

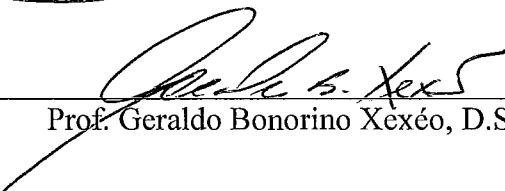
CUMBIA: UMA ARQUITETURA BASEADA EM AGENTES PARA DETECTAR
OPORTUNIDADES DE COLABORAÇÃO


Melfry M Moreno Molina

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE
SISTEMAS E COMPUTAÇÃO.

Aprovada por:


Prof. Jano Moreira de Souza, Ph.D.


Prof. Geraldo Bonorino Xexéo, D.Sc.


Prof.^a. Flavia Santoro, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 2005

MORENO MOLINA, MELFRY M.

CUMBIA: Uma arquitetura baseada em agentes para detectar oportunidades de Colaboração [Rio de Janeiro] 2005

XI, 80 p., 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Dissertação – Universidade Federal do Rio de Janeiro, COPPE

1. Trabalho Cooperativo Suportado por Computador
2. Sistemas Multiagentes
3. Colaboração Oportunística

I. COPPE/UFRJ II. Título (série)

*À memória da minha mãe, meu anjo
onde quer que esteja.*

*Ao meu pai,
quem me ensinou que ser pai é antes de tudo ser amigo.*

*A minha irmã,
que é como a voz da minha consciência.*

*A Claudia, minha esposa,
a luz que ilumina o meu caminho.*

*A Tawny e Emily, minhas filhas
fonte de inspiração e que fazem tudo valer a pena.*

Ao Edinson Daniel, meu sobrinho.

Agradecimentos

São muitas as pessoas que contribuíram de formas diversas e em momentos diversos nestes três longos anos de caminho que culminam com a conclusão desta dissertação. A todas elas meus mais sinceros agradecimentos. Vou aproveitar este espaço para mencionar de maneira especial a algumas delas.

Agradeço a minha Claudia, minha esposa, pela amizade, pelo amor, pelo apoio, pela paciência, e pela confiança.

Agradeço a Tawny e Emily, minhas filhas pelos muitos momentos de convívio perdidos durante a realização do mestrado devido à distância, e que espero retribuir em demasia a partir de agora.

Agradeço a minha mãe, a sua memória, quem sempre foi a primeira a se orgulhar dos meus logros e quem, apesar de ter partido deste mundo físico, permanece o tempo todo comigo dentro do meu coração me dando forças para empreender novos desafios.

Agradeço ao meu pai, o melhor pai do mundo, pela grande amizade, pelas críticas construtivas, pelos conselhos e pelo apoio ao longo da minha vida.

Agradeço a Merly, minha irmã e amiga, quem abriu o caminho desta longa travessia, e quem é sempre fonte de apoio e geradora de reflexões internas.

Agradeço a Adriana minha co-orientadora estrela que com a sua inteligência crítica e a sua dedicação na revisão dos meus textos constituiu-se na minha consciência acadêmica.

Agradeço a Jano meu orientador pela oportunidade de ter vivido uma experiência construtiva e formadora acadêmica, profissional e pessoal. Uma verdadeira experiência de vida.

Agradeço ao meu amigo Ricardo Barros, o meu Comandante, pela amizade desinteressada, pelos momentos de reflexão e pela confiança e apoio durante estes três longos anos. Agradeço a Uthy pela amizade, carinho e apoio. Não posso deixar de mencionar também os meus amigos Elder, Dionísio, Nayibe, Freddy, Guillermo, Dona Ana, Roberto, Margarete, Jonice, Patty e Manuel.

A todos “¡*Muchas gracias!*”.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CUMBIA: UMA ARQUITETURA BASEADA EM AGENTES PARA DETECTAR OPORTUNIDADES DE COLABORAÇÃO

Melfry M Moreno Molina

Setembro / 2005

Orientador: Jano Moreira de Souza

Programa: Engenharia de Sistemas e Computação

Com os recentes avanços nas tecnologias das comunicações e a descentralização das praticas de trabalho, se incrementaram os ambientes de trabalhos computacionais, distribuídos, remotos e automatizados. Na maioria dos casos, as pessoas trabalham em seus terminais de trabalho sem estar conscientes de seus pares. Com a mudança de ambientes físicos para ambientes virtuais, oportunidades de colaboração não são percebidas. Esta dissertação propõe uma arquitetura baseada em agentes para incentivar e dar suporte a colaborações não planejadas entre pessoas. Os agentes constroem perfis de usuário, analisando seus documentos e seu ambiente de trabalho buscando similaridade entre seus interesses e atividades. Analisando e comparando os contextos de trabalho, necessidades e recursos dos usuários, se espera descobrir oportunidades de colaboração que de outra maneira não seriam percebidas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CUMBIA: AN AGENT FRAMEWORK TO SUPPORT OPPORTUNISTIC
COLLABORATION

Melfry M Moreno Molina

September / 2005

Advisor: Jano Moreira de Souza

Department: Computer and Systems Engineering

With the recent advances in communications technologies and decentralization of work practices, there has been an increase in distributed, remote, computerized work environments. In most systems, individuals work from their personal computer terminals, unaware of their peers. With the change from a physical to a virtual environment, opportunities for collaboration often go unnoticed. In this Dissertation, we focus on how to bring unplanned collaboration about. We present an agent framework to encourage and support unplanned cooperation between people. Agents build user profiles through analysis of their documents and work environment and match them according to their interests, activities and opportunities for collaboration. By matching users' work contexts, needs and resources, we expect to uncover opportunities for collaboration that might otherwise go unnoticed.

Índice

Capítulo 1 – Introdução	1
1.1 – Motivação	1
1.2 – Caracterização do problema	1
1.3 – Objetivo da tese	3
1.4 – Enfoque da solução	3
1.5 – Organização da tese	4
Capítulo 2 – Trabalho Cooperativo com Suporte por Computador.....	6
2.1 – Groupware	8
2.2 – Tipos de grupos.....	9
2.2.1 – Equipes	9
2.2.2 – Organizações.....	10
2.2.3 – Grupos de interação social.....	11
2.2.4 – Grupos amorfos	11
2.3 – Percepção (<i>Awareness</i>).....	11
2.4 – Comunicação informal.....	14
2.5 – Privacidade	16
2.5.1 – Privacidade, segurança e anonimato.....	16
2.5.2 – Compartilhamento de informação, identificação e responsabilidade	16
2.5.3 – Controle e reciprocidade.....	17

2.6 – Similaridade e proximidade.....	17
Capítulo 3 – Agentes	19
3.1 – Definição de agente	20
3.2 – Características desejáveis	20
3.3 – Tipologia de agentes	21
3.4 – Comunicação entre agentes	24
3.5 – Arquiteturas de agentes.....	27
3.6 – Sistemas multiagentes.....	28
3.7 – Plataformas de desenvolvimento de agentes	28
3.7.1 – <i>Java Agent DEvelopment Framework (JADE)</i>	31
3.8 – Aplicações dos agentes	34
Capítulo 4 – CUMBIA.....	36
4.1 – Arquitetura	37
4.1.1 – Gestão de conhecimento	38
4.1.1.1 – Processamento em lotes	40
4.1.1.2 – Processamento on-line	41
4.1.2 – Percepção e <i>matchmaking</i>	42
4.1.3 – Colaboração	44
4.1.4 – Interface de usuário.....	44
Capítulo 5 – Protótipo implementado.....	45
5.1 – Agentes implementados.....	46

5.1.1 – Tipo <i>monitor</i>	46
5.1.2 – Tipo <i>profiler</i>	49
5.1.3 – Tipo <i>matchmaker</i>	51
5.1.4 – Interface do usuário	52
5.2 – Mapeando os agentes na arquitetura proposta	53
Capítulo 6 – Estudo de caso.....	55
6.1 – Ambientes acadêmicos	55
6.2 – O uso do protótipo	56
6.2.1 – Cenário.....	56
6.2.2 – Primeiros testes.....	56
6.2.3 – Segunda avaliação	57
6.2.3.1 – Participantes.....	57
6.2.3.2 – Procedimento	57
6.2.3.3 – Resultados esperados	58
Capítulo 7 – Conclusões	59
7.1 – Contribuições.....	60
7.2 – Trabalhos futuros	60
Referências Bibliográficas.....	62
Anexo.....	74

Índice de Figuras

Figura 1 - Matriz tempo/localização.....	6
Figura 2 - Espectro dos grupos (ELLIS e WAINER, 1998).....	9
Figura 3 – Sistema de Teleconferência VideoWindow (FISH, KRAUT et al., 1990)...	12
Figura 4 - Dimensão da formalidade na comunicação (KRAUT, FISH et al., 1990). ...	14
Figura 5 - Tipos de comunicação segundo (KRAUT, FISH et al., 1990).	15
Figura 6 - Tipologia de agentes proposta por (NWANA, 1996).	23
Figura 7 - Taxonomia de agentes proposta por (FRANKLIN e GRAESSER, 1996). ...	24
Figura 8 - Produtos reportados no estudo realizado por Mangina (MANGINA, 2002).	29
Figura 9 - Ciclo de vida dos agentes em JADE (BELLIFEMINE, CAIRE et al., 2004)	32
Figura 10 - Arquitetura padrão proposta pela FIPA (BELLIFEMINE, CAIRE et al., 2004).....	33
Figura 11 - Arquitetura de agentes CUMBIA (MORENO, VIVACQUA et al., 2003).	37
Figura 12 - Esquema da arquitetura.....	45
Figura 13 - Arquivo XML de registro da atividade do usuário gerado pelos agentes do tipo <i>monitor</i>	47
Figura 14 - Esquema do conjunto de agentes.....	52
Figura 15 - Radar.....	53

Índice de Tabelas

Tabela 1 - Esquema dos dados pessoais do perfil dos usuários baseado em (P3P, 2005).	50
Tabela 2 – Relação dos agentes implementados com a arquitetura proposta.....	54

Capítulo 1 – Introdução

1.1 – Motivação

Novas práticas de trabalho têm surgido graças aos recentes avanços em tecnologias de comunicação e ao amplo uso dos computadores nas organizações. É cada vez mais comum encontrar indivíduos trabalhando individualmente no computador e colaborando remotamente com outros. A tendência à descentralização do trabalho é cada vez maior e há uma tendência à formação de equipes de trabalho temporárias para trabalhar em projetos específicos. Com a adoção de ferramentas colaborativas de trabalho pelas organizações, os indivíduos passam a estabelecer colaborações remotas mais frequentemente e a trabalhar cada vez mais em ambientes virtuais (HERTEL, GEISTER et al., 2005).

Projetos multidisciplinares, em particular, requerem que grupos de especialistas de diferentes áreas trabalhem juntos na solução de problemas complexos. Neste caso, várias pessoas contribuem na busca da solução a um problema específico. Em outros casos, várias pessoas trabalham com problemas similares ou mesmo iguais, mas com um enfoque diferente. Em qualquer um desses casos, a oportunidade de discutir diferentes tópicos do projeto no qual se está trabalhando é sempre útil, permitindo a reflexão e exploração de diferentes opções (FISHER, 2001) bem como a geração de soluções inovadoras. Em todos os casos, a oportunidade de colaborar existe e ainda pode ser necessária.

1.2 – Caracterização do problema

Nesses ambientes, algumas oportunidades normais de interação desaparecem, tais como as conversas informais nos corredores e as sugestões espontâneas em encontros ao acaso (não planejados). No computador, a percepção do usuário sobre o ambiente fica seriamente limitada, sendo a ausência de informação visual, auditiva e ambiental a maior dificuldade. Além disso, as interações por computador são menos ricas que as interações face-a-face. Em um ambiente virtual, é muito difícil conhecer e confiar em alguém ou encontrar-se por acaso com alguém conhecido e colaborar com

ele.

Ferramentas de “bate-papo” (ou *messengers*) fornecem meios para que as pessoas estejam ao alcance de conhecidos: permitindo saber se eles estão on-line e facilitando a comunicação com eles quando necessário. A maioria desses ambientes permite a troca de arquivos e mensagens, discussões e co-edição. Alguns são ricos em mídias, incluindo interações com áudio e vídeo além das ferramentas padrões. Não obstante, nesses ambientes ainda é necessário que os usuários determinem a necessidade de comunicação e decidam pelo início da interação. É possível oferecer melhores serviços que levem a colaborações. Muitas oportunidades se perdem pela falta de percepção dos outros e do ambiente. As pessoas desconhecem as habilidades, interesses, disponibilidade ou motivação dos outros pra participar em um projeto.

Gerenciamento de perfis de usuário, competências, interesses e expertise são técnicas utilizadas para auxiliar a detecção de oportunidades de colaboração e a indução ao trabalho cooperativo. Provendo aos usuários informações de percepção, esses teriam oportunidade de melhor aproveitar as habilidades, competências e disponibilidade dos outros. Isso é especialmente importante em ambientes de trabalho não estruturados ou vagamente estruturados, onde as pessoas trabalham em grupos e equipes que são altamente re-configuráveis e não necessariamente preestabelecidas desde o início. Ambientes acadêmicos são um exemplo: nestes, equipes de pesquisadores podem estar envolvidos em diferentes linhas de trabalho e especialistas podem se juntar ao grupo para tratar de diferentes assuntos. Esses últimos poderiam se vincular temporalmente ao grupo, por exemplo, com o objetivo de resolver um problema em particular, ou poderiam aderir permanentemente no projeto.

Além disso, indivíduos frequentemente se conectam nos sistemas de troca de mensagens em “modo invisível”, eliminando a possibilidade de alguém tentar se comunicar com eles. Muitas vezes isso tem a finalidade de diminuir as conversações não desejadas, que por vezes são iniciadas quando o usuário se encontra em “modo disponível”. Isto demonstra que as mensagens podem ser elementos perturbadores no ambiente de trabalho: em grande parte, a informação sobre as atividades dos outros é irrelevante no contexto corrente de trabalho e só atrapalha a atividade em curso (SOHLENKAMP, 1999b). Torna-se então necessário um controle apurado do fluxo de informação para minimizar o nível de interrupção do usuário. Apresentação de

informação oportuna (Just in Time Information Delivery) é o estudo de como entregar a informação no momento preciso, para a pessoa precisa e da maneira adequada, ou seja, que não interrompa o trabalho individual (RHODES e MAES, 2000). As perguntas “que, quando, como e quem” (os famosos W’s do inglês “What, When, hoW and Who”) têm ganhado importância junto com o volume de informação disponível.

1.3 – Objetivo da tese

O objetivo deste trabalho é apresentar uma arquitetura *peer to peer* baseada em agentes para dar suporte à percepção (*awareness*) e à descoberta de oportunidades potenciais de colaboração. A idéia principal é incentivar colaboração espontânea ou não-planejada entre pessoas com perfis próximos. Agentes constroem perfis de usuários através da análise dos documentos e do ambiente de trabalho destes, e buscam proximidade entre eles com respeito aos seus interesses e às suas atividades. Através da busca de proximidade entre os contextos de trabalho, necessidades e recursos dos usuários, se espera descobrir oportunidades de colaboração que de outra maneira não seriam percebidas. Compartilhamento de recursos e a informação sobre esses recursos são facilitados com o intuito de estimular a colaboração entre os usuários.

1.4 – Enfoque da solução

Para enfrentar esses problemas, este trabalho utiliza técnicas de gerenciamento de perfis e matchmaking, através do uso de agentes inteligentes, que capturam e filtram informação, e fazem inferências sobre o contexto de trabalho dos usuários, assim como determinam o momento e os destinatários dessa informação.

Pesquisas anteriores ressaltam a importância da proximidade em esforços colaborativos (KRAUT, FISH et al., 1990), (OLSON e OLSON, 2000), (BRADNER, 2002). Em uma pesquisa de colaboração científica a proximidade foi o fator de maior importância na determinação da colaboração, superando a similaridade de interesses (KRAUT, EGIDO et al., 1990). Ou seja, se estabeleceram colaborações entre pessoas que estavam próximas mais frequentemente do que entre aquelas que tinham interesses similares, mas tinham que lidar com o fator da distancia física para colaborar.

Matsuura introduz o conceito de proximidade virtual, definido como situações nas quais os usuários acessam os mesmos dados ou invocam a mesma

aplicação em um ambiente virtual (MATSUURA, FUJINO et al., 1993). Este trabalho adota um enfoque similar de “proximidade”, usando o contexto de trabalho individual atual como base para busca de outras pessoas com as quais seria interessante colaborar. A interface de usuário implementada usa a metáfora do radar para representar a proximidade conceitual do usuário com os contatos escolhidos por este.

O fornecimento de informação sobre o usuário é invariavelmente associado à violação de privacidade, isto é, a privacidade do usuário pode estar sendo violada pelo compartilhamento de informações com outras pessoas sobre as suas atividades. De maneira geral, cada elemento de informação compartilhado com outros é uma violação potencial à privacidade. Para lidar com essa situação, o sistema deixa a critério do usuário qual informação compartilhar e em que quantidade ou nível de detalhe.

1.5 – Organização da tese

Neste trabalho é proposta uma arquitetura baseada em agentes para dar suporte à percepção e à descoberta de oportunidades potenciais de colaboração. Para uma melhor compreensão do trabalho, faz-se necessária uma revisão da literatura, apresentada nos capítulos 2 e 3.

No **capítulo 2** são apresentados conceitos de CSCW e *groupware* focando principalmente nos conceitos de percepção e comunicação informal, assim como em uma classificação dos diferentes tipos de grupos. São apresentados também trabalhos relacionados com percepção e comunicação informal desenvolvidos por diferentes autores.

O **capítulo 3** trata de conceitos relacionados com agentes partindo da sua definição, características desejáveis, tipologias diversas, formas de comunicação, arquiteturas, e finalizando com sistemas multiagentes. O capítulo 3 é fechado com uma breve discussão sobre o uso dos agentes na área de *groupware*.

No **capítulo 4** é apresentada CUMBIA, a arquitetura proposta, descrevendo cada uma das quatro equipes de agentes que a compõem: *Interface de Usuário*, que apresenta a informação ao usuário; *Colaboração*, que permite o estabelecimento fácil e rápido do contato quando há uma possibilidade de colaboração; *Awareness e matchmaking*, que faz a busca por outros usuários com quem pode ser interessante estabelecer contato; e

Gestão de Conhecimento, que administra os perfis e armazena informação sobre os documentos manipulados, as buscas realizadas, as colaborações e as pesquisas em andamento.

Para validação do trabalho, foi criado um protótipo, o qual é apresentado no **capítulo 5**, seguido de uma conclusão sobre o trabalho e direções futuras para pesquisa no **capítulo 6**.

Capítulo 2 – Trabalho Cooperativo com Suporte por Computador

Trabalho Cooperativo com Suporte por Computador (ou CSCW, do inglês *Computer-Supported Collaborative/Cooperative Work*) emergiu como uma área de pesquisa multidisciplinar na metade da década de 80 (BANNON e SCHMIDT, 1989). A pesquisa se concentra em formas efetivas de cooperação entre pessoas ou grupos que têm de realizar uma tarefa comum e no projeto, implementação e suporte de sistemas computacionais para essas formas de cooperação. Muito freqüentemente os sistemas de CSCW são classificados segundo a matriz tempo/localização descrita por (GRUDIN, 1994) e mostrada na Figura 1.

	MESMO LOCAL	LOCAL DIFERENTE
MESMO MOMENTO	INTERAÇÃO FACE-A-FACE	INTERAÇÃO SÍNCRONA DISTRIBUÍDA
MOMENTO DIFERENTE	INTERAÇÃO ASSÍNCRONA	INTERAÇÃO ASSÍNCRONA DISTRIBUÍDA

Figura 1 - Matriz tempo/localização

Liu e Conradi redefinem e reorganizam a classificação de tipos de trabalho cooperativo levando em conta as diferentes classes e a complexidade dos processos envolvidos que precisam de suporte. Eles definem quatro tipos de trabalho (LIU e CONRADI, 1998):

- Trabalho cooperativo *Ad Hoc*: o processo não segue um modelo formalmente definido, e sua arquitetura consiste de um ambiente que fornece ferramentas de comunicação com características básicas de percepção

(*awareness*). Alguns exemplos desse tipo de trabalho cooperativo são brainstorming, aprendizado cooperativo, encontros informais e trabalhos em design.

- *Workflow* predefinido ou rigoroso: sistemas como os de automação de escritórios representados por fluxos de documentos e processos. Tem duas subcategorias: uma pessoa só trabalhando em documentos complexos ou várias pessoas trabalhando num documento. Em ambos os casos, o processo é formal, mas simples. Na arquitetura do processo, encontram-se espaços de trabalho (*workspaces*), repositório de documentos e ferramentas de planejamento, coordenação e execução do processo. Exemplos de tais sistemas são o Lotus Notes (ORLIKOWSKI, 1992), o Active Mail (GOLDBERG, SAFRAN et al., 1992) e o MAFIA (LUTZ, RETZOW et al., 1990)
- *Workflow* coordenado: consiste de atividades de registro, processamento, check-out e execução das tarefas. Existe comunicação freqüente (síncrona e assíncrona) entre os participantes, mas o compartilhamento de dados é limitado entre eles, dado que é controlado principalmente pelo coordenador. O processo inclui um exemplo desse tipo de trabalho cooperativo que é a manutenção centralizada de *software*. Alguns protótipos interessantes desse tipo de sistema são o EPOS (CONRADI, JACCHERI et al.), o MARVEL (BEN-SHAUL e KAISER, 1995) e o APEL (DEMI, ESTUBLIER et al., 1998).
- *Workflow* cooperativo: nestes casos, o espaço de trabalho compartilhado e o planejamento da cooperação são os principais fatores do ponto de vista do processo. Comunicação (síncrona e assíncrona) é fluida entre todos os participantes e o coordenador, e o compartilhamento de dados é significativo. Um exemplo interessante de sistemas que dão suporte a esses tipos de organizações e processos distribuídos é o OZ (BEN-SHAUL e KAISER, 1995).

2.1 – Groupware

Os sistemas que dão suporte computacional a grupos de pessoas envolvidas em uma tarefa comum, assim como as interfaces para ambientes compartilhados, são chamados de *groupware* (ELLIS, GIBBS et al., 1991).

Há uma ampla variedade de produtos de *groupware* se concentram em aspectos específicos do trabalho cooperativo. Produtos de *groupware* incluem, entre outros, “quadros brancos” compartilhados e sistemas de comunicação estruturados, de agendas de grupo, de videoconferências, de co-autoria, e de gestão de *workflow* (DE FARIAS, PIRES et al., 1999).

Atualmente, cada vez mais pessoas precisam acessar remotamente informações da empresa e precisam colaborar com outros à distância. A *Web* tem assumido um papel central na forma como as pessoas compartilham informação local e remotamente, pois os browsers (ou navegadores) encontram-se disponíveis em qualquer lugar e a *Web* integra diferentes serviços em uma só interface de usuário comum, facilmente acessível e independente da plataforma. Essa é a razão pela qual a *Web* tem sido adotada como um dos maiores meios de suporte à colaboração entre pessoas. Mesmo assim, os mecanismos básicos oferecidos pela *Web* não são suficientes para dar suporte à colaboração interativa. As necessidades de comunicação que estimularam o desenvolvimento da *Web* foram a consulta estruturada de documentos e não o suporte à discussão interativa entre os membros de uma equipe virtual (BERGENTI, GARIJO et al., 2002).

A migração das tecnologias *Web* às redes móveis permite levar serviços e facilidades consolidados na *Web* aos usuários móveis. A ampla disponibilidade de banda das infra-estruturas móveis permite às pessoas compartilhar informação através de hardware heterogêneo, isto é, laptops e PDAs, usando sistemas operacionais e protocolos de comunicação diferentes (BERGENTI, GARIJO et al., 2002).

Ellis e Wainer apresentam um modelo de *groupware* onde as funcionalidades centrais são classificadas em três grupos ou aspectos diferentes. o “guardião”, aspecto relacionado com as operações sobre os dados compartilhados, o “coordenador”, aspecto relacionado à seqüência temporal das atividades executadas pelos usuários do

groupware, e o “comunicador”, aspecto relacionado com a comunicação irrestrita entre os usuários do *groupware*. Alguns sistemas de *groupware* fornecem funcionalidades pertencentes a apenas um desses aspectos (ELLIS e WAINER, 1994).

2.2 – Tipos de grupos

Ellis e Wainer classificam os grupos em quatro classes, baseados no estilo de trabalho, nas perspectivas e nas atitudes das pessoas envolvidas nesses grupos. As classes não são disjuntas, elas podem ser vistas como extremos de um ‘contínuo’, que une esses quatro extremos como mostrado na Figura 2. As quatro classes de grupos são equipes, organizações, grupos de interação social e grupos amorfos. Segue uma breve descrição extraída do trabalho desses autores (ELLIS e WAINER, 1998).

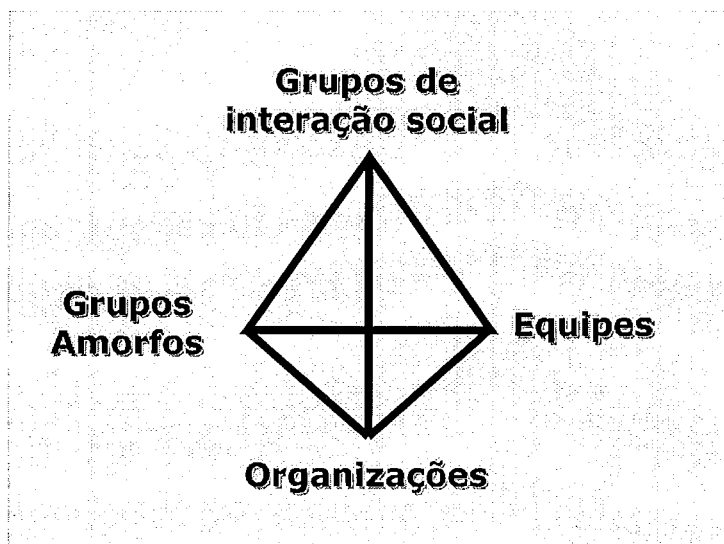


Figura 2 - Espectro dos grupos (ELLIS e WAINER, 1998)

2.2.1 – Equipes

Equipes são grupos coesos de pessoas que se conhecem mutuamente e trabalham em uma tarefa específica. Normalmente, equipes possuem poucos membros, que conhecem e confiam uns nos outros e têm um objetivo comum (terminar a tarefa que estão executando). É também normal, em equipes, a ausência de hierarquias ou estruturas de poder. Exemplos de equipes são: dois autores escrevendo um artigo, uma equipe de futebol jogando um jogo, engenheiros projetando um sistema de controle para

uma fábrica. É importante frisar que essa definição da equipe é relacionada à tarefa, e não persiste depois que esta é terminada. Logo, não podemos afirmar que um grupo de pessoas é uma boa equipe porque fazem muitas coisas juntas. Em vez disso, deveríamos dizer que quando executam uma tarefa determinada, esse grupo de pessoas trabalha como uma equipe.

Tipicamente, as equipes trabalham em reuniões ou encontros: os membros da equipe estão juntos, no mesmo lugar e tempo, para trabalhar na tarefa. A reunião não segue uma seqüência predefinida de passos, ela é dinamicamente planejada de acordo com as necessidades das pessoas e da tarefa mesma. A divisão do trabalho também não é predefinida, ela evolui na medida em que a tarefa evolui. O trabalho de cada membro da equipe é dependente do trabalho dos outros membros: cada um deles deve estar ciente do que os outros estão fazendo.

2.2.2 – Organizações

Organizações são grupos de pessoas regidas por estruturas e modelos de poder. Tipicamente, as organizações possuem um grande número de pessoas, que podem conhecer ou não as outras; e que podem compartilhar ou não um mesmo objetivo. A organização como um todo está envolvida em um grande número de atividades e tem muitos objetivos, e cada membro da organização pode estar envolvido em muitas atividades e ter seus próprios objetivos. Cada membro de uma organização pode ter diferentes papéis, e esses papéis definem um conjunto de direitos para executar ações, acessar informações, etc. Informação sobre a identidade dos membros da organização, os papéis que eles desempenham, e toda a estrutura formal dentro da organização estão normalmente disponíveis como modelos.

O trabalho tipicamente é de escritório, isto é, pessoas trabalhando individualmente em uma ou mais atividades que são parte de um grande procedimento predefinido. Portanto, a divisão do trabalho e a seqüência das atividades são também predefinidas. O trabalho é comumente dividido em atividades executáveis individualmente, ordenadas de maneira a ser possível garantir que toda a informação necessária para executar uma atividade estará disponível quando ela estiver sendo executada.

2.2.3 – Grupos de interação social

Um grupo de interação social é um grupo de pessoas que estão socializando. Normalmente, esse tipo de grupo é pequeno e os membros se conhecem, mas não tem um objetivo comum (além de socializar). A maneira típica de interagir é a conversação, que é uma atividade de comunicação não-estruturada e altamente interativa.

2.2.4 – Grupos amorfos

Grupos amorfos são grupos formados por um grande número de pessoas que não se conhecem, não têm objetivos comuns, não estão envolvidas em estruturas, não confiam umas nas outras etc. A única diferença entre um grupo amorfo e um conjunto matemático de pessoas é que existe um espaço de informação associado ao grupo amorfo. Para pertencer a um grupo amorfo um membro tem que fornecer um meio de tanto ele quanto a informação pública sobre seus interesses serem acessados pelos outros membros do grupo. Essa informação é que dá origem ao espaço de informação do grupo amorfo.

O principal modo de trabalho, nos grupos amorfos é a busca individual no espaço de informação, procurando a pessoa que se está precisando.

A Internet é o melhor exemplo de um grupo amorfo com um espaço de informação em formato digital, que pode ser explorado por programas (nesse caso, agentes).

2.3 – Percepção (*Awareness*)

O conceito de “*Awareness*” (ou percepção) tem recebido muita atenção dos pesquisadores da área de CSCW. Estes têm reconhecido à importância de estar cientes dos seus pares ou colaboradores dentro do ambiente de trabalho. Uma das primeiras definições encontradas na literatura diz que *awareness* é “o conhecimento das atividades dos outros que serve para dar contexto a nossa própria atividade” (DOURISH e BELLOTI, 1992). (SOHLENKAMP, 1999a) destaca que “*awareness* é o estado mental de um usuário... que envolve as atividades dos outros e fornece um contexto para as próprias... enquanto os mecanismos de *awareness* são técnicas empregadas por um sistema pra alcançar esse estado mental”.

São vários os aspectos a serem considerados na definição de *awareness*: o estado mental do usuário, o estado atual do computador e do sistema, a atividade e a presença de outros na *Web*, o contexto para a própria atividade e a possibilidade de re-inspecionar atividades passadas e de recriar o contexto apropriado (KANERVA, KOSKINEN et al., 2000).

A preocupação inicial foi fornecer suporte de vídeo e de áudio para apoiar a cooperação, mas outras ferramentas e métodos vêm surgindo desde então. Diversos trabalhos lidam com interfaces de vídeo para apoiar percepção pessoal e interações informais.

Cruiser (FISH, 1989; ROOT, 1988), por exemplo, é um ambiente virtual que usa canais de áudio e vídeo para dar suporte ao processo dinâmico de interação social informal. Foi projetado com o propósito de dar suporte à comunicação a distância entre colegas. Cruiser fornece mecanismos inovadores para estabelecer conexões entre usuários, procurando fomentar comunicação informal, freqüente (habitual) e não-planejada entre os membros de uma comunidade distribuída. A interface fornece uma espécie de “corredor virtual” onde o usuário pode descansar (*take a break*) quando desejar. A interface tenta simular um passeio pelo corredor. O usuário pode visualizar as outras pessoas no ambiente e pode tomar a decisão de conversar (ou não) com qualquer um no corredor.

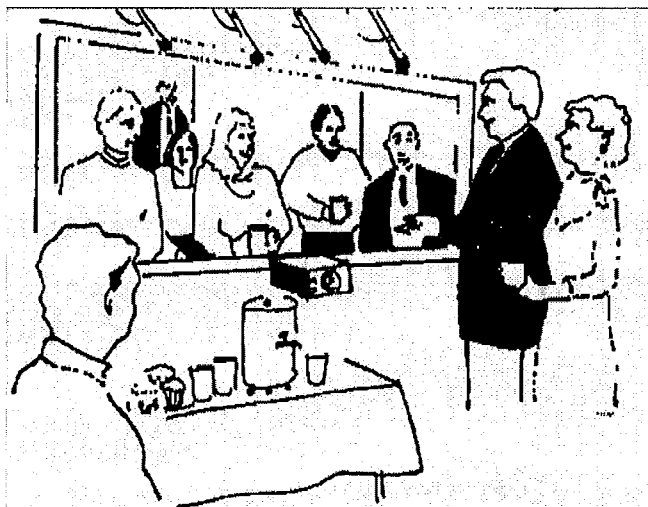


Figura 3 – Sistema de Teleconferência VideoWindow (FISH, KRAUT et al., 1990).

VideoWindow (BELLCORE, 1989; FISH, KRAUT et al., 1990) é um sistema de teleconferência que conecta duas “salas de café” localizadas em escritórios distantes com canais *full duplex* áudio e vídeo. As imagens de vídeo são projetadas em uma tela de três pés de altura e seis pés de largura (aproximadamente o dobro de uma tela normal de TV). Isso permite ver quase toda a sala de café e as pessoas aparecendo na tela próximas ao tamanho normal, como mostrado na Figura 3. O VideoWindow oferece a sensação de presença em um espaço compartilhado a um custo relativamente baixo. Para usar o VideoWindow, basta entrar na sala de café e dizer “oi” para quem estiver do outro lado. O sistema é utilizado para estudar as interações (mediante a tela) entre pessoas fisicamente distantes que estejam descansando nas salas de café.

Portholes (DOURISH e BLY, 1992) e Polyscope (BOURNING e TRAVERS, 1991) são espaços que dão suporte à percepção compartilhada com o objetivo de facilitar interações informais. O sistema fornece imagens de todos os usuários nos escritórios, permitindo saber se os demais estão ocupados ou não, e ver, de modo geral, o que os colegas estão fazendo.

Alguns trabalhos incluem motivação e incentivos no suporte à colaboração. Pinheiro propõe uma arquitetura que fornece percepção sobre eventos passados (isto é, informação sobre iterações passadas, resultado de trabalhos anteriores) para melhorar o suporte à colaboração atual (PINHEIRO, LIMA et al., 2002). (HOFFMAN e HERRMANN, 2004) propõem sistemas que permitem aos indivíduos vislumbrarem benefícios potenciais da colaboração com o intuito de motivá-la

Outros trabalhos lidam com percepção baseada nos documentos manipulados, ou nas tarefas que estão sendo realizadas, com a finalidade de fornecer informação aos usuários sobre quem está trabalhando com o mesmo documento ou está realizando tarefas similares no mesmo momento (ISAACS, TANG et al., 1996; MATSUURA, FUJINO et al., 1993; MORAN, DECOUCHANT et al., 2002; MORAN, FAVELA et al., 2002). Piazza (ISAACS, TANG et al., 1996), por exemplo, fornece informação sobre quem está trabalhando em uma tarefa similar no computador, descobrindo assim uma oportunidade de interação ou de cooperação. A plataforma PIÑAS (MORAN, DECOUCHANT et al., 2002) fornece informação sobre os espaços de trabalho potenciais e atuais, e serviços orientados ao suporte de edição colaborativa na *Web*. Esses trabalhos tentam claramente buscar coincidências entre indivíduos no momento

em que compartilham um contexto de trabalho. Trabalhos mais recentes concentram-se em ambientes de computação móvel, onde o problema central da pesquisa de percepção é a localização dos usuários (ALDUNATE, NUSSBAUM et al., 2002; ESBORJÖRNSSON e ÖSTERGREN, 2002; KORTUEN, GELLERSEN et al., 2002).

A forma mais básica de percepção é a informação sobre presença fornecida pelos sistemas de troca de mensagens como Yahoo! Messenger, MSN Messenger, AOL Instant Messenger etc. Nesses sistemas, os usuários criam uma lista de contatos que o sistema monitora. O sistema apresenta a disponibilidade e o estado de cada contato assim como a do próprio usuário. Uma ferramenta muito mais especializada é o GROOVE, que introduz o conceito de espaço compartilhado para incrementar o escopo da percepção pessoal ou, “*personal awareness*” (GROOVE, 2005). No espaço compartilhado do GROOVE, os usuários podem ver o que os outros estão fazendo e quais objetos do espaço estão manipulando.

2.4 – Comunicação informal

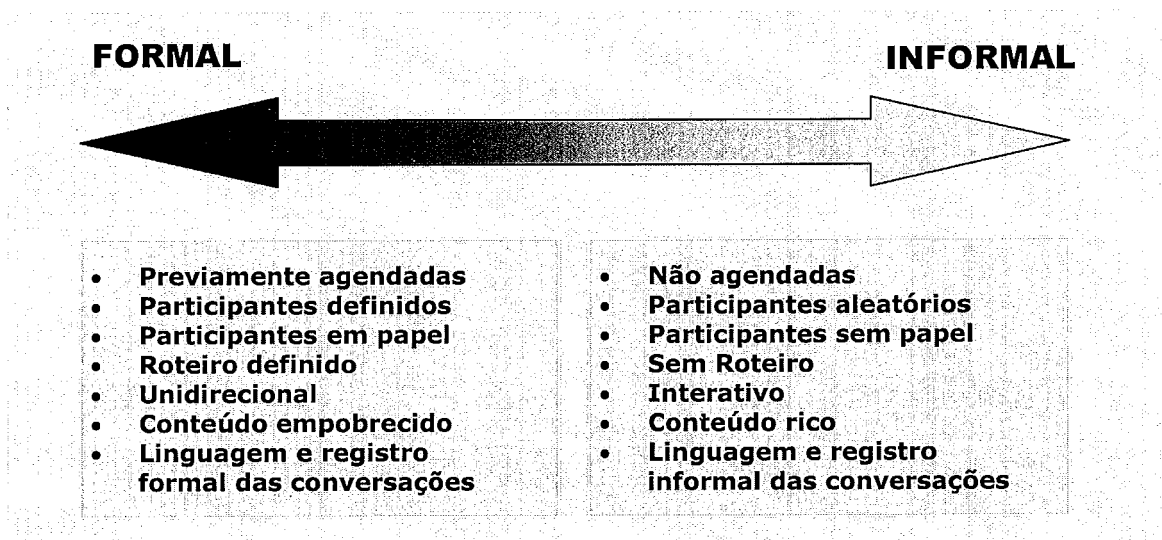


Figura 4 - Dimensão da formalidade na comunicação (KRAUT, FISH et al., 1990).

A maior parte do trabalho realizado nas organizações requer algum grau de comunicação e cooperação ativa com outros, tanto no trabalho rotineiro quanto no criativo (por exemplo, pesquisa científica ou desenvolvimento de engenharia). Os membros de um grupo precisam comunicar-se com os outros membros para assim cumprir suas funções produtiva e social, da mesma forma que os grupos também se comunicam entre si. A comunicação usada é tanto formal quanto informal. O grau de

formalidade da comunicação tem a ver com determinadas características: se foi ou não agendada, se os participantes foram previamente avisados, rol dos participantes e roteiro da reunião, entre outras (apresentadas na Figura 4).

O termo “informal” neste trabalho refere-se à forma em que a interação é iniciada e não necessariamente ao tom da interação; uma interação agendada poderia ser conduzida de maneira informal. (ISAACS, WALENDOWSKI et al., 2002) consideram que o termo “*impromptu*” caracteriza melhor a natureza desse tipo de interação, mesmo assim, eles preferem usar “informal”, dando continuidade à definição usada em trabalhos anteriores (KRAUT, FISH et al., 1990).

(KRAUT, FISH et al., 1990) apresentam uma classificação dos diferentes tipos de interações que acontecem no local de trabalho:

Agendadas – São conversações agendadas previamente por ambas as partes.

Intencionais – Quem inicia a conversação estabelece previamente a intenção de falar com a outra parte.

Oportunisticas – Quem inicia a conversação tinha planejado falar com a outra parte, mas aproveita um encontro para fazê-lo.

Espontâneas – Quem inicia a conversação não tinha planejado falar com a outra parte.

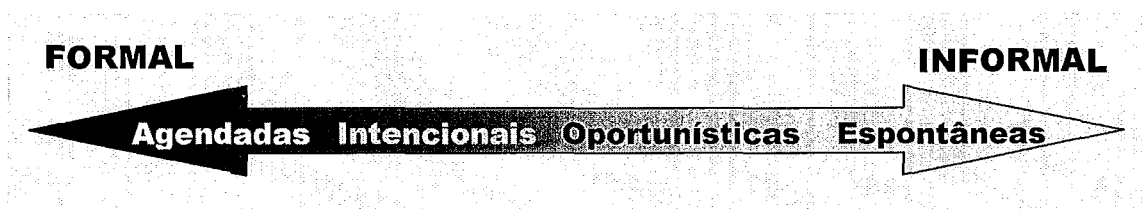


Figura 5 - Tipos de comunicação segundo (KRAUT, FISH et al., 1990).

(KRAUT, FISH et al., 1990) também afirmam que a maioria das conversações são naturalmente informais, freqüentemente breves e usualmente baseadas em discussões prévias. As conversações ocorrem porque uma pessoa está por perto no mesmo momento em que outra quer perguntar ou fornecer (compartilhar) alguma informação. Estudos mostram que as comunicações informais têm um papel muito

importante, ajudando os trabalhadores a aprender, entender, adaptar e aplicar processos e procedimentos formais (ISAACS, WHITTAKER et al., 1997). Poucos sistemas têm focado o suporte das interações oportunísticas e espontâneas.

Segundo (ESBORJÖRNSSON e ÖSTERGREN, 2002), interações espontâneas são ações que ocorrem quando participantes humanos (e/ou computacionais) coincidem ao mesmo tempo em um local e interagem para satisfazer necessidades imediatas. Um ponto de vista similar é adotado em (ALDUNATE, NUSSBAUM et al., 2002), onde a co-localização é essencial para a colaboração espontânea. Ambos os estudos lidam com ambientes de computação móvel, mas fornecem uma idéia útil para a implementação de ambientes de trabalho virtual, pois identificam fatores importantes para o estabelecimento de interações informais.

2.5 – Privacidade

2.5.1 – Privacidade, segurança e anonimato.

Qualquer ferramenta de *groupware* precisa compartilhar algum tipo de informação, mas existe a preocupação que essa informação permaneça privada e que um dado crítico esteja protegido contra intentos agressivos para obtê-lo. Em muitas ocasiões, os usuários escolhem ser anônimos ou decidem usar apelidos. O anonimato pode ser crucial para fomentar participações francas e pode ser muito útil fornecendo proteção contra agressões (FARKAS, ZIEGLER et al., 2002; SHÜMMER, 2004).

2.5.2 – Compartilhamento de informação, identificação e responsabilidade

Por outro lado, existe a contínua pressão de compartilhar mais informação. Quanto mais informação é compartilhada, maior afinidade pode ser alcançada. Compartilhar informação sobre os usuários de um sistema possibilita este fornecer melhor personalização e maior satisfação de interesses. Embora o anonimato sirva para proteger os usuários, existem razões para exigir a identificação destes para fins de responsabilização, em especial quando a segurança e o risco de comportamento abusivo devem ser controlados (FARKAS, ZIEGLER et al., 2002).

2.5.3 – Controle e reciprocidade

Para resolver essas necessidades conflituosas, é importante dar ao usuário tanto controle quanto seja possível sobre o que deve ser compartilhado e o que deve permanecer privado. Permitir aos usuários decidir sobre a quantidade de informação a ser compartilhada poderia ser usado para determinar a informação que ele pode acessar. Um exemplo de políticas de privacidade é o princípio de reciprocidade: se um usuário deseja informação sobre outro, ele deve fornecer a mesma informação sobre si mesmo. Reciprocidade não é sempre uma política correta, mas serve como um ponto de partida (SHÜMMER, 2004).

2.6 – Similaridade e proximidade

Informações relacionadas a conhecimento, habilidades físicas e cognitivas, distância e características psico-sociais como confiança e atitudes, são importantes para o estabelecimento de uma colaboração bem sucedida. Segundo (ALDUNATE, NUSSBAUM et al., 2002), similaridade em preferências por atividades, valores básicos, *hobbies*, cultura, passado comum e confiança na outra pessoa são as mais importantes garantias de contatos bem sucedidos. As pessoas têm modelos mentais de si mesmos e dos outros -- quanto mais próximos esses modelos, maior a probabilidade de se ter interações bem sucedidas.

Pesquisas anteriores ressaltam a importância da proximidade em esforços colaborativos (KRAUT, FISH et al., 1990), (OLSON e OLSON, 2000), (BRADNER, 2002). Em uma pesquisa de colaboração científica, a proximidade foi o fator de maior importância na determinação da colaboração, superando a similaridade de interesses (KRAUT, EGIDO et al., 1990). Ou seja, se estabeleceram colaborações entre pessoas que estavam próximas mais freqüentemente do que entre aquelas que tinham, não obstante ter interesses similares, mas tinham que lidar com o fator da distancia física para colaborar.

Matsuura introduz o conceito de proximidade virtual, definido como aquelas situações nas quais os usuários acessam os mesmos dados ou invocam a mesma aplicação em um ambiente virtual(MATSUURA, FUJINO et al., 1993). Este trabalho usa um enfoque similar, usando o contexto de trabalho individual atual como base na

busca de outras pessoas com as quais seria interessante colaborar e para técnicas de gerenciamento de perfis e *matchmaking*

Capítulo 3 – Agentes

'Agent-based computing (ABC) is likely to be the next significant breakthrough in software development.' (SARGENT, 1992)

A inteligência artificial distribuída (IAD) (AVOURIS, 1992; BOND e GASSER, 1988; LABROU, FININ et al., 1999) define-se como uma subárea da inteligência artificial (IA) que trata dos comportamentos inteligentes coletivos que são produto da cooperação de diversas entidades denominadas agentes. Inicialmente, a IAD (BOND e GASSER, 1988) centrava-se em temas de concorrência da IA, o que incluía também a inteligência artificial paralela, que trata do paralelismo nas arquiteturas de *software*, linguagens e algoritmos para melhorar a eficiência dos sistemas inteligentes. Hoje em dia (BOND e GASSER, 1992), a IAD está mais ligada ao conceito de agentes ou entidades colaboradoras.

Embora seja uma disciplina nova, podemos distinguir três períodos cronológicos bem diferenciados:

- A IAD “clássica” (AVOURIS, 1992; BOND e GASSER, 1988; MOULIN e CHAIB-DRAA, 1996) centra-se no estudo da conduta coletiva, em oposição à IA, que estuda a conduta individual.
- A IAD “autônoma” centra-se no estudo dos agentes individuais situados em um mundo social (CASTELFRANCHI e CONTE, 1996). Modifica-se a visão inicial da IAD em que a “perspectiva social” significa que a sociedade prima sobre os indivíduos, os quais se explicam a partir da sua função na sociedade, e centra-se no estudo de agentes autônomos em um mundo multi-agente.
- A IAD “comercial” (HEDBERG, 1996; NWANA, 1996) centra-se na aplicação da IAD clássica e autônoma, desenvolvendo agentes (denominados genericamente *agentes de software*) com características muito diferenciadas (agentes móveis, pessoais etc.) que estão sendo explorados comercialmente.

3.1 – Definição de agente

No Mini Aurélio Século XXI (FERREIRA, 2000) encontramos a seguinte definição: “*Agente*. 1. Que opera, agencia, age. 2. Quem trata de negócios por conta alheia. 3. Aquilo que produz, ou é capaz de produzir, determinado efeito”.

A questão sobre o que é um agente ainda está em debate, havendo o risco de qualquer programa ser denominado agente. Podem-se distinguir duas definições extremas de agentes (WOOLDRIDGE e JENNINGS, 1995b): Uma *definição fraca* consiste em definir um agente como uma entidade capaz de intercambiar mensagens utilizando uma linguagem de comunicação de agentes (GENESERETH, SINGH et al., 1994). Essa definição é a mais utilizada na engenharia de *software* baseada em agentes, que tem como objetivo conseguir a interoperabilidade entre aplicações no nível semântico utilizando a tecnologia emergente de agentes. Uma *definição mais forte*, ou restritiva, é a enunciada por (SHOHAM, 1993) na sua proposta de Programação Orientada a Agentes (AOP), onde um agente é definido como uma entidade cujo estado é visto como um conjunto de componentes mentais, tais como crenças, capacidades, eleições e acordos. Os agentes são considerados como sistemas intencionais (WOOLDRIDGE e JENNINGS, 1995b), ou seja, sistemas com comportamento previsível atribuindo crenças, desejos e um comportamento racional; são entidades que percebem o seu ambiente mediante sensores e atuam sobre ele (RUSSELL e NORVIG, 1995).

Técnicas orientadas a agentes estão sendo cada vez mais usadas em uma grande variedade de aplicações de telecomunicações, comerciais e industriais na medida em que os projetistas e desenvolvedores percebem seu potencial (JENNINGS, 2001). Os agentes são principalmente utilizados na construção de sistemas complexos ou *peer to peer*, por serem “leves”, poderem trabalhar em paralelo e permitirem uma fácil reconfiguração do sistema.

3.2 – Características desejáveis

Visto que a definição de agente tem sido tão controvertida quanto a definição de IA, se tem optado pela definição de um conjunto de propriedades que caracterizam os agentes. (note que um agente não tem que possuir todas essas propriedades). (SHEN,

NORRIE et al., 2001) apresentam uma lista dessas características desejáveis de um agente inteligente, embora eles refiram-se à utilização em sistemas de engenharia concorrente e manufatura, são aplicáveis a sistemas mais gerais . Características desejáveis de um agente são:

- *Centrado na rede*: distribuído e auto-organizado;
- *Comunicativo*: capaz de interagir com outros agentes;
- *Semi-autônomo*: humanos ou outros agentes podem controlar o nível de autonomia do agente;
- *Reativo*: capaz de reagir oportuna e apropriadamente a eventos e mudanças no seu ambiente;
- *Deliberativo*: com capacidade suficiente de deliberação para executar as tarefas de uma maneira orientada às metas;
- *Colaborativo*: capaz de interagir com humanos e outros agentes;
- *Pro-ativo*: Capaz de tomar iniciativa, não só reagir aos eventos, capaz de gerar metas e agir racionalmente para alcançá-las;
- *Preditivo*: capaz de fazer previsões sobre o futuro, especialmente sobre o efeito das suas ações;
- *Adaptativo*: capaz de se adaptar às mudanças nas necessidades dos usuários e das tarefas no ambiente;
- *Persistente*: capaz de permanecer por longos períodos de tempo inativo;
- *Móvel*: capaz de viajar pela rede se necessário.

3.3 – Tipologia de agentes

A grande quantidade de atributos discutidos anteriormente leva à conclusão de que seria muito difícil programar um agente que incorporasse todos eles. As características de um agente são dependentes do tipo de aplicação a que ele se propõe. A

escolha de quais atributos devem estar presentes em um agente depende da funcionalidade que o projetista pretende dar ao seu agente.

A análise dos atributos que estão presentes nos agentes tem sido utilizada pelos pesquisadores para organizar os agentes em tipologias. Uma tipologia é uma classificação por tipos de agentes que possuem atributos em comum.

(NWANA, 1996) propõe uma tipologia de agentes que identifica diferentes dimensões de classificação. Ele acredita que agentes podem ser classificados de acordo com:

Sua *mobilidade*: estático ou móvel. No caso de ser móvel, o agente pode ainda estar residente na máquina do cliente ou temporariamente no servidor.

A presença ou não de um modelo de *raciocínio simbólico*, ou seja, um agente pode ser deliberativo ou reativo.

A presença dos atributos primários: *autonomia*, *cooperação* e *aprendizado*. Combinando essas três características, quatro tipos de agentes podem ser derivados: agentes colaborativos, agentes colaborativos com capacidade de aprendizado, agentes de interface e agentes verdadeiramente inteligentes. É importante notar que os limites dessa classificação não devem ser interpretados como linhas bem definidas. Na verdade, o fato de agentes cooperativos terem mais ênfase em cooperação e autonomia que agentes com capacidade de aprendizado não exclui a possibilidade de aqueles desenvolverem características de aprendizado.

A *função* principal assumida pelo agente, como os agentes de informação ou de Internet.

Suas características *híbridas* que combinam duas ou mais filosofias diferentes em um mesmo agente.

A presença dos atributos secundários: versatilidade, veracidade, confiabilidade, continuidade temporal, tolerância a falhas e qualidades mentais e emocionais.

Após estabelecer sua tipologia, (NWANA, 1996) criou sete categorias de agentes: colaborativos, de interface, móveis, de informação, reativos, híbridos e inteligentes, mostrados na Figura 6.

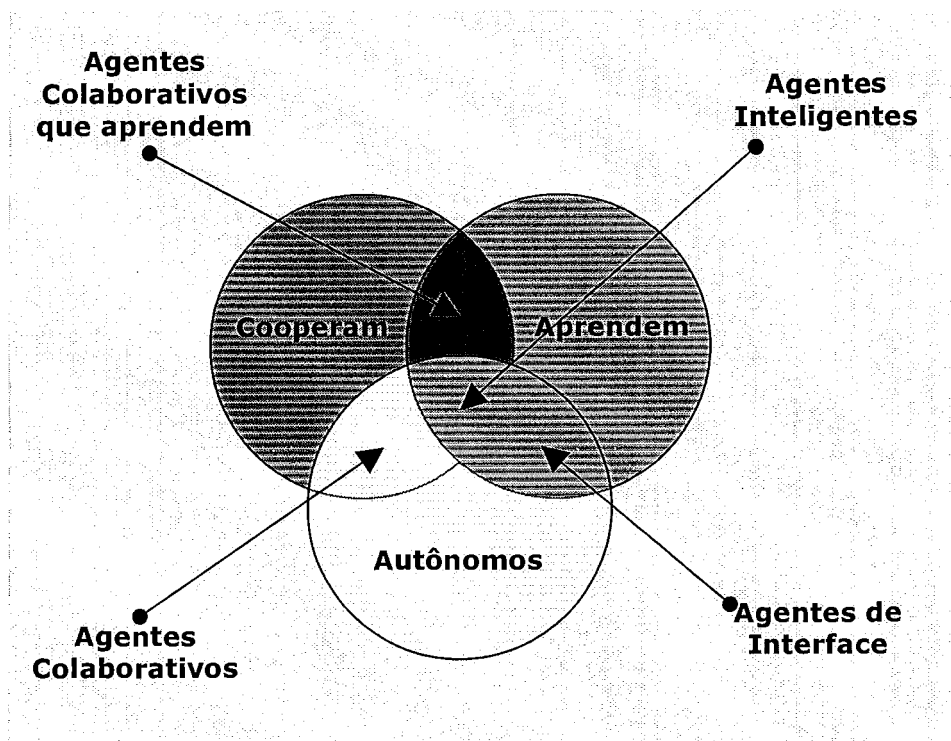


Figura 6 - Tipologia de agentes proposta por (NWANA, 1996).

(FRANKLIN e GRAESSER, 1996) acreditam que um agente, por definição, deve ser um processo em contínua execução e possuir, pelo menos, os atributos de autonomia, reatividade e proatividade. (WOOLDRIDGE e JENNINGS, 1995b) acrescentam ainda que um agente deve possuir habilidade social e, baseados em várias definições de agentes, apresentam uma taxonomia que engloba grande parte dos trabalhos em andamento.

(CAGLAYAN e HARRISON, 1997) estabelecem uma taxonomia através dos conceitos de ambiente, tarefa e arquitetura. Por ambiente deve-se entender o universo onde o agente atua, como um sistema operacional em particular ou a Internet. O conceito de tarefa está relacionado com o que um agente pode fazer. Arquitetura indica como o agente organiza internamente seu conhecimento. Utilizando estes conceitos, os

autores dividiram os agentes em três categorias: *desktop*, Internet e Intranet. Agentes *desktop* são basicamente agentes de interface que oferecem serviços de assistência aos seus usuários. Na categoria Internet, os agentes oferecem serviços de busca, filtragem e recuperação de informações. Os agentes móveis e de notificação também estão inclusos nessa categoria. Os agentes da categoria Intranet incluem aqueles que possuem capacidade para automatizar processos do fluxo de trabalho dentro das organizações. Outros agentes Intranet manipulam bancos de dados e alocam recursos em uma arquitetura cliente/servidor.

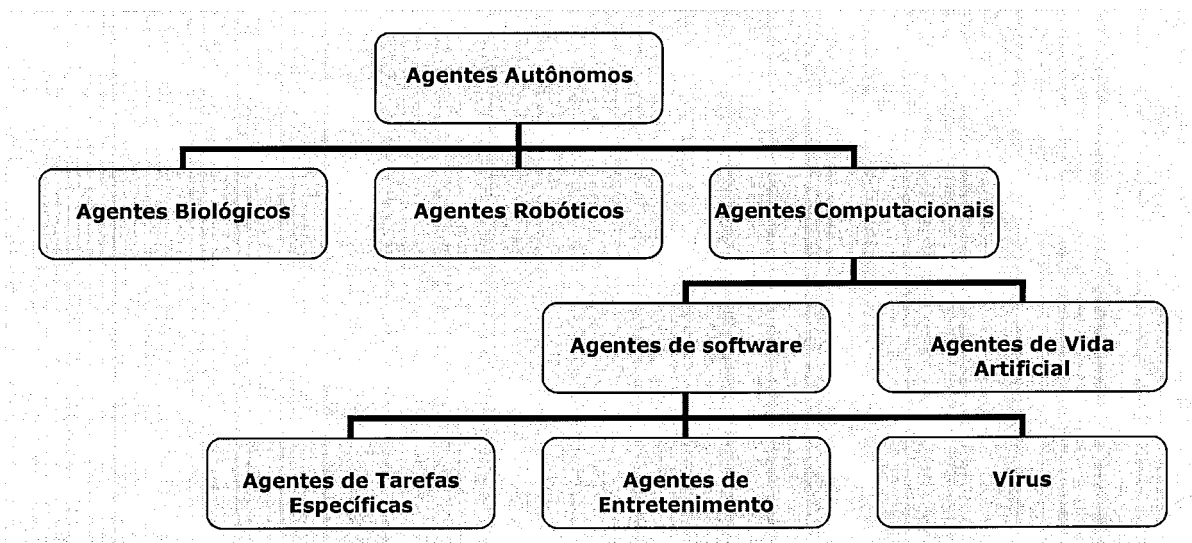


Figura 7 - Taxonomia de agentes proposta por (FRANKLIN e GRAESSER, 1996).

3.4 – Comunicação entre agentes

O poder dos sistemas de agentes depende da capacidade de comunicação entre eles. Os agentes devem ser capazes de comunicar-se com outros agentes e com os usuários para realizar suas tarefas efetivamente. Uma linguagem comum adotada amplamente para a representação de informações trocadas por agentes permitiria a interação e a interoperabilidade necessária nos ambientes distribuídos atuais.

(FININ e WEBER, 1994) estabelecem que existem diversos níveis nos quais sistemas baseados em agentes devem interagir:

- *Transporte*: como agentes enviam e recebem mensagens.
- *Linguagem*: qual o sentido de mensagens individuais.

- *Política*: como os agentes estruturam conversações.
- *Arquitetura*: como conectar sistemas em concordância com protocolos existentes.

Ainda que não exista uma linguagem padronizada, existem duas propostas amplamente aceitas: *Knowledge Query and Manipulation Language* (KQML) e *FIPA Agent Communication Language* (ACL). A primeira foi desenvolvida no início dos anos 90 como parte de um projeto do governo dos Estados Unidos, chamado *ARPA Knowledge Sharing Effort*. KQML é uma linguagem e um protocolo para a troca de informação e conhecimento que tem sido muito usada. A *Foundation for Intelligent Physical Agents* (FIPA) é uma organização sem fins lucrativos que tem a missão de produzir padrões para a interoperabilidade de agentes de *software* heterogêneos. Para evitar uma “guerra de padrões”, os participantes da FIPA têm incluído muitas das características de KQML (FIPA, 2005; LABROU, FININ et al., 1999).

A presença de uma linguagem comum para troca de mensagens é uma das características que diferencia um agente de um objeto na programação orientada a objetos. Uma mensagem no contexto dos agentes carrega uma semântica independente do agente, enquanto que uma mensagem no contexto dos objetos pode variar de um objeto para outro.

Apesar do mérito do KQML e do FIPA ACL, estes só se preocupam com o processo da comunicação agente-agente. Entendendo que um agente pode atuar por conta de uma pessoa ou organização, é necessário abordar também a comunicação usuário-agente. Continuam ainda sem resposta as questões sobre como uma tarefa é delegada ao agente, qual deve ser o formato no qual as preferências do usuário são transmitidas ao agente e como os estados de execução das tarefas podem ser monitorados e controlados pelo usuário.

Tanto o KQML quanto o FIPA ACL são baseados na teoria de Atos de Fala (em inglês, *Speech Act Theory*), desenvolvida por lingüistas com a intenção de entender como os humanos usam a linguagem em situações cotidianas (AUSTIN, 1962; SEARLE, 1969). A intenção de quem fala (ou seja, o efeito que se quer obter sobre quem escuta) é comunicada especificando o tipo de mensagem (isto é, se é uma

pergunta, uma afirmação, uma resposta etc.). Além disso, nenhum deles especifica a sintaxe ou a semântica do conteúdo das mensagens, visto que diferentes domínios de aplicação podem requerer diferentes linguagens de conteúdo. Algumas linguagens de representação de conteúdo de propósito geral foram desenvolvidas, como a KIF (do inglês *Knowledge Interchange Format*) (GENESERETH e FIKES, 1992), comumente usada com KQML, e a linguagem FIPA SL, mais utilizada com FIPA ACL.

O projeto de uma linguagem de comunicação entre agentes utiliza normalmente uma das seguintes abordagens: procedimental ou declarativa. Na abordagem procedimental, a comunicação acontece através de diretivas. Tanto comandos individuais quanto programas completos podem ser transmitidos e executados no lado receptor. Linguagens baseadas em *scripts*, tais como TCL (GRAY, 1995) e *Telescript* (WHITE, 1997), são exemplos da abordagem procedimental.

Na abordagem declarativa, a comunicação ocorre através da troca de estruturas declarativas, tais como definições ou asserções. Uma explicação bem detalhada da abordagem declarativa pode ser encontrada no trabalho de Labrou *et al*, onde é apresentada a ACL (LABROU, FININ *et al.*, 1999). A ACL foi o resultado do grupo ARPA KSE (*Knowledge Sharing Effort*) (NECHES, FIKES *et al.*, 1991). Pode-se dividir a ACL em três partes:

Vocabulário: funciona como um grande dicionário de palavras apropriadas às áreas de aplicação comum. Esse vocabulário compartilhado é chamado de ontologia.

Linguagem interna: KIF (*Knowledge Interchange Format*) é uma versão do cálculo de predicados de primeira ordem. Possui a capacidade de codificar dados simples, restrições, regras, expressões e outros fatores. Uma completa descrição da linguagem KIF e de como utilizá-la pode ser encontrada no trabalho de (GENESERETH e FIKES, 1992).

Linguagem externa: KQML é uma camada lingüística que pode encapsular estruturas KIF. Ela fornece informação contextual para uma comunicação mais eficiente. Uma das especificações iniciais de KQML pode ser encontrada em (FININ e WEBER, 1994).

Uma mensagem ACL é uma expressão KQML na qual os argumentos são termos ou sentenças KIF formadas por palavras no vocabulário ACL. Entretanto, é importante ressaltar que a linguagem KQML não foi idealizada para transmitir apenas conteúdo baseado em KIF. Na verdade, KQML pode ser bem mais abrangente, permitindo até a transmissão de conteúdo definido pelos próprios projetistas do sistema multi-agente em questão. Obviamente, a não utilização de uma linguagem padronizada de representação de conhecimento reduz a possibilidade de interação entre agentes desenvolvidos por projetistas diferentes.

3.5 – Arquiteturas de agentes

A arquitetura interna de um agente é, essencialmente, a descrição dos seus módulos e como eles trabalham juntos. É necessário definir a arquitetura interna de um agente para permitir ao projetista de um sistema baseado em agentes integrar os requisitos especiais da aplicação e definir a negociação entre eles (SHEN, NORRIE et al., 2001).

Para programar as características descritas na seção 3.2 – são necessários vários módulos apropriados. Agentes simples podem precisar de um número pequeno de módulos, enquanto outros, mais complexos, podem necessitar de um número muito maior. Os módulos mais comumente usados são: Interface de Comunicação, Percepção, Execução, Conhecimento Social, Auto-Conhecimento, Conhecimento do Domínio, Gestão de Conhecimento, Aprendizado, Raciocínio, Modelos de Solução de Problemas, Coordenação, Planejamento e Programação (*Scheduling*), Controle, Gestão de Conflito e Interfaces de Aplicações. É importante notar que um agente específico pode ser composto de um (sub)conjunto desses módulos. Em aplicações reais vários módulos podem ser convenientemente combinados em um só, assim como, um módulo pode ser subdividido em vários de nível menor (SHEN, NORRIE et al., 2001).

As arquiteturas usadas em sistemas baseados em agentes vão da muito simples até as muito complexas. Algumas revisões bibliográficas bastante completas podem ser encontradas nos trabalhos de (HUHNS e SINGH, 1998; SHEN, NORRIE et al., 2001; WOOLDRIDGE e JENNINGS, 1995a).

3.6 – Sistemas multiagentes

Atualmente, acredita-se que o uso de sistemas multiagentes (MAS, do inglês *Multi-Agent Systems*) é a melhor maneira de modelar e dar suporte a sistemas (e ambientes) abertos e distribuídos. Um MAS é uma rede de “solucionadores de problemas” (agentes) fracamente acoplados que trabalham juntos para solucionar um dado problema (WANG, CONRADI et al., 1999). As principais vantagens de um MAS são:

Descentralização: é possível a quebra de um sistema complexo em um conjunto de subsistemas cooperativos descentralizados.

Reuso (reutilização) de componentes/subsistemas: isto é, a construção de um novo sistema, possivelmente maior, mediante a interconexão e a interoperabilidade de subsistemas já existentes.

Suporte ao trabalho cooperativo: são capazes de melhor modelar e dar suporte às diferentes e variadas interações do trabalho cooperativo, dado que os agentes podem atuar como representantes interativos e autônomos das pessoas (dos humanos).

Flexibilidade: dão suporte à constante evolução dos sistemas, sendo possível adicionarem novos agentes ao sistema ou novas tarefas aos agentes já existentes, quando for necessário.

3.7 – Plataformas de desenvolvimento de agentes

MAS têm se constituído em uma área de pesquisa muito dinâmica, existindo na literatura numerosos trabalhos sobre plataformas e linguagens para o desenvolvimento desses sistemas. Em um estudo sobre *software* para sistemas multiagentes, realizado por Mangina, são apresentados 36 ferramentas e produtos para o desenvolvimento de sistemas multiagentes (MANGINA, 2002). Todos esses produtos são descritos brevemente. A Figura 8 apresenta uma relação dos produtos revisados nesse estudo.

	Academic	Commercial	Other
Free	AgentTool DECAF EXCALIBUR Agent JATLiteBean JESS LEE MAP / CSM MASSIVE KIT RePast SIM_AGENT StarLogo Xraptor	Bee-gent Comet Way CORMAS Cougaar FIPA-OS MAML ZEUS	Kaariboga SEMOA
Licensed	IMPACT JAFMAS / JIVE RETSINA TuCSon	ADK AgentSheets CABLE Grasshopper IDOL JACK JADE JADE / LEAP	ADK AgentSheets CABLE Grasshopper IDOL JACK JADE JADE / LEAP Living Markets NARVAL Voyager

Figura 8 - Produtos reportados no estudo realizado por Mangina (MANGINA, 2002).

Atualmente o site da AgentLink, a rede de excelência criada pela Comissão Européia para promover a pesquisa e o desenvolvimento da computação baseada em agentes lista 129 produtos. Dessa lista, se destacam o ZEUS Toolkit, o JACK, o Aglets e o JADE (AGENTLINK, 2005).

ZEUS foi desenvolvido pela British Telecom Lab e é um *framework* para o desenvolvimento de agentes colaborativos ou cooperativos e possui uma biblioteca de componentes escritos em Java. Cada agente ZEUS comporta uma camada de definição, uma camada de organização e uma camada de coordenação. A camada de definição representa o raciocínio do agente e as habilidades de aprendizado, suas metas, recursos, habilidades, crenças e preferências, entre outros. A camada de organização descreve as relações dos agentes com outros agentes. A camada de coordenação descreve a coordenação e as técnicas de negociação que o agente possui. Os protocolos de comunicação são construídos no topo da camada de coordenação e implementam a comunicação interagente (BRITISH TELECOM, 2003; NWANA, NDUMU et al., 1999).

JACK é um ambiente para construir, executar e integrar sistemas comerciais multiagentes usando uma metodologia baseada em componentes e uma linguagem de programação, chamada *JACK Agent Language* que estende a linguagem Java com conceitos de orientação a objetos (AGENT ORIENTED SOFTWARE GROUP, 2004).

Aglets são objetos Java, desenvolvidos no Laboratório de Pesquisas da IBM, em Tokyo, que podem se mover de uma máquina a outra através da Internet. Isto é, um aglet que está sendo executado em uma máquina pode parar de repente a sua execução, se despachar ele mesmo para uma máquina remota e continuar a execução nela. Quando o aglet se move, ele leva tanto o código quanto o estado (seus dados) (LANGE, 1997; LANGE e OSHIMA, 1998).

JADE é um *framework* de *software* para o desenvolvimento de sistemas multiagentes que segue as padronizações da FIPA. Sua meta é simplificar o trabalho ao mesmo tempo em que assegura o padrão. Trata-se de um *middle-ware* que implementa uma plataforma de agentes e um *software* de desenvolvimento, ou seja, não apenas facilita o desenvolvimento como também é utilizado para o gerenciamento de agentes. A plataforma pode ser distribuída por várias máquinas (que nem precisam compartilhar o mesmo sistema operacional) e sua configuração pode ser controlada por uma interface gráfica (GUI) remota (BELLIFEMINE, CAIRE et al., 2004). Dado que JADE foi a ferramenta escolhida para desenvolver o protótipo da arquitetura proposta, foi incluída uma seção com informações mais detalhadas sobre essa plataforma.

Shakshuki apresenta uma metodologia para avaliar as plataformas de sistemas multiagentes baseada nos critérios de disponibilidade, ambiente, desenvolvimento, propriedades características e desempenho (SHAKSHUKI, 2005). Vários estudos fazem comparações entre plataformas específicas (ALTMANN, GRUBER et al., 2001; CIANCARINI e WOOLDRIDGE, 2001; GERVAIS, GOMEZ et al., 2004; RICORDEL e DEMAZEAU, 2000; WANG e SORENSEN, 2003), enquanto outros fazem uma detalhada revisão bibliográfica sobre as tecnologias orientadas a agentes (CIANCARINI e WOOLDRIDGE, 2001; GERVAIS, GOMEZ et al., 2004)

3.7.1 – *Java Agent DEvelopment Framework (JADE)*

O JADE foi desenvolvido e continua sendo atualizado pela Universidade de Parma, na Itália. O principal objetivo do JADE é facilitar e simplificar o desenvolvimento de MAS, garantindo um padrão de interoperabilidade entre eles através de um conjunto abrangente de agentes de serviços de sistema, os quais tanto facilitam como possibilitam a comunicação entre agentes, utilizando as especificações da FIPA, como serviço de nomes e páginas amarelas, transporte de mensagens, serviços de codificação e decodificação de mensagens, e uma biblioteca de protocolos de interação pronta para ser usada. Toda comunicação entre agentes é feita via troca de mensagens. Além disso, o JADE lida com todos os aspectos que não fazem parte do agente em si e que são independentes das aplicações, tais como transporte de mensagens, codificação e interpretação de mensagens e ciclo de vida dos agentes (mostrado na Figura 9).

JADE pode ser considerado como um *middleware* de agentes que implementa um *framework* de desenvolvimento e uma plataforma de agentes. Em outras palavras, uma plataforma de agentes que segue especificações da FIPA e bibliotecas para desenvolvimento de agentes em Java. De acordo com (BELLIFEMINE, CAIRE et al., 2004), JADE foi escrito em Java devido a características particulares da linguagem, em especial a programação orientada a objeto em ambientes distribuídos heterogêneos. Foram desenvolvidos tanto pacotes Java com funcionalidades prontas para uso, quanto interfaces abstratas para se adaptar de acordo com a funcionalidade da aplicação de agentes.

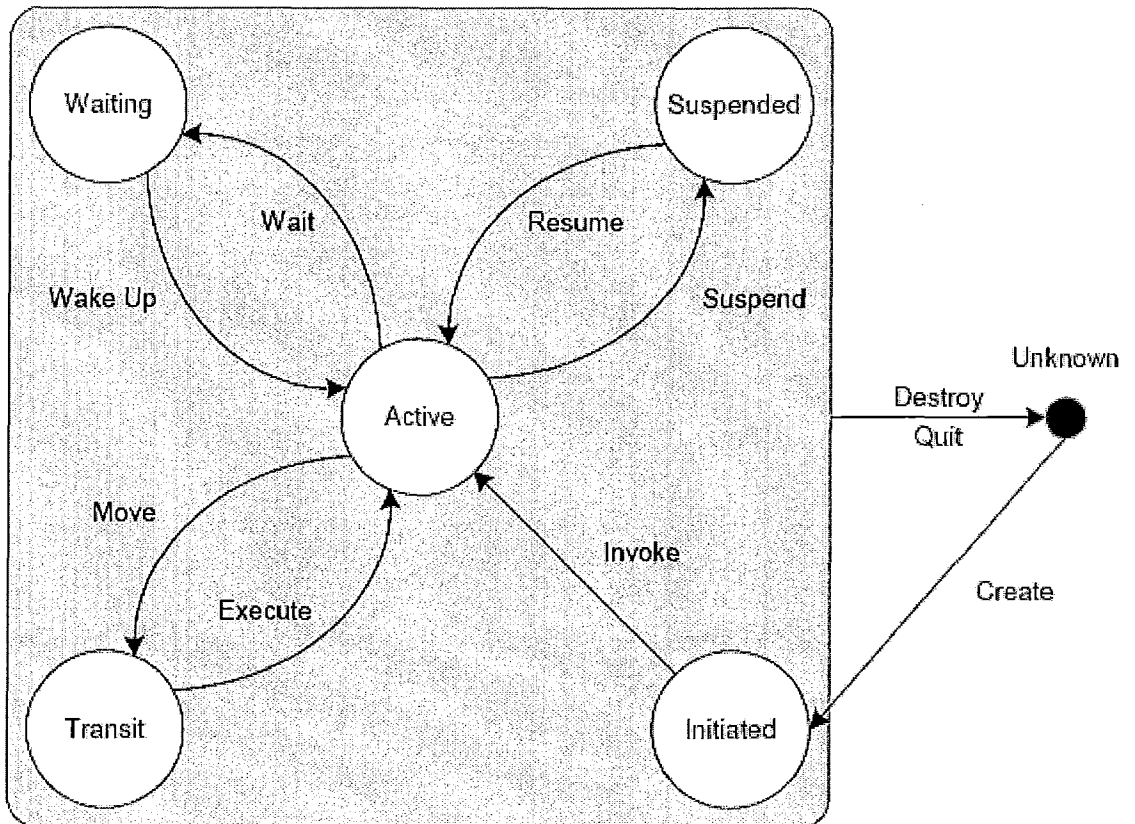


Figura 9 - Ciclo de vida dos agentes em JADE (BELLIFEMINE, CAIRE et al., 2004)

Seguem abaixo algumas características que JADE oferece para a programação de MAS (BELLIFEMINE, CAIRE et al., 2004):

- **Interface gráfica de usuário:** interface visual que gerencia vários agentes e contêineres de agentes, inclusive remotamente;
- **Ferramentas de depuração:** ferramentas que ajudam o desenvolvimento e a depuração de aplicações multiagentes baseadas em JADE;
- **Plataforma distribuída de agentes:** JADE pode ser dividido em vários *hosts* ou máquinas, desde que eles possam ser conectados via RMI (*Remote Method Invocation*). Apenas uma aplicação Java e uma Java Virtual Machine é executada em cada máquina. Os agentes são implementados como *threads* Java e inseridos dentro de repositórios de agentes chamados de contêineres (*Agent Containers*) que provêm todo o suporte para a execução do agente;
- **Ambiente de agentes conforme o padrão FIPA:** o qual inclui um sistema gerenciador de agentes (AMS - Agent Management System), um diretório

facilitador (DF - Directory Facilitator) e um canal de comunicação para os agentes (ACC - Agent Communication Channel) mostrados na Figura 10. Todos esses três componentes são automaticamente carregados quando o ambiente é iniciado;

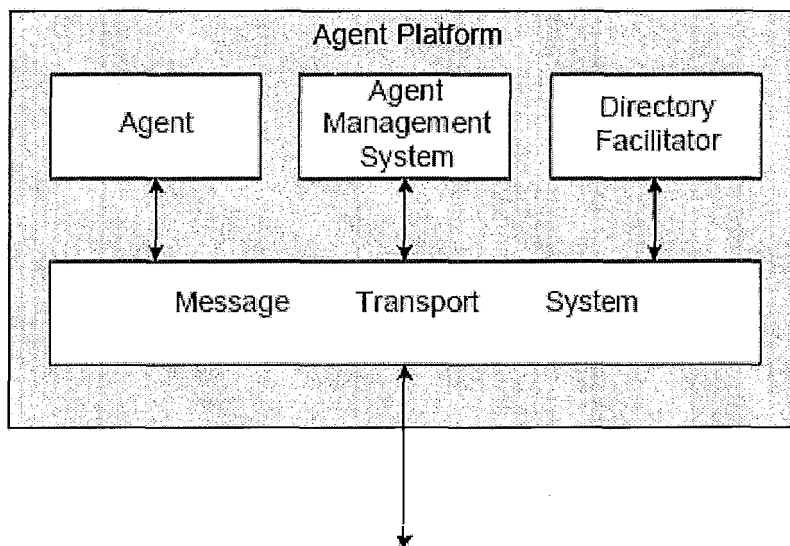


Figura 10 - Arquitetura padrão proposta pela FIPA (BELLIFEMINE, CAIRE et al., 2004)

- **Biblioteca de protocolos FIPA:** para interação entre agentes, o JADE dispõe de uma biblioteca de protocolos prontos para ser usados;
- **Suporte à execução de atividades:** as quais podem ser paralelas e concorrentes de agentes implementadas através dos modelos de comportamentos;
- **Transporte de mensagens:** transporte de mensagens no formato FIPA-ACL dentro da mesma plataforma de agentes;
- **Automação de registros:** registro e cancelamento automático de agentes com o AMS fazendo com que o desenvolvedor se abstraia disso; e,
- **Integração:** mecanismo que permite aplicações externas carregarem agentes autônomos JADE.

O JADE, além das características acima citadas, que por si só já facilitam muito o desenvolvimento de MAS, possui também algumas ferramentas muito úteis, que simplificam a administração da plataforma de agentes e o desenvolvimento de aplicações (BELLIFEMINE, CAIRE et al., 2004).

3.8 – Aplicações dos agentes

Agentes têm sido amplamente usados como assistentes de usuários, buscando informação, executando *matchmaking* entre usuários ou fornecendo recomendações. Algumas aplicações incluem Yenta, um sistema distribuído que relaciona usuários com interesses similares (FONER, 1997). Os agentes *Yenta* se comunicam com outros e compartilham informações sobre os interesses dos seus usuários, buscando similaridade ente eles. O *Remembrance Agent* é um assistente pessoal que observa os usuários enquanto escrevem e provêem recomendações de documentos possivelmente relacionados dentro de um conjunto de documentos fornecido pelo usuário (RHODES e STARNER, 1996). Uma aplicação baseada em agentes para *expertise*, o Expert Finder é um agente que estabelece o grau de *expertise* do usuário mediante o processamento de documentos e sugere pessoas que poderiam ter o conhecimento necessário (VIVACQUA e LIEBERMAN, 2000). Agentes também têm sido usados como guardiões de mecanismos de reputação mediante o processamento de redes sociais para estabelecer relacionamentos entre usuários (ZACHARIA e MAES, 2005) e buscando listas de discussão que poderiam interessá-lo, baseado no conteúdo da discussão (VAN DYKE, LIEBERMAN et al., 1999).

Agentes têm sido usados em *groupware* há muito tempo devido às habilidades sociais que possuem (BERGENTI, GARIJO et al., 2002). Uma revisão bibliográfica sobre o uso dos agentes em *groupware* pode ser encontrada em (ELLIS e WAINER, 1998). Ellis e Wainer apresentam os papéis que os agentes podem ter em sistemas de *groupware* e CSCW (ELLIS e WAINER, 1999). Eles identificam os seguintes papéis:

- *Guardiões*: agentes que tomam conta do ciclo de vida dos artefatos criados pelo grupo.
- *Coordenadores*: agentes que dão suporte à coordenação das atividades dentro do grupo.
- *Comunicadores*: agentes que dão suporte à comunicação entre os membros do grupo.

- *Agentes de equipes*: agentes que atuam como participantes ou membros artificiais das reuniões.

Sistemas como NEEM (ELLIS, BARTHELMESS et al., 2002), *Personal Assistant* (ENEMBRECK e BARTHÈS, 2002) e COLLABORATOR (BERGENTI, GARIJO et al., 2002) são exemplos bem sucedidos do uso de agentes no desenvolvimento de ferramentas colaborativas. AwServer, CScheduler e E_Places são bons exemplos de trabalhos de percepção baseada em agentes (ALARCÓN e FULLER, 2002).

Existem outros trabalhos também interessantes sobre o uso de agentes em *groupware*. Por exemplo, (CUNHA, FUKS et al., 2003) apresentam um MAS implementado para conformar grupos de trabalho entre alunos que assistem aulas no ambiente AulaNet (LUCENA, FUKS et al., 1999), um sistema de *groupware* desenvolvido para ensino e aprendizado via *Web*.

(ZHANG e TANNIRU, 2005) afirmam que a modelagem e a simulação baseada em agentes é uma metodologia apropriada pra o estudo de comunidades virtuais, (VLCs , do inglês *Virtual Learning Communities*). Onde cada participante é modelado como um agente dentro de sistema distribuído. Cada um possui características cognitivas e sociais como autonomia, interação, reatividade e persistência, que podem ser facilmente implementadas usando tecnologias representadas em agentes.

Capítulo 4 – CUMBIA

Esta dissertação propõe CUMBIA, uma arquitetura *peer to peer* baseada em agentes na qual cada usuário tem um conjunto de agentes especializados para auxiliá-lo nas tarefas de colaboração e gestão de conhecimento. Os agentes têm como principal objetivo identificar oportunidades potenciais de colaboração e prover meios para que ela seja o mais produtiva possível.

Dado que as interações espontâneas já não são tão frequentes como eram antes a troca e o compartilhamento de informações encontram-se comprometidos. Portanto, se faz necessário manter a distribuição de informações que acostumavam ser compartilhadas através de interações espontâneas e que podem propiciar colaborações futuras. Sistemas de computador podem fornecer essas informações aos usuários, permitindo a eles estar a par dos seus colegas, trocar conhecimento, discutir e colaborar.

Cada interação espontânea é uma oportunidade de aprender alguma coisa sobre os outros, tomar decisões, monitorar o estado de um trabalho e fazer correções se for necessário. A proximidade também facilita a verificação de disponibilidade, perceber intentos de comunicação e tomar parte em uma comunicação (KRAUT, FUSSELL et al., 2002). Essas informações se referem ao trabalho que está sendo feito, a quem o está fazendo e como está progredindo. Além disso, existe uma grande quantidade de informações adicionais de muita importância para o estabelecimento de colaborações: quem é a pessoa, quais são seus trabalhos anteriores, qual é sua experiência, qual é sua rede de contato e quais são seus atuais interesses e projetos. Toda essa informação é normalmente compartilhada com outros em interações informais e serve frequentemente na busca de ajuda ou quando se está pensando em entrar em contato com alguém a respeito de um assunto específico.

Matsuura introduz o conceito de proximidade virtual, definindo-o como situações nas quais os usuários acessam os mesmos dados ou invocam a mesma aplicação em um ambiente virtual (MATSUURA, FUJINO et al., 1993). Este trabalho adota um enfoque similar de “proximidade”, usando o contexto de trabalho individual atual como base para busca de outras pessoas com as quais seria interessante colaborar.

A interface de usuário implementada usa a metáfora do radar para representar a proximidade conceitual do usuário com os contatos escolhidos por este.

4.1 – Arquitetura

Os serviços oferecidos pela arquitetura podem ser classificados em quatro categorias como mostrado na Figura 11. Esses serviços são fornecidos por vários agentes que interagem entre si.

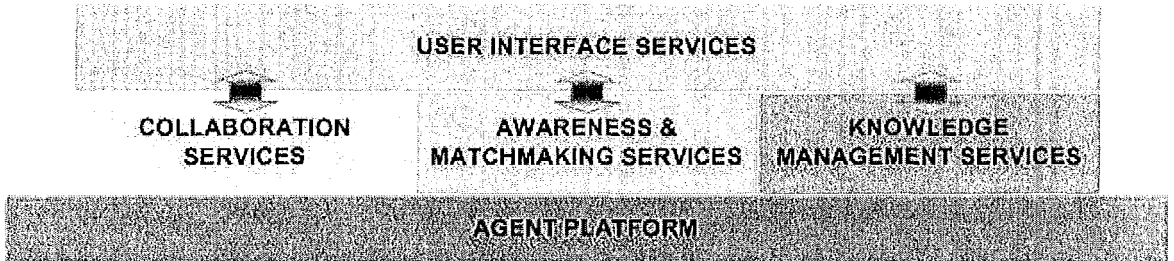


Figura 11 - Arquitetura de agentes CUMBIA (MORENO, VIVACQUA et al., 2003)

Gestão de Conhecimento – estes serviços administram os dados pessoais do usuário, constroem o perfil inicial baseado nesses dados; e armazenam informação sobre os documentos manipulados, as buscas realizadas, as colaborações e as pesquisas em andamento. Periodicamente verificam e re-avaliam a informação para garantir a fidelidade do perfil. Várias fontes de informação são usadas na construção do perfil de usuário: correio eletrônico, favoritos e publicações lidas ou escritas pelos usuários. Os documentos são analisados usando seu texto e são classificados por importância para o usuário, com, palavras e conceitos chaves. A classificação é também baseada no número, tempo e tipo (ler, escrever, imprimir) de acesso, bem como na distribuição (ou seja, se foi enviado para outras pessoas ou não).

Percepção e *matchmaking* – estes serviços procuram por outros usuários com quem poderia ser interessante estabelecer contato; interagem com outros agentes, solicitando o perfil e o contexto dos seus usuários; e comparam os perfis em concordância com o contexto atual e o ambiente de trabalho.

Colaboração – estes serviços permitem o estabelecimento fácil e rápido de comunicação quando a possibilidade de colaboração for detectada e o usuário decidir entrar em contato; e fornecem as ferramentas para o início de potenciais

cooperações/colaborações do tipo de ‘troca de mensagens’ o que é suficiente nesta fase, mas para as fases posteriores, tais como interações prolongadas ou colaborações mais complexas outro tipo de ferramentas é preciso.

Interface de Usuário – estes serviços apresentam a informação ao usuário e permitem especificar parâmetros e dar informação aos outros agentes. Dado que um dos objetivos da arquitetura é fornecer percepção e informação, que de outra forma não seria recebida esses agentes têm a função de decidir a forma e o momento apropriados de exibir a informação, evitando assim sobrecarga de informação e tentando ser discretos para não interromper o trabalho atual do usuário.

Nas seções seguintes se detalha as características de cada um dos serviços oferecidos por CUMBIA, a arquitetura *peer to peer* baseada em agentes para dar suporte à percepção (*awareness*) e à descoberta de oportunidades potenciais de colaboração, proposta nesta dissertação.

4.1.1 – Gestão de conhecimento

Os agentes do serviço de gestão de conhecimento são responsáveis pela funcionalidade de geração e persistência (armazenamento) dos perfis. As unidades básicas do perfil são os projetos e interesses do usuário. Uma parte do perfil é fornecida explicitamente ao sistema enquanto a outra é inferida automaticamente. Os usuários sempre têm o controle total sobre seus perfis, tendo a capacidade de mudar e corrigir todas as informações, assim como a capacidade de definir quando essas informações podem ser publicadas ou não.

Nessa arquitetura é assumido que o usuário sempre está trabalhando dentro de um *espaço de trabalho*. Portanto o contexto básico é formado pela definição desse espaço de trabalho que por sua vez é constituído pelos projetos, documentos, e contatos, entre outras características, do usuário.

Os projetos estão relacionados a documentos, pessoas, colaborações e pesquisas, mas, de maneira pessoal e local. Assim, um projeto cooperativo em andamento pode ter e certamente terá duas ou mais definições diferentes associadas, uma para cada usuário envolvido. Isso é coerente com o fato de que cada pessoa tem a sua própria visão da realidade e organiza o seu trabalho de acordo com essa visão. Logicamente, é útil

manter a correspondência entre os projetos individuais que representam o mesmo esforço grupal. Os projetos estão geralmente relacionados a um único espaço de trabalho, porém, é concebível que se esteja trabalhando em mais de um projeto simultaneamente, dentro de um mesmo espaço de trabalho, e também que se esteja envolvido em projetos representados por mais de um espaço de trabalho.

Os agentes do gerenciamento do perfil são responsáveis pela informação sobre os seguintes tópicos:

Informação de contato – informação necessária para que as outras pessoas possam contatar o usuário (nome, título, *e-mail*, telefone etc.).

Áreas de interesse – áreas nas quais os usuários têm algum tipo de interesse. Podem ser automática ou manualmente estabelecidas e “*rankeadas*” (ou qualificadas) por nível de interesse ou de atividade. As áreas de interesse podem estar relacionadas a projetos, pessoas ou contatos, documentos e históricos *web*.

Projetos – informação relacionada aos projetos nos quais o usuário está envolvido. O usuário cria manualmente projetos para organizar, no sistema, suas informações. Esses projetos são classificados de acordo com o estado de atividade: passados (já finalizados), presentes (em andamento, incluindo o atual) ou futuros (se tem a intenção de trabalhar). Os projetos podem ser associados a um prazo, ajudando na priorização do trabalho dos agentes. Parte da informação é coletada manualmente e parte será inferida do contexto. Projetos podem estar relacionados com áreas de interesse, pessoas, documentos e históricos *web*.

Pessoas – a lista de contato do usuário, a qual pode estar classificada em diferentes categorias, tais como contatos pessoais ou de trabalho, colaboradores (passados, atuais ou potenciais), pesquisadores, etc. Alguma informação é inferida dos correios eletrônicos (dos campos FROM, TO e CC). Os contatos estão relacionados aos projetos e às áreas de interesse quando os usuários têm interesses similares. Os usuários podem classificar e qualificar as pessoas de acordo a sua relevância em uma área de interesse ou projeto.

Histórico Web – Os agentes analisam as páginas que são visitadas na *Web*. O usuário também pode qualificar os sites de acordo com a relevância para o trabalho ou

projeto em andamento (essa informação pode ser inferida usando o tempo gasto na revisão do documento, o número de acessos, etc.).

O usuário dispõe de mecanismos de classificação para que se possa qualificar os demais usuários, os documentos, entre outros fatores, de acordo com a relevância para o projeto atual, fornecendo assim mais informação, que pode ser usada pelo sistema para melhorar sua efetividade.

4.1.1.1 – Processamento em lotes

Quando o ambiente é instalado, a primeira tarefa dos agentes é construir um perfil primário, analisando todas as informações disponíveis fornecidas pelo usuário. Primeiramente, os documentos de texto, os históricos *web* e a rede de contatos são processados da seguinte maneira:

Processamento de textos – os documentos são analisados usando técnicas de mineração de texto. As palavras e conceitos chaves são extraídos dos documentos e se estabelecem relacionamentos entre os documentos.

Processamento dos favoritos – os sites que apareçam na seção de *favoritos* do navegador são visitados, processados como documentos de texto e relacionados usando as palavras e conceitos chaves.

Processamento do histórico web – os sites que apareçam na seção histórico *web* são analisados também. Os sites são diferenciados das buscas, as quais são usadas para obter palavras chaves.

Extração de redes de contatos – os correios eletrônicos arquivados são processados extraindo palavras e conceitos chaves do corpo de texto e contatos dos campos FROM, TO e CC. Uma lista inicial de contatos é criada com base nos correios eletrônicos e nas ferramentas de mensagem instantânea. Os itens não classificados como pertencentes a um espaço de trabalho específico são relacionados com o espaço de trabalho genérico ou padrão, de onde podem ser recuperados mais tarde. Esse espaço de trabalho tem documentos, enlaces e contatos associados a ele que ainda não foram relacionados com outros criados pelo usuário. Periodicamente, os agentes analisam os itens do espaço genérico e sugerem possíveis relacionamentos, os quais podem ou não

ser aceitos pelo usuário.

4.1.1.2 – Processamento on-line

Enquanto o usuário trabalha, seu comportamento é observado pelos agentes, que tentam obter mais informações úteis. Os agentes processam, da mesma forma descrita no processamento em lotes, os documentos em que o usuário está trabalhando, assim como as ações tomadas em relação a esses documentos. A seguinte informação é extraída do comportamento:

Tempo gasto nos documentos – o tempo gasto em cada documento é medido.

Frequência de acesso – o sistema registra o número de acessos ao documento e a data do último acesso.

Tipo de acesso – o sistema também registra as ações tomadas em relação aos documentos, isto é, tenta inferir se o usuário está lendo, escrevendo ou publicando seus trabalhos (esta última é inferida quando se envia o documento para outros).

Essa informação é usada para representar a relevância ou importância dos documentos dentro do espaço de trabalho melhorando assim a efetividade do sistema.

As seguintes inferências são feitas:

Necessidades – a informação sobre as necessidades é extraída das palavras usadas nas buscas, referências ou nos documentos descarregados, colocados nos favoritos (“*bookmarks*”) ou guardados. Isso será usado para buscar informação relevante nos espaços de trabalho de outros usuários.

Uso dos recursos – para o sistema os recursos são os enlaces e documentos armazenados no computador do usuário e que este pode (ou está disposto a) compartilhar com outros usuários. O número de acessos e o tempo desses acessos (versus o tamanho do documento) determinam a importância que o documento tem nesse contexto. A distribuição também indica importância ou relevância. Ou seja, se o usuário considera um documento suficientemente bom para enviá-lo a outros então o sistema deve registrar esse comportamento e estabelecer relacionamentos com os contatos.

Conhecimento – os documentos escritos pelo usuário determinam seu conhecimento e os documentos lidos determinam o conhecimento que está sendo adquirido. A frequência e o tempo dos acessos indicam a intensidade do estudo realizado.

Toda a informação do perfil é armazenada em uma base de conhecimento. O sistema armazena a troca de mensagens (de texto, *e-mail*, discussões, etc.) e as relaciona com os espaços de trabalho, criando um histórico pessoal de trabalho. Esse histórico serve para orientar discussões futuras, para ajudar aos usuários a estabelecer entendimento entre eles e para a retomada de discussões não finalizadas. Adicionalmente, ajuda os usuários a estabelecer padrões de cooperação, determinando quais usuários foram mais cooperativos no passado e procurando novas interações, principalmente com aqueles mais cooperativos.

4.1.2 – Percepção e *matchmaking*

A partir do momento em que os perfis dos usuários são construídos, eles devem ser comparados a fim de se determinar um grau de proximidade e assim identificar oportunidades de colaboração entre os usuários. Os agentes baseados no contexto atual do usuário (determinado pelo espaço de trabalho) tentam detectar quando existe uma oportunidade de colaboração e alertam o usuário sobre a possibilidade.

A comparação de perfis é feita quando o usuário está *on-line* ou mesmo quando está *off-line*. A comparação *on-line* ocorre enquanto o usuário está trabalhando. O sistema procura achar outros usuários que estejam potencialmente interessados em uma colaboração dentro do contexto atual, isto é, usuários que possam estar interessados no que o usuário esteja fazendo ou que possam prestar assistência ou serem assistidos. Já a comparação *off-line* é feita baseada em toda a informação do usuário (não só a pertencente ao contexto do trabalho atual), ou seja, os agentes procuram informação sobre todos os projetos e/ou áreas de interesse nas quais o usuário está ou esteve envolvido.

A identificação da oportunidade de colaboração é um processo de três passos:

1. Dado o espaço atual de trabalho do usuário, buscam-se outros usuários trabalhando em espaços similares. Inicialmente, a busca é limitada aos

espaços ativos dos outros usuários. Se não existe um similar, a busca inclui então os espaços de trabalho inativos.

2. Dentro dos espaços de trabalho dos outros usuários similares encontrados é realizada uma segunda busca de documentos “próximos” ou parecidos e que não estejam presentes no espaço do usuário. Se o usuário está realizando uma busca, as palavras usadas nessa busca são usadas para buscar dentro desses espaços encontrados. Tenta-se determinar também neste passo a existência de reciprocidade. Entendendo por reciprocidade a possibilidade de dois usuários poderem oferecer algum recurso um para o outro, ou seja, a existência de necessidade mútua entre eles. Isso facilitaria o início de uma colaboração.
3. Os resultados do processo são enviados aos agentes de interface os quais se encarregam de notificar ao usuário da existência da oportunidade de colaboração.

O contexto atual é determinado manualmente pelo usuário. Os agentes então buscam no perfil a informação relacionada ao espaço de trabalho e constroem o “perfil do espaço”. Nesse perfil, se armazena a informação sobre os recursos associados ao espaço de trabalho (projetos, documentos, referências, enlaces etc.) e necessidades do usuário nos projetos, que são inferidas com base nas buscas realizadas e nos documentos baixados. Os agentes utilizam inicialmente só a informação do contexto atual dos usuários, pois é mais conveniente para estes não terem que mudar seu contexto para iniciar uma colaboração.

Quando a possibilidade é detectada, os agentes notificam o usuário. As oportunidades estão sempre ligadas ao tempo e, o usuário deve ser informado sobre a possibilidade, e a existência de reciprocidade, bem como receber um perfil disponível sobre o outro usuário que inclua informação sobre colaborações anteriores e o seu comportamento cooperativo. Outra informação útil, especialmente entre usuários desconhecidos, é tentar determinar a existência de contatos comuns entre os dois indivíduos.

4.1.3 – Colaboração

Uma vez que a oportunidade de colaboração foi identificada, os usuários podem se tornar colaboradores ocasionais ou colaboradores ativos. Colaboradores ocasionais são aqueles que fornecem sugestões ocasionais e atendem reuniões ocasionalmente. Colaboradores ativos são aqueles que estão envolvidos permanentemente e formalmente no projeto. Em ambos os casos, é importante conhecer o estado de cada participante, as tarefas pelas quais é responsável e o seu grau de disponibilidade. Nessa situação, também passam a ser úteis interfaces com ferramentas de gestão de projetos para dar assistência às colaborações ativas.

A idéia aqui é fornecer ferramentas básicas que dêem suporte à fase inicial de contato uma vez que a oportunidade de colaboração seja detectada e permitir a integração de ferramentas já existentes, que permitam mecanismos mais avançados de troca de informação tais como foros, correio eletrônico, lista de discussão etc. Essa integração é facilitada devido à arquitetura, que é baseada em agentes, e, em especial, à escolha de uma plataforma de desenvolvimento de agentes que cumpre os padrões propostos pela FIPA.

4.1.4 – Interface de usuário

A proximidade do usuário com cada um dos seus contatos é representada dentro de um radar dividido em setores, onde o centro representa ao usuário, como mostrado na Figura 2. Os diferentes setores representam as diferentes áreas de interesses ou projetos do usuário. Os outros usuários, relacionados na lista de contatos, são representados por ícones e aparecem em um setor do radar, onde a proximidade seja da maior relevância para o momento atual. A distância do ícone de cada usuário ao centro do radar representa a medida de proximidade calculada pelos agentes dos serviços de *matchmaking*. A cor e a forma do ícone representam o estado do usuário (isto é, *on-line*, *off-line*, ocupado, etc.) similar à maioria dos sistemas de troca de mensagens. Clicar ou deixar por um instante o cursor por cima do ícone fornece informação adicional sobre o usuário. Alguns serviços podem ser acessados pelo menu contextual (clicando com o botão direito do *mouse* sobre o ícone).

Capítulo 5 – Protótipo implementado

Neste capítulo, é apresentado o protótipo implementado para a validação do presente trabalho o qual foi desenvolvido usando predominantemente a linguagem de programação Java (SUN, 2005) e o JADE (TELECOM ITALIA LAB, 2005), embora alguns agentes específicos tenham sido desenvolvidos usando VBA (Visual Basic for Applications) da Microsoft.

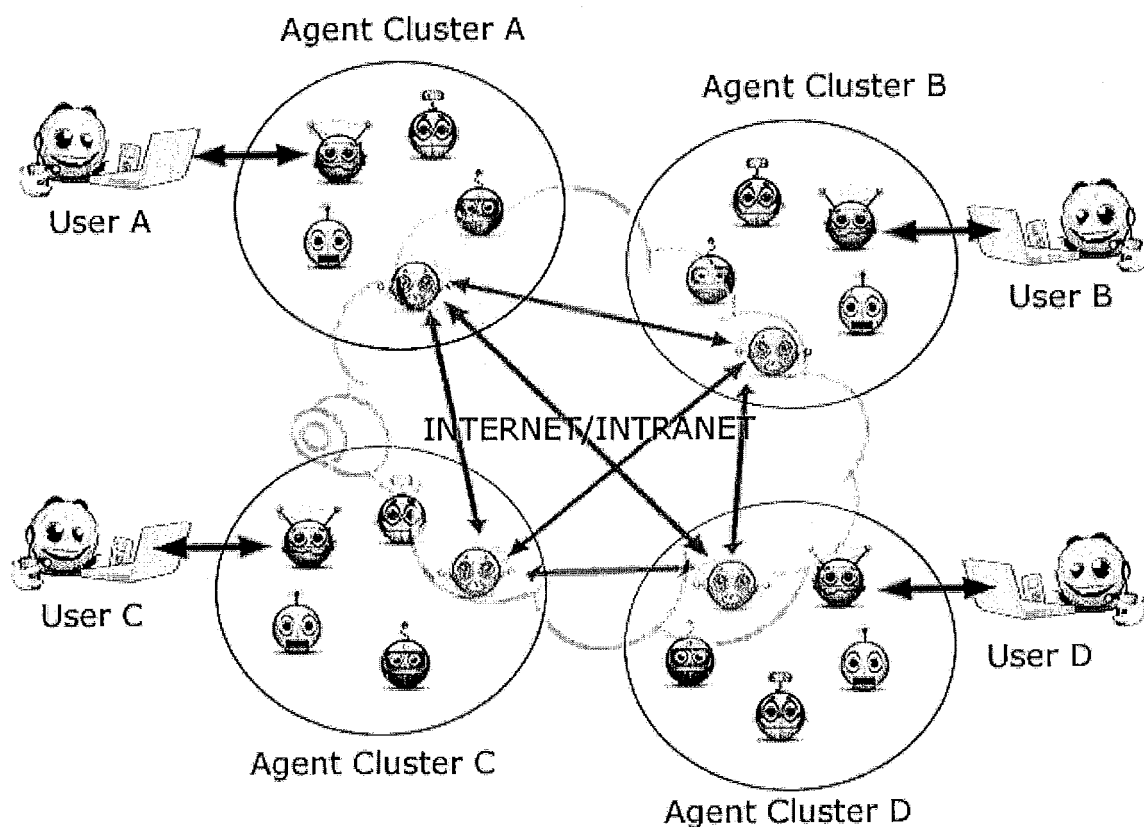


Figura 12 - Esquema da arquitetura

Dado que CUMBIA é um sistema *peer to peer* baseado em agentes, cada usuário tem um conjunto de agentes para auxiliá-lo, bem como a sua própria base de conhecimento e seu perfil. Este último pode ser compartilhado com os outros usuários, na medida em que esse comportamento seja definido nas políticas de privacidade. Ele é utilizado no processo de busca de “proximidade conceitual” através da análise e comparação com outros perfis.

Os agentes executam funções específicas e interagem entre eles a fim de proverem os serviços descritos no Capítulo 4. As três principais funções executadas são: extrair informação de aplicações e arquivos, calcular a proximidade entre os usuários e alertá-los da sua proximidade. A Figura 12 apresenta uma visão esquemática geral do sistema. Existem três tipos de agentes: *monitor*, encarregados de extrair informação do ambiente de trabalho, *profiler*, responsáveis pela construção dos perfis de usuários, e os *matchmaker*, que buscam e comparam as informações extraídas com as informações dos outros usuários, buscando oportunidades de colaboração.

5.1 – Agentes implementados

5.1.1 – Tipo *monitor*

Esses agentes são encarregados de monitorar o ambiente de trabalho de cada usuário e extrair informações sobre suas atividades. Eles são as principais fontes de informação do sistema e os agentes de outros tipos, processam essa informação a fim de cumprir suas funções.

Existem vários agentes do tipo *Monitor* que monitoram as operações do sistema operacional e cada aplicação instalada, sendo realizado, dessa forma, o monitoramento de todo o ambiente do usuário. Cada agente cria um arquivo XML de registro, que contém informação sobre os objetos manipulados, levando em consideração as ações executadas (abrir, salvar, imprimir, fechar etc.) para criar um histórico de manipulação dos objetos.

Um exemplo de arquivo XML de registro é apresentado na Figura 13. O arquivo especifica o objeto que está sendo manipulado e a ação que está sendo realizada nele. Dessa forma, é possível determinar a importância relativa do objeto no contexto do trabalho atual. Os agentes do tipo *matchmaker* utilizam esse arquivo com o intuito de detectar oportunidades de colaboração.

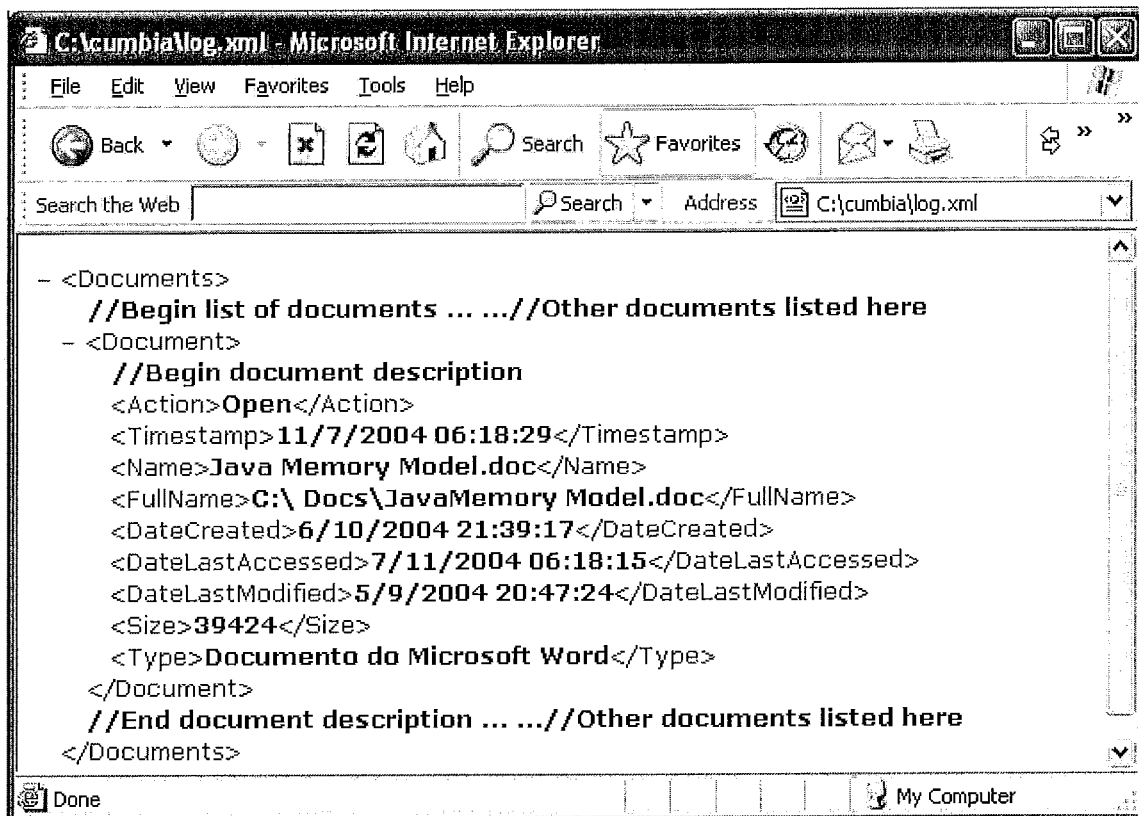


Figura 13 - Arquivo XML de registro da atividade do usuário gerado pelos agentes do tipo *monitor*.

O protótipo conta com um conjunto de agentes implementados para monitorar as principais aplicações de Microsoft Office usando VBA, o *WordAgent* para Word, o *ExcelAgent* para Excel e o *PPTAgent* para Power Point. Outro agente implementado em VBA monitora o Microsoft Windows e registra as aplicações utilizadas em um dado momento.

O *DirManagerAgent* é um agente desenvolvido usando JADE que monitora as pastas especificadas pelo usuário. Esse agente utiliza a ferramenta *Java KBtextmaster Natural Language Processing Toolkit*, uma API de mineração de textos de código aberto desenvolvida por Mark Watson (WATSON, 2004) e que foi modificada para atender as necessidades do protótipo. No desenvolvimento desse agente, foi utilizado também o *ConceptNet* (LIU e SINGH, 2004), uma ferramenta de raciocínio com sentido comum. O *ConceptNet* é uma API, desenvolvida na linguagem Python (PYTHON SOFTWARE FOUNDATION, 2005), com uma base de conhecimento de senso comum, criada no MIT Media Lab e baseada em uma rede de conceitos oriundos do projeto *Open Mind Commonsense* (MIT MEDIA LAB, 2004; SINGH, LIN et al., 2002), e de

algoritmos de processamento textual, com ênfase no raciocínio contextual.

Para atingir seus objetivos, o *DirManagerAgent* indexa e agrupa (*clustering*) todos os documentos acessíveis, isto é, todos os documentos que se encontram nas pastas especificadas pelo usuário e nos seguintes formatos: Microsoft Word™, Microsoft PowerPoint™, PDF™, OpenOffice.org Writer, AbiWord™, HTML e texto.. Dessa forma é possível fazer buscas de documentos por similaridade ou por comparação de texto (*full text search*).

Para cada um dos documentos processados é calculado um resumo, um vetor de categorias, dois de nomes (um com nomes de pessoas e outro com nome de lugares) e um com os conceitos (ou tópicos) chaves. O resumo, o vetor de categorias e os de nomes são calculados usando a API *KBtextmaster* do Mark Watson, já para o cálculo do vetor de conceitos chaves é usado o *ConceptNet*. Dado que este é desenvolvido em *Python*, os desenvolvedores disponibilizaram um servidor XML-RPC que viabiliza o uso em outras plataformas, sendo necessário um cliente XML-RPC. Esse foi desenvolvido com a utilização do Apache XML-RPC (APACHE SOFTWARE FOUNDATION, 2005b), uma implementação em Java do XML-RPC. O XML-RPC é um protocolo popular que usa XML sobre HTTP para implementar chamadas remotas a procedimentos.

O agente chamado *HTTPInterceptorAgent*, desenvolvido em JADE, captura o tráfego HTTP com o intuito de registrar e monitorar as páginas visitadas na Internet. Esse agente é um "Proxy sem cache". Ele guarda os endereços e código HTML das páginas visitadas em um arquivo de texto armazenado dentro de um diretório que faz parte daqueles monitorados pelo agente *DirManagerAgent*. Se existir permissão por parte do usuário, na primeira vez em que é executado, esse agente explora o histórico dos navegadores para realizar o processamento explicado anteriormente nas páginas anteriormente visitadas.

O agente *AwarenessAgent* é encarregado de monitorar a disponibilidade dos contatos, além da comunicação com os outros usuários, mediante troca de mensagens de textos e ainda pelo intercâmbio de perfis com os agentes dos demais usuários. Ele é desenvolvido em JADE e usa o serviço de "páginas amarelas" descrito no Capítulo 3 para esses fins. O intercâmbio de perfis é regulado pelas políticas de privacidade

estabelecidas pelo usuário no mesmo perfil e explicadas na seção seguinte.

5.1.2 – Tipo *profiler*

Esses agentes são encarregados de construir o perfil de cada usuário, com informações sobre interesses, contatos e documentos manipulados. A informação que serve de base para a construção do perfil é fornecida pelos agentes do tipo *monitor* e algumas, mais básicas, são fornecidas diretamente pelo usuário.

O perfil é um documento XML dividido em cinco seções ou categorias com a estrutura descrita a seguir:

Dados pessoais: essa seção corresponde à informação sobre a identidade do usuário e é fornecida por ele. Com o intuito de padronizar o perfil, foi escolhido o esquema de usuário do padrão P3P (*P3P "user" schema*)¹ para a especificação dos dados pessoais (P3P, 2005), cuja estrutura desta categoria é mostrada na Tabela 1. A maioria desses dados são opcionais sendo o apelido (ou *nickname*) e o ID de usuário (ou *login*) os únicos obrigatórios.

User	Category	Structure	Short display name
Name	Physical Contact Information, Demographic and Socioeconomic Data	personname	User's Name
Bdate	Demographic and Socioeconomic Data	date	User's Birth Date
Login	Unique Identifiers	login	User's Login Information
Cert	Unique Identifiers	certificate	User's Identity Certificate
Gender	Demographic and Socioeconomic Data	<i>unstructured</i>	User's Gender (Male or Female)
Employer	Demographic and Socioeconomic Data	<i>unstructured</i>	User's Employer
department	Demographic and Socioeconomic Data	<i>unstructured</i>	Department or Division of Organization where User is Employed
Jobtitle	Demographic and	<i>unstructured</i>	User's Job Title

¹ Ver esquema XML completo no apêndice 1

	Socioeconomic Data		
Home-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Home Contact Information
business-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Business Contact Information

Tabela 1 - Esquema dos dados pessoais do perfil dos usuários baseado em (P3P, 2005).

Documentos: essa seção corresponde à informação sobre os últimos documentos manipulados pelo usuário. A quantidade de documentos relacionados nessa seção depende das preferências do usuário. Quanto maior o número de documentos, maior será a precisão na busca. Entretanto, o volume de dados trafegando na rede e o tempo para o cálculo da proximidade, efetuado pelos agentes do tipo *Matchmaker*, que serão explicados em detalhes na próxima seção, também será maior.

Conceitos: essa seção é obrigatória e consta de um vetor ponderado de conceitos calculado com base na atividade do usuário, sendo atualizado a cada intervalo de tempo. Esse intervalo é definido pelo usuário e também especificado nesta mesma seção. O vetor é usado no cálculo de proximidade entre usuários.

Políticas de privacidade: o controle da privacidade da informação é responsabilidade do usuário e é ele mesmo quem decide quais informações podem ser compartilhadas e quais devem permanecer privadas. Para isso, ele deve preencher os campos correspondentes a essa seção, que são de tipo booleano, ou seja, do tipo sim ou não. Esses campos são *Personal*, *Documents* e *Reciprocity*. *Personal* refere-se a os dados pessoais que podem ser mostrados e compartilhados, *Documents* define se a seção de documentos será mostrada e compartilhada, *Reciprocity* determina se a reciprocidade será exigida ou não, ou seja, que só serão compartilhados aqueles dados que os outros usuários também estejam compartilhando. Caso o campo *Personal* esteja marcado para não ser compartilhado, o apelido e o ID seguiram sendo públicos, pois é a única maneira de identificar os usuários. Se o usuário só quer compartilhar parte dos dados pessoais, ele deverá compartilhar a seção, mas só preencherá os dados que deseja compartilhar, deixando vazios aqueles que preferir manter privados.

Para os fins do protótipo só foi desenvolvido um agente desse tipo chamado de *ProfilerAgent*, que realiza todas as funções descritas acima. Ele foi desenvolvido usando JADE e a ferramenta Apache XMLBeans (APACHE SOFTWARE FOUNDATION, 2005a). Esta permite acessar arquivos XML de programas desenvolvidos em Java através das classes geradas a partir do XML Schema.

5.1.3 – Tipo *matchmaker*

. Esse agente é encarregado de calcular a proximidade com os demais usuários. Cada usuário pode definir o escopo da busca, isto é, a busca pode ser limitada aos contatos ou pode ser aberta a qualquer pessoa que esteja usando o sistema nesse momento. A segunda opção é claramente mais custosa em termos de uso de CPU e de rede, mas as possibilidades de achar uma oportunidade de colaboração também são maiores. O cálculo da proximidade é baseado na informação extraída pelos agentes do tipo monitor sobre a atividade corrente do usuário.

Os agentes *MatchmakerAgent* comunicam-se entre si, enviando e recebendo as informações sobre seus usuários, os perfis, e calculam a proximidade baseados nos dados contidos neles. O intercâmbio de perfis sempre é feito respeitando as políticas de privacidade estabelecidas pelo usuário e descritas anteriormente.

Para os fins do protótipo, foi implementado só um agente desse tipo, o *MatchmakerAgent* que foi desenvolvido usando JADE e Apache XMLBeans. Esse agente usa o vetor de conceitos e um algoritmo clássico de cálculo de similaridade do modelo de espaço vetorial pra calcular a proximidade (LEE, CHUANG et al., 1997; SALTON, 1998). Foi escolhida a medida de similaridade do co-seno, que determina o ângulo entre os vetores dos usuários. Sejam X e Y os vetores correspondentes aos usuários “Usuário X” e “Usuário Y” respectivamente, a similaridade entre eles ($sim(X, Y)$) é calculada da seguinte forma:

$$sim(X, Y) = \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 + \sum_{i=1}^t y_i^2}}$$

Onde t representa o número total de conceitos nos vetores e, x_i e y_i são as

medidas de relevância do conceito i nos vetores X e Y respectivamente.

Por ser CUMBIA uma arquitetura baseada em agentes, é muito simples adicionar novos agentes com tarefas específicas para aumentar a capacidade de monitoramento. Na Figura 14, pode-se observar o esquema dos agentes implementados.

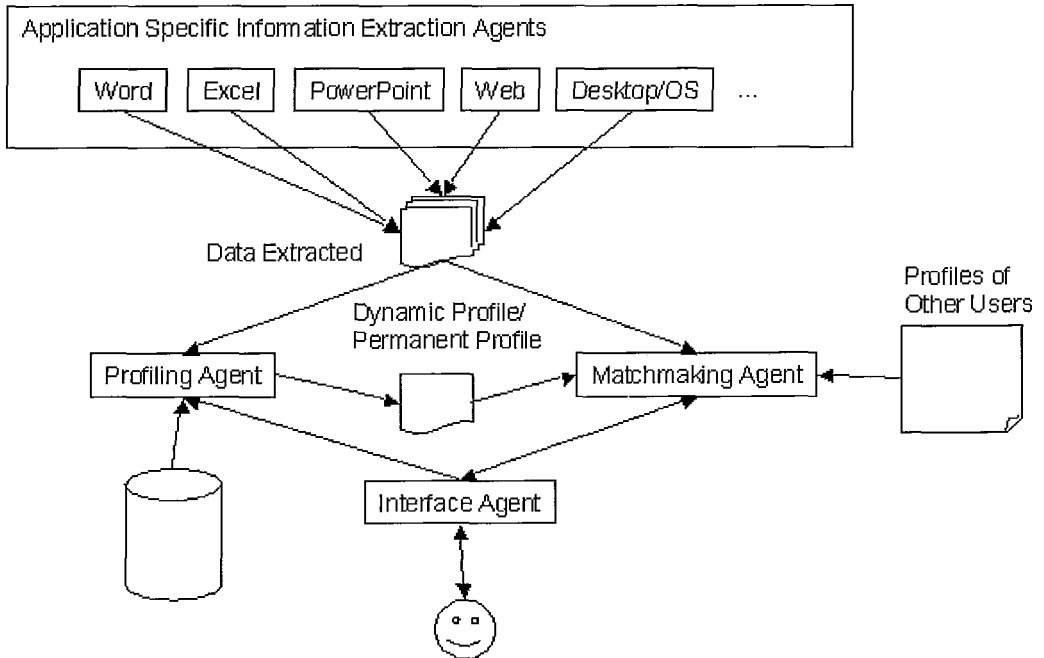


Figura 14 - Esquema do conjunto de agentes

5.1.4 – Interface do usuário

A proximidade de cada usuário é representada em uma tela de radar pelo *RadarAgent*, onde o centro representa o usuário, como apresentado na Figura 15 - Radar. Os outros usuários, relacionados na lista de contatos, são representados por ícones e aparecem em um setor do radar onde a proximidade seja de maior relevância para o momento atual. A distância do ícone de cada usuário ao centro do radar representa a medida de proximidade calculada pelos agentes dos serviços de *matchmaking*. A cor e a forma do ícone representam o estado do usuário (isto é, *on-line*, *off-line*, ocupado, etc.) similar à maioria dos sistemas de troca de mensagens. Clicar ou deixar por um instante o *cursor* por cima do ícone fornece informação adicional sobre o usuário. Alguns serviços podem ser acessados pelo menu contextual (clicando com o botão direito do mouse sobre o ícone).

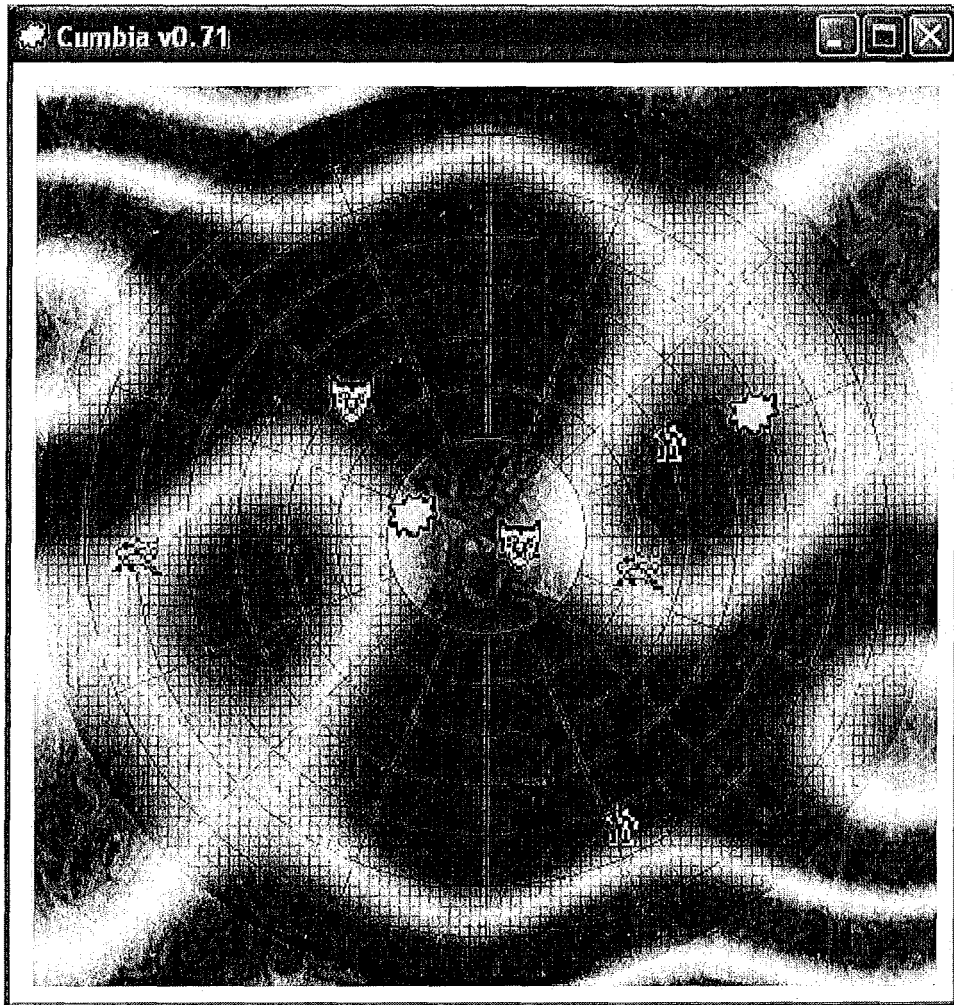


Figura 15 - Radar

5.2 – Mapeando os agentes na arquitetura proposta

Como visto no Capítulo 4, a arquitetura CUMBIA oferece um conjunto de serviços que podem ser classificados em quatro categorias: gestão de conhecimento, percepção e *matchmaking*, colaboração, interface de usuário. Além disso, se explicou que esses serviços são fornecidos por vários agentes que interagem entre si. A seguir, se explica a relação de cada um dos agentes implementados com os serviços oferecidos, que é resumida na Tabela 2 – Relação dos agentes implementados com a arquitetura proposta.

O agente do tipo *profiler* chamado *ProfilerAgent* e os agentes do tipo *monitor* chamados *WordAgent*, *ExcelAgent*, *PPTAgent*, *DirManagerAgent* e *HTTPInterceptorAgent* fornecem serviços de gestão de conhecimento, pois são os encarregados de capturar e armazenar os dados pessoais do usuário assim como de

construir o perfil.

Os serviços de *percepção e matchmaking* são oferecidos pelos agentes *AwarenessAgent* e *MatchmakerAgent* respectivamente. O *AwarenessAgent* fornece ainda a ferramenta de troca de mensagens prestando também um serviço de *colaboração*.

Agentes		Serviços da arquitetura			
Tipo	Nome	GC	PM	CO	IU
Monitor	<i>WordAgent</i>	●			
	<i>ExcelAgent</i>	●			
	<i>PPTAgent</i>	●			
	<i>DirManagerAgent</i>	●			
	<i>HTTPInterceptorAgent</i>	●			
	<i>AwarenessAgent</i>		●	●	●
	<i>RadarAgent</i>				●
Profiler	<i>ProfilerAgent</i>	●			
Matchmaker	<i>MatchmakerAgent</i>		●		

GC:Gestão Conhecimento - PM:Percepção e *matchmaking* – CO:Colaboração – IU:Interface de Usuário

Tabela 2 – Relação dos agentes implementados com a arquitetura proposta

O *RadarAgent* encarregado de apresentar a informação sobre a proximidade através de uma tela de radar, fornecendo um serviço de interface de usuário que poderia ser considerado também como um agente do tipo *monitor*. O agente *AwarenessAgent* possui a sua própria interface para permitir a troca de mensagens com os outros usuários.

Capítulo 6 – Estudo de caso

6.1 – Ambientes acadêmicos

Ambientes acadêmicos se caracterizam pela presença de pessoas com diferentes graus de experiência e conhecimentos sobre vários temas, possibilitando que especialistas ajudem aos principiantes no aprimoramento de suas habilidades. Todos os membros desses ambientes são geradores e consumidores de informação e conhecimento em maior ou menor grau.

A universidade é um ambiente acadêmico formado por professores e alunos. O professor é um gerente, facilitador e guia que cria espaços onde todos os participantes desenvolvam seus potenciais e alcancem os objetivos da educação (aprender a ser, aprender a aprender e aprender a fazer) e usa esses espaços para alcançar os objetivos das instituições de educação superior (gerar, preservar e difundir o conhecimento).

O advento da tecnologia permitiu que muitos processos fossem transferidos do papel para o meio eletrônico, agilizando assim a troca de informações e o acesso a diversos serviços. Um dos ambientes mais beneficiados neste contexto foi o ambiente acadêmico, sendo cada vez mais comum o uso dessa tecnologia no apoio aos processos educativos e de pesquisa. Nesse contexto há uma tendência à formação de equipes de trabalho temporárias para enfrentar projetos específicos de pesquisa ou de desenvolvimento ou simplesmente de projetos de disciplinas.

É também cada vez mais comum o uso de ambientes virtuais de aprendizagem na educação, que são disponibilizados com o intuito de facilitar o aprendizado mediante o trabalho em grupos, ou seja, através de aprendizado cooperativo/colaborativo. Espera-se que com a introdução desses ambientes a colaboração seja espontânea, não obstante, muitas vezes a cooperação simplesmente não se dá ou ela tem que ser estimulada externamente por um tutor (SANTORO, BORGES et al., 2000).

Sendo a pesquisa uma das atividades típicas desses tipos de ambientes, é comum a formação de grupos de trabalhos com o objetivo de apresentar um projeto e uma vez concluído esse projeto os grupos são desmontados. Além disso, com frequência pessoas

externas participam ocasionalmente como assessores, e algumas vezes, dado o interesse no projeto, passam a formar parte do grupo. Em geral, a construção dinâmica de grupos temporários é utilizada como estratégia colaborativa, e é característica do trabalho em ambientes de pesquisa.

Neste capítulo apresenta-se como CUMBIA poderia ajudar a incentivar a colaboração entre os participantes de uma comunidade acadêmica

6.2 – O uso do protótipo

6.2.1 – Cenário

Na universidade, especificamente na graduação, se exige o desenvolvimento e apresentação de um projeto final de curso. Este é normalmente desenvolvido por grupos de dois ou três estudantes. Mesmo que os temas desses projetos sejam diferentes eles podem pertencer à mesma área do conhecimento ou a áreas similares (ou relacionadas). É também comum que os estudantes, nessa altura do curso, não estejam muito tempo na universidade e dediquem mais tempo em casa para a realização desses projetos.

Uma vez definido o tema do projeto a fase inicial é uma revisão bibliográfica. E a fase final é uma edição colaborativa de um texto que envolve aos participantes do projeto. Nessas condições, é interessante que os estudantes estejam a par do que seus colegas, tanto os membros do seu grupo quanto de aqueles colegas que estejam também desenvolvendo projeto final de curso em outros grupos. Isto com o intuito de detectar possibilidades de colaboração entre diferentes grupos de projetos.

6.2.2 – Primeiros testes

Uma primeira avaliação foi feita tomando os documentos consultados por um grupo de alunos de graduação e construindo seus perfis baseados neles. Posteriormente foi rodado o algoritmo de matchmaking e feito o cálculo de proximidade.

Por outro lado, uma revisão manual foi feita sobre se os mesmo documentos e se avaliou a existência de proximidade entre os conjuntos de documentos revisados. Se questionou a possibilidade de ter sido feita uma recomendação no sentido de incentivar

colaboração entre esses estudantes.

Posteriormente essa avaliação manual foi comparada com os resultados fornecidos pelo sistema. Em quanto a os vetores gerados no perfil se encontrou bastante representatividade e enquanto a o resultado do algoritmo de *matchmaking* encontrou se certas similitudes na maioria dos casos mas algumas possibilidades não forao detectadas.

6.2.3 – Segunda avaliação

6.2.3.1 – Participantes

Professores orientadores de projetos e estudantes da graduação desenvolvendo projetos finais de curso serão convidados a participar de um teste do sistema implementado usando a interface implementada.

6.2.3.2 – Procedimento

O procedimento para a realização do estudo pode ser sintetizado pelos passos apresentados abaixo. A ordem cronológica de execução está de acordo com a ordem de apresentação dos mesmos:

- Explicação para todos os participantes dos objetivos da dissertação em geral
- Explicação dos objetivos específicos dos testes
- Explicação para todos os participantes da arquitetura CUMBIA
- Explicação para cada participante sobre o uso do protótipo implementado.
- Entrevista individual com os professores
- Entrevista grupal com os estudantes

Nas entrevistas se solicitará a os participantes avaliar os seguintes aspectos: (i) Nível de Interrupção Gerada, (ii) Facilidade de utilização, (iii) Utilidade do reconhecimento de proximidade, (iv) Aumento de produtividade individual e (v) Satisfação.

6.2.3.3 – Resultados esperados

Espera-se que os participantes concordem em que a interface em geral não atrapalha o trabalho em andamento.

Nas entrevistas os participantes serão questionados sobre a representatividade dos conceitos contidos nos vetores que fazem parte do perfil e que são usados no processo de busca de oportunidades. A facilidade de uso da interface, as opções sobre o controle da privacidade e a ferramenta de colaboração implementada. Também sobre as oportunidades notificadas assim como as ações tomadas a partir do conhecimento de alguma oportunidade de colaboração. Deseja analisar também, se o fato de saber que outros colegas estejam trabalhando em coisas parecidas no mesmo momento os incentiva ou não a estabelecer comunicação.

Capítulo 7 – Conclusões

Um dos principais problemas enfrentado quando parte do trabalho é feito em ambientes virtuais é a perda de oportunidades de interações espontâneas ou não planejadas. As interações oportunísticas usualmente ocorrem quando os indivíduos localizados em um mesmo espaço físico encontram-se por acaso nos corredores, ou passam no escritório dos outros para troca rápida de idéias. Isso tem causado um crescente interesse em sistemas de percepção (ou *awareness*) e no reconhecimento da importância das interações informais (KRAUT, FISH et al., 1990) e da proximidade em esforços colaborativos (BRADNER, 2002; KRAUT, EGIDO et al., 1990; KRAUT, FISH et al., 1990; OLSON e OLSON, 2000).

A comunidade de pesquisadores da área de CSCW reconhece a necessidade de sistemas de percepção específicos para o suporte das atividades colaborativas nos ambientes virtuais. Entretanto, a maioria dos sistemas propostos preocupa-se por fornecer informações sobre o estado atual do usuário (se está conectado ou não, etc.), as atividades correntes e passadas ou os objetos que eles está manipulando (KANERVA, KOSKINEN et al., 2000), esperando que as colaborações ocorram simplesmente porque o usuário pode observar os outros (SANTORO, BORGES et al., 2000). Esses sistemas não fornecem informação adicional que permita estimular colaboração entre esses usuários. Por outro lado, a maioria deles se preocupa por fornecer as informações, mas nem sempre tem um foco específico do motivo para fornecê-las.

Esta dissertação propõe a CUMBIA, uma arquitetura baseada em agentes que, mediante o uso desse tipo de informação, tem o objetivo de construir o perfil de cada usuário que serve como base para calcular similaridade de interesses, ou a proximidade conceitual entre os contextos dos usuários. Estas são informações que ajudam oportunidades de colaboração a serem descobertas em ambientes virtuais de trabalho a partir do análise do ambiente de trabalho individual de cada usuário. O fato de não precisar de um ambiente comum diminui a quantidade de informação que trafega pela rede, sendo fornecida só a informação relevante para cada um dos usuários.

Uma preocupação importante quando se compartilha qualquer tipo de

informação e à privacidade (FARKAS, ZIEGLER et al., 2002; SHÜMMER, 2004), por esse motivo o sistema proposto oferece mecanismos que permitem aos usuários decidirem quais informações podem ser compartilhadas e quais devem ser restritas.

Um primeiro protótipo foi implementado e alguns testes iniciais foram realizados, obtendo assim importantes informações que serão usadas para o aprimoramento do sistema e no desenvolvimento de novas versões do protótipo.

7.1 – Contribuições

Ainda que existam propostas similares que tentam analisar o contexto atual do usuário e descobrir oportunidades potenciais de colaboração, estas precisam analisar o espaço virtual comum de interação entre os usuários. Dessa forma, o principal diferencial da arquitetura CUMBIA, proposta nesta dissertação, é o cálculo da similaridade e da proximidade com base no espaço individual de trabalho e a sua representação através da metáfora do radar.

Sendo a CUMBIA baseada em uma plataforma de agentes padronizados, é permissível a integração de ferramentas desenvolvidas em qualquer outra plataforma de agentes que cumpra os padrões da FIPA (FIPA, 2005), assim como a integração de ferramentas já existentes, não necessariamente desenvolvidas com tecnologias baseadas em agentes, através do desenvolvimento de um agente tipo interface que a represente e permita a comunicação entre ela e os demais agentes.

7.2 – Trabalhos futuros

Como trabalho futuro será realizado um teste com o protótipo implementado, utilizando o cenário e o procedimento descrito no estudo de caso, mas com um maior número de participantes e por um tempo que permita coletar maiores informações sobre a eficácia do sistema na detecção e sugestão de oportunidades de colaboração, assim como validar de forma mais exata as técnicas de *matchmaking* utilizadas.

Um segundo protótipo encontra-se em fase de projeto, com maior número de agentes para monitorar outros aspectos do ambiente de trabalho individual dos usuários, o que permitirá um melhor detalhamento do perfil, com uma nova interface e com a integração de algumas ferramentas de colaboração que permitam aumentar a efetividade

do sistema proposto. Nesse novo protótipo se espera introduzir as recomendações obtidas dos testes que serão realizados. Projetaram-se novos testes com diversos cenários, usando esse novo protótipo e com uma bateria de testes mais abrangentes e detalhados.

O sistema pode ser usado em muitos outros ambientes organizacionais com o intuito de detectar oportunidades de colaboração que possam contribuir na realização de trabalhos e incrementar a produtividade das pessoas envolvidas.

Referências Bibliográficas

AGENT ORIENTED SOFTWARE GROUP, 2004, "JACK". In: <http://www.agent-software.com/>, Accessed in 28/08/2005.

AGENTLINK, 2005, "AgentLink Site". In: <http://www.agentlink.org>, Accessed in 28/08/2005.

ALARCÓN, R., FULLER, D., 2002, "Intelligent Awareness in Support of Collaborative Virtual Work Groups". In: *Groupware: Design, Implementation, and Use 8th International Workshop, CRIWG 2002 Lecture Notes in Computer Science*, v. 2440, pp. 147-167

ALDUNATE, R., NUSSBAUM, M., GONZALEZ, R., 2002, "An Agent Based Middleware for supporting Spontaneous Collaboration among Co-Located, Mobile and not Necessarily Known People". In: *Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, CSCW 2002*, New Orleans, USA,

ALTMANN, J., GRUBER, F., KLUG, L., et al, 2001, "Using Mobile Agents in Real World: A Survey and Evaluation of Agent Platforms". In: *2nd Workshop on Infrastructure for Agents, MAS, and Scalable MAS at Autonomous Agents*, May

APACHE SOFTWARE FOUNDATION, 2005a, "Apache XMLBeans: XML-Java binding tool". In: <http://xmlbeans.apache.org/>, Accessed in 07/02/2005a.

APACHE SOFTWARE FOUNDATION, 2005b, "Apache XML-RPC". In: <http://ws.apache.org/xmlrpc/>, Accessed in 07/02/2005b.

AUSTIN, J. L., 1962, *How to Do Things With Words*, Oxford, England, Oxford University Press

AVOURIS, N. M., 1992, "Distributed Artificial Intelligence: Theory and Praxis", v. 5, *Computer and Information Science*, Kluwer Academic Publishers

BANNON, L. J.,SCHMIDT, K., 1989, "CSCW: Four Characters in Search of a Context". In: *First European Conference on Computer Supported Cooperative Work (ECSCW'89)*, pp. 358-372,

BELLCORE, 1989, *The VideoWindow Teleconferencing Service Model. Special Report SR-ARH-001424*. Bell Communications Research

BELLIFEMINE, F., CAIRE, G., TRUCCO, T., et al, 2004, "JADE PROGRAMMER'S GUIDE". In: <http://jade.tilab.com/doc/programmersguide.pdf>. Accessed in 02/11/2004.

BEN-SHAUL, I.,KAISER, G. A., 1995, *A paradigm for decentralized process modeling*, Boston/Dordrecht/London, KluwerAcademic Publishers

BERGENTI, F., GARIJO, M., et al, 2002, "Enhancing Collaborative Work through Agents". In: *VIII Convegno dell'Associazione Italiana per l'Intelligenza Artificiale*,

BOND, A. H.,GASSER, L., 1988, "An analysis of problems and research in DAI". In BOND, A. H. and GASSER, L., *Readings in Distributed Artificial Intelligence*, San Mateo, CA, Morgan Kaufmann Publishers

BOND, A. H.,GASSER, L., 1992, "A subject-indexed bibliography of distributed artificial intelligence", *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, n. 6, pp. 1260-1281

BOURNING, A.,TRAVERS, M., 1991, "Two approaches to casual Interaction on Computer and Video Networks". In: *International Networking Conference*

BRADNER, E. M. G., 2002, "Why distance matters: effects on cooperation, persuasion and deception". In: *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pp. 226-235, New Orleans, Louisiana, USA

BRITISH TELECOM, 2003, "ZEUS". In: <http://more.btexact.com/projects/agents/zeus/index.htm>, Accessed in 05/11/2003.

CAGLAYAN, A., HARRISON, C., 1997, *Agent Sourcebook*, John Wiley & Sons

CASTELFRANCHI, C., CONTE, R., 1996, "Distributed artificial intelligence and social science: Critical issues". In O'HARE, G. M. P. and J.N.R., *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons

CIANCARINI, P., WOOLDRIDGE, M., 2001, "Agent-oriented software engineering". In: *Proceedings of the 1st International Workshop on Agent-Oriented Software Engineering, Lecture Notes in Computer Science*, v. 1957, pp. 1-24

CONRADI, R., JACCHERI, M. L., MAZI, C., "Use and Implementation of SPELL, a language for Software Process Modeling and Evolution". In: *Second European Workshop on Software*

CUNHA, L. M., FUKS, H., LUCENA, C. J. P., 2003, "Setting Groups of Learners using Matchmaking Agents". In: *Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education - CATE 2003*, pp. 321-326, Rhodes - Greece, June

DE FARIAS, C. R. G., PIRES, L. F., VAN SINDEREN, M., 1999, *AMIDST*

DEMI, S., ESTUBLIER, J., AMIOUR, M., 1998, "APEL: A Graphical Yet Executable Formalism for Process Modeling". In NITTO, E. and FUGGETTA, A., *Process Technology*, Milan, Politecnico di Milano and CEFRIEL - Kluwer Academic Publishers

DOURISH, P., BELLOTI, V., 1992, "Awareness and coordination in shared workspaces". In: *ACM Conference on Computer Supported Cooperative Work (CSCW)*, pp. 107-114,

DOURISH, P., BLY, S., 1992, "Portholes: Supporting Awareness in distributed Work Group". In: *Computer Human Interface*

ELLIS, C. A., BARTHELMESS, P., QUAN, B., et al, 2002, *NEEM: An Agent Based Meeting Augmentation System. Technical Report CU-CS-937-02*. University of Colorado at Boulder, Computer Science Department

ELLIS, C. A., GIBBS, S. J., REIN, G. L., 1991, "Groupware: Some issues and experiences", *Communications of the ACM*, v. 34, n. 1, pp. 38-58

ELLIS, C. A., WAINER, J., 1994, "A conceptual model of groupware". In: *ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, pp. 79-88,

ELLIS, C. A., WAINER, J., 1998, "Agents in Groupware Systems", Accessed in 2004.

ELLIS, C. A., WAINER, J., 1999, "Groupware and Computer Supported Cooperative Work". In Weiss, G, *Multiagent, Systems*, MIT Press

ENEMBRECK, F., BARTHÈS, J., 2002, "Personal Assistant to Improve CSCW". In: *Proceedings of the 7th International CSCWD*, pp. 329-335, Rio de Janeiro, Brasil

ESBORJÖRNSSON, M., ÖSTERGREN, M., 2002, "Issues of Spontaneous Collaboration and Mobility". In: *Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, UBICOMP'02*, Göteborg, Sweden,

FARKAS, C., ZIEGLER, G., MERETEI, A., et al, 2002, "Anonymity and accountability in self-organizing electronic communities". In: *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society WPES '02*, pp. 81-90, New York, NY, November

FERREIRA, A., 2000, *Aurelio Século XXI: O minidicionário da lingua portuguesa*, Rio de Janeiro, Nova Fronteira

FININ, T., WEBER, J., 1994, "DRAFT: Specification of KQML Agent Communication Language. The DARPA Knowledge Sharing Initiative External Interfaces Working Group". In: <http://logic.stanford.edu/sharing/papers/kqml.ps>.

FIPA, 2005, "The Foundation of Intelligent Physical Agents Web Site". In: <http://www.fipa.org/>, Accessed in 07/02/2005.

FISH, R. S., 1989, "Cruiser: A Multimedia System for Social Browsing", *SIGGRAPH Video Review*, v. 6, n. 45

FISH, R. S., KRAUT, R. E., CHALFONTE, B. L., 1990, "The VideoWindow System in Informal Communications". In: *Conference on Computer Supported Cooperative Work (CSCW)*,

FISHER, G., 2001, "External and Shareable Artifacts as Opportunities for Social Creativity in Communities of Interest". In: *Computational and Cognitive Models of Creative Design (Proceedings of the Fifth International Conference)*, pp. 67-89, Heron Island, Australia, December

FONER, L. N., 1997, "Yenta: A Multi-Agent, Referral-Based Matchmaking System". In: *First International Conference on Autonomous Agents (Agents '97)*, Marina del Rey, CA

FRANKLIN, S., GRAESSER, A., 1996, "Is it an agent, or just a program?: A taxonomy for autonomous agents". In: *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ECAI'96)*, Heidelberg, Germany. <http://www.msci.memphis.edu/franklin/AgentProg.htm>

GENESERETH, M. R., FIKES, R., 1992, *Knowledge Interchange Format Version 3.0 Reference Manual*. Computer Science Department, Stanford University

GENESERETH, M. R., SINGH, N. P., SYED, M. A., 1994, "A distributed and anonymous knowledge sharing approach to software interoperation". In: *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*,)

GERVAIS, M., GOMEZ, J., WEISS, G., 2004, "A survey on agent-oriented software engineering researches", *Methodologies and Software Engineering for Agent Systems*, New York (NY), Kluwer

GOLDBERG, Y., SAFRAN, M., SILVERMAN, W., et al, 1992, "Active Mail: A Framework for Integrated Groupware Applications". In: *Groupware'92*, pp. 222-224

- GRAY, R., 1995, "Agent Tcl: A transportable agent system". In: *Proceedings of the CIKM Workshop on Intelligent Information Agents*, Baltimore,
- GROOVE, 2005, "Groove Site". In: www.groove.net.
- GRUDIN, J., 1994, "Computer-Supported Cooperative Work: History and Focus", *IEEE Computer*, v. 27, n. 5, pp. 19-26
- HEDBERG, S., 1996, "Agents for sale: first wave of intelligent agents go commercial", *IEEE Expert*, v. 11, n. 6, pp. 16-18
- HERTEL, G., GEISTER, S., KONRADT, U., 2005, "Managing Virtual Teams: A review of current empirical research", *Human Resource Management Review*, v. 15, n. 1, pp. 69-95
- HOFFMAN, M., HERRMANN, T., 2004, "Prospect Awareness - Envisioning the Benefits of Collaborative Work". In: <http://iundg.informatik.unidortmund.de/iug-home/people/MH/ProspectAwareness/PAhome.html>, Accessed in 2004.
- HUHNS, M., SINGH, M. P., 1998, "Agents and Multiagent Systems: Themes, Approaches and Challenges". In HUHNS, M. and SINGH, M. P., *Readings in Agents*, chapter 1, San Francisco, CA, USA, Morgan Kaufmann
- ISAACS, E. A., TANG, J. C., MORRIS, T., 1996, "PIAZZA: A desktop Environment Supporting Impromptu and Planned Interactions". In: *Conference on Computer Supported Cooperative Work (CSCW)*, Cambridge, MA,
- ISAACS, E. A., WALENDOWSKI, A., RANGANATHAN, D., 2002, "Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions", *CHI Letters*, v. 4, n. 1, pp. 179-186
- ISAACS, E. A., WHITTAKER, S., FROHLICH, D., et al, 1997, "Informal Communication Re-Examined: New Functions for Video in Supporting Opportunistic Encounters". In FINN, K., SELLEN, A., and WILBUR, S., *Video Mediated Communication*, Mahwah, NJ, Lawrence Erlbaum
- JENNINGS, N. R., 2001, "An Agent-Based Approach for Building Complex Software Systems", *Communications of the ACM*, v. 44, n. 4, pp. 35-41

KANERVA, A., KOSKINEN, L., PITKÄNIEMI, T., 2000, *Awareness Support in Virtual Collaborative Systems*. University of Helsinki

KORTUEN, G., GELLERSEN, H. W., BILLINGHURST, M., 2002, "Mobile Ad Hoc Collaboration". In: *Conference on Computer Human Interface, CHI*, Minneapolis, USA

KRAUT, R. E., EGIDO, R., GALEGHER, C., et al, 1990, "Patterns of contact and communication in scientific research collaboration", *Intellectual teamwork: social and technological foundations of cooperative work*, Mahwah, NJ, USA, Lawrence Erlbaum Associates, Inc.

KRAUT, R. E., FISH, R. S., ROOT, R., et al, 1990, "Informal Communication in Organizations: Form, Function, and Technology". In OSKAMP, I. S. and SPACAPAN, S., *People's Reactions to Technology*, Beverly Hills, CA, Sage Publications

KRAUT, R. E., FUSSELL, S. R., BRENNAN, S. E., et al, 2002, "Understanding effects of proximity on collaboration: Implications for technologies to support remote collaborative work". In HINDS, P. and KIESLER, S., *Distributed Work*, Cambridge, MA, MIT Press

LABROU, Y., FININ, T., PENG, Y., 1999, "Agent communication languages: The current landscape", *IEEE Intelligent Systems*, v. 14, n. 2, pp. 45-52

LANGE, D., 1997, "Java Aglet Application Programming Interface (J-AAPI) White Paper - Draft 2". In: <http://www.trl.ibm.com/aglets/JAAPI-whitepaper.htm>, Accessed in 28/08/2005.

LANGE, D., OSHIMA, M., 1998, *Programming and Deploying Java Mobile Agents with Aglets*. 1st, Addison-Wesley Professional

LEE, D., CHUANG, H., SEAMONS, K., 1997, "Document Ranking and the Vector-Space Model", *IEEE Software*, v. 14, n. 2, pp. 67-75

LIU, C., CONRADI, R., 1998, "Process View of CSCW". In: *ISFST98 / International Workshop on Intelligent Agents in Information and Process Management*, pp. 12-12, Bremen, Germany, September

LIU, H., SINGH, P., 2004, "ConceptNet a practical commonsense reasoning tool-kit", *BT Technology Journal*, v. 22, n. 4, pp. 211-226

LUCENA, C. J. P., FUKS, H., MILIDIÚ, R., et al, 1999, "AulaNet: Helping Teachers to Do Their Homework". In: *Multimedia Computer Techniques in Engineering Education Workshop*, Technische Universitat Graz, Graz, Austria:

LUTZ, E., RETZOW, H. K., HOERNIG, K., 1990, "MAFIA - An Active Mail-Filter-Agent for an Intelligent Document Processing Support". In S. Gibbs and A. Verrijn-Stuart, IFIP, North Holland, Elsevier Science Publishers B.V.

MANGINA, E., 2002, "Review of Software Products for Multi-agent System". In: <http://www.agentlink.org/admin/docs/2002/MAS.pdf>, Accessed in 01/02/2003.

MATSUURA, N., FUJINO, G., OKADA, K., et al, 1993, "An Approach to Encounters and Interaction in a Virtual Environment". In: *ACM Conference on Computer Science*, Indiana, United States,

MIT MEDIA LAB, 2004, "Open Mind Commonsense Web Site". In: <http://commonsense.media.mit.edu>, Accessed in 15/11/2004.

MORAN, A. L., DECOUCHANT, D., FAVELA, J., et al, 2002, "PIÑAS: Supporting a Community of Authors on the Web". In: *Fourth International Conference on Distributed Communities on the Web*, Sydney, Australia,

MORAN, A. L., FAVELA, J., MARTINEZ-ENRIQUEZ, A. M., et al, 2002, "Before Getting There: Potential and Actual Collaboration". In: *Groupware: Design, Implementation and Use, International Workshop, CRIWG*, v. LNCS 2440, pp. 147-167, La Serena, Chile, September

MORENO, M., VIVACQUA, A., DE SOUZA, J., 2003, "An Agent Framework to Support Opportunistic Collaboration". In: *Groupware: Design,*

Implementation, and Use 9th International Workshop, CRIWG 2003 Lecture Notes in Computer Science, v. 2806, pp. 224-231, Autrans, France, September

MOULIN, B., CHAIB-DRAA, B., 1996, " An Overview of Distributed Artificial Intelligence", *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons

NECHES, R., FIKES, R., FININ, T., et al, 1991, "Enabling Technology for Knowledge Sharing", *AI Magazine*, v. 3, n. 12, pp. 36-56

NWANA, H. S., 1996, "Software agents: An overview", *Knowledge Engineering Review*, v. 11, n. 3, pp. 1-40. <http://www.sce.carleton.ca/netmanage/docs/AgentsOverview/ao.html>.

NWANA, H. S., NDUMU, D., LEE, L., et al, 1999, "ZEUS: a toolkit and approach for building distributed multi-agent systems". In: *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pp. 360-361, Seattle, Washington, United States,

OLSON, G., OLSON, J., 2000, "Distance Matters", *Human-Computer Interaction*, v. 2/3, pp. 139-178

ORLIKOWSKI, W. J., 1992, "Learning from Notes: Organizational Issues in Groupware Implementation". In: *Computer Supported Cooperative Work*, pp. 352-369, Toronto, Canada,

P3P, 2005, "Platform for Privacy Preferences (P3P) Project Home Page". In: <http://www.w3.org/P3P/>, Accessed in 22/07/2005.

PINHEIRO, M. K., LIMA, J. V., BORGES, M. R. S., 2002, "A Framework for Awareness Support in Groupware Systems". In: *7th International Conference on Computer Supported Cooperative Work in Design - CSCWiD'2002*, pp. 13-18, Rio de Janeiro, Brazil,

PYTHON SOFTWARE FOUNDATION, 2005, "Python programming language". In: <http://www.python.org>, Accessed in 2004.

RHODES, B., MAES, P., 2000, "Just-in-time information retrieval agents", *IBM Systems Journal special issue on the MIT Media Laboratory*, v. 39, n. 3&4, pp. 685-704

RHODES, B., STARNER, T., 1996, "The Remembrance Agent: A continuously running automated information retrieval system". In: *Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pp. 487-495, London, UK

RICORDEL, P., DEMAZEAU, Y., 2000, "From Analysis to Deployment: A Multi-agent Platform Survey". In: *ESAW '00: Proceedings of the First International Workshop on Engineering Societies in the Agent World*, pp. 93-105, London, UK

ROOT, R., 1988, "Design a Multi-Media Vehicle for Social Browsing". In: *Conference on Computer Supported Cooperative Work (CSCW)*,

RUSSELL, S., NORVIG, P., 1995, *Artificial Intelligence - A Modern Approach*. Third, Englewood Cliffs, NJ, Prentice Hall

SALTON, G., 1998, *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley Publishing

SANTORO, F. M., BORGES, M. R. S., SANTOS, N., 2000, "Cooperation Model for Learning: A System of Patterns". In: *Proceedings of the Conference on Educational Multimedia, Hypermedia and Telecommunications*, Montreal, Canada

SARGENT, P., 1992, "Back to school for a brand new ABC", *The Guardian*, pp. 28-28

SEARLE, J. R., 1969, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge, England, Cambridge University Press

SHAKSHUKI, E., 2005, "A Methodology for Evaluating Agent Toolkits". In: *International Conference on Information Technology: Coding and Computing (ITCC'05)*, v. 1, pp. 391-396, Canada,

SHEN, W., NORRIE, D., BARTHÈS, J., 2001, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Great Britain, Taylor and Francis

- SHOHAM, Y., 1993, "Agent-oriented programming", *Artificial Intelligence*, v. 60, n. 1, pp. 51-92
- SHÜMMER, T., 2004, *The Public Privacy – Patterns for Filtering Personal Information in Collaborative Systems - Technical Report*. FernUnivesität in Hagen
- SINGH, N. P., LIN, T., MUELLER, E. T., et al, 2002, "Open mind commonsense: knowledge acquisition from the general public", 2519 ed., Heidelberg, Springer
- SOHLENKAMP, M., 1999a, *Supporting Group Awareness in Muti-User Environments through Perceptualization*
- SOHLENKAMP, M., 1999b, *Supporting Group Awareness in Muti-User Environments through Perceptualization*
- SUN, 2005, "Java Technology". In: <http://java.sun.com>, Accessed in 2003.
- TELECOM ITALIA LAB, 2005, "Java Agent DEvelopment Framework (JADE)". In: <http://jade.tilab.com/>, Accessed in 2003.
- VAN DYKE, N., LIEBERMAN, H., MAES, P., 1999, "Butterfly: A Conversation-Finding Agent for Internet Relay Chat". In: *Proceedings of the 1999 International Conference on Intelligent User Interfaces*
- VIVACQUA, A.,LIEBERMAN, H., 2000, "Agents to Assist in Finding Help". In: *Proceedings of the ACM Conference on Computers and Human Interface (CHI-2000)*, The Hague, Netherlands
- WANG, A., CONRADI, R., LIU, C., 1999, "A Multi-Agent Architecture for Cooperative Software Engineering", v. Third International Conference on Autonomous Agents
- WANG, A.,SORENSEN, C., 2003, "A Comparison of two different Java Technologies to Implement a Mobile Agent System". In: *Proceedings of the IASTED International Conference on Applied Informatics 2003 (AI'2003)*, pp. 6-, Innsbruck, Austria, February

WATSON, M., 2004, "Java KBtextmaster Natural Language Processing Toolkit". In: http://www.markwatson.com/opensource/KBtextmaster_ver_2_user_guide.pdf, Accessed in 07/04/2004.

WHITE, J. E., 1997, "Mobile agents". In BRADSHAW, J., *Software agents*, Cambridge, MA, USA, MIT Press

WOOLDRIDGE, M., JENNINGS, N. R., 1995a, "Agent Theories, Architectures, and Languages: A Survey". In WOOLDRIDGE, M. and JENNINGS, N. R., *Intelligent Agents*, Berlin, Springer-Verlag

WOOLDRIDGE, M., JENNINGS, N. R., 1995b, "Intelligent agents: Theory and practice", *The Knowledge Engineering Review*, v. 10, n. 2, pp. 115-152

ZACHARIA, G., MAES, P., 2005, "Trust management through reputation mechanisms", *Applied Artificial Intelligence Journal*, v. 14, n. 9, pp. 881-908

ZHANG, Y., TANNIRU, M., 2005, "An Agent-based Approach to Study Virtual Learning Communities". In: *Proceedings of the 38th Hawaii International Conference on System Sciences*, v. 01, pp. 11.3-, Washington, DC, USA, January

Neste anexo se apresenta a definição formal do esquema base XML da P3P (*Platform for Privacy Preferences*) que é usado para a definição dos dados pessoais no perfil do usuário. Esse também pode ser consultado na internet no site <http://www.w3.org/TR/P3P/base>.

```
- <DATASchema xmlns="http://www.w3.org/2002/01/P3Pv1">
- <!-- ***** Base Data Structures ***** -->
- <!-- "date" Data Structure -->
<DATA-STRUCT name="date.ymd.year" short-description="Year" />
<DATA-STRUCT name="date.ymd.month" short-description="Month" />
<DATA-STRUCT name="date.ymd.day" short-description="Day" />
<DATA-STRUCT name="date.hms.hour" short-description="Hour" />
<DATA-STRUCT name="date.hms.minute" short-description="Minute" />
<DATA-STRUCT name="date.hms.second" short-description="Second" />
<DATA-STRUCT name="date.fractionsecond" short-description="Fraction of Second" />
<DATA-STRUCT name="date.timezone" short-description="Time Zone" />
- <!-- "login" Data Structure -->
- <DATA-STRUCT name="login.id" short-description="Login ID">
- <CATEGORIES>
<uniqueid />
</CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="login.password" short-description="Login Password">
- <CATEGORIES>
<uniqueid />
</CATEGORIES>
</DATA-STRUCT>
- <!-- "personname" Data Structure -->
- <DATA-STRUCT name="personname.prefix" short-description="Name Prefix">
- <CATEGORIES>
<demographic />
</CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="personname.given" short-description="Given Name (First Name)">
- <CATEGORIES>
<physical />
</CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="personname.middle" short-description="Middle Name">
- <CATEGORIES>
<physical />
</CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="personname.family" short-description="Family Name (Last Name)">
- <CATEGORIES>
<physical />
</CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="personname.suffix" short-description="Name Suffix">
- <CATEGORIES>
<demographic />
</CATEGORIES>
```

```

    </DATA-STRUCT>
- <DATA-STRUCT name="personname.nickname" short-description="Nickname">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- "certificate" Data Structure -->
- <DATA-STRUCT name="certificate.key" short-description="Certificate key">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="certificate.format" short-description="Certificate format">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- "telephonenumber" Data Structure -->
- <DATA-STRUCT name="telephonenumber.intcode" short-description="International Telephone Code">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="telephonenumber.loccode" short-description="Local Telephone Area Code">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="telephonenumber.number" short-description="Telephone Number">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="telephonenumber.ext" short-description="Telephone Extension">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="telephonenumber.comment" short-description="Telephone Optional
  Comments">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- "postal" Data Structure -->
  <DATA-STRUCT name="postal.name" structref="#personname" />
- <DATA-STRUCT name="postal.street" short-description="Street Address">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="postal.city" short-description="City">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="postal.stateprov" short-description="State or Province">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="postal.postalcode" short-description="Postal Code">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>

```

```

    </DATA-STRUCT>
- <DATA-STRUCT name="postal.organization" short-description="Organization Name">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="postal.country" short-description="Country Name">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
  </DATA-STRUCT>
- <!-- "telecom" Data Structure -->
- <DATA-STRUCT name="telecom.telephone" short-description="Telephone Number"
  structref="#telephonenumber">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="telecom.fax" short-description="Fax Number" structref="#telephonenumber">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="telecom.mobile" short-description="Mobile Telephone Number"
  structref="#telephonenumber">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="telecom.pager" short-description="Pager Number"
  structref="#telephonenumber">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
  </DATA-STRUCT>
- <!-- "online" Data Structure -->
- <DATA-STRUCT name="online.email" short-description="Email Address">
- <CATEGORIES>
  <online />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="online.uri" short-description="Home Page Address">
- <CATEGORIES>
  <online />
  </CATEGORIES>
  </DATA-STRUCT>
- <!-- "contact" Data Structure -->
  <DATA-STRUCT name="contact.postal" short-description="Postal Address Information"
    structref="#postal" />
- <DATA-STRUCT name="contact.telecom" short-description="Telecommunications Information"
  structref="#telecom">
- <CATEGORIES>
  <physical />
  </CATEGORIES>
  </DATA-STRUCT>
- <DATA-STRUCT name="contact.online" short-description="Online Address Information"
  structref="#online">
- <CATEGORIES>
  <online />
  </CATEGORIES>
  </DATA-STRUCT>
- <!-- "uri" Data Structure -->
  <DATA-STRUCT name="uri.authority" short-description="URI Authority" />
  <DATA-STRUCT name="uri.stem" short-description="URI Stem" />
  <DATA-STRUCT name="uri.querystring" short-description="Query-string Portion of URI" />

```

```

- <!-- "ipaddr" Data Structure -->
- <DATA-STRUCT name="ipaddr.hostname" short-description="Complete Host and Domain Name">
- <CATEGORIES>
  <computer />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="ipaddr.partialhostname" short-description="Partial Hostname">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="ipaddr.fullip" short-description="Full IP Address">
- <CATEGORIES>
  <computer />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="ipaddr.partialip" short-description="Partial IP Address">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- "loginfo" Data Structure -->
- <DATA-STRUCT name="loginfo.uri" short-description="URI of Requested Resource"
  structref="#uri">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="loginfo.timestamp" short-description="Request Timestamp"
  structref="#date">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="loginfo.clientip" short-description="Client's IP Address or Hostname"
  structref="#ipaddr" />
- <DATA-STRUCT name="loginfo.other.httpmethod" short-description="HTTP Request Method">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="loginfo.other.bytes" short-description="Data Bytes in Response">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="loginfo.other.statuscode" short-description="Response Status Code">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- "httpinfo" Data Structure -->
- <DATA-STRUCT name="httpinfo.referrer" short-description="Last URI Requested by the User"
  structref="#uri">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-STRUCT>
- <DATA-STRUCT name="httpinfo.useragent" short-description="User Agent Information">
- <CATEGORIES>
  <computer />
  </CATEGORIES>
</DATA-STRUCT>
- <!-- ***** Base Data Schemas ***** -->
- <!-- "dynamic" Data Schema -->

```



```

- <DATA-DEF name="dynamic.clickstream" short-description="Click-stream Information"
  structref="#loginfo">
- <CATEGORIES>
  <navigation />
  <computer />
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="dynamic.http" short-description="HTTP Protocol Information"
  structref="#httpinfo">
- <CATEGORIES>
  <navigation />
  <computer />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="dynamic.clientevents" short-description="User's Interaction with a Resource">
- <CATEGORIES>
  <navigation />
  </CATEGORIES>
</DATA-DEF>
  <DATA-DEF name="dynamic.cookies" short-description="Use of HTTP Cookies" />
- <DATA-DEF name="dynamic.searchtext" short-description="Search Terms">
- <CATEGORIES>
  <interactive />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="dynamic.interactionrecord" short-description="Server Stores the Transaction
  History">
- <CATEGORIES>
  <interactive />
  </CATEGORIES>
</DATA-DEF>
  <DATA-DEF name="dynamic.miscdata" short-description="Miscellaneous Non-base Data Schema =
  information" />
  - <!-- "user" Data Schema -->
- <DATA-DEF name="user.name" short-description="User's Name" structref="#personname">
- <CATEGORIES>
  <physical />
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.bdate" short-description="User's Birth Date" structref="#date">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.login" short-description="User's Login Information" structref="#login">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.cert" short-description="User's Identity Certificate"
  structref="#certificate">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.gender" short-description="User's Gender">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.jobtitle" short-description="User's Job Title">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>

```

```

    </DATA-DEF>
- <DATA-DEF name="user.home-info" short-description="User's Home Contact Information"
  structref="#contact">
- <CATEGORIES>
  <physical />
  <online />
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.business-info" short-description="User's Business Contact Information"
  structref="#contact">
- <CATEGORIES>
  <physical />
  <online />
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.employer" short-description="Name of User's Employer">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="user.department" short-description="Department or Division of Organization
  where User is Employed">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <!-- "thirdparty" Data Schema -->
- <DATA-DEF name="thirdparty.name" short-description="Third Party's Name"
  structref="#personname">
- <CATEGORIES>
  <physical />
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.bdate" short-description="Third Party's Birth Date"
  structref="#date">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.login" short-description="Third Party's Login Information"
  structref="#login">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.cert" short-description="Third Party's Identity Certificate"
  structref="#certificate">
- <CATEGORIES>
  <uniqueid />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.gender" short-description="Third Party's Gender">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.jobtitle" short-description="Third Party's Job Title">
- <CATEGORIES>
  <demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.home-info" short-description="Third Party's Home Contact
  Information" structref="#contact">
- <CATEGORIES>

```

```

<physical />
<online />
<demographic />
  </CATEGORIES>
</DATA-DEF>
- <DATA-DEF name="thirdparty.business-info" short-description="Third Party's Business Contact
  Information" structref="#contact">
- <CATEGORIES>
  <physical />
  <online />
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
- <DATA-DEF name="thirdparty.employer" short-description="Name of Third Party's Employer">
- <CATEGORIES>
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
- <DATA-DEF name="thirdparty.department" short-description="Department or Division of
  Organization where Third Party is Employed">
- <CATEGORIES>
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
- <!-- "business" Data Schema -->
- <DATA-DEF name="business.name" short-description="Organization Name">
- <CATEGORIES>
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
- <DATA-DEF name="business.department" short-description="Department or Division of
  Organization">
- <CATEGORIES>
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
- <DATA-DEF name="business.cert" short-description="Organization Identity certificate"
  structref="#certificate">
- <CATEGORIES>
  <uniqueid />
    </CATEGORIES>
  </DATA-DEF>
- <DATA-DEF name="business.contact-info" short-description="Contact Information for the
  Organization" structref="#contact">
- <CATEGORIES>
  <physical />
  <online />
  <demographic />
    </CATEGORIES>
  </DATA-DEF>
</DATASchema>

```