

**Universidade Federal de Santa Catarina**  
**Departamento de Engenharia de Produção e Sistemas**  
**Programa de Pós-Graduação em Engenharia de Produção**

# **Utilização de Agentes Inteligentes no Trabalho Colaborativo via Internet**

**Adriano Coser**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, como parte dos requisitos para a obtenção do grau de Mestre.

Área de concentração: Inteligência Aplicada

Orientador: Fernando Álvaro Ostuni Gauthier, Dr.

Florianópolis - SC  
1999

**Adriano Coser**

# **Utilização de Agentes Inteligentes no Trabalho Colaborativo via Internet**

Esta dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia de Produção, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

---

Ricardo Miranda Barcia, PhD.

**Coordenador**

Banca examinadora:

---

Fernando Álvaro Ostuni Gauthier, Dr.

**Orientador**

---

Alejandro Martins Rodriguez, Dr.

---

Roberto Carlos dos Santos Pacheco, Dr.

---

Andréa Valéria Steil, M. Eng.

Florianópolis, outubro de 1999.

Dedico este trabalho aos meus pais, que  
nunca mediram esforços para que seus  
filhos tivessem uma boa formação.

## **Agradecimentos**

Em primeiro lugar à minha família, pelo apoio essencial em todas as minhas realizações.

Ao Professor Fernando Gauthier, pela tranquilidade e segurança com que orientou este trabalho.

À amiga Fabiane Vavassori, pelas importantes contribuições na fase de elaboração dos requisitos da aplicação desenvolvida.

Aos diretores da AltoQi Tecnologia em Informática, José Carlos Pereira, Rui Luiz Gonçalves e Jano D'Araújo Coelho, pelo incentivo durante todo o curso.

Aos colegas da equipe de desenvolvimento da AltoQi Tecnologia, André Banki, André Gustavo Mendonça, Fernando Shauffert e Marco Aurélio Martins Melo, que sempre contribuíram com boas idéias durante a realização deste trabalho.

Aos amigos Edwin Maldonado e Ane Denise Piccinini, proprietários da Hidrosane Engenharia, que gentilmente cederam o espaço da empresa para o desenvolvimento da fase final deste trabalho.

Ao amigo Paulo de Tarso Mendes Luna, pelas idéias e discussões durante a definição do tema desta dissertação.

Ao amigo Gustavo Razzera, pelas contribuições na fase de desenvolvimento da aplicação.

# Sumário

|   |             |
|---|-------------|
| <b>LISTA DE FIGURAS.....</b>                                  | <b>IV</b>   |
| <b>LISTA DE TABELAS .....</b>                                 | <b>VI</b>   |
| <b>LISTA DE SIGLAS E ABREVIACÕES .....</b>                    | <b>VII</b>  |
| <b>RESUMO .....</b>   | <b>VIII</b> |
| <b>ABSTRACT .....</b>   | <b>IX</b>   |
| <b>1 INTRODUÇÃO .....</b>                                     | <b>1</b>    |
| 1.1 MOTIVAÇÃO E JUSTIFICATIVA .....                           | 1           |
| 1.2 OBJETIVOS .....   | 5           |
| 1.3 ESTRUTURA DO TRABALHO .....                               | 6           |
| <b>2 TRABALHO COLABORATIVO SUPORTADO POR COMPUTADOR .....</b> | <b>8</b>    |
| 2.1 CONSIDERAÇÕES INICIAIS .....                              | 8           |
| 2.2 HISTÓRICO .....   | 9           |
| 2.3 DEFINIÇÕES .....  | 10          |
| 2.3.1 CSCW.....   | 11          |
| 2.3.2 Groupware.....  | 13          |
| 2.3.3 Workflow .....  | 13          |
| 2.3.4 Teletrabalho.....                                       | 15          |
| 2.4 CLASSIFICAÇÃO .....                                       | 15          |
| 2.4.1 Classificação Espaço / Tempo .....                      | 16          |
| 2.4.2 Classificação Considerando a Previsibilidade .....      | 16          |
| 2.4.3 Outras Classificações.....                              | 17          |
| 2.5 CSCW E INTERNET .....                                     | 18          |
| 2.5.1 Massa Crítica.....                                      | 18          |
| 2.5.2 Internet e World Wide Web.....                          | 19          |
| 2.5.3 Razões para a Utilização da Internet.....               | 19          |
| 2.5.4 Tecnologias de CSCW sobre a Internet.....               | 20          |
| 2.6 ESTÁGIO DE DESENVOLVIMENTO .....                          | 22          |
| 2.6.1 Lotus Notes .....                                       | 22          |
| 2.6.2 Microsoft NetMeeting.....                               | 26          |
| 2.6.3 Novell GroupWise.....                                   | 27          |
| 2.6.4 Projeto BSCW .....                                      | 28          |
| 2.7 CONSIDERAÇÕES FINAIS .....                                | 29          |
| <b>3 AGENTES INTELIGENTES.....</b>                            | <b>31</b>   |
| 3.1 CONSIDERAÇÕES INICIAIS .....                              | 31          |
| 3.2 DEFINIÇÃO .....   | 33          |
| 3.3 PROPRIEDADES .....  | 37          |
| 3.3.1 Autonomia .....   | 37          |
| 3.3.2 Delegação.....  | 38          |
| 3.3.3 Comunicabilidade e Habilidade Social.....               | 38          |

|          |  |           |
|----------|--|-----------|
| 3.3.4    | <i>Reatividade</i>                               | 38        |
| 3.3.5    | <i>Pró atividade</i>                             | 39        |
| 3.3.6    | <i>Robustez</i>                                  | 39        |
| 3.3.7    | <i>Inteligência</i>                              | 40        |
| 3.3.8    | <i>Aprendizagem</i>                              | 40        |
| 3.3.9    | <i>Outras Características</i>                    | 40        |
| 3.4      | CLASSIFICAÇÃO                                    | 41        |
| 3.4.1    | <i>Franklin e Graesser, 1996</i>                 | 42        |
| 3.4.2    | <i>Caglayan e Harrison, 1997</i>                 | 43        |
| 3.4.3    | <i>Wooldridge e Jennings, 1995</i>               | 44        |
| 3.5      | BENEFÍCIOS DA TECNOLOGIA DE AGENTES              | 46        |
| 3.6      | APLICAÇÕES DE AGENTES                            | 47        |
| 3.7      | AGENTES EM CSCW                                  | 50        |
| <b>4</b> | <b>MODELAGEM DO SISTEMA</b>                      | <b>52</b> |
| 4.1      | CONSIDERAÇÕES INICIAIS                           | 52        |
| 4.2      | ABRANGÊNCIA                                      | 53        |
| 4.2.1    | <i>Grupos de Pesquisa</i>                        | 53        |
| 4.2.2    | <i>Equipes Virtuais</i>                          | 54        |
| 4.2.3    | <i>Determinação da Abrangência do Modelo</i>     | 55        |
| 4.3      | REQUISITOS                                       | 56        |
| 4.3.1    | <i>Tipos de Usuário</i>                          | 56        |
| 4.3.2    | <i>Cadastro de Usuários e Grupos de Trabalho</i> | 57        |
| 4.3.3    | <i>Comunicação On-line</i>                       | 57        |
| 4.3.4    | <i>Reuniões a Distância</i>                      | 58        |
| 4.3.4.1  | <i>Agendamento</i>                               | 58        |
| 4.3.4.2  | <i>Organização da Pauta</i>                      | 59        |
| 4.3.4.3  | <i>Sala de Reuniões</i>                          | 59        |
| 4.3.4.4  | <i>Geração de Atas</i>                           | 60        |
| 4.3.5    | <i>Compartilhamento de Informações</i>           | 60        |
| 4.3.6    | <i>Outras Características</i>                    | 61        |
| 4.4      | PROJETO  | 63        |
| 4.4.1    | <i>Arquitetura</i>                               | 63        |
| 4.4.2    | <i>Comunicação</i>                               | 66        |
| 4.4.2.1  | <i>Recursos de Programação</i>                   | 67        |
| 4.4.2.2  | <i>Hierarquia de Mensagens</i>                   | 68        |
| 4.4.3    | <i>Agentes</i>                                   | 70        |
| 4.4.3.1  | <i>Análise Estado - Evento</i>                   | 70        |
| 4.4.3.2  | <i>Geração de Eventos</i>                        | 71        |
| 4.4.3.3  | <i>Requisitos para os Agentes</i>                | 71        |
| 4.4.3.4  | <i>Modelo para os Agentes</i>                    | 72        |
| 4.4.4    | <i>Servidor</i>                                  | 74        |
| 4.4.4.1  | <i>Arquitetura</i>                               | 74        |
| 4.4.4.2  | <i>Interface com o Administrador</i>             | 76        |
| 4.4.4.3  | <i>Roteador</i>                                  | 76        |
| 4.4.4.4  | <i>Agente de Comunicação</i>                     | 78        |
| 4.4.4.5  | <i>Agente de Chat</i>                            | 78        |
| 4.4.4.6  | <i>Agente de Reunião</i>                         | 79        |
| 4.4.4.7  | <i>Agente de Biblioteca</i>                      | 83        |
| 4.4.4.8  | <i>Caixas Postais</i>                            | 84        |
| 4.4.5    | <i>Cliente</i>                                   | 85        |
| 4.4.5.1  | <i>Comunicador</i>                               | 86        |
| 4.4.5.2  | <i>Agente do Usuário</i>                         | 86        |
| 4.5      | CONSIDERAÇÕES FINAIS                             | 87        |
| <b>5</b> | <b>IMPLEMENTAÇÃO DO SISTEMA</b>                  | <b>89</b> |

|   |   |            |
|---|---|------------|
| 5.1   | CONSIDERAÇÕES INICIAIS .....                            | 89         |
| 5.2   | FERRAMENTAS DE DESENVOLVIMENTO .....                    | 90         |
| 5.2.1   | <i>Escolha da Plataforma de Desenvolvimento</i> .....   | 90         |
| 5.2.1.1   | Ambiente de Desenvolvimento .....                       | 91         |
| 5.2.2   | <i>A Linguagem Java</i> .....                           | 92         |
| 5.2.2.1   | Surgimento da Linguagem .....                           | 92         |
| 5.2.2.2   | Linguagem e Plataforma .....                            | 92         |
| 5.2.2.3   | Principais Características .....                        | 94         |
| 5.3   | O SERVIDOR DO VMEETING .....                            | 96         |
| 5.3.1   | <i>Cadastro de Usuários</i> .....                       | 97         |
| 5.3.2   | <i>Composição dos Grupos de Trabalho</i> .....          | 98         |
| 5.4   | O CLIENTE DO VMEETING .....                             | 99         |
| 5.4.1   | <i>O Login do Usuário</i> .....                         | 99         |
| 5.4.2   | <i>Recursos de Comunicação</i> .....                    | 100        |
| 5.4.2.1   | Comunicação Usuário a Usuário .....                     | 101        |
| 5.4.2.2   | Salas de Conversação .....                              | 102        |
| 5.4.3   | <i>Reuniões a Distância</i> .....                       | 103        |
| 5.4.3.1   | Agendamento .....                                       | 104        |
| 5.4.3.2   | Composição da Pauta .....                               | 110        |
| 5.4.3.3   | A Sala de Reuniões .....                                | 112        |
| 5.4.3.4   | Geração da Ata .....                                    | 114        |
| 5.4.4   | <i>Bibliotecas</i> .....                                | 117        |
| 5.5   | CONSIDERAÇÕES FINAIS .....                              | 119        |
| 5.5.1   | <i>Cumprimento dos Requisitos</i> .....                 | 119        |
| 5.5.2   | <i>Consistência entre Projeto e Implementação</i> ..... | 120        |
| 5.5.3   | <i>Performance</i> .....                                | 120        |
| <b>6</b>  | <b>CONCLUSÕES .....</b>                                 | <b>122</b> |
| 6.1   | TRABALHOS FUTUROS .....                                 | 124        |
| <b>APÊNDICE I – NOTAÇÃO PARA O PROJETO ORIENTADO A OBJETOS ..</b> |   | <b>127</b> |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>                           |   | <b>130</b> |

## Lista de Figuras

|   |     |
|---|-----|
| Figura 2-1 – Um processo de <i>Workflow</i> . .....                             | 14  |
| Figura 2-2 – Custo / benefício dos sistemas colaborativos. ....                 | 18  |
| Figura 2-3 – Modelo de <i>groupware</i> da Lotus. ....                          | 23  |
| Figura 3-1 – Uma visão da arquitetura de agente. ....                           | 36  |
| Figura 3-2 – Classificação dos agentes segundo sua natureza. ....               | 42  |
| Figura 4-1 – Visão simplificada da arquitetura do sistema. ....                 | 65  |
| Figura 4-2 – Comunicação através <i>streams</i> . ....                          | 68  |
| Figura 4-3 – Classes do protocolo de comunicação do sistema. ....               | 69  |
| Figura 4-4 – Modelo para os agentes do sistema. ....                            | 73  |
| Figura 4-5 – Arquitetura do servidor. ....                                      | 75  |
| Figura 4-6 – Modelo para o agente de reunião. ....                              | 81  |
| Figura 4-7 – Modelo para o agente de biblioteca. ....                           | 83  |
| Figura 4-8 – Arquitetura da aplicação cliente. ....                             | 85  |
| Figura 5-1 – Compilação e execução de programas em Java. ....                   | 93  |
| Figura 5-2 – Uma compilação, diversas plataformas. ....                         | 93  |
| Figura 5-3 - Janela principal do servidor do sistema. ....                      | 97  |
| Figura 5-4 – Diálogo utilizado no cadastro de usuários. ....                    | 97  |
| Figura 5-5 – Diálogo para composição dos grupos de trabalho. ....               | 98  |
| Figura 5-6 – Janela de <i>login</i> do usuário. ....                            | 99  |
| Figura 5-7 – As configurações do usuário. ....                                  | 100 |
| Figura 5-8 – A janela principal da aplicação cliente. ....                      | 100 |
| Figura 5-9 – A comunicação usuário a usuário. ....                              | 101 |
| Figura 5-10 – Sala de conversação de um grupo de trabalho. ....                 | 102 |
| Figura 5-11 – Menu com os comandos sobre as reuniões. ....                      | 103 |
| Figura 5-12 – Diálogo para solicitação de uma nova reunião. ....                | 104 |
| Figura 5-13 – O agente de reunião solicita a opinião do usuário. ....           | 106 |
| Figura 5-14 – O agente de reunião informa o coordenador sobre as opiniões. .... | 107 |
| Figura 5-15 – Aviso sobre a confirmação de uma reunião. ....                    | 108 |
| Figura 5-16 – Uma reunião deve ser iniciada. ....                               | 108 |
| Figura 5-17 – Aviso para o usuário sobre o início de uma reunião. ....          | 109 |
| Figura 5-18 – A agenda do grupo de trabalho. ....                               | 110 |

|  |     |
|--|-----|
| Figura 5-19 – Diálogos para composição da pauta da reunião. ....   | 111 |
| Figura 5-20 – A sala de reuniões.....                              | 112 |
| Figura 5-21 – Anotação das decisões tomadas durante a reunião..... | 114 |
| Figura 5-22 - Arquivo de atas de um grupo de trabalho. ....        | 115 |
| Figura 5-23 – Uma ata de reunião gerada pelo agente.....           | 116 |
| Figura 5-24 – Manipulação das bibliotecas do sistema.....          | 117 |
| Figura 5-25 – Visualização de uma biblioteca. ....                 | 118 |

## **Lista de Tabelas**

|   |    |
|---|----|
| Tabela 2-1 – Classificação espaço / tempo para os ambientes em CSCW. .... | 16 |
| Tabela 2-2 – Ambientes em CSCW considerando previsibilidade. ....         | 17 |
| Tabela 3-1 – Classificação de agentes por ambientes e tarefas.....        | 44 |
| Tabela 4-1 – Estados do agente de reunião. ....                           | 80 |

## Lista de Siglas e Abreviações

|        |  |
|--------|--|
| API    | <i>Application Program Interface</i>                                       |
| BSCW   | <i>Basic Support for Cooperative Work</i>                                  |
| CASE   | <i>Computer Aided Software Engeneering</i>                                 |
| CGI    | <i>Common Gateway Interface</i>  |
| CESAR  | <i>Collaboration Environments and Service Architectures for Researches</i> |
| CSCW   | <i>Computer Supported Collaborative (Cooperative) Work</i>                 |
| CMC    | <i>Computer Mediated Communication</i>                                     |
| DLL    | <i>Dynamic Linked Library</i>  |
| FTP    | <i>File Transfer Protocol</i>  |
| HCI    | <i>Human-Computer Interaction</i>  |
| HTTP   | <i>Hipertext Transfer Protocol</i>   |
| IA     | Inteligência Artificial  |
| IDE    | <i>Integraded Development Environment</i>                                  |
| IT     | <i>Information Technology</i>  |
| MIT    | <i>Massachusetts Institute of Technology</i>                               |
| TCP/IP | <i>Transfer Control Protocol / Internet Protocol</i>                       |
| URL    | <i>Unique Resource Locator</i>   |

## Resumo

O trabalho colaborativo tem sido uma prática cada vez mais frequente em diversos tipos de organizações. As mudanças estruturais na busca por maior competitividade, as novas expectativas geradas pela evolução tecnológica, o aumento da complexidade nos processos e a necessidade de se trabalhar a distância, estão entre os fatores que incentivam a adoção deste paradigma.

No aspecto tecnológico, a evolução das redes de computadores, incluindo o advento da Internet, transformaram os computadores em elementos centrais das chamadas tecnologias da informação. A adoção dessas tecnologias, somada à melhoria das interfaces com os usuários, incentivou a disseminação de computadores nos ambientes de trabalho. A nova realidade atraiu a atenção de pesquisadores e desenvolvedores de diversas áreas, que vislumbraram a idéia de utilizar sistemas computacionais no apoio ao trabalho colaborativo nas organizações.

Em torno dessa idéia, surgiu a área de pesquisa denominada CSCW – Trabalho Colaborativo Suportado por Computador, reunindo pessoas de diversas áreas de conhecimento, que compartilhavam o objetivo de entender as formas de trabalho humano e como os sistemas poderiam suportá-las. Na busca por tecnologias aplicáveis, a área de CSCW vale-se de todo o campo das ciências da computação. Neste contexto, a tecnologia de agentes inteligentes apresenta-se como uma alternativa promissora, visando a construção de *softwares* que agem em favor dos usuários de maneira autônoma. Este trabalho explora a utilização de agentes como elementos essenciais na arquitetura de uma aplicação para suporte ao trabalho colaborativo sobre a Internet.

Palavras-chaves: CSCW, agentes inteligentes, Internet.

## **Abstract**

Collaborative work has been increasingly used in several types of organizations. Growing competitiveness, new expectancies generated by the technological evolution, raising in processes complexity and the need to work at distance are some of the reasons that lead to this paradigm.

On the technological side, the evolution of computer networks, including Internet, changed the role of computers, making them the central elements of information technologies. These technologies, associated with the evolution of the user interfaces, contributed to the proliferation of computers on work environments. This new reality attracted researchers and developers from several areas, realizing the idea of using computer systems to support collaborative work in organizations.

Around this idea emerged the research area called CSCW – Computer Supported Collaborative Work, joining people who shared the goal of understanding the ways humans work together and how computer systems could support them. In its search for applicable technologies, CSCW research uses the entire computer science field. Under this context, intelligent agents show up as a promising alternative, aiming at the construction of softwares that help users in an autonomous way. This work explores the use of agents as essential elements of an application to support collaborative work over the Internet.

Keywords: CSCW, intelligent agents, Internet.

# 1 Introdução

## 1.1 Motivação e Justificativa

Desde o seu surgimento, os sistemas computacionais têm passado por uma rápida e constante evolução, tanto nas tecnologias empregadas, quanto na quantidade de usuários beneficiados. Esta evolução é impulsionada, em grande parte, pelas necessidades evidenciadas na sociedade. Ao mesmo tempo, transformações importantes ocorridas na sociedade são motivadas pelas novas possibilidades trazidas pela evolução dos sistemas. Estabelece-se assim um ciclo evolutivo, que afeta diretamente a vida das pessoas e das organizações. Importantes transformações relacionadas com este ciclo referem-se às formas como as pessoas trabalham. Neste contexto, a possibilidade de se utilizar computadores como meios de comunicação e compartilhamento de informações exerce papel fundamental.

As primeiras redes de computadores surgiram com o principal objetivo de prover o compartilhamento de recursos, como arquivos e impressoras, nos ambientes informatizados. A evolução desta tecnologia permitiu que a interligação de computadores excedesse aos limites dos escritórios e organizações, com o surgimento das redes de médio e longo alcance. A criação da Internet, há 30 anos, traçaria definitivamente a tendência de utilização de computadores como meios de comunicação, possibilitando a união de sistemas heterogêneos espalhados por todo mundo.

Com a evolução das redes, computadores tornaram-se o centro de uma área de conhecimento denominada *Tecnologia da Informação* (IT), que abrange todos os aspectos do gerenciamento e processamento de informações nos diversos tipos de organizações, envolvendo indústria, comércio, ambientes acadêmicos, etc. [AOL 99]. Segundo Ljungberg e Kristoffersen [LJU 99], a adoção crescente das novas tecnologias de informação constitui uma das grandes motivações para a prática do trabalho colaborativo no dia a dia das organizações.

Outro fator tecnológico relacionado com a adoção do trabalho colaborativo, destacado por Bannon [BAN 93a], refere-se à “*extensão da utilização do computador a pessoas para as quais a facilidade de uso constitui um requisito fundamental*”. Isto trouxe uma rápida evolução das interfaces homem – computador (HCI), tornando os sistemas utilizáveis por pessoas sem conhecimentos específicos em informática. Ainda segundo Bannon, mais recentemente as interfaces passaram a visar menos o usuário individual, voltando-se mais para o trabalho e para as organizações.

O suporte computacional ao trabalho colaborativo passou a despertar o interesse de pesquisadores e desenvolvedores em diversas áreas, incluindo automação de escritórios, sistemas de informações, comunicação mediada por computador (CMC), sistemas distribuídos, HCI, entre outras [BAN 93b]. Em 1984, Irene Greif e Paul Cashman criaram o termo *Trabalho Colaborativo Suportado por Computador (CSCW)*, reunindo sob esta denominação o interesse comum de diversas áreas, em torno do entendimento das formas como as pessoas trabalham e como a tecnologia pode apoiar este trabalho [GRU 94].

Conforme Bannon [BAN 91], o termo *trabalho colaborativo*, foco das pesquisas em CSCW, possui uma história longa, tendo sido empregado pelos economistas já na primeira metade do século XIX, para designar o trabalho envolvendo múltiplos atores. O termo foi definido mais formalmente por Karl Max, em 1867, como “*múltiplos indivíduos trabalhando juntos, de maneira planejada, no mesmo processo de produção ou em processos diferentes mas conectados*”.

A disseminação da prática do trabalho colaborativo nas organizações não é tão antiga quanto a história do termo, justamente pela necessidade de suporte tecnológico. Colaboração requer comunicação direta entre os participantes, compartilhamento eficiente de informações, conhecimento e, possivelmente, controle de cada participante sobre as ações dos demais, entre outros recursos [DIX 96]. Atualmente, muitas pessoas passam boa parte do seu dia de trabalho interagindo e cooperando com outras, seja através de encontros físicos ou de meios eletrônicos [LJU 99]. Isto é devido, em grande parte, aos frutos já colhidos na área de CSCW.

Nos últimos anos, o nível de interesse pela área cresceu muito. O número de sistemas desenvolvidos com o objetivo de apoiar as atividades colaborativas é cada vez

maior. A soma de esforços já produziu um grande entendimento sobre a complexidade do trabalho em grupo e sobre a flexibilidade requerida pelos sistemas que visam administrar esta complexidade. [BEN 97b] A atração por esta área de aplicação é bastante devida a diversas mudanças que estão ocorrendo, por exemplo, no ambiente das organizações, nas expectativas das pessoas em relação aos sistemas, e na própria tecnologia [BAN 93a]. Estas mudanças podem ser ilustradas por alguns exemplos:

- ✓ As relações das empresas com seus clientes e fornecedores são cada vez mais baseadas na tecnologia. O compartilhamento de informações entre organizações já é quase tão importante quanto o compartilhamento dentro das próprias organizações. [TAM 97]
- ✓ A estrutura das organizações está mudando. Na busca por competitividade, recursos como *downsizing*, reestruturação, reengenharia e incremento da qualidade são frequentemente aplicados. Estas mudanças requerem esforços de equipe, exigindo que as pessoas trabalhem de forma colaborativa. [CWC 98]
- ✓ As facilidades trazidas pelos modernos recursos de comunicação, destacando-se aqui a Internet, associadas ao fato de que muitas pessoas trabalham atualmente sobre informações, possibilitam que estas trabalhem a distância, muitas vezes no seu próprio lar, sem a necessidade de se deslocarem diariamente para um local determinado. [BRI 95]
- ✓ A busca por novos mercados de *software* atraiu o interesse dos desenvolvedores, que visualizaram CSCW como uma área emergente para a criação de novas aplicações, os *groupwares*, para suportar atividades de grupos ao invés de suportar somente atividades individuais. [BAN 93a]
- ✓ Quando novas tecnologias são desenvolvidas, frequentemente para atender necessidades já evidenciadas, acabam por gerar novas possibilidades, que aumentam a expectativas das pessoas. É crescente a demanda por interfaces mais flexíveis e personalizáveis, e por funcionalidades que permitam às pessoas executarem suas tarefas de maneira mais eficiente [BAN 93a].

Segundo Laamanen [LAA 97], CSCW é um termo que cobre uma grande variedade de aplicações, desde *e-mail* e *newsgroups*, até aplicações avançadas, que

suportam reuniões a distância em tempo real. Não existe uma definição aceita universalmente sobre o termo, o que conduz também à falta de metodologias bem fixadas para o projeto de aplicações para a área. Bannon [BAN 91] ressalta que CSCW também não pode ser definido através das tecnologias empregadas na construção dos sistemas: “CSCW é uma área de pesquisa voltada para o projeto de sistemas, e como qualquer outra área de aplicação, em sua busca por técnicas aplicáveis, pode valer-se potencialmente de todo o campo das ciências da computação”.

A motivação central deste trabalho parte justamente da necessidade de aplicação de tecnologias computacionais que possam incrementar a eficiência dos sistemas de suporte ao trabalho colaborativo. Neste sentido, a tecnologia de agentes inteligentes e a Internet apresentam-se como recursos cujas possibilidades, principalmente no caso dos agentes, estão apenas começando a ser exploradas. Acredita-se que essas tecnologias podem trazer um incremento significativo nas interfaces e nos recursos oferecidos aos usuários. Além disso, podem produzir benefícios importantes em termos de desenvolvimento, possibilitando arquiteturas mais simples, modulares e extensíveis.

Agentes inteligentes são entidades computacionais que se destacam de outras categorias de *software*, principalmente pelas suas características de autonomia e delegação. Agentes são capazes de realizar tarefas em favor de usuários ou de outros agentes, sem a intervenção direta dos mesmos. Trata-se de uma tecnologia em pleno desenvolvimento e aplicável a diversas áreas, como comércio eletrônico, gerenciamento de informações pessoais, gerenciamento de redes, etc.

A natureza das aplicações em CSCW justifica a opção pela utilização de agentes como elementos essenciais na arquitetura dos sistemas. Segundo Jennings e Wooldridge [JEN 96], agentes não beneficiam somente indivíduos, contribuindo também para o bom desempenho de organizações inteiras. Nestas, os processos têm se tornado cada vez mais complexos, envolvendo vários indivíduos, muitas vezes dispersos geograficamente. Para Hermans [HER 96], os usuários nestes ambientes necessitam de ferramentas que, além de viabilizarem comunicação e compartilhamento de recursos, atuem efetivamente na organização do trabalho em equipe. Liu e Dix [LIU 97] afirmam que agentes inteligentes podem contribuir grandemente para o atendimento dessas exigências, atuando como procuradores, mediadores ou coordenadores, em favor dos usuários envolvidos.

Além da tecnologia de agentes, este trabalho explora a utilização da Internet como plataforma de comunicação e transmissão de dados. Aplicações em CSCW são tipicamente distribuídas e, cada vez mais, necessitam transpor os limites físicos das organizações. Com isso, surgem os problemas relacionados à interligação de plataformas heterogêneas, que pode ser um grande fator limitante quando se utiliza uma arquitetura de rede que não constitui um padrão universal. Segundo Dix [DIX 96], a Internet oferece uma infraestrutura globalmente acessível e independente de plataforma operacional, o que justifica, em grande parte, sua adoção como ambiente para a criação de aplicações colaborativas. Além do aspecto técnico, a escolha da Internet é justificada também pelo número de usuários e de organizações que esta alcança. Para Dix, o sucesso de uma aplicação em CSCW está fortemente ligado ao número de usuários que esta beneficia.

## **1.2 Objetivos**

### Objetivo Geral

Explorar a tecnologia de agentes inteligentes na modelagem e implementação de uma aplicação de suporte ao trabalho colaborativo via Internet.

### Objetivos Específicos

- ✓ Realizar um estudo sobre a área de CSCW, buscando principalmente identificar as necessidades dos ambientes onde o trabalho colaborativo se faz necessário, ou pode ser empregado como alternativa na busca por melhores resultados. Este estudo deve fornecer também uma visão sobre as tendências do desenvolvimento nesta área, essencial para a elaboração de um modelo com reais possibilidades de aplicação.
- ✓ Pesquisar a tecnologia de agentes inteligentes, procurando compor um embasamento teórico para o desenvolvimento de agentes que contribuam efetivamente para os requisitos da área de aplicação.
- ✓ Delimitar um campo de abrangência dentro da área de CSCW, a partir do qual se possa propor uma arquitetura de suporte ao trabalho colaborativo. Esta delimitação terá como base os estudos realizados e a observação empírica de atividades

colaborativas, como o trabalho de grupos de pesquisa no ambiente acadêmico.

- ✓ Dentro do escopo delimitado, definir um conjunto bem determinado de requisitos a serem atendidos pela aplicação.
- ✓ Propor um modelo computacional que atenda aos requisitos especificados, buscando a utilização de agentes como entidades fundamentais e a Internet como plataforma de comunicação. Este modelo deve englobar tanto a arquitetura externa do sistema quanto a estrutura interna dos seus componentes.
- ✓ Implementar o modelo proposto em linguagem de programação Java.

### 1.3 Estrutura do Trabalho

Os estudos e contribuições realizados neste trabalho, são relatados da seguinte maneira:

O capítulo 2 traz um levantamento sobre a área de CSCW. Origem, evolução e tecnologias associadas são alguns dos aspectos abordados. A importância da Internet e as tendências atuais de desenvolvimento, são questões essenciais neste levantamento.

O capítulo 3 relata os estudos sobre a tecnologia de agentes inteligentes, identificando as características que diferem estas entidades dos *softwares* tradicionais. Os benefícios trazidos por esta tecnologia, suas principais áreas de aplicação e suas possibilidades em CSCW, também são assuntos abordados.

No quarto capítulo é relatada toda a fase de modelagem do sistema, envolvendo a delimitação do escopo de atuação, a especificação dos requisitos e a elaboração do projeto do sistema. Durante a exposição do projeto, diversos problemas são identificados e soluções são apresentadas. Estes problemas referem-se, por exemplo, ao desenvolvimento de aplicações distribuídas, à construção de agentes e ao suporte para a comunicação entre as entidades do modelo.

O capítulo 5 apresenta a implementação do modelo proposto. Inicialmente são apresentadas questões sobre a escolha da linguagem e ferramenta de desenvolvimento. Toda a interface do sistema implementado é apresentada através de exemplos

ilustrativos. Durante a apresentação, são relatados aspectos importantes sobre o comportamento dos agentes e sobre a implementação propriamente dita.

O sexto capítulo conclui a dissertação, trazendo uma análise sobre os resultados obtidos e sugestões para trabalhos futuros.

O apêndice I apresenta resumidamente a notação desenvolvida por Coad e Yourdon [COA 93] para o projeto orientado a objetos. Esta notação é utilizada no capítulo 4 na modelagem das entidades do sistema.

## 2 Trabalho Colaborativo Suportado por Computador

### 2.1 Considerações Iniciais

Em 1984, Paul Cashman e Irene Grief organizaram um *workshop*, onde pessoas de diversas áreas de estudo puderam trocar idéias sobre a maneira como as pessoas trabalham e sobre como a tecnologia poderia apoiar suas atividades. O termo Trabalho Colaborativo (ou Cooperativo) Suportado por Computador foi empregado naquela ocasião para designar o interesse comum das pessoas envolvidas [GRU 94]. Atualmente, é seguro afirmar que este evento deu origem a uma área de pesquisa e desenvolvimento com enorme potencial, cujos frutos mal começaram a ser colhidos. A sigla CSCW, do inglês *Computer-Supported Cooperative (Collaborative) Work*, é utilizada universalmente para referenciar a área de conhecimento.

Segundo Bentley *et al.* [BEN 97b], durante os anos que sucederam seu surgimento, o nível de interesse no campo de CSCW tem crescido muito. Um grande número de sistemas foram desenvolvidos com objetivo de fornecer suporte ao trabalho colaborativo. Esses esforços conduziram a um entendimento cada vez maior sobre a complexidade associada ao trabalho em grupo e sobre a flexibilidade necessária nos sistemas de suporte. Apesar dos avanços, pouquíssimos sistemas colaborativos são utilizados em grande escala, principalmente no que se refere a grupos de trabalho dispersos geograficamente, que ainda utilizam o correio eletrônico e os programas de transferência de arquivos como principais ferramentas de apoio.

O fato de uma sigla ser largamente utilizada para designar uma área de estudo, não significa a existência de um entendimento claro sobre o seu significado. Segundo Bannon, em [BAN 93b], diversos termos, como *groupware*, *collaborative computing*, *workgroup computing*, *coordination technology*, são cada vez mais utilizados. No entanto, apesar do interesse crescente, existe muita confusão sobre qual o foco exato da área denominada CSCW. A afirmação de Bannon parece ser válida até hoje. O interesse na área é cada vez maior, impulsionado pelo crescimento da utilização das redes de

computadores e pela própria tendência de reestruturação das organizações. Um número cada vez maior de novas tecnologias, incluindo vídeo conferências, sistemas de autoria colaborativa, sistemas de aprendizado colaborativo e outros, têm sido agrupados sob o domínio da sigla CSCW, que já havia sido tratada por Bannon [BAN 91] como um “*termo guarda-chuva*” para diversas tecnologias.

Esta área de pesquisa, relativamente nova e extremamente abrangente, é a atividade fim do modelo e da implementação propostos neste trabalho. O objetivo deste capítulo é descrever a área de pesquisa, buscando definições, aspectos importantes e aplicações, que contribuam para um melhor entendimento sobre os seus objetivos e sobre o seu estágio atual de desenvolvimento.

## **2.2 Histórico**

Liam Bannon [BAN 93b] [BAN 92] traz um breve histórico sobre o surgimento da área de pesquisa em CSCW, que será resumido nesta seção. Segundo o autor, o surgimento da expressão “Trabalho Colaborativo Suportado por Computador” pode ser atribuído a Irene Greif, que na época trabalhava no MIT, e a Paul Cashman, da Digital, no início dos anos 80. Estes organizaram uma pequena convenção em Endicott House, Massachusetts, em agosto de 1984, que reuniu cerca de 34 pessoas interessadas em áreas relativamente distintas, como sistemas de informação, hipertexto, sistemas distribuídos, computação mediada por computador e outras. O objetivo era discutir aspectos comuns àquelas áreas de estudo e identificar possibilidades para o desenvolvimento de sistemas para suportar atividades distribuídas no trabalho das pessoas.

A primeira conferência no assunto ocorreu em Austin, Texas, em 1986, reunindo cerca de 300 pessoas. Entre os principais tópicos estavam o desenvolvimento e utilização de sistemas de conferência, experiências com salas de reuniões via computador, ferramentas de filtro para correio eletrônico e a edição colaborativa de hipertextos. Portland, Oregon, sediou uma conferência maior em 1988, com cerca de 485 participantes, que assistiram à apresentação de artigos voltados principalmente para a qualidade no projeto de sistemas colaborativos. A interação com o usuário, desde o

início do projeto, foi colocada como um requisito fundamental para a qualidade dos produtos na área.

O interesse no assunto era crescente e a primeira conferência Européia aconteceu em 1989, em Gatwick, na Inglaterra. As conferências passaram a ser anuais, tanto na Europa como na América do Norte. Os encontros na área tornaram-se frequentes, com diversos *Workshops* e conferências tratando de tópicos específicos, como a tecnologia para colaboração, sistemas de suporte à decisão para grupos, sistemas multi-usuários, etc. Em cada evento podia-se observar a reunião de pessoas com experiências bastante distintas e a formação de duas correntes: a primeira formada por pessoas interessadas na modelagem de sistemas de comunicação para escritórios e afins; a segunda mais voltada para o desenvolvimento de práticas cada vez mais sofisticadas para o trabalho colaborativo.

Além do crescimento no número de eventos, muitas publicações em áreas como interação homem - computador, engenharia de *software*, suporte à decisão, etc., passaram a incluir CSCW em sua lista de tópicos.

Nos anos que sucederam o histórico descrito por Bannon, até os dias de hoje, é evidente o aumento de interesse pela área. Uma pesquisa sobre o termo CSCW na *Web*, por exemplo, resulta em um número espantoso de *links*, que conduzem à revistas eletrônicas (*Journal of Computer-Mediated Communication*, *Journal of Organisational Computing*, *Collaborative Computing at Nottingham University*, etc.), artigos, dissertações, teses e grandes projetos de pesquisa (como os projetos BSCW e CESAR, no centro de pesquisas GMD na Alemanha). A pesquisa revela também que o interesse nesta área não é restrito à comunidade científica. O interesse comercial é enorme, com empresas oferecendo soluções que vão desde a automatização de escritórios até a realização de vídeo conferências.

## 2.3 Definições

Não existe um consenso sobre a definição e os objetivos exatos da área denominada CSCW. Alguns autores sequer consideram CSCW como uma linha independente de pesquisa. Além disso, outros termos como *groupware*, automatização de *workflow*, teletrabalho, que representam aplicações com certo grau de distinção, são

frequentemente utilizados de forma indiscriminada, provocando ainda mais confusão sobre os limites e objetivos da área de pesquisa. Esta seção apresenta definições e considerações lançadas por alguns autores sobre CSCW, procurando também explicar o significado de outros termos importantes associados a esta área.

### 2.3.1 CSCW

Ljungberg e Kristoffersen [LJU 99] definem CSCW como o termo que “*captura todos os tipos de tecnologias de informação explicitamente projetadas para facilitar a cooperação e a comunicação entre as pessoas*”. O trabalho de muitas pessoas envolve interação com outras pessoas durante boa parte do tempo, seja através de reuniões presenciais ou de meios eletrônicos. Uma das maiores razões para o desenvolvimento dessas tecnologias de informação é o aumento acentuado da interação humana nos ambientes produtivos, presenciado nos últimos anos.

Para Renata Araujo *et al.* [ARA 95], CSCW é a “*disciplina de pesquisa para o estudo das técnicas e metodologias de trabalho em grupo e das formas como a tecnologia pode auxiliar este trabalho*”. Os autores salientam, no entanto, que os limites da área não são bem estabelecidos. Além do trabalho, também são frutos desta área de pesquisa aplicações visando atividades de lazer, interação social ou com objetivos educacionais. As aplicações nem sempre apresentam um caráter cooperativo ou colaborativo, envolvendo também atividades de competição e negociação.

Lars Bergman [BER 95] define CSCW como “*a área que enfatiza a importância de se trabalhar eficientemente em grupo no processo de desenvolvimento*”, não importando a natureza deste. Bergman lembra que esta área de pesquisa refere-se primeiramente à interação entre atores humanos, e não entre computadores e componentes de *software*. Sendo assim, este tema deve ser tratado sob o ponto de vista do comportamento colaborativo dos humanos. “*A combinação deste comportamento e das situações envolvidas, formam os requisitos a serem alcançados pelo software e hardware de suporte*”.

Liam Bannon [BAN 93b], após destacar a falta de uma definição universalmente aceita para CSCW, caracteriza a área de estudo de quatro formas diferentes:

- ✓ De maneira mais simples, CSCW pode ser visto como um “*termo guarda-chuva*”,

com pouco conteúdo, além da preocupação com pessoas, computadores e atividades envolvendo cooperação e até mesmo competição. A utilidade desta definição vaga é permitir que pessoas de diferentes áreas de estudo tenham uma arena comum para discutir os pontos em que suas áreas se interceptam e se relacionam com a idéia de trabalho colaborativo.

- ✓ CSCW como um novo paradigma: Bannon traz argumentos de alguns autores caracterizando CSCW como uma nova maneira de visualizar o projeto de diversos tipos de sistemas de suporte, ao invés de distinguir CSCW como uma nova área de pesquisa.
- ✓ Outra visão para o termo, é caracterizá-lo como a área de estudos preocupada principalmente com o desenvolvimento de *software* para suportar grupos ou equipes. De acordo com Bannon, esta visão é criticada por alguns autores, por apresentar na maioria das vezes a idéia de grupos pequenos e homogêneos, formados por pessoas que se relacionam diretamente no trabalho. Este conceito muitas vezes despreza questões importantes nas organizações, como poder e política.
- ✓ CSCW como o suporte tecnológico para as formas de trabalho colaborativo: nesta abordagem, CSCW teria como foco principal o entendimento da natureza do trabalho colaborativo, objetivando o desenvolvimento de tecnologias de suporte adequadas.

### O Significado do “C”

Uma dúvida pertinente ao se tratar da definição de CSCW é o significado do segundo “C” da sigla. Grudin [GRU 94] esclarece que a dúvida provém da diferença de visão entre as pessoas envolvidas em projetos da área. De um lado, pessoas focalizam pequenos grupos, onde os membros compartilham uma meta comum e têm a necessidade de comunicarem-se para atingir esta meta. Para estas pessoas, o “C” significa **cooperativo**. De outro lado estão as pessoas interessadas em sistemas organizacionais e gerenciamento de grandes projetos, envolvendo usualmente grandes grupos com metas muitas vezes não coincidentes ou até conflitantes. Para estes, o “C”

significa **colaborativo**. O enfoque deste trabalho não está restrito a um determinado tipo ou tamanho de organização, e o termo “colaborativo” foi escolhido arbitrariamente.

### **2.3.2 Groupware**

Este é um termo de mercado, utilizado para designar produtos empregados no apoio ao trabalho em grupo. Segundo Araujo *et al.* [ARA 95], o termo costuma ser utilizado frequentemente como sinônimo de CSCW. No entanto, as expressões representam idéias diferentes. CSCW representa uma linha de pesquisa, onde pesquisadores de diversas disciplinas buscam entender o processo colaborativo e identificar as formas como a tecnologia pode apoiar este processo. *Groupware* pode ser considerado como um dos frutos desta linha de pesquisa, caracterizando o *software* e o *hardware* que viabilizam o trabalho em grupo, independente de tamanho e dispersão geográfica do grupo. Sistemas de correio eletrônico, teleconferencias, suporte à decisão em grupo e editores de texto colaborativos, são exemplos de *groupwares*.

Para Tamura [TAM 97], a característica essencial dos *softwares* contidos no domínio do *groupware*, é tornar o grupo mais produtivo. O aumento na produtividade seria fruto principalmente do compartilhamento e gerenciamento de informações.

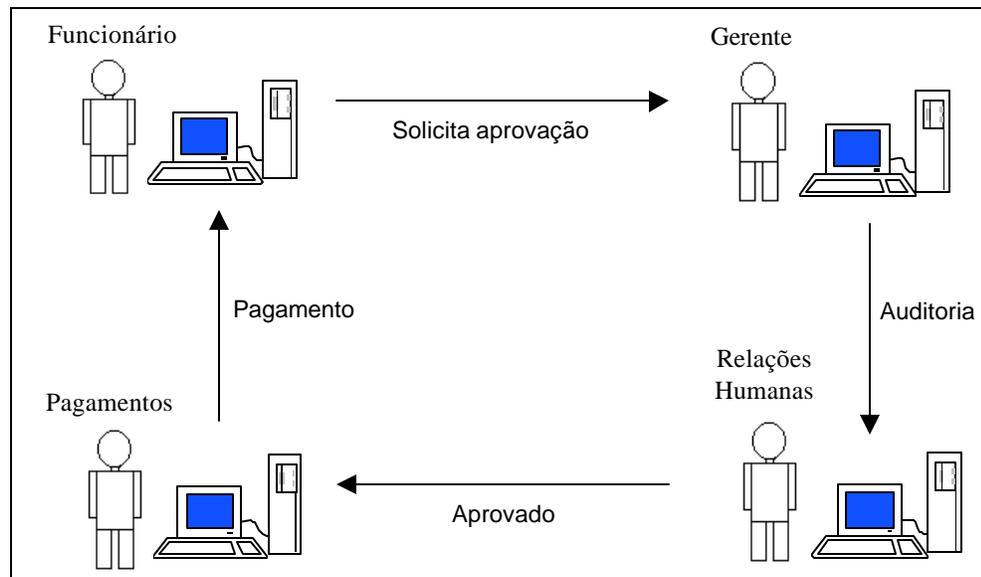
A definição encontrada em [AOL 99] caracteriza *groupware* como o *software* que auxilia colegas em um grupo de trabalho a organizar suas atividades, suportando operações como: agendamento de reuniões, alocação de recursos, correio eletrônico, compartilhamento e proteção de documentos, utilidades para telefonia, distribuição de arquivos e circulação de notícias. Segundo esta enciclopédia eletrônica, “*groupware é algumas vezes chamado de software de produtividade para grupos de trabalho*”.

### **2.3.3 Workflow**

*Workflow* é outra expressão mercadológica associada à área de CSCW. A automação de processos de *workflow* representa um grande campo de aplicação para as tecnologias de *groupware*.

Atualmente nas empresas, uma tarefa normalmente faz parte de uma meta maior. O exemplo ilustrado na Figura 2-1, mostra uma situação simples, de um empregado

solicitando o reembolso de suas despesas de viagem a trabalho. Mesmo em pequenas instituições, esta tarefa trivial pode envolver várias pessoas: a solicitação é enviada pelo funcionário ao gerente, que solicita a confirmação das despesas ao departamento de relações humanas. Este aprova as despesas e solicita seu pagamento ao responsável por esta tarefa, que finalmente efetua o pagamento. [TAM 97]



**Figura 2-1 – Um processo de *Workflow*.**

Fonte: [TAM 97]

Independente do número de pessoas envolvidas, as tarefas em uma empresa frequentemente são decompostas e acabam constituindo um processo com um fluxo de trabalho associado, que precisa ser seguido corretamente para que a tarefa original se complete.

Automatização de *workflow* significa justamente automatizar o fluxo de informações e de tarefas, associado aos processos dentro de uma instituição. *Groupwares*, além de automatizar estes processos, podem fornecer importantes informações que incrementam os relatórios de gerenciamento, como por exemplo: medidas de tempos sobre os processos, quantos passos faltam para a realização de um processo, etc. [TAM 97]

### 2.3.4 Teletrabalho

Outro campo frequentemente confundido com CSCW é o teletrabalho. Na verdade, teletrabalho representa uma área de estudos distinta, e constitui também um grande campo para a aplicação de *groupwares* e outros produtos da pesquisa em CSCW. Mais ainda, o assunto constitui uma das grandes motivações para a pesquisa em CSCW.

“Trabalho a distância” é a expressão utilizada pela maioria para traduzir o significado de teletrabalho. No entanto, não existe uma única definição aceita universalmente, e muitas pessoas utilizam outras expressões, como *homeworking*, trabalho flexível, trabalho remoto, etc. [BRI 95]

Teletrabalho, que em inglês também é designado por *telecommuting*, é um termo cunhado no início dos anos 70 por Jack Nilles, para descrever um tipo de ambiente onde os trabalhadores estão dispersos geograficamente, trabalhando em seus lares ou em outros escritórios, comunicando-se com um escritório central e enviando dados através de linhas telefônicas. [AOL 99]

O teletrabalho representa uma verdadeira revolução na maneira como as pessoas trabalham. A expressão “revolução da informação” é comumente utilizada para sintetizar as mudanças que vêm ocorrendo nos últimos anos nos sistemas produtivos. Antes da revolução industrial a maioria dos trabalhadores vivia e produzia em pequenos vilarejos. O surgimento das grandes fábricas fizeram com que os trabalhadores se deslocassem para viver em cidades e localidades próximas ao local de trabalho. A revolução da informação traz uma nova abordagem, onde boa parte das pessoas trabalham sobre informações, ao invés de objetos físicos. Com as facilidades de comunicação, algumas pessoas passam a poder trabalhar em qualquer lugar, sem a necessidade de deslocar-se todos os dias para um ambiente de trabalho centralizado. [BRI 95]

## 2.4 Classificação

Sendo CSCW uma área de pesquisa bastante abrangente, diversos esforços foram realizados no sentido de obter uma classificação para as aplicações na área, não

existindo porém um consenso sobre o assunto [ARA 95]. Algumas propostas de classificação são apresentadas nesta seção.

#### 2.4.1 Classificação Espaço / Tempo

Idealizada por DeSanctis e Gallupe<sup>1</sup>, esta classificação é largamente utilizada e divide os ambientes de CSCW em quatro classes, caracterizadas na Tabela 2-1:

|                           | <b>Mesmo tempo</b>               | <b>Tempos diferentes</b>         |
|---------------------------|----------------------------------|----------------------------------|
| <b>Mesmo espaço</b>       | Interação face a face            | Interação assíncrona             |
| <b>Espaços diferentes</b> | Interação distribuída assíncrona | Interação distribuída assíncrona |

**Tabela 2-1 – Classificação espaço / tempo para os ambientes em CSCW.**

**Fonte: [KAL 99]**

Esta classificação reforça a idéia de que esta área de pesquisa abrange os mais diversos ambientes operacionais. Implicações importantes desta abrangência são visíveis tanto no aspecto social como técnico. No aspecto social, uma das grandes preocupações na implantação de *groupwares* refere-se à diferença entre o encontro físico e a interação através de ambientes virtuais, mesmo que em tempo real. A aceitação dessas tecnologias depende muito de quanto a simulação de ambientes consegue realmente aproximar as pessoas. No aspecto técnico, os ambientes distribuídos trazem problemas associados à transmissão de grandes volumes de dados e à coordenação eficiente das atividades realizadas pelos participantes. [ARA 95]

#### 2.4.2 Classificação Considerando a Previsibilidade

Uma extensão à classificação espaço / tempo é proposta por Grudin [GRU 94], considerando o fator de previsibilidade das dimensões. A Tabela 2-2 mostra esta classificação trazendo aplicações representativas para cada classe.

As atividades podem ocorrer no mesmo espaço, em espaços diferentes mas conhecidos pelos participantes (como nas correspondências via correio eletrônico), ou em inúmeros lugares, muitas vezes desconhecidos pelos participantes (como na emissão de uma mensagem a um grupo de notícias eletrônico). A mesma divisão ocorre com o

tempo: em uma reunião, por exemplo, as atividades ocorrem em tempo real. No envio de um correio eletrônico, o tempo em que este é recebido é previsível (um ou dois dias). Existem ainda as atividades que ocorrem sem previsão de tempo, como a edição colaborativa de documentos. [GRU 94]

|   | <b>Mesmo tempo</b>  | <b>Tempos diferentes mas previsíveis</b>    | <b>Tempos diferentes e imprevisíveis</b> |
|---|---|---|--|
| <b>Mesmo espaço</b>                       | Suporte a reuniões e sistemas de suporte à decisão em grupo | <i>Work shift</i>                           | Salas de discussão para grupos.          |
| <b>Espaços diferentes mas previsíveis</b> | Áudio e vídeo conferências                                  | Correio eletrônico                          | Edição colaborativa                      |
| <b>Espaços diferentes e imprevisíveis</b> | Seminários de interação <i>multicast</i>                    | <i>Bulletin boards</i> ou <i>newsgroups</i> | <i>Workflow</i>                          |

**Tabela 2-2 – Ambientes em CSCW considerando previsibilidade.**

**Fonte: [GRU 94]**

Grudin salienta que esta classificação é “*fácil de entender, facilita a comunicação e é amplamente utilizada, especialmente por desenvolvedores de groupware*”, mas que esconde um risco: diversas atividades dos ambientes produtivos não são perfeitamente associados a nenhuma categoria. Uma aplicação desenvolvida para suportar determinada atividade em uma célula da tabela, pode provocar um impacto negativo quando aplicada a outra célula, inviabilizando sua utilização em determinada instituição. Por esta e outras razões, alguns autores criticam esta classificação e clamam por suporte completo, em qualquer dimensão de espaço e tempo [ARA 95].

### **2.4.3 Outras Classificações**

Kaliannan [KAL 99] considera mais duas abordagens para classificação. A primeira leva em conta a **dinâmica dos grupos**, e classifica as aplicações de acordo com os seguintes critérios:

- ✓ Nível do grupo: considera principalmente o número de entidades que formam o grupo.
- ✓ Padrão de acesso dos membros: considera como os grupos são formados e

---

<sup>1</sup> Em seu artigo “A foundation for the study of group decision support systems”. *Management Science*, 33, 5, 589-609, 1987. Referenciado por Grudin [GRU 94].

destruídos e como a composição do grupo evolui no tempo.

- ✓ Disseminação das informações: considera os padrões de comunicação e interação entre os membros e os tipos de informações trocadas.

A segunda abordagem classifica as aplicações de acordo com três critérios de **escala**:

- ✓ Escala espacial: caracteriza a dispersão geográfica dos participantes.
- ✓ Escala numérica: considera o número de entidades em uma aplicação distribuída, tratando normalmente de um mesmo tipo de entidade (pessoas, nós na rede, escritórios, etc.)
- ✓ Escala organizacional: leva em conta as necessidades geradas pela implantação das tecnologias nas organizações, como segurança de dados, por exemplo.

## 2.5 CSCW e Internet

### 2.5.1 Massa Crítica

Uma das principais causas de insucesso das aplicações em CSCW é o problema da *massa crítica* de usuários. Este problema é relacionado com a questão do custo / benefício da aplicação. Considerando o caso de um usuário em particular, o custo devido à utilização de uma aplicação distribuída tende a ser constante, independente do número total de usuários da mesma. No entanto, o benefício obtido por este usuário cresce juntamente com o número total de usuários. O ponto onde o benefício começa a superar o custo é chamado de *ponto de massa crítica* (Figura 2-2).[DIX 96]

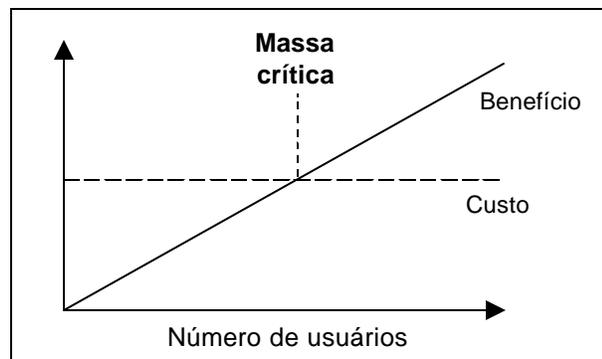


Figura 2-2 – Custo / benefício dos sistemas colaborativos.

A idéia de atingir o maior número possível de usuários interessados sugere a utilização da Internet e da *Web* como plataformas para as aplicações em CSCW. A Internet, sem dúvida, ultrapassou o ponto de massa crítica, sendo esse um dos fatores para o seu próprio sucesso e para as aplicações que ela suporta, como a *Web*, o correio eletrônico, etc.

### **2.5.2 Internet e *World Wide Web***

É importante salientar a diferença existente entre Internet e *World Wide Web*, frequentemente utilizados como termos equivalentes. O termo Internet refere-se a um grande número de tecnologias associadas, como TCP/IP e a própria *Web*, e a uma vasta estrutura que une redes de todo o mundo [LAA 97]. A *Web*, por sua vez, constitui uma das grandes aplicações que operam sobre a Internet.

Segundo Alain Dix [DIX 96], muitos sistemas são rotulados como “*baseados na Web*”, quando na verdade estabelecem conexões de Internet independentes com servidores especializados ou com outros clientes. Muitas vezes esta independência é necessária e sadia para a aplicação, mas os termos corretos devem ser aplicados. Uma aplicação baseada na *Web* é construída para executar sobre *browsers*, utilizando os protocolos de operação e transferência de dados compatíveis com os mesmos.

O fato de uma aplicação para a Internet não ser baseada na *Web* não impede que esta goze dos benefícios trazidos pela rede, principalmente no que diz respeito à alcançabilidade e independência de plataforma. A aplicação desenvolvida como resultado final deste trabalho utiliza a Internet, mais especificamente o protocolo TCP/IP, como plataforma de comunicação e transmissão de dados, mas não é baseada na *Web*. No entanto, como será visto adiante, o projeto foi totalmente voltado para a obtenção de independência da plataforma operacional, visando alcançar usuários tanto em redes locais quanto à distância.

### **2.5.3 Razões para a Utilização da Internet**

A maior parte das tecnologias de *groupware* existentes, como as aplicações construídas sobre o Lotus Notes, fornecem recursos que atendem as necessidades das organizações para as quais são destinadas. Quando se deseja cruzar os limites de uma

organização, os problemas tendem a ser muito maiores, envolvendo integração e interoperabilidade entre diferentes plataformas operacionais. [BEN 97a]

Neste contexto, a Internet oferece uma infraestrutura globalmente acessível e independente de plataforma operacional. Por esta razão, muitos desenvolvedores passaram a ver a Internet e a *Web* como ambientes promissores para a criação de aplicações colaborativas [DIX 96]. Bentley *et al.* [BEN 97a] [BEN 97b] enumeram algumas características que alimentam esse interesse:

- ✓ A Internet, com seus serviços de correio eletrônico, seus navegadores *Web* e demais serviços, está disponível para praticamente todas as plataformas computacionais e sistemas operacionais, provendo acesso às informações e aplicações que hospedam a partir de qualquer lugar do mundo.
- ✓ Muitas organizações já possuem seus próprios servidores *Web* ou seu endereço eletrônico na Internet, facilitando a implantação e aceitação de ferramentas baseadas na *Web* ou na própria Internet.
- ✓ Disponibilizar a Internet para os membros de uma instituição está cada vez mais fácil, tanto no aspecto financeiro quanto tecnológico. Financeiramente, os serviços de acesso estão cada vez melhores e mais baratos, em função da crescente concorrência. No campo tecnológico, entre outras facilidades, os sistemas operacionais passaram a integrar o acesso à Internet, praticamente eliminando a necessidade de instalação de outros *softwares* para esta finalidade.

#### **2.5.4 Tecnologias de CSCW sobre a Internet**

Laamanen [LAA 97] enumera diversas classes de aplicações que são, ou poderiam ser, construídas sobre a Internet. Um dos aspectos interessantes da Internet diz respeito ao espaço. A mesma tecnologia pode ser aplicada para abranger somente uma instituição, através de uma Intranet, ou para ter um alcance global.

## Tecnologias Assíncronas

Segundo Laamanen, estas aplicações adaptaram-se à Internet logo no seu início, sendo mais tolerantes às deficiências de largura de banda e latência da rede. Os principais exemplos são:

- ✓ **E-mail:** provavelmente é a aplicação mais antiga relacionada com CSCW. Sem dúvida, o *e-mail* atingiu o ponto de massa crítica em certos grupos de usuários, como universitários, funcionários de grandes empresas, etc.
- ✓ **Sistemas de conferência:** são aplicações para discussão assíncrona em grupo, através de mensagens de texto que ficam armazenadas em um servidor. *Newsgroups*, grupos de discussão e *bulletin boards*, são outras denominações para a tecnologia.
- ✓ **Groupware e Workflow:** a maioria dos sistemas existentes nesta área foi constituída para *mainframes* e redes locais, mas a utilização da Internet é uma tendência fortíssima.
- ✓ **Calendários de Grupos:** surgiram com os primeiros sistemas operacionais multi-usuários. A maioria existente utiliza as tecnologias antigas e estão ligados a *workflows* e outros *groupwares*.

## Tecnologias Síncronas

Tecnologias síncronas suportam interações em tempo real, incluindo áudio e vídeo. As aplicações baseadas em texto são as menos exigentes em termos de taxas de transferência, funcionando plenamente via Internet. Os recursos de transmissão ainda são insuficientes para suportar vídeo em tempo real, mas o futuro é promissor. O áudio em tempo real já atinge hoje um nível satisfatório, chegando a funcionar plenamente quando o tráfego da rede favorece. Alguns representantes:

- ✓ **Sistemas de co-autoria e projeto:** não existem padrões na Internet para suportar compartilhamento de documentos e outros objetos, mas esta tem sido utilizada para suportar soluções proprietárias.
- ✓ **Sistemas de chat:** surgiram logo no início do ambiente UNIX, com sistemas como

o WRITE e o TALK. Depois do IRC (*Internet Relay Chat*), surgiram as salas de *chat*, que operam sobre os *browsers* e são extremamente comuns atualmente.

- ✓ **Reuniões a distância e Vídeo conferência:** apesar dos problemas ainda existentes devido às limitações de desempenho da rede, é possível visualizar a Internet como plataforma padrão para este tipo de aplicação.
- ✓ **Tele presença, avatars e realidade virtual:** aplicações nesta área procuram recriar ambientes para pessoas interagindo de locais remotos. *Avatars* são utilizados para representar as pessoas na rede em ambientes de realidade virtual. Dependendo de áudio e vídeo em tempo real, estas aplicações ainda sofrem com as limitações de transferência da Internet.

## 2.6 Estágio de Desenvolvimento

Com objetivo de fornecer uma visão sobre o estágio atual em que se encontra o desenvolvimento na área de CSCW, esta seção traz uma breve descrição de alguns produtos e de um projeto de pesquisa. Dentre os produtos citados, o Lotus Notes é tido atualmente como a principal ferramenta para a constituição de soluções em *groupware*. Também são descritos o Microsoft NetMeeting, um sistema bastante completo em termos de colaboração a distância, e o Novell GroupWise, que permite a personalização de seus serviços através da programação sobre mais de 20 API's.

No campo da pesquisa o exemplo apresentado é o projeto BSCW, do centro de pesquisas GMD, que tem sua importância reconhecida tanto na bibliografia pesquisada, quanto nas páginas sobre CSCW na Internet. Sua principal característica é o suporte ao compartilhamento de informações sobre a *Web*.

### 2.6.1 Lotus Notes

O Lotus Notes pode ser caracterizado como uma ferramenta de *groupware* que atende prontamente a diversos requisitos na área de trabalho colaborativo. Além disso, trata-se de uma plataforma de desenvolvimento para soluções adequadas à realidade de cada organização.

O produto foi concebido a partir de um modelo de *groupware* desenvolvido pela Lotus (Figura 2-3), que procura tratar os diferentes tipos de informações e tarefas de gerenciamento inerentes aos ambientes colaborativos. Este modelo define três áreas de tecnologia: comunicação, colaboração e coordenação. A **comunicação** é baseada no correio eletrônico. A **colaboração** habilita os grupos de usuários a compartilharem informações através de fóruns ou áreas de trabalho comuns. A **coordenação**, também chamada de automatização de *workflow*, envolve a automatização de processos nas organizações. [TAM 97]

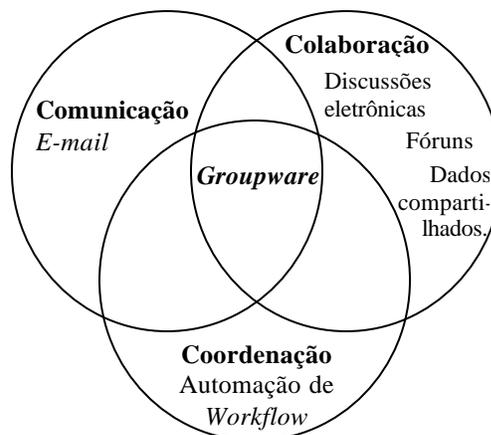


Figura 2-3 – Modelo de *groupware* da Lotus.

Fonte: [TAM 97]

Em [LOT 96] são descritos os principais recursos das versões 4.x do Lotus Notes, que atendem aos requisitos das três áreas citadas acima:

#### Base de Dados de Documentos

O elemento básico em um bando de dados no Notes é um **documento**. A estrutura de um documento é definida por uma tabela, que contém determinado número de campos, cada um armazenando uma informação, estruturada ou não. Os documentos podem ser agrupados hierarquicamente, ou seja, cada documento pode conter um conjunto de documentos filhos. O usuário navega pelos documentos através das **visões**, que são flexíveis para possibilitar vários tipos de ordenação, expansão ou não dos documentos pais e outras configurações.

O Notes pode armazenar diversos tipos de documentos ainda não atendidos por bases de dados relacionais ou outros sistemas de bancos de dados. Devido à diversidade dos documentos, uma série de recursos e facilidades são fornecidas:

- ✓ Suporta formatações complexas de textos e objetos multimídia, agindo como um ponto de acesso central para todas as informações em uma corporação.
- ✓ Fornece uma máquina de pesquisa de texto, que permite aos usuários indexar e procurar documentos através de expressões chaves.
- ✓ Fornece um sistema de controle de versões, que viabiliza a alteração do mesmo documento por vários usuários, sem que um sobrescreva as alterações de outro. O controle de versões pode ser estendido para atender às necessidades particulares dos grupos de trabalho.
- ✓ Possibilita a criação de *links* para documentos de maneira simples, por ser um sistema baseado em hipertexto.

### Replicação

O Lotus Notes permite que um documento armazenado em um servidor seja duplicado em outro servidor remoto em qualquer lugar do mundo, através da Internet. Desta forma, os usuários possam trabalhar em diferentes lugares sem a necessidade de acessar uma base de dados remota. O mais interessante é que os documentos replicados são sincronizados pela ferramenta, de maneira bi-direcional e eficiente.

### Segurança

Através de criptografia e outras facilidades, o Lotus Notes provê quatro níveis de segurança sobre os documentos compartilhados: autenticação, controle de acesso, privacidade a nível de campo de documento e assinaturas digitais.

### Troca de Mensagens (*Messaging*)

A capacidade de troca de mensagens é utilizada pelo Lotus Notes em quatro níveis diferentes:

- ✓ no nível mais básico, como recurso para comunicação pessoal, através de um completo sistema *e-mail*;
- ✓ na colaboração entre os membros dos grupos;

- ✓ como um componente crítico para aplicações de *workflow*, onde existem necessidades de notificação de usuários ou atualização de documentos em função de eventos ocorridos no processo;
- ✓ como uma plataforma para a composição de calendários de grupos e agendas.

### Ambiente para o Desenvolvimento de Aplicações

O Lotus Notes inclui um ambiente de desenvolvimento integrado, que atende a desenvolvedores com diversos graus de experiência. Com este recurso, o Notes pode ser utilizado para coordenar processos estratégicos nas instituições, gerenciando o fluxo de trabalho das equipes que utilizam os documentos armazenados. Aplicações desenvolvidas neste ambiente, localizado em um servidor, são imediatamente acessíveis pelos clientes ou até mesmo replicáveis para outros servidores.

O projeto de uma aplicação Notes é baseado em cinco elementos básicos: **formulários, campos, visões, agentes e pastas**. Os agentes constituem entidades lógicas criadas pelos desenvolvedores para automatizar tarefas, tanto no cliente quanto no servidor. Por exemplo, um agente pode ser criado para monitorar o recebimento de mensagens de *e-mail* e repassar para uma lista de colegas as mensagens contendo determinada expressão no campo “assunto”.

O ambiente inclui uma série de ferramentas de desenvolvimento, além dos elementos básicos citados acima. A mais importante é o *LotusScript*: uma linguagem de programação orientada a objetos, que provê acesso a todos os serviços do Notes e inclui um *debugger* e um *object browser* integrados.

### Outras características

As características citadas acima estão longe de esgotar a descrição sobre a plataforma, mas são suficientes para dar uma idéia sobre sua filosofia de trabalho e sobre sua abrangência. O artigo da Lotus descreve ainda outros recursos, entre os quais estão a total integração com a Internet, o suporte à múltiplas plataformas e o suporte à computação móvel.

A partir da versão 4.5, o Lotus Notes foi dividido em duas partes: o cliente, que continua sendo chamado de **Notes**, e o servidor, que passou a ser chamado **Domino**. Por

praticidade, o Lotus Domino pode ser considerado a mesma entidade que o antigo *Notes Server*, mas a evolução é bastante significativa. [TAM 97]

## 2.6.2 Microsoft NetMeeting

Segundo Keizer [KEI 97], que realizou um estudo comparativo com quatro ferramentas colaborativas, o NetMeeting foi o único a apresentar os seis recursos mais desejáveis para as aplicações nesta área: áudio, vídeo, transferência de arquivos, compartilhamento de documentos e aplicações, *chat* e *whiteboard*. Além disso, o produto tem a vantagem de ser gratuito. As principais características do produto, que roda em ambiente Windows e atualmente está na versão 3, são enumeradas na página de informações do produto [MIC 99] e resumidas a seguir:

- ✓ **Áudio e vídeo conferência:** permite que as pessoas conversem via Internet como se estivessem face a face. O recurso é limitado a duas pessoas, mas o usuário pode facilmente alternar entre várias conversas [KEI 97].
- ✓ **Whiteboard:** permite a colaboração em tempo real através de informações gráficas.
- ✓ **Chat:** conversas em tempo real em modo texto com quantos participantes se desejar.
- ✓ **Diretório na Internet:** mantido pela Microsoft, este *site* permite localizar e contatar pessoas por toda a rede.
- ✓ **Transferência de arquivos:** permite que arquivos sejam enviados em *background* durante as conferências.
- ✓ **Compartilhamento de aplicações:** permite que o usuário compartilhe uma ou mais aplicações durante uma conferência, mantendo o controle sobre a forma como cada aplicação é utilizada.
- ✓ **Compartilhamento remoto de desktop:** permite a um usuário operar remotamente um computador.

### 2.6.3 Novell GroupWise

O produto da Novell constitui uma ferramenta cliente-servidor para colaboração e comunicação, que provê o gerenciamento dos mais diversos tipos de informações. Na versão atual, 5.5, o servidor roda sobre WindowsNT Server 4.0, Unix, NetWare e IntraNetWare. O cliente está disponível para Windows NT 4.0, Windows 3.1/95/98, Unix e Mac OS. Essas informações são extraídas da página do produto na Internet [NOV 99], onde são apresentadas as principais características do produto, algumas delas resumidas a seguir:

- ✓ **Mailbox Universal:** um único *mailbox* gerencia todos os tipos de informações: *e-mail*, faxes, planilhas, documentos, imagens, *workflows*, calendários de grupos, etc.
- ✓ **E-mail:** provê comunicação com usuários do sistema e com todos os usuários de *e-mail* mapeados pelos chamados *GroupWise Gateways*.
- ✓ **Acesso, conversão, integração e segurança de documentos:** entre outros recursos, documentos podem ser copiados de um servidor para serem alterados em uma estação remota. Os documentos são organizados em bibliotecas, que são gerenciadas por uma janela específica. Tarefas, compromissos, mensagens de *e-mail* e outras informações são convertidas diretamente para documentos nas bibliotecas.
- ✓ **Colaboração sobre a Internet:** sistemas *GroupWise* podem conectar-se dinamicamente via Internet para trocar documentos.
- ✓ **Agenda e calendário interativos:** o calendário pessoal pode ser configurado para alertar o usuário sobre os compromissos e para deslocar ou remarcar automaticamente as tarefas ainda não finalizadas. A agenda do grupo pode ser acessada por todos os membros para facilitar a marcação de tarefas.
- ✓ **Gerenciamento de mensagens baseado em regras:** para auxiliar na organização das mensagens, regras predefinidas podem ser aplicadas sobre cada mensagem recebida. O sistema pode tomar uma ou mais ações de acordo com o conteúdo da mensagem.

Além dos recursos oferecidos, mais de 20 API's estão disponíveis para que todo o gerenciamento de documentos possa ser personalizado de acordo com as necessidades

específicas de cada indivíduo, grupo ou organização. O *GroupWise* inclui ainda o suporte a *workflow*, onde as tarefas são enviadas e recebidas como se fossem mensagens de *e-mail* convencionais.

#### 2.6.4 Projeto BSCW

O projeto BSBW (*Basic Support for Cooperative Work*) é desenvolvido na unidade de pesquisa FIT (*Institute for Applied Information Technology*), sediado pelo Centro Nacional de Pesquisa da Alemanha – GMD.

##### Visão Geral

A principal meta do projeto é a integração de serviços de colaboração com os ambientes já existentes, suportando grupos de trabalho com qualquer dispersão geográfica e com diferentes infraestruturas computacionais. O sistema desenvolvido é baseado na noção de uma área de trabalho compartilhada, estabelecida pelos membros dos grupos para organizar e coordenar seu trabalho. A área de trabalho compartilhada é um repositório de informações, acessível pelos membros através de um esquema simples de identificação e senha. [BEN 97b]

Os objetos armazenados na área de trabalho podem ser de diversos tipos, como documentos, tabelas, gráficos, planilhas e *links* para páginas na *Web*. O sistema mantém os membros do grupo constantemente informados a respeito das ações dos demais integrantes sobre objetos da área de trabalho. [BSC 98]

Um servidor BSCW pode gerenciar diversas áreas de trabalho. Em cada uma, os objetos de informação, como são chamados os diversos tipos de documentos, são armazenados em uma hierarquia de pastas. Os usuários podem transferir objetos de suas máquinas para o servidor e especificar restrições de acesso quanto à visualização e quanto às operações executáveis sobre os mesmos. [BEN 97a]

Para prover acesso a diversas plataformas e larga alcançabilidade, o BSCW foi totalmente constituído sobre a *Web*. O servidor do BSCW constitui a extensão de um servidor *Web* através da programação em CGI [BEN 97a].

## O Serviço de Eventos

Este serviço mantém os usuários informados sobre as atividades dos outros membros em uma área de trabalho. Cada operação, como criação de uma pasta, remoção de um objeto, etc., gera um evento específico. Os eventos mais recentes são representados por uma lista de ícones na área de trabalho. [BEN 97b]

## Registro de Membros

O acesso a uma área de trabalho é restrito, inicialmente, a usuários registrados em determinado servidor. Novos membros podem ser adicionados através de um “convite” de algum membro já existente. É possível se configurar o sistema para permitir que um novo usuário se cadastre em uma área de trabalho sem ser convidado. [BEN 97b]

## Outros Serviços

Além das funções básicas de gerenciamento de objetos, envolvendo criação de pastas, cópia, transferência, etc., o servidor BSCW foi idealizado para permitir funções mais avançadas de colaboração. O autoria colaborativa de documentos é uma necessidade em diversos ambientes. Esta necessidade é atendida no projeto através de um sistema de gerenciamento de versões. [BEN 97a]

O suporte para reuniões a distância e conferências baseadas em texto ganharam destaque nas versões mais recentes do sistema. Um agenda com as atividades do grupo está à disposição dos usuários. [BSC 98]

O projeto mantém um servidor público que, até o final de 1998, mantinha cerca de 8000 pessoas registradas. [BSC 98]

## **2.7 Considerações Finais**

A área de estudos introduzida neste capítulo se apresenta como um campo extremamente fértil para a aplicação de sistemas computacionais. Os sistemas já desenvolvidos vão de encontro às necessidades de vários setores da sociedade, incluindo organizações industriais e comerciais, ambientes acadêmicos, etc.

Uma das fortes tendências observadas é o desenvolvimento de aplicações que vão além dos limites das organizações, permitindo interação e colaboração independente da situação geográfica. Neste ponto, a Internet se apresenta como uma alternativa excelente, provendo uma plataforma de rede que já integra boa parte dos ambientes informatizados em todo o mundo.

Os produtos e projetos observados durante a pesquisa, demonstram a viabilidade de utilização da Internet como suporte para a comunicação entre servidores e clientes. Por outro lado, aplicações puramente baseadas na *Web* parecem não constituir uma tendência muito forte. Esta tecnologia tende a limitar o poder das aplicações, que muitas vezes realizam tarefas complexas, que vão além do gerenciamento de informações.

A pesquisa mostrou que a utilização de agentes inteligentes em CSCW ainda está dando os primeiros passos. Pode-se dizer que as pretensões dos agentes incluídos em algumas aplicações, como no Lotus Notes, ainda são tímidas, diante das possibilidades oferecidas por esta tecnologia.

## 3 Agentes Inteligentes

### 3.1 Considerações Iniciais

O objetivo deste capítulo é introduzir alguns conceitos sobre a tecnologia de agentes inteligentes, que constitui uma importante ramificação da inteligência artificial (IA). Recriar aspectos do comportamento humano em máquinas é um dos grandes objetivos da IA. Certamente, muitos esforços neste sentido contribuíram para o surgimento e desenvolvimento da tecnologia de agentes. Em síntese, a motivação central desta tecnologia é a construção de entidades computacionais autônomas, capazes de realizar tarefas em benefício de usuários ou de outras entidades do seu relacionamento, como outros agentes.

Como será visto neste capítulo, agentes têm sido utilizados com sucesso em uma grande quantidade de aplicações, que incluem desde tarefas corriqueiras, como a monitoração do correio eletrônico, até atividades críticas, como o controle do tráfego aéreo. Através dos agentes, os frutos de anos pesquisa em IA deverão ficar mais próximos da grande massa de usuários, cujas necessidades já não são totalmente satisfeitas pelo modelo clássico de aplicações.

Wooldridge e Jennings [WOO 95] lembram que a IA é alguma vezes definida como “*o campo das ciências da computação que visa a construção de agentes que exibem aspectos do comportamento inteligente*”. Apesar do conceito de agente ser central para a IA, até meados dos anos 80 os pesquisadores da IA tradicional deram relativamente pouca importância ao assunto. A partir daí, no entanto, o interesse cresceu muito. Agentes são atualmente discutidos por pesquisadores de vários campos das ciências da computação, incluindo comunicação de dados, sistemas concorrentes, robótica e projeto de interfaces.

## A Terceira Revolução da Informática

Chorafas [CHO 98] cita uma predição muito interessante feita pelo Dr. Alan Kay, que tem sob seu crédito a utilização do termo *desktop* como uma metáfora para denotar a área de trabalho no vídeo do computador. Dr. Kay acredita que o final deste século está sendo marcado pelo início da terceira grande revolução nas ciências da computação. A primeira teria ocorrido no início dos anos 50, com a invasão institucional dos *mainframes* nas corporações de todo o mundo. A segunda revolução conduziu a computação à herança dos microcomputadores, com suas interfaces baseadas em janelas e ícones. O final dos anos 90 assiste ao surgimento crescente de ambientes computacionais cada vez mais poderosos, que unem intimamente três elementos: computadores, comunicações e *softwares* inteligentes.

Este novo paradigma difere bastante do que se está acostumado atualmente. Ao invés de apenas utilizar os *softwares* dedicados a funções específicas, que recheiam os computadores pessoais, o usuário será servido por agentes, com os quais irá interagir para alcançar seus objetivos. Chorafas acrescenta às colocações de Dr. Kay, que o novo conceito de *software* será caracterizado por agentes instruídos pelo usuário. A habilidade de aprendizado pode fornecer ao usuário uma assistência verdadeiramente inteligente. Isto pode ser atingido com o aumento das capacidades de inferência e de relacionamento dos agentes.

Ainda segundo Chorafas, *“a diferença chave entre os agentes e as ferramentas atualmente disponíveis, é que o agente observa e se comunica com o usuário, enquanto uma ferramenta tradicional, como um objeto passivo, é algo que o usuário enxerga e manipula”*.

Caglayan e Harrigon [CAG 97] reafirmam que a tecnologia de agentes pode ser utilizada profissionalmente nos dias de hoje, e que se tornará ainda mais poderosa e útil nos próximos anos. Segundo os autores, a tecnologia de agentes tem sido alvo de grande especulação e publicidade. Os excessos na repercussão do assunto deixam o público em geral, e até mesmo os desenvolvedores de aplicações, com a impressão de que esta tecnologia ainda depende de anos de investimento até poder ser realmente utilizada. Entretanto, inúmeras experiências pelo mundo demonstram que a tecnologia possui uma

base sólida e pode colaborar imediatamente para a evolução dos sistemas computacionais em diversos setores.

### O Surgimento da Tecnologia

O conceito de agentes é fruto de cerca de 40 anos de pesquisa em IA e robótica. Em meados dos anos 70, a idéia de uma entidade de *software* capaz de realizar tarefas em favor de um usuário já era bem estabelecida. Esta pesquisa trouxe para a computação diversos conceitos inerentes à inteligência humana, como raciocínio, representação do conhecimento e aprendizagem. Entretanto, a aplicação prática de agentes tem uma origem mais pragmática. Durante anos foram desenvolvidas aplicações para automatizar tarefas específicas, até se chegar aos agentes, que trazem uma abordagem mais genérica para a resolução de problemas, integrando aplicações e bases de dados, inclusive através das redes [CAG 97].

Os agentes têm atraído a atenção dos desenvolvedores por uma série de razões:

- ✓ Constituem um meio de se introduzir inteligência nas interfaces, possibilitando que os usuários, por mais inexperientes, tirem máximo proveito das aplicações [CAG 97].
- ✓ Podem ser utilizados para personalizar aplicações e serviços, indo de encontro às preferências, metas e desejos dos usuários [CAG 97].
- ✓ Podem gerenciar a recuperação, disseminação e a pesquisa de informações através das redes de computadores, especialmente a Internet [CAG 97].
- ✓ São aplicáveis de várias maneiras em sistemas de comércio eletrônico, agindo em favor de consumidores e fornecedores [CAG 97], [JEN 96].
- ✓ Não beneficiam apenas os indivíduos, podendo auxiliar no funcionamento mais eficiente de organizações como um todo [JEN 96].

## **3.2 Definição**

A tecnologia de agentes tem sido foco de interesse cada vez maior por parte de pesquisadores e desenvolvedores, nas mais diversas áreas de aplicação. Entretanto, não

existe atualmente um conceito universalmente aceito sobre o termo “agente” [WOO 95], [JEN 96], [CAG 97]. Os autores envolvidos nesta área de pesquisa fornecem várias definições, quase sempre baseadas no conjunto de características que cada autor considera fundamentais para que um *software* venha a ser chamado de agente. Este conjunto varia bastante de autor para autor, deixando a impressão de que cada um formula sua definição de acordo com os exemplos de agente que tem em mente [FRA 96].

A falta de consenso não constitui necessariamente um problema: se muitas pessoas estão desenvolvendo aplicações úteis e interessantes, não é tão importante o fato delas não concordarem na terminologia utilizada. No entanto, existe o perigo de que o termo “agente” seja utilizado indiscriminadamente, provocando certa falta de clareza dentro da comunidade de pesquisa no tratamento da tecnologia [WOO 95].

Como não existe uma definição universal, é interessante se examinar os conceitos lançados por alguns autores, no intuito de formar uma idéia própria sobre o que diferencia um agente de um *software* convencional.

Wooldridge e Jennings [WOO 95] fornecem dois conceitos:

- ✓ A noção mais fraca utiliza o termo agente para denotar um *hardware* ou (mais usualmente) um *software* que seja dotado de autonomia para a realização de suas tarefas, habilidade para interagir com outros agentes e entidades, e reatividade ao meio em que está inserido.
- ✓ A noção mais forte é mais utilizada por pesquisadores da inteligência artificial. Nesta definição, um agente é um *software* que, além das propriedades citadas acima, são planejados ou implementados através de conceitos usualmente aplicados aos seres humanos, como conhecimento, crença, intenção e obrigação.

Franklin e Graesser [FRA 96], após reverem conceitos lançados por diversos autores, formalizaram os requisitos essenciais que constituem um agente na seguinte definição: “*um agente autônomo é um sistema situado em um ambiente, fazendo parte, percebendo e alterando este ambiente no decorrer do tempo, adequando-se à sua própria agenda e afetando sua percepção do ambiente no futuro*”.

Segundo Genesereth e Ketchpel [GEN 94], a engenharia de *software* baseada em agentes foi desenvolvida para facilitar a criação de programas capazes de operar entre si, em ambientes dinâmicos e heterogêneos. Um agente seria então “*um componente de software que se comunica com outros pela troca de mensagens em uma linguagem de comunicação. Agentes podem ser tão simples quanto subrotinas, mas normalmente constituem grandes entidades, com algum tipo de controle de persistência*”.

Goodwin [GOO 93], define um agente como “*uma entidade criada para executar uma tarefa ou um conjunto de tarefas*”. Para o autor, qualquer propriedade de um agente deve ser definida com base nas suas tarefas e no ambiente em que ele atua.

Coen [COE 94] considera os agentes como entidades autônomas, capazes de agir em favor dos interesses dos usuários. Agentes devem ser robustos e capazes de manipular informações particulares com segurança. Agentes tendem a ser altamente interativos, participando ativamente do seu universo computacional, reagindo às alterações e provocando mudanças neste universo.

De acordo com Foner [FON 93], um *software* deve obedecer a um conjunto rígido de critérios para ser considerado um agente. Foner considera crucial a presença de noções como autonomia, capacidade de personalização, discurso, cooperação, confiança e outras.

Caglayan e Harrison [CAG 97] baseiam sua definição em dois atributos extraídos do conceito genérico de agente: um agente realiza coisas; um agente age em favor de alguém ou de algo. Os autores definem agente como “*uma entidade computacional que realiza autonomamente as tarefas delegadas pelo usuário*.”

As definições citadas acima não chegam a constituir grandes divergências em relação ao que caracteriza um *software* como um agente. O que se observa é uma faixa de variação relativamente grande com respeito às características consideradas importantes na visão de cada autor. Em todas as definições, no entanto, parece clara a imagem de uma entidade autônoma, que age em favor do usuário ou de outros agentes. Talvez seja este um dos aspectos mais importantes do desenvolvimento desta tecnologia.

A motivação primeira para a criação de agentes não deve ser a adequação aos critérios de agência<sup>2</sup> considerados essenciais por determinados autores. Trata-se da busca por uma tecnologia que, embora não esteja ainda firmemente constituída, fornece idéias avançadas sobre a interação entre usuários e computadores. Esta busca deve partir da verificação das novas necessidades, geradas, em grande parte, pela explosão no número de usuários de sistemas computacionais que se observa atualmente. Um *software* não é considerado um agente somente porque passa por uma espécie de *check list* com as características desejáveis, embora isto também seja importante. O essencial é que o *software* se comporte como um assistente eficiente, que livre o usuário da realização de tarefas passíveis de automatização e que, até mesmo, viabilize a realização de tarefas operacionalmente impraticáveis antes do seu advento.

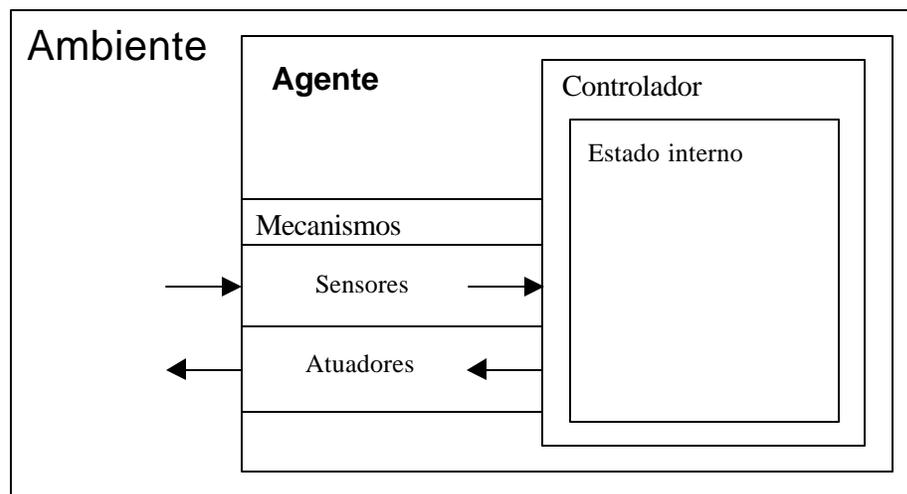


Figura 3-1 – Uma visão da arquitetura de agente.

Fonte: [GOO 93]

Para complementar o entendimento das definições apresentadas, a Figura 3-1 mostra uma arquitetura genérica para um agente, proposta por Goodwin [GOO 93]. O agente está inserido em um ambiente, com o qual interage através dos *mecanismos* de interação, os *sensores* e *atuadores*. Esses mecanismos possibilitam que o agente perceba o ambiente e provoque alterações no mesmo. Os mecanismos fazem a interface do ambiente com outra parte do agente: o *controlador*. Este recebe informações dos sensores e as utiliza para, possivelmente, alterar o *estado interno* do agente. Os *atuadores* são acionados pelo *controlador* quando é necessário enviar mensagens para o ambiente.

<sup>2</sup> A agência é o termo utilizado para designar a propriedade que um *software* apresenta quando é um agente

A representação de Goodwin sugere a idéia de que o conceito de agente é uma evolução natural do conceito de objeto, da programação orientada a objetos. Fazendo-se uma analogia, os mecanismos de interação corresponderiam aos serviços ou métodos de um objeto. O estado interno do agente corresponderia ao conjunto de atributos, que descrevem o estado de um objeto. O *controlador* é a porção que de fato diferencia o agente de um simples objeto. Este componente dá vida ao agente, independente do seu grau de inteligência e da complexidade das tarefas que realiza.

### 3.3 Propriedades

Como visto até aqui, as definições de agente encontradas na literatura são geralmente baseadas nas propriedades que diferem um agente de um *software* convencional. Esta seção apresenta algumas dessas características. Estas podem ser consideradas essenciais ou não, dependendo do autor. No entanto, é praticamente um consenso que características como *autonomia*, *comunicabilidade* e *reatividade* são altamente desejáveis, senão obrigatórias, para que um *software* desempenhe o papel de agente. A própria noção intuitiva de um *software* que faz coisas em favor do usuário, sugere a necessidade dessas capacidades especiais.

#### 3.3.1 Autonomia

Jennings e Wooldridge, em [JEN 96], dizem que “*agentes devem ser capazes de realizar a maior parte de suas tarefas sem a intervenção direta de humanos ou de outros agentes, e devem possuir um grau de controle sobre suas próprias ações e sobre seu estado interno.*”

A idéia de que esta propriedade é obrigatória é reforçada por Foner [FON 93], quando diz que “*qualquer agente deve ter um certo grau de autonomia em relação ao seu usuário*”. De outra forma, segundo Foner, ele seria apenas um programa com passos fixos e determinados pelas ações do usuário. A partir de determinado grau de autonomia, o agente pode contar com uma agenda própria, independente do usuário. Para tanto, o agente deve implementar as noções de *ação periódica*, *execução espontânea* e *iniciativa*. Com essas noções, o agente é capaz agir de forma preemptiva

(por iniciativa de uma entidade externa, como o usuário ou outro agente), ou de forma independente, visando a realização das tarefas que lhe foram atribuídas.

De acordo com Caglayan [CAG 97], um agente opera sem a intervenção direta do usuário para realizar as tarefas por este delegadas. A autonomia de um agente pode variar deste iniciar um processo de *backup* noturno, até negociar o melhor preço de um produto para o usuário.

### **3.3.2 Delegação**

A autonomia de um agente está condicionada às tarefas que lhe são delegadas pelo usuário. Um agente realiza tarefas em favor de um usuário ou de outros agentes e suas iniciativas devem ser explicitamente aprovadas pelo seu usuário [CAG 97].

### **3.3.3 Comunicabilidade e Habilidade Social**

Os agentes devem ter a capacidade de interagir com os usuários e com outras entidades de *software*, visando resolver os problemas associados às suas próprias tarefas e também auxiliar outros agentes quando for viável [JEN 96].

A interação com usuários e outros agentes é essencial para que tarefas possam ser delegadas a um agente. Esta característica é também necessária para que o agente possa manter informados os interessados sobre o progresso das tarefas que executa. Interfaces com os usuários e linguagens de comunicação entre agentes são recursos essenciais para garantir a comunicabilidade dos agentes [CAG 97].

### **3.3.4 Reatividade**

As propriedades de *reatividade* e *pró atividade* podem ser consideradas ferramentas essenciais para que os agentes possam ser dotados de autonomia. Segundo Wooldridge e Jennings [WOO 95], os agentes devem ser capazes de perceber o ambiente em que estão inseridos e responder rapidamente às mudanças que ocorrem no mesmo. O ambiente a que os autores se referem pode incluir o mundo físico, o usuário (através de uma interface gráfica), outros agentes ou a Internet.

O conceito de reatividade está relacionado com a capacidade de **monitoração**, citada por Caglayan e Harrison [CAG 97] como uma das propriedades essenciais na sua definição de agente: *‘Um agente deve ser capaz de monitorar seu ambiente para conseguir realizar suas tarefas autonomamente.’*

Na arquitetura proposta por Goodwin [GOO 93], mostrada na Figura 3-1, mecanismos denominados *sensores* ajudam a compor a interface do agente com o seu ambiente. Seu papel é informar ao *controlador* sobre qualquer alteração relevante no ambiente, para que o agente possa reagir de maneira adequada.

### 3.3.5 Pró atividade

De acordo com Jennings e Wooldridge [JEN 96], um agente não deve agir somente em resposta às alterações sentidas em seu ambiente, devendo apresentar também iniciativa própria, agindo oportunamente em favor das metas que deseja alcançar.

Além de monitorar as alterações no seu ambiente, o agente deve ser capaz de provocar alterações no mesmo. Caglayan e Harrison [CAG 97] consideram a **atuação** uma capacidade essencial para a realização autônoma das tarefas. A arquitetura de Goodwin (Figura 3-1) define mecanismos *atuadores* na camada de interface do agente, para que o *controlador* possa provocar alterações no ambiente.

### 3.3.6 Robustez

Meyer [MEY 88] define robustez como *“a habilidade dos sistemas de software funcionarem mesmo sob condições anormais”*.

Embora esta seja uma característica altamente desejável em qualquer tipo de *software*, sua importância cresce ainda mais quando se trata de agentes. Coen [COE 94] diz que a importância desta propriedade vem principalmente do fato de que os agentes agem de forma autônoma, monitorando o ambiente e agindo sobre o mesmo. *“As tarefas realizadas pelos agentes são presumidamente importantes e estes devem ser capazes de responder a alterações inesperadas em seu mundo computacional.”*

### 3.3.7 Inteligência

Segundo Caglayan e Harrison [CAG 97], um agente realiza suas tarefas através das atividades de *monitoração* e *atuação* no seu ambiente. O agente deve ser capaz de interpretar os eventos monitorados para decidir suas próximas atuações em busca da realização das tarefas. Esta capacidade constitui, segundo os autores, a inteligência do agente, essencial para o seu comportamento autônomo.

Já Auer [AUE 95], considera um tanto inapropriado definir a inteligência simplesmente como uma propriedade de decisão. Para realizar suas tarefas, o agente realiza algum tipo de “tradução”, procurando representar cada tarefa através de um conjunto de ações a serem tomadas. Durante este processo, o agente pode se deparar com várias ambiguidades. A inteligência do agente consistiria então na sua capacidade de lidar eficientemente com estas ambiguidades.

### 3.3.8 Aprendizagem

Goodwin [GOO 93], divide os agentes em dois grupos: deliberativos e não deliberativos (ou reflexivos). Agentes deliberativos possuem internamente um modelo do seu mundo computacional, possivelmente incluindo eles mesmos. O agente deliberativo utiliza este modelo para avaliar o efeito das ações possíveis em determinado passo da resolução de um problema. Os agentes reflexivos, por outro lado, não contam com um modelo do seu mundo, representando seu conhecimento através de tabelas, que mapeiam as ações a serem tomadas em resposta a cada evento percebido.

Segundo Goodwin, a aprendizagem é uma propriedade inerente apenas aos agentes deliberativos, na verdade, aos mais avançados dentro deste grupo. Esta propriedade consiste na capacidade do agente modificar, por si próprio, a representação do seu mundo computacional.

### 3.3.9 Outras Características

O conjunto de propriedades apresentado acima, representa apenas uma parte das características inerentes aos agentes. Dependendo da área de aplicação, da complexidade dos problemas a serem resolvidos pelos agentes e das técnicas

empregadas no projeto dos mesmos, determinadas características podem assumir papel crucial, ou podem ser simplesmente dispensadas. As propriedades apresentadas devem ser suficientes para o escopo deste trabalho e suas descrições colaboram para o entendimento das definições citadas na seção anterior.

Wooldridge e Jennings [WOO 95] definem também as propriedades de *mobilidade*, *veracidade*, *benevolência* e *racionalidade*, que constituem seu conceito mais forte de agência.

Auer [AUE 95] traz discussões sobre *mobilidade*, *responsabilidade*, *obrigação* e *confiança*.

Foner [FON 93] define as características de *cooperação*, *capacidade de personalização*, *degradação suave*, *risco*, *confiança* e *domínio*.

Franklin e Graesser [FRA 96] acrescentam a propriedade de *caráter*, que visa dotar os agentes de certa personalidade e estado emocional.

Para finalizar, Coen [COE 94] caracteriza os agentes como entidades “*não invariantes em relação ao tempo*”, podendo alterar as ações que são tomadas em determinadas situações. Tais alterações são baseadas na memória do agente sobre os efeitos das suas ações anteriores.

### **3.4 Classificação**

Buscar uma forma consensual para classificar os agentes não é uma tarefa simples, a exemplo do que ocorre com o problema da definição. A tecnologia de agentes não possui limites conhecidos, sendo aplicada a áreas cada vez mais diversas. Cresce também a variedade de formatos com que a tecnologia se apresenta. Esses fatores contribuem grandemente para a falta de classificações aceitas universalmente.

Esta seção apresenta as propostas de classificação de alguns autores. Essas propostas não são necessariamente excludentes, podendo até ser consideradas complementares, por envolverem diferentes componentes da tecnologia, como ambiente, natureza das tarefas e arquitetura.

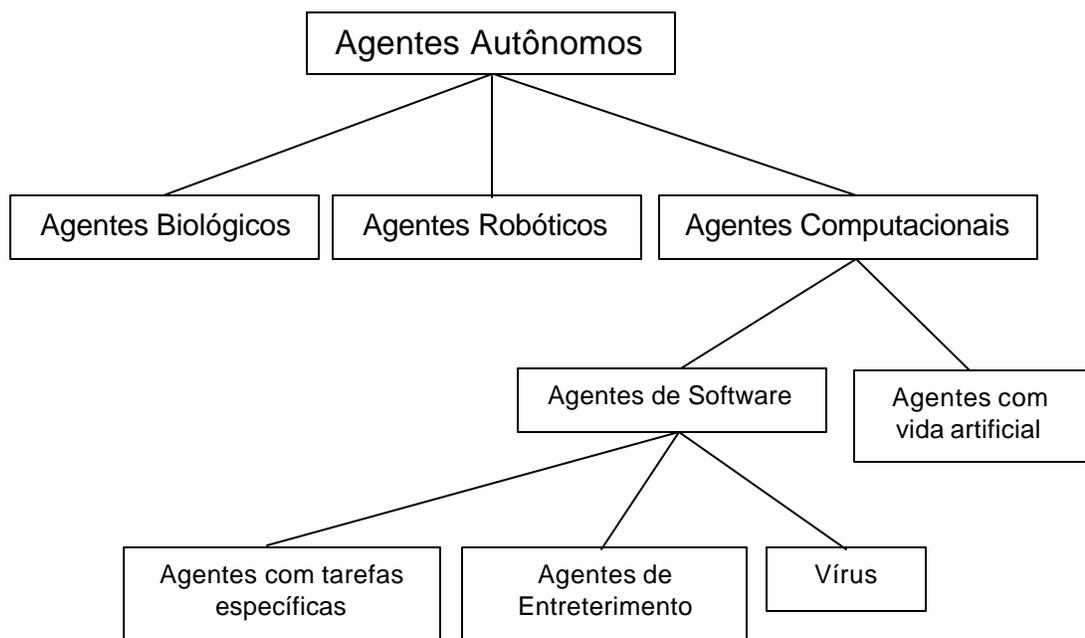
### 3.4.1 Franklin e Graesser, 1996

Algumas estratégias interessantes de classificação são apresentadas por Franklin e Graesser [FRA 96]. A primeira delas é baseada nas propriedades dos agentes, como as apresentadas na seção anterior. Segundo a definição dos autores, as propriedades de *reatividade*, *autonomia*, *pró atividade* e *continuidade temporal* são comuns a todos os agentes. A partir daí, a adição de outras propriedades produz classes particulares, como agentes móveis, agentes comunicantes, agentes aprendizes, etc.

Outra classificação citada por Franklin e Graesser divide inicialmente os agentes em três grupos, que podem ser novamente refinados em uma segunda camada:

- ✓ **Agentes reativos:** utilizam sensores para monitorar diversos “sinais de entrada” e sempre sabem o que fazer quando estes são recebidos.
- ✓ **Agentes planejadores:** solucionam os problemas através da elaboração de um plano de ação.
- ✓ **Agentes adaptativos:** não só planejam, como aprendem. Alteram seus planos em função das experiências passadas.

#### Classificação dos Agentes Segundo sua Natureza



**Figura 3-2 – Classificação dos agentes segundo sua natureza.**

Mais uma proposta de Franklin e Graesser consiste na classificação dos agentes baseada em um modelo biológico. Este modelo tem a forma de uma árvore, onde a raiz representa os seres vivos e as folhas representam as espécies. Cada nível da árvore traz um critério de classificação, por exemplo: reino, filo, classe, ordem, família, etc.

A classificação mostrada na Figura 3-2 segue essa idéia. No nível **reino**, os agentes são classificados em *biológicos*, *robóticos* e *computacionais*. No nível **filo** os agentes computacionais são classificados em *agentes de software* e *agentes com vida artificial*. No nível da **classe** os *agentes de software* são classificados em *agentes com tarefas específicas*, *agentes de entretenimento* e *vírus*. Esta classificação pode se estender ainda mais, de acordo com a necessidade de especificação de quem utiliza o modelo.

### 3.4.2 Caglayan e Harrison, 1997

Em [CAG 97], os autores consideram muitas vezes confusos os termos utilizados na literatura para classificar os agentes (agente de Internet, *wizard*, agente de pesquisa, etc.), já que estes termos referem-se a diferentes componentes da tecnologia. Por isso, antes de se iniciar uma classificação, são definidos os aspectos mais importantes da natureza dos agentes:

- ✓ **Ambiente:** os agentes são projetados para atuarem em determinados ambientes, como a *Web*, o sistema operacional, uma aplicação, etc. Denominações como *agente de Internet* e *agente de desktop*, são baseadas neste aspecto.
- ✓ **Tarefa:** agentes com tarefas específicas normalmente são nomeados com base na natureza de suas atividades. Por exemplo, *agentes de pesquisa* e *agentes de e-mail*.
- ✓ **Arquitetura:** segundo este aspecto, “*os agentes são rotulados de acordo com a arquitetura do seu conhecimento interno*”. *Agentes aprendizes* e *agentes neurais* são exemplos desta terminologia.

Em seu livro, Caglayan e Harrison decidem classificar os agentes de acordo com seu ambiente, dividindo cada categoria de acordo com as tarefas. A Tabela 3-1 mostra esta classificação.

| Ambiente            | Tarefa                     | Descrição   |
|---------------------|----------------------------|---|
| Agentes de Desktop  | Agentes de S.O.            | Provêm assistência ao usuário no sistema operacional.   |
|                     | Agentes de aplicação       | Auxiliam o usuário na utilização de aplicações específicas  |
|                     | Agentes de pacotes         | Facilitam a utilização de pacotes compostos de diversas aplicações.   |
| Agentes de Internet | Agentes de pesquisa        | Fornecem serviços de pesquisa na <i>Web</i> .   |
|                     | Filtros de informações     | Agentes que filtram as informações na Internet de acordo com a preferência do usuário.                                  |
|                     | Recuperação de informações | Trazem para o <i>desktop</i> do usuário pacotes personalizados com informações, de acordo com as preferências do mesmo. |
|                     | Agentes de notificação     | Notificam o usuário sobre eventos de seu interesse verificados na Internet.   |
|                     | Agentes servidores         | Situam-se em um endereço fixo da <i>Web</i> para prover serviços específicos.   |
|                     | Agentes de serviço         | Provêm serviços especializados aos usuários.  |
|                     | Agentes móveis             | Transportam-se de um lugar para outro para executar suas tarefas.   |
| Agentes de Intranet | Agentes colaborativos      | Automatizam processos de <i>work-flow</i> em empresas.  |
|                     | Agentes de automação       | Também são dedicados à automação de processos de <i>work-flow</i> .   |
|                     | Agentes de bancos de dados | Provêm serviços específicos para usuários de bancos de dados.   |
|                     | Agentes de recursos        | Realizam a alocação de recursos em arquiteturas cliente/servidor.   |

Tabela 3-1 – Classificação de agentes por ambientes e tarefas.

### 3.4.3 Wooldridge e Jennings, 1995

Em [WOO 95], os agentes são divididos de acordo com a arquitetura empregada na sua construção. Três classes são apresentadas:

#### Agentes Deliberativos

Um agente deliberativo é definido como “*um agente que contém uma representação explícita, um modelo simbólico do seu mundo, e cujas decisões (...) são tomadas através de um raciocínio lógico (ou pelo menos pseudo lógico), baseado em reconhecimento de padrões e manipulação simbólica.*”

Esta arquitetura é fruto do paradigma conhecido como *inteligência artificial simbólica*, que vê os agentes como tipos especiais de sistemas baseados em

conhecimento. Embora a idéia de se representar agentes somente através de raciocínio lógico seja atraente, surgem duas questões importantes:

- ✓ Tradução: como representar o mundo real em uma descrição simbólica adequada?
- ✓ Representação e raciocínio: como representar simbolicamente informações sobre entidades e processos do mundo real, e como fornecer aos agentes o raciocínio sobre estas informações?

Segundo Wooldridge e Jennings, apesar de todo esforço direcionado à resolução dessas questões, ambas estão longe de uma solução definitiva. A dificuldade inerente à prova de teoremas, até mesmo para lógicas simples, associada à complexidade dos algoritmos de manipulação simbólica em geral, tornam impraticável a idéia de se construir agentes como “*provadores de teoremas*”.

### Agentes Reativos

Devido às dificuldades com a representação simbólica do mundo real, pesquisadores têm buscado técnicas alternativas, como a idéia de agentes reativos. Wooldridge e Jennings definem esses agentes como entidades “*que não incluem nenhum tipo de modelo simbólico do mundo e não utilizam raciocínio simbólico complexo*”. Um dos maiores problemas desta arquitetura é a falta de uma metodologia de desenvolvimento associada, como a IA simbólica para a arquitetura deliberativa.

### Arquiteturas Híbridas

Segundo Wooldridge e Jennings, vários pesquisadores têm explorado a construção de sistemas híbridos, unindo a arquitetura deliberativa com o esquema alternativo de agentes reativos. Uma abordagem possível seria construir o agente através de dois (ou mais) subsistemas: o **deliberativo**, que contém o modelo do mundo real, seria encarregado de atividades como planejamento e tomadas decisão. O subsistema **reativo** seria capaz de prover uma resposta rápida aos eventos importantes detectados no ambiente, sem nenhuma espécie raciocínio complexo.

Arquiteturas híbridas são bastante exploradas e possuem algumas vantagens sobre as arquiteturas puramente deliberativas ou reativas. Um dos grandes problemas relativos a esta estratégia é a interação entre os subsistemas deliberativos e reativos.

### 3.5 Benefícios da Tecnologia de Agentes

Caglayan e Harrison [CAG 97] fazem um levantamento sobre os benefícios trazidos pelos agentes em função das particularidades desta tecnologia:

#### Automação

Agentes podem ser utilizados para automatizar:

- ✓ comportamentos repetitivos de um único usuário;
- ✓ comportamentos similares de um grupo de usuários;
- ✓ comportamentos sequenciais e repetitivos de vários usuários num sistema colaborativo.

Um comportamento repetitivo pode estar baseado tanto no tempo, quanto nos eventos relacionados. Um comportamento baseado no tempo refere-se a *“algo que o usuário realiza num tempo em particular, como visitar determinada página da Web toda manhã às nove horas”*. Um comportamento baseado em evento *“é algo que o usuário faz em relação a outra tarefa”*. Por exemplo, abrir sempre uma planilha eletrônica depois de abrir o sistema de controle de estoque, é um comportamento baseado em evento.

#### Customização

O agente age em favor do usuário, ou do grupo de usuários. A capacidade de customização do agente aumenta os benefícios fornecidos pelo mesmo. As preferências do usuário são levadas em conta na realização das tarefas pelo agente. Um exemplo deste benefício aparece no campo da obtenção de informações.

O volume de informação disponível atualmente nos meios de comunicação de massa (TV, rádio, cabo, listas de correio eletrônico, etc.) e nos meios de publicação (jornais, revistas, páginas da *Web*, etc.) é algo incalculável. Agentes podem trazer

grandes benefícios quando posicionados entre o usuário e as fontes de informação. No caso dos meios de massa, que emitem a mesma informação a todos os receptores, um agente pode monitorar e filtrar as informações recebidas, apresentando-as ao usuário de acordo com os interesses do mesmo. Nos meios de publicação, o agente pode localizar as informações relevantes para o usuário e repassá-las quando for devido.

### Notificação

Um agente que fornece serviços de notificação pode poupar muitos esforços do usuário, monitorando eventos relevantes em suas atividades. Por exemplo, um agente pode monitorar alterações em páginas da *Web* de interesse do usuário e avisá-lo sempre que alguma novidade for publicada.

### Aprendizado

Agentes com capacidade de aprendizado podem beneficiar o usuário ou o grupo de usuários, detectando atividades que podem ser automatizadas e as preferências que podem ser customizadas.

### Tutores

Agentes que funcionam como tutores podem reduzir muito os requisitos de treinamento na utilização de sistemas. Esses agentes podem monitorar as ações do usuário e, através de sua capacidade de inferência, oferecer orientações para guiar os passos do usuário dentro de uma aplicação.

## **3.6 Aplicações de Agentes**

Agentes são atualmente a base para o desenvolvimento de uma grande variedade de aplicações, que vão desde as mais simples, como um filtro de *e-mail*, até sistemas complexos e críticos, como o controle de tráfego aéreo [JEN 98]. Esta seção introduz algumas áreas de desenvolvimento em que a tecnologia de agentes tem sido aplicada com sucesso.

## Computação Móvel

A migração dos sistemas isolados para os sistemas baseados em rede já é realidade há algum tempo. Segundo Hermans [HER 96], esta evolução gerou novas necessidades: os usuários precisam de mobilidade, acessando os recursos de uma rede a partir de qualquer lugar, independente das limitações de largura de banda e volatilidade da rede. Os agentes inteligentes, que neste caso residem na rede e não no computador pessoal do usuário, podem atender a esses requisitos, encarregando-se das solicitações do usuário. Os agentes podem processar as informações na fonte, retornando para o usuário somente os dados importantes, evitando o congestionamento da rede com grandes quantidades de dados não processados.

## Controle de Processos Industriais

Segundo Jennings e Wooldridge [JEN 98], “*o controle de processos é uma área natural para a aplicação de agentes inteligentes e sistemas multi-agentes, já que os controladores são, por natureza, sistemas autônomos e reativos.*” O maior exemplo desta aplicação é uma plataforma denominada ARCHON, destinada à construção de sistemas multi-agentes. Agentes desenvolvidos sobre o ARCHON têm sido aplicados em sistemas complexos, como transporte de energia e aceleração de partículas.

## Gerenciamento de Redes e Sistemas

Esta foi uma das primeiras áreas de aplicação a serem beneficiadas pela utilização de agentes. A migração dos sistemas computacionais para arquiteturas baseadas em rede, como o modelo cliente/servidor, aumentaram a complexidade do gerenciamento. Os usuários destes ambientes, especialmente os administradores, necessitam de ferramentas que simplifiquem o gerenciamento dos recursos. Agentes inteligentes são recursos valiosos nesta área, podendo, por exemplo, detectar e reagir a determinadas situações no sistema, gerenciar configurações dinamicamente e tratar de funções específicas, como a contabilização de recursos em uma rede. [HER 96]

## Gerenciamento de Informações Pessoais

A explosão na utilização da Internet e das tecnologias associadas, como a *Web*, trouxe enormes mudanças no campo da informação pessoal. No setor acadêmico, por

exemplo, tornou-se comum a publicação eletrônica de artigos em listas de *e-mail* e, principalmente, na *Web*. No entanto, a riqueza e diversidade das fontes de informação trazem consigo alguns problemas: muitas vezes é difícil localizar as informações apropriadas em um espaço sem limites como a *Web*. Em uma lista de *e-mail*, por exemplo, é um verdadeiro desafio filtrar as informações de interesse em meio à grande massa de informações recebidas. [JEN 96]

Esses problemas motivam a utilização de agentes inteligentes no gerenciamento de informações pessoais. Neste contexto, a meta dos agentes é incrementar a produtividade do usuário, permitindo que o usuário focalize sua atenção nas informações que realmente interessam e fornecendo o acesso a informações que seriam obtidas com muita dificuldade sem a presença desses assistentes. [JEN 96]

Especificamente nos sistemas de correio eletrônico, os usuários necessitam de vários serviços que podem ser resolvidos por agentes. Exemplos desses serviços seriam a organização automática das mensagens, de acordo com assunto ou prioridade, e a eliminação de mensagens que não são de interesse do usuário. Os agentes podem agir de acordo com as regras estabelecidas pelo usuário ou até deduzir suas regras, através da identificação de certos padrões no comportamento do usuário. [HER 96]

### Comércio Eletrônico

Atualmente, o comércio é praticado, quase que na sua totalidade, através de interações diretas entre as pessoas, que decidem quando comprar, quanto podem pagar, etc. No entanto, diversas decisões e atividades relacionadas ao comércio podem tranquilamente ser automatizadas por agentes. [JEN 98]

Segundo Hermans [HER 96], “*o comércio eletrônico é uma área em expansão, impulsionada pela popularidade da Internet*”. Os compradores precisam encontrar os fornecedores dos produtos e serviços que necessitam, precisam encontrar informações sobre produtos que atendam aos seus requisitos, e, após a compra, frequentemente necessitam de algum tipo de suporte. Os fornecedores, por sua vez, necessitam de meios para alcançar o maior número possível de compradores com seus produtos. Agentes podem ser aplicados para auxiliar tanto compradores quanto fornecedores, realizando compras para o usuário, buscando informações técnicas sobre produtos, promovendo a

venda de produtos, solucionando problemas pós-venda para os clientes, negociando o melhor preço e forma de pagamento, etc.

### Outras Aplicações

Além das aplicações brevemente discutidas acima, Jennings e Wooldridge [JEN 98] fornecem outros exemplos interessantes. Na área industrial, destacam o controle de processos, a manufatura e o controle de tráfego aéreo. Nas aplicações comerciais, destacam o gerenciamento de negócios, além do gerenciamento de informações pessoais e do comércio eletrônico, já citados acima. No setor médico, agentes são aplicados na monitoração do estado dos pacientes e na coordenação do atendimento em clínicas e hospitais. Na diversão, a tecnologia de agentes tem sido empregada em jogos e no teatro e cinema interativos.

## **3.7 Agentes em CSCW**

A aplicação da tecnologia de agentes não beneficia apenas indivíduos, contribuindo para que organizações inteiras operem de forma mais eficiente. Em diversos setores de produção, os processos têm se tornado mais complexos, envolvendo vários indivíduos, muitas vezes dispersos geograficamente, que necessitam trabalhar juntos na direção das metas da sua organização. Tradicionalmente, o gerenciamento de tais processos é feito manualmente, o que tende a trazer erros, perdas de informações e outros problemas. A aplicação de ferramentas tradicionais, como os sistemas de *workflow*, resolve em parte o problema, mas estas tendem a ser rígidas no que diz respeito à mudanças no ambiente e surgimento de eventos inesperados. [JEN 96]

Neste novo e crescente paradigma de trabalho, os usuários trabalham juntos, compartilhando documentos, utilizando recursos de comunicação e compartilhando uma série de outros recursos através das redes. Os usuários nesta área não necessitam apenas de ferramentas que viabilizem o compartilhamento eficiente de recursos. Necessitam também de funções que suportem o gerenciamento de equipes de trabalho, fazendo com que as pessoas realmente consigam produzir nesta nova situação, que envolve também dispersão geográfica e temporal. [HER 96]

Agentes inteligentes são idealizados como entidades autônomas e robustas, capazes de conduzir a realização de tarefas complexas, que lhes são atribuídas com alto grau de abstração. Estas e outras características sugerem sua aplicação em sistemas de suporte ao trabalho colaborativo.

Liu e Dix [LIU 97], reforçam a idéia de utilizar agentes inteligentes em sistemas de CSCW, visando o aumento da eficiência e qualidade no trabalho. Os autores enquadram a área de CSCW na tecnologia dos sistemas de informação. Tais sistemas são considerados como organizações ou sistemas sociais. Para que as metas de uma organização sejam atingidas, é necessário que os “atores” dentro da organização ajam de forma coordenada e colaborativa. O sucesso da organização depende de todos os seus membros. Agentes são entidades sociais, capazes de interagir com usuários, com outros agentes e com o ambiente em que estão inseridos. Esta característica vai de encontro às necessidades dos sistemas que representam organizações sociais, como ocorre em CSCW.

Independente do papel que desempenham, os agentes inseridos tornam-se essenciais “atores” nos sistemas em que são aplicados. Ainda segundo Liu e Dix, os agentes passam a atuar de maneira muito semelhante às pessoas no ambiente colaborativo. Isto reforça a crença de diversos pesquisadores da IA, de que os agentes devem ser capazes de expressar características típicas da natureza humana, como desejo, intenção e crença. Apresentando ou não tais características, é certo que os agentes devem atuar também de forma coordenada e responsável dentro do sistema, visando o sucesso da organização como um todo.

São muitos os papéis possíveis para agentes em sistemas colaborativos. Agentes podem atuar como **procuradores**, realizando ações e interagindo com outros agentes e usuários em favor de um ou mais usuários. Podem ser **mediadores**, transmitindo ou roteando mensagens. Podem ainda atuar como **coordenadores**, sugerindo ou determinando ações dentro do ambiente colaborativo [LIU 97].

Neste trabalho, agentes são utilizados como componentes fundamentais da aplicação desenvolvida, exercendo atividades relacionadas com a coordenação de atividades, compartilhamento de informações e mediação da comunicação entre os colaboradores.

## 4 Modelagem do Sistema

### 4.1 Considerações Iniciais

Este capítulo traz a elaboração de um modelo para um sistema de suporte ao trabalho colaborativo via Internet. O sistema foi batizado como **VMeeting** (*Virtual Meeting*) e sua implementação será apresentada no capítulo 5.

A estratégia empregada na modelagem pode ser representada por três atividades: a definição da abrangência do sistema, a especificação dos requisitos e o projeto do modelo computacional. As atividades ocorrem na ordem em que são apresentadas e o resultado de cada etapa serve como ponto de partida para a seguinte, até que o produto do modelo possa ser utilizado como base para a implementação.

A primeira etapa da modelagem tem como objetivo determinar a área de abrangência do sistema. Como foi possível observar no capítulo 2, CSCW é um campo bastante vasto. As possibilidades e necessidades evidenciadas são tantas, que é adequado se definir inicialmente quais atividades, dentro do trabalho colaborativo, devem ser explorados pela aplicação.

Definidas as áreas de atuação, a etapa seguinte consiste no levantamento dos requisitos para o sistema. Cada área de atuação identificada na primeira fase abrange ainda um número muito grande de necessidades específicas. As necessidades que se deseja atender com o novo sistema devem ser identificadas e sintetizadas, gerando um conjunto bem definido de requisitos. Esta etapa da modelagem é comumente chamada de “análise de requisitos”. No ciclo tradicional de desenvolvimento de sistemas, as duas etapas descritas até aqui estão situadas na fase de *análise*, que é seguida pelas fases de *projeto*, *implementação* e *validação*.

A modelagem segue com o projeto do sistema, visando atender a todos os requisitos estabelecidos. Em síntese, o projeto define a arquitetura da aplicação, a estrutura das entidades que a compõem e o relacionamento entre as mesmas. Todo o

projeto, bem como a implementação em linguagem Java, seguem o paradigma da programação orientada a objetos. Na apresentação dos modelos de classes de objetos é utilizada a notação desenvolvida por Coad e Yourdon, descrita [COA 93], assim como partes de sua metodologia de projeto. Os símbolos utilizados são descritos no Apêndice I. Durante o projeto, destaca-se a modelagem dos agentes, que são então utilizados como elementos centrais em diversos pontos da arquitetura.

As classes utilizadas na elaboração da interface do sistema não serão detalhadas neste capítulo. A idéia é tratar o projeto da interface o mais separadamente possível do restante da aplicação. As classes que representam os elementos da interface normalmente fazem parte de bibliotecas pré concebidas, que podem ser fornecidas pela ferramenta de desenvolvimento ou por outros pacotes.

## **4.2 Abrangência**

CSCW é uma área cuja abrangência exata ainda não pode ser determinada. A falta de um campo de aplicação bem definido dificulta a determinação dos requisitos para as ferramentas computacionais. Estas são aplicáveis a uma grande variedade de ambientes, onde as necessidades dos usuários, de maneira geral, não são plenamente atendidas por aplicações que apresentam soluções genéricas.

Este fator gera a primeira dificuldade nesta modelagem: a aplicação proposta não é aplicável somente a um tipo de ambiente ou organização, tendo uma abordagem mais genérica. Desta forma, torna-se interessante limitar o escopo da aplicação aos tipos de atividades que podem ser consideradas mais comuns e essenciais dentro do trabalho colaborativo. Com este objetivo, além dos estudos relatados no capítulo 2, foi tomada como base a observação de uma atividade onde o trabalho colaborativo é utilizado com muita frequência: a realização de trabalhos acadêmicos em grupos de pesquisa.

### **4.2.1 Grupos de Pesquisa**

A realização de trabalhos acadêmicos, como dissertações, teses, projetos de pesquisa e outros, envolve professores e alunos que normalmente possuem interesses comuns, no que diz respeito às áreas de conhecimento em que atuam. Nesta atividade, o mesmo professor é normalmente responsável pela orientação simultânea de vários

trabalhos. Frequentemente, os trabalhos orientados pelo mesmo professor são relacionados, já que são originados pelas linhas de pesquisa em que este atua. O trabalho colaborativo surge como uma opção para que os esforços dos orientandos, e dos próprios orientadores, sejam melhor coordenados na busca do aumento da produtividade e da qualidade dos trabalhos. A necessidade de colaboração é um dos grandes fatores que incentivam a criação de grupos de pesquisa.

A afinidade dos membros de um grupo no aspecto científico, dificilmente se repete no que diz respeito a horários e locais de trabalho. É frequente, inclusive, a composição de grupos com pessoas situadas em diferentes instituições de ensino. A dispersão temporal e geográfica do grupo de trabalho traz a necessidade da utilização de ferramentas que viabilizem o desenvolvimento dos trabalhos com o menor número possível de reuniões presenciais.

A comunicação constante entre os membros dos grupos, a realização de reuniões a distância, o compartilhamento de informações, a autoria colaborativa de documentos, os fóruns de discussões, são apenas algumas das necessidades inerentes a esta atividade, que podem ser suportadas por uma ferramenta colaborativa.

#### **4.2.2 Equipes Virtuais**

O conceito de equipe virtual também traz aspectos importantes a serem observados no projeto de uma ferramenta colaborativa. As pessoas envolvidas em atividades suportadas por ferramentas colaborativas estão inseridas em um novo mundo, onde os conceitos de distância e tempo assumem outras dimensões. Segundo Lipnack e Stamps [LPI 97], a maioria das pessoas nunca recebeu um treinamento sobre como viver e trabalhar nesta nova realidade. Por isso, são necessários novos modelos de equipes que incorporem a idéia de trabalho em grupo independente de tempo e espaço.

Ainda segundo Lipnack e Stamps, três palavras capturam a essência do sucesso de equipes virtuais: **pessoas**, **propósitos** e *links*.

Pessoas estão presentes em grupos de qualquer tipo e de qualquer nível, desde comitês executivos até associações de pais de uma escola.

Para as equipes, os propósitos representam as tarefas a serem realizadas, ou as metas compartilhadas entre as pessoas participantes.

Os *links* são os canais de comunicações, as interações e as relações que sustentam a estrutura de uma equipe. A grande diferença entre o trabalho presencial e o trabalho virtual reside na natureza e variedade desses *links*.

Durante a elaboração do modelo, no decorrer deste capítulo, o modelo **peçoas / propósitos / links** deverá ser constantemente lembrado, por incorporar elementos essenciais do trabalho a distância suportado por computador.

### **4.2.3 Determinação da Abrangência do Modelo**

A atividade dos grupos de pesquisa serviu como guia básico para a identificação das linhas de atuação do sistema. No entanto, cuidados foram tomados para que os aspectos considerados fossem comuns à maioria das atividades onde o trabalho colaborativo é aplicado. Três elementos foram escolhidos:

#### Comunicação

Sistemas de suporte ao trabalho colaborativo a distância são projetados para prover uma interface a um ambiente compartilhado, onde os usuários são ligados de diversos modos, para que possam se comunicar quase como se estivessem no mesmo espaço (Gay *et al.* [GAY 97]). Uma atividade de trabalho em grupo, seja presencial ou a distância, exige que a comunicação entre os integrantes da equipe seja facilitada e o mais frequente possível. Desde uma rápida troca de idéias entre duas pessoas, até uma discussão envolvendo vários membros do grupo, a comunicação é muitas vezes essencial para a continuação de determinadas atividades.

#### Reuniões a Distância

Uma atividade produtiva precisa, desde o seu início e durante todo seu ciclo de existência, ser delineada de acordo com seus objetivos. Em diversos momentos uma atividade precisa ser discutida: os objetivos podem ser alterados, os recursos disponíveis precisam ser adequadamente aplicados, o cenário envolvendo a atividade pode ser alterado a qualquer instante, etc. As reuniões são normalmente marcos importantes no

ciclo de uma atividade. Nelas são discutidas as questões importantes para a sequência do trabalho, de maneira formal e decisiva. Quando se parte para o trabalho colaborativo a distância, a importância da realização de reuniões cresce ainda mais, já que o acompanhamento das atividades pelos envolvidos torna-se mais difícil quando a equipe está geograficamente dispersa.

### Compartilhamento de informações

Segundo Bentley *et al.* [BEN 97a], diversas razões mostram que o compartilhamento de informações está tornando-se cada vez mais necessário no âmbito do trabalho colaborativo. A tendência de descentralização das empresas, as parcerias nos negócios e o suprimento de certas funções através de serviços externos às instituições, são exemplos que salientam tal necessidade. Embora aplicações como as vídeo conferências venham a resolver alguns problemas, evidências empíricas sugerem que sistemas que provêm acesso a informações compartilhadas, em qualquer tempo e lugar, são um dos principais requisitos para os grupos de trabalho a distância.

Documentos de diversas naturezas são gerados como resultado do trabalho em grupo e precisam ser facilmente acessíveis aos membros da equipe. Além disso, os indivíduos podem obter documentos importantes, relacionados às áreas de interesse do grupo, como artigos, manuais, endereços na *Web* e outros. O compartilhamento desses documentos é um fator muitas vezes essencial para a produtividade do grupo.

## **4.3 Requisitos**

A partir das áreas de interesse identificadas até aqui, foram levantados os requisitos fundamentais para o sistema, que serão apresentados nesta seção. Além dos requisitos funcionais, também são identificadas diversas características desejáveis no sistema, como simplicidade e alcançabilidade, importantes para o sucesso de qualquer aplicação colaborativa.

### **4.3.1 Tipos de Usuário**

O sistema deverá suportar inicialmente três categorias de usuários:

- ✓ **Administrador:** é a pessoa que terá acesso às funções fornecidas pelo servidor do sistema.
- ✓ **Usuários comuns:** são as pessoas cadastradas no sistema e que deverão participar dos grupos de trabalho.
- ✓ **Coordenadores:** são usuários do sistema aos quais se atribui o papel especial de coordenar os grupos de trabalho. A atividade de coordenação envolve funções exclusivas, que visam principalmente organizar a sequência dos trabalhos dentro do grupo.

#### **4.3.2 Cadastro de Usuários e Grupos de Trabalho**

Além dos usuários, outra entidade fundamental no sistema é o **grupo de trabalho**, que reúne usuários em torno de um interesse ou objetivo comum. O primeiro requisito funcional para o sistema é disponibilizar ao administrador as funções de cadastro, envolvendo inserção, edição e exclusão, para usuários e grupos de trabalho.

Sobre os usuários deverão ser armazenadas as seguintes informações pessoais: nome, endereço, endereço de correio eletrônico e telefone. Além disso, devem ser armazenados o *login* e uma senha para acesso ao sistema.

O cadastro de um grupo de trabalho deve armazenar o nome, uma descrição, o conjunto de usuários participantes e o usuário nomeado como coordenador. Os usuários poderão fazer parte de mais de um grupo simultaneamente.

#### **4.3.3 Comunicação *On-line***

No intuito de facilitar ao máximo a comunicação entre os membros dos grupos de trabalho, foram estabelecidos os seguintes requisitos funcionais para o sistema:

##### Usuários on-line nos grupos de trabalho

O usuário conectado ao sistema deverá estar constantemente informado sobre os demais usuários conectadas nos grupos de trabalho dos quais participa.

### Troca de mensagens *on-line*

O sistema deverá oferecer a possibilidade de estabelecer uma linha de comunicação entre dois usuários que estejam conectados, para que estes troquem mensagens entre si através de texto.

### Salas de conversação para os grupos de trabalho

Além da comunicação usuário a usuário, haverá o acesso dos usuários às salas de conversação. Cada grupo deve contar com uma sala própria.

## **4.3.4 Reuniões a Distância**

A realização de reuniões a distância pode ser considerado o ponto mais importante do modelo proposto. Os requisitos expostos a seguir refletem dois aspectos fundamentais no que diz respeito às reuniões: o primeiro e mais importante refere-se à organização, envolvendo os preparatórios que viabilizam a ocorrência da reunião e ajudam a garantir o seu bom aproveitamento. O segundo trata da reunião propriamente dita, envolvendo as ferramentas necessárias para a conexão dos usuários a uma sessão *on-line* e o suporte para as discussões e tomadas de decisões.

### **4.3.4.1 Agendamento**

À medida que cresce o número de membros em um grupo de trabalho, é provável que aumente a dificuldade de conciliação de horários. A tarefa de agendar uma reunião, que aparentemente é simples, acaba tornando-se um tanto complexa. O problema cresce com a dispersão geográfica, onde a comunicação entre os membros pode seguir a passos muito mais lentos se comparada à comunicação dentro do espaço de uma organização.

O sistema deverá ser dotado de inteligência suficiente para tratar o problema de agendar uma reunião de um grupo a partir de uma solicitação do seu coordenador. Nesta versão, o coordenador deverá sugerir um conjunto de horários para a realização da reunião e determinar a forma como o sistema deve agir, dentre três formas possíveis:

- ✓ Solicitar as opiniões dos membros do grupo e informar o resultado da pesquisa ao coordenador, para que este tome a decisão sobre o horário.

- ✓ Solicitar as opiniões dos membros do grupo e tomar a decisão de acordo com a maioria.
- ✓ Marcar um horário determinado sem consultar os membros do grupo.

#### **4.3.4.2 Organização da Pauta**

Uma reunião normalmente é guiada por uma pauta previamente estabelecida, o que certamente assegura maior produtividade na sessão. O objetivo principal da pauta é descrever quais assuntos serão tratados na reunião.

A partir do momento em que uma reunião foi agendada, os usuários poderão preparar-se para a realização da mesma através da pauta. É desejável que cada membro do grupo possa adicionar contribuições à pauta e acompanhar a evolução da mesma até a realização da reunião.

Durante uma reunião, o andamento dos assuntos é frequentemente apoiado pela apresentação de documentos pelos membros da equipe. O termo “documento” é utilizado aqui para designar os diversos materiais que possam ser expostos na hora da reunião, como textos, planilhas, desenhos, páginas na Internet, vídeos e outros. No sistema proposto, os documentos ou materiais a serem apresentados na reunião devem fazer parte da pauta. É necessário que o sistema conte com ferramentas que possibilitem aos usuários incluírem seus documentos, basicamente arquivos e endereços de páginas *Web*, como anexos nos assuntos da pauta, para que estes possam ser apresentados durante a reunião.

#### **4.3.4.3 Sala de Reuniões**

Outro requisito importante para o sistema é a interface para a realização das reuniões. A proposta para a esta versão é utilizar janelas baseadas em texto, semelhantes às salas de conversação, com ferramentas especiais que permitam o gerenciamento da reunião por parte do coordenador e um melhor acompanhamento de todos os participantes.

A sala de reuniões é o ponto do sistema onde abre-se o maior leque de possibilidades em termos de interface. A evolução das redes de computadores possibilitou a difusão de importantes tecnologias aplicáveis às reuniões a distância

através dos computadores. Um exemplo dessas tecnologias são as vídeo conferências, cuja utilização deverá ser bastante viável num futuro próximo. Outra tecnologia interessante e disponível em alguns *softwares* comerciais permite a visualização e até a utilização simultânea de uma aplicação por usuários em computadores remotos.

O fato dessas tecnologias mais modernas não serem incluídas nos requisitos para a elaboração da sala de reuniões deve-se basicamente ao seguinte: o modelo inicial do sistema está mais voltado para o aspecto organizacional das reuniões, onde é requerida maior inteligência por parte do sistema. Independente da interface utilizada, os recursos que tratam da organização terão sempre um papel essencial, que não será diminuído pela tecnologia empregada.

#### **4.3.4.4 Geração de Atas**

As reuniões realizadas em uma instituição são normalmente sintetizadas através de uma ata, que fica armazenada de com diversas finalidades: informar aos interessados sobre as decisões tomadas, servir como base para a pauta de reuniões subsequentes, etc. A ata normalmente é elaborada por uma pessoa nomeada pelo grupo, que acumula o material necessário durante a reunião, para posteriormente escrever a ata e repassá-la aos participantes. As informações acumuladas podem incluir o nome das pessoas presentes, assuntos em pauta, discussões importantes, decisões, documentos apresentados, etc.

As atividades relacionadas com a ata da reunião, embora sejam relativamente simples, rendem um certo trabalho principalmente no que diz respeito ao seu armazenamento e divulgação. No caso da reunião a distância, essas atividades tornam-se menos triviais se realizadas manualmente. Daí surge outro requisito: o sistema deverá automatizar o tanto quanto for possível as tarefas de elaboração, armazenamento e divulgação das atas.

#### **4.3.5 Compartilhamento de Informações**

No trabalho a distância a atividade de o compartilhamento de informações entre os membros dos grupos de trabalho tende a ser pouco eficiente quando realizada através de ferramentas convencionais, como o correio eletrônico, por exemplo. Surge então a

idéia de dotar o sistema de ferramentas que possibilitem que os usuários construam, ao longo do tempo, bibliotecas contendo documentos de interesse comum, como artigos, manuais, tutoriais, endereços na *Web*, etc.

Quando uma pessoa localiza ou recebe um documento de seu interesse, é comum que ela procure armazenar este documento para poder acessá-lo a qualquer momento. Se a pessoa faz parte de um grupo de trabalho, ela pode querer não só armazenar o documento para si, mas também disponibilizá-lo aos demais participantes. A ferramenta de trabalho em grupo deve facilitar ao máximo essa atividade, incentivando os usuários a sempre compartilharem suas descobertas e também a acessarem as contribuições dos demais.

#### Manutenção das Bibliotecas

O primeiro requisito nesta área é possibilitar aos usuários a edição e o acesso facilitado às bibliotecas. O sistema deverá permitir a criação de várias bibliotecas, nas quais os documentos serão organizados por assuntos. Os documentos poderão ser arquivos ou endereços na *Web*.

#### Notificações aos Usuários

No intuito de incentivar e tornar mais eficiente o acesso dos usuários às bibliotecas, surge a idéia de outra funcionalidade para o sistema: os usuários deverão ser notificados sempre que forem incluídos novos documentos nos assuntos de seu interesse. Cada usuário deve poder cadastrar-se em quantos assuntos desejar dentre os disponíveis nas bibliotecas.

### **4.3.6 Outras Características**

Até aqui foram enumerados os requisitos funcionais para o sistema de apoio ao trabalho colaborativo. Antes de se iniciar a elaboração do projeto do sistema, devem ser definidas também algumas características desejáveis, relativas principalmente à utilização e à evolução do sistema.

### Alcançabilidade

Segundo Bentley *et al.* [BEN 97b], apesar dos esforços já dispendidos em um grande número de projetos buscando dar suporte ao trabalho colaborativo, poucas aplicações são utilizáveis a longa distância e a maioria é representada por protótipos em laboratórios. No que se refere a grupos de trabalho dispersos geograficamente, o correio eletrônico e os programas simples para transferências de arquivos continuam sendo o estado da arte em sistemas que suportam o trabalho colaborativo a distância.

O sistema proposto deve ser utilizável tanto em redes de pequeno alcance, como as *Intranets*, quanto à longa distância. A plataforma de rede que melhor atende a este requisito de alcançabilidade global é a Internet.

### Migração dos Usuários

Outra característica desejável é a não vinculação do usuário do sistema a uma estação de trabalho. Um mesmo usuário deve ser capaz de acessar o ambiente de qualquer computador ligado à Internet e que possua o sistema instalado, sem perdas de informações ou eventos relativos aos grupos dos quais participa.

### Simplicidade

A proposta do VMeeting é ser uma aplicação utilizada de forma contínua, fazendo parte do ambiente de trabalho dos usuários. A aceitação de uma ferramenta deste tipo depende bastante da sua simplicidade, que deve se estar presente tanto no processo de instalação quanto na utilização dos recursos.

### Estendibilidade

Meyer [MEY 88] enumera os principais fatores de qualidade de um *software*, sendo estes divididos em duas categorias. Os **fatores externos** dizem respeito aos aspectos que podem ser detectados pelos usuários do produto, como corretude, robustez, estendibilidade, reusabilidade e compatibilidade. Os fatores percebidos apenas pelos profissionais de informática, como modularidade e legibilidade do código, são ditos **fatores internos**.

Sem diminuir a importância dos demais fatores, a estendibilidade deve receber atenção especial durante a fase de projeto. Como foi dito na primeira seção deste capítulo, a grande abrangência da área de CSCW torna necessária a delimitação dos campos de interesse da aplicação. No entanto, para que a evolução da aplicação esteja garantida, a arquitetura do modelo deve estar aberta para a inclusão de novos conjuntos de requisitos.

Ainda segundo Meyer, estendibilidade é definida como “*a facilidade com que os produtos de software podem ser adaptados às mudanças nas especificações.*” Dois princípios são essenciais na busca deste fator:

- ✓ Simplicidade: uma arquitetura simples será sempre mais adaptável a mudanças se comparada a uma arquitetura mais complexa.
- ✓ Descentralização: quanto maior a autonomia dos módulos na arquitetura do *software*, maior a chance de que uma alteração simples afete somente um ou poucos módulos do sistema, ao invés de desencadear um grande número de mudanças por toda a arquitetura.

Esses princípios deverão ser perseguidos durante toda a fase de projeto.

## **4.4 Projeto**

### **4.4.1 Arquitetura**

As características do sistema proposto sugerem a uma arquitetura baseada no modelo cliente/servidor. O servidor é a unidade central do modelo, possuindo um endereço fixo na rede, através do qual diversas instâncias da aplicação cliente podem estar conectadas.

Na aplicação proposta, como ocorre frequentemente nas aplicações cliente/servidor, o servidor assume as maiores responsabilidades dentro do sistema, sendo encarregado de tarefas como as seguintes :

- ✓ realizar o processamento necessário para atender às solicitações dos clientes;
- ✓ mediar a comunicação entre as aplicações clientes;

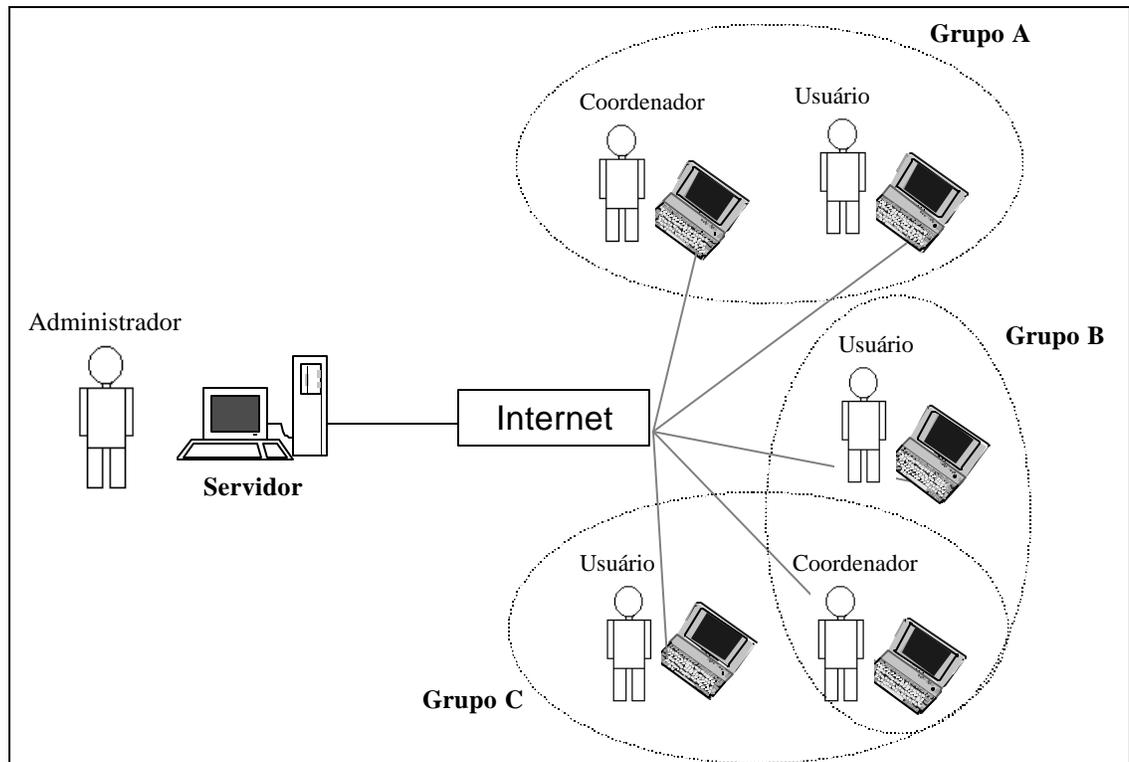
- ✓ armazenar os dados utilizados pelo sistema;
- ✓ fornecer a interface para o administrador do sistema.

Sua interface deverá ser bastante simples, por atender somente aos requisitos do administrador. No entanto, sua arquitetura interna é relativamente extensa.

O cliente, por sua vez, é caracterizado pela simplicidade no que diz respeito à arquitetura interna, já que grande parte da inteligência do sistema está associada ao servidor. A complexidade, neste caso, vai para o campo da interface. É através do cliente que os usuários (usuários comuns e coordenadores de grupos) terão acesso a todos os recursos do sistema.

A Figura 4-1 apresenta a idéia mais abstrata da arquitetura do sistema. O administrador tem acesso ao sistema através do módulo servidor, instalado em uma estação com um endereço fixo na Internet. Os usuários que compõem os grupos de trabalho acessam o sistema através da unidade cliente. A Internet é o meio de comunicação entre os clientes e o servidor. O endereço na Internet onde o servidor está instalado é o elo de ligação entre este e os clientes.

O cliente e o servidor não são aplicações convencionais. Juntas elas compõem um sistema distribuído. Este tipo de sistema é composto por entidades que, embora apresentem certo grau de independência, colaboram entre si buscando atingir os objetivos globais da aplicação. Tais entidades podem ser modeladas de diversas maneiras, como agentes, componentes distribuídos, *applets* que executam em navegadores da *Web*, aplicações convencionais e outros.



**Figura 4-1 – Visão simplificada da arquitetura do sistema.**

No caso deste modelo, as entidades são representadas principalmente por agentes, que encapsulam praticamente todas as tarefas do sistema, e pelas aplicações que fazem a interface entre os usuários e os agentes. Cada tipo de agente está localizado em um dos módulos da aplicação (cliente ou servidor), a maioria no servidor.

Para que os agentes e demais entidades que compõem o modelo possam cooperar para atender às requisições dos usuários, é estritamente necessário que se comuniquem. A comunicação entre os elementos de uma aplicação deve ser suportada por alguma espécie de protocolo ou linguagem entendida por todos.

Antes do detalhamento das arquiteturas dos módulos cliente e servidor, são apresentados dois aspectos fundamentais para o modelo: o primeiro trata da comunicação entre as entidades que compõem os módulos da aplicação. O segundo trata da definição de uma base para a construção dos agentes.

#### 4.4.2 Comunicação

Para se estabelecer a forma de comunicação entre os agentes e demais componentes do sistema, duas possibilidades devem ser previamente analisadas: o uso de um protocolo já estabelecido e a concepção de um novo protocolo.

Os protocolos de comunicação já estabelecidos apresentam como maior atração a possibilidade de se estabelecer comunicação também com componentes externos à aplicação. Um exemplo deste tipo de protocolo é a linguagem KQML (*Knowledge Query and Manipulation Language*), que está sendo bastante utilizada em aplicações multi-agentes. Um exemplo de aplicação foi desenvolvido por Faraco [FAR 98], onde agentes cooperam em um sistema para comércio eletrônico.

No entanto, certos problemas devem ser analisados antes da adoção de um protocolo. O primeiro é o fato de não existir na prática um consenso sobre qual protocolo irá tornar-se um “padrão de fato” (como o protocolo TCP/IP para a Internet). Outro problema é o custo de implementação. Adequar um sistema a um protocolo pode conduzir à escrita de bastante código de suporte, por exemplo, para tratar da interpretação das mensagens e até para a composição das mesmas.

A definição de um protocolo de comunicação dedicado, por outro lado, traz como grande vantagem a total flexibilidade quanto ao tipo de informação a ser transportada. Além disso, sua interpretação pelos componentes do sistema, pode ser bastante simples dependendo da implementação, sem a necessidade de processos de conversão muitas vezes complexos.

No caso deste modelo, a definição de um protocolo dedicado é uma boa alternativa. Não existe a necessidade dos agentes do sistema comunicarem-se com agentes externos, tornando bastante interessante a flexibilidade de um protocolo próprio. Agentes desenvolvidos posteriormente, para a mesma aplicação ou para novos fins, poderão fazer uso do mesmo protocolo, já que este, como será justificado a seguir, pode ser facilmente estendido.

#### 4.4.2.1 Recursos de Programação

O projeto do protocolo de comunicação está apoiado em dois recursos de programação disponíveis na linguagem Java.

O primeiro é denominado RTTI (*Run Time Type Identification*) e permite que um objeto seja perfeitamente identificado em tempo de execução, como o nome sugere. Esta identificação traz informações sobre a classe a qual o objeto pertence, incluindo as possíveis super classes, a lista de atributos e métodos, etc. O RTTI está embutido na especificação da linguagem Java. A maioria dos compiladores mais novos para C++ também implementam este recurso.

O segundo recurso trata da transmissão e persistência de objetos. Objetos podem ser representados em cadeias de *bits*, denominadas *streams*. Estas representam dispositivos de armazenamento ou de entrada e saída, como arquivos e portas de comunicação. Objetos inseridos em uma *stream* podem ser restaurados posteriormente a partir da mesma *stream*, ou de outra *stream* representando o mesmo dispositivo. Uma nova instância do objeto é criada quando ocorre a extração a partir de uma *stream*. Este recurso pode fazer parte da própria linguagem, pode ser fornecido nas bibliotecas que normalmente acompanham os ambientes de desenvolvimento, ou pode ser implementado diretamente pelo usuário da linguagem. No caso da linguagem Java, este recurso é oferecido diretamente pela API, através de uma *interface* específica<sup>3</sup>.

O protocolo TCP/IP fornece canais de comunicação denominados *sockets*, que serão utilizados na transmissão de mensagens na aplicação. Então, as referidas *streams*, onde os objetos são inseridos e extraídos na transmissão, são mapeamentos lógicos sobre os *sockets* alocados para a comunicação.

A Figura 4-2 esquematiza o processo de transmissão de objetos através de *streams*. A **Aplicação Emissora** possui um objeto - **Objeto A** - que deve ser transmitido para a **Aplicação Receptora**. O objeto é inserido em uma *stream* que representa a porta de comunicação (um *socket*, por exemplo) . A **Aplicação Receptora** recebe o **Objeto A** através de uma *stream* de entrada que mapeia a mesma porta de

---

<sup>3</sup> Uma *interface* Java define um conjunto de funcionalidades que podem ser implementadas por determinada classe de objetos. A *interface* denominada *Serializable*, do pacote *java.io*, define as funções relativas à persistência dos objetos. Objetos que implementam esta interface podem ser inseridos em *streams*.

comunicação. Uma instância idêntica do **Objeto A** passa a existir nesta aplicação, que decide gravar o objeto em uma base de dados utilizando o mesmo recurso.

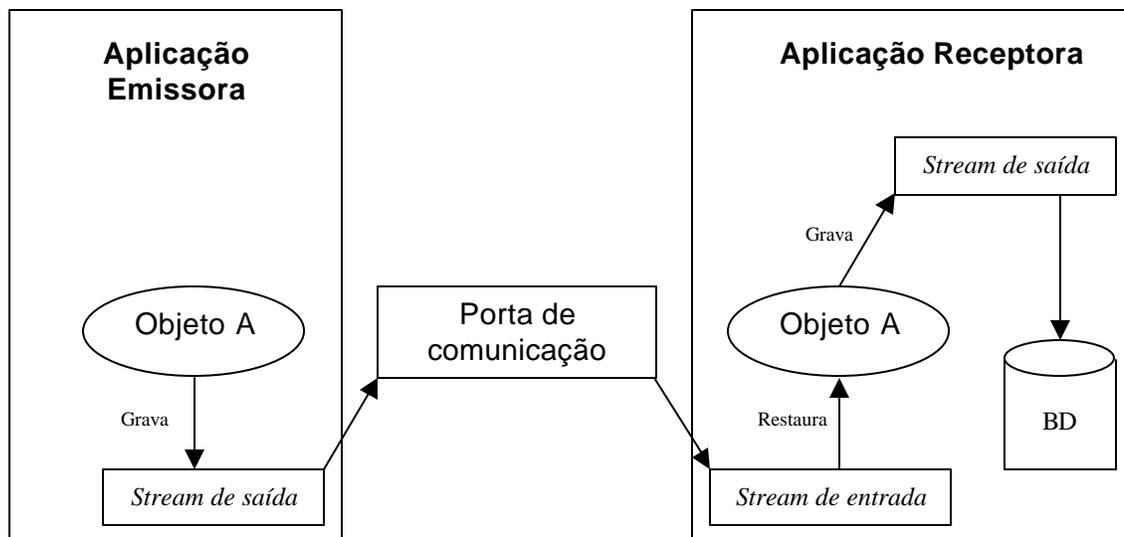


Figura 4-2 – Comunicação através streams.

A combinação desses recursos torna possível a definição do protocolo de comunicação através de uma hierarquia de classes. Uma mensagem pode ser enviada na forma de um objeto, desde que o mecanismo de transporte seja suportado por *streams*. O receptor, ao receber a mensagem, restaura o objeto e o identifica via RTTI. A informação sobre a classe à qual o objeto pertence permite o processamento da mensagem representada.

#### 4.4.2.2 Hierarquia de Mensagens

As mensagens utilizadas pelos componentes do sistema serão representadas por uma hierarquia de classes. Como vantagens desta abordagem salientam-se a flexibilidade quanto aos tipos de informações transportadas e a facilidade com que o protocolo pode ser estendido, através da definição de novas classes.

A Figura 4-3 apresenta parte da hierarquia de classes utilizada como protocolo de comunicação do sistema. Todas as mensagens são especializações da super classe **Mensagem**, que contém sempre a identificação do remetente e do destinatário da mensagem. Todas as entidades do modelo, como os agentes, possuem um identificador único, como será visto adiante neste capítulo.

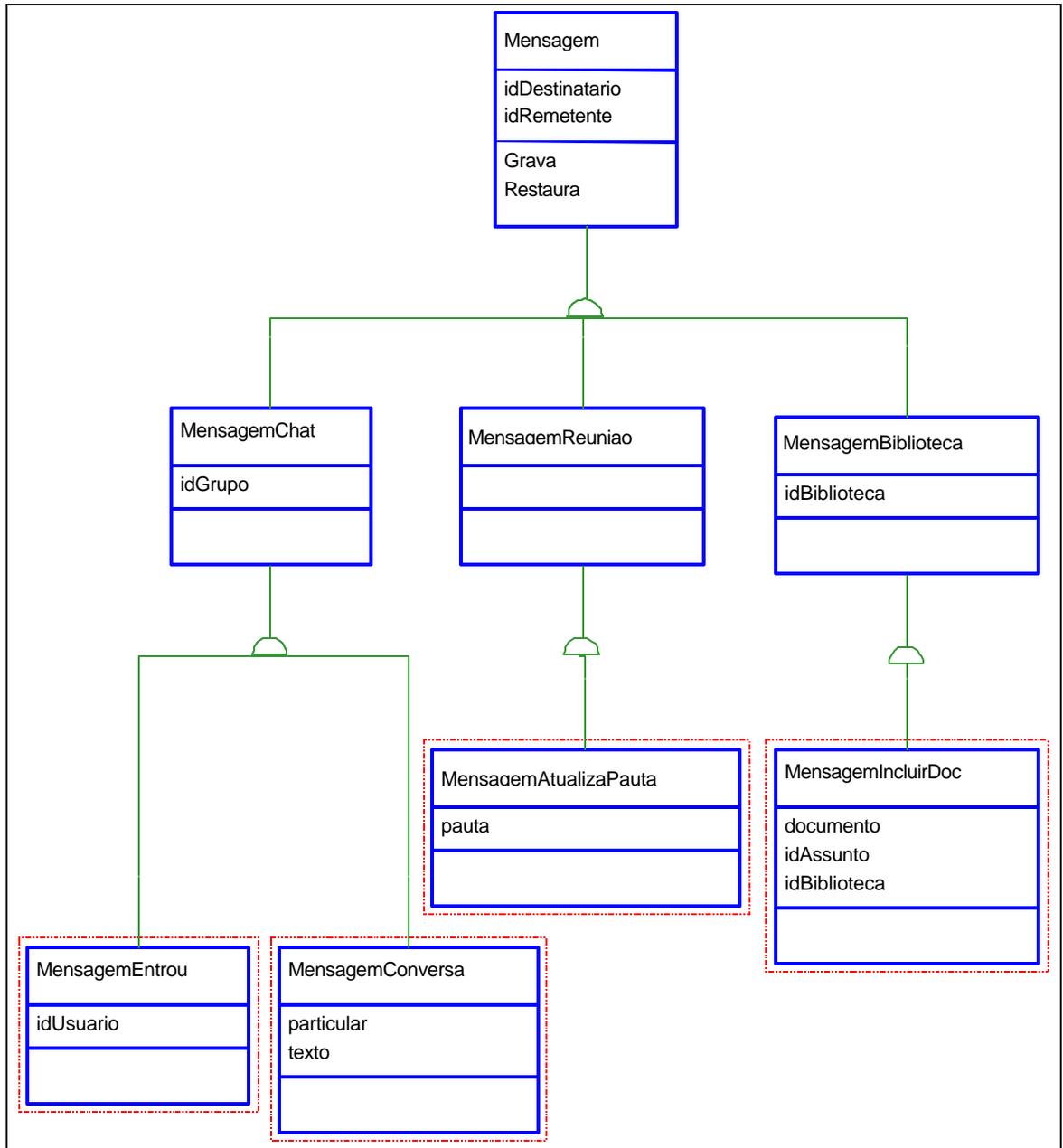


Figura 4-3 – Classes do protocolo de comunicação do sistema.

Para facilitar o roteamento e interpretação das mensagens, a hierarquia é decomposta em outras super classes, cujo papel é formar agrupamentos de mensagens relacionadas ao mesmo assunto. Por exemplo, a classe **MensagemChat** é a super classe para todas as mensagens relacionadas às atividades nas salas de chat. A classe **MensagemConversa**, por exemplo, em adição ao remetente, destinatário e identificação do grupo, transporta o texto do enviado e uma *flag* indicando se a mensagem é particular ou não.

### 4.4.3 Agentes

Os agentes são as entidades mais importantes da arquitetura do sistema, já que neles está concentrada praticamente toda a inteligência necessária à realização das tarefas. Neste modelo as aplicações principais – cliente e servidor – são decompostas em agentes. Cada agente constitui uma entidade com independência para tratar das tarefas que lhe são atribuídas. Para que os objetivos globais do sistema sejam alcançados, os agentes fornecem seus serviços para a aplicação à qual estão associados e também utilizam serviços básicos oferecidos pela mesma.

O objetivo desta seção é definir a forma de funcionamento e as estrutura dos agentes que compõem o modelo, iniciando pelo mecanismo empregado para garantir aos agentes a autonomia e o conhecimento necessários para sua atuação.

#### 4.4.3.1 Análise Estado - Evento

Dix [DIX 98] define a análise estado - evento como “*uma coleção de técnicas (formais e semi formais) baseadas nas diferenças entre **eventos** (fatos que acontecem) e **estados** (atributos que sempre possuem um valor)*”. Ainda segundo Dix, a análise estado - evento tem sido utilizada, dentre outras aplicações, em arquiteturas distribuídas baseadas em agentes.

A noção de estado é pouco considerada em aplicações de escopo local, onde a resposta aos eventos tende a ocorrer imediatamente. Porém, na interação a distância, a definição de estados é crucial, já que as interações tendem a ser lentas e o contexto em que um evento foi gerado não pode ser perdido até que todo o processamento relativo ao mesmo seja realizado.

A natureza dos requisitos funcionais do modelo desenvolvido neste trabalho sugere a utilização da análise estado – evento como guia para o funcionamento dos agentes. O passo das interações entre os usuários e o sistema é algo imprevisível. Por exemplo, o tempo decorrente entre a solicitação de uma reunião e a consulta a determinado membro do grupo sobre sua disponibilidade de horário, pode variar de poucos minutos a vários dias, dependendo da frequência com que este acessa o sistema. O armazenamento de estados por parte dos agentes possibilita sua adaptação a este tipo de situação. O estado ou conjunto de estados de um agente deve garantir que o escopo

de uma determinada tarefa seja recuperado no momento que um evento relevante ocorre. Por exemplo, quando o usuário responde sobre a solicitação de uma reunião, o agente precisa saber do que se trata a resposta recebida e como deve proceder. Neste caso, o estado do agente deve manter informações sobre o tempo em que a solicitação foi enviada ao usuário.

#### 4.4.3.2 Geração de Eventos

Os eventos são quase sempre os responsáveis por alterações de estados nos agentes. Essas alterações geralmente caracterizam o progresso dos agentes na realização de suas tarefas. Neste modelo existem dois tipos de eventos: as mensagens e os eventos temporizados.

As mensagens são eventos externos aos agentes e podem ser originadas diretamente pelo usuário ou por outros agentes que compõem a aplicação. Frequentemente as mensagens constituem requisições de serviços para os agentes ou respostas às solicitações deste.

As mensagens não fornecem elementos suficientes para que um agente realize suas atribuições. Um agente precisa ser dotado de algum tipo de iniciativa própria, precisa tomar atitudes em resposta a alterações percebidas no seu ambiente. Para tanto, são definidos os eventos temporizados. A cada agente pode ser associado um ou mais temporizadores, que geram um evento a cada intervalo de tempo configurado pelo agente. Quando o agente recebe um evento deste tipo, tem a oportunidade de tomar atitudes segundo suas observações sobre o ambiente em que está inserido. Tais observações podem envolver, por exemplo, o horário, o tempo restante para o término de uma tarefa, os usuários conectados no sistema e o próprio estado interno do agente.

#### 4.4.3.3 Requisitos para os Agentes

Antes da definição do modelo para a implementação dos agentes é necessária a observação dos requisitos essenciais para o seu funcionamento dentro da aplicação:

- ✓ **Tratamento de mensagens:** os agentes devem possuir uma forma comum de receber as mensagens, que caracterizam os eventos externos de maior importância, e tratá-las de acordo com o seu estado interno.

- ✓ **Iniciativa própria:** no intuito de resolver os problemas que lhe são atribuídos, um agente não deve depender apenas da recepção de mensagens. É necessário que ele responda também a estímulos do meio em que está inserido e que tenha inteligência suficiente para tomar as atitudes necessárias no momento certo.
- ✓ **Persistência:** os agentes não permanecem sempre ativos no sistema. Por exemplo, se o servidor da aplicação é fechado, todos os agentes associados ao mesmo são desligados. Por esta razão os agentes precisam ser persistentes, podendo ser gravados e restaurados a qualquer momento, sem perder nenhuma informação sobre seu estado interno. O mesmo mecanismo utilizado para a transmissão de mensagens através dos *sockets* pode ser aplicado para a persistência dos agentes e outros objetos em arquivos. Embora a persistência não seja útil para determinados agentes do modelo, como será visto em seguida, torna-se interessante modelar este conceito para todas as classes de agentes. Desta forma, todos podem ser tratados da mesma maneira e a extensão da persistência para todos os agentes já fica assegurada. Então, este conceito é modelado na super classe **Agente** (Figura 4-4).
- ✓ **Estendibilidade:** os requisitos estabelecidos no início deste capítulo cobrem apenas uma parcela das possibilidades para uma aplicação deste tipo. A extensão do modelo para atender a novas necessidades no futuro depende de uma estrutura que suporte a adição de novos recursos, possivelmente implementados por novos tipos de agentes. O objetivo deste modelo é permitir que a adição de novos agentes cause o mínimo de alterações na estrutura da aplicação.

#### 4.4.3.4 Modelo para os Agentes

O diagrama da Figura 4-4 mostra o modelo de classes para o desenvolvimento dos agentes. A super classe **Agente** modela os aspectos essenciais a todos os agentes. Cada instância será identificada no sistema pelo seu ID, um número inteiro. O estado interno pode ser representado de diversas maneiras, não sendo possível se determinar o tipo desta informação na super classe. Mesmo assim, o conceito é modelado na classe **Agente** para indicar a obrigatoriedade da presença deste em todos os agentes. O estado é representado de acordo com os objetivos do agente, podendo variar desde uma simples enumeração, como [*iniciando*, *aguardando\_mensagem*, *processando\_mensagem*, *finalizando*], até um conjunto complexo de informações.

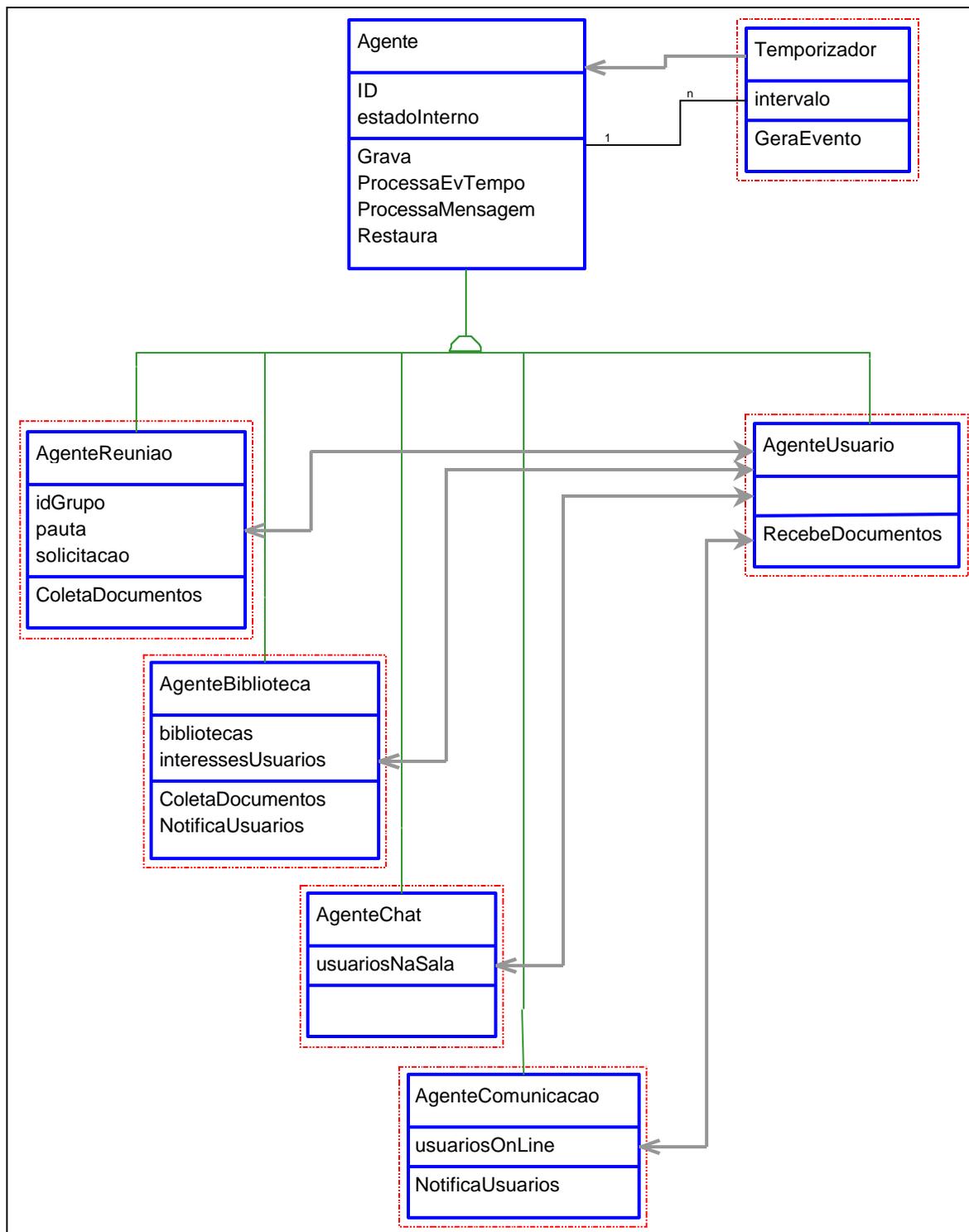


Figura 4-4 – Modelo para os agentes do sistema.

Os serviços essenciais também são mostrados na classe **Agente** e dizem respeito à persistência (*Grava* e *Restaura*) e ao processamento de eventos gerados por mensagens ou pelo temporizador (*ProcessaMensagem* e *ProcessaEvTempo*). Um agente pode ter um ou mais temporizadores associados. Cada objeto da classe **Temporizador** notifica o

agente ao qual está associado a cada intervalo de tempo, que é configurado pelo próprio agente.

Os agentes constituem entidades com vida própria dentro do modelo. Para representar esta característica na implementação, cada agente pode ser representado por uma *thread* dedicada. Sucintamente, uma *thread* representa uma porção de uma aplicação principal que executa de maneira independente.

O diagrama mostra as cinco classes de agentes que compõem a arquitetura do sistema. Seus atributos e serviços foram ilustrados de maneira resumida e serão detalhados nas próximas seções. O **AgenteComunicação** tem como principal tarefa monitorar a conexão de usuários ao sistema, mantendo os mesmos informados sobre quem está *on-line* em cada grupo de trabalho. O **AgenteReunião** trata da realização de uma reunião a distância desde o seu agendamento até a geração da ata. Cada instância desta classe é criada para tratar de uma reunião em particular. O **AgenteBiblioteca** possui uma única instância e é encarregado de manter as bibliotecas de documentos, atendendo as solicitações dos usuários e garantindo que todos tenham a visão atualizada das bibliotecas durante todo o tempo de conexão. O **AgenteUsuário** é o único que não está associado ao servidor do sistema. Cada cliente possui uma instância desta classe, que é encarregado, por exemplo, de recuperar as mensagens para o usuário no tempo em que este ficou *off-line*.

#### 4.4.4 Servidor

Definidos os modelos para a construção dos agentes e para comunicação entre os componentes do sistema, inicia-se agora o projeto do servidor. Este constitui o núcleo da aplicação, sendo responsável pelo armazenamento de dados e contendo a maioria das entidades inteligentes do sistema.

##### 4.4.4.1 Arquitetura

O esquema da Figura 4-5 mostra a arquitetura do servidor do sistema, que é composto por um conjunto dinâmico de entidades ligadas a um elemento denominado **Roteador**. O número de agentes rodando no sistema varia basicamente de acordo com

as reuniões em andamento e o número de salas de *chat* ativas em cada instante, já que as demais classes de agentes possuem instâncias únicas.

Outra parte fundamental do servidor é a interface com o administrador do sistema, através da qual é realizado o cadastrado de usuários e a composição dos grupos de trabalho. A base de dados do sistema, que está representada no diagrama por uma única entidade (BD), na realidade está distribuída em diversas bases de informações. Algumas, como o cadastro de usuários, são utilizadas por praticamente todas as entidades do modelo. Outras são de acesso exclusivo de determinadas entidades.

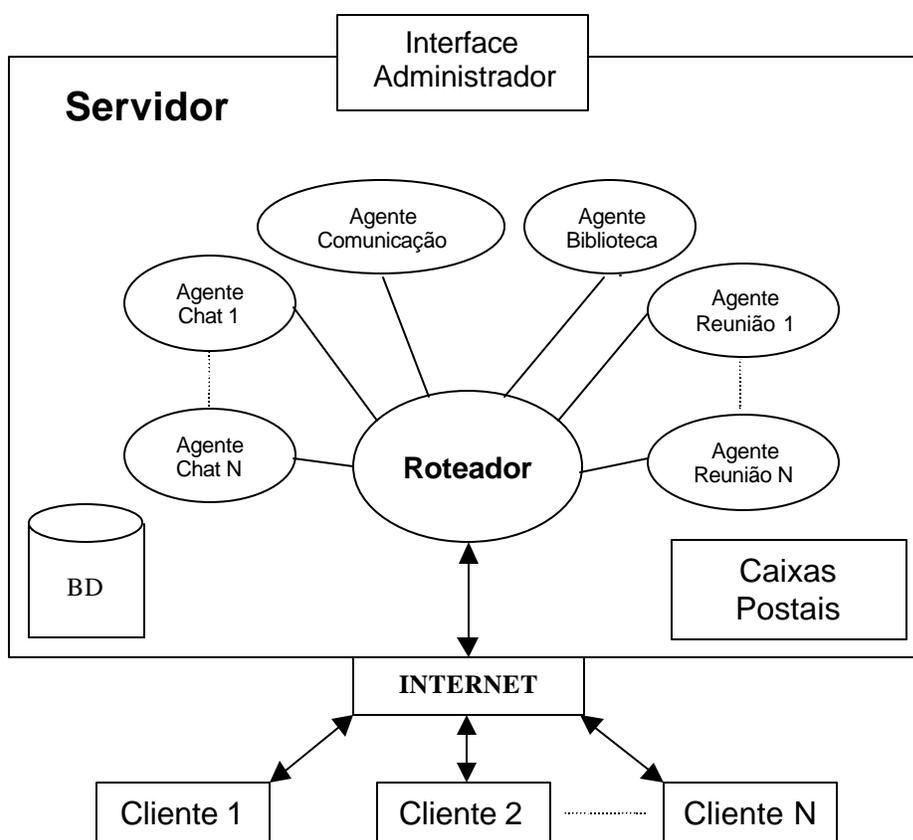


Figura 4-5 – Arquitetura do servidor.

O esquema mostra ainda a Internet representando a plataforma de comunicação entre as instâncias do módulo cliente e o servidor. Dois clientes nunca estabelecem comunicação de forma direta. Todas as mensagens são distribuídas pelo **Roteador**. A seguir serão fornecidos detalhes sobre os papéis e a estrutura de cada entidade da arquitetura.

#### 4.4.4.2 Interface com o Administrador

A interface com o administrador do sistema é composta basicamente pelo cadastro de usuários e grupos de trabalho. Este cadastro foi modelado como uma base de dados relacional. Isto constitui uma exceção em termos de modelagem, já que o restante do sistema utiliza o paradigma da orientação a objetos. A razão principal para esta abordagem é a necessidade de se armazenar os dados do cadastro através de um Sistema Gerenciador de Bancos de Dados (SGBD). Esta necessidade é devida ao fato deste ser o cadastro mais importante da aplicação, podendo atingir um volume considerável dependendo do tamanho da instituição. Os SGBDs oferecem, acima de tudo, uma boa garantia quanto à integridade dos dados, além de agilizarem bastante as operações de pesquisa e manutenção.

O cadastro é constituído por duas tabelas relacionadas. A tabela de usuários armazena entradas com os seguintes dados: nome, *login* no sistema, senha de acesso, endereço de correio eletrônico, endereço residencial e telefone. Para cada grupo é armazenado o nome, a descrição, os usuários que o compõem e o usuário nomeado como coordenador.

A linguagem Java oferece uma API específica para o acesso a bases de dados relacionais. O JDBC (*Java Data Base Connectivity*) possibilita a conexão de uma classe do programa a uma base de dados através de um *driver* instalado no sistema operacional. Diversos tipos de *drivers* podem ser acessados, o que leva a uma independência entre o programa Java e o sistema de gerenciamento de dados. Através do JDBC pode-se manipular a base de dados através de consultas SQL (*Structured Query Language*).

#### 4.4.4.3 Roteador

Esta entidade tem como função principal viabilizar a comunicação entre os componentes do sistema. Sempre que um cliente se conecta ao sistema, o faz solicitando uma conexão ao Roteador, que possui um endereço conhecido na Internet ou em outra rede TCP/IP (uma rede local, por exemplo) sobre a qual rodam as aplicações. Ao receber a solicitação, o Roteador cria uma porta de comunicação (*socket*) para o novo cliente, que será mantida enquanto este permanecer conectado. Como o Roteador

mantém as portas de comunicação, todas as mensagens passam obrigatoriamente por ele.

Quando o Roteador recebe uma mensagem, decide como deve proceder dependendo da classe a qual ela pertence e do seu destinatário. Por exemplo, uma mensagem que seja uma especialização da classe `MensagemAgente` é enviada diretamente ao agente identificado como destinatário. Uma mensagem que não se destina a uma entidade do sistema (um agente ou o próprio Roteador) deve ser destinada a outro usuário, identificado como destinatário. Neste caso, a mensagem é imediatamente repassada ao mesmo pelo Roteador.

Além dos clientes enviarem suas mensagens através do Roteador, os agentes que compõem o módulo servidor também solicitam ao mesmo o envio de mensagens aos clientes. O Roteador gerencia mensagens que podem ser destinadas diretamente de um usuário para outro, de um usuário para um agente e vice-versa, de um agente para outro e de um usuário ou agente para o próprio Roteador.

Além da distribuição de mensagens, o Roteador exerce outro papel importante: controlar a ativação e desativação dos agentes quando o servidor do sistema é iniciado ou finalizado. O Roteador é indicado para a realização desta tarefa por já possuir referências a todos os agentes do sistema, necessárias para a distribuição das mensagens. Quando o sistema está para ser finalizado, todos os agentes ativos são notificados pelo Roteador e armazenados em um arquivo, através do mecanismo de persistência de objetos descrito anteriormente. Quando o servidor é inicializado, o Roteador é o primeiro componente ativado, tendo como primeira função restaurar os agentes do arquivo.

O projeto do Roteador é um ponto crucial para o requisito de estendibilidade do sistema. Tanto o conjunto de agentes quanto as mensagens do sistema são totalmente abstratos. O Roteador trata todos os agentes de maneira uniforme independente das suas funções no sistema. Novas categorias de agentes podem ser facilmente adicionadas ao modelo através da especialização da classe **Agente**, sem que nenhuma alteração seja necessária na estrutura do Roteador. A hierarquia de mensagens também pode evoluir livremente abaixo da super classe **Mensagem**, quase sempre sem alterar o processo de roteamento. Tal processo só será alterado se uma nova classe de mensagens necessitar

de alguma diferenciação além daquela feita entre as mensagens destinadas aos agentes e aos usuários.

#### 4.4.4.4 Agente de Comunicação

A classe **AgenteComunicação**, que aparece na Figura 4-4, modela um agente único no sistema. Este é o agente mais simples da arquitetura, tendo como única função monitorar as conexões dos usuários com objetivo de manter os usuários que estão *on-line* constantemente informados sobre a situação dos grupos de trabalho.

Sempre que o Roteador recebe uma mensagem de *login* ou *logout*, de um usuário, esta mensagem é repassada também para o agente de comunicação. Este por sua vez, atualiza a situação de usuários *on-line* e *off-line* para os grupos do usuário que enviou a mensagem, informando todos os membros conectados nesses grupos sobre a nova situação.

O **estado interno** deste agente é representado por um mapa dos grupos de trabalho, contendo usuários *on-line* e *off-line*. Alterações neste estado só ocorrem com o recebimento de mensagens de *login* e *logout*. A persistência não é importante para este agente, já que seu estado interno só tem significado durante a mesma execução do programa.

#### 4.4.4.5 Agente de Chat

As salas de conversação do sistema são voltadas para os grupos de trabalho. Sempre que um usuário solicita a entrada na sala de conversação de um grupo, o Roteador verifica se a sala já existe. Se já existir, o usuário é imediatamente conectado ao agente que coordena a sala, senão, uma nova instância da classe **AgenteChat** é criada no sistema. A partir daí, todas as mensagens de *chat* do grupo em questão são encaminhadas para este agente.

O **estado interno** do agente representa a cada instante os usuários presentes na sala de conversação de um grupo de trabalho. Este agente é responsável por manter os usuários sempre informados sobre quem está presente na sala do grupo e repassar as mensagens que constituem a conversa de acordo com as solicitações dos participantes.

Quando o agente percebe que não há mais usuários na sala, trata de finalizar sua execução. Assim como o agente de comunicações, estes agentes não persistem de uma execução para outra no servidor, já que as salas de *chat* são voláteis, não havendo relação entre duas sessões de conversação de um mesmo grupo, embora este conceito possa ser facilmente alterado caso se considere importante manter uma memória sobre as conversações.

#### 4.4.4.6 Agente de Reunião

A organização e coordenação das reuniões a distância é o ponto do modelo onde a aplicação de agentes se torna mais interessante. Esta atividade exige do sistema um acompanhamento constante sobre cada reunião, desde a sua solicitação até o seu fechamento definitivo, possivelmente após a geração da ata. Durante todo o ciclo de vida de uma reunião, o sistema toma atitudes por iniciativa própria no intuito de atender às orientações do coordenador e garantir a sequência dos trabalhos do grupo.

O número de atividades relativas ao gerenciamento de uma reunião sugere a necessidade de um agente exclusivo para tratar de cada reunião. A classe **AgenteReunião** modela este conceito. Quando o coordenador de um grupo de trabalho solicita uma nova reunião, um novo agente é instanciado no sistema, tornando-se o responsável tanto pela coordenação das atividades da reunião, quanto pelo armazenamento dos dados relativos à mesma.

A Figura 4-6 detalha a parte do modelo envolvendo a classe **AgenteReunião**. Uma instância desta classe armazena basicamente a identificação do grupo de trabalho da referida reunião, a solicitação do coordenador do grupo e a pauta de assuntos, além das propriedades herdadas da super classe **Agente**.

##### A Classe SolicitacaoReunião

Ao ser inicializado, o agente recebe uma instância da classe **SolicitacaoReunião**, que contém as determinações do coordenador do grupo. Nesta classe são armazenados o assunto principal, a duração prevista e os horários sugeridos para a reunião. Além disso, o coordenador determina como o agente deve agir em relação ao agendamento da reunião. Como especificado na análise de requisitos, são três as possibilidades de ação do agente:

- ✓ Solicitar as opiniões dos membros do grupo sobre os horários e retornar as respostas ao coordenador para que este tome a decisão final.
- ✓ Solicitar as opiniões dos membros do grupo e decidir pela opinião da maioria.
- ✓ Marcar a reunião sem solicitar opiniões. Neste caso o horário é determinado diretamente pelo coordenador.

O capítulo sobre a implementação fornece maiores detalhes sobre a interação do agente de reunião com os usuários, porém as possibilidades de ação do agente são essenciais neste contexto para que o seu estado interno seja definido, lembrando que a análise estado - evento é a base para o funcionamento do agente.

### O Estado Interno

O estado interno do agente de reunião é representado por uma enumeração que define em que curso estão as atividades relativas à reunião. Com base nas possibilidades de ação do agente, tanto durante o agendamento da reunião quanto nas atividades subsequentes, foram definidos os estados apresentados na Tabela 4-1. O estado evolui em função das mensagens recebidas dos usuários e dos horários estipulados pelo próprio agente como limite para a execução de certas tarefas – da início à reunião, por exemplo. A evolução dos estados não ocorre num único sentido e não necessariamente passa por todos os estados definidos. Dependendo da solicitação do coordenador, certos estados podem ser omitidos e outros repetidos várias vezes, como será observado mais tarde.

| <b>Estado</b>          | <b>Significado</b>   |
|------------------------|--|
| Iniciou                | O agente acaba de ser criado e receber a solicitação da reunião.   |
| Opiniões solicitadas   | Se o coordenador assim determinou, foram solicitadas as opiniões dos membros do grupo sobre os horários. O agente aguarda pelas respostas.   |
| Coordenador consultado | Quando todos os membros do grupo responderam sobre os horários ou quando o horário limite foi alcançado, o agente consulta o coordenador sobre a marcação da reunião, se a solicitação for esta. |
| Reunião marcada        | A reunião foi marcada. A agenda do grupo deve ser atualizada.  |
| Início solicitado      | Quando o horário da reunião se aproxima, o agente pergunta ao coordenador se a sessão pode ser iniciada.   |
| Reunião iniciada       | O agente de início à reunião e passa a monitorar as atividades desta.  |
| Reunião finalizada     | Todas as atividades da reunião foram encerradas. O agente está pronto para acabar seu ciclo de vida.   |

**Tabela 4-1 – Estados do agente de reunião.**

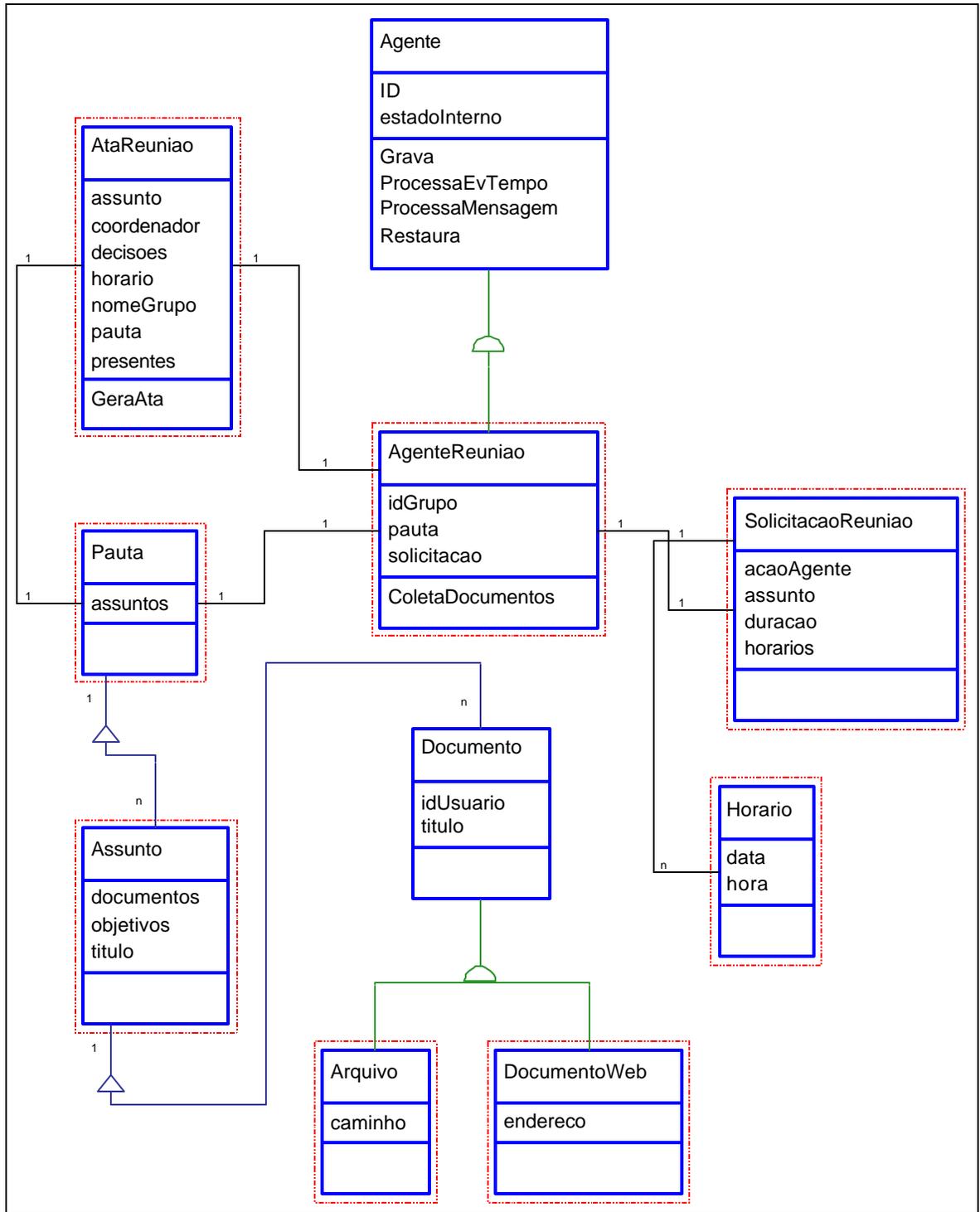


Figura 4-6 – Modelo para o agente de reunião.

A Pauta

Como foi estipulado na definição dos requisitos, uma reunião a distância é guiada por uma pauta previamente estabelecida pelos membros do grupo. A pauta é

armazenada pelo agente da reunião, que a fornece aos membros do grupo sempre que estes desejarem editá-la ou simplesmente analisá-la.

A classe **Pauta** é modelada como um conjunto dinâmico de assuntos, que são representados pela classe **Assunto**. Cada assunto possui um título, um texto explicando seus objetivos e um conjunto de documentos. A classe **Documento** é definida por um título e armazena também a identificação do usuário que anexou o documento ao assunto. Esta classe é especializada pela classe **Arquivo**, que representa documentos armazenados no sistema local do usuário, e pela classe **DocumentoWeb**, que representa o endereço de uma página na **Web**.

### Coleta de Documentos

Outra função importante do agente de reunião é a coleta dos documentos anexados aos assuntos da pauta. Sempre que o agente está desocupado ele verifica a pauta em busca de documentos pendentes. Quando o usuário que possui um documento pendente está conectado, o agente inicia o processo de transferência do documento do sistema local do usuário para o servidor. O objetivo do agente é disponibilizar todos os documentos no servidor antes que a reunião tenha início, o que facilita a visualização dos mesmos pelos membros do grupo durante a sessão.

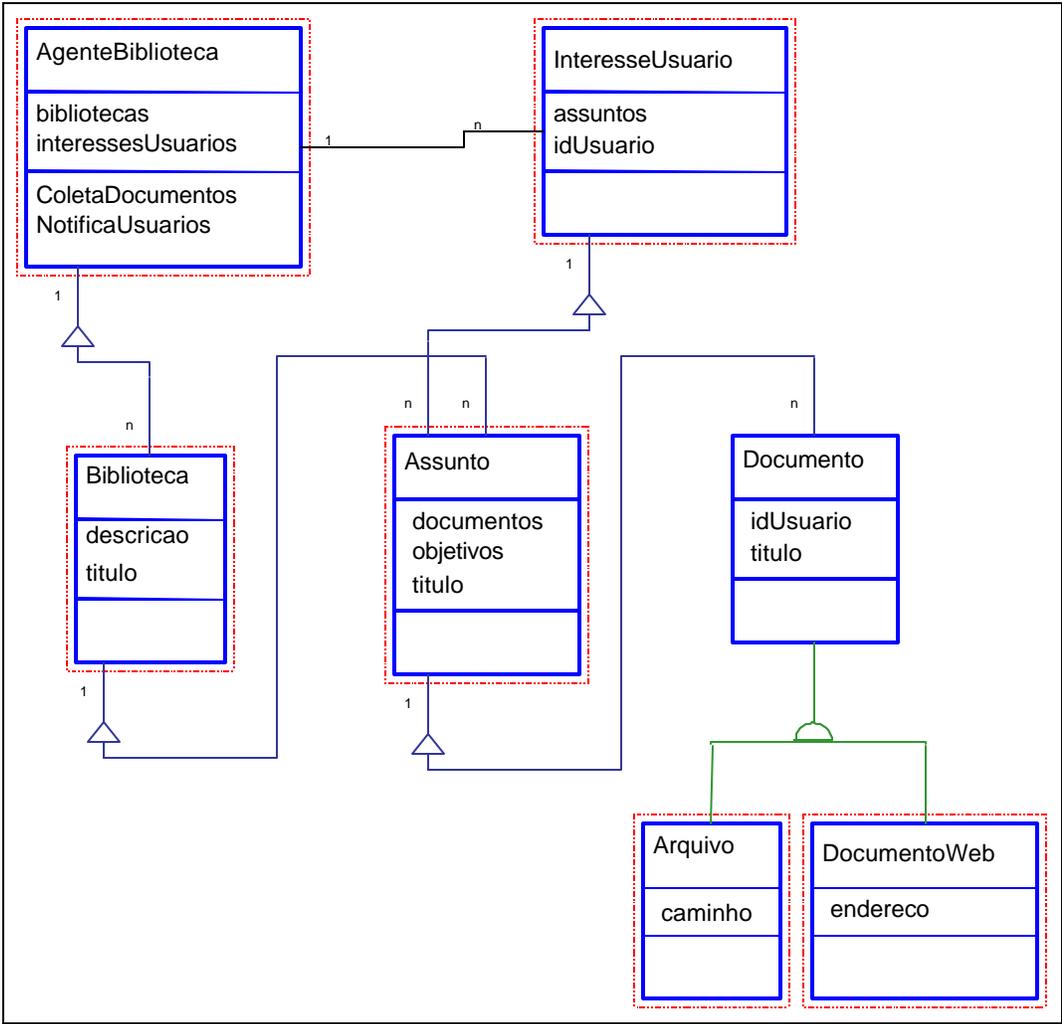
### Geração da Ata

A última tarefa do agente de reunião antes de ser finalizado é a geração da ata. O agente compõem a ata através das informações constantes na pauta da reunião e das conclusões tomadas durante as discussões dos membros do grupo. Como será mostrado no capítulo seguinte, a interface da sala de reunião deve fornecer meios para que o coordenador do grupo tome nota das decisões ou outros detalhes de importância que devam constar na ata. As anotações do coordenador são enviadas ao agente da reunião, que trata de armazená-las para posterior geração da ata.

Quando o agente é solicitado pelo coordenador a gerar a ata da reunião, reúne todas as informações relevantes – horário, usuários presentes, assuntos em pauta, decisões tomadas, etc. – em um novo objeto da classe **AtaReunião**. Este objeto é então armazenado no arquivo de atas do sistema.

Geralmente o agente da reunião é finalizado logo após a geração da ata. As atas ficam armazenadas em arquivos no servidor. Quando um usuário solicita a visualização das atas de um grupo, o Roteador abre o arquivo correspondente, recupera as atas e as envia através de uma mensagem específica. Objetos da classe **AtaReunião** têm a capacidade de gerar um texto com a ata da reunião a partir dos dados armazenados em sua estrutura.

**4.4.4.7 Agente de Biblioteca**



**Figura 4-7 – Modelo para o agente de biblioteca.**

A Figura 4-7 mostra as classes envolvidas no projeto do agente de biblioteca. A classe **AgenteBiblioteca** é uma especialização da classe **Agente** e é responsável pelo armazenamento e manutenção das bibliotecas de documentos. Este agente é uma instância única no sistema e é composto por um conjunto dinâmico de bibliotecas. A

classe **Biblioteca** possui uma identificação (título e descrição) e representa um conjunto de assuntos. Os assuntos são representados exatamente pelas mesmas classes já utilizadas na definição da pauta de reunião.

Além de reutilizar o modelo de assuntos da pauta de reunião, este agente também aproveita a mesma idéia de coleta de documentos definida para aquele agente. Quando o agente está ocioso, deve verificar por documentos pendentes e tentar obtê-los a partir do sistema local dos usuários, para que fiquem armazenados no servidor.

Este agente foi modelado para armazenar também os assuntos de interesse dos usuários. Cada usuário do sistema pode se cadastrar para receber notificações do agente sobre atualizações nos assuntos de seu interesse. Sempre que um assunto é alterado, o agente envia notificações segundo as informações armazenadas pelas instâncias da classe **InteresseUsuario**.

O agente deve também manter sempre atualizada a visão dos usuários sobre as bibliotecas. Sempre que ocorre uma alteração, a nova estrutura das bibliotecas deve ser enviada para todos os usuários conectados.

#### **4.4.4.8 Caixas Postais**

Para realizarem suas tarefas, os agentes do sistema precisam enviar notificações ou solicitações aos usuários em diversas situações, independente destes estarem conectados ou não. Um agente de reunião, por exemplo, precisa interagir com todos os membros do grupo de trabalho algumas vezes até que a reunião efetivamente ocorra. Para facilitar o processo de comunicação dos agentes com os usuários, foi criado um esquema de caixas postais como parte da estrutura do Roteador. As caixas postais visam o armazenamento temporário das mensagens destinadas a usuários que não estejam conectados quando as mensagens são enviadas.

Este recurso tira dos agentes a responsabilidade de verificar se o usuário que precisa ser contado está ou não conectado. O Roteador trata de realizar esta verificação sempre que é solicitado a transmitir uma mensagem de um agente para um usuário. Se o usuário está fora do sistema, a mensagem é armazenada na sua caixa postal. Por outro lado, dependendo da atividade do agente, outra preocupação passa a existir: certas mensagens podem se tornar obsoletas antes que cheguem aos seus destinatários. O

Roteador fornece aos agentes o recurso de filtrar e eliminar mensagens das caixas postais antes que estas sejam efetivamente enviadas. Cabe aos agentes tomar esta iniciativa quando perceberem que certas mensagens não têm mais sentido.

Dependendo da implementação, a caixa postal de cada usuário pode ser criada e destruída dinamicamente pelo Roteador, conforme existam ou não mensagens pendentes na mesma.

#### 4.4.5 Cliente

A Figura 4-8 esquematiza a arquitetura do módulo cliente, através do qual os usuários acessam o sistema. Em contraste com o módulo servidor, o modelo do cliente é bastante simples, contendo apenas um agente, além da *thread* que fornece os serviços de comunicação.

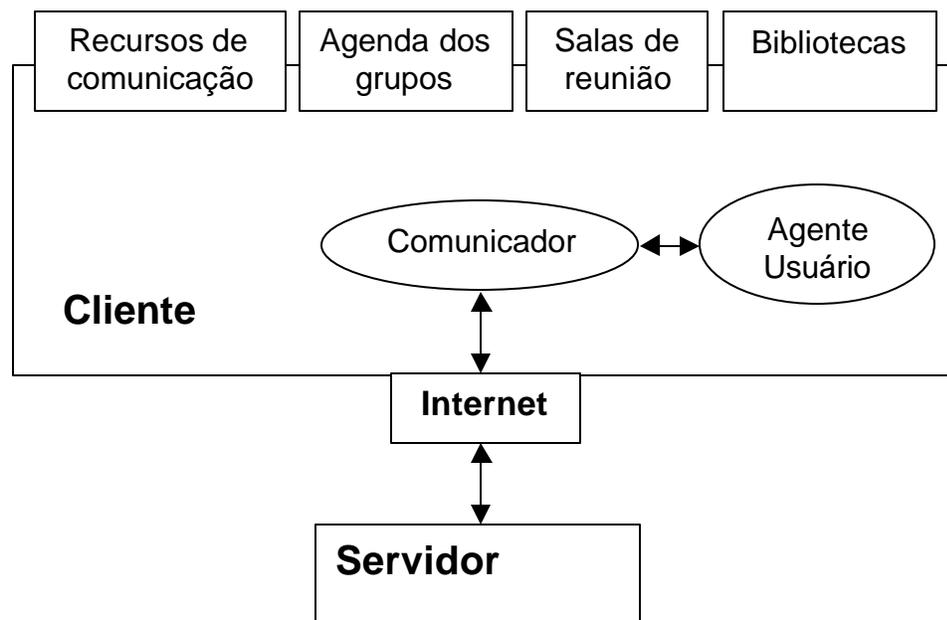


Figura 4-8 – Arquitetura da aplicação cliente.

O ponto principal desta parte do modelo é a interface. O diagrama mostra a interface dividida em quatro grupos funcionais:

- ✓ Recursos de comunicação: este grupo engloba as salas de conversação, o envio de mensagens *on-line* e a visualização constante da situação dos grupos de trabalho.
- ✓ Agenda dos grupos: é a parte da interface que trata principalmente do agendamento

de reuniões. Certas funções fornecidas por esta interface são acessíveis somente pelos coordenadores dos grupos de trabalho.

- ✓ Salas de reunião: as reuniões dos grupos de trabalho são realizadas através desta interface, que na primeira versão será composta por janelas baseadas em texto.
- ✓ Bibliotecas: esta interface permite que o usuário acesse as bibliotecas do servidor, podendo consultar e alterar seu conteúdo.

#### **4.4.5.1 Comunicador**

O Comunicador é a parte da aplicação responsável pela ligação do cliente com o servidor. Quando o usuário acessa o sistema, o Comunicador é disparado como uma *thread*, estabelecendo uma conexão com o Roteador, localizado no servidor.

Uma vez inicializado, o Comunicador passa a monitorar uma porta de comunicação para receber as mensagens destinadas ao usuário. Todas as mensagens recebidas são repassadas para o Agente do Usuário, que toma a atitude adequada para cada mensagem. O Comunicador também fornece o serviço de envio de mensagens, utilizado pelo Agente do Usuário ou diretamente pelos elementos da interface.

#### **4.4.5.2 Agente do Usuário**

A primeira versão do modelo utiliza somente um tipo de agente no cliente da aplicação. A função primordial do Agente do Usuário é notificar o usuário sobre as mensagens recebidas. Logo após inicializado, o agente se comunica com o servidor do sistema para verificar se existem mensagens pendentes para o usuário na caixa postal. O agente fica então disponível para receber as mensagens através do Comunicador e para atender às solicitações do usuário.

As requisições do usuário frequentemente fazem com que seu agente comunique-se com os agentes localizados no servidor. Por exemplo, quando um usuário solicita a visualização da pauta de uma reunião, o Agente Usuário envia uma mensagem com a solicitação para o agente responsável pela reunião. Quando este responde, possivelmente enviando a ata, o agente trata de fornecê-la ao usuário.

Este agente encapsula todo o conhecimento referente aos procedimentos de comunicação necessários para disponibilizar os serviços do sistema ao usuário. O acréscimo de novos serviços ao modelo certamente conduziria à criação de agentes mais específicos, visando a decomposição da arquitetura para obtenção de componentes mais simples. Por hora, a arquitetura baseada em um agente único atende bem os objetivos do modelo.

## 4.5 Considerações Finais

As fases de análise e projeto de um sistema computacional são cruciais para o seu desenvolvimento. As chances de sucesso na fase de implementação são bem maiores quando esta é baseada em um projeto consistente, que por sua vez parte de um conjunto bem estabelecido de requisitos.

Este capítulo buscou fornecer este suporte para a fase de implementação. Dentro de uma área de pesquisa cuja abrangência não é bem determinada, foi levantado um conjunto de requisitos que cobrem aspectos importantes do trabalho colaborativo, envolvendo a comunicação entre os usuários, as reuniões a distância e o compartilhamento de informações.

O projeto apresentou uma arquitetura distribuída, dividida em dois módulos: servidor e cliente. O servidor é o elemento centralizador e sua estrutura envolve a maioria das entidades inteligentes, além de todo o armazenamento de dados da aplicação. Várias instâncias do módulo cliente são ligadas a um servidor único, e fornecem aos usuários o acesso às funcionalidades do sistema.

A utilização de agentes é marcante em toda a arquitetura. Entre outros benefícios, os agentes possibilitam uma melhor administração da complexidade do sistema. Os conhecimentos necessários para a realização das tarefas não ficam centralizados apenas em uma aplicação. Cada agente constitui uma entidade autônoma que encapsula a inteligência relativa a uma parte dos requisitos do sistema. Desta forma, diversas tarefas podem ser tratadas paralelamente. A descentralização das atividades também contribui para a obtenção de projetos mais simples, mesmo quando o conjunto de requisitos cresce ou se torna mais complexo.

Quanto à adequação dos agentes modelados às definições apresentadas no capítulo 3, pode-se dizer que estes possuem as propriedades consideradas essenciais pela maioria dos autores pesquisados. O modelo apresenta agentes que executam de forma autônoma, seja através do conceito de *thread* ou de outros recursos de implementação aplicáveis. Os agentes executam tarefas delegadas pelos usuários sem a intervenção direta dos mesmos, como no agendamento de reuniões, por exemplo. Para realizar as tarefas, eles monitoram aspectos de seu interesse no ambiente e atuam sobre o mesmo quando for adequado.

Embora ainda não utilizem conceitos mais avançados de IA, como aprendizado e reconhecimento de padrões, os agentes do modelo possuem certo grau de inteligência, modelada através da análise estado – evento. Quando estimulado por eventos externos ou por eventos temporizados, cada agente sabe que atitudes devem ser tomadas em função do seu estado interno. Neste aspecto, o Agente de Reunião possui a evolução de estados mais interessante, agindo de diversas maneiras dependendo do tipo de solicitação de reunião que recebe.

A preocupação com a estendibilidade e descentralização do modelo conduziram à obtenção de estruturas que poderão ser utilizadas em outras aplicações distribuídas, independente dos seus campos de atuação. Neste sentido, destacam-se principalmente as partes do modelo que especificam a arquitetura dos agentes, o protocolo de comunicação e o roteamento de mensagens. Essas estruturas são facilmente extensíveis dentro do próprio sistema e também podem ser especializadas para a composição de novas aplicações.

## 5 Implementação do Sistema

### 5.1 Considerações Iniciais

O objetivo deste capítulo é apresentar o produto da implementação do modelo elaborado no capítulo anterior. A implementação foi realizada em linguagem Java, que tem como característica marcante a independência entre a aplicação e o sistema operacional. Além da utilização do protocolo TCP/IP, que possibilita a utilização do sistema tanto sobre a Internet como sobre redes locais, a independência de plataforma é outra característica importante, contribuindo para que um maior número de usuários possa ser atingido. Uma implementação em Java pode ser distribuída por diversas plataformas sem a necessidade de versões específicas.

A interação dos usuários com o sistema é descrita à medida que são mostrados os recursos disponíveis na interface das aplicações cliente e servidor. No decorrer da apresentação são esclarecidas questões importantes sobre a implementação e sobre o comportamento dos agentes e demais entidades do modelo.

Durante a implementação, um dos grandes objetivos foi mapear com a maior fidelidade possível as estruturas levantadas na fase de projeto. O paradigma da programação orientada a objetos encoraja esta abordagem, mantendo uma notação uniforme durante todo o ciclo de desenvolvimento.

Além das estruturas previstas no modelo, é frequente a identificação de novas classes e objetos na fase de implementação. Estas novas estruturas normalmente não fazem parte no domínio do problema, o que revelaria deficiências no projeto. Seu papel é frequentemente dar suporte à implementação dos serviços da aplicação. Neste trabalho, as estruturas acrescentadas ao modelo durante a implementação visam basicamente a composição da interface do sistema.

No modelo apresentado, é notável a ausência de um detalhamento sobre as estruturas que compõem a interface, que foi representada apenas por agrupamentos de

funções nos diagramas com a arquitetura dos módulos. A principal razão para esta omissão é a vantagem obtida quando se separa o modelo da interface e o modelo que representa o domínio do problema. Quando a interface não está fortemente ligada à estrutura do sistema, sua flexibilidade é muito maior. Isto facilita, entre outras alterações, os processos de modernização aos quais os *softwares* são frequentemente submetidos. Além disso, nos ambientes de desenvolvimento mais modernos, a interface é facilmente constituída através de componentes que representam janelas, menus, planilhas e outros elementos. Estes componentes são totalmente personalizáveis e diminuem muito a necessidade de escrita de classes para finalidades específicas. A facilidade gerada pela utilização dos componentes incentiva a atividade de **prototipação**, onde vários modelos para a interface do sistema são apresentados até que se obtenha o resultado esperado.

## 5.2 Ferramentas de Desenvolvimento

### 5.2.1 Escolha da Plataforma de Desenvolvimento

Um grande número de linguagens de programação de propósito geral estão à disposição do desenvolvedor que está para iniciar um projeto. A opção por determinada linguagem depende de inúmeros fatores, dentre os quais podem estar a metodologia de desenvolvimento empregada, a facilidade na construção dos componentes da interface, os ambientes disponíveis no mercado e até mesmo o gosto pessoal dos envolvidos no projeto. No entanto, as características particulares de cada projeto tendem a exercer muita influência nesta decisão. Aplicações que envolvem a manipulação de bases de dados, por exemplo, exigem uma plataforma de desenvolvimento que forneça acesso a sistemas de gerenciamento de bases de dados.

A linguagem Java foi escolhida para a implementação realizada neste trabalho. Os principais fatores que levaram a esta escolha foram os seguintes:

- ✓ Suporte a aplicações *multithread*: o sistema desenvolvido é composto por duas aplicações principais (cliente e servidor), que são constituídas por diversas entidades que executam paralelamente, como é o caso dos agentes. A plataforma Java suporta naturalmente este tipo de aplicação.

- ✓ Acesso à rede: as facilidades fornecidas na API da linguagem Java para aplicações que necessitam de acesso a algum protocolo de rede simplificam bastante o projeto.
- ✓ Independência de plataforma: um programa totalmente escrito em Java pode ser utilizado em qualquer ambiente (*hardware* e sistema operacional) que disponha de uma Máquina Virtual Java.

Além disso, é muito importante o fato da linguagem ser totalmente orientada a objetos e possuir recuperação automática da memória alocada, sem nenhuma interferência do programador. Esta última característica contribui muito para a obtenção de um código mais robusto e confiável.

### 5.2.1.1 Ambiente de Desenvolvimento

Associada à escolha da linguagem está a opção por um ambiente integrado de desenvolvimento (IDE). De maneira geral, um IDE é composto por um compilador, um conjunto de bibliotecas para diversas finalidades e o ambiente de desenvolvimento propriamente dito, com recursos específicos para a construção da interface, edição dos arquivos, gerenciamento de projetos, depuração e outros.

No caso da linguagem Java, o compilador utilizado pela ferramenta de desenvolvimento pode ser proprietário, assim como a Máquina Virtual e as APIs. No entanto, algumas plataformas utilizam o JDK – *Java Development Kit* - da Sun Microsystems. O JDK é por si só uma ferramenta de desenvolvimento completa, porém sem oferecer nenhum recurso de edição. Além do compilador, o *kit* contém uma API padrão e o JRE – *Java Run Time Environment* – uma máquina virtual que pode também ser instalada separadamente onde se desejar executar uma aplicação Java.

Três ambientes foram avaliados na fase inicial do desenvolvimento: o *JBuilder*, da Borland; o *VisualAge for Java*, da IBM; e o *Visual J++*, da Microsoft. Todos incorporam filosofias de programação visual e baseada em componentes. O *JBuilder* da Borland foi a ferramenta escolhida. A opção se deve principalmente à diversidade de componentes oferecidos e a um critério puramente pessoal: a familiaridade com os ambientes do mesmo fabricante para a linguagem C++.

## 5.2.2 A Linguagem Java

### 5.2.2.1 Surgimento da Linguagem

A linguagem de programação Java foi projetada para atender as necessidades do desenvolvimento de aplicações no contexto de ambientes heterogêneos e distribuídos. Teve origem como parte de um projeto de pesquisa na Sun Microsystems, que desenvolvia *softwares* avançados para uma grande variedade de dispositivos de rede. A meta era desenvolver uma plataforma operacional pequena, confiável, portátil e distribuída. No início do projeto a linguagem C++ foi escolhida. Com o passar do tempo as dificuldades encontradas com C++ chegaram a um limite. Concluiu-se que os problemas seriam melhores resolvidos através de uma plataforma totalmente nova. As diretivas para o projeto da nova plataforma foram fruto do exame de diversas linguagens, como Eiffel, SmallTalk, Objective C e Cedar/Mesa. O resultado é uma linguagem que tem provado ser ideal para o desenvolvimento de aplicações seguras e distribuídas por ambientes que vão desde uma rede local até a Internet. [GOS 96]

### 5.2.2.2 Linguagem e Plataforma

Java não é apenas uma linguagem de programação, mas também uma plataforma operacional.

O mundo da informática conta atualmente com diversas plataformas, entre as quais estão o Microsoft Windows, Macintosh, OS/2, UNIX e NetWare. Compilações distintas de um mesmo *software* devem ser geradas para rodar em cada plataforma, já que o código gerado pelos compiladores está fortemente ligado à arquitetura da mesma. [KRA 96] Além disso, os programas normalmente fazem uso intensivo da API fornecida pelo sistema operacional, criando um grau de dependência ainda maior.

Um dos grandes objetivos da plataforma Java é tornar os programas independentes de máquina ou sistema operacional. Uma API e uma Máquina Virtual constituem a plataforma operacional Java. A API é composta por uma completa hierarquia de classes, que representa deste os tipos mais simples, como números e caracteres, até mapeamentos sobre protocolos de rede. A Máquina Virtual funciona

como um interpretador, que traduz o código gerado pelo compilador para o código de máquina específico em tempo de execução.

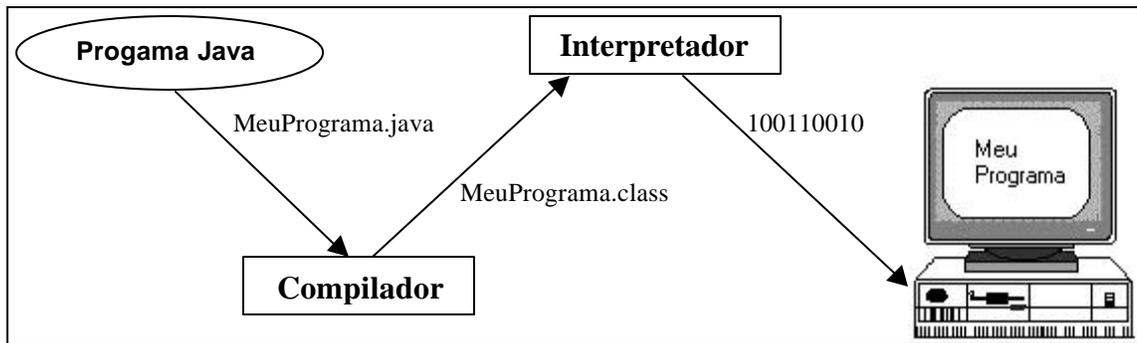


Figura 5-1 – Compilação e execução de programas em Java.

Fonte: The Java Tutorial [SUNa].

A Figura 5-1 ilustra este mecanismo. Cada classe que compõem um programa Java, assim como a função que constitui o ponto de entrada da aplicação, são escritos em arquivos com a extensão ‘java’, que são submetidos a um compilador. A saída do compilador é um código chamado *bytecode*, gravado em um arquivo com a extensão ‘class’. A Máquina Virtual gera o código binário específico para o *hardware* e sistema operacional a partir do *bytecode* quando o programa é executado.

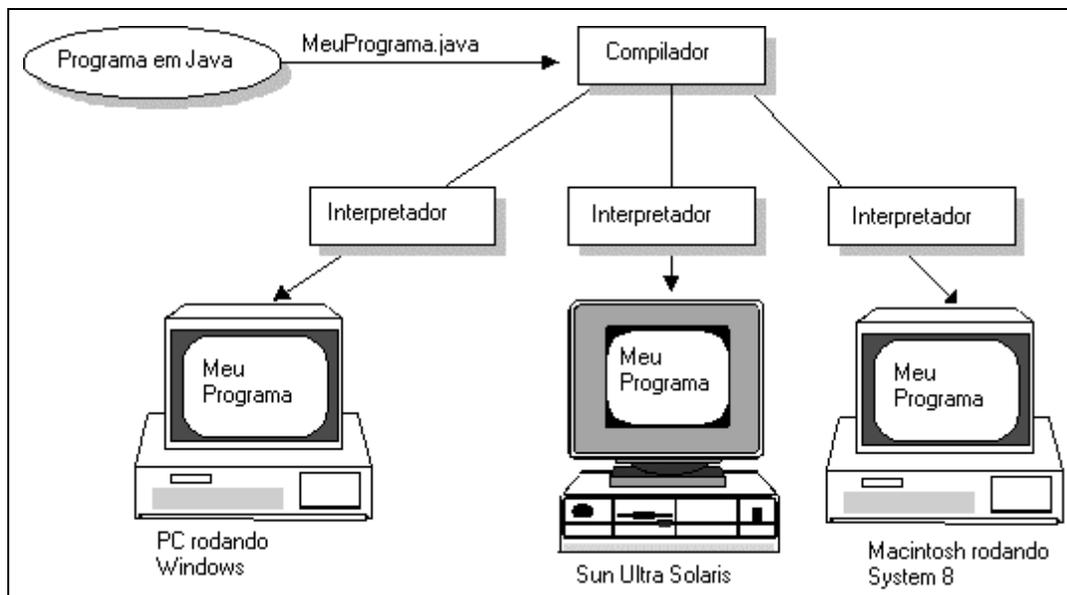


Figura 5-2 – Uma compilação, diversas plataformas.

Fonte: The Java Tutorial [SUNa].

A plataforma Java forma uma camada que isola o programa de aplicação do ambiente operacional. Os programas são construídos sobre a API e as particularidades de cada sistema são encapsuladas pela Máquina Virtual. A Figura 5-2 traz a idéia de uma única compilação do programa rodando em diferentes plataformas operacionais.

### 5.2.2.3 Principais Características

No artigo “*The Java Language: An Overview*” [SUNb] os idealizadores da linguagem descrevem suas características mais marcantes:

#### Simplicidade

Java foi projetada para ser uma linguagem simples, que não necessitasse de grandes períodos de treinamento e que tirasse proveito dos padrões atualmente utilizados. A maioria dos programadores utiliza atualmente C++ como linguagem de desenvolvimento orientado a objetos. Por este motivo, a sintaxe da linguagem Java foi construída para ser tão próxima quanto possível da sintaxe C++. No entanto, algumas características marcantes do C++ foram omitidas, como a sobrecarga de operadores e a herança múltipla.

Um coletor automático da memória alocada é um dos recursos que tornam o programa em Java muito mais simples. O gerenciamento de memória é uma das principais fontes de problema da programação em C ou C++.

#### Orientação a Objetos

A linguagem Java incorpora praticamente todas as facilidades da orientação a objetos disponíveis em C++, com exceção da herança múltipla. A ausência deste recurso é em grande parte compensada pelas **interfaces**. Uma interface Java define um conjunto de funcionalidades, que podem ser implementadas por qualquer classe. Uma classe pode implementar diversas interfaces, podendo assim assumir personalidades além daquela herdada da super classe.

Todos os tipos de dados, inclusive os tipos básicos, são representados por classes de objetos. Todas as classes derivam da super classe **Object**.

## Programação em Rede

A API Java fornece uma extensa biblioteca de classes cuja função é facilitar o acesso aos protocolos da família TCP/IP, como HTTP e FTP. Aplicações Java podem abrir e acessar objetos através da rede via URLs com a mesma facilidade com que se acessa um arquivo do sistema local.

A implementação realizada neste trabalho utiliza *sockets* TCP para prover a comunicação entre as entidades dos módulos cliente e servidor. O *socket* é uma porta de comunicação que, em Java, é encapsulada por *buffers* ou *streams* de entrada e saída, que fornecem os serviços essenciais de comunicação.

## Robustez

Java foi projetada para proporcionar a geração de programas confiáveis. Diversas verificações em tempo de compilação e de execução são realizadas para prevenir erros. A linguagem é fortemente tipada (como C++), não permitindo declarações implícitas como na programação em C. Isto possibilita que vários problemas em potencial sejam detectados no processo de compilação.

A maior vantagem sobre C++ neste aspecto está na coleta automática da memória alocada. Em Java não existem ponteiros e o desenvolvedor tem o mínimo de preocupação com o gerenciamento e a integridade da memória alocada.

## Segurança

Java foi idealizada para o uso em ambientes distribuídos. A ênfase dada no aspecto de segurança foi bastante grande. Técnicas de autenticação possibilitam a criação de programas seguros, inclusive contra vírus. O aspecto “robustez” está fortemente ligado com a segurança. O fato do programador não ter acesso a ponteiros impede que os dados protegidos das classes sejam atingidos, fechando as portas para muitos dos mecanismos empregados na construção dos vírus.

## Portabilidade

Como já foi mostrado, a plataforma Java isola a aplicação do ambiente operacional. Este é um grande passo para a portabilidade. Além disso, não existem

aspectos dependentes de implementação na especificação da linguagem. Todas as Máquinas Virtuais seguem normas rígidas visando garantir que o mesmo código execute da mesma maneira em todas as plataformas. Um exemplo dessas normas é a definição dos dados e operações aritméticas.

As bibliotecas que compõem a API definem interfaces portáteis. Para representar uma janela, por exemplo, existe uma classe abstrata chamada “*window*” e uma implementação específica da mesma para Windows, Unix, Macintosh, etc.

### Multithread

Aplicações *multithread* suportam a execução simultânea de diversos processos de forma sincronizada. Escrever programas com este tipo de arquitetura é bastante complicado em linguagens que não suportam este mecanismo.

Java integra um conjunto sofisticado de primitivas para a sincronização de processos, que libera o desenvolvedor das preocupações relativas à operacionalidade da arquitetura *multithread*. A facilidade de uso e a robustez destes recursos incentivam o uso deste modelo de aplicação, que possui um comportamento excelente em sistemas de tempo real.

## **5.3 O Servidor do VMeeting**

A partir de agora será apresentado o resultado da implementação do modelo elaborado no capítulo 4, iniciando pelo servidor. A Figura 5-3 mostra a janela principal do servidor, que representa a interface do sistema com o Administrador. Esta interface fornece acesso às funções de cadastro de usuários e de grupos de trabalho. O Administrador pode visualizar a lista com os nomes dos usuários ou com os grupos de trabalho. Em cada opção estão disponíveis as operações de inclusão, exclusão e edição de propriedades.

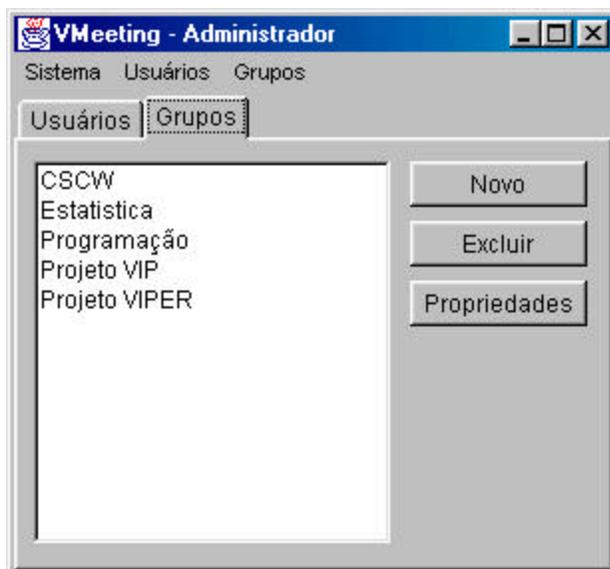


Figura 5-3 - Janela principal do servidor do sistema.

### 5.3.1 Cadastro de Usuários

O diálogo da Figura 5-4 é utilizado pelo Administrador para incluir um novo usuário ou para alterar os dados de um usuário já cadastrado. Além das informações pessoais são solicitados os dados para acesso do usuário ao sistema.

The image shows a dialog box titled "Propriedades do Usuário". It is divided into two sections. The first section, "Dados pessoais", contains four text input fields: "Nome" with the value "Adriano Coser", "Endereço" with "R. Lauro Linhares - Florianópolis - SC", "E-Mail" with "coser@eps.ufsc.br", and "Telefone" with "331-9498". The second section, "Acesso ao sistema", contains three text input fields: "Login" with "Adriano", "Senha" with "\*\*\*\*\*", and "Senha (conf.)" with "\*\*\*\*\*". At the bottom of the dialog are "Ok" and "Cancelar" buttons.

Figura 5-4 – Diálogo utilizado no cadastro de usuários.

### 5.3.2 Composição dos Grupos de Trabalho

A partir dos usuários cadastrados são compostos os grupos de trabalho. Um usuário que não faz parte de nenhum grupo de trabalho fica praticamente sem possibilidades ao acessar o sistema, podendo apenas utilizar os serviços da biblioteca de documentos.

Através do diálogo da Figura 5-5 o Administrador compõem um novo grupo de trabalho ou altera as propriedades de um grupo existente. Cada grupo é identificado pelo seu nome e descrição. No agrupamento chamado **Usuários**, todos os usuários cadastrados no sistema são listados à esquerda. À direita estão os usuários que compõem o grupo e o nome do coordenador escolhido. Entre as duas listas estão os botões que permitem a inclusão e exclusão de usuários do grupo.



Figura 5-5 – Diálogo para composição dos grupos de trabalho.

## 5.4 O Cliente do VMeeting

O módulo denominado cliente desde a fase de modelagem, é representado por outro aplicativo Java, que só é ligado ao servidor através da rede. Logo após a instalação, o endereço do servidor na rede precisa ser configurado. Nesta versão, este dado encontra-se em um arquivo texto instalado junto com o cliente.

O usuário usufrui melhor dos recursos do sistema se o mantiver ativo sempre que sua estação de trabalho estiver ligada. Isto propicia o acompanhamento contínuo dos eventos relativos às atividades dos grupos de trabalho aos quais ele pertence. No entanto, se o sistema permanecer fechado por determinado período de tempo, nenhuma mensagem destinada ao usuário será perdida. Todas as mensagens geradas enquanto o usuário esteve *off-line* são recuperadas pelo Agente do Usuário quando o cliente for iniciado. As únicas mensagens perdidas são as que foram geradas e perderam o sentido dentro do intervalo de tempo em que o usuário esteve *off-line*.

### 5.4.1 O Login do Usuário



Figura 5-6 – Janela de login do usuário.

Quando o usuário inicia a aplicação cliente, as informações de *login* são solicitadas através do diálogo da Figura 5-6. Quando o usuário pressiona o botão **Ok**, as informações são enviadas ao Roteador do sistema e o cliente entra no estado de espera. Instantes depois, o Comunicador recebe a resposta do servidor. Se o *login* foi bem sucedido, a aplicação é inicializada, senão, as informações são novamente solicitadas.

Quando o sistema for utilizado por um único usuário, este pode gravar as informações de *login* na configuração do sistema (Figura 5-7). Desta forma, quando o sistema é iniciado, as informações lidas na configuração são diretamente enviadas ao

servidor. Existe também uma configuração que permite ao usuário não receber mensagens mesmo estando conectado.



Figura 5-7 – As configurações do usuário.

#### 5.4.2 Recursos de Comunicação



Figura 5-8 – A janela principal da aplicação cliente.

Realizado o *login*, o usuário tem acesso à janela principal da aplicação, mostrada na Figura 5-8. Na barra de título está a identificação do usuário que está conectado. Três opções de visualização são representadas pelas pastas abaixo do menu de comandos: **Grupos**, **Agenda** e **Bibliotecas**.

A opção **Grupos**, possibilita que o usuário selecione, a qualquer momento, o grupo de trabalho com o qual deseja interagir. Também nesta opção, o usuário fica

continuamente informado sobre a situação dos usuários no grupo de trabalho selecionado. Sempre que alguém entra ou sai do sistema, as listas de usuários *on-line* e *off-line* são imediatamente atualizadas. O **Agente de Comunicação**, localizado no servidor, encarrega-se de enviar uma notificação aos usuários dos grupos que tiveram a situação alterada.

#### 5.4.2.1 Comunicação Usuário a Usuário

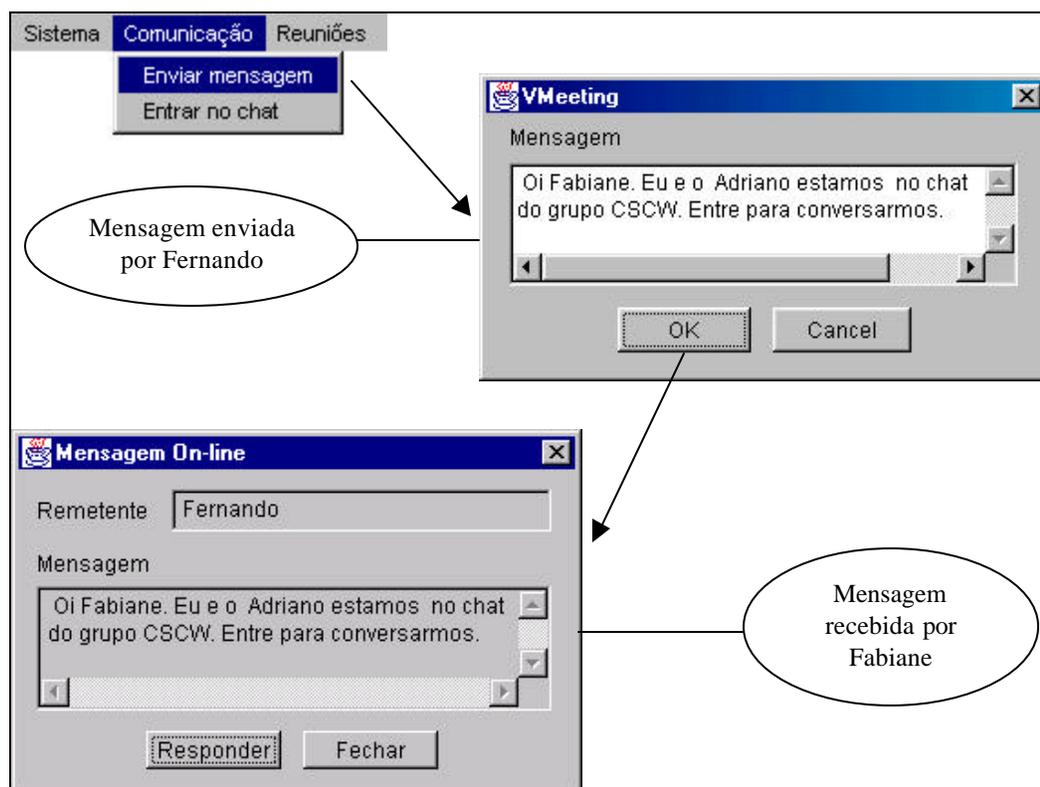


Figura 5-9 – A comunicação usuário a usuário.

Este recurso de comunicação permite que se estabeleça um diálogo rápido entre dois usuários conectados. O usuário que inicia a comunicação deve selecionar um membro do grupo de trabalho na lista de usuários *on-line* e acessar o comando **Enviar mensagem**, no menu **Comunicação**. Como mostra a Figura 5-9, o remetente deve digitar sua mensagem na janela que surge logo após o comando. Do outro lado, o destinatário recebe a mensagem e tem a opção de finalizar a conversa ou de responder ao remetente. Esses passos podem repetir-se indefinidamente, de acordo com a vontade dos usuários.

### 5.4.2.2 Salas de Conversação

Nesta implementação as salas de conversação do sistema destinam-se exclusivamente aos grupos de trabalho. Através do comando **Entrar no chat**, do menu **Comunicação**, o usuário tem acesso à sala de conversação do grupo selecionado. No servidor, um Agente de Chat é instanciado para cada sala em utilização. O mesmo usuário pode participar de mais de uma sala simultaneamente, bastando trocar o grupo selecionado e utilizar novamente o comando **Entrar no chat**.

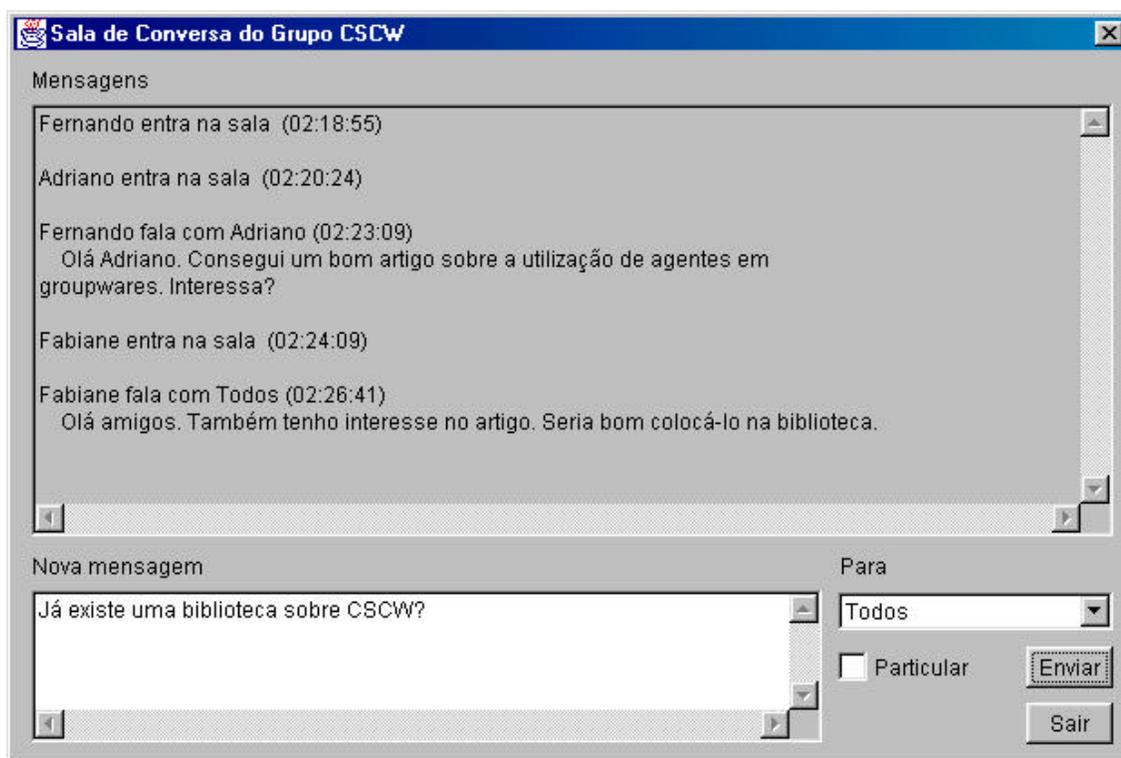


Figura 5-10 – Sala de conversação de um grupo de trabalho.

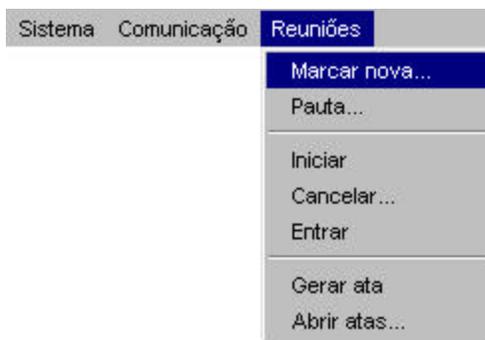
A Figura 5-10 mostra uma sala de conversação de um grupo de trabalho. As mensagens digitadas na parte inferior podem ser enviadas para todos os presentes ou para um usuário escolhido na lista. Uma mensagem pode ser marcada como particular, sendo visível somente para o destinatário. Na parte superior estão as mensagens enviadas e as informações sobre a entrada e saída de usuários.

### 5.4.3 Reuniões a Distância

A arquitetura baseada em agentes rendeu benefícios a todos os pontos da aplicação. Porém, a organização das reuniões a distância é o ponto onde esses benefícios são mais claros. Graças à descentralização de funções promovida pelos agentes, o sistema pode controlar o agendamento e a realização de várias reuniões simultaneamente.

O **Agente de Reunião** foi implementado para tratar de uma reunião desde o momento em que o coordenador do grupo faz a solicitação de agendamento até a geração da ata. Como idealizado no projeto, uma nova instância desta classe de agente é criada sempre que uma reunião é solicitada. Além das atividades de coordenação, o agente é o responsável pelo armazenamento e fornecimento de todos os dados relativos à reunião, como a solicitação do coordenador e a pauta.

A Figura 5-11 mostra o menu que dá acesso aos recursos sobre as reuniões. As opções de *marcar*, *iniciar*, *cancelar* e *gerar ata* só aparecem habilitadas quando o usuário é coordenador do grupo de trabalho visualizado naquele instante. As funções *pauta*, *iniciar*, *cancelar*, *entrar* e *gerar ata* são aplicadas sempre sobre a reunião selecionada na agenda do grupo de trabalho, que será mostrada a seguir.



**Figura 5-11 – Menu com os comandos sobre as reuniões.**

Para descrever a interação dos usuários com os agentes de reuniões, será mostrado passo a passo o processo de agendamento e realização de uma reunião a distância. No exemplo hipotético, o coordenador do grupo de trabalho denominado CSCW deseja marcar uma reunião para tratar do projeto de uma nova ferramenta CASE, destinada a facilitar o desenvolvimento colaborativo de sistemas.

### 5.4.3.1 Agendamento

Quando o coordenador de um grupo de trabalho solicita a marcação de uma nova reunião, deve passar as informações necessárias para o que o agente de reunião que será instanciado possa realizar o agendamento. Através do diálogo mostrado na Figura 5-12, o coordenador fornece o assunto da reunião, sua provável duração e sugere os horários para a sua realização. O agente realiza o agendamento de acordo com a determinação do coordenador, que escolhe entre três opções:

|   | Data       | Hora     |
|---|------------|----------|
| 1 | 28-06-1999 | 21:15:00 |
| 2 | 02-07-1999 | 20:00:00 |
| 3 | 03-07-1999 | 17:30:00 |
| 4 | 04-07-1999 | 20:00:00 |

Figura 5-12 – Diálogo para solicitação de uma nova reunião.

- ✓ **Confirme pela opinião da maioria:** o agente solicita a opinião de todos os membros do grupo e decide pelo horário que recebe o maior número de votos.
- ✓ **Mostre-me as opiniões antes de confirmar:** nesta opção o agente também solicita a opinião dos participantes, mas retorna o resultado ao coordenador para que este decida por algum horário ou faça uma nova solicitação.
- ✓ **Marque a reunião sem consultar o grupo:** neste caso a agente passa direto pela tarefa de consultar os usuários e marca a reunião para o horário informado pelo

coordenador.

O coordenador pode ainda incluir assuntos para a pauta da reunião já no momento da solicitação. O botão ‘Pauta...’ dá acesso à função de composição da pauta da reunião, que será mostrada logo a seguir.

### Procedimentos do Agente de Reunião

Na fase de projeto, no capítulo anterior, a análise estado – evento foi introduzida como o mecanismo que define o comportamento dos agentes dentro do sistema. O agente de reunião possui o diagrama de estados mais interessante dentre as entidades do modelo. Sendo assim, especial atenção será dada na apresentação dos procedimentos tomados por estes agentes na coordenação das reuniões.

Nos próximos parágrafos, à medida que são apresentados os pontos principais da interação entre o agente e os usuários, também é mostrada a evolução no estado interno do agente. Os estados do agente de reunião estão definidos na Tabela 4-1. Os diálogos apresentados são a forma como as mensagens do agente chegam aos usuários. As solicitações e respostas dos usuários são sintetizadas em mensagens específicas e remetidas ao agente correspondente, que está rodando no servidor.

Quando o Roteador recebe a mensagem com a solicitação do coordenador para uma nova reunião, ele instancia um novo agente, que passa a ser o responsável pela reunião solicitada.

Quando o agente é criado, seu estado passa a ser *iniciou*. Se a solicitação do coordenador exigir a opinião dos usuários, o agente solicita as opiniões e passa para o estado *opiniões solicitadas*. Senão, o agente marca a reunião e passa para o estado *reunião marcada*.

No exemplo da Figura 5-12, o coordenador pede que o agente solicite a opinião dos usuários. Os usuários recebem a solicitação através do diálogo mostrado na Figura 5-13. Os horários sugeridos são listados e o usuário pode assinalar quantos horários desejar, podendo inclusive não opinar, caso nenhum horário lhe seja adequado.

Enquanto estiver no estado *opiniões solicitadas*, o agente fica armazenando as respostas dos usuários. O agente só toma a próxima atitude quando **percebe** a ocorrência de uma das seguintes condições:

- ✓ todos os usuários, exceto o coordenador, já opinaram;
- ✓ o horário limite para a coleta das opiniões foi alcançado.

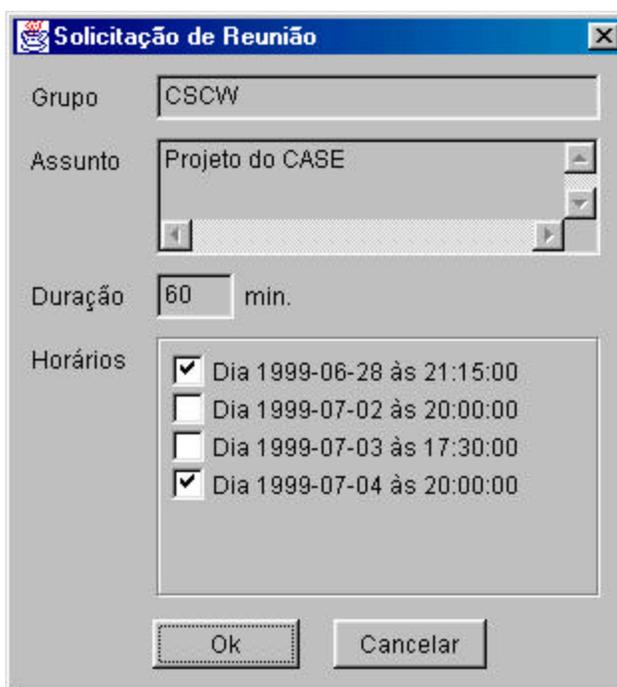


Figura 5-13 – O agente de reunião solicita a opinião do usuário.

Se a solicitação for marcar o horário pela opinião da maioria, o agente marca a reunião para o horário com o maior número de votos. Em caso de empate, o horário mais próximo é escolhido. O estado do agente passa então para *reunião marcada*. Se a solicitação for retornar as opiniões para o coordenador, como no exemplo que está sendo apresentado, o agente envia uma mensagem para o coordenador e fica em novo estado de espera: *coordenador consultado*. O coordenador recebe a mensagem através do diálogo da Figura 5-14. Com base nos resultados, o este deve fornecer uma nova orientação ao agente.

Na opção “*fazer uma nova solicitação*”, o diálogo de solicitação (Figura 5-12) é novamente apresentado. O agente retorna ao estado *iniciou* e passa a agir de acordo com a nova solicitação recebida. Na opção “*marcar o horário selecionado*” o agente

marca a reunião e vai para o estado *reunião marcada*. Existe ainda a possibilidade de o coordenador cancelar a reunião, o que leva o agente para o estado *reunião finalizada*.

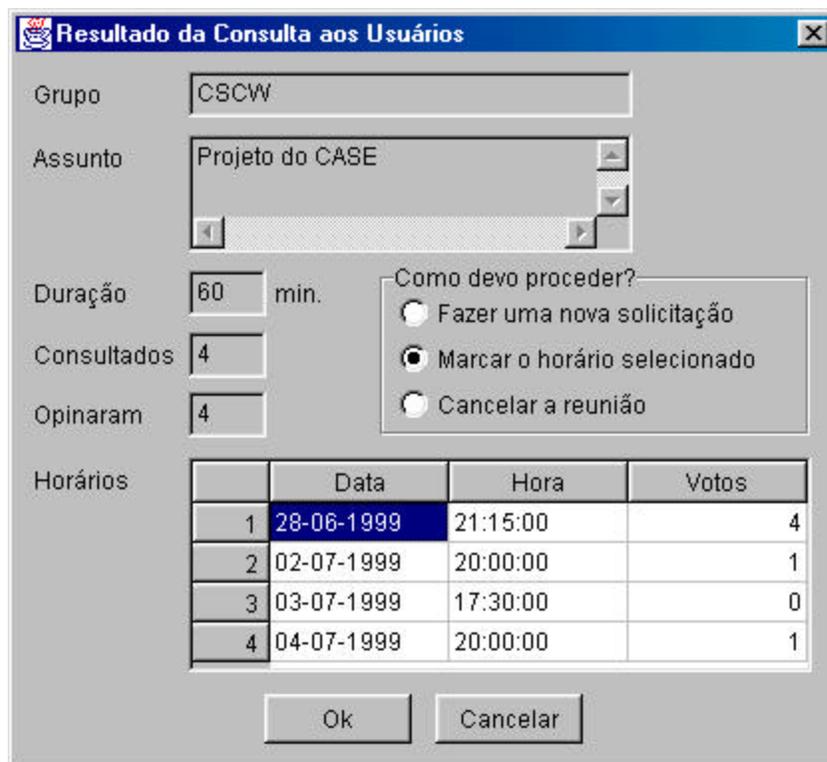
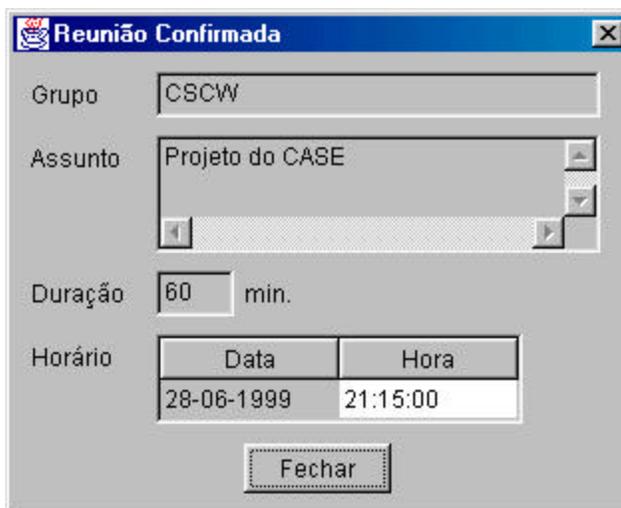


Figura 5-14 – O agente de reunião informa o coordenador sobre as opiniões.

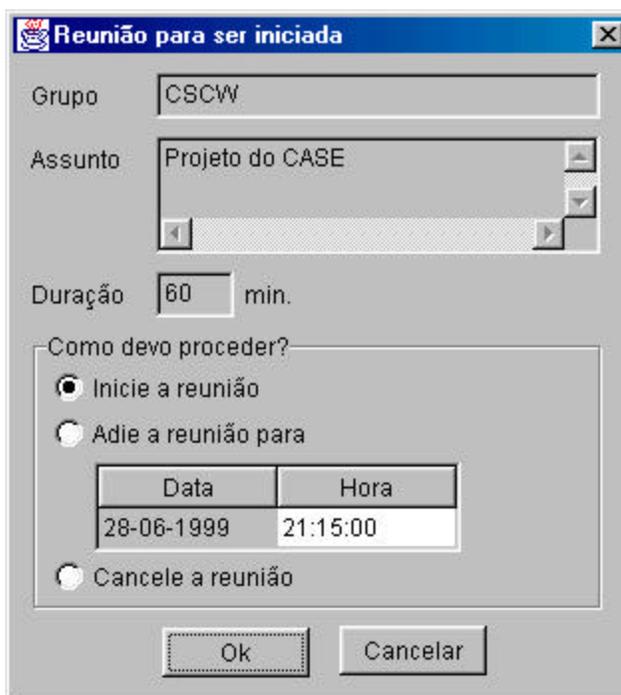
No exemplo apresentado, o coordenador solicita a confirmação da reunião para o horário que recebeu mais votos. O agente envia mensagens de confirmação para todos os usuários no grupo e atualiza a agenda. Sempre que uma reunião é marcada, transferida ou cancelada, o agente responsável repete o procedimento de comunicar os usuários e atualizar a agenda. A Figura 5-15 mostra o diálogo que reporta os avisos de confirmação, adiamento ou cancelamento de reuniões.

Quando o agente está no estado *reunião marcada*, fica no aguardo de novas solicitações. A qualquer momento um usuário pode solicitar a pauta da reunião ou a reunião pode até mesmo ser cancelada pelo coordenador. Além dos eventos externos, o agente monitora a hora do sistema para dar início à reunião no horário determinado.



**Figura 5-15 – Aviso sobre a confirmação de uma reunião.**

Quando o tempo limite é atingido, o agente envia uma mensagem ao coordenador solicitando nova orientação e vai para o estado *início solicitado*. O coordenador recebe o aviso através do diálogo da Figura 5-16.



**Figura 5-16 – Uma reunião deve ser iniciada.**

Neste momento, o coordenador pode orientar o agente a tomar uma das seguintes atitudes: iniciar, adiar para um novo horário ou cancelar a reunião. Se a solicitação for adiar a reunião, o agente notifica os usuários, atualiza a agenda do grupo e retorna para

o estado *reunião marcada*. Se o coordenador optar pelo cancelamento, o agente faz as notificações e vai para o estado *reunião finalizada*.

No exemplo apresentado, o coordenador solicita que o agente inicie a reunião. O agente envia uma notificação a todos os membros do grupo que estiverem *on-line* naquele momento, para que entrem na sala de reunião. A Figura 5-17 mostra o aviso recebido por cada usuário, que pode ser conduzido imediatamente para a sala de reunião, se desejar.

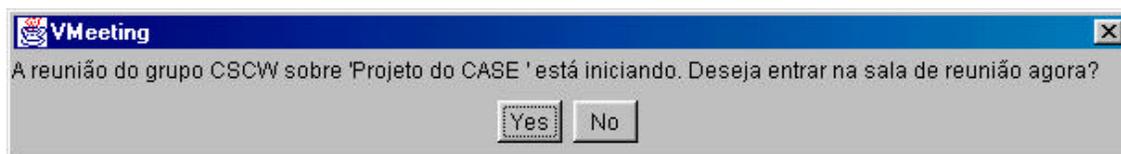


Figura 5-17 – Aviso para o usuário sobre o início de uma reunião.

O estado do agente passa então para *reunião iniciada*, que indica que sua tarefa daqui para diante é receber as mensagens decorrentes da realização da reunião e realizar o processamento adequado.

O aviso do agente não é a única oportunidade para que os usuários ingressem na sala de reunião. Na verdade, o agente suporta o processo de comunicação entre os usuários a partir do momento que a reunião é confirmada na agenda do grupo. Através do comando “**Entrar...**” do menu “**Reuniões**” (Figura 5-11), o usuário entra na sala da reunião selecionada na agenda.

### A Agenda do Grupo

Além de receberem avisos sobre os eventos relativos às reuniões dos grupos, os usuários dispõem de uma agenda com as reuniões confirmadas. Além de informar sobre os compromissos, a agenda possibilita o acesso às funções do menu **Reuniões** (Figura 5-11). Parte dessas funções são aplicadas à reunião selecionada na agenda. A Figura 5-18 mostra a interface da aplicação cliente com a visão “Agenda”. A agenda mostrada refere-se ao grupo selecionado na visão “Grupos”. Os compromissos listados são atualizados *on-line* pelos agentes sempre que ocorrem alterações.

| Data       | Hora     | Dur. | Assunto               |
|------------|----------|------|-----------------------|
| 28-06-1999 | 21:15:00 | 60   | Projeto do CASE       |
| 04-07-1999 | 18:00:00 | 60   | Compra de livros      |
| 05-07-1999 | 20:00:00 | 30   | Congresso de IA       |
| 12-07-1999 | 21:30:00 | 90   | Andamento dos artigos |

Figura 5-18 – A agenda do grupo de trabalho.

### 5.4.3.2 Composição da Pauta

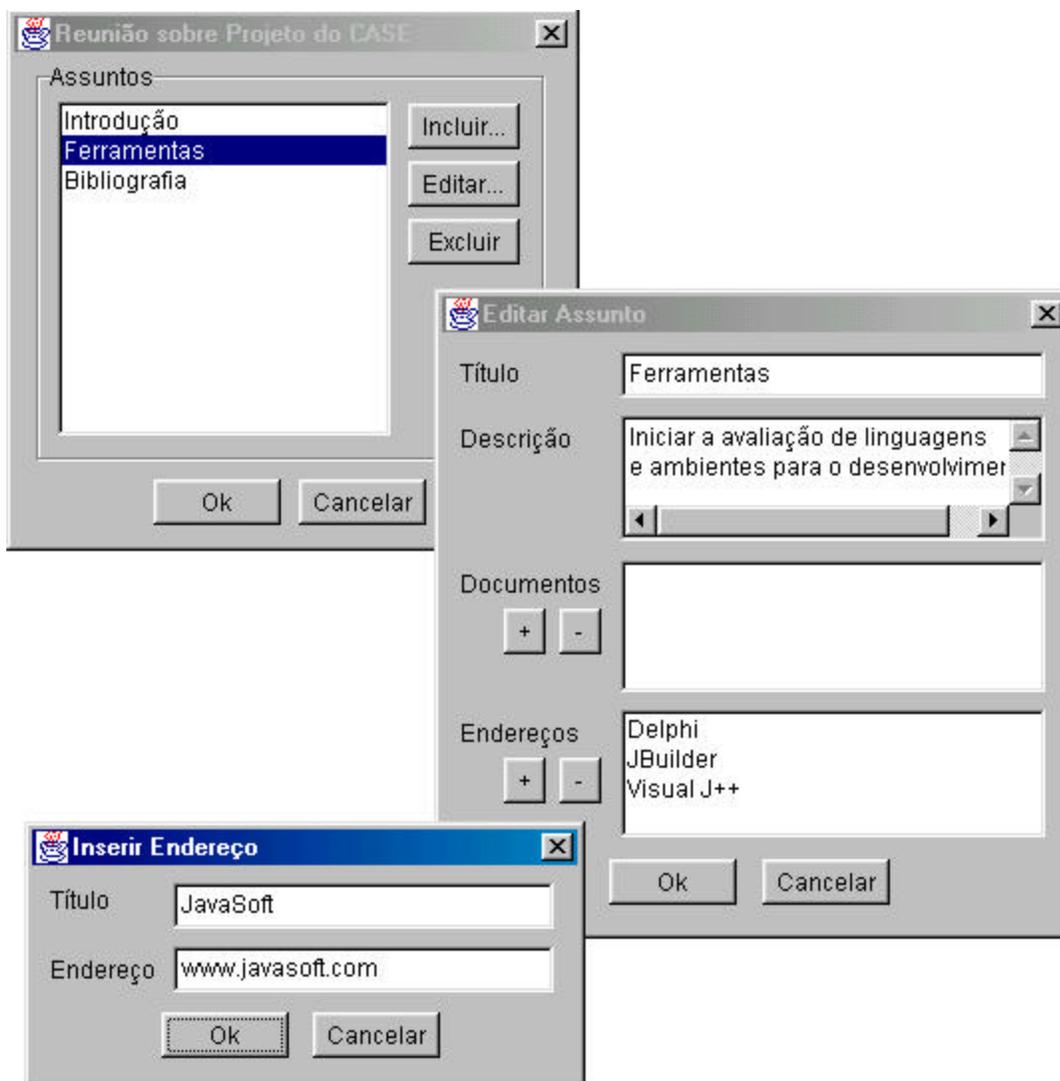
No momento em que faz a solicitação, o coordenador do grupo já pode incluir assuntos na pauta. Acessando o botão “Pauta...” do diálogo da Figura 5-12, o coordenador faz com que a pauta seja instanciada e enviada juntamente com a solicitação para o Roteador, que repassará estas informações para o novo agente. Se o agente não recebe a pauta no momento em que é criado, trata de criar uma pauta vazia.

A partir do momento em que uma reunião é agendada, até a sua realização, todos os membros do grupo de trabalho podem contribuir para a composição da pauta. Para ter acesso à mesma, o usuário seleciona a reunião na agenda do grupo (Figura 5-18) e dispara o comando “**Pauta...**” do menu “**Reuniões**” (Figura 5-11). O agente do usuário recebe a solicitação da interface, envia a mensagem adequada ao agente da reunião selecionada e continua sua atividade de monitoramento de mensagens.

Ao receber a mensagem com a solicitação, o agente da reunião cria uma mensagem de resposta cujo conteúdo é um objeto que representa a pauta. O agente do usuário recebe esta mensagem e usa um serviço da interface para mostrar a pauta ao usuário.

Como especificado no projeto, a pauta foi implementada como um conjunto de assuntos. Cada assunto é composto por documentos, que podem ser arquivos ou

endereços na Internet. Os diálogos mostrados na Figura 5-19 refletem justamente esta estrutura.



**Figura 5-19 – Diálogos para composição da pauta da reunião.**

O primeiro diálogo, na parte superior da figura, surge para o usuário quando o seu agente recebe a pauta solicitada. A partir deste, é possível incluir novos assuntos, excluir ou alterar assuntos já existentes. O segundo diálogo, mais à direita, é utilizado na criação ou edição de um assunto. Além de definir o título e a descrição do assunto, está disponível a inclusão e exclusão de documentos a serem utilizados durante a realização da reunião. O usuário pode incluir arquivos do seu sistema local ou endereços de páginas na Internet. O último diálogo apresentado é utilizado para a inclusão de um novo endereço de Internet no assunto editado.

Quando o usuário confirma as alterações, fechando com “Ok” o diálogo principal, a pauta é enviada pelo agente do usuário ao agente da reunião, responsável pelo seu armazenamento.

### 5.4.3.3 A Sala de Reuniões

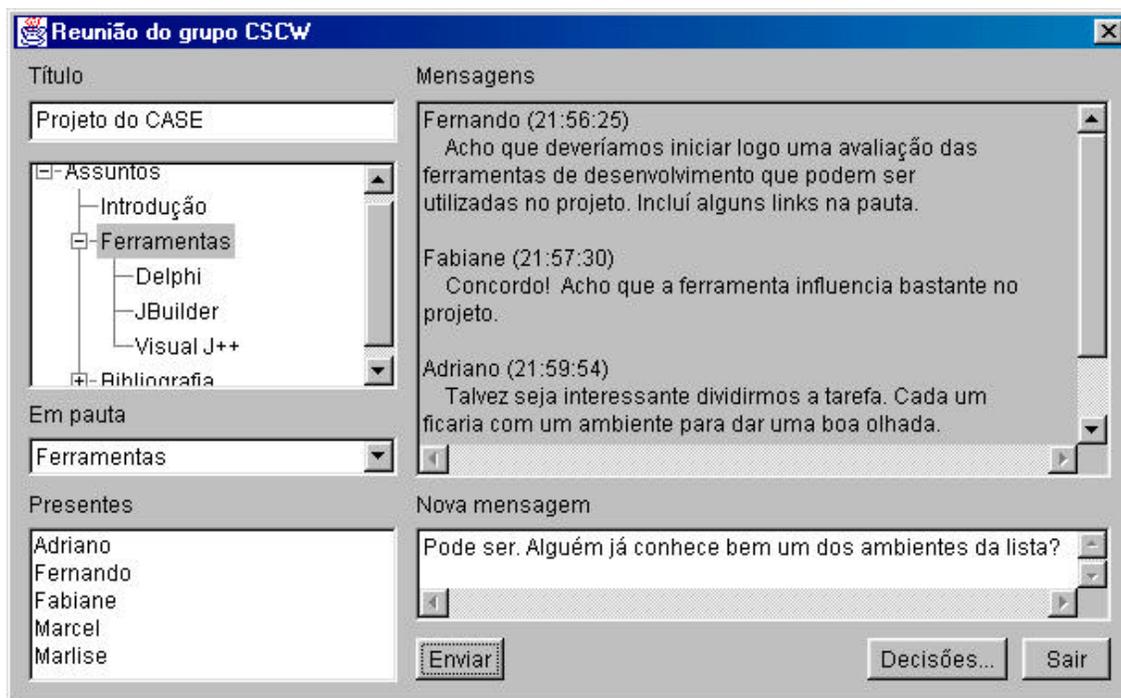


Figura 5-20 – A sala de reuniões.

Ao solicitar a entrada em uma reunião, seja por iniciativa própria ou respondendo ao aviso do agente, o usuário é conduzido para a sala da reunião, representada pela janela da Figura 5-20. O sistema permite que um usuário participe de mais de uma reunião simultaneamente.

Na barra de título da janela está indicado o nome do grupo de trabalho. Na coluna da esquerda estão as informações relativas à pauta da reunião e a relação dos presentes. Na direita está o espaço para a conversação e o botão que dá acesso à janela para anotação das decisões, que será mostrada a seguir.

#### Apresentação da Pauta

Logo abaixo do título da reunião é apresentada a pauta de assuntos. Através da estrutura de “árvore” a visualização é facilitada. Os nós representando os assuntos

podem ser expandidos, exibindo assim os documentos relacionados com cada assunto. Um duplo clique do mouse sobre um nó da árvore exhibe o seu conteúdo. Se o nó representa um assunto, o diálogo de assunto (Figura 5-19) é aberto para visualização, não permitindo alterações. Se o nó for um documento, ou seja, um arquivo ou um endereço na Internet, uma solicitação ao sistema operacional é feita para que o seu conteúdo seja exibido. No caso de um endereço na Internet, por exemplo, o navegador configurado no sistema operacional é aberto e a página endereçada é carregada.

A exibição de documentos é o único ponto da implementação onde se fez necessária a utilização de funções específicas do sistema operacional. Para tanto, foi implementada uma pequena DLL em linguagem C++. Para cada sistema operacional em que se deseja utilizar a aplicação é necessária uma implementação específica desta DLL. A máquina virtual Java carrega a versão adequada da DLL em tempo de execução quando alguma função da mesma for utilizada. Para este trabalho, apenas a versão para o ambiente Windows foi implementada.

Logo abaixo da pauta está uma caixa de escolha que define o assunto em discussão num dado instante. Este controle está habilitado somente para o coordenador do grupo. Quando este muda o assunto em pauta, o agente da reunião trata de notificar os agentes de todos os usuários presentes. O assunto em pauta é então alterado automaticamente na janela de cada um.

### Usuários Presentes

Abaixo do assunto em pauta está a relação dos usuários que estão conectados à reunião a cada instante. O agente da reunião recebe notificações dos agentes dos usuários sempre que alguém entra ou sai da sala. Todos os presentes são então informados pelo agente da reunião sobre a nova situação da sala.

### Conversação

A conversação entre os usuários na sala também é intermediada pelo agente da reunião, já que este está constantemente informado sobre quem está presente. Na direita da janela está o espaço destinado à conversação. Acima são visualizadas as mensagens enviados pelos participantes e logo abaixo está o espaço para o envio das mensagens, que neste caso são abertas a todos os participantes.

## Decisões para a Ata

Para que a ata da reunião possa ser posteriormente gerada pelo agente, os participantes devem anotar as decisões tomadas e outras considerações importantes. O botão “**Decisões...**” abre uma janela para esta finalidade. Esta janela reflete sempre o assunto colocado em pauta pelo coordenador. A Figura 5-21 mostra o coordenador do grupo anotando uma nova decisão no assunto que trata das ferramentas de desenvolvimento.

Sempre que as decisões são editadas, todos os participantes têm suas janelas atualizadas pelo agente da reunião. Quando o assunto em pauta é alterado, a janela das decisões passa a exibir as anotações relativas ao novo assunto.

Todas as anotações realizadas são armazenadas pelo agente da reunião. Elas deverão posteriormente compor a ata, juntamente com a pauta e outras informações relevantes.

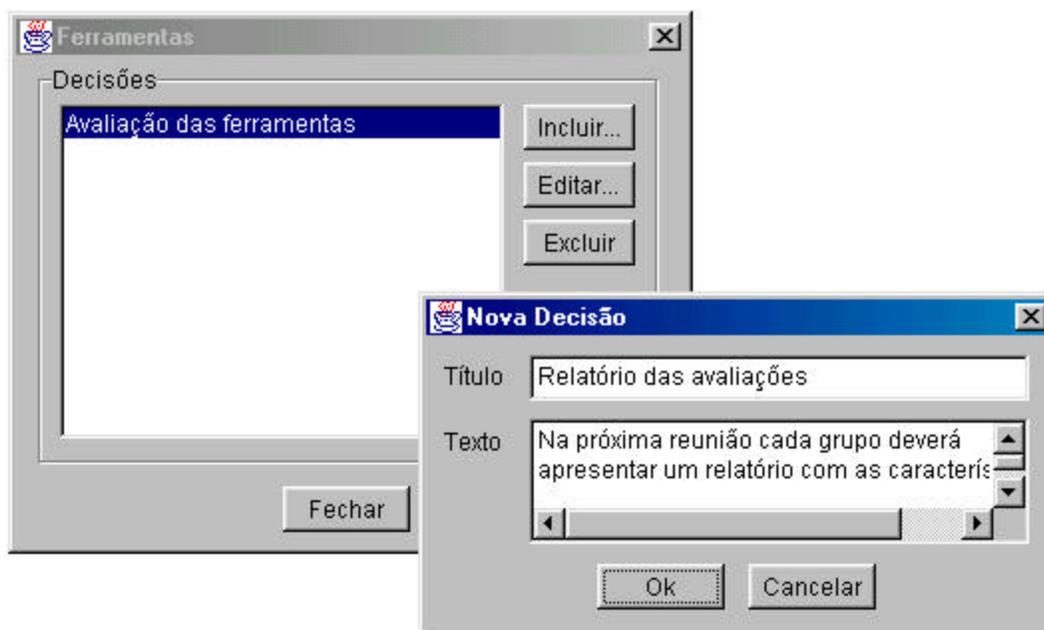


Figura 5-21 – Anotação das decisões tomadas durante a reunião.

### 5.4.3.4 Geração da Ata

Antes de finalizar a reunião, o coordenador pode solicitar que o agente realize a geração de uma ata, através do comando “**Gerar ata**”, do menu “**Reuniões**” (Figura

5-11). Ao receber a solicitação, o agente da reunião utiliza as informações contidas na solicitação, na pauta e nas decisões, para criar um objeto que representa a ata da reunião. Este objeto é armazenado em um cadastro especial no servidor, onde as atas são separadas por grupo de trabalho. A responsabilidade sobre este cadastro é do Roteador.

O sistema possibilita que o coordenador solicite mais que uma vez a geração da ata de uma reunião. Por exemplo, o coordenador pode atualizar a ata caso a reunião tenha sido retomada por algum motivo e novas decisões tenham sido tomadas. Neste caso, o agente simplesmente substitui a ata anterior.

Normalmente a geração da ata é a última atividade associada a uma reunião. Depois disso o coordenador pode eliminar a reunião da agenda através do comando “**Cancelar...**”, do menu “**Reuniões**” (Figura 5-11).

#### Visualização das Atas



| Data       | Hora     | Assunto          |
|------------|----------|------------------|
| 28-06-1999 | 21:15:00 | Projeto do CASE  |
| 04-07-1999 | 18:00:00 | Compra de livros |
| 05-07-1999 | 20:00:00 | Congresso de IA  |

Figura 5-22 - Arquivo de atas de um grupo de trabalho.

Um usuário pode, a qualquer momento, solicitar a visualização das atas de um grupo de trabalho através do comando “**Abrir atas...**”, do menu “**Reuniões**” (Figura 5-11). A solicitação é recebida pelo Roteador do sistema, que abre o arquivo referente ao grupo e envia para o usuário os objetos representado as atas em uma única

mensagem. Quando as informações são recebidas na aplicação cliente, a janela da Figura 5-22 é aberta para o usuário.

|   |
|---|
| <p>ATA DE REUNIÃO</p> <p>Grupo: CSCW<br/>Assunto: Projeto do CASE<br/>Coordenador: Fernando Gauthier</p> <p>No dia 1999-06-28 às 21:15, os membros do grupo reuniram-se para tratar dos seguintes assuntos:</p> <ul style="list-style-type: none"> <li>- Introdução</li> <li>- Ferramentas</li> <li>- Bibliografia</li> </ul> <p>Os objetivos iniciais e as decisões tomadas sobre cada assunto são colocados a seguir.</p> <p>-----</p> <p>Assunto: Introdução</p> <p>Objetivo: Falar sobre trabalho colaborativo no desenvolvimento de software. Discutir as Possibilidades de um CASE para desenvolvimento colaborativo de software.</p> <p>-----</p> <p>Assunto: Ferramentas</p> <p>Objetivo: Iniciar a avaliação de linguagens e ambientes para o desenvolvimento.</p> <p>Decisões tomadas:</p> <ol style="list-style-type: none"> <li>1. Avaliação das ferramentas<br/>Esta tarefa fica assim dividida: Adriano e Fabiane fazem testes com o JBuilder e com o Visual J++. Marcel e Marlise avaliam o Delphi.</li> <li>2. Relatório das avaliações<br/>Na próxima reunião cada grupo deverá apresentar um relatório com as características da ferramenta avaliada. A opinião pessoal sobre a usabilidade é fundamental.</li> </ol> <p>-----</p> <p>Assunto: Bibliografia</p> <p>Objetivo: Iniciar a pesquisa de bibliografia que contribua para o projeto.</p> <p>Decisões tomadas:</p> <ol style="list-style-type: none"> <li>1. Compra de livros<br/>O Adriano e a Marlise ficam encarregados de procurar livros que interessem para o desenvolvimento do projeto. Na próxima reunião veremos o que é viável para compra.</li> <li>2. Pesquisa na Internet<br/>A Fabiane e o Marcel ficam responsáveis pela pesquisa de artigos na Internet. Pode ser criada uma biblioteca específica sobre CSCW no sistema.</li> </ol> |
|---|

**Figura 5-23 – Uma ata de reunião gerada pelo agente.**

O botão “Abrir” faz com que a ata associada à linha selecionada na tabela seja efetivamente gerada em um arquivo texto, a partir das informações estruturadas armazenadas. Uma solicitação ao sistema operacional é feita para que o arquivo seja aberto pelo editor de texto configurado, como foi feito na visualização dos documentos

da pauta da reunião. A Figura 5-23 mostra a ata gerada pelo agente da reunião hipotética sobre “Projeto do CASE”.

Ainda na janela de visualização das atas, o coordenador do grupo tem a possibilidade de solicitar a exclusão das atas cujo armazenamento não seja mais interessante.

#### 5.4.4 Bibliotecas

O último requisito funcional para a aplicação diz respeito ao compartilhamento de informações através de bibliotecas. Como especificado na fase de projeto, um agente específico foi implementado com a finalidade de cuidar do armazenamento e viabilizar a composição e utilização das bibliotecas. Todas as funções da aplicação cliente relativas à manipulação das bibliotecas geram requisições ou notificações destinadas ao agente de biblioteca, que está executando no servidor.

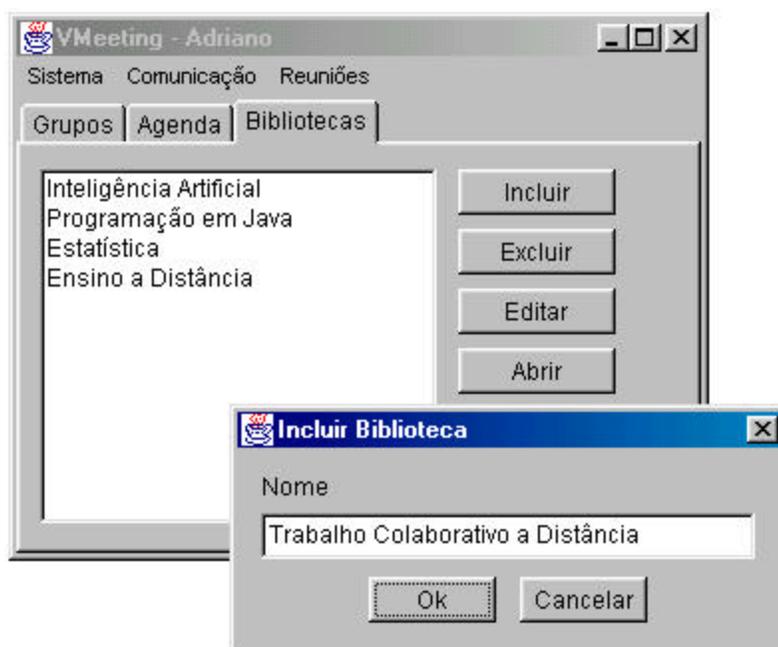
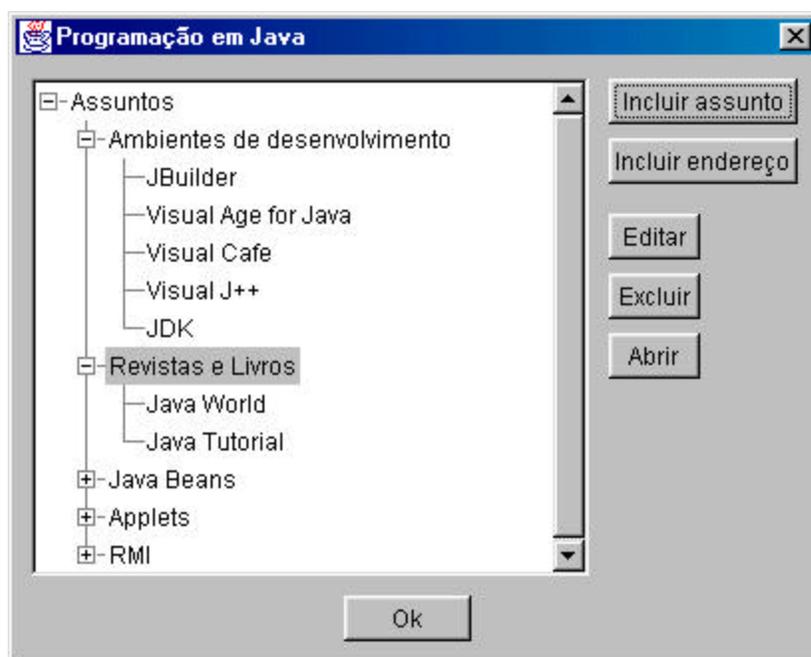


Figura 5-24 – Manipulação das bibliotecas do sistema.

Para prover uma melhor organização das informações, várias bibliotecas podem ser criadas no sistema. Cada uma constitui um conjunto dinâmico de assuntos, que contêm diversos documentos. As classes que representam os assuntos e os documentos

são exatamente as mesmas utilizadas na implementação da pauta de reunião. As bibliotecas são comuns a todos os grupo de trabalho.

A Figura 5-24 mostra a visão “Bibliotecas” da interface da aplicação cliente. Ao lado da lista de bibliotecas estão os botões com as funções para inclusão, exclusão, edição e visualização das bibliotecas. Quando uma biblioteca é criada, o agente de biblioteca armazena a identificação do usuário responsável pela criação. Somente a este é permitido excluir a biblioteca ou alterar seu nome. As funções excluir, editar e abrir, sempre dizem respeito à biblioteca selecionada na lista.



**Figura 5-25 – Visualização de uma biblioteca.**

Quando o usuário abre uma biblioteca, seu conteúdo é exibido através da janela da Figura 5-25. A estrutura de “árvore” mostra os assuntos com seus documentos. Um duplo clique sobre o nó selecionado (equivalente ao botão “Abrir”), faz com que este seja visualizado. Se o nó representar um assunto, o diálogo para edição é aberto. Se o nó for um documento, sua exibição é solicitada ao sistema operacional. Os botões “Editar” e “Excluir” também referem-se sempre ao nó selecionado na árvore. O botão “Incluir endereço” facilita a inclusão de um endereço Internet no assunto selecionado, sem a necessidade de abrir o diálogo de edição. Os diálogos para a edição dos assuntos e documentos são os mesmos empregados na composição da pauta de uma reunião (Figura 5-19).

Quando um usuário realiza uma alteração sobre as bibliotecas, como a criação de uma nova biblioteca ou a inclusão de um documento, o agente trata de notificar todos os usuários *on-line* no sistema. Desta forma, os usuários estão sempre com a visão atualizada sobre a situação das bibliotecas.

## **5.5 Considerações Finais**

### **5.5.1 Cumprimento dos Requisitos**

Dos requisitos estabelecidos na fase de modelagem, apenas dois não foram completamente atendidos na fase de implementação: a coleta de arquivos e as notificações do agente de biblioteca.

#### Coleta de Arquivos

Os documentos incluídos nas pautas de reuniões e nas bibliotecas do sistema podem ser representados por endereços na Internet ou por arquivos do sistema local do usuário. Com os endereços na Internet não há problema, já que estes constituem uma informação de fácil representação. Já os arquivos precisam ser transportados do sistema local dos usuários até o servidor, para que possam ser disponibilizados aos demais usuários interessados.

Os agentes de reunião e de biblioteca foram projetados para realizar a tarefa de coleta dos arquivos incluídos pelos usuários. Quando estes agentes recebem um evento do temporizador e constatam que nenhuma tarefa precisa ser realizada, eles verificam os documentos nos assuntos da pauta ou da biblioteca (dependendo do agente) e tentam buscar os arquivos pendentes caso os usuários responsáveis estejam conectados. Toda a inteligência da parte dos agentes foi implementada para este fim. No entanto, o serviço de transferência de arquivos não foi concluído até a apresentação deste trabalho.

#### Notificações do Agente de Biblioteca

Um dos requisitos relativos ao compartilhamento de informações era a possibilidade dos usuários solicitarem o recebimento de notificações sobre alterações nos assuntos de seu interesse. Estava prevista uma espécie de cadastro associado às

bibliotecas, onde cada usuário poderia selecionar os assuntos sobre os quais gostaria de ser notificado quando surgissem novidades.

O modelo para o agente de biblioteca prevê esta possibilidade. Este agente deveria se responsabilizar pelo armazenamento deste cadastro e por realizar as notificações na medida em que os assuntos recebessem alterações. Na implementação do agente esta funcionalidade não foi incluída. O agente gera apenas notificações para todos os usuários conectados sempre que uma alteração ocorre nas bibliotecas, com objetivo de que os usuários tenham sempre uma visão consistente sobre a situação sobre as mesmas.

### **5.5.2 Consistência entre Projeto e Implementação**

Um dos grandes problemas da metodologia de desenvolvimento conhecida como programação estruturada, já obsoleta, diz respeito à falta de uma notação consistente durante as fases de análise, projeto e implementação. Em cada etapa, mecanismos diferentes são utilizados, criando uma barreira que dificulta, entre outras coisas, a alteração do projeto e da implementação em função de mudanças nos requisitos.

A programação orientada a objetos soluciona este problema através da adoção de uma notação uniforme durante todas as fases do desenvolvimento. Os conceitos de classes, objetos, atributos e serviços podem ser utilizados desde a fase de análise até a implementação.

Neste trabalho este benefício foi bastante notado. O modelo obtido até a fase de projeto foi mapeado praticamente na sua totalidade na implementação. Isto facilita a manutenção e evolução da aplicação, já que os diagramas de classe obtidos no projeto fornecem uma idéia clara sobre como a implementação foi realizada.

### **5.5.3 Performance**

A independência entre a aplicação e o sistema operacional é uma das grandes vantagens da programação em Java. Graças ao mecanismo conhecido como Máquina Virtual Java, o mesmo código pode rodar em diferentes plataformas sem qualquer necessidade de rescrita ou recompilação.

Este benefício, entretanto, gera certos custos em termos de performance. O código de um aplicativo em Java não é executável diretamente pelo sistema operacional. O programa precisa ser interpretado pela máquina virtual para então ser efetivamente executado. À medida que cresce a complexidade do código, a performance da aplicação tende a cair.

Na implementação realizada neste trabalho, é possível se notar esta deficiência. Na interface, as janelas exibidas demoram mais do que o normal para surgir na tela, o que se torna um pouco incômodo para o usuário. Além disso, a performance geral da aplicação cai à medida que cresce o número de *threads* executando na mesma máquina. Embora estas deficiências não impeçam a utilização do sistema, uma proposta com objetivo comercial deveria levá-las em conta e buscar alternativas, como compiladores e máquinas virtuais mais otimizados.

## 6 Conclusões

Neste trabalho foram apresentadas as principais fases do desenvolvimento de uma aplicação de suporte ao trabalho colaborativo. Cada etapa relatada foi fundamental para o resultado obtido, desde o levantamento teórico, envolvendo a área de aplicação e as tecnologias utilizadas, até a implementação em linguagem Java.

A pesquisa realizada sobre CSCW revelou uma área de aplicação extremamente promissora, que atrai a atenção de pesquisadores e empresas de desenvolvimento por todo o mundo. O interesse pela área é impulsionado por uma série de transformações que têm ocorrido nas organizações, na sociedade como um todo e na própria tecnologia. A tendência de reestruturação das organizações visando competitividade, a necessidade crescente de se trabalhar remotamente e a integração das empresas com seus fornecedores, são alguns entre os muitos fatores que incentivam o desenvolvimento de aplicações neste campo.

O estudo das tendências de desenvolvimento e a observação empírica de atividades colaborativas serviram para traçar as linhas de interesse da aplicação desenvolvida. O suporte à comunicação entre os usuários, a coordenação e realização de reuniões em tempo real, e o compartilhamento de informações, foram identificados como objetivos centrais.

Dentre as tecnologias aplicadas na implementação, destaca-se o conceito de agentes inteligentes. Estes têm sido empregados na composição de alguns sistemas comerciais e suas possibilidades na área de CSCW são consideradas por diversos autores. No entanto, o estágio atual de desenvolvimento revela que a exploração desta tecnologia ainda está no seu início. Os agentes ainda são inseridos nas aplicações colaborativas como elementos auxiliares, capazes de executar tarefas interessantes em favor do usuário, principalmente no campo do gerenciamento de informações pessoais.

O modelo apresentado no capítulo 4 propõe a utilização de agentes como entidades fundamentais na arquitetura da aplicação. Nesta abordagem, a complexidade

associada à aplicação como um todo pôde ser dividida de maneira eficiente. Cada tipo de agente que compõe a arquitetura administra um conjunto bem determinado de problemas, ou seja, uma parte da complexidade do sistema. Quando o problema atribuído a determinado tipo de agente foi considerado muito complexo, adotou-se uma abordagem onde um agente é criado para tratar de cada instância do problema. É o que ocorre, por exemplo, com os agentes de reunião. Cada instância trata da coordenação de apenas uma reunião, o que permite ao sistema tratar de várias reuniões simultaneamente, sem a necessidade de estruturas mais complexas.

O paradigma da programação orientada a objetos, largamente utilizado já há algum tempo, é extremamente compatível com a proposta de agentes inteligentes. Sucintamente, agentes podem ser considerados extensões de objetos, que além de atributos e serviços, possuem alguma espécie de mecanismo que lhe garante um comportamento ativo no seu ambiente computacional. Não seria muita pretensão referir-se à tecnologia de agentes como uma grande revolução no desenvolvimento de sistemas, onde os conceitos da orientação a objetos são estendidos para a obtenção de entidades dotadas de autonomia e inteligência, capazes de agir em favor dos usuários e de outros sistemas mesmo longe destes.

Durante a elaboração do modelo foram tratados diversos problemas que não são inerentes somente ao sistema desenvolvido, mas também a diversos tipos de aplicações distribuídas. Sempre que possível, as soluções para tais problemas foram modeladas de forma genérica. Desta forma, foram obtidas algumas estruturas interessantes, que podem ser utilizadas tanto na evolução do próprio sistema quanto na elaboração de novas aplicações. O protocolo de comunicação orientado a objetos e a estrutura extensível para o desenvolvimento de agentes constituem as principais contribuições neste sentido.

Além de explorar a tecnologia de agentes, a aplicação desenvolvida utiliza o Internet como plataforma de rede. Pode-se afirmar que esta escolha se traduz em tranquilidade, tanto para desenvolvedores como para usuários. A Internet é uma rede de alcance global, integrando plataformas heterogêneas de maneira transparente para as aplicações que rodam sobre ela. O protocolo TCP/IP é largamente utilizado, constituindo um padrão já há alguns anos. Além disso, o protocolo não está restrito à Internet, sendo bastante utilizado em redes locais e aplicável à redes dedicadas com

qualquer abrangência. Desta forma, a aplicação está preparada para ser utilizada tanto em Intranets quanto à longa distância, dependendo da necessidade de cada organização.

A integração de plataformas computacionais heterogêneas, altamente desejável em aplicações colaborativas, foi fator determinante na escolha da linguagem de programação. A linguagem Java foi idealizada para que os programas gerados executem sobre uma máquina virtual, que funciona como uma camada isolante entre os programas de aplicação e o sistema operacional. Durante a implementação, entretanto, foi possível observar que os benefícios desta linguagem vão bastante além. O conceito de aplicação *multithread* está embutido na plataforma Java, o que facilitou muito a implementação dos agentes e de outras entidades autônomas que compõem a aplicação, como o Roteador e o Comunicador do cliente. A API fornece um conjunto completo de classes para comunicação, isentando o desenvolvedor das preocupações relativas à utilização dos protocolos de rede.

Em contrapartida, os benefícios trazidos pela máquina virtual implicam em certos custos na performance da aplicação. Quando aumenta o número e a complexidade das *threads* executando no mesmo ambiente, a performance global acaba sendo prejudicada. Outro problema é a ausência de plena compatibilidade entre máquinas virtuais de diferentes fabricantes, e até entre versões do mesmo fabricante. Esta situação pôde ser verificada durante o desenvolvimento, nas tentativas de executar o sistema sobre diferentes máquinas virtuais.

Acredita-se estarem cumpridos os objetivos propostos no início deste trabalho. A adoção da tecnologia de agentes inteligentes possibilitou a obtenção de uma aplicação com elementos diferenciais, principalmente no que diz respeito às facilidades de comunicação entre os usuários e à coordenação de reuniões a distância.

## 6.1 Trabalhos Futuros

As possibilidades para o sistema desenvolvido neste trabalho estão longe de ser esgotadas, mesmo sem se expandir o campo de abrangência delimitado no início do capítulo 4. Algumas possibilidades para a continuação do projeto são sugeridas nesta seção.

### Agendamento de Reuniões

Na funcionalidade de agendamento de reuniões, o coordenador, ao solicitar uma nova reunião, pode sugerir um conjunto de horários. A escolha definitiva é baseada na opinião dos usuários do grupo. A elaboração de um modelo de agenda individual, onde os usuários poderiam fornecer os horários em que estão disponíveis para o grupo de trabalho, aumentaria bastante o grau de autonomia dos agentes nesta atividade. Na solicitação de uma reunião, o próprio sistema poderia sugerir horários ao coordenador, procurando conciliar a agenda dos membros do grupo. A possibilidade de se conseguir marcar um horário com apenas uma interação aumentaria ainda mais.

### Atas das Reuniões

O gerenciamento das atas de reuniões também dá margem a alguns aperfeiçoamentos. Por exemplo, durante a realização de uma reunião, o agente poderia estar preparado para recuperar, através de uma pesquisa inteligente, atas de reuniões anteriores, relacionadas com a reunião em andamento. Normalmente, decisões tomadas em uma reunião acabam constituindo a pauta de reuniões futuras, às vezes para efeito de verificação do seu cumprimento.

### Compartilhamento de Documentos

Um recurso previsto no levantamento de requisitos, mas que não foi totalmente implementado, previa a coleta de documentos (arquivos) a partir do sistema local dos usuários para serem armazenados no servidor. Esses documentos podem fazer parte das pautas de reuniões ou das bibliotecas do sistema. Disponíveis no servidor, os arquivos poderiam ser transportados até os sistemas dos usuários que desejassem consultá-los. Seria interessante finalizar a implementação deste recurso, através do transporte dos arquivos pela rede, ou de outras abordagens. Atualmente, a publicação automática de documentos em páginas *Web* é facilitada por alguns aplicativos e também poderia ser considerada neste ponto.

### Notificações sobre as Bibliotecas

Outro recurso que não passou da fase de projeto, previa a geração de notificações aos usuários quando ocorressem alterações de interesse nas bibliotecas do sistema.

Embora a visão do usuários sobre as bibliotecas seja constantemente atualizada pelo agente de biblioteca, a idéia do usuário se cadastrar em assuntos de interesse e receber avisos sobre as novidades seria um complemento interessante.

### Personificação dos Agentes

Outra sugestão interessante diz respeito à personificação dos agentes, principalmente do agente do usuário. Este agente executa no cliente da aplicação, recebendo as mensagens dos agentes do servidor e repassando-as ao usuário quando for necessário. A maneira como as mensagens são reportadas ao usuário lembra muito o modelo rígido das aplicações tradicionais. A figura de um agente, um auxiliar do usuário, não está realmente clara na interface. Para ilustrar a viabilidade dessa idéia, vale citar uma biblioteca da Microsoft, denominada *MS Agent*<sup>4</sup>. Sua API permite que os agentes de uma aplicação sejam personificados de diversas maneiras, incluindo emissão de sons, vozes sintetizadas e representação corporal.

---

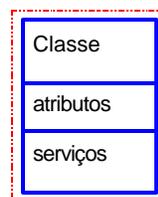
<sup>4</sup> Uma visão sobre esta biblioteca pode ser obtida em [http://msdn.microsoft.com/library/periodic/period98/html/sl\\_agents.htm?RLD=79](http://msdn.microsoft.com/library/periodic/period98/html/sl_agents.htm?RLD=79)

## Apêndice I – Notação para o Projeto Orientado a Objetos

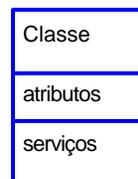
O modelo de classes e objetos apresentados no capítulo 4 utilizam a notação desenvolvida por Coad e Yourdon, descrita em [COA 93]. O *software* OOTHER, distribuído gratuitamente, foi empregado na composição dos diagramas de classes. Os símbolos utilizados são descritos resumidamente neste apêndice, como suporte para um melhor entendimento do modelo apresentado.

### Símbolo Classe & Objeto

Representa uma classe e seus objetos. O retângulo mais claro indica que a classe pode ser diretamente instanciada na aplicação, não representando apenas um conceito abstrato. O símbolo é dividido internamente em três espaços, que armazenam, de cima para baixo, o nome da classe, os atributos e os serviços.



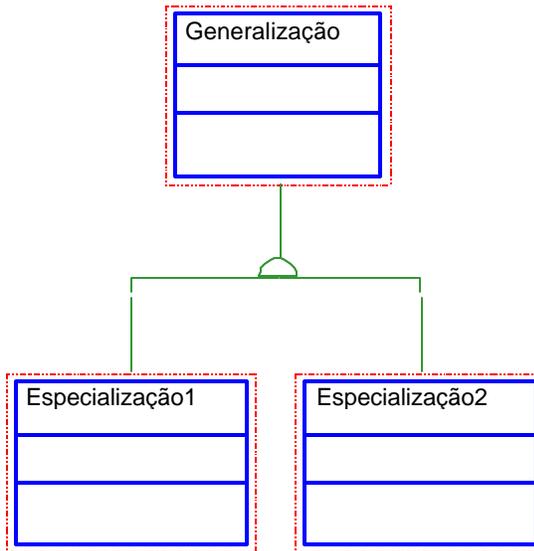
### Símbolo Classe



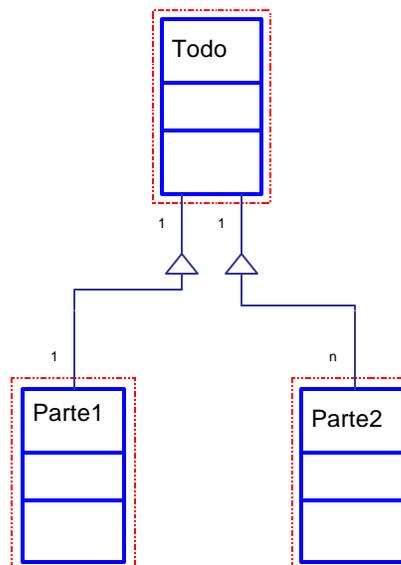
É uma especialização do símbolo Classe & Objeto, utilizada para representar classes abstratas, ou seja, que não representam objetos do modelo. A única diferença é a ausência do retângulo mais externo.

### Estrutura Generalização – Especialização

Utilizada para representar a relação de herança entre as classes. A classe mais genérica está acima das classes especializadas e um símbolo com um semi-círculo representa a relação entre elas.



### Estrutura Todo – Parte

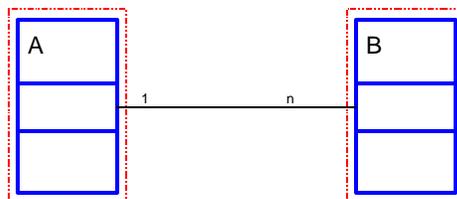


Representa a relação de composição entre as classes. A classe “Todo” pode ser composta por uma ou várias “Partes”. Esta estrutura é representada por um triângulo

com o vértice superior apontado para a classe que representa o todo. Em cada extremidade da linha é indicado número de partes que pode compor o todo e vice-versa.

### Conexão de Ocorrência

Serve para designar mapeamentos entre objetos individuais, não entre classes. Não representam uma composição fixa no modelo. São conexões que ocorrem durante a existência dos objetos, quando, por exemplo, um objeto da classe A instancia e mantém referências para diversos objetos da classe B por algum tempo. Uma linha entre os objetos, contendo a multiplicidade dos mesmos, representa esta relação.



### Conexão de Mensagem

É uma seta clara, apontando no sentido do emissor para o receptor, indicando a emissão de mensagem, possivelmente para solicitar algum serviço.



## Referências Bibliográficas

### Trabalho Colaborativo Suportado por Computador

- [AOL 99] AOL COMPUTING'S WEBOPEDIA. <http://aol.pcwebopedia.com>
- [ARA 95] Renata Mendes de Araujo, Maria C. R. Cavalcanti, Maria Luiza Campos, Marcos R. S. Borges. **CSCW, Groupware & Internet**. Relatório Técnico, Universidade Federal do Rio de Janeiro, 1995.  
<http://www.cos.ufrj.br/~renata/cscw>
- [BAN 91] Liam J. Bannon, Kjeld Schmidt. **CSCW: Four Characters in Search of a Context**. Em "J. Bowers & S. Benford (Eds.) *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*. Amsterdam North-Holland, pp 3-16. 1991 –  
<http://www.ul.ie/~idc/library/papersreports/LiamBannon/4/ECCSCW89.html>
- [BAN 92] Liam J. Bannon. **Discovering CSCW**. 5<sup>th</sup> Information Systems Research in Scandinavia (IRIS), Larkollen, Norway, 1992, pp 507-520.  
<http://www.ul.ie/~idc/library/papersreports/LiamBannon/10/IRISCSCW.html>
- [BAN 93a] Liam J. Bannon. **The Context of CSCW**. Artigo introdutório em "k. Schmidt (ed.) *Report of COST14 'CoTech' Working Group 4 (1991-1992)*, pp. 9-36", Fevereiro de 1993 –  
<http://www.ul.ie/~idc/library/papersreports/LiamBannon/1/BannonHughes.html>
- [BAN 93b] Liam J. Bannon. **CSCW: An Initial Exploration**. Scandinavian Journal of Information Systems, Vol. 5, pp 3-24, Agosto de 1993.  
<http://www.ul.ie/~idc/library/papersreports/LiamBannon/26/SJISFin.html>
- [BEN 97a] R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor, G. Woetzel. **Basic Support for Cooperative Work on the World Wide Web**. International Journal of Human Computer Studies: Special issue on Novel Applications of the WWW, Spring 1997, Academic Press, Cambridge. <http://bscw.gmd.de/Papers/IJHCS.html>
- [BEN 97b] Richard Bentley, Thilo Horstmann, Jonathan Trevor. **The World Wide Web as Enabling Technology for CSCW: The Case of BSCW**. CSCW: The Journal of Collaborative Computing, 2-3, 1997, Kluwer Academic Press, Amsterdam. [http://bscw.gmd.de/Papers/CSCWJ-WWW/CSCW\\_journal.html](http://bscw.gmd.de/Papers/CSCWJ-WWW/CSCW_journal.html)

- [BER 95] Lars Bergman. **On the Meaning of “C” in CSCW**. ERCIM News Online Edition, Abril de 1995.  
[http://www.ercim.org/publication/Ercim\\_News/enw21](http://www.ercim.org/publication/Ercim_News/enw21)
- [BRI 95] British Telecommunications. **Na Overview of Teleworking**. Study report, 1995. <http://btlabs1.labs.bt.com/projects/telework/doc.htm>
- [BSC 98] **BSCW Project**. FIT – GMD, 1998. <http://orgwis.gmd.de/projects/BSCW>
- [CWC 98] **Eletronic Learning Environment**. Collaborative Work Center Home Page, Northeast Lousiana University. 1998.  
<http://198.79.216.222/ele/collab.html>
- [DIX 96] Alain Dix. **Challenges and Perspectives for Cooperative Work on the Web**. Proceedings of the ERCIM workshop on CSCW and the Web, Sankt Augustin, Alemanha, 1996.
- [DIX 98] Alain Dix, Devina Ramduny, Julie Wilkinson. **Interaction in the Large**. Interacting with Computers, Special Issue on Temporal Aspects of Usability. 1998. <http://www.soc.staffs.ac.uk/~cmtajd/papers/IwCtau98/>
- [GAY 97] Gery Gay, Marc Lentini. **Use of Communication Resources in a Networked Collaborative Design Environment**. CSCW’s International Conference on Work Teams Proceedings, 1997.  
[http://jcmc.huji.ac.il/vol1/issue1/IMG\\_JCMC/ResourceUse.html](http://jcmc.huji.ac.il/vol1/issue1/IMG_JCMC/ResourceUse.html)
- [GRU 94] Jonathan Grudin. **CSCW: History and Focus**. IEEE, 1994.  
<http://www.ics.uci.edu/~grudin/Papers/IEEE94/IEEEComplastsub.html>
- [KAL 99] Sureshkumar Kaliannan. **A Software Platform to Enable Multi-Domain Collaborative Applications**. Tese para o grau “*Master of Science in Computer Science*”, Graduate College of The University of Iowa, 1999.  
<http://www.cs.uiowa.edu/~skaliann/thesis/thesis.html>
- [KEI 97] Gregg Keizer. **Meeting of the Minds (4 Net Conferencing Programs)**. CNET Special Report, 1997.  
<http://www.cnet.com/Content/Reviews/Compare/Netconference>
- [LAA 97] Petteri Laamanen. **Internet: the enabler of more efficient Computer Supported Collaborative Work**. 1997. Publicado na Web:  
<http://www.tcm.hut.fi/Opinnot/Tik-110.551/1997/cscw.html>
- [LPI 97] Jessica Lipnack, Jeffrey Stamps. **Virtual Teams. Reaching Across Space, Time, and Organizations with Technology**. John Wiley & Sons, Inc., EUA, 1997.
- [LIU 97] Kecheng Liu e Alan Dix. **Norm Governed Agents In CSCW**. 1<sup>st</sup> International Workshop on Computational Semiotics, Paris, 1997.  
<http://www.soc.staffs.ac.uk/~cmtajd/papers/paris97>

- [LJU 99] Fredrik Ljungberg e Steinar Kristoffersen. **An Empirical Study of How People Establish Interaction: Implications for CSCW Session Management Models**. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99), Pennsylvania, USA, Maio de 1999.
- [LOT 96] **Lotus Notes R4 Overview White Paper**. Lotus, 1996.  
<http://www.lotus.nl/ntsd96/24a2.htm>
- [MIC 99] **NetMeeting Features Page**. Site da Microsoft, 1999.  
<http://www.microsoft.com/windows/netmeeting/features/default.asp>
- [NOV 99] **GroupWise 5.5 Product Brief**. Novell Products and Programs, 1999.  
<http://www.novell.com/catalog/qr/sne44602.html>
- [SCO 97] Jan Scott. **CSCW**. Material de curso. Charles Sturt Univertisy, 1997.  
<http://farrer.riv.csu.edu.au/~jscott>
- [TAM 97] Randall A. Tamura, *et al.* **Lotus Notes and Domino Server 4.6**. SAMS Publishing, EUA, 1997.

#### Agentes Inteligentes

- [AUE 95] Karl Auer. **Agents**. Swiss Federal Institute of Techonology, 1995.  
<http://www.tip.net.au/~kauer/project/title.htm>
- [CAG 97] Alper K. Caglayan, Colin G. Harrison. **Agent Sourcebook**. Jhon Wiley & Sons, Inc. EUA, 1997.
- [CHO 98] Dimitris N. Chorafas. **Agent Technology Handbook**. McGraw-Hill, EUA, 1998.
- [COE 94] Michael Coen. **SodaBot: A Software Agent Environment and Construction System**. MIT AI Lab Technical Report 1493, 1994.  
<http://www.ai.mit.edu/people/sodabot/sodabot.html>
- [FAR 98] Rafael A. Faraco. **Uma Arquitetura de Agentes para Negociação dentro do Domínio do Comércio Eletrônico**. Dissertação de Mestrado apresentada ao Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 1998.
- [FON 93] Leonard N. Foner. **What's an Agent, Anyway? A Sociological Case Study**. Agents Group, MIT Media Lab., 1993.
- [FRA 96] Stan Franklin, Art Graesser. **Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents**. Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1996.
- [GEN 94] Michael R. Genesereth, Steven P. Ketchpel. **Software Agents**. Communications of the ACM, vol. 37, n. 7, pp. 49-53, Julho de 1994.

- [GOO 93] Richard Goodwin. **Formalizing Properties of Agents**. School of Computer Science, Carnegie Mellon University, Pittsburgh, 1993.  
<http://www.site.uottawa.ca/~babdulla/Library/FormalizingPropertiesOfAgents.ps>
- [HER 96] Björn Hermans. **Intelligent Software Agents on the Internet: an Inventory of Currently Offered Functionality in the Information Society & a Prediction of (near-)Future Developments**. Tese de Doutorado, Tilburg University, Tilburg, Netherlands, 1996.  
[http://www.comedia.com/broadcatch/agent\\_thesis/index.html](http://www.comedia.com/broadcatch/agent_thesis/index.html)
- [JEN 96] Nicholas R Jennings , Michael Wooldridge. **Software Agents**. IEE Review, pp. 17-20, Janeiro de 1996.
- [JEN 98] Nicholas R Jennings , Michael Wooldridge. **Agent Technology: Foundations, Applications and Markets**. Springer-Verlag, 1998.
- [LAS 94] Yezdi Lashkari, Max Metral, Pattie Maes. **Collaborative Interface Agents**, *Proceedings of AAAI '94 Conference*, Seattle, Washington, 1994.  
<http://agents.www.media.mit.edu/groups/agents/publications/aaai-ymp/aaai.html>
- [MAE 94] Pattie Maes. **Agents that Reduce Work and Information Overload**. Communications of the ACM, pp.31-40, 1994.  
<http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html>
- [VAV 98] Fabiane Barreto Vavassori. **Ferramentas e Agentes para um Ambiente de Aprendizagem na Web**. Dissertação de Mestrado apresentada ao Programa de Pós Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, 1998.
- [WOO 95] Michael Wooldridge, Nicholas R Jennings. **Intelligent Agents: Theory and Practice**. Knowledge Engineering Review, 1994. Revisado em Janeiro de 1995.

#### Programação Orientada a Objetos

- [COA 93] Peter Coad e Edward Yourdon. **Projeto Baseado em Objetos**. Editora Campus, Rio de Janeiro, 1993.
- [MEY 88] Bertrand Meyer. **Object-oriented Software Construction**. Prentice Hall International (UK) Ltd. 1988.

## Programação em Java

- [FAR 98] Jim Farley. **Java Distributed Computing**. O'Reilly & Associates, Inc. EUA, 1998.
- [GOS 96] James Gosling, Henry McGilton. **The Java Language Environment – A White Paper**. 1996. <http://java.sun.com/docs/white/langenv>
- [KRA 96] Douglas Kramer. **The Java Platform – A White Paper**. 1996. <http://java.sun.com/docs/white/platform/javaplatformTOC.doc.html>
- [SUNa] **The Java Tutorial**. Sun Microsystems. <http://www.java.sun.com/docs/books/tutorial>
- [SUNb] **The Java Language: An Overview**. Sun Microsystems. <http://java.sun.com/docs/overviews/java/java-overview-1.html>
- [WUT 97] Mark Wutka, *et al.* **Java: Técnicas Profissionais**. Berkeley, São Paulo, 1997.