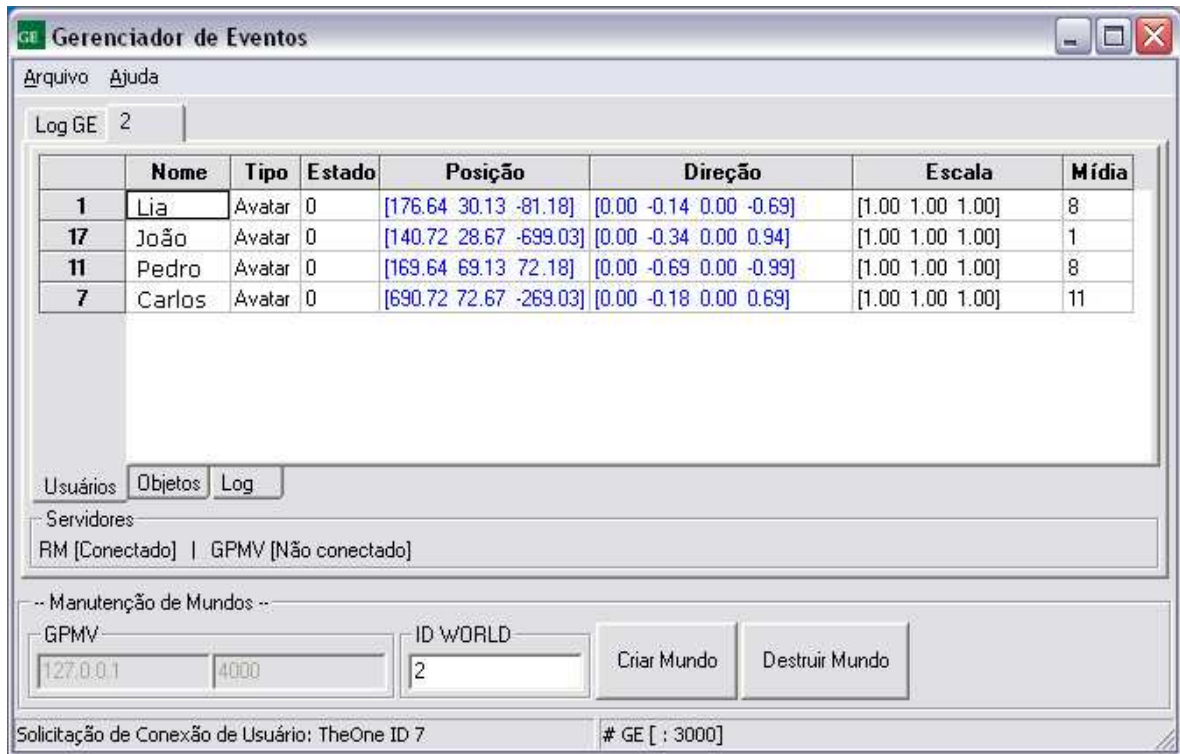


Figura 6.6. GE mostrando os usuários conectados.

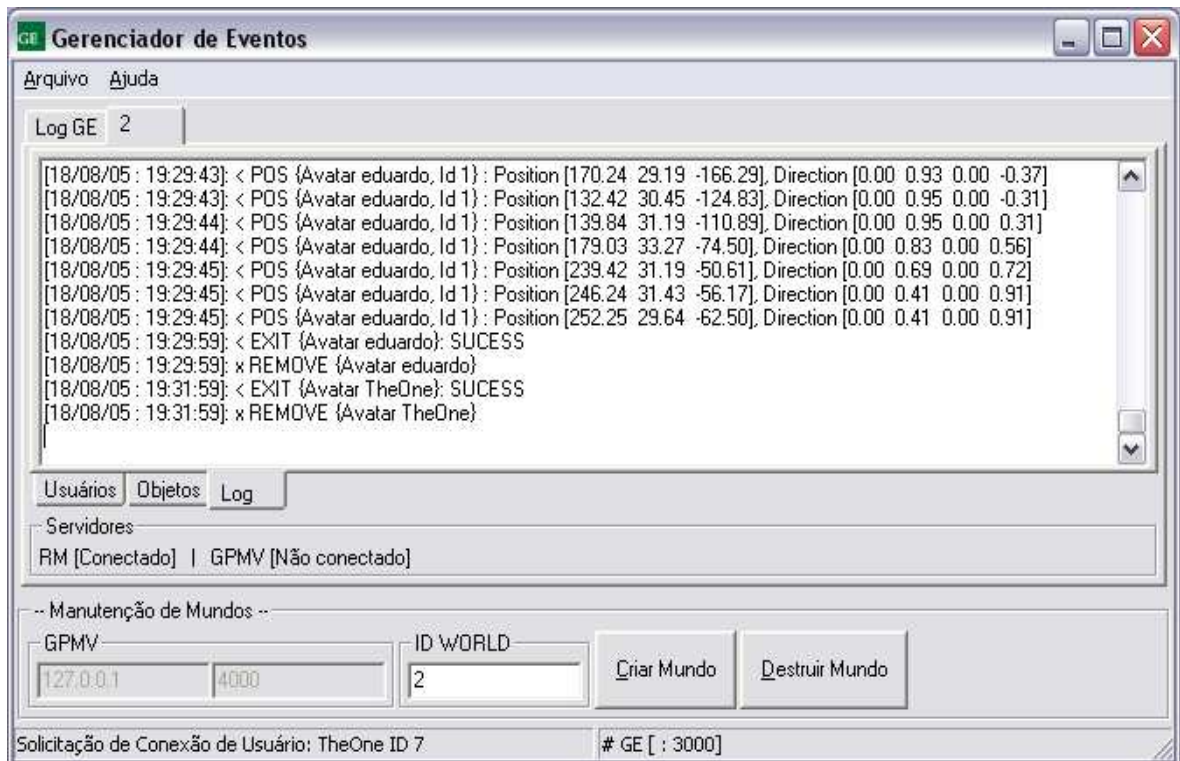


Figura 6.7. GE mostrando registro de PDUs que chegam e que são enviadas.

6.4.5 Cliente de Interação e Edição em RV

Os recursos do ambiente virtual estão disponíveis para os usuários através do CIERV (descrito na Seção 4.4.2), que se conecta diretamente ao RM. Essa aplicação cliente faz uso

de uma interface gráfica tridimensional (GUI) (Figura 6.8, diagrama de classes no Anexo C) para a realização de todo o processo de geração e apresentação do mundo virtual.

O modelo de CVE utilizado neste estudo de caso é *Desktop Virtual Reality*¹⁹, e o gerenciamento dos elementos gráficos do ambiente virtual é feito através do motor gráfico CRAbGE (Maia, 2003). O CRAbGE é um motor gráfico aberto, portátil e extensível, voltado para aplicações de realidade virtual e particularmente para jogos. Também visando ser portátil, o CIERV foi implementado com a linguagem C++ utilizando a biblioteca gráfica wxWidgets (wxWidgets, 2005) para construção de sua interface gráfica. A comunicação do CIERV com os *WebServices* GPS e FM é feita através do pacote gSOAP (gSOAP, 2005). Esse pacote foi desenvolvido para permitir que aplicações clientes que utilizem a linguagem C/C++ se comuniquem via SOAP (Box *et al.*, 2000) com *Web Services*. Já a comunicação do CIERV com o RM é feita através de um cliente do protocolo PASSÁRGADA 2.0 que utiliza a biblioteca ENet. Por último, a leitura e a escrita dos arquivos XML (descritores de mundos virtuais e objetos) são realizadas pela biblioteca TinyXML (TinyXML, 2005).

A seguir serão apresentados os cenários mais relevantes da implementação do CIERV para o estudo de caso proposto. Esses cenários estão relacionados com os apresentados na Seção 4.5.1 Cenários de Utilização.



Figura 6.8. Tela do Cliente de Interação e Edição em RV (CIERV).

¹⁹ *DesktopVirtual Reality* utilizam computadores comuns e não necessitam de hardware especial.

Cenário 1: Conectando ao GPS

Dos seis usuários deste estudo de caso apenas cinco irão entrar no ambiente virtual da cozinha: Lia, João, Pedro, Carlos e Lúcia. Marcos, o administrador do sistema, não participará da interação. Para ter acesso ao ambiente, todos esses cinco usuários deverão acessar antes o GPS para se autenticarem (Figura 6.9).



Figura 6.9. Tela de autenticação do CIERV com o GPS.

Cenário 2: Conectando a um Mundo Virtual

Após ser autenticado pelo GPS, cada um dos usuários recebe uma lista personalizada de mundos virtuais que estão disponíveis para ele. Por exemplo, Lia e João possuem listas de mundos virtuais diferentes, veja Figura 6.10. Entre esses mundos virtuais está o ambiente da cozinha utilizada neste estudo de caso.

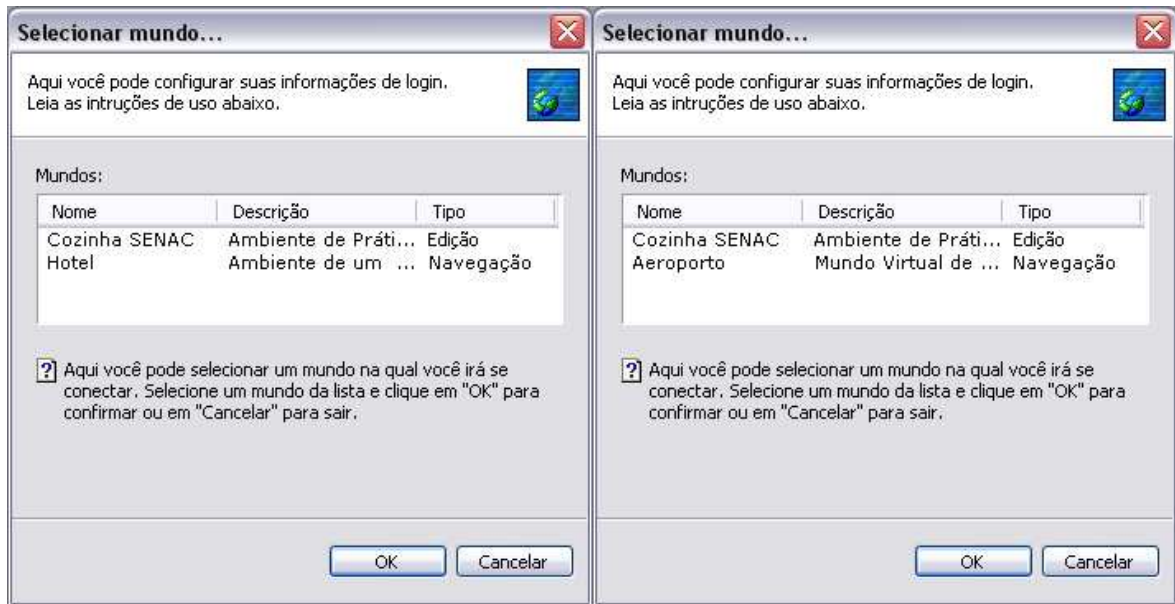


Figura 6.10. À esquerda, tela com a lista de mundos do usuário Lia, e a direita, tela com a lista de mundos do usuário João.

Cenário 3: Inserindo Tarefas

Lia, a editora do ambiente, tem a responsabilidade de analisar o ambiente virtual e definir as tarefas que serão necessárias para atingir o objetivo, que é construção da cozinha seguindo as especificações descritas no curso de Boas Práticas. Além de definir as tarefas, ela também é responsável por definir as dependências temporais dessas tarefas. Os tipos de dependências utilizadas neste estudo de caso foram quatro: ‘antes de’, ‘depois de’, ‘durante’ e ‘ao mesmo tempo’. Os tipos ‘antes de’ ou ‘depois de’ são restrições que definem que uma tarefa só pode começar, ou antes, ou depois de outra tarefa, respectivamente. O tipo de dependência ‘durante’ define que uma tarefa deve começar depois de outra tarefa começar e antes desta terminar. O tipo ‘ao mesmo tempo’ define que duas tarefas devem começar e terminar no mesmo instante. As tarefas estão disponíveis no CIERV em uma árvore, que define tarefas e subtarefas. A partir dessa árvore, o usuário poderá modificar as informações de uma tarefa, como nome, responsável, dependências e estado (veja Figura 6.11). As tarefas deste estudo de caso e suas dependências estão definidas na Tabela 21.

Tabela 21. Lista de tarefas e suas dependências

Tarefas	Tarefas dependentes	Tipo de dependência
Tarefa 1. Dispor móveis	Distribuir eletrodomésticos Distribuir eletroportáteis	antes de antes de
Tarefa 2. Distribuir eletrodomésticos	Dispor móveis Distribuir eletroportáteis	depois de ao mesmo tempo
Tarefa 3. Distribuir eletroportáteis	Dispor móveis Distribuir eletrodomésticos	depois de ao mesmo tempo
Tarefa 4. Colocar utensílios domésticos	Distribuir eletrodomésticos Distribuir eletroportáteis	depois de durante
Subtarefa 4.1. Colocar prataria e panelas	Colocar material de limpeza	depois de
Subtarefa 4.2. Colocar material de limpeza	Colocar prataria e panelas	antes de



Figura 6.11. À esquerda, tela de criação de tarefas, à direita, tela de inserção de dependências.

Cenário 4: Montando a Hierarquia de Grupos

Todos os usuários que entram pela primeira vez em um ambiente virtual em edição têm o papel no trabalho colaborativo de espectador, com exceção do editor desse ambiente que tem o papel de coordenador geral. Assim, neste estudo de caso, à medida que os usuários

vão entrando no ambiente como espectadores, a coordenadora geral Lia vai atribuindo papéis a cada um deles. Inicialmente, Lia atribuiu o papel de colaborador a Pedro e a Carlos e o papel de coordenador a João, tendo como seus subordinados Pedro e Carlos. Lúcia, que entrou por último, ficou como espectadora. A hierarquia de grupo é disponibilizada no CIERV através de uma árvore, a partir da qual, os coordenadores poderão atribuir tarefas a usuários, e criar ou mudar a formação de grupos (veja Figura 6.12).

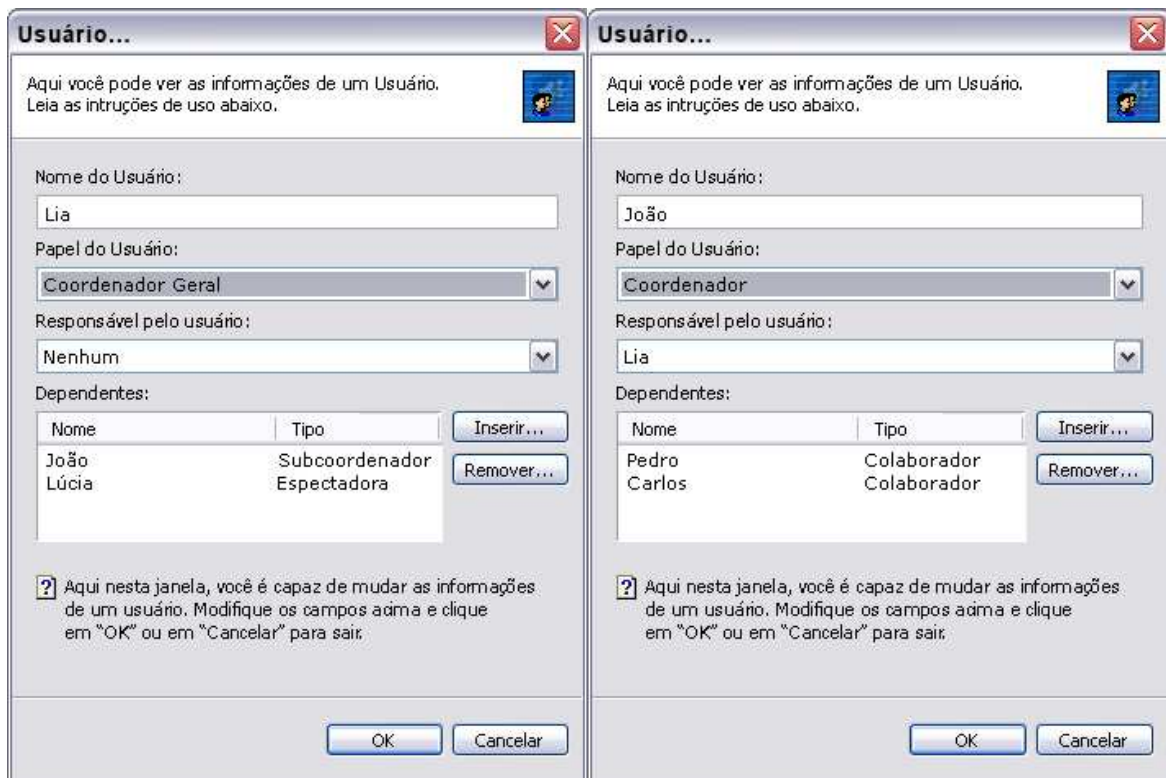


Figura 6.12. À esquerda, a tela com informações sobre o usuário Lia, coordenador geral. À direita, a tela com informações sobre subcoordenador João.

Cenário 5: Inserindo e Manipulando Objetos

Para que os usuários consigam atingir o objetivo de montar o ambiente virtual da cozinha, eles precisam inserir e manipular objetos nesse ambiente. Para isso, o CIERV disponibiliza uma biblioteca de objetos e um conjunto de ferramentas para manipular esses objetos (Figura 6.13).

Com a biblioteca e as ferramentas, os usuários presentes no ambiente poderão realizar as tarefas de sua responsabilidade. Por exemplo, João é responsável pela tarefa de distribuir eletrodomésticos. Assim, com os modelos de eletrodomésticos presentes na biblioteca e as ferramentas de manipulação, ele poderá concluir essa tarefa. Entretanto, ele deverá tomar cuidado com as recomendações do curso de Boas Práticas (não colocar uma

geladeira ao lado de um fogão, por exemplo). Ao concluir a tarefa, João deverá atualizar o estado da tarefa para “concluída” (Figura 6.11).

No final, quando todas as tarefas estiverem concluídas, Lia irá avaliar o trabalho realizado por seus companheiros, analisando e mostrando os erros cometidos por cada um deles. Em uma outra abordagem, ela avisa que o usuário cometeu erros, e dá uma chance para que ele os corrija, não prejudicando, assim, os outros usuários, já que as tarefas são dependentes.

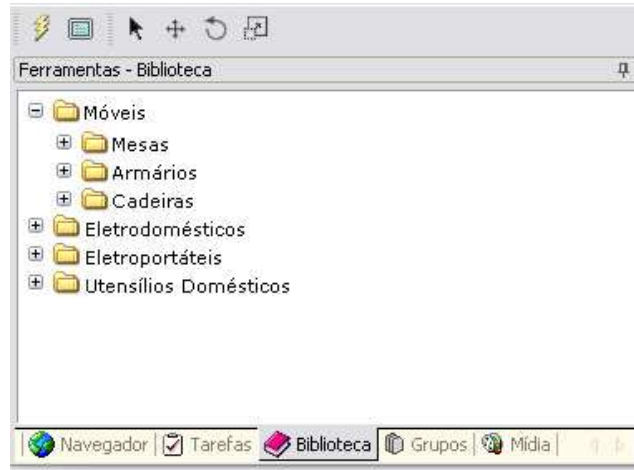


Figura 6.13. Biblioteca de objetos para o estudo de caso.

6.4.6 Protocolo PASSÁRGADa 2.0

Para o protocolo PASSÁRGADa 2.0 (descrito na Seção 4.6 Protocolo PASSÁRGADa 2.0) foram implementados dois componentes: um cliente e um servidor. O componente servidor é utilizado no GE e no RM, já o componente cliente é utilizado no CIERV. Esses componentes foram implementados com a Linguagem C++ utilizando a biblioteca de *sockets* ENet. O objetivo da ENet é propor uma camada de comunicação fina e simples sobre o protocolo UDP (*User Datagram Protocol*). O protocolo UDP não garante a chegada da mensagem, mas é mais rápido que o protocolo TCP. Para resolver esse problema, a ENet emprega um mecanismo que garante a chegada das mensagens sem perder a simplicidade do protocolo UDP. As PDUs implementadas para este estudo de caso são descritas na Tabela 22.

Tabela 22. Lista de PDUs implementadas para o protocolo PASSÁRGADa 2.0.

PDU	Descrição
<i>connect</i>	Inicia conexão com um servidor.
<i>requestConnect</i>	Solicita conexão com um mundo virtual específico
<i>disconnect</i>	Pede desconexão de um mundo virtual.
<i>reconnect</i>	Solicita reconexão com um servidor.

<i>accept</i>	Aceitação ou rejeição de conexão de um usuário a um mundo virtual.
<i>text</i>	Mensagem de texto.
<i>sound</i>	Mensagem de som.
<i>Url</i>	Mensagem de URL.
<i>pos</i>	Posição sem deslocamento.
<i>posDisplace</i>	Posição com deslocamento.
<i>posQ</i>	Posição sem deslocamento usando <i>quaternions</i> ²⁰ .
<i>posQDisplace</i>	Posição com deslocamento usando <i>quaternions</i> .
<i>scale</i>	Mudança de escala sem deslocamento.
<i>scaleDisplace</i>	Mudança de escala com deslocamento.
<i>move</i>	Movimentação de elementos.
<i>turn</i>	Rotação de elementos.
<i>action</i>	Mensagem de ação.
<i>status</i>	Mensagem de mudança de estado.
<i>inUser</i>	Entrada de usuário.
<i>exitUser</i>	Saída de usuário.
<i>inObj</i>	Inserção de objeto.
<i>outObj</i>	Remoção de objetos.
<i>group</i>	Formação de grupo.
<i>unGroup</i>	Mensagem para desfazer grupo.
<i>log</i>	Registro de ocorrência de componentes da CRAbCVE.
<i>permission</i>	Mensagem de permissão.
<i>publish</i>	Mensagem de publicação.
<i>gen</i>	Mensagem genérica.

6.5 Considerações Finais

Este capítulo apresentou o estado atual das implementações dos componentes da arquitetura CRAbCVE utilizando o modelo MAC. As implementações foram aplicadas a um estudo de caso relacionado ao projeto Sintegra – Sistema integrado de Educação a Distância com suporte a Avaliação de Competências. Apesar de o estudo de caso ser simples, envolvendo apenas seis usuários, e não ter todos os componentes e funcionalidades da arquitetura CRAbCVE implementados, ele foi importante para demonstrar o funcionamento e a integração dos componentes da arquitetura CRAbCVE.

Um aspecto interessante nas implementações foi a preocupação com a portabilidade dos componentes, sempre utilizando tecnologias que pudessem ser usados em diferentes plataformas. Além disso, essas tecnologias são de domínio público, o que reduz os custos da implementação. Apesar dessas vantagens, o uso dessas tecnologias causou uma consequência negativa: o aumento da complexidade e do tempo de desenvolvimento.

²⁰ *Quaternion* é um vector de três dimensões mais um escalar.

Outro aspecto interessante foi a facilidade de utilização do CIERV que possui uma interface amigável não necessitando que seus usuários tenham profundos conhecimentos de computação gráfica ou programação para criarem seus próprios conteúdos.

Capítulo 7

Conclusões e Trabalhos Futuros

Há aproximadamente uma década, a colaboração *desktop* tem recebido o suporte de computadores. Observou-se também, na última década, a utilização de técnicas de realidade virtual para o desenvolvimento de atividades colaborativas com imersão dos participantes em um ambiente 3D gerado por computador. Atualmente, vários usuários podem solucionar uma grande variedade de problemas colaborativamente em um ambiente de trabalho virtual compartilhado. Esses ambientes tornam a colaboração mais envolvente e mais bem sucedida do que outras tecnologias de colaboração, pois suportam um grau maior de interação social. Apesar dos bons resultados obtidos com esses ambientes, a maioria deles não dá suporte apropriado ao trabalho colaborativo, por falta de um mecanismo de controle das ações dos participantes. Essa falta de controle pode ser considerada a maior deficiência desses ambientes.

Desenvolver ambientes virtuais colaborativos que forneçam suporte apropriado ao trabalho colaborativo é uma tarefa complicada, pois é preciso seguir conceitos e recomendações de várias áreas de pesquisa: a de Realidade Virtual, a de Trabalho Colaborativo Assistido por Computador e a de Sistemas Computacionais Distribuídos. A RV preocupa-se com o fornecimento de condições adequadas para que os usuários experimentem sentimentos de imersão e presença dentro do ambiente virtual, e possam interagir entre si e com os elementos do ambiente. A CSCW define as condições que o ambiente deve prover para suporte adequado aos processos de comunicação, negociação, coordenação, compartilhamento, colaboração e percepção. Por último, a área de Sistemas Distribuídos preocupa-se com os elementos necessários para que o ambiente seja um sistema computacional robusto e escalável capaz de dar suporte o acesso de múltiplos usuários.

7.1 Principais contribuições

A principal contribuição desse trabalho é a definição de uma arquitetura de uso genérico, a CRAbCVE, projetada para viabilizar vários ambientes virtuais colaborativos (CVEs) distribuídos em uma rede de servidores na Internet. A CRAbCVE é uma evolução da arquitetura Ataxia (Leite-Junior, 2000) que inclui elementos para permitir o trabalho

colaborativo. Para o desenvolvimento da arquitetura CRAbCVE, as funcionalidades dos elementos da Ataxia foram redefinidas, um novo componente foi adicionado e um novo protocolo de comunicação foi especificado.

Cada CVE disponibilizado pela CRAbCVE é formado por um conjunto de componentes especializados, os quais podem ser executados simultaneamente em diversas máquinas. Essa é uma alternativa de baixo custo aos atuais sistemas de suporte a ambientes virtuais em rede, permitindo a criação de uma estrutura distribuída, com uma grande capacidade de expansão.

Outra importante contribuição é a definição do modelo MAC, que visa auxiliar o emprego da arquitetura CRAbCVE no trabalho colaborativo. Para isso, o MAC especifica uma ferramenta de autoria, políticas de acesso e mecanismos de coordenação, percepção e comunicação para possibilitar eficiência no trabalho colaborativo. Por conjugarem várias das funcionalidades necessárias a ambientes de realidade virtual que apóiam o trabalho colaborativo, esses modelos podem servir de base para desenvolvedores que desejam implantar autoria colaborativa em ambientes virtuais.

A identificação dos processos envolvidos na colaboração assistida por computador e suas taxonomias, como também o levantamento dos requisitos e aspectos de *software* e *hardware* de realidade virtual distribuída para desenvolvimento de um CVE também constituem contribuições substanciais deste trabalho.

Por último, o protótipo desenvolvido como estudo de caso deste trabalho pode ser utilizado como ferramenta auxiliar em aplicações educacionais.

7.2 Trabalhos futuros

O uso indiscriminado de fábricas para viabilizar a instanciação remota dos componentes de CVEs pode causar sobrecarga nos servidores hospedeiros. Uma solução para esse problema seria a utilização de um sistema de gerenciamento de recursos da CRAbCVE. Esse sistema seria responsável pelo gerenciamento de servidores, instanciação transparente de CVEs, balanceamento de cargas, recuperação de estado de falha e segurança no acesso aos componentes.

O gerenciamento de servidores se resume ao controle dos servidores hospedeiros disponíveis para instanciação de CVEs. A instanciação transparente refere-se à alocação de CVEs no servidor com maior quantidade de recursos livres. O balanceamento de cargas consiste na transferência de instâncias de CVEs de um servidor sobrecarregado para outro. A recuperação de estado de falha refere-se à detecção, ao registro e à solução de problemas

nos componentes de um CVE. A segurança no acesso aos componentes estabelece técnicas de codificação na comunicação entre os componentes da CRAbCVE.

A especificação de um modelo de interação mais sofisticado que utilize técnicas de inteligência artificial para tratar os comportamentos dos objetos presentes no ambiente virtual seria bastante interessante. Esse modelo poderá utilizar, para tomadas de decisão, máquinas de estado ou até mesmo redes neurais. Além disso, o modelo poderá também especificar mecanismos de consciência para determinar o nível de percepção que uma entidade tem de outra.

Para possibilitar uma maior quantidade de usuários em um mesmo ambiente, poderão ser utilizadas técnicas de carga dinâmica da estrutura tridimensional do ambiente e de técnicas de partição do ambiente. A carga dinâmica facilitará a utilização de grandes mundos virtuais. A utilização desses mundos poderá causar sobrecarga nos servidores pela quantidade de usuários conectados, assim, a partição do ambiente poderá resolver esse problema, dividindo-o em setores, cada um dos quais poderá ser tratado por um conjunto de componentes da arquitetura CRAbCVE.

Por último, seria interessante realizar um estudo estatístico sobre a capacidade de usuários que cada CVE suporta em tipos diferentes de redes de computadores. Por exemplo:

- ♦ Qual a quantidade média de usuários que é suportada com uma conexão de 128kbs realizando autoria em mundos virtuais?
- ♦ Quantos CVEs são suportados para um mesmo FM para uma largura de banda específica?
- ♦ Qual a média de PDUs geradas em uma interação simples com N usuários?

A resposta a essas questões iria auxiliar a implantação da arquitetura para uma aplicação específica, levando-se em conta a quantidade e o tipo de recursos disponíveis.

Referências Bibliográficas

- ActiveWorld (2004). *Virtual Reality experience*. Disponível em: <<http://www.activeworld.com>>. Acesso em 07 dez. 2005.
- Allison, D.; Wills, B.; Hodges, L. F. & Wineman, J. (1997). *Gorillas in the Bits*. In Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'97), IEEE Computer Society Press, California, EUA, p. 69–77.
- Almendra, C. C. (2003). *Administração Remota de Ambientes Virtuais em Rede para Integração com Sistemas de Gerenciamento de Aprendizado*. Dissertação de Mestrado, UFC, Fortaleza.
- Andrade, A. (1999). *Uma Proposta Metodológica Para Criação de Roteiros em Ambientes Virtuais Para Aplicação Educacional*. Dissertação de Mestrado, UFSC, Florianópolis.
- Applegate, L. M.; Konsynski, B. R. & Nunamaker, J. F. (1986). *A group decision support system for idea generation and issue analysis in organization planning*. In Proceedings of the First Conference on Computer Supported Cooperative Work, ACM Press, Nova York, EUA, p. 16-34.
- Araujo, R. C. (2000). *Arquitetura para Desenvolvimento de Aplicações de Educação a Distância*. Dissertação de mestrado, UFPE, Recife.
- Araújo, R. M.; Dias, M. S. & Borges, M. R. S. (1997). *Suporte por Computador ao Desenvolvimento Cooperativo de Software: Classificação e Propostas*. Anais do XI Simpósio Brasileiro de Engenharia de Software, Fortaleza, p. 299-314.
- Balen, H. (2000). *Distributed Object Architectures with CORBA*. Nova York: SIGS Publications, EUA.
- Barros, L. A. (1994). *Suporte a ambientes Distribuídos para Aprendizagem Cooperativa*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro.
- Benford, S. D. & Greenhalgh, C. (2001). *Collaborative Virtual Environments*. In Proceedings of the Communications of the ACM, vol. 44, no. 7, ACM Press, Nova York, EUA, p. 79.

- Benford, S.; Bowers, J.; Fahlen, L.; Mariani, J. & Rodden, T. (1994) *Supporting Cooperative Work in Virtual Environments*. In Proceedings of the Computer Journal, Oxford University Press, vol. 37, no. 8, Londres, Inglaterra, p. 653-668.
- Blaxxun (2004). *Blaxxun Contact*. Disponível em: <<http://www.blaxxun.com>>. Acesso em 14 dez. 2004.
- Borges, M.; Campos, M. L. & Cavalcanti, M. C. R. (1995). *Suporte por computador ao trabalho cooperativo*. Anais do XV Congresso Brasileiro da Sociedade Brasileira de Computação, XIV Jornada de Atualização em Informática, Canela, p. 45.
- Borges, V. M. C. (2001). *Proposta Metodológica para o Ensino de Línguas Estrangeiras no Projeto AVAL - O Emprego da Abordagem com Base em Tarefas*. Anais do VII Workshop de Informática na Escola, Fortaleza.
- Borghoff, U. M. & Schlichter, J. H. (2000). *Computer-Supported Cooperative Work: Introduction to Distributed Applications*, 1ª ed, Nova York: Springer, EUA.
- Boulic, R.; Rezzonico, S. & Thalmann, D. (1996). *Multi-Finger Manipulation of Virtual Objects*. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '96), Hong Kong, China, p. 67-74.
- Box, D.; Ehnebuske, D.; Kakivaya, G.; Layman, A.; Mendelson, N.; Nielsen, H. F.; Thatte, S. & Winner, D. (2000). *Simple Object Access Protocol (SOAP) 1.1*. World Wide Web Consortium (W3C). Disponível em: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>. Acesso em 21 ago. 2005.
- Brutzman, D. (1998). *Graphics Internetworking: Bottlenecks and Breakthroughs*. Proceedings of the Digital Illusion: Entertaining the Future with High Technology, ACM Press, Nova York, EUA, p. 61-95.
- Burgoon, M. & Ruffner, M. (1978). *Human Communication*. Nova York: Holt, Rinehart and Winston, EUA.
- Çapin, T. K.; Pandzic, I. S.; Thalmann, N. M. & Thalmann, D. (1999). *Avatars in Networked Virtual Environments*. Chichester: John Wiley & Sons, Inglaterra.
- Cardoso, F.; Rocha, F.; Lima, L.; Costa, M.; Souza T.; Ferreira, L.; Cardoso, A. & Lamounier-Junior, E. (2003). *Jogo: Fogo Amigo - Uso de Técnicas de Realidade Virtual no Desenvolvimento de Jogos Interativos*. In Proceedings of the 6th Symposium on Virtual Reality, Ribeirão Preto, p. 431.

- Conklin, J. & Begeman, M. L. (1987). *gIBIS: A Hypertext Tool for Team Design Deliberation*. In Proceeding of the ACM conference on Hypertext, Carolina do Norte, EUA, p. 247 - 251.
- CORBA (2005). *Common Object Request Broker*. Disponível em: <[http:// www.omg.org](http://www.omg.org)>. Acesso em 08 jun. 2005.
- Crowston, K. & Malone, T. W. (1988). *Intelligent Software Agents*. In Byte Magazine, vol. 13, McGraw-Hill Inc., Hightstown, NJ, EUA p. 267-272.
- Cruz-Neira, C.; Sandin, D. J.; DeFanti, T. A.; Kenyon, R. V. & Hart, J. C. (1992). *The CAVE Audio Visual Experience Automatic Virtual Environment*. In Proceeding of the Communication of the ACM, vol. 35, no. 6, ACM Press, Nova York, EUA p. 64-72.
- Csordas, T. J. (2000). *Computerized Cadavers: Shades of Being and Representation in Virtual Reality*. Case Western Reserve University. Disponível em: <<http://www.focusing.org/compuCAD.html>>. Acesso em 05 jan. 2001.
- CVE (2005). *Collaborative Virtual Environments*. Disponível em: <<http://www.crg.cs.nott.ac.uk/~sdb/CVEs.html>>. Acesso em 02 fev. 2005.
- Das, T. K.; Singh, G.; Mitchell, A.; Kumar, P. S. & McGee, K. (1997a). *Developing Social Virtual Worlds using NetEffect*. In Proceedings of the 6th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises, IEEE Computer Society Washington, EUA, p. 148-154.
- Das, T. K.; Singh, G.; Mitchell, A.; Kumar, P. S. & McGee, K. (1997b). *NetEffect: A Network Architecture for Large-scale Multi-user Virtual Worlds*. In Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST '97), Lausanne, Suíça, p. 157-163.
- DIVE (2004). *Distributed Interactive Virtual Environment*. Disponível em: <<http://www.sics.se/dive/>>. Acesso em 15 dez. 2004.
- Dourish, P. & Bellotti, V. (1992). *Awareness and coordination in shared workspace*. In Proceedings of 4th Conference on Computer Supported Cooperative Work, Chapel Hill, EUA, p.107-114.
- Dourish, P. (1997). *Extending awareness beyond synchronous collaboration*. In Proceedings of Conference on human factors in computing systems (CHI'97),

- Workshop on Awareness in Collaboration Systems. Atlanta, EUA. Disponível em <<http://www.best.com/~jpd/chi97-awareness.html>>. Acesso em 07 jul. 1999.
- Eduardo, V. (2001). *Protótipo de um ambiente virtual distribuído multiusuário*. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação), Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- Ellis, C. A.; Gibbs, S. J. & Rein, G. L. (1991). *Groupware: Some issues and experiences*. In Proceedings of the Communication of the ACM, v.34, no.1, ACM Press, Nova York, EUA, p.1-29.
- Ellis, C. A. & Wainer, J. (1994). *A Conceptual Model of Groupware*. In Proceedings of the Conference on Computer Supported Cooperative Work, ACM Press, Chapel Hill, EUA, 79-88.
- Emmerich, W. (2000). *Engineering Distributed Objects*. Chichester: Wiley.
- ENet (2005). *UDP oprn library*. Disponível em <<http://enet.bespin.org/>>. Acesso em 17 ago. 2005.
- Frécon, E. & Stenius, M. (1998). *DIVE: A scaleable network architecture for distributed virtual environments*. In Proceedings of the Distributed Systems Engineering Journal (DSEJ), Special Issue on Distributed Virtual Environments, vol. 5, no. 3, p. 91-100.
- Freitas, C. M. D. S.; Manssour, I. H.; Nedel, L. P.; Gavião, J. K.; Paim, T. C. & Maciel, A. (2003). *Framework para Construção de Pacientes Virtuais: Uma Aplicação em Laparoscopia*. In Proceedings of the 6th Symposium on Virtual Reality, Ribeirão Preto, p. 117-128.
- Fuks, H.; Raposo, A. B. & Gerosa, M. A. (2002). *Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas*. Anais do XXII Congresso da Sociedade Brasileira de Computação, Jornada de Atualização em Informática, vol. 2, cap. 3, Florianópolis, p. 89-128.
- Funkhouser, T. A. (1995). *RING: A Client-Server System for Multi-User Virtual Environments*. In Proceedings of Symposium on Interactive 3D Graphics, ACM SIGGRAPH Special Issue, Symposium, Monterey, EUA, p. 85-92.
- Funkhouser, T. A. (1996). *Network Topologies for Scaleable Multi-User Virtual Environments*. In Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'96), Santa Clara, EUA, p. 222-229.

- Gamma, E.; Helm, R.; Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Software*. Nova York: Addison Wesley, USA.
- Gossweiler, R.; Laferriere, R. J.; Keller, M. L. & Pausch, R. (1994). *An Introductory Tutorial for Developing Multi-User Virtual Environments*. In the Presence: Teleoperators and Virtual Environments, vol.3, no.4, p.255-264.
- Greenhalgh, C. (1997). *Large Scale Collaborative Virtual Environments*. Tese de Doutorado, Computer Science Department, University of Nottingham, Nottingham, UK.
- Greenhalgh, C. M. & Benford, S. D. (1995). *Massive: a collaborative virtual environment for teleconferencing*. In Proceedings of the of ACM Transactions on Computer-Human Interactions (TOCHI'95), vol. 2, no. 3, New York, EUA, p. 239-261.
- Greenhalgh, C. M.; Benford, S. D.; Taylor, I.M.; Bowers, J. M.; Walker, G. & Wyver, J. (1999). *Creating a live broadcast from a virtual environment*. In Proceedings of the SIGGRAPH'99, Los Angeles, EUA, p. 375-384.
- Grudin, J. (1994). *Computer-Supported Cooperative Work: History and Focus*. In IEEE Computer Magazine, vol. 27, no. 5, IEEE Computer Society Press, p. 19-26.
- gSOAP (2005). *Generator Tools for Coding SOAP/XML Web Services in C and C++*. Disponível em <<http://www.cs.fsu.edu/~engelen/soap.html>>. Acesso em 27 ago 2005.
- Gutmann, S.; Burgard, W.; Fox, D. & Konolige, K. (1998). *An experimental comparison of localization methods*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98), vol. 2, Victoria, Canada, p. 736 - 743.
- Gutwin, C. & Greenberg, S. (1999). *A framework of awareness for small groups in shared-workspace groupware*. Technical Report 99-1, Department of Computer Science, University of Saskatchewan, Canadá. Disponível em <<http://www.cpcs.ucalgary.ca/grouplab/papers/1999/99-AwarenessTheory/html/theory-tr99-1.html>>. Acesso em 07 set 1999.
- Haake, J. M. & Wilson, B. (1992). *Supporting Collaborative writing of hyperdocuments in SEPIA*. In Proceedings of 4th ACM Conference on Conference on Computer-Supported Cooperative Work (CSCW'92), ACM Press, Toronto, Canada, p. 138-146..
- Hangsand, O. (1996). *Interactive Multiuser Ves in the DIVE System*. In IEEE Multimedia Magazine, IEEE Computer Society, vol. 3, no. 1, p. 30-39.

- IEEE (1995). *IEEE Standard for Distributed Interactiv Simulation – Application Protocols*. In Proceedings of the IEEE Std 1278.1–1995. Piscataway: IEEE Standard Press.
- ISO/IEC (1991). *Basic Reference Model of Open Distributed Processing*. International Organization for Standardization and International Electrotechnical Committee, Available through national standards bodies.
- Johnson, A.; Roussos, M.; Leigh, J.; Barnes, C.; Vasilakis, C. & Moher, T. (1998). *The NICE Project: Learning Together in a Virtual World*. In Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'98), Atlanta, EUA, p. 176-183.
- Joslin, C.; Molet, T.; Magnenat-Thalmann, N.; Esmerado, J.; Thalmann, D.; Palmer, I.; Chilton, N. & Earnshaw, R. (2001). *Sharing Attractions on the Net with VPark*. In Proceedings of the IEEE Computer Graphics and Applications, vol. 21, no. 1, Genebra, Suíça, p. 61-71.
- Kaufmann, H.; Schmalstieg, D. & Wagner, M. (2000). *Construct3D: A Virtual Reality Application for Mathematics and Geometry Education*. In Proceedings of the Education and Information Technologies, vol.5 no.4, p.263-276.
- Kawamoto, A. L. S.; Cantão, J.; Pinto, A.; Aquino, P. T.; Álvares, J. V.; Kirner, T. G.; Kirner, C. & Wazlawick, R. S. (2001). *AVC-MV: Um Ambiente Colaborativo para Aplicações Educacionais*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 226-237.
- Kraemer, K. L. & King, J. L. (1988). *Computer-based systems for cooperative work and group decision making*. In Proceedings of ACM Computing Surveys, vol. 20, no.2, p. 115-146.
- Lee, J. S. (1990). *A tool for managing group decision rationale*. In Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90), Los Angeles, EUA, p.79-92.
- Leite-Junior, A. J. M. (2000). *Ataxia: Uma Arquitetura para a Viabilização de NVEs voltados para a educação a distância através da Internet*. Dissertação de Mestrado, UFC, Fortaleza.

- 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*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 258-269.
- 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*. In Proceedings of the 5th Symposium on Virtual Reality, Fortaleza, p. 138-149.
- Lewis, B. T. & Hodges, J. D. (1988). *Shared books: collaborative publication management for an office information system*. In Proceedings of the Conference on Office Information Systems, Association for Computing Machinery, New York, EUA, Association for Computing Machinery, p.197-204.
- Lewis, M. & Jacobson, J. (2002). *Introduction (Game engines in scientific research special issue)*. In Proceedings of the Communications of the ACM, ACM Press, vol. 45, no. 1, New York, EUA, p. 27-31.
- Locke, J. (1994). *An introduction to the internet networking environment and SIMNET/DIS*. Computer Science Department, Naval Postgraduate School, Monterey, EUA. Disponível em: <<http://web.nps.navy.mil/~code09/techreports.html>>. Acesso em 07 fev. 2001.
- Lubich, H. P. (1995). *Towards a CSCW Framework for Scientific Cooperation in Europe*. In Lecture Notes in Computer Science, vol. 889, Berlin: Springer-Verlag, Alemanha.
- Macedo, A. A. (1999). *Explorando Tecnologias Hipermídia e de Trabalho Cooperativo em um Ambiente de Apoio ao Ensino*. Dissertação de Mestrado, USP, São Paulo (Capital).
- Macedonia, M. R. & Zyda, M. (1997). *A Taxonomy for Networked Virtual Environments*. In IEEE MultiMedia Magazine, IEEE Computer Society, vol. 4, no.1, p. 48-56.
- Macedonia, M. R.; Brutzman, D. P.; Zyda, M. J.; Pratt, D. R.; Barham, P. T.; Falby, J. & Locke, J. (1995). *NPSNET: A Multi-Player 3D Virtual Environment Over the Internet*. In Proceedings of the Symposium on Interactive 3D Graphics, ACM SIGGRAPH, Monterey, EUA, p.93-94.

- Maia, J. G. R.; Cavalcante-Neto, J. B. & Vidal, C. A. (2003). *CRAbGE: Um Motor Gráfico Customizável, Expansível e Portável Para Aplicações de Realidade Virtual*. In Proceedings of the 6th Symposium on Virtual Reality, Ribeirão Preto, p. 03-14.
- Makrakis, D.; Hafid, A.; Nait-Abdesselem, Kasiolas, F. A. & Qin, L. (1998). *Quality of Service Management in Distributed Interactive Virtual Environment*. Progress Report of DIVE project. Disponível em <http://www.mcrlab.uottawa.ca/research/QoS_DIVE_Report.html>. Acesso em 21 ago. 2005.
- Malone, T. W. & Crowston, K. (1990). *What Is Coordination Theory and How Can It Help Design Cooperative Work Systems?* In Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'90), Los Angeles, EUA, p. 371-380.
- Mania, K. & Chalmers, A. (1998). *A Classification for User Embodiment in Collaborative Virtual Environments*. In Proceedings of the 4th International Conference on Virtual Systems and Multimedia, IOS Press - Ohmsha, Ltd., Bristol, Inglaterra, p. 177-182..
- Miles, V. C.; McCarthy, J. C.; Dix, A. J.; Harrison, M. D. & Monk, A. F. (1993). *Reviewing Designs for a Synchronous-Asynchronous Group Editing Environment*. Mike Sharples (editor), Computer Supported Collaborative Writing, Londres: Springer-Verlag, Inglaterra, p. 137-160.
- Moura-Filho, C. O. & Oliveira, A. M. (1998). *Videoconferência em educação a distância*. Fortaleza: Editora CEFET-CE.
- Neisser, U. (1976). *Cognition and Reality*. San Francisco: W.H. Freeman, EUA.
- NetMeeting (2005). Disponível em <<http://www.microsoft.com/windows/netmeeting/>>. Acesso em 14 ago. 2005.
- NPSNET (2004). Disponível em <<http://movesinstitute.org/~npsnet/>>. Acesso em 07 dez. de 2004.
- Nunamaker, J. F.; Dennis, A. R.; Valacich, J. S.; Vogel, D. & George, J. F. (1991). *Electronic Meeting Systems to Support Group Work*. In Proceedings of the Communications of the ACM, vol. 34, no. 7, ACM Press, New York, p. 40-61.
- Oliveira, J. C. & Georganas, N. D. (2003). *Velvet: An adaptive hybrid architecture for very large virtual environments*. In Presence Teleoperators And Virtual Environments, vol. 12, no. 6, p. 555-580.

- Ozgate (2004). *Ozgate International*. Disponível em <<http://www.Ozgate.com>>. Acesso em 27 dez. 2004.
- Pandzic, I. S.; Çapin, T. K.; Magnenat-Thalmann, N. & Thalmann, D. (1996). *Motor Functions in the VLNET Body-Centered Networked Virtual Environment*. In Proceedings of the Eurographics Workshops: Virtual Environments and Scientific Visualization '96, Monte Carlo, Monaco, p. 94-103.
- Panitz, T. (1996). *A definition of collaborative vs cooperative learning*. Disponível em <<http://www.lgu.ac.uk/deliberations/collab.learning/panitz2.html>>. Acesso em 07 nov. de 2004.
- Pequeno, M. C. (1998). *Infra-Estrutura Física e Tecnológica para Rede de Videoconferência do Estado do Ceará*. Projeto submetido à FINEP, Fortaleza.
- Pequeno, M. C.; Loureiro, R. C.; Sarmiento, W. W. F.; Pinho, G.; Pequeno, H. S. L.; Paula, P. S.; Silva, C. O. *Videoconferência: A Experiência da Formação de Gestores de Novas Tecnologias da Informação e Comunicação*. Anais do IX Workshop de Informática na Escola, XXIII Congresso da Sociedade Brasileira de Computação, Campinas, p. 40-59.
- Pinheiro, M. K.; Lima, J. V. & Borges, M. R. S. (2002). *A Framework for Awareness Support in Groupware Systems*. In Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design. CSCWD'2002, Rio de Janeiro, p. 13-18.
- Poupyrev, I. & Ichikawa, T. (1999). *Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques*. In Journal of Visual Languages and Computing, Academic Press, vol. 10, no.1, p. 19-35.
- R. S. Wazlawick; L. C. Fagundes & T. G. Kirner (1999). *Museu Virtual: Ferramenta de Autoria para Criação de Museus em Realidade Virtual para Apoiar a Aprendizagem Colaborativa via Internet*. Projeto ProTem-CPNq PTI/PEDU.
- Ragnarok (2004). *Ragnarok Online*. Disponível em <<http://www.Ragnarok.com.br>>. Acesso em 28 dez. de 2004.
- Raposo, A. B (2001). *A Component-based infrastructure for the coordination of collaborative activities in virtual environments*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 127-138.

- Raposo, A. B.; Magalhães, L. P. & Ricarte, I. L. M (2000). *Mecanismos de Coordenação para Ambientes Colaborativos*. Anais do VI Simpósio Brasileiro de Sistemas Multimídia e Hipermídia (SBMIDIA'00), Natal, p. 247-258.
- Raposo, A. B. & Fuks, H. (2002). *Defining Task Interdependencies and Coordination Mechanisms for Collaborative Systems*. In Proceedings of the Cooperative Systems Design (Frontiers in Artificial Intelligence and Applications, vol. 74, IOS Press, Amsterdam, p. 88-103.
- Raposo, A. B.; Magalhães, L. P. & Ricarte, I. L. M. (2000). *Petri Nets Based Coordination Mechanisms for Multi-Workflow Environments*. In International Journal of Computer Systems Science & Engineering, vol. 15, no.5, p. 315-326.
- Raposo, A. B.; Magalhães, L. P.; Ricarte, I. L. M. & Fuks, H. (2001). *Coordination of collaborative activities: A framework for the definition of tasks interdependencies*. In Proceedings of the 7th International Workshop on Groupware (CRIWG 2001), Darmstadt, Alemanha. p. 170-179.
- Reinhard, W.; Schweitzer, J. & Völksen, G. (1994). *CSCW Tools: Concepts and Architectures*. In IEEE Computer Magazine, vol.27, no. 5, p.28-36.
- Rezende, J. L. (2003). *Aplicando Técnicas de Comunicação para a Facilitação de Debates no Ambiente AulaNet*. Dissertação de Mestrado, PUC-Rio, Rio de Janeiro.
- Robinett, W. (1998). *Interactivity and Individual Viewpoint in Shared Virtual Worlds: the Big Screen Versus Networked Personal Displays*. In Dodsworth, C. (Editor). Digital Illusion: Entertaining the Future with High Technology, New York: ACM Press, EUA, p. 331-342.
- Rodello, I. A., Kubo, M. M., Kirner, C. & Tori, R. (2001). *Análise dos Principais Fatores para o Desenvolvimento de Sistemas Distribuídos de Realidade Virtual*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 314- 323.
- Rosa-Junior, O. (2001). *Ambientes Virtuais Cooperativos LRVCHAT3D, Um Estudo de Caso*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 148-159.
- Santos, A. (1993). *Cooperative Hypermedia editing with CoMediA*. In Journal of Computer Science & Technology, vol.8, no.2, p. 257-269.

- Santos, E. M. (2001). *VEGA – um modelo de integração de aplicações externas a ambientes virtuais através do uso de aplicações mediadoras e de simulóides*. Dissertação de Mestrado, UFC, Fortaleza.
- Santos, E. M.; Vidal, C. A.; Leite-Júnior, A. J. M. & Almendra, C. C. (2001 B). *Beremiz: Integrando o Mathematica a um ambiente virtual em rede*. In Proceedings of the 4th SBC Symposium on Virtual Reality, Florianópolis, p. 238-248.
- Sarin, S. (1985). *Computer-Based real time Conferencing Systems*. In IEEE Computer Magazine, vol. 7, no.10, New York, EUA, p. 33-45.
- Schlichter, J.; Koch, M. & M. Bürger (1997). *Workspace Awareness for Distributed Teams*. In Lecture Notes on Computer Science, vol. 1364, Berlin: Springer-Verlag, Alemanha, p. 199-219.
- Schrage, M. (1995). *No more teams! Mastering the dynamics of creative collaboration*. New York: Currency Doubleday, EUA.
- Schulzrinne, H. & Casner, S. (2003). *RTP Profile for Audio and Video Conferences with Minimal Control*. In Technical Report, Request for Comments 1890 (RFC 1890), Internet Engineering Task Force.
- Schulzrinne, H.; Casner, S.; Frederick, R. & Jacobson, V. (2003). *RTP: A Transport Protocol for Real-Time Applications*. In Technical Report, Request for Comments 3550 (RFC 3550), Internet Engineering Task Force.
- Shaw, C. & Green, M. (1993). *The MR toolkit peers package and experiment*. In Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS 93), Washington, EUA, p. 463-469.
- Shirmohammadi, S. & Georganas, N. D. (2001). *An End-to-End Communication Architecture for Collaborative Virtual Environments*. In Computer Networks Journal, vol. 35, no. 2, p. 351-367.
- Shum, S. & Hammond, N. (1994). *Argumentation-Based design rationale: what use at what cost?* In International Journal of Human-Computer Studies, vol. 40, no. 4, p. 603-652.
- Silva, R. J. M.; Wagner, G. N.; Raposo, A. B. e Gattas, M. (2003). *Experiência de Portais em Ambientes Arquitetônicos Virtuais*. In Proceedings of the 6th Symposium on Virtual Reality, Ribeirão Preto, p. 117-128.

- SIMNET (2005). *Simulator Networking*. Disponível em <http://www.sisostds.org/webletter/siso/iss_39/art_202.htm>. Acesso em 07 jun. 2005.
- Singhal, S. & Zyda, M. (1999). *Networked Virtual Environments*. New York: Addison-Wesley, ACM Press, EUA.
- Sintegra (2005). *Sistema integrado de Educação a Distância com suporte a Avaliação de Competências*. Disponível em <<http://www.vdl.ufc.br/sintegra>>. Acesso em 21 ago. 2005.
- Slater, M.; Ush, M. & Steed, A. (1994). *Depth of Presence in Virtual Environments*. In Proceedings of Presence: Teleoperators and Virtual Environments, vol. 3, no. 2, p. 130-144.
- Smith, G. (1996). *Co-operative virtual environments: Lessons from 2D multiuser interfaces*. In Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'96), Boston, EUA, p. 390-398.
- Sohlenkamp, M. (1998). *Supporting group awareness in multi-user environment through perceptualization*. Dissertação de Mestrado. Fachbereich Mathematik-Informatik der Universität, Gesamthochschule, Paderborn, Alemanha.
- SOLAR (2005). *Sistema Online de Aprendizagem*. Disponível em <<http://www.vdl.ufc.br/solar>>. Acesso em 07 ago. 2005.
- Souza, C. R. B.; Wainer, J. & Rubira, C. M. F (1997). *Um Modelo de Anotações para o Desenvolvimento Cooperativo de Software*. In Proceedings of the Workshop on Hipermedia and Multimedia (WOHM'97), São Carlos. p. 143-154.
- Stallings, W. (1998). *High-Speed Networks TCP/IP and ATM Design Principles*. Nova Jersey: Prentice Hall, EUA.
- Stephenson, N. (1993). *Snowcrash*. Londres: ROC, Inglaterra.
- Stytz, M. R. (1996). *Distributed Virtual Environments*. In Proceedings of the IEEE Computer Graphics and Applications, vol. 16, no. 33, p. 19-31.
- Tanenbaum, A. S. (1995). *Distributed Operating Systems*. Nova Jersey: Prentice Hall, EUA.
- Tanenbaum, A. S. (1996). *Computer Networks*. Nova Jersey: Prentice Hall, 3ª edição, EUA.

- Tanenbaum, A. S. & Woodhull, A. S. (1997). *Operational Systems - Design and Implementation*. Nova Jersey: Prentice Hall, 2ª edição, EUA.
- Teufel, S. & Teufel, B (1995). *Bridging Information Technology and Business — Some Modelling Aspects*. In Proceedings of the the ACM SIGOIS Bulletin, vol. 16, no.1, ACM Press, Nova York, EUA, p. 13–17.
- Thalmann, D. (2001). *The Role of Virtual Humans in Virtual Environment Technology and Interfaces*. In Frontiers of Human-Centred Computing, Online Communities and Virtual Environments. Londres: Springer-Verlag, Inglaterra, p. 27-38.
- TinyXML (2005). Disponível em <<http://www.grinninglizard.com/tinyxml/>>. Acesso em 17 ago. de 2005.
- Universal Time (2000). *UniversalTime.org: Promoting a metric based system*. Disponível em <<http://www.ucaid.eu/ucaid>>. Acesso em 07 ago. 2005.
- Usenet (2004). *Newsgroup*. Disponível em <<http://www.usenetbinaries.com>>. Acesso em 17 nov. 2004.
- Valle-Filho, A. M.; Souza, P.; Alves, J. B. M., Wazlawick, R. S. e Luz, R. P. (2000). *Ferramentas de Autoria de Realidade Virtual - Um Estudo Comparativo*. Anais do Congresso Internacional de Ingeniería Informática, Buenos Aires, Argentina, p. 47-65.
- Vicentin, V. J., Kubo, M. M., Dizeró, W. J., Kirner, C. (1999). *Sistemas Cooperativos de Realidade Virtual Usando Java e VRML*. Anais do XXV Congresso Latino-Americano de Informática (CLEI'99), Assunção, Paraguai. p. 1085-1096.
- Vidal, C. A.; Borges, V. M. C. & Barros Filho, E. M. (2001a). *Virtual Environments for Language Learning*. In Proceedings of the Projects Evaluation Workshop in Informatics in Education, ProTeM-CC, Rio de Janeiro, p. 196-213.
- Vidal, C. A.; Leite Jr., A. J. M.; Almendra, C. C.; Santos, E. M. & Oliveira, J. P. C. (2000). *Uma Proposta de Integração de Ferramentas Externas a Ambientes Virtuais através de Simulóides e Aplicações Mediadoras*. Anais do 3º Workshop on Virtual Reality, Gramado, p. 131-142.
- Vidal, C. A.; Santos, E. M.; Leite-Júnior, A. J. M.; Almendra, C. C. & Borges, V. M. C. (2003). *Collaborative Virtual Environments for Language Learning*. In Proceedings of the 6th Symposium on Virtual Reality, Ribeirão Preto, p. 309-320.

- Vidal, C. A.; Leite Júnior, A. J. M.; Almendra, C. C.; Santos, E. M.; Oliveira, J. P. C. & Costa-Filho, J. C. S. (2001b). *A Virtual Environment for Distance Learning*. In Proceedings of the 7th World Conference on Computers in Education – Networking the Learner, Copenhagen: Kluwer Academic Publishers and IFIP, Dinamarca, p. 95-103.
- Vinoski, S. (1998). *New Features for CORBA 3.0*. In Proceedings of the Communications of the ACM, vol. 41, no. 10, ACM Press, New York, p.44-52.
- W3C (2005). *World Wide Web Consortium*. Disponível em <<http://www.w3.org/XML/>> Acesso em 17 ago. 2005.
- Wikipedia (2005). Disponível em <<http://www.wikipedia.org>>. Acess 23 ago. 2005.
- Wilcox S. K. (1998). *Web Developer's Guide to 3D Avatars*. Nova York: John Wiley and Sons Inc, EUA.
- wxWidgets (2005). Disponível em <<http://www.wxwidgets.org/>>. Acesso em 07 ago. 2005.
- Xu, C. (2000) *Interaction and Collaboration mechanisms for Distributed Communities and Groups in Educational Settings*, PhD thesis. Institut fur Informatik der Technischen Universitat Munchen, Munique, Alemanha.
- Zyda, M. J. (1996), *Networked Virtual Environments*. California.
- Zyda, M. J.(1995). *VRAIS Panel on Networked Virtual Environments*. In Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'95), Carolina do Norte, EUA, p. 67-77.

Anexos

Anexo A – Exemplo de arquivo descritor de mundo virtual

```
<!-- Arquivo descritor (XML) do mundo virtual da cozinha utilizado no estudo de caso
desse trabalho -->

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<world id="1" name="cozinha.zip" version="1">

  <media id="17" name="cozinha.3ds"/>

  <!-- Árvore com os objetos inseridos no mundo. Neste arquivo existem dois: um
objeto composto (uma mesa) e um simples. Este arquivo não descreve os objetos,
apenas faz referência a eles -->

  <objects>

    <!-- O objeto 23 usa a mídia 14 descrita em outro arquivo xml, 14.xml -->
    <object localId="23" name="mesa-pequena" status="0" type="1" media="14">

      <pos x="1.340" y="1.450" z="0.000" /> <!-- Posição -->

      <orient x="1.470" y="-1.100" z="1.210" w="1.000" /> <!-- Orientação -->

      <scale x="2.000" y="2.000" z="2.000" /> <!-- Escala -->

      <speed x="0.000" y="0.000" z="0.000" /> <!-- Velocidade -->

      <angle x="1.000" y="0.000" z="0.000" w="1.000" /> <!-- Ângulo -->

      <!-- Lista de objetos que compõe a mesa (duas cadeiras) -->

      <children>

        <!-- Objeto cadeira descrito no arquivo 12.xml -->

        <object localId="35" name="cadeira01" status="0" type="1" media="12">

          <pos x="-12.340" y="-11.450" z="42.470" />

          <orient x="-1.760" y="1.530" z="1.021" w="1.000" />

          <scale x="2.000" y="2.000" z="2.000" />

          <speed x="0.000" y="0.000" z="0.000" />

          <angle x="-1.000" y="0.000" z="0.000" w="1.000" />

        </object>

        <!-- Objeto cadeira descrito no arquivo 12.xml -->

        <object localId="36" name="cadeira02" status="0" type="1" media="12">

          <pos x="-21.240" y="11.000" z="37.470" />

          <orient x="1.760" y="-1.530" z="1.021" w="1.000" />

          <scale x="2.000" y="2.000" z="2.000" />

          <speed x="0.000" y="0.000" z="0.000" />

          <angle x="1.000" y="0.000" z="-1.000" w="1.000" />

        </object>

      </children>

    </object>

  </objects>

</world>
```

```
        </object>
    </children>
</object>
<!-- Objeto armário descrito no arquivo 12.xml -->
<object localId="16" name="armario" status="0" type="1" media="22">
    <pos x="55.000000" y="20.000" z="25.000" />
    <orient x="1.000" y="0.000" z="0.000" w="1.000" />
    <scale x="1.000" y="1.000" z="1.000" />
    <speed x="0.000" y="0.000" z="0.000" />
    <angle x="1.000" y="0.000" z="0.000" w="1.000" />
</object>
</objects>
<portals>
    <!-- Portal atribuído ao objeto armário que conecta ao CVE no endereço ip
         200.17.41.189 na porta 3000 -->
    <portal id="15" spId="33" oId="16">
        <address ip="200.17.41.189" port="3000" />
    </portal>
</portals>
<spawnPoints>
    <!-- Nascedouro atribuído a avatares e simulóides -->
    <spawnPoint id="18">
        <elementType>2</elementType> <!-- Avatares -->
        <elementType>3</elementType> <!-- Simulóides -->
    </spawnPoint>
</spawnPoints>
<!-- Árvore que descreve a hierarquia de grupo do trabalho colaborativo. Nessa
     árvore só aparece os usuários colaboradores ou coordenadores (Espectadores não
     aparece)-->
<users>
    <user id="5" name="Lia"> <!-- Subordinados de Lia -->
        <subordinates>
            <user id="6" name="João">
```

```

        <subordinates> <!-- Subordinados de João -->
            <user id="7" name="Pedro" />
            <user id="8" name="Carlos" />
        </subordinates>
    </subordinates>
</user>
</users>
<!-- Árvore que descreve a organização das tarefas listando suas dependências e
      responsáveis -->
<tasks>
    <task id="61" responsible="5" description="Montar uma Cozinha Ideal">
        <!-- Lia é responsável pela tarefa 62 (responsible = "6") -->
        <subtask id="62" responsible="6" description="Dispor Móveis">
            <!-- Tipos de dependência (dependence type):
                  (0) antes de; (1) depois de; (2) durante; (3) ao mesmo tempo -->
            <dependences>
                <dependence type="0" id="63" />
                <dependence type="0" id="64" />
            </dependences>
        </subtask>
        <subtask id="63" responsible="7" description="Distribuir
            Eletrodomésticos">
            <dependences>
                <dependence type="1" id="62" />
                <dependence type="3" id="64" />
            </dependences>
        </subtask>
        <subtask id="64" responsible="8" description="Distribuir Eletroportáteis">
            <dependences>
                <dependence type="1" id="62" />
                <dependence type="3" id="63" />
            </dependences>
        </subtask>
    </task>
</tasks>

```

```
<subtask id="65" responsible="6" description="Colocar Utensílios
Domésticos">
  <dependences>
    <dependence type="1" id="63" />
    <dependence type="2" id="64" />
  </dependences>
  <!-- Subtarefas de subtask 65 -->
  <subtask id="66" responsible="7" description="Colocar Prataria e
Panelas">
    <dependences>
      <dependence type="1" id="67" />
    </dependences>
  </subtask>
  <subtask id="67" responsible="8" description="Colocar Material de
Limpeza">
    <dependences>
      <dependence type="0" id="66" />
    </dependences>
  </subtask>
</subtask>
</task>
</tasks>
</world>
```

Anexo B – Exemplos de arquivo descritor de mídia

```
<!-- Arquivo descritor (XML) de uma mídia composta de uma mesa e duas cadeiras. Nesse
arquivo são guardadas as referências das mídias (cadeiras) que compõe a mesa e as
posições relativas das cadeiras -->

<?xml version="1.0" encoding="UTF-8" ?>

<element id="22" name="mesa-composta" type="8"> <!-- type 8 : mídia composta -->

  <root id="23"></root> <!-- Mídia 23 (23.xml) descreve a mesa -->

  <children> <!-- Lista de mídias que compõe a mesa -->

    <media id="24"> <!-- Mídia 24 (24.xml) descreve uma cadeira -->

      <pos x="-1.34" y="-3.78" z="0"></pos> <!-- Posição relativa à mesa -->

      <orient x="-1" y="0" z="0" w="1"></orient> <!-- Orientação relativa à
      mesa -->

      <scale x="2.0" y="2.0" z="2.0"></scale> <!-- Escala relativa à mesa -->

    </media>

    <media id="24">

      <pos x="-4.89" y="1.45" z="0"></pos>

      <orient x="1" y="0" z="0" w="1"></orient>

      <scale x="2.0" y="2.0" z="2.0"></scale>

    </media>

    <media id="24">

      <pos x="3.56" y="-1.45" z="0"></pos>

      <orient x="-1" y="0" z="0" w="1"></orient>

      <scale x="2.0" y="2.0" z="2.0"></scale>

    </media>

  </children>

</element>
```

<!-- Arquivo descritor (XML) de uma mídia simples de um personagem virtual (avatar). Foi escolhido neste exemplo um avatar por possuir mais informações no arquivo, mas esse modelo de arquivo também serve para descrever um objeto estático (como exemplo, uma cadeira). Nesse caso, alguns campos do arquivo não são necessários -->

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<element id="87" name="Robert" type="3"> <!-- Modelo do avatar -->
```

```
  <body currentGeometry="1" currentMaterial="1"> <!-- Geometria e textura usadas -->
```

```
    <!-- Cada mídia possui uma lista de corpos (geometria + textura) -->
```

```
    <geometries> <!-- Lista de geometrias -->
```

```
      <geometry>
```

```
        <model id="1">human.md2</model> <!-- Modelo de um humano -->
```

```
          <!-- Lista de animações com um identificador da animação  
            do modelo (MD2) utilizado -->
```

```
          <anims>
```

```
            <anim name="stand" id="0"></anim>
```

```
            <anim name="run" id="2"></anim>
```

```
            <anim name="attack" id="1"></anim>
```

```
            <anim name="pain" id="5"></anim>
```

```
            <anim name="jump" id="4"></anim>
```

```
            <anim name="flip" id="3"></anim>
```

```
            <anim name="salute" id="8"></anim>
```

```
            <anim name="taunt" id="6"></anim>
```

```
            <anim name="wave" id="7"></anim>
```

```
            <anim name="point" id="9"></anim>
```

```
          </anims>
```

```
        </geometry>
```

```
      <geometry>
```

```
        <model id="2">dog.md2</model> <!-- Modelo de um cachorro -->
```

```
        <anims>
```

```
          <anim name="stand" id="0"></anim>
```

```
          <anim name="run" id="2"></anim>
```

```
          <anim name="attack" id="3"></anim>
```

```
          <anim name="pain" id="5"></anim>
```

```
          <anim name="jump" id="6"></anim>
```

```
<anim name="flip" id="4"></anim>
<anim name="salute" id="7"></anim>
<anim name="taunt" id="1"></anim>
<anim name="wave" id="9"></anim>
<anim name="point" id="8"></anim>
</anim>
</geometry>
</geometries>
<materials> <!-- Lista de texturas -->
<material id="1">human_mat1.jpg</material>
<material id="2">human_mat2.jpg</material>
<material id="3">dog_mat3.jpg</material>
<material id="3">dog_mat4.jpg</material>
</materials>
<sounds> <!-- Lista de sons -->
<sound id="1">walk.wav</sound>
<sound id="2">hi.mp3</sound>
<sound id="3">bark.wav</sound>
<sound id="3">pain.wav</sound>
</sounds>
</body>
</element>
```

Anexo C – Diagramas de Classes

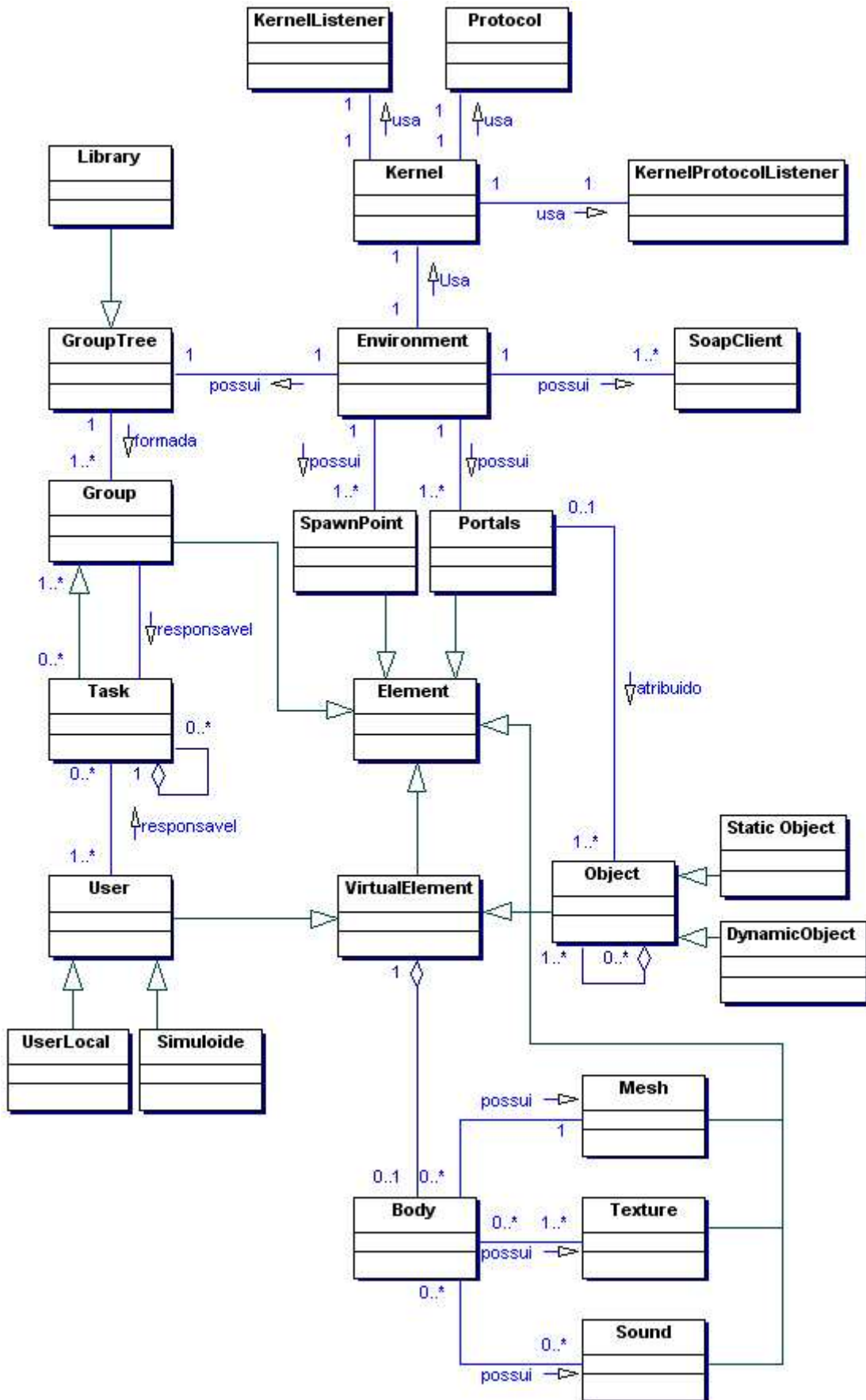


Figura C.7.1. Diagrama de Classes do CIERV.

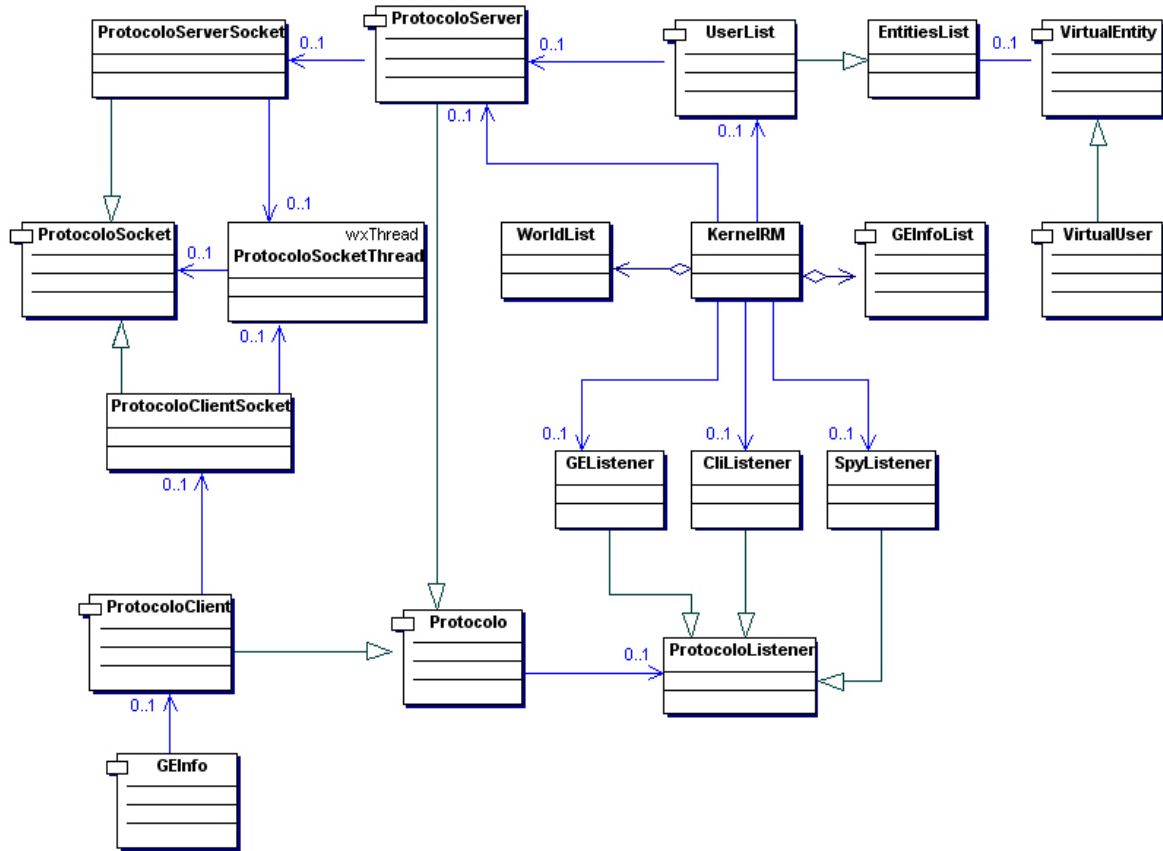


Figura C.7.2. Diagrama de Classes do RM.

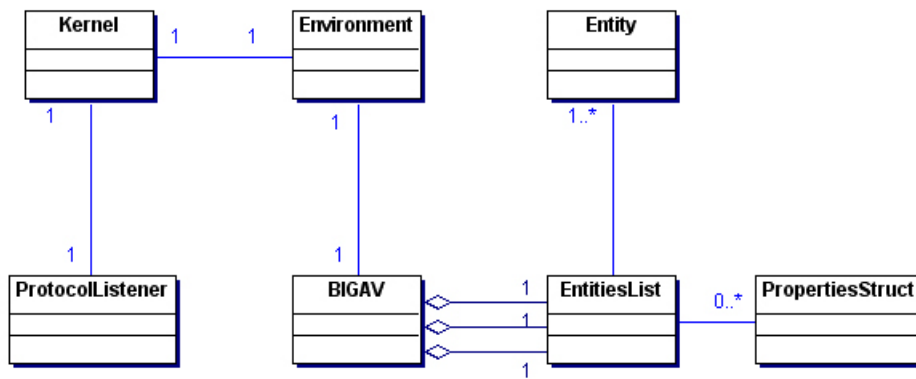


Figura C.7.3. Diagrama de Classes do GE.

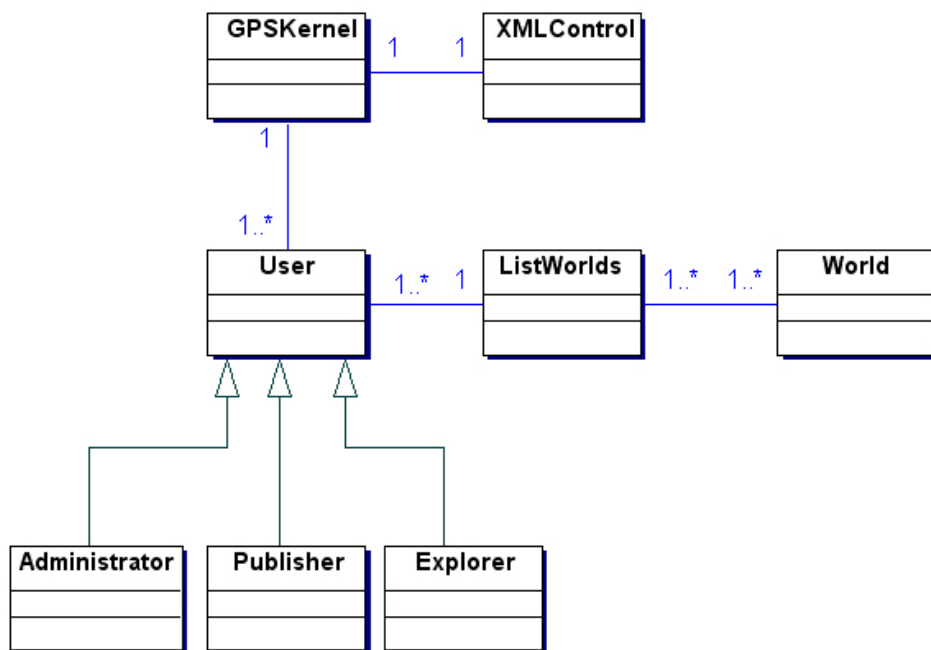


Figura C.7.4. Diagrama de Classes do GPS.

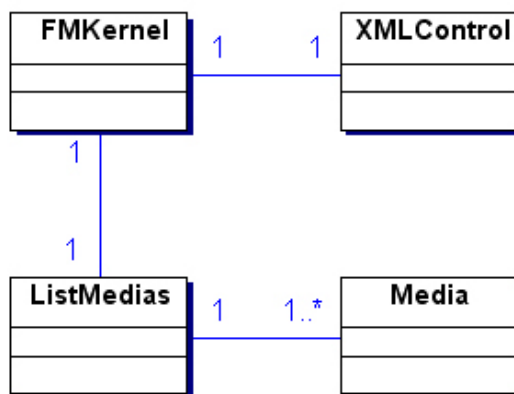


Figura C.7.5. Diagrama de Classes do FM.