

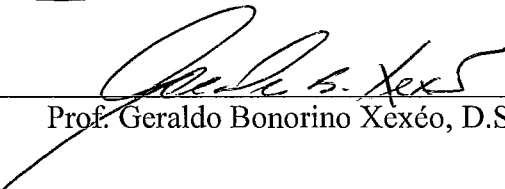
CUMBIA: UMA ARQUITETURA BASEADA EM AGENTES PARA DETECTAR  
OPORTUNIDADES DE COLABORAÇÃO

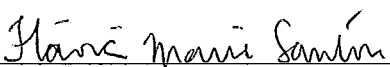
Melfry M Moreno Molina

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.<sup>a</sup>. 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	<b>INTERAÇÃO FACE-A-FACE</b>	<b>INTERAÇÃO SÍNCRONA DISTRIBUÍDA</b>
MOMENTO DIFERENTE	<b>INTERAÇÃO ASSÍNCRONA</b>	<b>INTERAÇÃO ASSÍNCRONA DISTRIBUÍDA</b>

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).

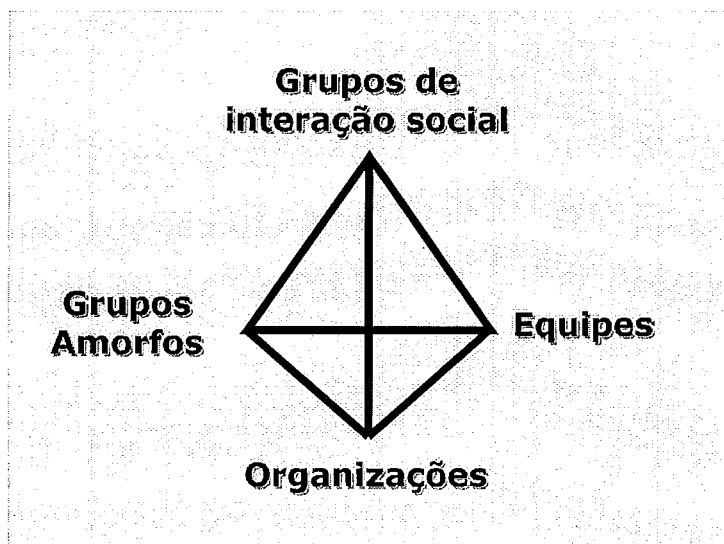


Figura 2 - Espectro dos grupos (ELLIS e WAINER, 1998)

### 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.

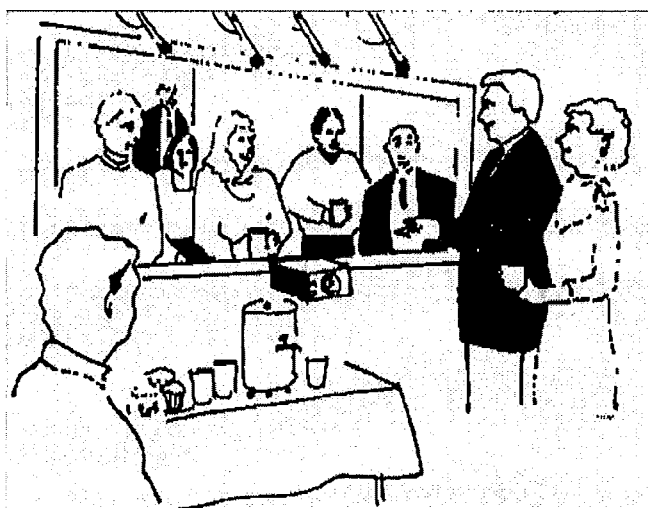


Figura 3 – Sistema de Teleconferência VideoWindow (FISH, KRAUT et al., 1990).

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

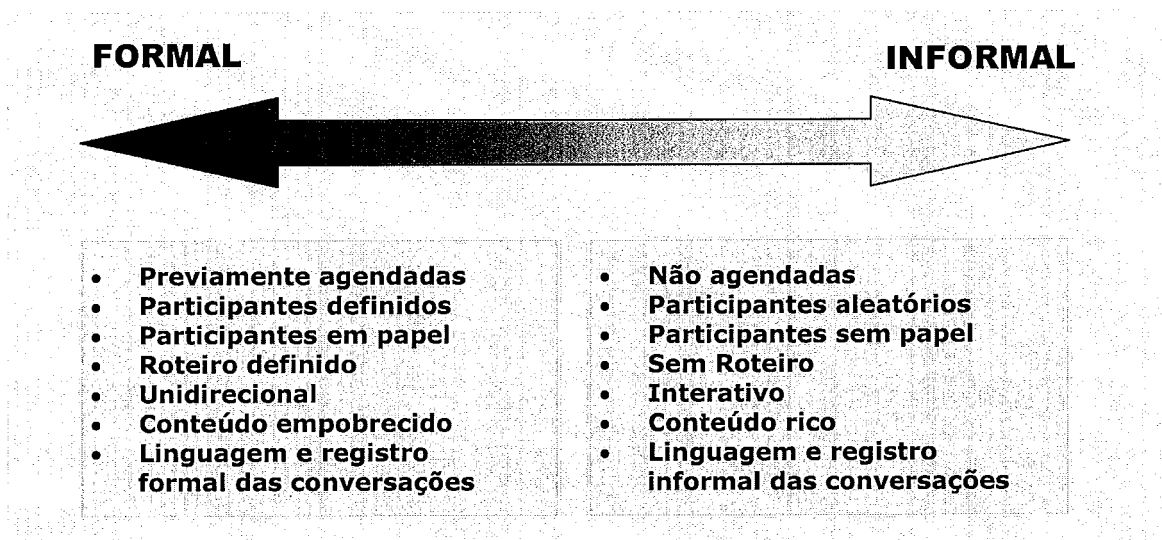


Figura 4 - Dimensão da formalidade na comunicação (KRAUT, FISH et al., 1990).

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.

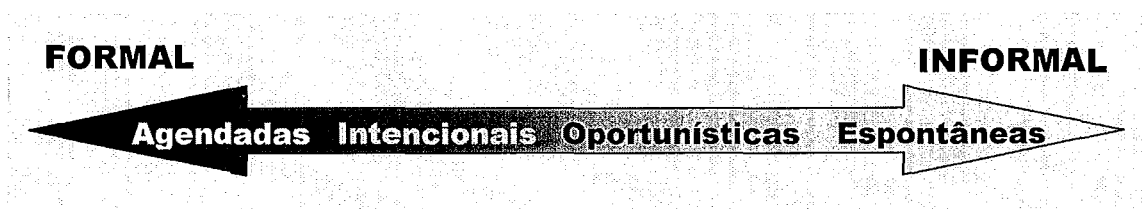


Figura 5 - Tipos de comunicação segundo (KRAUT, FISH et al., 1990).

(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.

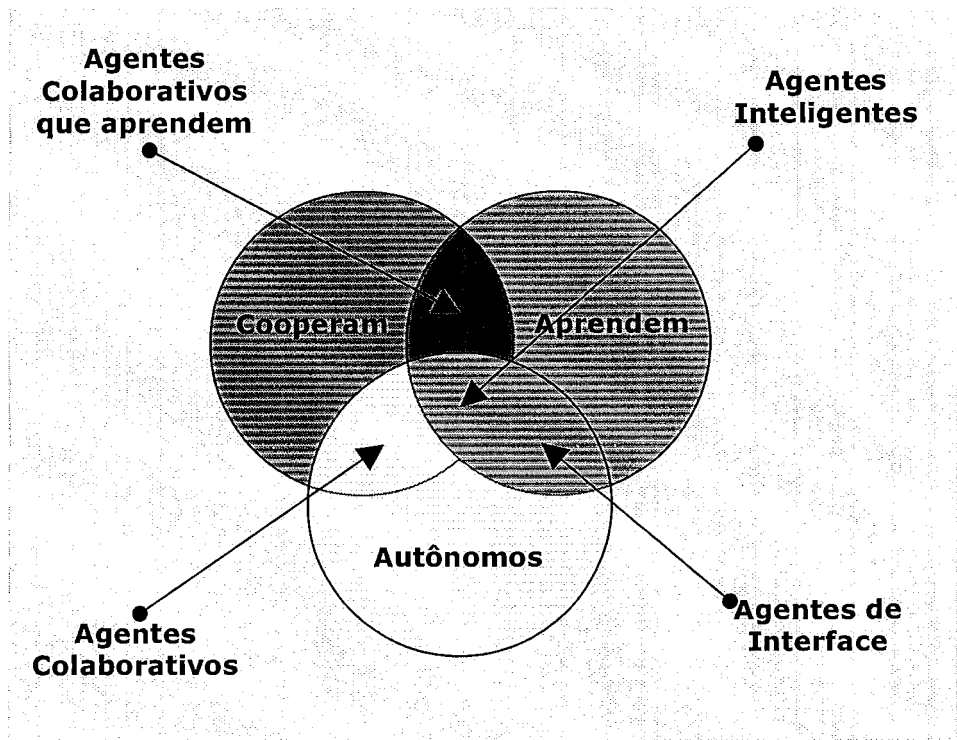


Figura 6 - Tipologia de agentes proposta por (NWANA, 1996).

(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.

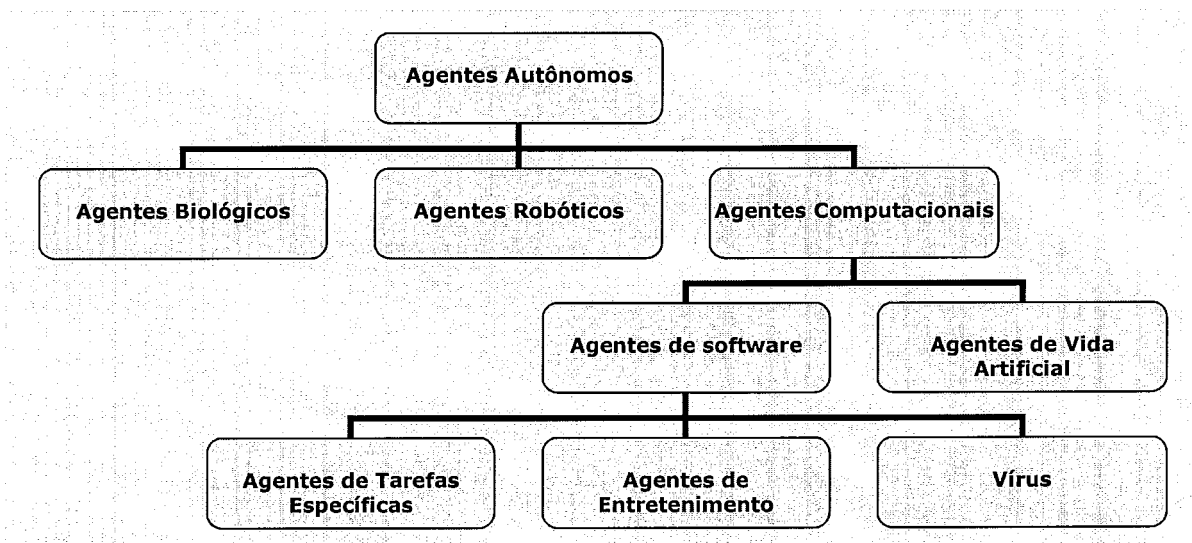


Figura 7 - Taxonomia de agentes proposta por (FRANKLIN e GRAESSER, 1996).

### 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.













































































































