

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção

UM MODELO BASEADO EM AGENTES PARA
SUORTE A AUTOMAÇÃO DE SERVIÇOS

Luciano José Savio

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia de Produção da
Universidade Federal de Santa Catarina
como requisito parcial para obtenção do título
de Mestre em Engenharia de Produção

Florianópolis

2000

Luciano José Savio

**Um Modelo baseado em Agentes para
Suporte a Automação de Serviços**

Esta dissertação foi julgada e aprovada para a
obtenção do título de **Mestre em Engenharia de
Produção** no **Programa de Pós-Graduação em
Engenharia de Produção** da
Universidade Federal de Santa Catarina

Florianópolis, 28 de Agosto de 2000.

Prof. Ricardo Miranda Barcia, Ph D.

Coordenador do Curso

BANCA EXAMINADORA

Prof. Oscar Ciro Lopez Vaca, Dr.
Orientador

Prof. Aran Morales, Dr.

Prof. Rafael Ávila Faraco, Msc.

Prof Ricardo Villares Davalos, Dr.

Agradecimentos

Agradeço primeiramente a Deus, amparando-me em todos os momentos e sempre sendo fonte de inspiração divina em todas as etapas desta caminhada.

Meus Pais, que com muita dedicação e firmeza me auxiliaram a descobrir o mundo e a usufruir das suas maravilhas.

A minha esposa Sílvia, pela compreensão e incentivo para esta realização.

À ACAFE, UNISUL e UFSC, instituições cujo apoio foram fundamentais para o desenvolvimento deste trabalho.

Aos professores do curso de Pós-Graduação em Engenharia de Produção e, particular agradecimento ao Prof. Dr. Oscar Ciro Lopez, pela dedicação na orientação e condução deste trabalho aos seus objetivos.

Aos colegas de trabalho da Assessoria de Tecnologia da Informação, pelo apoio, solidariedade e auxílio em muitos momentos tão necessário.

Aos membros da banca examinadora, Prof. Dr. Aran Morales, Prof. Msc. Rafael Ávila Faraco e Prof. Dr. Ricardo Villares Davalos, pela avaliação e contribuição na melhoria deste trabalho.

Ao acadêmico Diego Pablo Hidalgo, por sua contribuição e dedicação na codificação do protótipo.

Sumário

Lista de Figuras.....	p.vi
Lista de Tabelas.....	p.vii
Resumo.....	p.viii
Abstract.....	p.ix
1 Introdução.....	p.1
1.1 Objetivo.....	p.2
1.2 Justificativa.....	p.2
1.3 Estrutura do Trabalho.....	p.3
2 Teoria de Agentes.....	p.5
2.1 Introdução.....	p.5
2.2 Histórico.....	p.6
2.2.1 Inteligência Artificial.....	p.6
2.2.2 Inteligência Artificial Distribuída.....	p.7
2.3 Definição de Agentes.....	p.9
2.4 Tipologia de Agentes.....	p.17
2.5 Áreas de Atuação de Agentes.....	p.22
2.6 Comunicação entre Agentes.....	p.25
2.6.1 Linguagens de Comunicação entre Agentes.....	p.27
3 Gerenciamento Indireto.....	p.30
3.1 Introdução.....	p.30
3.2 Execução e Gerenciamento de Tarefas.....	p.32
3.3 Manipulação Direta.....	p.33
3.4 Gerenciamento Indireto.....	p.36
3.5 Competência e Confiabilidade.....	p.40
3.6 Conclusão.....	p.43
4 Modelo Proposto.....	p.45
4.1 Apresentação do Modelo.....	p.46
4.2 Características dos Softwares Agentes empregadas no Modelo.....	p.48
4.3 Arquitetura do Modelo Proposto.....	p.49
4.4 Tipologia de Agentes utilizada no modelo.....	p.50
4.5 Módulos e Elementos Componentes.....	p.51
4.6 Organização da Sociedade de Agentes no Modelo.....	p.54
4.6.1 Resolução Distribuída de Problemas.....	p.55
4.6.2 Sistemas Multi-Agentes.....	p.56
4.7 Comunicação entre os Agentes do Modelo.....	p.59
4.8 Conclusão.....	p.61

5 Implementação do Modelo	p.63
5.1 Serviços Contemplados Pelo Protótipo.....	p.64
5.2 Prestação dos Serviços.....	p.65
5.3 Execução dos Serviços Implementados.....	p.70
5.4 Resultados Obtidos.....	p.78
6 Conclusões e Recomendações	p.85
7 Referências Bibliográficas	p.87
8 Anexos	p.92
8.1 Resumo das Mensagens KQML.....	p.92
8.2 Ferramentas Utilizadas na Implementação do Protótipo.....	p.94
8.3 Regras de Produção.....	p.105
8.4 Arquitetura de Processamento Baseada em Três Camadas.....	p.107

Lista de Figuras

Figura 1: Modelo de Agentes segundo Gaglayan e Harrison.....	p.16
Figura 2: Classificação Segundo Nwana [Nwana,1996].....	p.18
Figura 3: Manipulação direta de um objeto – [Apple,1996].....	p.34
Figura 4: Manipulação direta vs. Gerenciamento Indireto.....	p.38
Figura 5: Agente Assistente – [Maes,1997].....	p.39
Figura 6: Agentes Assistentes e aquisição de conhecimento – [Maes,1997].....	p.43
Figura 7: Arquitetura Geral do Modelo.....	p.49
Figura 8: Módulos e Elementos do Modelo.....	p.51
Figura 9: Resolução Distribuída de Problemas.....	p.56
Figura 10: Sistemas Multi-Agentes.....	p.57
Figura 11: Comunicação entre os elementos do modelo.....	p.59
Figura 12: Interface do Usuário.....	p.66
Figura 13: Agentes Coordenadores e Base de Conhecimento.....	p.67
Figura 14: Agentes executores.....	p.68
Figura 15: Agentes de busca.....	p.68
Figura 16: Agentes de Interface Responsável e Base de Conhecimento.....	p.69
Figura 17: Sociedade de Agentes e Roteador AMR.....	p.69
Figura 18: Interface Principal do Protótipo.....	p.70
Figura 19: Emissão de Histórico.....	p.71
Figura 20: Documentos Pendentes.....	p.73
Figura 21: Solicitação de Notas Semestre.....	p.74
Figura 22: Apresentação de Notas.....	p.75
Figura 23: Interface para Troca de Curso.....	p.76
Figura 24: Pedido de Transferência de Curso.....	p.77
Figura 25: Fluxograma para Solicitação de Documentos.....	p.79
Figura 26: Fluxograma para Entrada de Processos.....	p.80
Figura 27: Fluxograma para Solicitação de Informações Acadêmicas.....	p.80
Figura 28: Agent Message Router.....	p.98
Figura 29: Níveis de Especialização do JATLite.....	p.97
Figura 30: Arquitetura Cliente/Servidor em duas camadas – Two Tier.....	p.107
Figura 31: Arquitetura cliente/servidor – Three Tier.....	p.109

Lista de Tabelas

Tabela 1 – Tempos para a prestação de Serviços.....	p.81
Tabela 2 – Ocorrência de Falhas na Execução dos Serviço.....	p.82
Tabela 3 – Ocorrência de Atrasos na Prestação dos Serviço.....	p.83
Tabela 4 - Definição sintática da KQML.....	p.101

Resumo

A automação das atividades realizadas pelo homem é uma constante em nossos dias, visando aprimorar a maneira de como estas atividades são executadas.

Objetivando auxiliar neste processo de automação, através do uso da tecnologia de agentes, do gerenciamento indireto e de técnicas de inteligência artificial, esse trabalho propõe uma arquitetura de agentes especializados aliado a aplicação da teoria do gerenciamento indireto, para suporte a automação de serviços. Nesta arquitetura será possível identificar os serviços a serem prestados, bem como a maneira como serão prestados, se utilizando de toda a gama de recursos proporcionada pela sociedade de agentes presente neste modelo.

Esta arquitetura contempla principalmente 4 elementos: Base de conhecimentos, arquitetura de quadro-negro, bases de dados e uma sociedade de agentes. Esta sociedade de agentes através de seus agentes especializados na busca de informações, interfaces, coordenação e execução de tarefas se utilizam dos demais elementos da arquitetura para prover os serviços da forma mais automatizada possível.

Esta arquitetura propõe ainda a utilização de uma arquitetura de execução distribuída em três camadas, proporcionando maior flexibilidade e utilização dos recursos para a execução dos serviços.

Neste trabalho, a arquitetura foi aplicada tendo como finalidade o gerenciamento automatizado das solicitações de serviços para uma secretaria de ensino de nível superior, tendo em vista a carência e a crescente demanda por automação dos processos ao atendimento de seus clientes, obtendo ótimos resultados.

Palavras-chave: Agentes, Automação, Gerenciamento Indireto, Inteligência Artificial

1. INTRODUÇÃO

A busca pela melhoria na qualidade dos serviços prestados pelos diversos setores da sociedade é uma constante em nosso cotidiano, objetivando, num âmbito global, a melhoria da sociedade como um todo.

A obtenção da qualidade nos serviços oferecidos implica, em muitos casos, na utilização de tecnologias modernas que venham a proporcionar uma maior agilidade e eficiência, bem como maior comodidade e segurança aos usuários dos serviços.

Gradativamente, à medida que novas tecnologias vão sendo criadas e disponibilizadas, esses serviços vão se adaptando às mesmas e obtendo um maior nível de automação, como exemplo concreto, cabe citar as tecnologias na área de negócios e comércio eletrônicos, denominados *e-commerce* e *e-business*, fazendo parte de uma tendência denominada *e-services* ou serviços eletrônicos.

O emprego de uma nova tecnologia pode representar profundas modificações na maneira como o serviço é prestado, implicando em novas atitudes para a execução do serviço.

Uma das tecnologias disponíveis e que está em evidência é a tecnologia de agentes. Essa tecnologia proporciona a construção de sistemas com características de autonomia, proatividade, reatividade, dentre outras, características essas que, aliadas a características de outras tecnologias, permitem elevar o nível de automação na prestação de serviços.

1.1 Objetivo

O presente trabalho se propõe a desenvolver um modelo de arquitetura de software que permita a automação da prestação de serviços de forma a minimizar a interferência do homem durante as diversas etapas do processo.

Os objetivos específicos do trabalho são:

- Analisar e demonstrar a viabilidade da integração de sociedades de agentes com as características da teoria do gerenciamento indireto.
- Analisar o emprego de agentes e gerenciamento indireto na automação de serviços.
- Propor uma arquitetura de agentes especializados com gerenciamento indireto para dar suporte a automação de serviços.
- Implementar computacionalmente o modelo proposto.
- Demonstrar a viabilidade do modelo da arquitetura proposta usando como exemplo a automação de prestação de serviços de uma secretaria geral de ensino, numa instituição de nível superior.

1.2 Justificativa

Os benefícios e facilidades oferecidos pelas grandes redes de computadores permitem uma comunicação e acesso a informações muito mais rápidos, dinamizando o cotidiano dos usuários. Esta nova dinâmica atua diretamente no comportamento das pessoas, direcionando-as cada vez mais para a utilização de serviços automatizados, proporcionando-lhes comodidade e rapidez na prestação dos serviços.

Por outro lado, as empresas fornecedoras de serviços buscam se posicionar diante do

mercado globalizado, tendo como requisitos fundamentais a competência, a qualidade e a agilidade, exigidas por esse mercado, encontrando, na automação de processos, uma aliada na obtenção de suas metas.

Os serviços eletrônicos começam a se difundir por todas as áreas onde possam atuar, permitindo a construção de verdadeiras organizações virtuais.

O estudo da interação entre homem e máquina, área de estudos denominada HCI (Human Computer Interaction) e, em particular, a área do gerenciamento indireto contribuem em muito para o sucesso deste estilo na prestação de serviços, uma vez que disponibilizam técnicas que quebram as barreiras encontradas na interação do usuário com os sistemas automatizados.

A utilização de sistemas baseados em agentes tem contribuído com características importantes necessárias à aplicação de sistemas automatizados, tais como autonomia e pró-atividade que, aliadas a teoria do gerenciamento indireto e às facilidades disponibilizadas pelas grandes redes de computadores, tem permitido a construção de sistemas com a flexibilidade, agilidade e autonomia necessárias à execução de serviços havendo pouca ou nenhuma intervenção do homem no processo.

A flexibilidade, agilidade e autonomia são um diferencial muito importante para sistemas desta natureza, onde a automação deve ser implementada de forma a prestar o serviço da maneira mais transparente possível para o cliente, evitando-se dificuldades de interação ou no fornecimento de informações a respeito do andamento da prestação do serviço.

1.3 Estrutura do trabalho

O presente capítulo é de caráter introdutório ao tema da pesquisa, apresentando a motivação e os objetivos que levaram a realização da mesma.

O segundo capítulo apresenta a teoria dos agentes, enfatizando seus conceitos, arquitetura e áreas de atuação mais evidentes na atualidade, bem como uma explanação do mecanismo de comunicação necessário aos sistemas agentes, considerando sua importância

fundamental neste tipo de aplicação.

O terceiro capítulo é tema da área de interação entre homem e máquina, denominada Gerenciamento Indireto, onde procura situar o papel desempenhado por essa área no que tange a automação na prestação de serviços.

No quarto capítulo, é apresentada uma proposta para automação de serviços através de um modelo baseado numa arquitetura de agentes especializados e na teoria do gerenciamento indireto, visando viabilizar a prestação de serviços de maneira autônoma.

O quinto capítulo apresenta uma aplicação construída sob o embasamento do modelo proposto no capítulo anterior, automatizando a prestação de serviços numa secretaria geral de ensino de nível superior, onde estão apresentados ainda os resultados obtidos com a aplicação do modelo.

Ao sexto capítulo cabem as conclusões finais da presente pesquisa, apresentando, ainda, possíveis melhorias ao modelo proposto.

Por fim, está arrolada a bibliografia utilizada para a realização dessa pesquisa, bem como anexos que fazem parte deste trabalho, compreendendo os capítulos sétimo e oitavo, respectivamente.

2. TEORIA DE AGENTES

2.1 Introdução

Observando a literatura da área da ciência da computação dos últimos anos, constata-se a presença de um novo elemento, precursor de uma nova tecnologia no ambiente da tecnologia da informação cada vez mais presente, mais discutido e, nem por isso, mais bem definido: os agentes.

Nas primeiras décadas os computadores vinham sendo utilizados pelos usuários em geral como um meio para conseguir maior produtividade em seus afazeres, porém esse ganho somente se realiza com o aprendizado de novas ferramentas e na adaptação do uso de novas tecnologias, necessitando um maior esforço do usuário devido ao ganho de produtividade, um nível maior de conhecimento, passando a existir então um grau de aprendizagem.

A tecnologia dos agentes representa um grande impacto no modo como o usuário vem a utilizar o computador, adicionando novas características no uso e vindo a auxiliar no desempenho de suas tarefas, podendo reduzir substancialmente o grau de aprendizagem necessário, aproximando ainda mais o usuário do computador através de facilidades e novos atrativos criados através desta tecnologia.

Essa tecnologia vem ganhando espaço no âmbito da pesquisa, reunindo grupos multidisciplinares objetivando construir novas aplicações envolvendo agentes que venham a auxiliar, direta ou indiretamente os usuários de computadores.

Estas facilidades normalmente se baseiam em conceitos advindos da inteligência artificial e da inteligência artificial distribuída, tendo em vista as suas características.

2.2 Histórico

Os softwares agentes podem ser considerados como uma evolução dos Sistemas

Multi-agentes que, juntamente com a Resolução Distribuída de Problemas e a Inteligência Artificial Paralela se fundem na Inteligência Artificial Distribuída.

Os pesquisadores em Inteligência Artificial sempre almejavam a construção de sistemas onde pudesse ser tomada alguma conduta inteligente. Na década de 50, John McCarthy e Oliver Selfridge imaginaram a construção de um sistema que habitaria e realizaria suas operações num ambiente virtual. Este sistema consistia de uma meta e, para atingí-la existiam uma série de ações a serem executadas.

2.2.1 Inteligência Artificial

Pesquisas envolvendo a Inteligência Artificial são realizadas a mais de 30 anos, tendo seu início formal por volta dos anos 50, nos Estados Unidos. Entretanto a Inteligência Artificial derivou de dois temas de estudo muito enfatizadas nos anos 30 e 40, a lógica matemática e a computação.

A medida em que se traçava uma relação inicial entre lógica e inteligência, a relação da inteligência com a computação também se afirmava, sendo formalizada através de modelos numéricos simples, onde a computação destes modelos se comportava de maneira inteligente para a época.

Minski, na década de 50 definiu a Inteligência Artificial como sendo “*a ciência de levar as máquinas a realizarem algo que exigiria inteligência caso fossem realizadas por humanos*”.

Flávio Oliveira [OLIVEIRA,1996], apresenta uma definição para a Inteligência Artificial proporcionando uma visão mais prática da área:

“Inteligência Artificial é o estudo da construção de sistemas computacionais com base em algum paradigma de inteligência”[OLIVEIRA,1996].

A Inteligência Artificial baseia seus fundamentos no comportamento humano individual, onde o ponto de apoio é a representação do conhecimento e métodos de inferência.

Segundo Charniak [CHARNIAK,1985], a inteligência artificial é o ramo da computação que estuda as faculdades mentais pelo uso de modelos computacionais.

2.2.2 Inteligência Artificial Distribuída

Segundo Bond e Gasser, a Inteligência Artificial Distribuída pode ser definida sob o seguinte aspecto:

“Inteligência Artificial Distribuída é a sub-área da Inteligência Artificial concernente à concorrência em computações de Inteligência Artificial, em diversos níveis” [BOND,1988]

Flávio Oliveira [OLIVEIRA,1996], lança uma crítica a esta definição, sob o argumento de que a Inteligência Artificial Distribuída não se limita a implementações em plataformas distribuídas de sistemas tradicionais da Inteligência Artificial, mas seu foco principal enfatiza a distribuição dinâmica do controle e ações, sob o aspecto da interação e cooperação entre as entidades.

A Inteligência Artificial Distribuída baseia seus fundamentos no comportamento social, enfatizando as cooperações, interações e as negociações de troca de conhecimento entre diversos elementos do sistema.

Vários foram os motivos pelos quais a Inteligência Artificial Distribuída ganhou atenção especial assim que a teoria dos agentes se firmou no meio científico, dentre eles cabe citar a necessidade de estender as relações homem-máquina, baseando-se em métodos distribuídos de trabalho, a melhor adequação no tratamento de problemas que tratam de conhecimento distribuído e também a oportunidade de conseguir uma representação do conhecimento sob uma perspectiva mais realista, permitindo um modelo do conhecimento mais claro.

Bond e Gasser [BOND,1998], argumentam também como fatores que levaram a impulsionar a Inteligência Artificial Distribuída, o crescimento do poder computacional com o uso de novas tecnologias na construção dos equipamentos, a disseminação dos sistemas de redes de computadores, a descentralização dos centros computacionais e a introdução do trabalho em grupo na resolução de problemas.

A Inteligência Artificial Distribuída subdivide-se em duas sub-áreas:

- Resolução Distribuída de Problemas – DPS (Distributed Problem Solving)
- Sistemas Multi-Agentes – MAS (Multi-Agent Systems)

Tal classificação não pretende distinguir os trabalhos segundo a tecnologia empregada, mas na maneira como se dá a resolução dos problemas, pois tanto os sistemas multi-agentes como a resolução distribuída de problemas podem comportar vários agentes interagindo em conjunto.

Já nos primórdios das investigações em Inteligência Artificial Distribuída, nos anos 70, foi proposto o paradigma dos atores [HEWITT,1977]. Tal modelo conceitualizava um objeto auto-suficiente, interativo e

concorrente, tendo estados internos que respondiam as mensagens de outros objetos que se comunicassem sob um mesmo padrão.

O desenvolvimento das pesquisas na área de agentes se subdividem em dois grupos principais. Segundo Nwana [NWANA,1996], o primeiro grupo se inicia em 1977, tendo os agentes como entidades vivas e deliberativas, tendo um estado interno simbólico, sendo definidos como um modelo simbólico do mundo, representado de maneira explícita, onde as decisões e ações são tomadas mediante a razão simbólica. O segundo grupo surge a partir de 1990, promovendo a diversificação das classes de agentes, resultando uma evolução na sua autonomia e inteligência. Segundo Jeffrey Bradshaw [BRADSHAW,1997], há uma evolução da deliberação para a execução e do raciocínio para a ação remota. Alguns pesquisadores afirmam que esta linha de pesquisa surge pois atualmente todos chamam de agente a praticamente qualquer tipo de implementação com um mínimo de proximidade com a área.

A visão de M. Knapik [KNAPIK e JOHNSON,1998], é a de que os agentes se apresentem cada vez mais e em maior número nos sistemas de computação e de comunicação, devido a sua versatilidade em atuar em áreas como comércio eletrônico, interfaces inteligentes, sistemas de busca de informações, dentre outras.

2.3 Definição de Agentes

Para muitos pesquisadores, a definição do que seja um agente não é menos difícil de ser alcançada do que a definição do que seja inteligência, tendo em vista a inexistência de um consenso para o tema que seja aclamado no âmbito geral dos pesquisadores.

Gilber [IBM,1995] tem uma plausível definição de agente inteligente:

“Inteligência é o grau de raciocínio e aprendizado: a habilidade do agente em aceitar os objetivos e realizar as tarefas delegadas para atingi-los. No mínimo devem existir algumas preferências, talvez em forma de regras com uma máquina de inferência ou outro mecanismo de raciocínio que haja sobre estas preferências. Níveis mais altos de inteligência incluem um modelo ou outra forma de compreensão e raciocínio sobre o que o usuário deseja realizar, planejando o significado do seu objetivo. Mais ainda na escala da inteligência estão os sistemas que aprendem e se adaptam ao ambiente em que estão inseridos, de acordo com seus objetivos, e de acordo com os recursos disponíveis ao agente. Esse sistema pode, como um assistente humano, descobrir novos relacionamentos, conexões, ou conceitos independentemente do ser humano, e explorá-los, antecipando e satisfazendo as necessidades do usuário”.

H. Bjorn em seu trabalho “Intelligent Software in the Internet” [BJÖRN HERMANS,1996], conceitua agentes em dois níveis de funcionalidade. Num primeiro grupo, estão presentes os agentes que possuem as seguintes propriedades:

Autonomia

A autonomia diz respeito a característica que o agente deve possuir para poder agir por si só, visando atingir suas metas de acordo com seu estado interno.

É a habilidade que torna o agente capaz de exercer o controle sobre suas próprias ações, sem a interferência de outrém [FRANKLIN & GRAESSER, 1996].

Segundo Castelfranchi [CASTELFRANCHI,1995], os agentes devem atuar sem nenhum tipo de intervenção direta de outros humanos ou até mesmo de agentes, devendo ter ainda algum tipo de controle sobre suas ações e seu estado interno.

A pró-atividade torna-se uma característica importante para a autonomia, visto que esta proporciona ao agente a tomar iniciativas próprias, não agindo mais apenas segundo mudanças no ambiente em que está envolvido [WOOLDRIDGE & JENNINGS, 1994].

Segundo Lenny Foner [FONER,1993], os agentes que apresentarem capacidade de ação periódica, execução espontânea e iniciativa, levam o agente a realizar ações independentes, proporcionando ao agente um alto grau de autonomia tornando suas agendas independentes das de seus usuários.

Habilidade Social

A habilidade social representa a capacidade que o agente deve possuir para que haja uma interação com outros elementos.

Segundo Emerson Paraíso [PARAÍSO,1997], os agentes devem identificar quando tais interações são convenientes através de algum mecanismo interno, sendo capazes ainda de interagir com outros elementos segundo uma representação comum a ambos os envolvidos. Para tanto, existe uma relação íntima desta característica com a característica da comunicabilidade entre os agentes.

Reatividade

Este atributo deve estar presente nas definições de agentes, pois a reatividade diz respeito a capacidade do agente em reagir segundo a mudanças no ambiente em que está inserido.

Segundo Michael Wooldridge [WOOLDRIDGE & JENNINGS,1994], a reatividade é a propriedade que permite aos agentes a percepção do seu ambiente e o fornecimento de uma resposta adequada às mudanças que está ocorrendo. Engloba o ambiente do agente: o mundo real, o usuário, outros agentes, Internet, ou qualquer combinação dos itens relatados.

Pró-atividade

Esse atributo torna o agente capaz de resolver problemas quando encontrar situações inesperadas, tornando-o flexível na solução dos problemas, pois o agente possui iniciativa própria ou ainda comportamento independente. Desta forma, o agente não age mais somente segundo mudanças em seu ambiente, mas também segundo suas metas.

Segundo Michael Wooldridge [WOOLDRIDGE & JENNINGS,1996] para que um agente possua a propriedade da pró-atividade, este deve manter um comportamento oportunístico, sempre na direção da satisfação de seus objetivos.

Continuidade Temporal

A continuidade temporal representa a capacidade do agente em executar continuamente seus processos, independente se estes se encontram ativos no ambiente ou adormecidos.

Orientação a Objetivos

O agente deve ser autônomo suficiente para lidar com tarefas complexas em alto nível, decidindo como uma tarefa pode ser cumprida, qual a melhor forma de subdividi-la e como deve ser executada, seguindo os objetivos a que fora proposto.

Os softwares que manifestam tais características, ou propriedades, são agrupados nos softwares que possuem uma fraca noção de agente, sendo representados por sistemas computacionais baseados em software tais como os processos simples executados sob o comando do sistema operacional que apresentam as características acima citadas. Estão agrupados numa segunda conceitualização os sistemas que, além das características acima definidas, se utilizem de conceitos normalmente aplicados a atitudes humanas, utilizando-se de noções mentais ou emocionais, tais como conhecimento, crença/conduta, intenção e obrigação. Para

alguns pesquisadores, este conceito de agentes possui um significado mais específico e, portanto mais preciso sobre o assunto, considerando ainda esta área como a de agentes emocionais, tendo em vista os conceitos envolvidos. Esses sistemas agentes usualmente possuem as seguintes características:

Mobilidade

Esta característica representa a capacidade que o agente possui em se mover de uma máquina para outra [FRANKLIN & GRAESSER,1996].

Essa característica é muito interessante para os agentes, de forma a auxiliar os usuários na busca de informações, ou até mesmo na busca de informações para a solução de suas próprias tarefas e várias outras benéficas que a mobilidade pode trazer. Porém essa capacidade pode resultar na geração de muitos outros problemas, como sobrecarga da rede por onde os agentes ‘vagam’, ou ainda segundo Knapik [KNAPIK & JOHNSON,1998] apresentar problemas relacionados com a segurança dos sistemas, podendo carregar consigo vírus destruidores, ou apresentar problemas com a execução de seu código. Portanto, para o desenvolvimento de agentes móveis deve-se atentar para a implantação de processos de autorização e apresentar garantias de que o ambiente da máquina hospedeira não seja prejudicado, podendo colocá-lo em dificuldades de operação.

Adaptabilidade

Segundo Gilbert [GILBERT,1996], deve ser característica dos agentes a capacidade de observação do ambiente onde está inserido, adaptando suas ações segundo outras anteriormente executadas em condições similares, visando um maior acerto no alcance dos objetivos e metas definidas.

Stan Franklin [FRANKLIN & GRAESSER,1996], citam ainda que o comportamento adaptativo se baseia na modificação do comportamento atual do agente tendo como parâmetros outras experiências passadas.

Colaboração

A colaboração entre agentes vêm da necessidade do cumprimento de tarefas complexas na solução de um problema.

A colaboração entre os agentes ocorre sob dois aspectos:

- *Compartilhamento de tarefas (Task Sharing)*, onde o agente tem uma tarefa a concluir e necessita do auxílio de outros agentes.
- *Compartilhamento de resultados (Result Sharing)*, neste caso o agente detentor de alguma informação a disponibiliza de forma que algum outro agente possa fazer o uso da mesma em benefício próprio. [PARAÍSO, 1997]

Michael Wooldridge [WOOLDRIDGE & JENNINGS,1994] atentam para a necessidade da característica da habilidade social, de modo a permitir uma melhor interação com outros agentes ou até mesmo com humanos segundo algum padrão de comunicação.

Comunicabilidade

A comunicação se torna necessária num ambiente onde possua mais de um agente, e onde esses devem atingir objetivos comuns através da realização de tarefas pré-definidas. O mecanismo de comunicação deve especificar a forma pela qual os agentes vão trocar informações.

Genesereth e Ketchpel [GENESERETH & KETCHPEL,1994] classificam um software como agente se houver algum mecanismo de comunicação entre os agentes.

Segundo Stan Franklin [FRANKLIN & GRAESSER,1996], os agentes podem estar preparados para se comunicarem com outros elementos além de outros agentes, tais como humanos ou outro elemento qualquer pertencente ao seu ambiente.

Aprendizagem

O ato de tomar uma ação que vá atingir o objetivo de forma mais precisa através da avaliação de ações passadas em conjunto com uma avaliação da configuração de seu ambiente vem a consolidar a autonomia deste agente e permitindo-o a aprender sobre seu comportamento diante de situações inesperadas.

Desta forma, não necessariamente o agente vá tomar a ação correta sempre, mas a medida em que as situações forem se configurando, este vai aprendendo constantemente, através de experiências passadas.

B. Lemon [LEMON et alii.,1996] sugere ainda que o aprendizado pode ser interagido com um treinador, onde este passa o conhecimento ao agente no momento em que não possuir um mínimo de conhecimento para tomar alguma decisão.

Pattie Maes, um dos pioneiros na pesquisa em agentes, em seu trabalho *The Maes Agent* [MAES,1995] cita que agentes autônomos são sistemas computacionais que residem em um ambiente dinâmico e complexo, percebendo e agindo de forma autônoma visando a realização de seu conjunto de objetivos ou tarefas para as quais fora designado.

Spector [SPECTOR,1997] nos diz que um agente é qualquer sistema autônomo, onde perceba e aja, objetivando um conjunto de metas específicas num ambiente real ou virtual bem definido e delimitado.

Demazeau [FERBER in DEMAZEAU et alii.,1992], amplia os horizontes de agentes, definido-os como sendo uma entidade real ou virtual que emerge num ambiente onde pode tomar algumas ações, que é capaz de perceber e representar parcialmente este ambiente, que é capaz de comunicar-se com outros agentes e que possui um comportamento autônomo, que é uma consequência de sua observação, seu conhecimento e suas interações com outros agentes.

Agente é uma entidade persistente, dedicada a um propósito específico. Persistente distingue agentes de programas tradicionais, onde estes possuam noções próprias de como concluir suas tarefas. Propósito específico particulariza a aplicação, distinguindo-a de aplicações multifunção, sendo tipicamente mais inteligentes [SMITH, CYPLER e SPOHRER,1994 in FRANKLIN e GRAESSER,1996].

Para Nissen [NISSSEN,1995], um agente pode ser definido como alguma coisa que aja como sendo um procurador de outrém, tendo como objetivo específico a realização de ações que sejam benéficas a quem está sendo representado.

Lizotte [LIZOTTE et alii,1990], define os agentes como sendo entidades físicas ou abstratas, tendo capacidade para:

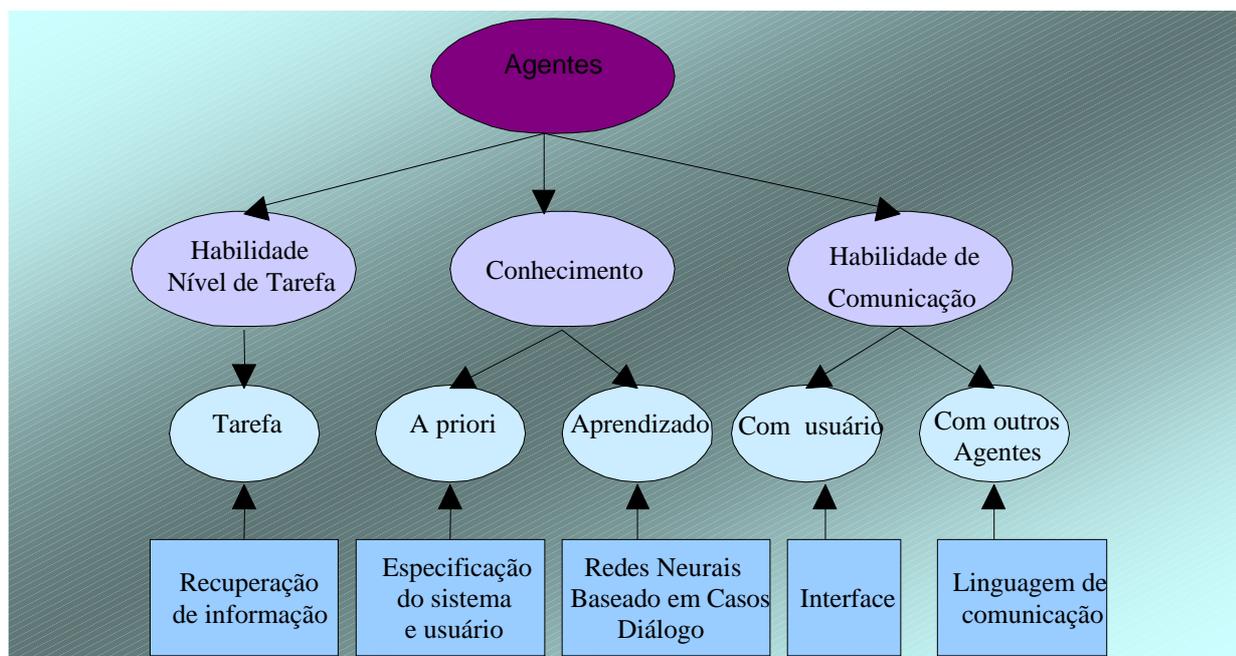
- comunicação com outros agentes;
- agir sobre seu ambiente e sobre suas próprias atuações;

- manipular parte do ambiente onde se encontra inserido;

Stuart Russel e Peter Norvig [RUSSELL e NORVIG,1995], entendem que o ato de agir racionalmente é uma maneira de atingir metas previamente definidas por alguém e desta forma traçam uma forte ligação entre inteligência e ação racional. Em seu trabalho, definem agente como algo que seja capaz de interagir num ambiente, atuando sobre o mesmo devido a sua capacidade de percebê-lo.

Alper Gaglayan e Colin Harrison [GAGLAYAN E HARRISON,1997], representam os agentes sob a forma de um modelo, procurando facilitar o entendimento da funcionalidade, apresentando como os mecanismos funcionam entre si e como interagem com o usuário. A figura 1 destaca o modelo apresentado.

Figura 1 – Modelo de Agentes segundo Gaglayan e Harrison



Segundo o modelo acima são destacados tres módulos principais:

- *Habilidades na execução de tarefas*: Este módulo abrange as habilidades presentes no agente que podem ser utilizadas para que este atinja suas metas, percebendo e agindo sobre o ambiente em que está inserido.
- *Conhecimento*: Este módulo diz respeito às regras que o agente utiliza para realizar suas tarefas. Tal conhecimento pode ser obtido de algumas maneiras: especificado diretamente durante a fase de desenvolvimento do agente, especificado pelo usuário, advindas de outras fontes de conhecimento e ainda advindas do conhecimento do usuário e do próprio ambiente a que o agente pertença.
- *Comunicabilidade*: As habilidades de comunicação pertencentes ao agente implica diretamente na interação com os outros agentes e com os usuários do sistema.

Como podemos perceber, encontramos alguns pontos em comum nas definições a respeito de agentes inteligentes, tais como as características de autonomia e percepção do ambiente, porém uma definição aceita amplamente pelos pesquisadores ainda não se concretizou, pois a cada nova definição proposta mantém-se muitos argumentos relacionados ao trabalho específico do proponente.

Neste trabalho, os agentes são definidos como entidades comunicativas, persistentes e autônomas, tendo a capacidade de representar o ambiente onde estão inseridas, percebendo-o e tomando ações sobre o mesmo a fim de atingir seus objetivos de maneira eficiente.

A tecnologia de agentes está diretamente relacionada ao paradigma do gerenciamento indireto, onde o ser humano e agentes comunicam-se entre si, monitoram os eventos e executam suas tarefas. Sob este aspecto, cabe aos agentes a execução de tarefas para o usuário, o treinamento do usuário, auxílio a usuários que colaboram entre si e o monitoramento de processos e eventos.

2.4 Tipologia de Agentes

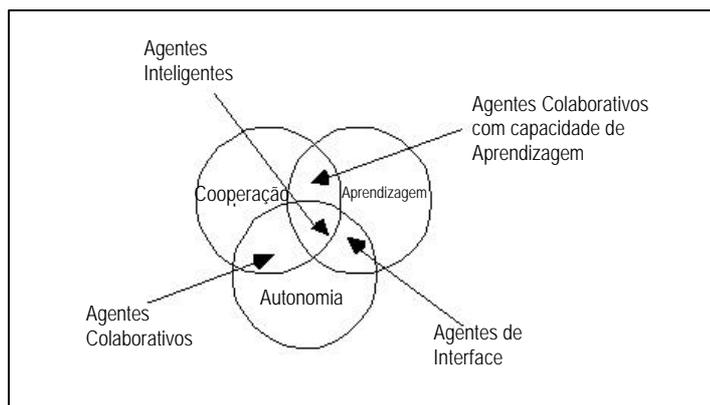
Devido ao fato de cada agente possuir apenas um conjunto de características, estes são agrupados segundo sua tipologia.

Uma tipologia nada mais é do que um grupo de determinadas características normalmente atuantes em conjunto.

Segundo Nwana [NWANA,1996] uma tipologia se refere ao estudo de tipos de entidades e propõe uma classificação dos tipos de agentes de acordo com várias dimensões:

- *Mobilidade*: De acordo com a habilidade que o agente possui para se mover por entre uma rede de computadores, se possuir essa habilidade, é classificado como estático ou móvel.
- *Modelo de Raciocínio Simbólico*: Um agente pode trazer consigo um modelo de raciocínio simbólico pré-definido, norteando suas ações, ditos agentes deliberativos; ou não, tendo suas ações baseadas num sistema de estímulo e resposta, levando em consideração o atual estado do seu ambiente, estes ditos agentes reativos.
- *Atributos Básicos*: Os agentes podem ser classificados de acordo com alguns atributos essenciais, tais como autonomia, aprendizado e cooperação, sendo que a combinação destas características, representadas graficamente pela figura 2, resultam em quatro tipos de agentes:
 - *Colaborativos*: com ênfase na cooperação e autonomia;
 - *Colaborativos com capacidade de aprendizado*: cooperação, autonomia e aprendizagem;
 - *Interface*: com ênfase na autonomia e aprendizagem;
 - *Inteligentes*: Enfatizando a autonomia, aprendizagem e cooperação.

Figura 2 – Classificação Segundo Nwana [Nwana,1996]



NWANA ressalta ainda que não se pretende impor limites quanto ao agrupamento dos atributos básicos.

Um agente pode ser classificado de acordo com a função principal que desenvolve, tais como os agentes de informação, que atuam na busca ou tratamento de informações.

Estas características atuando em conjunto podem formar algumas tipologias, como por exemplo:

- Agentes Colaborativos

Os agentes colaborativos cumprem suas tarefas de acordo com sua especialidade, colaborando para a solução de um problema maior. Nestes agentes, destacam-se as características de autonomia e cooperação.

Agentes encaixados nesta tipologia necessitam de mecanismos de negociação, a fim de promover o comprometimento dos parceiros na execução das tarefas deliberadas, além de possuir mecanismos que possibilitem a comunicação com seus parceiros.

- Agentes de Interface

Este tipo de agente provê uma assistência direta ao usuário do computador no sentido de auxiliá-lo no uso de algum software. Sua forma de atuação baseia-se na monitoração das ações do usuário naquele ambiente onde o agente se faz presente. Desta forma, suas características principais baseiam-se na autonomia e no aprendizado, uma vez que deva observar e absorver a maneira como o usuário executa suas ações, sugerindo formas de execução similares, sempre a critério do usuário permiti-la ou não.

Este agente vem a atuar geralmente em segundo plano, procurando encontrar em sua base de conhecimento certos padrões que se repetem frequentemente, permitindo então que o agente forneça o auxílio ao usuário na forma de uma sugestão do uso de algum mecanismo disponível ou na sugestão de uma forma de execução de alguma tarefa.

Podemos concluir então que este tipo de agente tem como objetivo principal promover a aproximação entre usuários e computadores, sob a forma de prover os usuários com algum tipo de auxílio no desempenho das tarefas que necessita executar, com base na observância da maneira como o usuário vem desenvolvendo suas atividades naquele software.

Para Nwana, o agente desempenha algumas tarefas em conjunto com o usuário, podendo ser classificado também como um assistente pessoal para aquela aplicação.

Pattie Maes [MAES,1997], aponta três diferentes maneiras pelas quais o agente pode aprimorar seu aprendizado:

- Através de uma comunicação direta entre o agente e usuário, onde este confirme ou não a assimilação daquela ação por parte do agente;
- Através de ordens explícitas do usuário ao agente, instruindo-o na execução das tarefas;
- Através da informação de outros agentes, passando a ter como colaborador outro elemento que não o usuário;

Dentre estas formas de aprendizado podemos perceber claramente que o agente depende de duas fontes para aprender suas habilidades: o usuário e outros agentes que convivem no mesmo ambiente.

Tais agentes também são conhecidos como agentes assistentes, sendo que neste caso além de desempenharem funções de auxílio sob a aprovação do usuário, também são habilitados a desenvolverem algumas ações de forma autônoma sem nenhuma intervenção do usuário.

- Agentes Móveis

Os agentes ditos móveis são aqueles que possuem a habilidade de se locomover de uma máquina a outra através dos mecanismos de comunicação existentes nas redes de computadores. A presença da característica da mobilidade é essencial para este tipo de agente. Aqui devemos levar em consideração alguns problemas que esta característica pode proporcionar, tais como aumento de tráfego nas redes, segurança, dentre outros.

Segundo Nwana [NWANA,1996], os agentes móveis são processos computacionais capazes de percorrer através de grandes rede de computadores, interagir com outros computadores que servem estas redes com algum serviço ou informação, coletar informações pelos quais sejam interessantes na obtenção de seu objetivo, e retornar a seu computador originário após concluído suas tarefas propostas por seu usuário. Ainda segundo Nwana a mobilidade não é uma condição essencial para um agente, mas um agente móvel estão presentes as características de autonomia e cooperação.

- Agentes de informação

Os agentes de informação vêm atuar na busca de informações em ambientes onde as informações encontram-se distribuídas, havendo a necessidade de reuni-las da melhor forma possível. Desta forma tais agentes não tem sua atuação na busca simples, tais como muitos mecanismos de busca, mas a proposta é promover mecanismos inteligentes autônomos, onde estes agentes venham a reconhecer padrões de informação, classificar as que sejam mais importantes dentro de um contexto preestabelecido. Segundo Nwana, os agentes de informação vieram à tona devido ao rápido crescimento do volume de informações a ser gerenciado e da necessidade de extrair novas informações a partir das existentes.

Estes agentes desenvolvem atividades de gerenciamento, manipulação ou agrupamento das informações por diferentes fontes distribuídas.

- Agentes Reativos

Os agentes reativos não possuem um modelo de seu ambiente para determinar suas ações. Eles atuam sob um estímulo, devolvendo uma resposta ao ambiente em que está inserido. Desta forma, são agentes simples que não modelam o ambiente para tornar as ações, mas possuem um mapeamento de determinadas situações que podem se configurar em seu ambiente e um conjunto de respostas associadas a estas configurações.

Segundo Maes [MAES,1991], existem três conceitos básicos associados aos agentes reativos:

- Funcionalidade emergente, onde devido ao fato de tais agentes serem relativamente simples e interagirem com outros agentes com métodos simples, quando observados de uma maneira global no ambiente, tais interações podem requerer padrões complexos de conduta.
- Decomposição de tarefas: Os agentes reativos são vistos como uma coleção de módulos que operem de forma autônoma e são responsáveis por tarefas específicas. A comunicação entre estes módulos é mínima e realizada sob baixo nível, não existindo um modelo global de comunicação.
- Os agentes reativos não trabalham com representações simbólicas.

- Agentes híbridos

Apesar da existência dos tipos de agentes vistos até o momento, cada qual compondo seu conjunto particular de características, os agentes ainda podem apresentar a necessidade da junção num mesmo projeto de diferentes tipologias, configurando um agente híbrido. Segundo Nwana [NWANA,1996], agentes híbridos se referem a sua constituição como uma combinação de duas ou mais filosofias de agentes num modelo de agentes único, podendo incluir uma filosofia de agentes móveis, de agentes de interface, de agentes

colaborativos, etc... Diz ainda que, para algumas aplicações, os benefícios obtidos com a combinação destas filosofias num único agente é maior do que os obtidos no mesmo agente, baseado inteiramente numa única filosofia.

2.5 Áreas de Atuação de Agentes

Atualmente, os sistemas de agentes estão presentes em inúmeras áreas, devido a sua versatilidade na resolução de problemas, seja sob a abordagem de Resolução Distribuída de Problemas ou sob a abordagem de Sistemas Multi-Agentes. A seguir estão descritas algumas áreas de atuação onde os sistemas de agentes estão presentes, áreas estas que nos são mais enfáticas no momento presente, sendo que seria praticamente inviável arrolar todas.

Sistemas e Gerenciamento de Redes de Computadores

Arquiteturas de agentes já existiam nos sistemas e gerenciamento de redes, porém tais agentes possuíam uma função estática. Os agentes inteligentes vêm atuar nesta área de forma a incrementar os softwares de gerenciamento, filtrando dados ou tomando certas ações devido a alguma nova configuração no seu ambiente, proporcionando certa dinamicidade no gerenciamento de redes.

Gerenciamento do acesso móvel

Os usuários de uma rede procuram, além de ter acesso aos recursos disponíveis na rede de qualquer lugar, necessitam acessar tais recursos independente da limitação de banda existente no meio de comunicação das tecnologias móveis disponíveis, tais como comunicação de redes sem fio, e independentemente da volatilidade desta rede. Aqui, agentes inteligentes podem residir na rede a fim de solicitar suas necessidades de comunicação ao sistema de gerência, ou ainda procurar informações e solicitações dos usuários, transmitindo apenas o resultado das operações.

Correio e Sistema de Mensagens

Estes sistemas necessitam de sub-sistemas que atuem na organização e priorização de mensagens. Os agentes inteligentes podem atuar nestas funções, permitindo que o usuário interaja com o agente personalizando suas tarefas através da definição de regras que venham a satisfazer suas necessidades, podendo modificá-las assim que suas necessidades forem outras.

Gerenciamento e acesso à Informação

Nesta área de intensa atividade os agentes inteligentes atuam de forma a auxiliar os usuários na busca de informações realizando, quando necessário, tarefas de categorização, disseminação, anotação (registro) e compartilhamento sobre as informações. Podem atuar ainda na mineração de dados, utilizando-se de técnicas de pesquisa específicas.

Colaboração

Uma área já bastante difundida e em rápido crescimento é o trabalho em grupo, ou trabalho em conjunto através de compartilhamento de documentos, videoconferência, etc..., onde existem dois denominadores comuns: um é o compartilhamento de recursos, o outro é a equipe de trabalho. Aqui, os usuários necessitam, além de infra-estrutura de suporte aos recursos necessários, de agentes que as auxiliem na montagem e gerenciamento de grupos de pessoas e na gerência da produção de seu trabalho.

Fluxo de Trabalho e Gerenciamento Administrativo

Nesta área onde visa a melhoria e a automação dos processos de uma empresa, os agentes inteligentes vêm a atuar de forma a agilizar os processos, tornando-os mais eficientes, através da sua automação, reduzindo custos com o gerenciamento automatizado destes processos, ou parte deles.

Comércio Eletrônico

Esta área da rede mundial abrange a necessidade de compradores encontrarem quem venda os produtos e serviços necessários, bem como informações adicionais a respeito do produto ou serviço oferecido. Da mesma maneira, os vendedores precisam levar seus produtos e serviços com suas características aos compradores. Enfim, compradores e vendedores necessitam automatizar suas necessidades de compra e venda de produtos e serviços. Neste caso, agentes inteligentes podem sair às compras, levando consigo uma série de restrições e especificações técnicas, negociando com outros agentes a compra ou venda do produto ou serviço.

Interfaces Adaptativas e/ou Inteligentes

Acompanhando o crescimento do número de usuários dos sistemas computacionais as interfaces necessitam aprender os hábitos e preferências de seus usuários, afim de se adaptar aos mesmos. Os agentes de interface atuam nesta área monitorando as atividades dos usuários, criando um modelo de ações para cada usuário, de forma a auxiliá-los na resolução de possíveis problemas, ou indicar ações de forma a facilitar seu

trabalho. Combinando esta tecnologia com a habilidade de sistemas de reconhecimento de voz, tornam as interfaces mais amigáveis ao interagir com as pessoas.

2.6 Comunicação entre Agentes

Os agentes enquanto elementos independentes e autônomos convivendo num sistema computacional necessitam obviamente da manutenção de algum mecanismo de comunicação, seja ela interna para com outros agentes de um mesmo ambiente, ou externa ao seu ambiente. Esse mecanismo vem a atuar como uma linguagem de comunicação que seja comum entre os agentes e, apesar deste consenso existir no âmbito dos pesquisadores, ainda não se obteve um consenso sobre a melhor linguagem a ser utilizada na comunicação entre os agentes.

A comunicação entre os agentes vem permitir e até mesmo determinar o nível de colaboração alcançado por um sistema de agentes. Segundo Lesser [LESSER,1987], a cooperação total entre agentes só é encontrado em sistemas onde os agentes vem a atuar em conjunto na solução de um problema, podendo superar suas capacidades individuais, tais sistemas são denominados sistemas de resolução de problemas distribuído e cooperativo, onde a cooperação é exigida ao máximo para que, através da ajuda mútua dos elementos, cada qual com sua especialidade, se atinja o objetivo de maneira mais eficaz.

Quanto maior a cooperação, menor a probabilidade de que os objetivos dos agentes entrem em conflito. Para a solução deste problema Sycara [SYCARA,1989], apresenta a necessidade de um processo de resolução de conflitos, um mediador, atuando como um árbitro para os demais agentes.

Os sistemas agentes podem ser classificados segundo o tipo de comunicação que é capaz de estabelecer, podendo haver desde nenhum tipo de comunicação direta com outros agentes até uma comunicação em alto nível, conforme a classificação apresentada a seguir.

Comunicação inexistente

Os agentes podem interferir no trabalho dos demais agentes sem obter nenhuma comunicação. Segundo Shelling [SHELLING,1960], este modelo de comunicação é o que obtém os melhores resultados quando os objetivos dos agentes não estão interrelacionados, não havendo conflitos de nenhum tipo. Certamente a falta de mecanismos de comunicação representa uma grande barreira para o sistema, havendo a necessidade de incorporar sistemas complexos de raciocínio acerca dos demais, trabalhar com probabilidades e

incertezas, necessitando de um maior esforço computacional.

Primitivas de comunicação

C. Hoare [HOARE,1978], cita que neste caso a comunicação acontece a partir de um número fixo de sinais previamente estabelecidos, podendo ser um problema pois devido a limitação dos sinais vem a limitar também a cooperação entre os agentes.

Troca de planos de informações

Neste caso os agentes podem trocar seus planos, podendo desta forma adaptar e rever suas estratégias de ação. Apesar da grande vantagem proporcionada pela troca de planos, Rosenschein [ROSENSCHEIN,1986], apresenta alguns inconvenientes, tais como o alto custo computacional exercido sobre o meio de comunicação e também a falta de garantias de que o plano resultante possa ser aceito por completo pela base de conhecimentos do agente.

Troca através de quadro negro

Segundo Hayes-Roth [HAYES,1985], este tipo de comunicação baseia-se no compartilhamento de informações através de uma área de memória onde os agentes possam se utilizar ou depositar dados parciais ou encontrar algum tipo de informação que lhes seja útil.

Este método pode apresentar problemas dependendo da maneira como fluem as informações no sistema, se é baseada nos dados ou em objetivos.

Troca de mensagens

Esta estratégia apresentada [AGHA,1988] se baseia na atuação de agentes em resposta ao processamento de uma comunicação, podendo resultar numa comunicação enviada ao próprio agente, a criação de outros agentes do mesmo tipo e a remodelagem do comportamento, que consiste na descrição do novo estado do agente após a comunicação se efetivar.

Comunicação em alto nível

Permite levar ao receptor as intenções, compromissos, enfim, o estado do emissor através da geração e interpretação de declarações, intencionando que o receptor assuma o mesmo estado que o emissor [GROSZ,1990].

Para Tim Finin [FININ,1993], existem alguns níveis nos quais os sistemas baseados em agentes devem acordar, ao menos no que se refere a sua interface de comunicação. São eles:

Transporte:	Cuida do modo como os agentes enviam e recebem suas mensagens.
Linguagem:	Trata do significado das mensagens.

Política: Trata de como as conversações (comunicações) entre os agentes são estruturadas.
Arquitetura: Trata do modo como os sistemas baseados em agentes podem se conectar.

2.6.1 Linguagens de Comunicação entre Agentes

A ausência de uma padronização da linguagem de comunicação entre agentes atrapalha na construção dos sistemas, pois podem surgir problemas ao ser necessário que os agentes construídos sob uma linguagem de comunicação se comuniquem diretamente com agentes que não tenham sido construídos sob o mesmo padrão. Os problemas vão desde inconsistências no uso da sintaxe do vocabulário até ao uso de expressões idênticas com significados distintos para ambos os agentes ou ainda o uso de expressões diferentes tendo o mesmo significado, prejudicando em muito o estabelecimento ou a manutenção da comunicação.

A construção de uma linguagem de comunicação pode ser abordada segundo os enfoques procedural e declarativo, descritos a seguir:

Enfoque Procedural

Está baseado no modelo de comunicação através de diretivas procedurais, onde as linguagens devem permitir aos agentes o intercâmbio tanto de comandos individuais como de um conjunto deles, formando um procedimento. Tais linguagens possuem uma execução eficiente, porém não há garantias de que as informações necessárias aos procedimentos que estão sendo intercambiados estejam residentes também no agente receptor do procedimento. São exemplos deste tipo de linguagens a Telescript e TCL.

Enfoque Declarativo

Está fundamentado no modelo de comunicação através de informações declarativas, podendo ser asserções, definições, dentre outras. Para que tal enfoque tenha utilidade prática, deve ser capaz de comunicar informações de vários tipos, inclusive um conjunto delas.

Um trabalho que segue o enfoque declarativo desenvolvido pelo ARPA Knowledge Sharing Effort, propondo a ACL (Agent Communication Language). Esta linguagem é formada por três módulos principais:

Vocabulário

Consiste de um dicionário de dados onde encontramos as palavras apropriadas para cada área de aplicação. Relacionado a cada palavra, temos uma descrição da palavra, permitindo uma melhor explanação da mesma, anotações formais escritas na linguagem KIF (Knowledge Interchange Format), para utilização nos programas.

Linguagem Interna KIF(Knowledge Interchange Format)

A linguagem KIF é uma versão do cálculo de predicados de primeira ordem, possuindo algumas extensões, permitindo a codificação de dados, de restrições, de regras, etc... As mensagens podem ainda levar outras informações de maneira implícitas sobre seu emissor, receptor, tempo, histórico, etc... Genesereth e Fikles [GENESERETH,1992], descrevem por completo a linguagem KIF e sua utilização.

Linguagem Externa KQML (Knowledge Query and Manipulation Language)

A KQML é uma linguagem, definindo um formato para as mensagens, bem como um protocolo de comunicação para estas mensagens. As estruturas KIF podem ser encapsuladas nas operações da KQML, que são definidas através das *performatives*. Desta forma, uma expressão da linguagem KQML se torna uma mensagem ACL, onde os argumentos são formados pelas sentenças da linguagem KIF que, por sua vez, são formados pelas palavras do vocabulário ACL. O uso das sentenças KIF não é exigido pela KQML, porém o uso de uma linguagem não padronizada ou não tão aceita no ambiente de agentes pode levar a dificuldades na comunicação com agentes desenvolvidos por outras equipes de trabalho.

3. GERENCIAMENTO INDIRETO

“Suddenly the computer is a window into a world of information, people, software... And this world is vast, unstructured, and completely dynamic.

It's no longer the case that a person can be in control of this world and master it”

Pattie Maes

3.1 Introdução

Diariamente nos deparamos com uma série de tarefas a desempenhar, procurando atingir nossos objetivos. Tais tarefas podem ser executadas de diferentes maneiras, dependendo das ferramentas disponíveis para fazê-la. Percebe-se, no mundo moderno, uma preocupação constante com o aprimoramento na maneira como o homem realiza seus afazeres, visando sempre melhorias, de forma a proporcionar maior conforto, agilidade e rapidez no seu desenvolvimento.

A invenção do computador, juntamente com a evolução tecnológica, trouxe muitos benefícios ao homem, proporcionando-lhe novas descobertas e conseqüentemente novos desafios a serem vencidos.

Atualmente os computadores são o meio que o homem utiliza para resolver suas tarefas, seja pelas facilidades que oferece ou pela comodidade proporcionada por eles. As ferramentas disponíveis para satisfazer as necessidades via computadores são executadas, em sua grande maioria, de forma passiva, ou seja, a ferramenta é iniciada pelo usuário de forma direta, e este detém total controle dos mecanismos que estão sendo executados.

A complexidade dos ambientes computacionais cresce a cada dia, ou pode-se dizer, a

cada atualização de software realizada por seus fabricantes, adicionando novas funções ao software ou novas características às funções já existentes proporcionadas pela evolução tecnológica.

Desta forma, o ambiente onde o usuário opera vem se transformando, tornando-se mais dinâmico; agora o usuário trabalha com informações locais juntamente com informações provenientes de outras fontes, tais como acessos remotos a bases de dados da grande rede mundial.

Está cada vez mais presente a necessidade de um novo modelo na forma como o usuário interage com suas ferramentas de trabalho, uma vez que este não detém o controle, ou o conhecimento de tudo o que pode ser utilizado em benefício próprio ao desenvolver suas tarefas.

O impulso proporcionado pelo emprego das novas tecnologias, tais como comércio eletrônico, modelo de negócios eletrônico e serviços eletrônicos, normalmente tratados como *e-commerce*, *e-business* e *e-services*, está levando à criação de verdadeiras organizações virtuais, refletindo diretamente na interação do usuário com tais organizações.

A metáfora do gerenciamento indireto vem sendo proposta como uma alternativa ao método atualmente utilizado, denominado manipulação direta, objetivando mudanças na interação homem-computador, de forma que o usuário não necessite iniciar e controlar todas as tarefas necessárias à realização de algum trabalho, pois elas são, e serão, cada vez mais, em maior número.

Tal metáfora está baseada na utilização de alguma tecnologia que venha a proporcionar que as tarefas a serem desempenhadas possam ser totalmente gerenciadas, isso inclui execução, controle e monitoração, por entidades que estejam presentes no ambiente computacional do usuário, sem que este necessite de conhecimentos específicos a respeito do uso das ferramentas utilizadas, isto é de domínio destas entidades.

3.2 Execução e Gerenciamento de Tarefas

A operação de um ambiente computacional baseia-se na execução de aplicativos ou softwares de computador. Para cada aplicativo que se executa, o usuário deve ter um conhecimento específico para poder realizar suas tarefas.

Com a evolução tecnológica e com a conseqüente disponibilização de uma gama maior de operações, levam os aplicativos a uma maior complexidade no seu manuseio. Isso vem a exigir do operador, leia-se usuário, um nível de conhecimento cada vez mais específico e detalhado a respeito do software no que se refere a sua abrangência de conteúdo. Outro ponto importante é o da diversificação no número de tarefas que o usuário desempenha frente ao computador, exigindo do usuário, além de um conhecimento mais aprimorado a respeito do aplicativo, conhecer um número cada vez maior de softwares que satisfaça suas necessidades.

Frente a essa tendência no aumento da complexidade no uso dos softwares, necessita-se de ferramentas que auxiliem o usuário de forma pró-ativa, que seja iniciada, gerenciada e finalizada sem a determinação explícita do mesmo, pois este está preocupado com os resultados da execução e não com a forma de como é executado.

Essa nova abordagem revolucionária não somente o uso do software, mas principalmente a maneira como se comporta o ambiente onde o usuário está inserido, pois este passa a contar com auxiliares autônomos que existem para facilitar a execução das tarefas, seja através da busca de informações, filtros de conteúdo, auxílio com sugestões de uso, demonstração automática de ações pretendidas pelo usuário, etc.

Tais entidades auxiliadoras estão ativas sempre que o usuário estiver manipulando alguma informação que seja da sua área de domínio específica, possuindo conhecimento a respeito das informações que é capaz de manipular, bem como informações a respeito do usuário que opera a máquina, como preferências, forma de tomar as principais ações, etc.

3.3 Manipulação Direta

Desde a invenção do computador até meados dos anos 90, a interação entre o usuário e

a máquina era realizada por meio de comandos enviados à máquina através de algum dispositivo, como teclado, cartões perfurados, dentre outros. Esse tipo de interação denomina-se interface do usuário baseada em comandos, sendo que o usuário informa à máquina as ações a serem tomadas através de comandos de linguagem reconhecidos pela mesma, desde que obedecida uma determinada sintaxe apropriada para cada comando.

Essa forma de interação, apesar de eficiente para as necessidades da época, impunha certas barreiras, gerando algumas dificuldades, como a necessidade de usuários especializados no conhecimento dos comandos e da sintaxe utilizada, e o limite em suas produções, pois o computador era visto como um intermediário entre o usuário e a realização da tarefa propriamente dita.

Com a proliferação dos computadores e a conseqüente disseminação da microinformática, foram criadas novas necessidades para o estabelecimento da interação entre o usuário e a máquina. A partir de então, quem estava no comando não era mais um usuário especialista em comandos e sintaxes complexas, mas um usuário em potencial, cuja área de atuação era outra qualquer, e que se utilizava da máquina, tendo-a como um meio mais eficiente para a execução de suas tarefas.

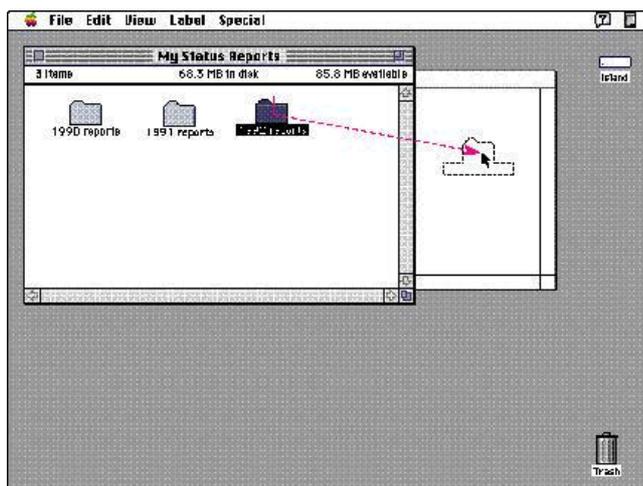
Com este novo panorama, a interação via comandos de linguagem tornava-se deveras complicada para o usuário, surgindo então uma necessidade urgente no âmbito de aumentar a produtividade, o aprendizado mais rápido e as facilidades no uso das ferramentas de informática. Surgiu então o conceito de Manipulação Direta, o qual teve seu início formal a partir de 1983, com Bem Shneiderman [SHNEIDERMAN,1993], que introduz a manipulação de objetos pertencentes ao ambiente computacional de forma direta através do uso de periféricos tais como o mouse.

A manipulação direta refere-se à maneira como é estabelecida a interação entre homem e máquina, caracterizando-se por um campo interdisciplinar, envolvendo áreas de estudo em ciências da computação, psicologia, lingüística, interface gráfica, dentre outras.

Mike Dennehy [DENNEHY,1993], define manipulação direta como sendo um estilo de interação entre homem e máquina, o qual se caracteriza por uma representação natural de objetos e ações, dando a impressão de que o próprio usuário está realizando a tarefa, sem ter a máquina como intermediador.

Para [APPLE,1996] a manipulação direta, além de proporcionar ao usuário a sensação de controle direto sobre os objetos, permite que as operações realizadas sobre eles tenham um impacto direto sobre os mesmos e este impacto é imediatamente visível.

Figura 3 – Manipulação direta de um objeto – [Apple,1996]



A figura 3 consiste em uma demonstração de uma ação sobre um objeto do ambiente. Neste caso, uma pasta de documentos sendo movida de uma região do sistema para uma outra, caracterizando muito bem a noção de que o usuário possui o domínio da ação, tendo controle direto sobre o objeto em questão, inexistindo aqui o diálogo com a máquina como pré-requisito para a tomada da ação.

Hutchins [HUTCHINS et alii,1996], define manipulação direta como sendo a sensação de estar envolvido por um mundo de objetos ao invés de intermediários.

Carrie Heeter [HEETER,1991], cita que uma interface não necessariamente deva ser gráfica para que possa incorporar conceitos de manipulação direta, sendo que este tipo de interface vem a auxiliar em muito a aplicação do conceito.

Bill Mcmillan [MCMILLAN & PATTERSON,1998], apresenta dois estilos onde o conceito de manipulação direta possa ser aplicado:

- Preenchimento de formulários: permitindo uma navegação pelos campos que já foram preenchidos, adicionando-se elementos de apoio, tais como botões de rádio, caixas de diálogo, campos tipo caixa de checagem, menus tipo pop up.
- Manipulação direta através de ambiente gráfico: onde se aplica a manipulação dos objetos através de ações físicas tais como arrastar, eliminar os objetos, ao invés da digitação de comandos.

Sjoerd Michels [MICHELS,1995], apresenta alguns benefícios e possíveis problemas ao se utilizar de Manipulação Direta. Como benefícios obtidos cabe citar:

- Como o sistema não possui sintaxes complexas, usuários iniciantes aprendem rapidamente as funcionalidades básicas, geralmente através de demonstrações e tutoriais.
- Usuários experientes têm sua produtividade aumentada, realizando um grande número de tarefas rapidamente.
- Sem a presença de uma rígida sintaxe, os erros são cada vez menos frequentes.
- Os usuários percebem claramente se suas ações estão conduzindo ao objetivo esperado, caso contrário pode mudar de atitudes.
- As ações tomadas podem ser revertidas, dando maior confiabilidade e segurança ao usuário.

Como possíveis problemas na aplicação deste conceito são citados pelo autor:

- A necessidade de maiores recursos do sistema.
- Operações repetitivas são executadas mais eficientemente através de scripts de comando ou macros.
- A manipulação direta requer do usuário um substancial conhecimento na manipulação dos objetos.

Um estudo da empresa americana Microsoft Corporation realizado em 1990 apontou que o uso de interface gráfica aliada ao uso de conceitos de Manipulação Direta ao invés do uso de interface por comandos de linguagem proporcionam ao usuário vários benefícios, dentre os quais cabe citar:

- Maior produtividade;
- O aprendizado se dá em maior quantidade e em menos tempo;
- Maior segurança no uso das ferramentas, demonstrando uma tendência de auto aprendizagem e exploração do ambiente.

A título de exemplos práticos do uso de Manipulação Direta, temos softwares tais como Windows e Office da Microsoft Corporation, System V da Apple Computer, dentre outros.

3.4 Gerenciamento Indireto

O ambiente onde o usuário opera o computador vem se tornando mais complexo, tendo que abrigar e manipular um número cada vez maior de informações, além de maior diversidade, como imagens, sons, etc.

Tal aumento na complexidade se justifica devido a alguns fatores tais como:

- *Aprimoramento dos softwares existentes:* A cada dia, os softwares são lançados ao mercado contemplando novas funcionalidades ou aprimorando as já existentes, exigindo do usuário, o qual manipula diretamente todas as ações, um nível de conhecimento cada vez mais específico com relação à ferramenta que utiliza.
- *Aumento de tarefas a serem realizadas via computador:* Cada vez mais, novos aplicativos são criados com a finalidade de informatizar a execução de tarefas até então executadas de forma manual, visando principalmente maior facilidade na execução, maior produtividade e qualidade.
- *Diversificação de usuários:* Até pouco tempo, o computador era geralmente utilizado por um usuário, o qual tinha conhecimento das informações ali depositadas bem como dos softwares disponíveis. Atualmente a realidade se faz um pouco diferente. Temos usuários que operam mais de um computador, tendo, conseqüentemente, os computadores sendo operados por mais de um usuário. Essa mudança na utilização dos computadores se deve ao fato do aumento de tarefas informatizadas, da evolução tecnológica das redes de computadores, permitindo

que usuários se conectem remotamente a outras máquinas, da necessidade de busca de informações a locais remotos informatizados através do uso, das redes de computadores.

À medida em que o ambiente computacional onde o usuário opera se torna mais complexo, este usuário é pressionado a obter um maior nível de conhecimento desse ambiente, permitindo que se usufrua dos benefícios oferecidos.

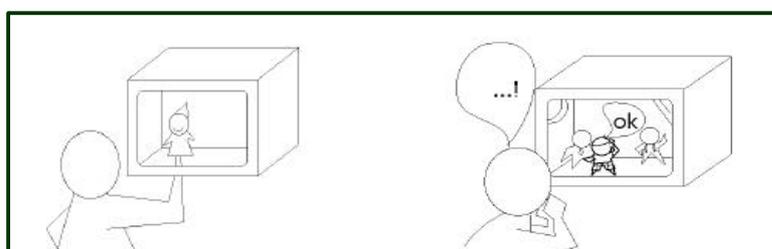
Porém os usuários nem sempre podem reservar algum tempo para se aprimorarem no uso de seus aplicativos, mas sentem a necessidade do uso de novas ferramentas que lhes proporcionem maior qualidade e produtividade. Pattie Maes [MAES,1997/2], afirma: “Eu pessoalmente acredito na importância crucial de softwares mais pró-ativos e personalizados, porque nosso ambiente computacional está se tornando mais e mais complexo, e nós enquanto usuários não podemos nos manter sempre no topo dos acontecimentos”.

Surge então a necessidade de criarmos ambientes computacionais pró-ativos, capazes de personalizar o ambiente de acordo com as preferências daquele usuário que opera a máquina naquele momento.

A metáfora do gerenciamento indireto está baseada, principalmente, nas características de pró-atividade e personificação. A pró-atividade ou ainda capacidade de iniciativa, segundo Mor Enric [ENRIC,1998], compreende a exteriorização de algum tipo de comportamento orientado a alcançar seus objetivos por si só, não havendo a necessidade de outrem comandar a ação. A personificação do software diz respeito à tomada da ação levando em consideração o perfil do usuário que está operando o software ou seja, este deve agir de acordo com o perfil do usuário.

A figura 4 procura demonstrar graficamente a principal diferença entre Manipulação Direta e Gerenciamento Indireto.

Figura 4 – Manipulação direta vs. Gerenciamento Indireto

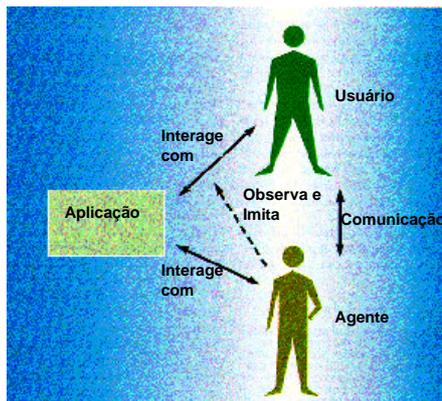


Essas duas características podem ser identificadas diretamente num software construído sob a ótica do gerenciamento indireto. No âmbito da pró-atividade, o software deve abrigar funcionalidades que permitem a tomada de atitudes baseadas na percepção da configuração do ambiente, sem a necessidade da interferência direta do usuário. Sob o aspecto da personificação, o software deve possuir mecanismos para identificar a maneira como o usuário conduz a ferramenta, construindo um perfil e, podendo, futuramente, auxiliar o usuário no uso da ferramenta sob a forma de sugestões, como por exemplo, o uso ou a configuração do ambiente.

O emprego do conceito de gerenciamento indireto está sendo muito bem aceito, aliado à tecnologia de construção de agentes, mais especificamente a de agentes de interface, tendo em vista que a teoria dos agentes propõe o emprego de certas características básicas para que se caracterize como um agente, tais como autonomia, habilidade social, pró-atividade, colaboração, dentre outras. Muitas delas tendo um relacionamento direto com o emprego do gerenciamento indireto.

Pattie Maes [MAES,1997], comenta que a idéia da utilização de agentes como interface para delegar tarefas a sistemas computadorizados foi inicialmente introduzida por Nicholas Negroponte [NEGROPONTE,1970] e Alan Kay [KAY,1984] e posteriormente adotada por diversas empresas atuantes na área tecnológica.

O uso da tecnologia de agentes e gerenciamento indireto deve produzir, não necessariamente um agente de interface que atue como um intermediário entre o usuário e a aplicação, mas pode ter um comportamento semelhante ao de um assistente, atuando de maneira cooperativa com o usuário e com os recursos oferecidos pelo ambiente computacional na resolução da tarefa. Na figura 5, temos um esquema gráfico de como este assistente pode atuar.



Como pode ser percebido, tal agente assistente pode interagir com a aplicação que está sendo executada, bem como diretamente com o usuário responsável por aquela execução, a fim de solicitar informações necessárias num dado momento ou sugerir algum auxílio a este usuário.

Para Pattie Maes a construção de agentes com a finalidade de prover ao usuário algum nível de delegação de tarefas tem uma série de obstáculos, pois tais software-agentes devem por exemplo, ter percepção de quando o usuário quer ser auxiliado e qual a melhor maneira de auxiliá-lo naquele momento.

3.5 Competência e Confiabilidade

A construção de agentes com as características de assistente vista nos moldes da figura 5 se depara com dois principais problemas: o da competência do assistente na resolução das tarefas e o da confiabilidade que este agente possui ao acionar determinada tarefa sem a aprovação explícita do usuário.

Para Pattie Maes [MAES,1997], o problema da competência abrange necessariamente:

- A forma como o agente adquire o conhecimento de que necessita;
- Com que mecanismos auxiliar o usuário;

- De que forma auxiliar o usuário;

O problema da confiabilidade diz respeito a como garantir que o usuário se sinta satisfeito delegando determinadas tarefas a um agente.

Atualmente existem duas abordagens que tratam desse problema. Na primeira, o software do usuário é o próprio agente, o qual consiste de um conjunto de regras definidas pelo usuário para o processamento de informações relacionadas a determinada tarefa. Uma vez definidas, essas regras realizam tarefas sem que o usuário as acione diretamente.

A segunda abordagem para o problema da competência e confiabilidade é denominada abordagem baseada no conhecimento, a qual consiste em um agente de interface atuando em conjunto com uma base de conhecimentos atuantes, especificamente, no domínio da aplicação e do usuário. Tal abordagem é utilizada pela maioria das pessoas que atualmente se utilizam de inteligência artificial aplicada a interfaces inteligentes, onde o agente consulta a base de conhecimentos e, aliado ao contexto atual do usuário, procura reconhecer as atitudes do usuário e encontrar oportunidades para auxiliá-lo.

Pattie Maes [MAES,1997] aponta possíveis problemas com tais abordagens. No caso da primeira, o critério da competência não é tratado de maneira satisfatória, pois requer muito esforço de conhecimento por parte do usuário final, uma vez que este deve:

- Reconhecer a oportunidade para o emprego do agente;
- Tomar a iniciativa para criar o agente;
- Especificar o conhecimento de forma explícita ao agente;
- Dar manutenção às regras constantemente;

O problema da confiabilidade é praticamente descartado, uma vez que nesse tipo de abordagem o próprio usuário provê a programação das tarefas.

Quanto a problemas concernentes à segunda abordagem, Maes cita que tanto a competência quanto a confiabilidade podem ser problemas em potencial. No que diz respeito à competência, essa abordagem requer um trabalho exaustivo por parte da máquina que trata do conhecimento, pois um grande volume de informações pode ser necessário à base de conhecimentos para viabilizar o funcionamento do agente de maneira satisfatória. Um

segundo problema apontado para essa abordagem se refere à diversidade de usuários. Uma vez inserido o conhecimento, esse é estático, não pode ser customizado para usos e preferências individuais, e a robustez e agilidade para prover um agente com todo o conhecimento de que necessita para atender satisfatoriamente a vários usuários é questionável.

Diante dos problemas e obstáculos levantados, Pattie Maes [MAES,1997], propõe uma nova abordagem para tratar questões referentes a competência e confiabilidade. É trabalhada a hipótese de que, sobre certas condições, o agente assistente pode se autoprogramar e adquirir por si só o conhecimento que necessita para auxiliar o usuário.

Inicialmente é dado ao assistente um mínimo de conhecimento necessário para que este, então, comece a aprender através de experiências do usuário e também a partir de outras entidades com as quais possa trocar informações, tais como outros agentes.

Para obter sucesso no emprego dessa abordagem, duas condições devem ser essencialmente satisfeitas: o uso das aplicações por parte do usuário deve envolver características de repetição de ações, para que o assistente passe a identificar as repetições e assimilar as ações, sendo que essas devem ser potencialmente diferentes, ou executadas de várias maneiras por diversos usuários. Caso os usuários executem suas ações de maneira idêntica, a abordagem baseada no conhecimento produzirá resultados mais satisfatórios que a abordagem agora apresentada. Da mesma forma, se não houver repetição freqüente de ações, o sistema não identificará regularidades na execução dessas ações e por conseguinte não será capaz de aprender através de exemplos do usuário.

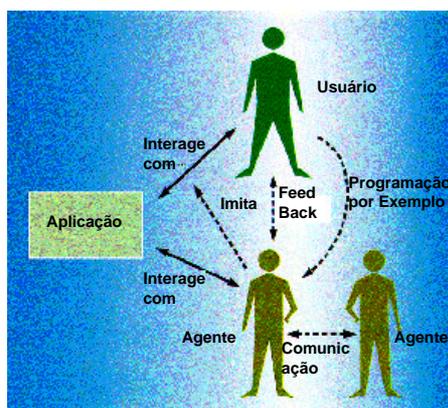
A idéia proposta por Maes é lançar um assistente inicialmente com poucos conhecimentos, apenas os necessários para que consiga, aliado a técnicas de aprendizagem, adquirir outros. Desta forma, o assistente vai se tornando cada vez mais familiar com os métodos de trabalho dos usuários, solucionando o problema da confiabilidade, pois, gradualmente, pode desenvolver novas habilidades e construir novos modelos para tomada de decisões, dependendo da conduta do usuário.

Quanto ao problema da competência, Maes propõe que o assistente a adquira através de quatro diferentes meios:

- Monitorando ações do usuário, desta forma o assistente estará aprendendo de maneira contínua a partir do próprio usuário;
- Através de ações diretas ou indiretas do usuário para com o assistente, neste caso o usuário rejeita uma sugestão do assistente desfazendo alguma ação proposta ou informa, diretamente que este deve assimilar tal fato.
- Através do treinamento direto do usuário, através de exemplos, o usuário pode informar ao assistente suas preferências e ações normalmente executadas.
- Através da troca de informações com entidades que assistam a outros usuários e que possuam mais experiência na realização da mesma tarefa.

Na figura 6, é apresentado um esquema gráfico de como pode ser estruturado o sistema e permitir que o assistente possa adquirir conhecimento.

Figura 6 – Agentes Assistentes e aquisição de conhecimento – [Maes,1997]



Como pode ser percebido nesse esquema gráfico, o agente assistente é capaz de trocar informações seja com a própria aplicação que o usuário utiliza, seja diretamente com o usuário, ou seja, ainda com outros agentes que podem ter mais experiência em alguma área de domínio na qual este agente ainda não possua um nível de informações satisfatório.

Este enfoque leva à concepção de comunidades de agentes, cada qual com algum tipo

de especialidade. Essas comunidades, atuando de maneira colaborativa através da troca de informações a respeito de seus conhecimentos específicos, podem direcionar a uma solução mais rápida e eficiente do problema.

3.6 Conclusão

Com as informações vistas no capítulo, podemos concluir que a interação entre homem e computador é de fundamental importância para que se obtenha melhores resultados com o uso destas máquinas.

Essa interação vem se aprimorando a cada nova fase pela qual passa. Inicialmente, com a interface baseada em comandos de linguagem, a seguir com a interface baseada na manipulação direta e depois com a proposta de se trabalhar com uma interface que, sob certo ângulo de visão, pense e aja por si só, trabalhando cooperativamente com o usuário e disparando a execução de tarefas que julgue necessárias para a obtenção de algum objetivo identificado.

Essa nova visão de interação entre homem e computador está fundamentada no uso de técnicas de inteligência artificial e no uso da tecnologia de agentes, a fim de propiciar ao mecanismo competência e confiabilidade ao tomar alguma decisão ou atitudes de maneira autônoma, proporcionando ao usuário conforto e segurança na delegação de tarefas a estes mecanismos que se utilizem de técnicas de gerenciamento indireto.

4. MODELO PROPOSTO

Atualmente, percebe-se uma necessidade premente por automação de serviços. A principal razão para essa necessidade reside no fato de que o cliente solicitante não dispõe, ou não quer dispor, de tempo hábil para solicitar tais serviços na maneira tradicional, ou seja, deslocando-se até ao local da prestadora e contratando o serviço.

Por parte da prestadora, os serviços devem ser oferecidos garantindo-se dentre outras características, rapidez, agilidade, qualidade e comodidade ao cliente, objetivando maior competitividade com o mercado em que atua.

A automação apenas do recebimento dos serviços pela prestadora não garante maior competitividade, uma vez que a avaliação da qualidade se dá no processo como um todo, desde o recebimento até a apresentação do resultado do serviço prestado. Pode-se então garantir comodidade ao cliente ao receber um pedido através de canais eletrônicos, tais como Internet ou diversos tipos de terminais de atendimento, mas na execução e resposta do serviço solicitado é que se encontram as maiores barreiras.

Quanto maior for o nível de automação na prestação de serviços, menores serão as ocorrências de falhas na execução ou atrasos e imprevistos, devido a padronização que a automação impõe na execução das tarefas. Quando o ambiente não está automatizado estes problemas podem ser mais frequentes.

Exemplos de sucesso nesse tipo de automação envolvem empresas atuantes nos mais diversos segmentos do mercado, cabendo citar a rede bancária, agências de viagens, comércio de produtos, dentre outros.

Há um fator agravante neste contexto: a abertura dos mercados internacionais ou a globalização dos negócios, exigindo que as organizações se tornem cada vez mais competitivas, definindo inclusive o sucesso ou não da empresa.

Uma maneira eficiente na execução de tarefas e, num âmbito mais global, na prestação de um serviço, pode ser encontrada na utilização de programas de computador, ou softwares, que tenham características específicas capazes de tratar de maneira automática e autônoma os eventos que vierem a ocorrer, aliado à teoria do gerenciamento indireto.

Este software deve ser capaz de resolver por si só todos os possíveis eventos que por ventura venham a ocorrer durante a prestação de um serviço solicitado. Desta forma, deve contemplar em seu modelo, características especiais que lhe proporcione autonomia para a tomada de decisões, que pode resultar por exemplo, numa ação direta sobre o sistema, sem que haja a necessidade de uma interferência ou aprovação de

algum usuário, ou ainda, comunicar-se com outras entidades a fim de obter informações que eventualmente sejam necessárias.

4.1 Apresentação do Modelo

O modelo se propõe a auxiliar na automação da prestação de serviços no sentido de minimizar o envolvimento de pessoas com a execução de serviços. Os serviços devem ter sua forma de execução padronizada, a fim de que possam ser executados de maneira automática, objetivando minimizar falhas, bem como, maximizar a agilidade e eficiência na sua execução.

Para que um serviço possa ser executado automaticamente, ou com o mínimo de interferência humana, deve-se dispor de algum mecanismo que registre como o mesmo deve ser executado. O modelo propõe a construção de uma base de conhecimentos onde estará registrado, em forma de regras, o conhecimento para a sua execução, bem como, regras para eventos inesperados que possam ocorrer durante alguma etapa.

Além do registro das regras que compõem a execução de um serviço, devem ser registradas as solicitações desses serviços, bem como todos os passos de sua execução, permitindo que o modelo responda, a qualquer instante, qual a situação em que se encontra a execução de um serviço ou se houve algum problema com a execução de alguma etapa. Esse registro se dá através de uma arquitetura de quadro-negro (blackboard), formada por uma estrutura de dados que permite aos demais elementos do modelo consultar ou depositar informações a respeito do andamento dos serviços solicitados.

O modelo propõe ainda a construção de uma sociedade de agentes específicos capazes de executar os serviços através da interação com a base de conhecimentos, que é o mecanismo que informa ao agente as regras para a execução de cada etapa do serviço, e também da interação com a arquitetura de quadro-negro.

Esta sociedade de agentes é composta por elementos com capacidades específicas para cada etapa que deva ser executada, estando presentes agentes de interface, responsáveis pela comunicação com os usuários externos ao modelo e agentes de informação, responsáveis pela busca de informações a entidades externas e que sejam necessárias à execução das etapas.

Os agentes pertencentes a esta sociedade são construídos sob o enfoque da teoria do gerenciamento indireto, enfatizando características de autonomia, pró-atividade, comunicabilidade e habilidade social, permitindo um comportamento que visa maximizar a pró-ação e a autonomia desses elementos.

No que diz respeito à execução de um serviço por parte da sociedade de agentes, os elementos estão organizados em quatro grupos principais de ação. São eles:

- Interface – Agentes que provêm a comunicação entre os usuários externos e os agentes do modelo.
- Coordenadores – Agentes que coordenam a execução das etapas de um serviço.
- Executores – Agentes que executam a etapa designada pelo agente coordenador.
- Busca – Agentes que auxiliam os agentes executores na busca de informações necessárias.

4.2 Características dos Softwares Agentes empregadas no Modelo

Como visto no capítulo 2 seção 2.4, os agentes possuem uma série de características, as quais muitas vezes extrapolam o ambiente computacional, estando também relacionadas com os campos da psicologia e sociologia, devido ao fato de que os agentes inteligentes procuram substituir a atividade humana nas tarefas a que se propõem executar.

A capacidade de um agente está relacionada às características que a ele estão agregadas, características essas que estão organizadas nos dois grupos principais, softwares com fraca noção de agente e softwares com forte noção de agente, já discutidos no capítulo 2. Desta forma, um agente não necessariamente deve ter todas as características pertencentes aos agentes, e geralmente não as tem, mas apenas aquelas as quais as tarefas a serem executadas por ele exigirem.

Através do emprego da tecnologia de agentes, pode-se valer de características como autonomia, comunicabilidade, pró-atividade, colaboração, dentre outras e aplicar o conceito envolvendo o gerenciamento indireto, construindo um modelo para um software capaz de prestar serviços com a mínima interferência do homem no andamento do processo. No modelo proposto, estão presentes algumas dessas características, destacando-se:

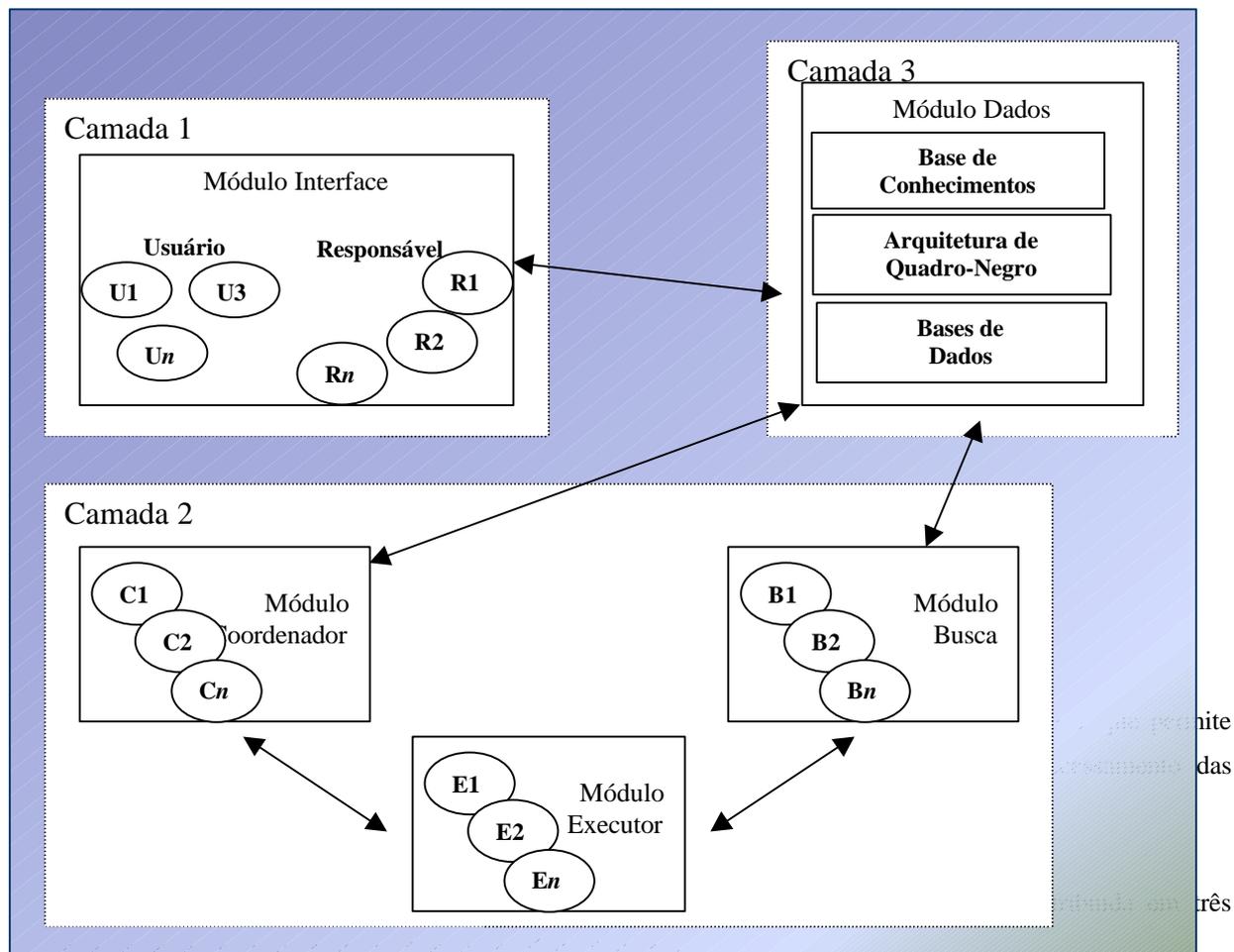
- Autonomia: imprescindível para prover a estabilidade necessária à sociedade dos agentes enquanto entidades autônomas pertencentes ao sistema.
- Comunicabilidade: característica esta muito exigida devido à arquitetura do modelo, composta por módulos independentes, porém com um volume constante de troca de mensagens entre estes módulos.
- Colaboração e Habilidade Social: Componente imprescindível num modelo que se utilize da resolução distribuída de problemas. Esta característica está presente em todos os módulos do sistema, constituindo-se de característica muito empregada pelo modelo.

- Pró-atividade: A pró-atividade é uma característica de suma importância para a aplicação da teoria de gerenciamento indireto, permitindo aos agentes agirem ‘por si próprios’ em determinadas circunstâncias.
- Adaptabilidade: uma vez que a execução de tarefas pode se configurar de diferentes formas dependendo dos parâmetros fornecidos e da situação atual do ambiente, essa característica também se faz presente no modelo.

4.3 Arquitetura do Modelo Proposto

A arquitetura deste modelo baseia-se numa sociedade de agentes e num conjunto de informações agrupadas em módulos de atuação que atendem às necessidades específicas dos seus elementos componentes, conforme ilustrado na figura 7.

Figura 7 – Arquitetura Geral do Modelo



4.4 Tipologia de Agentes utilizada no Modelo

Conforme visto no capítulo 2 seção 2.4, os agentes são classificados de acordo com as características que possuem, sendo agrupadas essas características em tipos de agentes.

Neste modelo, são encontrados os seguintes tipos de agentes:

- Agentes de Interface

Os agentes de interface atuam no modelo realizando a comunicação entre os usuários externos e a arquitetura de quadro-negro, promovendo a troca de informações entre os usuários e o modelo.

- Agentes de Informação:

Uma das necessidades do modelo se configura na busca de informações necessárias à execução dos serviços. Como essas informações podem ser mantidas por diversas fontes, se faz necessário a existência de agentes com capacidade para realizar as buscas e entregar essas informações ao agente solicitante.

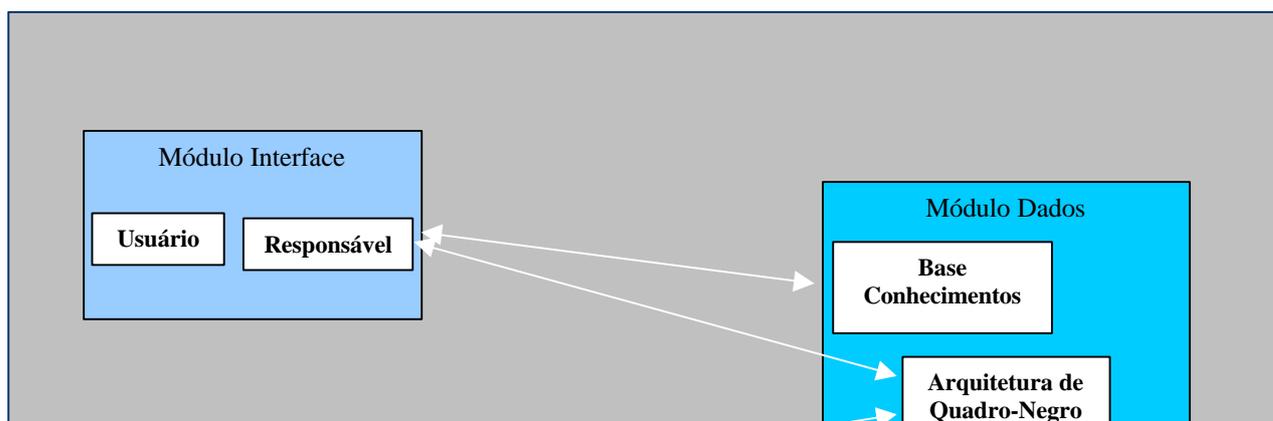
- Agentes Colaborativos:

A colaboração entre os agentes é uma característica fundamental para a manutenção da sociedade de agentes proposta neste modelo. Os agentes colaborativos atuam de forma a permitir o estabelecimento da cooperação entre os agentes que executam o serviço. Desta forma o modelo necessita geralmente acionar mais de um agente para realizar um serviço, através da característica da colaboração o serviço é executado.

4.5 Módulos e Elementos Componentes

A arquitetura do modelo é composta por cinco módulos distintos, com funções específicas conforme ilustrado na figura 8.

Figura 8 – Módulos e Elementos do Modelo



- **Módulo Interface:**

O módulo Interface representa os elementos responsáveis pela comunicação do modelo com os usuários externos. Através desse módulo, o usuário pode interagir com o modelo, depositando ou solicitando informações a arquitetura de quadro-negro. Os elementos que compõem esse módulo são agentes de interface que possuem duas funções distintas:

Agentes de Interface com o usuário: Esse agente é responsável pela interação do modelo com o usuário solicitante de um serviço. Através dele, o usuário pode solicitar ou cancelar uma solicitação ou, ainda, solicitar informações a respeito da situação em que se encontra o seu pedido. Dessa forma, o agente de interface com o usuário também consulta a arquitetura de quadro-negro em busca de informações. Esse agente, ao solicitar um serviço, consulta a base de conhecimentos em busca das etapas a serem cumpridas para a execução daquele tipo de serviço e as registra na arquitetura de quadro-negro, utilizada na execução do serviço.

Agentes de Interface com o Responsável: Determinadas etapas na execução de um serviço podem requerer a intervenção de algum usuário que seja responsável por aquela etapa ou serviço, como por exemplo a autorização da execução da etapa mediante uma situação anormal ou duvidosa encontrada pelo agente executor. Nesse caso, o usuário responsável por aquela etapa é notificado através desse agente, o qual mantém a execução do serviço suspenso até a decisão ser tomada pelo usuário responsável. O usuário pode acionar esse mesmo agente e informar ao modelo a informação solicitada.

- **Módulo Coordenador:**

Neste módulo estão presentes os elementos que são responsáveis por coordenar a execução dos serviços, sendo responsáveis por:

- Verificar a existência de novas solicitações de serviços na arquitetura de quadro-negro registrados pelo agente de interface com o usuário;
- Repassar ao agente executor uma nova etapa a ser executada;

- Registrar na arquitetura de quadro-negro a situação em que se encontra a execução da etapa.

- **Módulo Executor:**

Compete a esse módulo a execução das etapas propriamente dita. Os elementos que o compõem, os agentes executores, devem ter objetivos claros, tendo sua construção voltada para a execução de tarefas específicas, com total domínio da forma e dos recursos necessários que a execução da etapa exige, constituindo-se, assim, de vários elementos cada qual com a habilidade para executar uma etapa específica. Caso houver necessidade, um agente executor pode acionar um outro agente executor, a fim de manterem troca de informações ou colaborarem de alguma forma para a execução da etapa.

O módulo Executor, através do agente executor, aciona o módulo busca sempre que houver a necessidade de informações para dar prosseguimento à execução da tarefa.

Assim que a etapa é executada, o módulo coordenador é comunicado através do agente coordenador.

- **Módulo Busca:**

Ao módulo busca cabe a localização e o resgate das informações requisitadas pelo módulo executor através de um agente executor. Esse módulo é formado por agentes de informação, cada qual responsável por localizar um tipo de informação específica. Dessa forma, temos também agentes específicos para localizar cada tipo de informação. Após localizadas e resgatadas, as informações são enviadas ao agente executor solicitante.

- **Módulo Dados:**

O módulo dados representa as informações necessárias à manutenção do funcionamento do modelo a respeito do conhecimento da execução dos serviços, do andamento dos serviços e informações a respeito do estado interno dos elementos do modelo além de representar também as entidades externas detentoras de informações que o modelo necessita para executar os serviços. Compõem o módulo dados três elementos distintos:

- *Base de Conhecimentos:* Consiste num conjunto de regras que norteiam a execução dos serviços. Essas regras formam o conjunto de ações a serem executadas nas diversas etapas para a realização do serviço.
- *Arquitetura de Quadro-Negro:* Essa arquitetura consiste numa estrutura de dados onde os elementos do modelo podem depositar e consultar informações. Assim sendo, os agentes coordenadores registram nessa estrutura de dados o estado da

execução de cada etapa das solicitações, bem como podem manter um controle sobre quais agentes estão em atividade num determinado instante.

- *Base de Dados Externa ao Modelo:* Conforme visto na descrição do módulo busca, sempre que houver a necessidade de informações externas ao modelo, os agentes de busca são acionados, localizando e resgatando as informações. Essas bases de dados são formadas por entidades que não mantêm um vínculo direto com a arquitetura do modelo, mas são de fundamental importância, uma vez que cabe a estas entidades fornecerem informações de que o modelo necessita.

4.6 Organização da Sociedade de Agentes no Modelo

As sociedades de agentes são classificadas de acordo com a forma como se organizam os elementos internos, os agentes, na busca da solução para os problemas.

A forma de organização dos elementos da sociedade na busca de seus objetivos pode seguir a abordagem da Resolução Distribuída de Problemas ou a abordagem que trata de Sistemas Multi-Agentes.

4.6.1 Resolução Distribuída de Problemas

Solucionar problemas nada mais é do que caminhar. Partir de um ponto de saída e percorrer um caminho até a chegada. A solução é, portanto, o caminho e um método de solução, uma forma de construir caminhos e andar por eles. Esse caminhar se desenvolve em um espaço e esse espaço é justamente o problema, e a representação do espaço é a representação do problema. Cada problema determina seu próprio espaço, configurado por seus possíveis resultados, por seus resultados corretos, por seus lugares de partida, e por seus diferentes caminhos que se podem traçar e recorrer enquanto que se traçam [CORTES,1993].

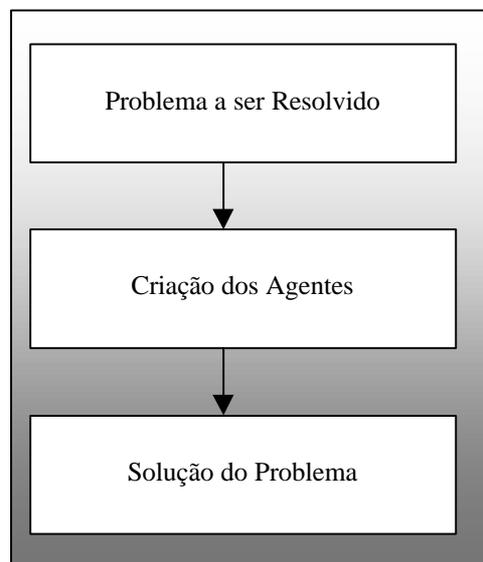
A solução consiste no caminho, no espaço (problema) e em seus resultados. Para que se consiga a solução deve-se apoiar em métodos que são utilizados para construir os caminhos, através de uma busca pelos estados do espaço(problema), que ligam os estados iniciais do problema (espaço) com os estados finais.

Uma busca distribuída requer um sistema de controle, alguns operadores e um espaço de estados particionado, onde vários elementos do processo podem realizar buscas simultâneas nas diferentes partes desse espaço. Os resultados destas buscas são, então, analisados e comparados a fim de gerarem uma resposta.

A Inteligência Artificial Distribuída pode também se configurar como uma resolução de problemas cooperativos, segundo Lesser [LESSER,1994], consistindo numa rede de nós autônomos ou semi-autônomos que trabalham em conjunto para resolver um problema único, através da cooperação e do compartilhamento de informações disponíveis nos nós.

Com a resolução distribuída de problemas, os agentes são construídos e orientados de forma a resolver um problema específico ou classe de problemas restritos, através da cooperação, divisão de tarefas e compartilhamento do conhecimento com outros agentes a respeito do problema e de como solucioná-lo. A figura 9 representa esta abordagem.

Figura 9 – Resolução Distribuída de Problemas



Da mesma maneira que os humanos superam suas limitações individuais associando-se a grupos de pessoas de forma a compartilhar informações e dividir tarefas complexas em outras tarefas mais simples, os sistemas de agentes inteligentes baseiam-se na idéia de superar as limitações de qualquer sistema inteligente através do mecanismo de agrupamento de sistemas mais simples, compartilhando suas capacidades específicas e conhecimentos sobre o assunto.

Na resolução distribuída de problemas, a solução é buscada através da decomposição de um problema em vários subproblemas que representam as metas a serem alcançadas, tendo um controle centralizado, o qual é responsável por coordenar as atividades que venham a solucionar o problema.

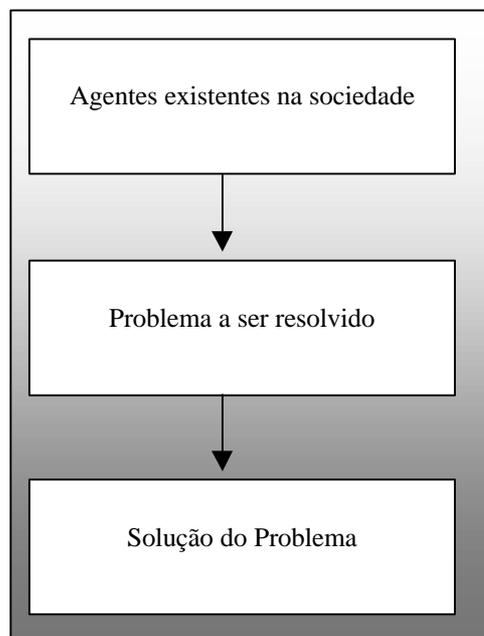
4.6.2 Sistemas Multi-Agentes

Os sistemas multi-agentes consistem na coordenação de um grupo de agentes autônomos, segundo seu comportamento inteligente, em que a própria existência destes elementos pode ser anterior ao surgimento de um problema naquele domínio de atuação.

Os sistemas multi-agentes têm seu enfoque voltado não para a resolução de um problema especificamente, mas para a visão de um domínio, onde várias situações podem ser configuradas, tendo uma arquitetura mais flexível. A organização dos elementos componentes pode mudar devido a variações no ambiente ao qual está inserido ou do problema que se está resolvendo.

Como visto, num sistema multi-agente os agentes podem tratar de problemas surgidos após sua criação, através da percepção do ambiente. Para permitir esta flexibilidade ao sistema, os agentes são agrupados em uma sociedade de agentes autônomos, onde a coordenação do comportamento está baseada no compartilhamento das metas, capacidades, planos de ação e do conhecimento adquirido. Pode-se dizer, então, que o problema é apresentado à sociedade de agentes para que a mesma busque a solução. A figura 10 representa esta abordagem.

Figura 10 – Sistemas Multi-Agentes



Flávio Oliveira [OLIVEIRA,1996], comenta sobre a problemática na coordenação quando ocorrem conflitos entre agentes num sistema multi-agente. Sempre que houver um conflito, os agentes deverão entrar num processo de negociação a fim de sanar as posições conflitantes, seja para definição de objetivos ou até mesmo para estabelecer consistência aos seus conhecimentos.

Em seu trabalho Durfee e Rosenschein [DURFEE & ROSENCHIN,1994], não isolam as duas áreas de Resolução Distribuída de Problemas e Sistema Multi-agentes, mas procuram estabelecer alguns relacionamentos entre ambas, em que se configuram as seguintes situações:

- O sistema multi-agente garante propriedades como metas comuns e projeto centralizado, tendo o resolvidor de problemas distribuídos como um subconjunto do sistema multi-agente.
- O solucionador de problemas distribuídos detém a visão global do sistema, tendo como base o sistema multi-agente, que tratará da individualidade de cada agente no seu relacionamento com os

demais. Assim, o solucionador distribuído de problemas atua como coordenador geral, direcionando o sistema para a solução do problema global e não do específico de cada agente.

- O sistema multi-agente e o solucionador distribuído de problemas são entendidos como sistemas complementares. O sistema multi-agente resolve como as propriedades coletivas são percebidas e o solucionador distribuído de problemas resolve como obter um desempenho eficiente num conjunto particular de agentes, tendo as propriedades do ambiente em que está inserido totalmente dinâmicas e não controláveis.

No modelo aqui apresentado, foi definida uma arquitetura de agentes baseada na Resolução Distribuída de Problemas (DPS – Distributed Problem Solved), em que os diversos agentes convivem numa sociedade. Cada elemento é especializado em alguma função, não necessariamente sendo única, mas as funções que são de sua responsabilidade estão sob seu domínio de atuação.

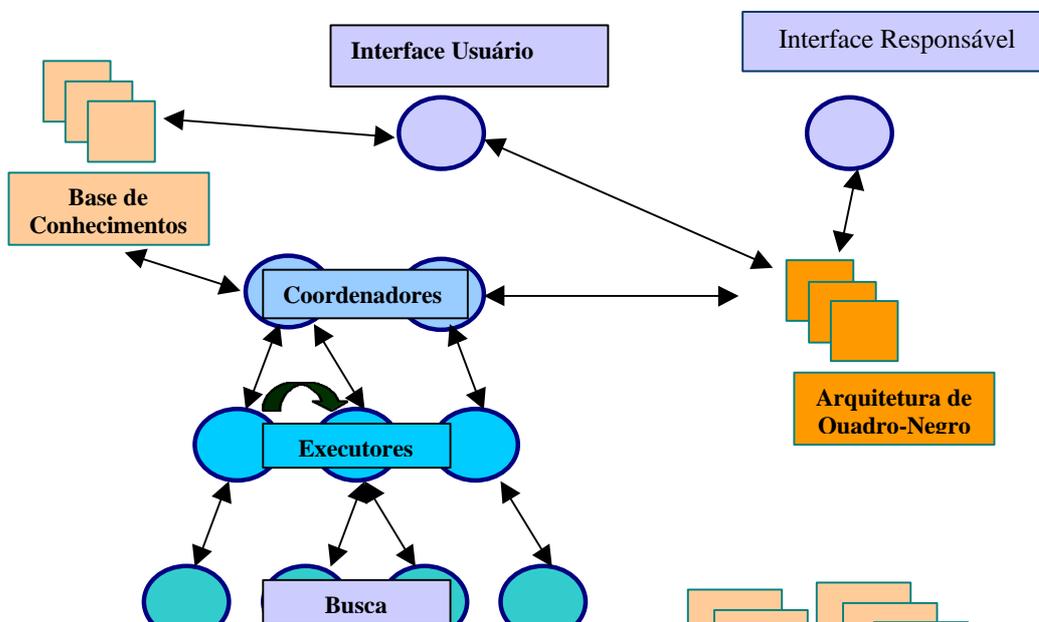
A Resolução Distribuída de Problemas se adequa com a forma de execução de um serviço proposta neste trabalho através da divisão do serviço em etapas e estas delegadas ou distribuídas entre os agentes da sociedade, com uma seqüência de execução pré-estabelecida através das regras existentes na base de conhecimentos.

4.7 Comunicação entre os Agentes do Modelo

A comunicação entre os elementos do modelo deve acontecer de forma simples e direta, visando maximizar a agilidade no fluxo de informações entre os agentes.

A figura 11 ilustra graficamente como ocorre a comunicação entre os elementos pertencentes ao modelo.

Figura 11 – Comunicação entre os elementos do modelo



Este modelo é comandado por uma base de conhecimentos cujos serviços prestados pela comunidade de agentes estão subdivididos em diversas etapas. Dessa forma, a base de conhecimentos através de suas regras, mantém o controle de quais etapas e em que ordem devem ser executadas para a prestação de um determinado serviço.

Além de manter este controle na base de conhecimentos, o modelo dispõe de uma arquitetura de quadro-negro, sendo considerada uma memória de trabalho e utilizada pelos elementos componentes da sociedade de agentes, onde são registradas as informações necessárias à execução dos serviços, como por exemplo a situação de cada etapa dos serviços ou informações adicionais necessárias à execução de determinada etapa.

Através do agente de interface usuário, o usuário registra uma solicitação de um serviço. Esse agente pesquisa, na base de conhecimentos, as regras necessárias à execução das diversas etapas do serviço e as registra na memória de trabalho a fim de que sejam executadas pelos demais componentes da sociedade de agentes.

Os agentes coordenadores de serviços verificam, periodicamente, a memória de trabalho à procura de registros de novas solicitações de serviços. Assim que encontrada, o coordenador que estiver coordenando o menor número de solicitações naquele instante será o responsável por coordenar essa nova solicitação. Assim, o agente coordenador retira da base de conhecimentos a primeira etapa a ser executada para aquela solicitação e a encaminha ao agente executor que esteja ocioso. Esse agente deve ter sobre seu domínio todo o plano de execução da etapa e pode, desde que necessário, acionar outros agentes executores, solicitando colaboração na execução de parte daquela etapa e da mesma forma acionar os agentes de busca a fim de receber as informações necessárias ao cumprimento das etapas.

Aos agentes de busca, cabem realizar as devidas consultas a bases de dados ou ainda a outros sistemas, desde que estejam disponíveis ferramentas que possibilitem tal ação, a fim de suprir as necessidades dos agentes executores.

Uma vez colhidas as informações solicitadas, os agentes de busca as entregam aos agentes executores que, ao completarem a execução da etapa, informam ao agente coordenador, o qual registra na memória de trabalho e retira a próxima etapa a ser cumprida. Caso não existam mais etapas, a prestação do serviço se dá por encerrada.

O agente de interface responsável é acionado sempre que a execução de algum serviço necessitar da interferência de algum usuário do sistema. Esta situação pode ser necessária devido à exigência de registro

específico da autorização ou não da execução do serviço, seja por qualquer motivo que possa impedir a execução normal do mesmo.

Havendo a impossibilidade da execução de alguma etapa, seja por motivos internos ou externos ao domínio do modelo, a memória de trabalho deve ser notificada, cabendo ao módulo Coordenador a responsabilidade de informar ao usuário responsável o motivo do cancelamento da execução do referido serviço.

Este modelo, devido às características de modularidade e de seus módulos serem fortemente acoplados, exige muito da comunicação entre os seus componentes, havendo a necessidade de ser adotada uma linguagem de comunicação que permita a exploração de recursos eficientes no estabelecimento destas comunicações.

4.8 Conclusão

Após a exposição dos elementos componentes do modelo bem como a funcionalidade do modelo como um todo, cabem ainda ressaltar algumas considerações obtidas após a análise da prototipação desse modelo.

O modelo descrito procura viabilizar a execução de serviços de maneira autônoma amparando-se num conhecido ditado: *dividir para conquistar*. Dessa forma, os serviços são divididos em etapas específicas e em quantas etapas forem necessárias, dependendo da complexidade da execução do serviço e dos recursos computacionais disponíveis.

A decomposição dos serviços em etapas proporciona flexibilidade ao modelo, visto que serviços distintos podem se utilizar de algumas etapas comuns na sua execução, permitindo que um agente executor programado para realizar uma etapa possa ser utilizado por diversos serviços.

A comunicação entre os elementos componentes desse modelo é exigida ao máximo constituindo-se de item fundamental para a viabilidade ou não de sua implantação sob o ambiente computacional escolhido. Isso se deve à possibilidade do envolvimento de diversos elementos na execução de uma única etapa, gerando então a necessidade de comunicação entre eles. Porém cabe ressaltar que a maior ou menor necessidade de comunicação não tem uma relação direta com o número de elementos envolvidos na execução das etapas, pois este é um fator definido caso a caso entre os elementos.

O estabelecimento de uma hierarquia entre os módulos Coordenador, Executor e Busca se faz necessária devido à necessidade de se manter um sentido na resolução distribuída de problemas, no que se refere ao fluxo de informações e controle da execução.

A organização da base de conhecimentos é de suma importância para o funcionamento do modelo, sendo que nela residem informações vitais como, por exemplo, os serviços que são prestados, a ordem de

execução das etapas por cada serviço, os elementos que executam determinada etapa, enfim, na base de conhecimentos está toda a organização lógica do modelo.

A arquitetura de quadro-negro, aqui referenciada muitas vezes como memória de trabalho, mantém a configuração do ambiente de trabalho dos elementos da sociedade de agentes, mantendo atualizados o status das etapas em execução, os serviços que não puderam ser executados e o motivo, as informações utilizadas para a execução de um serviço e os elementos que estão atualmente ociosos, dentre outras informações necessárias à prestação dos serviços.

Os *agentes de interface responsável* figuram como uma válvula de escape, acionada assim que houver a necessidade de algum tipo de intervenção no fluxo de execução de algum serviço, sendo necessários em situações que exijam aprovação explícita à execução, ou ainda na entrada de alguma informação de caráter momentâneo.

5 IMPLEMENTAÇÃO DO MODELO

Com base no modelo apresentado no capítulo 4 foi implementado o protótipo de um sistema capaz de atingir os objetivos contemplados pelo modelo.

Tal protótipo tem a finalidade de gerenciar solicitações de serviços para uma secretaria geral de ensino de nível superior, objetivando prestar o serviço com a mínima intervenção do corpo de funcionários da secretaria bem como de funcionários de outros setores envolvidos no processo.

Desta forma, este software visa permitir que os usuários em geral daquele departamento usufruam dos seus serviços com a máxima comodidade, pois a solicitação pode ser efetuada através de qualquer terminal que possua um canal de comunicação com o software, tendo como exemplo mais popular a Internet.

O que levou a adaptação do modelo para a prestação de serviços numa secretaria geral de nível superior foi a necessidade por um sistema que produzisse ao usuário respostas mais confiáveis no andamento dos processos, uma vez que através deste software o usuário pode consultar qual a etapa da sua solicitação está sendo executada naquele instante ou ainda receber informações de forma automática sobre qualquer atraso na sua solicitação; um maior controle sobre a execução das solicitações do usuário, uma vez que os agentes coordenadores mantêm total controle das operações que estão em andamento; e por fim maior agilidade no atendimento ao público, uma vez que em determinados períodos do ano letivo o volume de solicitações cresce de maneira considerável, resultando em atrasos no atendimento e até mesmo dificultando o usuário em dar entrada à sua solicitação, devido a demanda a ser atendida.

Com a implantação deste protótipo, o usuário poderá, através de qualquer terminal que acesse a Internet, via applets java, dar entrada em serviços que estejam disponíveis, consultar sobre o andamento de sua solicitação e receber sua solicitação sem a necessidade de se deslocar ao departamento, pois o sistema se encarrega de enviar a resposta da solicitação na forma que o usuário preferir, podendo ser via correio eletrônico, caso seja possível, via correspondência normal ou ainda retirar no próprio balcão de atendimento.

As ferramentas utilizadas na implementação do protótipo foram as seguintes:

- Linguagem de desenvolvimento Java
- JATLite – Java Agent Template
- Linguagem de Comunicação entre Agentes KQML
- JDBC – Java Database Connectivity

Em Anexos, no capítulo 8 sessão 8.2, contém uma descrição das principais características que levaram ao uso de cada uma destas ferramentas.

5.1 Serviços Contemplados pelo Protótipo

O protótipo implementado tem seu foco de atuação na prestação de serviços realizados por uma Secretaria Geral de Ensino para uma instituição de nível superior. Este departamento mantém estreitas relações com diversos outros setores, sejam eles pertencentes ao meio acadêmico ou administrativo da instituição, e até mesmo com entidades externas ao meio universitário, como por exemplo o MEC, Secretarias de Educação, dentre outros.

Diante da diversidade de serviços executados por uma Secretaria Geral de Ensino destas instituições, foram contemplados pela implementação do protótipo os serviços abaixo relacionados, estando subdivididos em três categorias:

- **Documentos**
 - Emissão de Histórico Escolar
 - Emissão de Atestado de Freqüência
 - Emissão de Atestado de Matrícula
 - Emissão de Atestado de Vaga
 - Emissão de Certificado de Conclusão
 - Emissão de Revisão de Currículo

- **Entrada de Processos**
 - Processo de Trancamento de Disciplina
 - Processo de Troca de Curso
 - Processo de Reingresso de alunos
 - Processo de Revisão de Prova
 - Processo de Reaplicação de Prova

- **Informações ao Usuário**
 - Informações de Notas Semestrais
 - Informações da Situação Atual do Processo
 - Informações de Espelho de Matrícula
 - Informações de Grade Horária dos Cursos

A escolha destes não foi meramente casual, mas apontados pela Secretaria Geral de Ensino da UNISUL, instituição onde o protótipo foi concebido, como serviços críticos e que necessitam de maior agilidade na sua prestação, uma vez que envolvem diretamente os alunos da instituição, representando um grande número de clientes do departamento, e que produzem maior demanda de serviços.

5.2 Prestação dos Serviços

Para que o sistema possa operar, são necessárias informações mínimas à base de conhecimentos, proporcionando aos agentes subsídios para realizarem seus trabalhos. Estas informações consistem em:

- Subdivisão dos serviços a serem prestados em etapas específicas e que sejam de domínio de pelo menos um agente da sociedade.
- Cadastramento de Responsáveis pela emissão dos serviços
- Cadastramento de prazos mínimos para a prestação dos serviços
- Cadastramento da agenda de despachos para os documentos

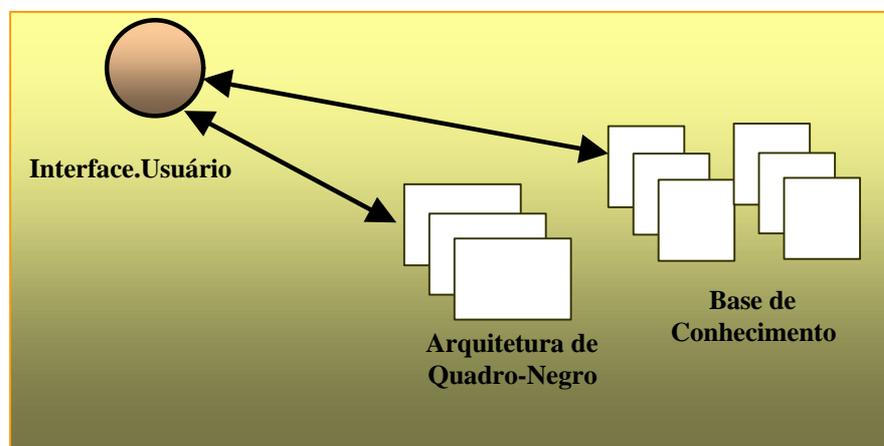
Uma vez alimentada a base de conhecimentos o sistema pode entrar em operação.

A sociedade de agentes é ativada no momento em que o usuário solicita algum serviço através do agente de Interface Usuário.

Este agente consulta a base de conhecimentos verificando se o serviço solicitado necessita de autorização explícita de algum funcionário responsável pelo serviço. Em caso afirmativo, verifica se este funcionário possui intenção de despachar documentos até a data limite do prazo de entrega do serviço e informa ao usuário a data em que o serviço estará concluído.

Este serviço é registrado na memória de trabalho, formada por uma arquitetura de quadro-negro, a qual registra as etapas a serem cumpridas, resgatadas através das regras mantidas pela base de conhecimentos, para que o serviço seja prestado por completo.

Figura 12 – Interface do Usuário

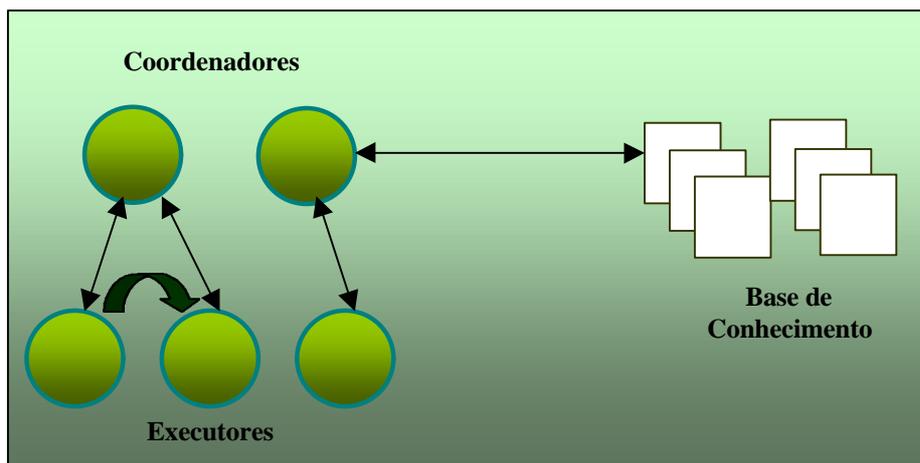


A partir de então, um agente tipo *coordenador* é acionado para coordenar a execução da solicitação. Tal agente retira cada etapa registrada na base de conhecimentos, registrando seu estado atual. O estado de uma etapa pode assumir os seguintes valores:

- Aguardando execução
- Executando
- Executada
- Impossibilitada/Interrompida

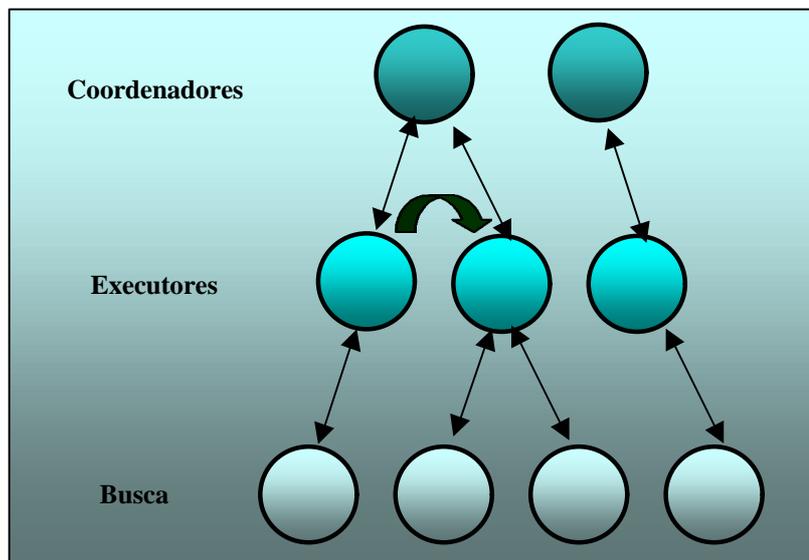
Ao retirar uma etapa da base de conhecimentos, o agente coordenador aciona um agente *executor*, o qual possua domínio suficiente para executar a etapa.

Figura 13 – Agentes Coordenadores e Base de Conhecimento



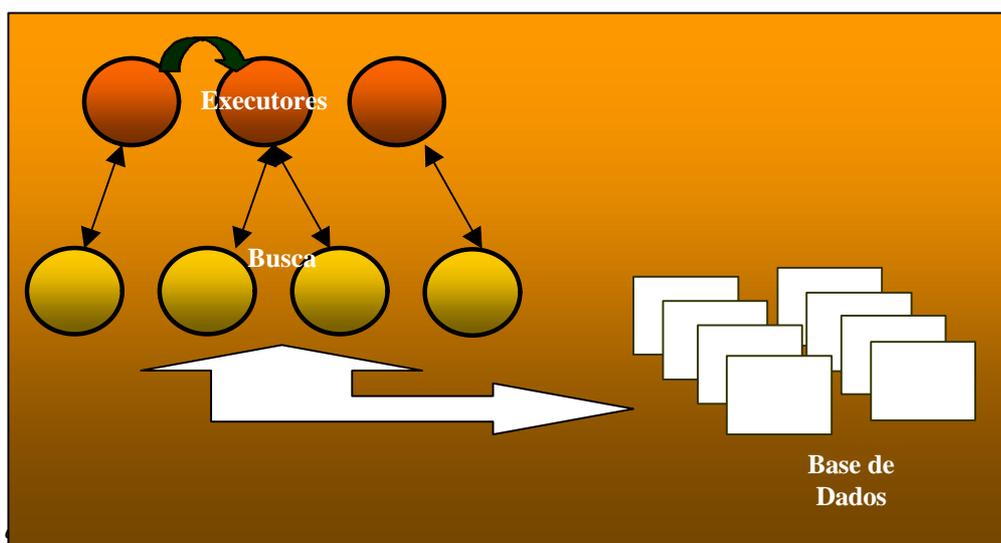
O agente executor respectivo passa a ser acionado e executa a etapa solicitada. Durante a execução, o agente pode solicitar auxílio de outros agentes executores ou ainda solicitar aos *agentes de busca* informações necessárias à execução daquela etapa. Uma vez realizada a etapa, o agente executor informa ao agente coordenador daquela solicitação o compromisso cumprido.

Figura 14 - Agentes executores



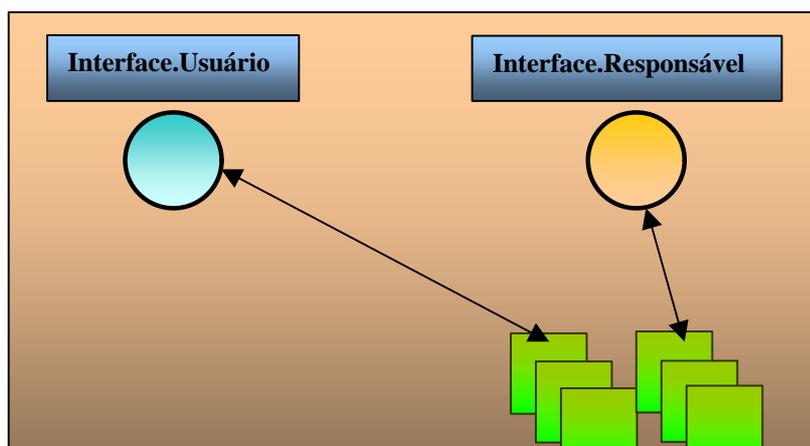
O agente de busca por sua vez é responsável por resgatar informações em seu domínio de atuação. Este, através de interfaces JDBC aciona consultas nas bases de dados disponíveis, bem como na base de conhecimentos, devolvendo ao agente executor as informações solicitadas.

Figura 15 – Agentes de busca



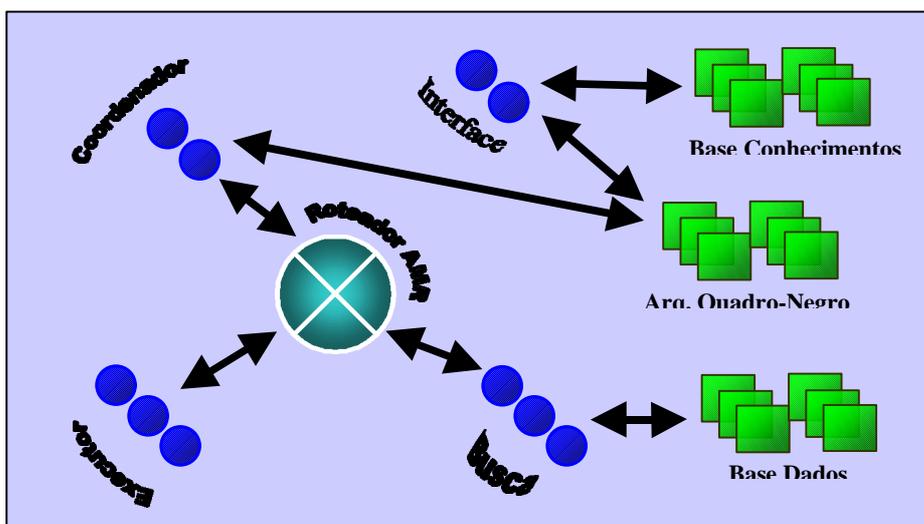
Ao longo do processo de aprovação de alguma etapa à arquitetura de quadro-negro. Tal informação ou aprovação consistem em procedimentos de análise realizados pelos responsáveis pela prestação do serviço, os quais estão aquém das capacidades do sistema.

Figura 16 – Agentes de Interface Responsável e Base de Conhecimento



No centro do sistema encontra-se o Roteador AMR, cujas características encontram-se descritas em Anexos, no capítulo 8 sessão 8.2, responsável por permitir de maneira simples e ágil a comunicação entre toda a sociedade dos agentes.

Figura 17 – Sociedade de Agentes e Roteador AMR



5.3 Execução dos Serviços Implementados

A seguir estão relatados a execução de alguns dos serviços descritos na sessão 5.1, discriminando a atuação dos elementos componentes do modelo no protótipo para a realização dos objetivos de cada processo. Será abordada a execução de um serviço para cada uma das categorias descritas na sessão 5.1, devido a similaridade na execução dos serviços para cada categoria, se utilizando dos mecanismos disponíveis de maneira idêntica ou com pouca variação.

A execução dos referidos serviços só pode ser realizada após prévia identificação do usuário através da interface ilustrada pela figura 18.

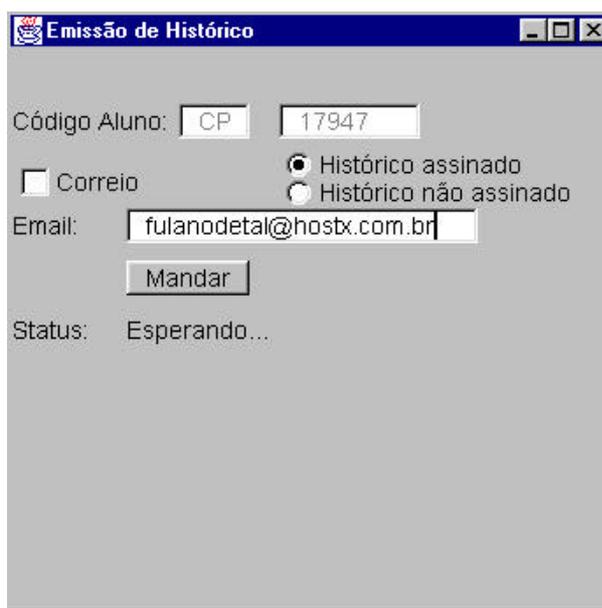
Figura 18 - Interface principal do protótipo



EMISSÃO DE DOCUMENTOS: Histórico Escolar

Através da interface ilustrada pela figura 19, o usuário pode solicitar a emissão de dois tipos de documentos: Histórico assinado, constituindo de um documento legal da instituição, devidamente assinado e autorizado, e o Histórico não assinado, onde será emitida uma relação das disciplinas e notas obtidas para as disciplinas em que o aluno já cursou, válido para simples conferência. Devido a similaridade na geração de ambos os documentos, estes foram agrupados numa única interface e o serviço é executado pelos mesmos agentes.

Figura 19 – Emissão de Histórico



Emissão de Histórico

Código Aluno: CP 17947

Correio Histórico assinado
 Histórico não assinado

Email: fulanodetal@hostx.com.br

Mandar

Status: Esperando...

Ao enviar o pedido, este é registrado na arquitetura de quadro-negro pelo agente de interface Usuário, a agenda de despachos do responsável por assinar o documento é consultada e retorna uma data prevista para a retirada do documento. Este agente subdivide os serviços em diversas etapas obtidas através da consulta na base de conhecimentos. Uma vez registradas as etapas na arquitetura de quadro-negro a serem cumpridas, dá-se início a execução das mesmas. O agente coordenador retira a primeira etapa a ser executada, notificando a arquitetura de quadro-negro de que esta etapa encontra-se em execução. É acionado então o agente executor responsável por aquela etapa específica, o qual detém o conhecimento de como executá-la, acionando, sempre que necessário, os agentes de busca para obter informações necessárias ao cumprimento da etapa. Uma vez executada, o agente coordenador é comunicado, o qual a registra como concluída, retirando a próxima etapa a executar.

As etapas necessárias a execução deste serviço são:

- *Resgate de dados*

Esta etapa consiste na verificação e coleta dos dados necessários ao documento, envolvendo agentes executores e de busca. Sempre que houver a necessidade de informações que não estão em poder do agente executor, os agentes de busca são acionados.

- *Geração do documento*

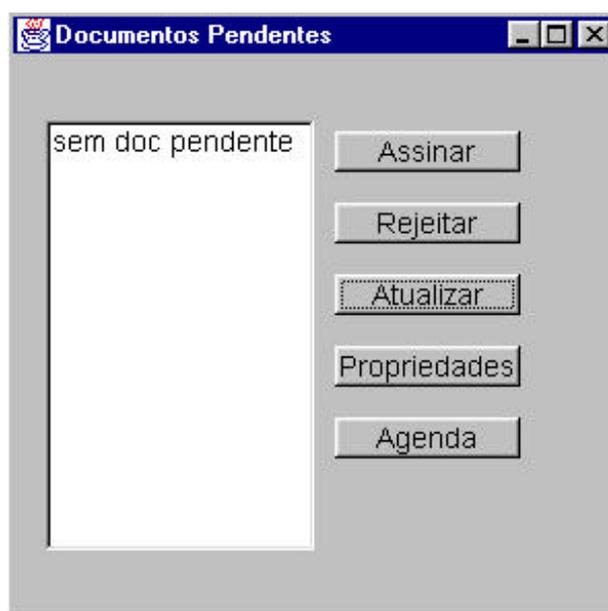
Uma vez em poder dos dados para a execução da etapa, os agentes executores constroem o documento propriamente dito, realizando as devidas formatações de acordo com um modelo previamente estabelecido.

- *Autorização*

Esta etapa é executada apenas para documentos que necessitam de autorização explícita, como por exemplo o histórico assinado, sendo realizada por um agente específico que provê ao usuário responsável informações a respeito do documento a ser autorizado.

Para executar esta etapa, o usuário responsável se identifica perante o sistema e o agente Usuário Responsável é acionado, o qual pesquisa na arquitetura de quadro-negro quais documentos necessitam autorização. Conforme ilustrado pela figura 20 o responsável tem disponível as opções de: Assinar o documento, onde o responsável autoriza a emissão do documento, imprimindo-o para assinatura e registro legal; Rejeitar, onde o usuário responsável rejeita a emissão do documento, indeferindo o pedido; Atualizar, opção que atualiza a interface com os últimos documentos que por ventura não estão sendo exibidos; Propriedades, utilizada na visualização para apreciação e conhecimento do documento; Agenda, permite ao usuário responsável o registro de sua agenda para despachos de documentos e/ou processos. Estas datas são utilizadas para fornecer ao solicitante uma previsão de conclusão do documento.

Figura 20 – Documentos Pendentes



- *Envio*

Uma vez autorizada a emissão do documento, o funcionário da Secretaria Geral de Ensino, responsável pelo envio do documento, é informado por meio de correio eletrônico da autorização do documento, bem como procedimentos para envio ao destino, opção esta que o solicitante faz ao enviar o pedido, podendo optar por receber o documento via postagem normal ou retirar no balcão de atendimentos. Um comunicado é enviado também ao solicitante a respeito da decisão tomada, seja o documento autorizado ou rejeitado.

SOLICITAÇÃO DE INFORMAÇÕES: Notas Semestre

Por meio da interface principal do sistema, demonstrada pela figura 18, o usuário pode solicitar as suas notas semestrais, acionando o agente de interface ilustrado pela figura 21.

Figura 21 – Solicitação de Notas Semestre



A imagem mostra uma janela de diálogo com o título "Notas no Semestre". No topo, há uma barra azul com o ícone de uma pasta e o texto "Notas no Semestre". Abaixo, há dois campos de entrada: "Código Aluno:" com o valor "CP" e "17947", e "Semestre:" com um campo vazio. Um botão "Mandar" está centralizado abaixo dos campos. Na base da janela, o texto "Status: Esperando..." indica o estado atual da solicitação.

Após a solicitação do serviço, ocorre o registro das etapas nos mesmos moldes descrito para a categoria de serviços tipo Emissão de Documento, onde a base de conhecimentos é consultada e as etapas são registradas na arquitetura de quadro-negro.

Neste serviço, são cumpridas as seguintes tarefas:

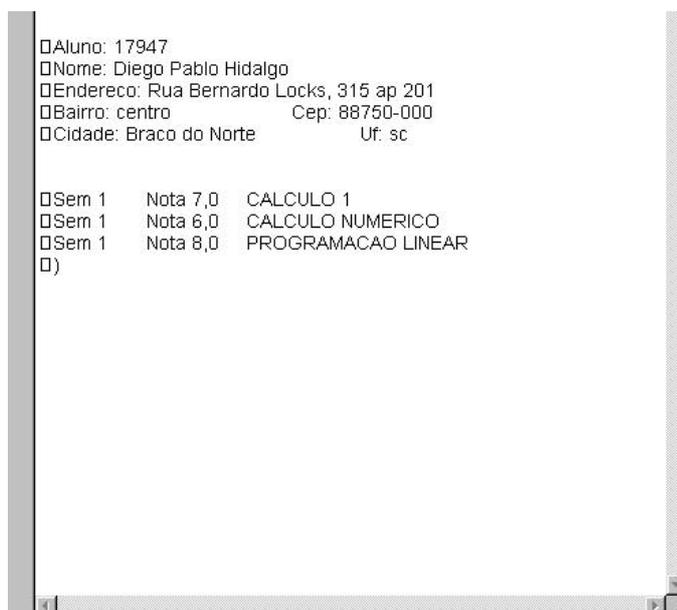
- *Resgate de dados*

Esta etapa consiste na verificação e coleta dos dados necessários ao documento, execução similar a de serviços da classe de Emissão de documentos.

- *Geração de notas semestre*

Nesta etapa, o agente coordenador designa o agente executor para o cumprimento da tarefa. O agente executor aciona os agentes de busca responsáveis pela busca de notas de alunos. Após recebida as informações, é gerada a interface para apresentação das notas, conforme ilustrado pela figura 22.

Figura 22 – Apresentação de Notas



□Aluno: 17947
□Nome: Diego Pablo Hidalgo
□Endereco: Rua Bernardo Locks, 315 ap 201
□Bairro: centro Cep: 88750-000
□Cidade: Braco do Norte Uf: sc

□Sem 1 Nota 7,0 CALCULO 1
□Sem 1 Nota 6,0 CALCULO NUMERICO
□Sem 1 Nota 8,0 PROGRAMACAO LINEAR
□)

- *envio ao solicitante*

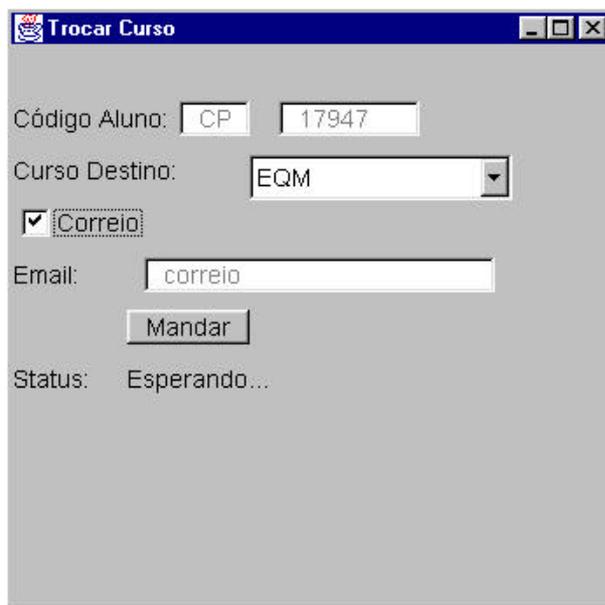
Uma vez gerada a interface com as notas, esta é apresentada ao usuário por meio do agente executor.

O objetivo deste serviço é somente apresentar as notas para conferência e conhecimento.

ENTRADA DE PROCESSOS: Troca de Curso

Este serviço consiste em permitir ou não que o aluno realize a transferência do seu curso de origem para um outro curso existente na instituição. Ao acionar a opção no menu principal, o agente de interface apresenta a tela correspondente para que o aluno informe os dados necessários, conforme ilustrado pela figura 23.

Figura 23 – Interface para Troca de Curso



Trocar Curso

Código Aluno: CP 17947

Curso Destino: EQM

Correio

Email: correio

Mandar

Status: Esperando...

Após a solicitação do serviço, ocorre o registro nos mesmos moldes descrito para a categoria de serviços tipo Emissão de Documento, onde a base de conhecimentos é consultada e as etapas são registradas na arquitetura de quadro-negro. As etapas a serem cumpridas para este tipo de serviço são:

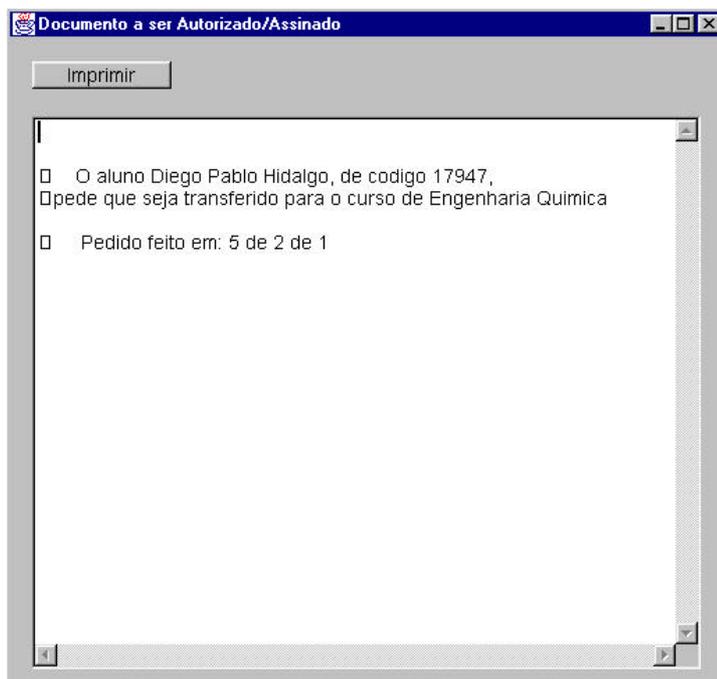
- *Resgate de dados*

Nesta etapa o agente executor realiza uma verificação da situação do aluno no curso de origem, a fim de munir o responsável por autorizar a transferência com algumas informações úteis, tais como semestre letivo do aluno no curso de origem, número de transferências já solicitadas, dentre outras.

- *Geração do documento*

Neste procedimento, o agente executor formula o pedido formal da transferência e o registra na arquitetura de quadro-negro. A figura 24 ilustra este pedido.

Figura 24 – Pedido de Transferência de Curso



- *Aprovação do responsável*

Esta etapa é similar a etapa de autorização, para solicitações da classe de emissão de documentos comentada anteriormente, onde o usuário responsável se identifica perante o sistema acionando o agente Usuário Responsável, o qual pesquisa na arquitetura de quadro-negro quais documentos necessitam autorização, conforme ilustrado pela figura 20. Ao autorizar uma solicitação desta natureza, o agente responsável pelo cumprimento desta tarefa proverá as operações necessárias para a devida transferência do curso.

- *Avisar aluno*

Dependendo da decisão tomada na autorização ou não da transferência, uma mensagem é enviada ao solicitante e ao responsável pelo registro de transferências da Secretaria Geral de Ensino, através do sistema de correio eletrônico. Esta etapa é realizada através do acionamento de um agente executor pelo agente coordenador, o qual após receber o comunicado de que a etapa foi cumprida encerra a solicitação do cliente.

5.4 Resultados Obtidos

Após a prototipação e implementação do modelo, foram realizados testes a fim de avaliar o comportamento do modelo na obtenção dos objetivos a que fora proposto.

A construção do protótipo foi conduzida a partir de consultas e entrevistas ao corpo funcional da Secretaria Geral de Ensino da UNISUL – Universidade do Sul de Santa Catarina. Este departamento é responsável pela prestação dos mais variados tipos de serviços, seja para a área acadêmica, administrativa ou até mesmo para entidades externas ao meio universitário.

Uma das principais funções desta Secretaria Geral de Ensino consiste no atendimento aos alunos e ex-alunos da Universidade, fornecendo por exemplo, documentação relativa a sua vida acadêmica, recebendo processos de transferências, prestando informações, dentre outros.

Atualmente no campus de Tubarão(SC), este departamento possui uma clientela de 8000 alunos, havendo em determinados períodos do ano letivo um volume de atendimento aos clientes acima da média, devido ao encerramento de prazos legais para a solicitação de alguns serviços.

Diante deste contexto, foram realizados testes visando contemplar alguns dos serviços atualmente prestados por esta Secretaria Geral de Ensino que estão implementados no protótipo, uma vez que esta Secretaria executa uma vasta gama de serviços.

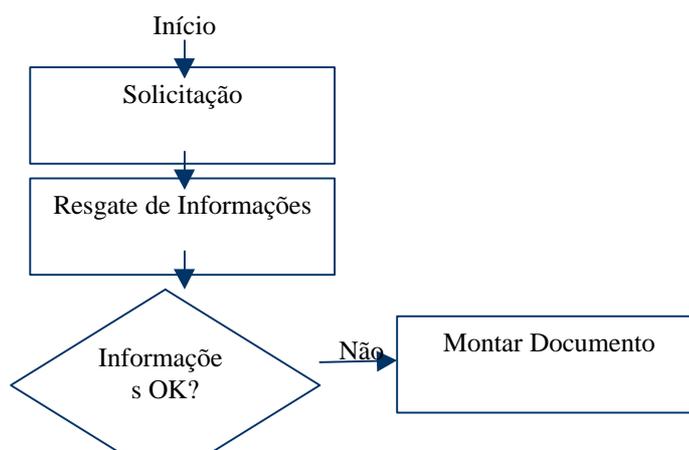
Para viabilizar a prestação dos serviços, o referido departamento envolve um corpo funcional composto por 10 colaboradores, distribuídos nas tarefas de:

- Atendimento e Protocolo de Pedidos: 03 Colaboradores
- Construção dos Documentos: 05 Colaboradores
- Conferência: 02 Colaboradores

Além deste corpo funcional, são envolvidos ainda colaboradores de outras áreas, como por exemplo a área acadêmica através dos coordenadores de curso e chefes de departamento, cuja responsabilidade é de atestar a veracidade e legitimidade do processo.

De maneira geral, o fluxo para a solicitação de documentos, entrada de processos e solicitação de informações ao usuário ocorre de acordo com os fluxos de informações ilustrados nas figuras 25, 26 e 27, respectivamente.

Figura 25 – Fluxograma para Solicitação de Documentos



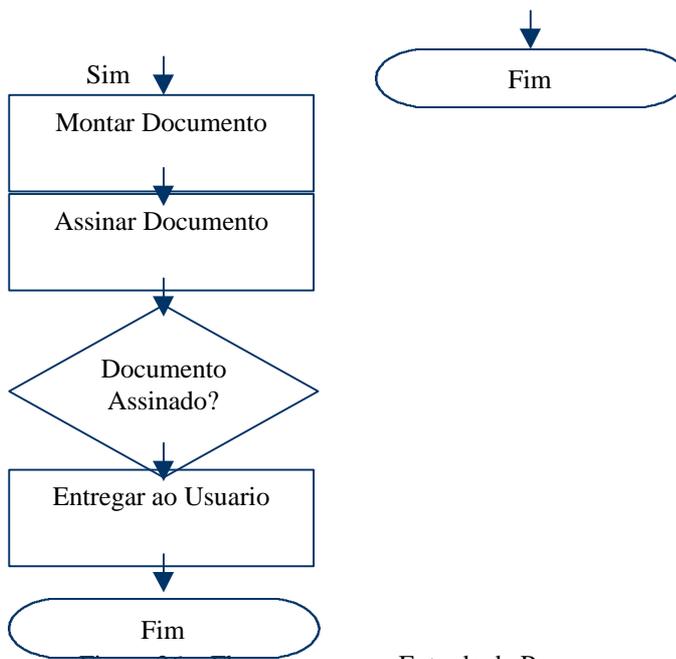


Figura 26 – Fluxograma para Entrada de Processos

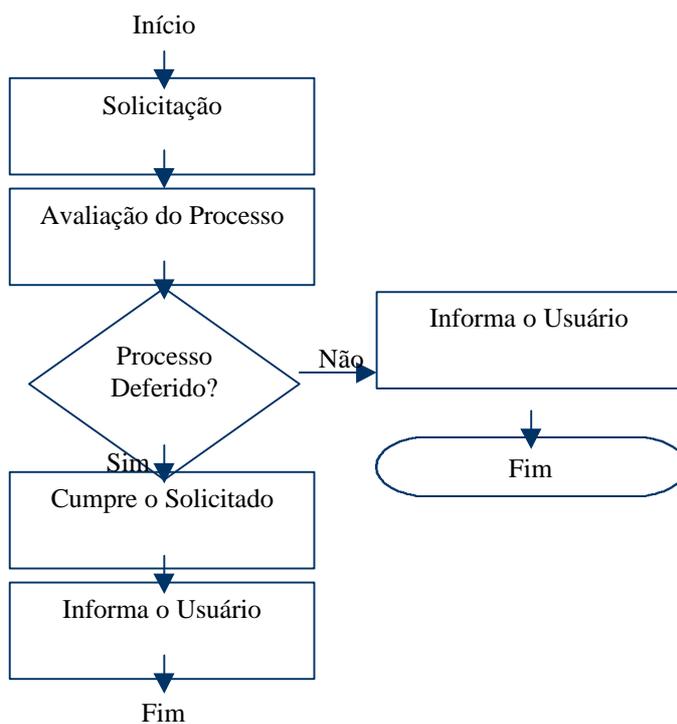
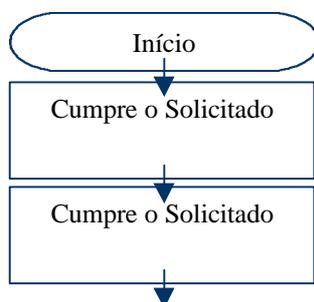
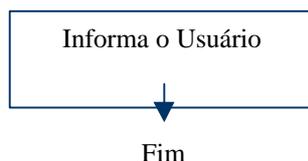


Figura 27 – Fluxograma para Solicitação de Informações Acadêmicas





O prazo para a conclusão destes tipos de solicitações varia de 24 a 96 horas. O usuário, ao solicitar um serviço é previamente informado do tempo necessário para o atendimento da sua solicitação, devendo, após decorrido este tempo, retornar ao local da solicitação para retirar o pedido. Em caso de insuficiência de informações ou atraso na liberação da solicitação, o usuário é comunicado a retornar após um novo prazo fixado.

Para a realização dos testes foi construída uma base de dados com as informações acadêmicas de 20 alunos, número este suficiente para contemplar as diversas situações que podem ocorrer no decorrer da prestação dos serviços.

Durante a fase de testes foram constatadas a possibilidade de melhorias de ordem administrativa com relação a aplicação do modelo para a área de solicitação de serviços para a Secretaria Geral de Ensino, principalmente no que se refere ao envolvimento do corpo funcional com a execução de tarefas meramente manuais e rotineiras, agora executadas automaticamente, liberando o corpo funcional para outras atividades mais importantes, sendo esta uma necessidade do departamento.

O protótipo proporcionou maior agilidade e eficiência na prestação dos serviços, principalmente pelas características da sociedade de agentes de pró-atividade, autonomia e colaboração entre os elementos aplicados sob o enfoque da teoria do gerenciamento indireto, reduzindo o tempo de prestação do serviço em até 30%, sendo que alguns serviços passaram a fornecer uma resposta imediata ao solicitante. Através da tabela 1 pode ser observada, em números absolutos e percentuais, a redução neste tempo da prestação do serviço.

Tabela 1 – Tempos para a prestação de Serviços

SERVIÇO	Serviço Prestado Manual (Tempo Máximo)	Serviço Prestado Automático (Tempo Máximo)	Redução
Emissão de Documentos	96 horas	74 horas	22 horas (22,9%)
Emissão de Informações	72 horas	43 horas	29 horas (40,1%)
Processos Acadêmicos	96 horas	70 horas	26 horas (27,0%)
Redução Total Média			(30,0%)

A ocorrência de falhas durante o processo, tanto na confecção dos documentos solicitados quanto na execução do serviço como um todo foi diminuída na ordem de aproximadamente 50%, devido a padronização das etapas automatizadas, contemplando todas as informações necessárias à execução das etapas.

A Tabela 2 contempla os percentuais de falhas obtidos para as últimas 50 ocorrências de cada tipo de serviço. É importante destacar aqui que as falhas não necessariamente ocorreram devido a problemas técnicos ou falha humana, mas também estão classificados como falhas a falta de registro de informações básicas para a execução do serviço ou a falta de documentos necessários à prestação do serviço.

Tabela 2 – Ocorrência de Falhas na Execução dos Serviços

SERVIÇO	Falhas Ocorridas no Serviço Prestado Manual	Falhas Ocorridas no Serviço Prestado Automático	Redução nas Falhas
Emissão de Documentos	8% dos Serviços	6% dos Serviços	25%
Emissão de Informações	6% dos Serviços	3% dos Serviços	50%
Processos Acadêmicos	10% dos Serviços	3% dos Serviços	70%
Redução Total Média			48,3%

A ocorrência de atrasos na prestação do serviço foi reduzida em aproximadamente 40%, uma vez que ocorre um planejamento na fixação da data prevista para a conclusão da solicitação bem como da execução do serviço, podendo ocorrer atrasos onde a interferência do usuário é necessária, como por exemplo na autorização da execução de alguma etapa ou na assinatura de algum documento.

A tabela 3 reflete os percentuais de atrasos ocorridos referentes a amostra já mencionada para obtenção de percentuais de ocorrência de falhas.

Tabela 3 – Ocorrência de Atrasos na Prestação dos Serviços

SERVIÇO	Atrasos Ocorridos no Serviço Prestado Manual	Atrasos Ocorridos no Serviço Prestado Automático	Redução nos Atrasos
Emissão de Documentos	12% dos Serviços	8% dos Serviços	33,3%
Emissão de Informações	14% dos Serviços	6% dos Serviços	42,8%
Processos Acadêmicos	16% dos Serviços	7% dos Serviços	43,8%
Redução Total Média			39,9%

Um fator bastante positivo proporcionado pela aplicação do protótipo foi a maior comodidade e segurança ao solicitante. Comodidade ao ter a possibilidade de solicitar o serviço de algum ponto remoto através da internet, e segurança ao poder, a qualquer momento, solicitar a situação atual em que se encontra a execução do pedido solicitado.

Após a realização dos testes, foi identificado em alguns serviços a necessidade de emissão de documentos fora de um padrão anteriormente especificado, como a necessidade de anotações e/ou observações a serem registradas, informações estas advindas de outras fontes não informatizadas. Devido a esta característica, estes serviços não puderam ser totalmente contemplados pelo protótipo.

A implementação deste modelo foi realizada em várias fases, a cada uma novos serviços foram sendo incluídos. Numa primeira fase foram implementados alguns serviços envolvendo as categorias de Emissão de Documentos e de Processos, como por exemplo os serviços de Atestados de Frequência, Atestados de Vaga, Processo de Trancamento de Disciplinas e Processo de Troca de Curso. Numa Segunda etapa foram implementados os demais serviços pertencentes a estas categorias, a fim de comprovar a eficiência e a viabilidade da reutilização de etapas pertencentes aos serviços já implementados. Na última etapa foram incluídos ao rol de serviços aqueles pertencentes a outras categorias, como por exemplo os da categoria de Emissão de Informações, a fim de comprovar a versatilidade das etapas em servir a serviços de outras categorias.

Com esta forma de implementação pôde ser comprovada a flexibilidade do modelo à incorporação de novos serviços aos já existentes, podendo ser reutilizadas as etapas já implantadas desde que sejam comuns aos serviços.

De modo geral, pôde ser comprovada a viabilidade da implementação do modelo e os testes realizados puderam demonstrar que a implantação deste modelo pode trazer vários benefícios no que diz respeito a automação de serviços.

6 CONCLUSÕES E RECOMENDAÇÕES

Este trabalho teve sua concepção na automação de tarefas e, por conseguinte, na automação de prestação de serviços como um todo.

Como objetivo final, este trabalho se propõe a apresentar um modelo, o qual venha a permitir a execução de serviços com a mínima intervenção humana, quando necessária. Para tanto, está baseado em técnicas de gerenciamento indireto e na utilização da tecnologia de agentes.

A utilização da tecnologia de agentes aliado a técnica de gerenciamento indireto nos permite explorar amplamente o desenvolvimento de sistemas cujo objetivo vá de encontro a automação de serviços. Os agentes, devido às características de autonomia, habilidade social, colaboração, dentre outras, constituem em importantes ferramentas para se tratar de sistemas autônomos, apresentando facilidades de implementação e atingindo resultados altamente satisfatórios.

A modularidade do modelo e a implantação em três camadas constituem-se em características importantes no modelo, permitindo que se alie flexibilidade e agilidade ao sistema.

A base de conhecimentos do modelo permite um nível alto de flexibilidade e escalabilidade, sendo possível a qualquer instante a construção de agentes responsáveis por novos domínios de atuação e o registro de suas funções na base de conhecimentos.

Um dos cuidados durante a concepção do modelo, foi o de não criá-lo especificamente para a solução de automação de serviços específicos em alguma área de conhecimento, restringindo sua aplicabilidade, mas a de se ter um modelo que seja amplamente utilizado devido as suas características.

A título de sugestão, é recomendável que a implementação do sistema se dê através de mecanismos de comunicação eficientes, sendo um elemento decisivo entre o sucesso ou não da implementação, uma vez que o sistema se utiliza de ampla comunicação entre os agentes.

No modelo aqui apresentado não foram esgotadas as possibilidades de sua aplicação. A seguir estão arroladas algumas melhorias identificadas que podem futuramente ser incorporadas ao modelo, como por exemplo:

- Adaptar ao modelo mecanismos de negociação entre os elementos da sociedade de agentes, permitindo o estabelecimento de prioridades na execução das etapas.

- Incorporar ao modelo elementos que permitam a visualização das regras mantidas pela base de conhecimentos, de maneira a permitir que o usuário possa entender de que maneira o serviço é prestado.
- Incluir escalas de prioridades na execução dos serviços
- Contemplar a sociedade de agentes com elementos que eliminem da arquitetura de quadro-negro informações lá depositadas e que não mais interessem à sociedade de agentes, permitindo maior agilidade na pesquisa por informações nesta memória de trabalho.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- [AGHA,1988] Agha, G., Hewitt, C. **Concurrent programming using actors. Object-Oriented Concurrent Programming.** MIT Press, 1988
- [APPLE,1996] **Human Interface Principles In: Macintosh Human Interface Guidelines.** Apple Computer Inc. 1996. URL: <http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html>
- [BJÖRN, HERMANS,1997] Bjorn, H. **Intelligent Software Agents on the Internet,** 1997 URL: http://www.firstmonday.dk/issues/issue2_3/index.html
- [BOND,1998] Bond, A., Gasser, L. **Readings in Distributed Artificial Intelligence.** Morgan Kaufmann Publishers, Los Altos, CA, 1988
- [BRADSHAW,1977] Bradshaw, Jeffrey M. **Software Agents.** The MIT Press, 1997
- [CASTELFRANCHI,1995] Castelfranchi, C. **Guarantees for autonomy in cognitive agent architecture.** Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890) pp56-70. Springer-Verlag: Heidelberg, Germany. 1995
- [CHARNIAK,1985] Charniak. **Introduction to Artificial Intelligence** (with Drew McDermott), Addison-Wesley, 1985.
- [CORTES,1993] Cortes, U., Bejar, J., Moreno, A. **Inteligencia Artificial.** Ed. UPC, 1993
- [DENNEHY,1993] Dennehy Mike. **Direct Manipulation.** Seminário The Encyclopedia of Virtual Environments, 1993. URL: <http://www.hitl.washington.edu/scivw/EVE/index.html>
- [DURFEE & ROSENSCHEIN,1994] Durfee E. H., Rosenschein J. S. **Distributed Problem Solving and Multi-Agent Systems: Comparison and Examples.** In: International Workshop on Distributed Artificial Intelligence, 1994.
- [ENRIC,1998] Enric Mor i Pera. **Diseño de un Agente que habita en Internet.** URL: <http://www-lsi.upc.es/~emor/agencia/> Setembro 1998. UNIVERSITAT POLITÈCNICA DE CATALUNYA – Espanha
- [FERBER in DEMAZEAU et al.,1992]DEMAZEAU, Yves, SICHMAN, Jaime Simão, BOISSIER, Oliver. **"When can Knowledge-based Systems be Called Agents?"**. XII Congresso da Sociedade Brasileira de Computação - IX Simpósio Brasileiro de Inteligência Artificial. pp 172-185, Rio de Janeiro, 1992.
- [FININ et alii 1993] Finin, Tim; Labrou, Yannis, Mayfield, James. **KQML as an agent communication language.** Computer Science Department. University of Maryland Baltimore County. Baltimore USA, 1993
- [FININ et alii, 1992a] Finin, Tim; Labrou, Yannis. **A Semantics approach for KQML - a general purpose communication language for software agents.** University of Maryland Baltimore County. Baltimore USA 1992a.
- [FININ et alii, 1997] Finin, Tim; Labrou, Yannis. **A Proposal for a new KQML Specification.** University of Maryland Baltimore County. (UMBC)

- [FONER,1993] Foner Lenny. **What's An Agent, Anyway?** A Sociological Case Study. Agents Memo 93-01, MIT Media Lab, Cambridge, MA. 1993
- [FRANKLIN & GRAESSER,1996] Franklin Stan, Graesser, **Is it an Agent or Just a Program?: A Taxonomy for Autonomous Agents.** University of Memphis, 1996. URL: <http://www.msci.memphis.edu/~franklin/AgentProg.html>
- [GAGLAYAN E HARRISON,1997] Caglayan, Alper e Harrison, Colin **Agent Sourcebook : A Complete Guide to Desktop, Internet, and Intranet Agents.** John Wiley & Sons, 1997.
- [GENESERETH & KETCHPEL,1994] Genesereth M. and Ketchpel, S. **Software Agents.** Communications of the ACM Julho 1994, Vol. 37 n. 7
- [GENESERETH,1992] Genesereth, M. e Fikes, R. **Knowledge Interchange Format, Version 3.0 – Reference Manual.** Logic Group, Computer Science Department, Stanford University. 1992
- [GILBERT,1996] Gilbert, D. & et alii. **The role of intelligent agents in the information infrastructure**
- [GROSZ,1990] Grosz, B., Sidner, C. **Plans for discourse. Intentions for Communication,** pp 417-444, MIT Press, 1990
- [GRUBER,1993] Gruber, T.R. **A traslation approach to portable ontologies.** Knowledge Acquisiton, 5(2):199-220, 1993
- [GRUBER,1997] Gruber, Tom. **What is an Ontology.** 1997. Web: www-si.stanford.edu/what-is-an-ontology.html
- [HAYES,1985] Hayes-Roth, B. **A blackboard architecture for control.** Artificial Intelligence, 26(3):251-321, 1985
- [HEETER,1991] Heeter Carrie. **The Look and Feel of Direct Manipulation.** Journal of Hypermedia and Multimedia Studies, 1991. URL:<http://commtechlab.msu.edu/randd/research/LookFeel/>
- [HEWITT,1977] Hewitt, C. **Viewing Control Structures as Patterns of Parsing Messages.** Artificial Intelligence 8 (3), 323-364, 1977
- [HOARE,1978] Hoare, C.A.R. **Communicating sequential processes.** Communications of the ACM, 21:666-677, 1978
- [HUTCHINS,1986] Hutchins, E.L., Hollan, J.D., and Norman, D.A. **"Direct manipulation interfaces"** in Norman, D.A. and Draper, S.W.(Eds.) User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.
- [IBM,1995] Gilber, A., et alii. **The Role of Intelligent Agents in the Information Infrastructure.** IBM, United States 1995
- [KAY,1984] Kay, A. **Computer Software.** In: Scientific American. Vol. 251, No. 3, 1984.
- [KNAPIK e JOHNSON,1998] Knapik, M. e Johnson, J. **Developing Intelligent Agents for Distributed Systems.** McGraw-Hill, 1998
- [LEAL e NUNES,1999] LEAL Tatiane dos Santos e NUNES, Darlan F. **Estudo da Arquitetura Network Computing com a Implementação e Implantação de um Protótipo.** Monografia - Universidade do Sul de Santa Catarina - UNISUL, 1999.
- [LEMON et al.,1996] Lemon, B., Pynadath, D. Taylor, G. e Wray, B. **Cognitive Architectures,** URL: <http://krusty.eecs.umich.edu/cogarch4/toc.html>, 1996.
- [LESSER,1987] Lesser, V., Durfee, E., Corkill, D. **Coherent cooperation among communicating problem solvers.** IEEE Transactions on Computers, 36(11):1275-1291, 1987

- [LESSER,1990] Lesser, V. An Overview of DAI: **Viewing Distributed AI as Distributed Search**. Journal of Japanese Society for Artificial Intelligence, Vol.5, No.4, 1990
- [LESSER,1994] Lesser, V. **Cooperative Distributed Problem Solving**. URL: <http://dis.cs.umass.edu/research/cpds.html>
- [MAES,1991] Maes, P. (1991b), "**Situated Agents Can Have Goals**", In Maes, P. (ed) (1991a), *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, London: The MIT press, 49-70
- [MAES,1995] Maes, P., "**Artificial Life Meets Entertainment: Life like Autonomous Agents**," *Communications of the ACM*, 38, 11, 108-114, 1995
- [MAES,1997/2] Maes, P. "**Software Agents: Humanizing the Global Computer**". Internet Computing, 7-8/97.
- [MAES,1997] Maes, P. "**Agents that Reduce Work and Information Overload**." In: Software Agents, edited by Jeffrey M. Bradshaw. AAAI Press/MIT Press, 1997.
- [MCMILLAN & PATTERSON,1998] Mcmillan Bill, Patterson Gary. HCI 1999/2000 **Lectures Notes**. URL: <http://starform.infj.ulst.ac.uk/BillsWeb/HCI/Lectures/lect8.html>. UNIVERSITY OF ULSTER - Belfast - Irlanda do Norte, 1998
- [MICHELS,1995] Michels Sjoerd. Tese: **The Cooperator**. Tilburg University - Netherland, 1995
- [NEGROPONTE,1970] Negroponte N. **The Architecture Machine**; Towards a More Human Environment. MIT Press, 1970.
- [NISSEN,1995] Nissen, M. **Intelligent Agents: A Technology and Business Application Analysis**. URL: <http://haas.berkeley.edu/~heilmann/agents/#exnum>, 1995.
- [NWANA,1996] Nwana, H.S. Software Agents: An Overview. Knowledge Engineering Review, Vol 11, No 3, pp.1-40, Sept 1996
- [OLIVEIRA,1996] Oliveira, Flávio M. Inteligência Artificial Distribuída. Instituto de Informática - PUCRS, Porto Alegre 1996.
- [PARAÍSO, 1997] Paraíso Emerson C. Concepção e Implementação de um Sistema Multi-Agentes para Monitoração e Controle de Processos Industriais. CPGEEII - CEFET/PR, 1997.
- [PIACINI] VANDERLEY J. PIACINI. Revista T & D - Treinamento & Desenvolvimento Ano V - Edição 51
- [ROSENSCHEIN,1986] Rosenschein, J.S. Rational Interaction: Cooperating Among Intelligent Agents. PhD thesis, Computer Science Department, Stanford University, 1986
- [RUSSELL e NORVIG,1995] RUSSEL, Stuart J e NORVIG, Peter. Artificial Intelligence - A Modern Approach. Prentice Hall. Upper Saddle River, NJ., 1995.
- [SHELLING,1960] Schelling, T.C. The Strategy of Conflict. Harvard University Press, Cambridge, MA, 1960
- [SHNEIDERMAN,1993] Shneiderman, Ben. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Publishing Company: Reading, MA, 1987.
- [SMITH, CYPLER e SPOHRER,1994 in FRANKLIN e GRAESSER,1996] Smith, D. C., A. Cypher and J. Spohrer (1994), "KidSim: Programming Agents Without a Programming Language" *Communications of the ACM*, 37, 7, 55-67
- [SPECTOR,1997] Spector, L. Automatic Generation of Intelligent Agent Programs. IEEE Expert Intelligent Systems, February 1997, pp. 3-4

[SYCARA,1989] Sycara, K.R. Multiagent compromise via negotiation. Distributed Artificial Intelligence, vol.2, pp.119-137, Morgan Kaufmann Publishers, Los Altos, CA, 1989

[WOOLDRIDGE & JENNINGS,1994] Wooldridge Michael, Jennings Nicholar R. Intelligent Agents: Theory and Praticce. Knowledge Engineering Review, 1994. URL:
<http://www.doc.mmu.ac.uk/STAFF/mike/ker95.ps>

[WUTKA,1997] Wutka Mark. **Java Técnicas Profissionais. Ed. Berkeley**, São Paulo, 1997.

JATLITE.*Java Agent Template Lite*,1997. URL:http://java.stanford.edu/java_agent/html/index2.html

JDBC™ Guide: Getting Started. Sun Microsystems, 1997.

8 ANEXOS

8.1 Resumo das mensagens KQML

As *performatives* ou mensagens KQML são apresentadas abaixo, onde a letra S (*sender*) identifica o emissor, R(*receiver*) o receptor e VKB(*virtual knowledge base*) é a base de conhecimento virtual.[Finin *et al.*, 1997]

Nome	Significado da Mensagem ou Performative
ask-if	S quer saber se o :content está na BKV de R
ask-all	S quer todas as respostas de R quando o :content for verdade para R
ask-one	S quer uma resposta de R quando o :content for verdade para R
stream-all	versão de respostas múltiplas de ask-all
Eos	indica o fim de um fluxo de respostas
Tell	informa que a sentença está na BKV de S
Untell	informa que a sentença não está na BKV de S
deny	a negação da sentença está na BKV de S
Insert	S pede a R adicionar o :content em sua BKV
Uninsert	S quer reverter uma ação de inserção
delete-one	S quer que R remova uma sentença de sua BKV
delete-all	S quer que R remova todas as sentenças indicadas na sua BKV
Undelete	S quer reverter uma ação de deleção
Achieve	S quer que R faça alguma coisa verdadeira no ambiente
Unachieve	S quer reverter a ação de um achieve
Advertise	S quer que R saiba que ele pode processar uma mensagem
Unadvertise	S quer que R saiba que ele não pode mais processar uma mensagem
Subscribe	S quer mudar a resposta de R à uma mensagem
Error	S considera a mensagem de R mal formada
Sorry	S entende a mensagem de R mas não pode dar uma resposta adequada
Standby	S quer que R esteja pronto para responder uma mensagem
Ready	S está pronto para responder uma mensagem recebida

	anteriormente de R
Next	S quer as próximas respostas de R
Rest	S quer as respostas restantes de R
Discard	S não quer as respostas restantes de R
Register	S anuncia a R sua presença no ambiente
Unregister	S quer reverter a ação de registro
Forward	S quer que R redirecione uma mensagem para o agente especificado em :to
Broadcast	S quer que R envie uma mensagens para todos os agentes
Transport-address	S quer associar seu nome com um novo endereço
broker-one	S quer que R encontre uma resposta para uma mensagem
broker-all	S quer que R encontre todas as respostas para uma mensagem
Recommend-one	S quer saber qual agente pode responder a uma mensagem
Recommend-all	S quer saber todos os agentes que podem responder a uma mensagem
recruit-one	S quer que R tome quais agentes podem responder a uma mensagem
recruit-all	S quer que R tome todos os agentes que podem responder a uma mensagem

8.2 Ferramentas Utilizadas Na Implementação do Protótipo

Para que seja suportada toda a funcionalidade do modelo foram utilizadas ferramentas específicas para a construção dos agentes e para a comunicação de toda a sociedade dos agentes pertencentes ao sistema. A seguir são descritas as características principais destas ferramentas.

Linguagem de Desenvolvimento JAVA

A Linguagem de Programação Java foi desenvolvida por engenheiros da empresa Sun Microsystems, baseada em objetos e sistemas abertos onde os desenvolvedores podem criar aplicações inclusive com características de distribuição de objetos. Um componente importante desta linguagem é a Máquina Virtual Java, que permite que todas as aplicações baseadas nesta linguagem sejam executadas em qualquer ambiente, independente de hardware ou sistema operacional.

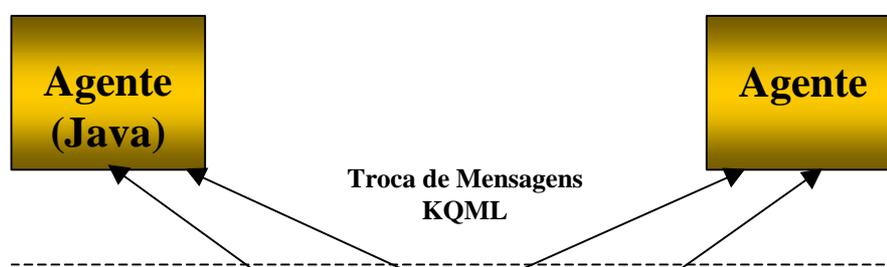
Dentre as características desta linguagem de desenvolvimento pode ser destacado:

- *Simplicidade*: Java oferece toda a funcionalidade de uma linguagem de programação potente, sendo que as características menos utilizadas nas linguagens tradicionais não estão presentes, facilitando o aprendizado. Possui desalocador de memória automático, liberando o desenvolvedor deste controle.
- *Orientação a Objetos*: Java possui incorporado a tecnologia básica de orientação a objetos presentes em outras linguagens tais como C++, suportando as três características básicas envolvendo o paradigma da orientação a objetos: a encapsulação, herança e polimorfismo.
- *Distribuição*: Esta linguagem foi construída com inúmeras capacidades de interconexão, constituindo-se de bibliotecas para interagir com protocolos tais como HTTP e FTP, permitindo facilidades no envio de informações através das redes de comunicação.
- *Robustez*: Java detecta a existência de problemas tanto em tempo de compilação quanto em tempo de execução, obriga a declaração explícita de métodos, reduzindo as possibilidades de erro, manipula a memória de modo que o programador se omita do gerenciamento de espaço disponível ou de corrupção.
- *Arquitetura Neutra*: O código-fonte de uma aplicação Java pode ser executada em qualquer máquina, independente do hardware ou sistema operacional, devido ao componente da Máquina Virtual.

Java Agent Template – JATLite

JatLite é um conjunto de programas escritos em Java, o qual permite a criação de softwares capazes de estabelecer uma comunicação robusta sobre uma rede de computadores, disponibilizando uma infra estrutura básica para registro de agentes pertencentes a sociedade através do Roteador de Mensagens de Agentes (AMR – Agent Message Router), permitindo o envio e recebimento de mensagens, transferência de arquivos, enfim, permite troca de informações em geral entre agentes existentes na sociedade.

Figura 28 – Agent Message Router



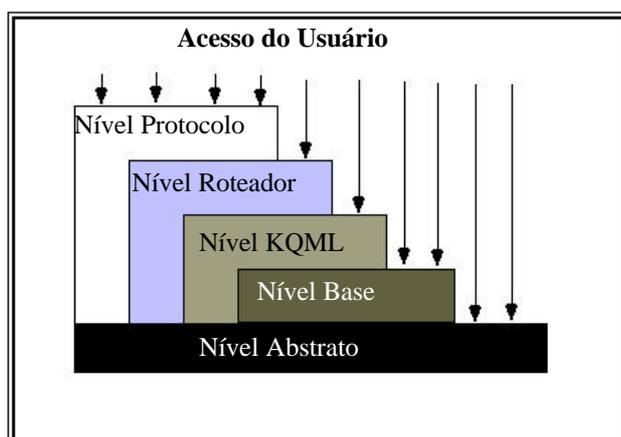
Tradicionalmente os sistemas de agentes realizam suas conexões entre os agentes através do ANS (Agent NameServer). Os agentes se utilizam do ANS para ter acesso ao endereço IP do outro agente, utilizando-o no estabelecimento de uma conexão TCP com o agente de destino. Qualquer problema que ocorra com o agente destino que impossibilite a troca de mensagens é de responsabilidade do emissor, devendo este tratar o problema, caso necessite reenvio ou armazenamento da mensagem que falhou. Desta forma o roteador é quem trata dos erros na entrega das mensagens, não mais o agente emissor.

Com o Agent Message Router, cada agente estabelece uma comunicação com o AMR, o qual envia as mensagens através do nome do registro dos agentes, não do endereço IP. O AMR armazena todas as mensagens até receber a confirmação do recebimento pelo agente destino.

As comunicações estabelecidas são construídas através dos padrões abertos da Internet, tais como TCP/IP, SMTP, FTP.

A arquitetura do JATLite é organizada através de uma hierarquia de bibliotecas com níveis de especialização crescentes, conforme indica a figura 29.

Figura 29 – Níveis de Especialização do JATLite



O *nível abstrato* provê uma coleção de classes necessárias a implementação da ferramenta JATLite, assumindo todas as conexões realizadas sobre o protocolo TCP/IP.

O *nível de base* provê o armazenamento e “parsing” das mensagens KQML, linguagem padrão sugerida para tratar as conversações entre os agentes.

O *nível de roteamento* provê registro dos agentes bem como o roteamento e o alinhamento das mensagens a serem enviadas.

O *nível de protocolo* provê o suporte aos diversos padrões de serviços Internet, tais como o SMTP, FTP, POP3 e HTTP.

Linguagem de Comunicação KQML

A KQML (Knowledge Query and Message Language) é uma linguagem e também um protocolo de comunicação voltado para a troca de informações e de conhecimento, fazendo parte do DARPA Knowledge Sharing Effort, que procura desenvolver técnicas e metodologias reutilizadas e compartilhadas [FININ et al 1993].

Desta forma a KQML pode ser utilizada como uma linguagem, visando uma interação com outros sistemas inteligentes proporcionando uma possível troca de conhecimento para a solução de determinado problema.

Esta linguagem possui um conjunto de mensagens padrão, denominadas performatives. Cada performative possui um objetivo e realiza uma ação específica. Desta forma, assim que o agente recebe uma mensagem KQML, toma uma ação baseado no significado e conteúdo desta mensagem, podendo ainda responder as mensagens sem ter conhecimento da estrutura do agente emissor.

Um agente deve assumir um comportamento próprio e autônomo, podendo interagir com outros agentes. O estado mental de um agente vem a representar seu comportamento. Tal estado pode ser constituído por uma base de conhecimentos, de forma a manter informações sobre suas crenças e seus objetivos. Segundo Tim Finin, as crenças de um agente são formados por informações de seu ambiente bem como informações sobre o próprio agente.

A KQML normalmente carrega como informação estruturas de outras linguagens ditas linguagens de conteúdo, tais como a KIF, visto na seção 3.1. Estas estruturas são encapsuladas numa expressão KQML, tendo formatos e sintaxes independentes.

Finin, ressalta que a KQML é uma linguagem de uso geral, podendo ser utilizada na comunicação de sistemas que fazem o uso de agentes independentemente da estrutura ou do ambiente em que estes agentes estejam inseridos.

A especificação da KQML segundo Tim Finin, não procura padronizar a infra-estrutura para o transporte de mensagens, visto que isto vai depender diretamente da linguagem utilizada na implementação do sistema, mas define algumas abstrações a respeito do transporte das mensagens:

- A conexão entre os agentes é realizada através de links de comunicação unidirecionais;
- Ao receber uma mensagem o agente deve saber por qual link esta foi transmitida;
- Ao enviar uma mensagem o agente deve escolher um link de saída existente;
- As mensagens enviadas a um mesmo destino devem chegar na mesma ordem de envio;
- Deve-se ter confiabilidade e segurança na entrega das mensagens.

Estes links de comunicação podem ser estabelecidos de diversas formas, cabendo citar:

- Uma conexão TCP/IP, que estaria ativa apenas durante a transmissão da mensagem;
- Conexões IPC entre processos numa máquina;
- Conexões entre endereços de correio eletrônico;

Estrutura da KQML

Para a KQML a comunicação a nível de agentes é um método de comunicação ponto-a-ponto, ou seja, abrange a esfera do agente emissor até o próximo agente receptor.

De acordo com a especificação da KQML Tim Finin, estão definidos um grande conjunto de performatives, sendo que os sistemas não necessitam compreender a todos, apenas os que julgar necessário para a realização de suas tarefas. Da mesma forma, novas performatives podem ser criadas, desde que obedeçam a forma de especificação da linguagem, criando um conjunto de mensagens estendidas ao conjunto original.

Ainda segundo Finin [FININ et al, 1992a], a linguagem KQML é formada por uma estrutura subdividida em três camadas. Estas camadas se relacionam de forma direta com uma mensagem KQML, no que se refere as informações que contém. São elas:

CAMADA DE CONTEÚDO: A camada de conteúdo trata da mensagem a ser transmitida, expressa numa linguagem que encapsule a informação. Desta forma o conteúdo da mensagem é uma informação totalmente independente da linguagem KQML.

Este conteúdo deve ser de conhecimento do agente que os envia e recebe, pois a ele cabe interpretá-lo e tomar as ações correspondentes.

A título de exemplo da atuação desta camada numa conversação, temos destacada uma mensagem KQML, tendo em seu conteúdo uma mensagem escrita em linguagem natural:

```
(ask-if :sender EMISSORX :receiver RECEPTORZ :in-reply-to MENSAGEMA
:reply-with MENSAGEMB      :language NATURAL :ontology 'ENVIO DE CON
TEÚDO' :content ('ESTE É O CONTEÚDO DE UMA MENSAGEM KQML'))
```

CAMADA DE MENSAGEM: Esta camada se refere aos diversos tipos de mensagens que podem ser comunicadas entre os agentes, definindo que tipo de performative deva ser usada ao transmitir um conteúdo específico [FININ et al 1993].

Para Tim Finin esta camada é também conhecida como uma camada de ação de discurso, representando os tipos de atos que podem ser tomados, tais como perguntas, respostas, erros, afirmações e negações.

Como visto anteriormente, o conteúdo da mensagem não é de domínio da linguagem KQML, fazendo-se necessário a presença de alguns atributos os quais têm a finalidade de fornecer informações adicionais a respeito do conteúdo da mensagem.

Os parâmetros são palavras precedidas por ':' (dois pontos), tendo um valor associado. A seguir estão relacionados alguns dos principais parâmetros com uma discussão sucinta:

- :content – Este parâmetro contém o conteúdo da mensagem propriamente dito. Este conteúdo deve estar de acordo com o parâmetro :language.
- :from - Este parâmetro identifica a origem da performative
- :in-reply-to –Este parâmetro define um identificador único para a mensagem, que é o que identifica a mensagem para o agente receptor e emissor.
- :language – Este parâmetro especifica qual a linguagem na qual foi descrito o conteúdo da mensagem.
- : ontology – Este parâmetro especifica o domínio de conhecimento no qual o conteúdo da mensagem deve ser interpretado.
- :receiver – Este parâmetro especifica o identificador do agente receptor de uma performative.
- :reply-with –Este parâmetro define um identificador único para a mensagem, que é o que identifica a mensagem para o agente receptor e emissor.
- :sender – Este parâmetro especifica o identificador do agente emissor de uma performative.

CAMADA DE COMUNICAÇÃO: A camada de comunicação é representada por parâmetros responsáveis por munir de informações a mecânica da comunicação da KQML. Estas informações consistem

de parâmetros de baixo nível de comunicação, tais como emissor, receptor, identificadores únicos das mensagens.

Sintaxe da Linguagem KQML

Uma mensagem, ou performative é representada como uma cadeia de caracteres ASCII, tendo uma sintaxe bem definida para formalizar a troca de mensagens entre os agentes.

A sintaxe da KQML possui algumas características vantajosas, sendo fácil de ser transportada a outras aplicações, legível para os humanos e simples a programas que executam o "parse" [FININ et al, 1993].

A seguir está descrito a definição sintática para a linguagem KQML sob o formato BNF (Bakus-Naur Form), especificando facilmente as mensagens utilizando-se de poucas regras e símbolos.

Tabela 4 - Definição sintática da KQML [Finin et al., 1993]

```

<performative> ::= (<word> { <whitespace> :<word> <whitespace> <expression>}*)
<expression> ::= <word> | <quotation> | <string> | (<word> {<whitespace> <expresion>}*)
<word> ::= <character><character>*
<character> ::= <alphanumeric> | <numeric> | <special>
<special> ::= < > | = | + | - | * | / | & | ^ | ~ | _ | @ | $ | % | : | . | ! | ?
<quotation> ::= ' <expression> | ' <comma-expression>
<comma-expression> ::= <word> | <quotation> | <string> | , <comma-expression>
                    (<word> { <whitespace> <comma-expression>}*)

```

```
<string> ::= "<stringchar>*" | #<digit><digit>*"<ascii>*"
<stringchar> ::= \<ascii> | <ascii>-\<double-quote>
```

Semântica das Mensagens

O modelo semântico das mensagens formam um contexto simples e uniforme, permitindo que os agentes consigam perceber as capacidades dos demais, tendo a impressão de que os agentes manipulam uma base de conhecimento. Assim sendo, uma comunicação entre agentes pode ser entendida como operações que venham a manipular tal base de conhecimentos, tais como solicitar o uso das informações, solicitar inclusões ou exclusões de sentenças ou questionar sobre que informações são de domínio de tal base.

É importante mencionar que não necessariamente o agente esteja estruturado como uma base de conhecimento, mas se utilize de outras ferramentas para armazenar suas informações, tais como bases de dados ou estruturas próprias de armazenamento de dados.

Desta forma, o agente necessita gerenciar uma base de conhecimento virtual, traduzindo a representação original para uma base de conhecimento abstrata, beneficiando assim os outros agentes que venham a se utilizar destas informações.

O conteúdo de uma base de conhecimento virtual pode ser classificado em duas categorias:

Crenças – Codificam informações a respeito do próprio agente e do ambiente externo em que está inserido, inclusive sobre bases virtuais de outros agentes.

Objetivos – Codificam informações a respeito dos estados do ambiente externo importantes para os objetivos do agente na sua atuação.

Performatives KQML

Como visto anteriormente, uma mensagem é uma performative que, ao ser recebida propõe uma ação.

A linguagem KQML foi especificada pela primeira vez em 1993, sendo proposta uma revisão em 1997 com modificações no conjunto de mensagens, significado e intenção de uso. [FININ et al, 1997]

As performatives podem ser classificadas em três categorias:

- Performatives de discurso: Estas performatives são utilizadas para proporcionar a troca de informações entre os agentes.

- Performatives de intervenção mecânica e de conversação: Estas performatives tem sua função baseada na intervenção do fluxo normal do diálogo assim que ocorra uma anormalidade. Tal intervenção ocorre interrompendo-se o fluxo de mensagens ou como a introdução de protocolos na interação, suprimindo a anormalidade da comunicação.

- Performatives de rede: As performatives de rede atuam no sentido de facilitar o diálogo entre os agentes, onde a presença de um agente facilitador pode organizar e gerenciar um grupo de agentes. Tais performatives podem auxiliar por exemplo na localização de agentes que estão aptos a responder por determinada solicitação.

As performatives se utilizam de parâmetros, permitindo uma diferenciação no comportamento das mensagens segundo o conteúdo mantido por tais parâmetros. A título de exemplificação do uso das performatives, serão descritas algumas delas:

Performative tell: A performative de discurso tell informa para o receptor que o conteúdo de :content faz parte da base de conhecimento virtual do emissor, validando tal conteúdo.

```
(tell
: sender <word>
: receiver <word>
: in-reply-to <word>
: reply-with <word>
```

: language <word>
 : ontology <word>
 : content <expression>)

Performative register: Esta é uma performative pertencente à classe das performatives de rede, sendo utilizada pelo emissor para anunciar ao facilitador a sua presença na sociedade bem como informar seu nome e endereço físico. No parâmetro :content estão as informações a respeito do agente emissor que o facilitador deve conhecer.

(register
 : sender <word>
 : receiver <word>
 : in-reply-to <word>
 : reply-with <word>
 : language <word>
 : ontology <word>
 : content <expression>)

Performative error: O emissor utiliza esta performative é usada pelo emissor para informar se algum erro ocorre na mensagem identificada pelo parâmetro :in-reply-to. Normalmente, existem três tipos de erros: erro sintático, erro nos parâmetros ou erro com a política ou protocolo de conversação.

(error
 : sender <word>
 : receiver <word>
 : in-reply-to <word>
 : reply-with <word>)

Uma especificação resumida da linguagem KQML está disponível no anexo I deste trabalho.

O JDBC é uma interface de comunicação cuja função é proporcionar uma comunicação entre fontes de dados, quando estas fontes se utilizam de padrões não compatíveis entre si para estabelecer uma comunicação. Segundo afirma Mark Wutka [Wutka,1997], os dados, inclusive de áudio e vídeo, podem ser recuperados e armazenados em objetos construídos sob a linguagem JAVA utilizando-se para tanto de interfaces JDBC.

8.3 Regras de Produção

A seguir estão descritas algumas regras de produção pertencentes a base de conhecimentos que foram implementadas no protótipo a fim de nortear a execução dos serviços.

Regra Nr.	Descrição da Regra
1	Se DOCUMENTO_ERRO = OK Então FIM_EXECUÇÃO = OK;
2	Se EXECUTE_RESGATE_DADOS = OK Então RESGATE_DADOS = OK;
3	Se DOCUMENTO = OK Então FIM_EXECUÇÃO_ERRO = OK;
4	Se FIM_EXECUÇÃO_ERRO = OK Então Stop;
5	Se FIM_EXECUÇÃO = OK Então Stop;
6	Se ENVIO_DOCUMENTO = OK E NOTIFICAÇÃO_USUARIO = OK Então DOCUMENTO = OK;
7	Se AUTORIZAÇÃO = OK Então EXECUTE_ENVIO_DOCUMENTO; EXECUTE_NOTIFICACAO_USUARIO;
8	Se EXECUTE_ENVIO_DOCUMENTO = OK Então ENVIO_DOCUMENTO = OK; Senão DOCUMENTO_ERRO = OK;
9	Se EXECUTE_NOTIFICACAO_USUARIO = OK Então NOTIVICAÇÃO_USUARIO = OK Senão DOCUMENTO_ERRO = OK;
10	Se GERAÇÃO = OK

	Então EXECUTE_AUTORIZAÇÃO;
11	Se EXECUTE_AUTORIZAÇÃO = OK Então AUTORIZAÇÃO = OK Senão DOCUMENTO_ERRO = OK;
12	Se RESGATE_DADOS = OK Então EXECUTE_GERAÇÃO;
13	Se EXECUTE_GERAÇÃO Então GERAÇÃO = OK Senão DOCUMENTO_ERRO = OK;

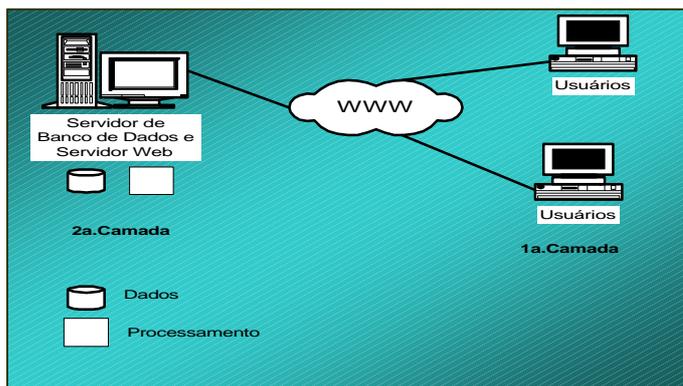
8.4 Arquitetura de Processamento Baseada em Três Camadas

Até a década de 90, o padrão implantado nos sistemas computacionais era uma arquitetura de processamento baseada em duas camadas, também denominada Arquitetura Cliente/Servidor Two-Tier, já uma evolução dos sistemas centralizados, advindos com os Mainframes.

Essa arquitetura, segundo Tatane Leal [Leal e Nunes,1999], é composta na primeira camada pelo cliente, sendo considerado o usuário e o programa propriamente dito, detendo o código da aplicação para si, localmente em sua máquina. A segunda camada contém o servidor de banco de dados, onde estão depositados os dados que servem às aplicações cliente.

A figura 30 ilustra graficamente a arquitetura baseada no processamento em duas camadas.

Figura 30 – Arquitetura Cliente/Servidor em duas camadas – Two Tier



Essa arquitetura apresenta algumas vantagens em relação às arquiteturas centralizadas, por exemplo, melhoria do poder de processamento para o cliente independente do servidor, redução do volume de tráfego através da rede de comunicação, dentre outras. Entretanto, essa arquitetura apresenta pontos fracos, como por exemplo, ao promover uma atualização da aplicação, é exigido que seja feita a atualização em cada máquina cliente, onde reside a aplicação; redundância de código, uma vez que nessa arquitetura as regras do negócio podem estar contidas em ambas as camadas.

Tatiane Leal [Leal e Nunes, 1999] apresentam ainda uma solução para estes problemas encontrados na arquitetura Two-Tier: uma arquitetura baseada em três camadas, também denominada arquitetura Three-Tier.

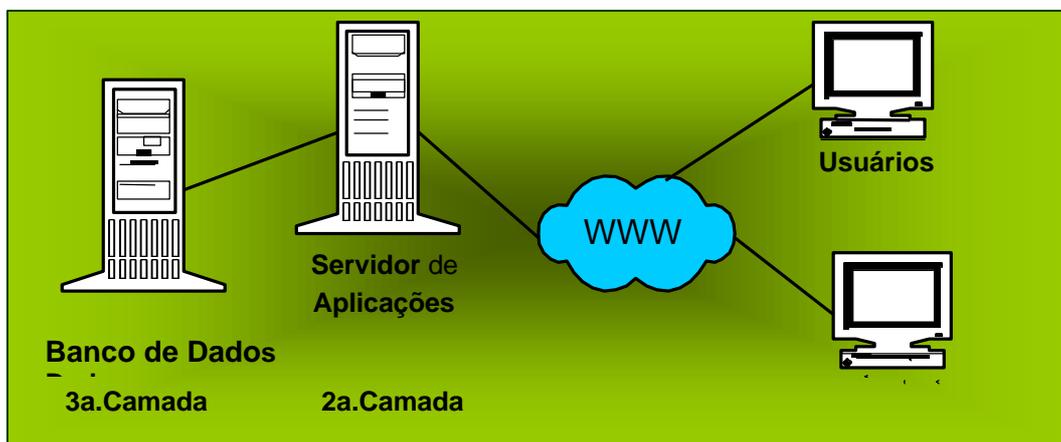
Nesta implementação é criada uma camada intermediária entre o cliente e o servidor de dados, denominado servidor de aplicações, ao qual cabe o papel de receber as chamadas dos clientes às aplicações, requisitar ao servidor de dados as informações necessárias, processá-las de acordo com o exigido pelas aplicações e enviá-las ao cliente.

Um dos pontos fortes da arquitetura Three-Tier é o isolamento da aplicação do cliente e a centralização de quase toda a regra do negócio na aplicação.

Ao cliente cabe a apresentação da interface ao usuário. Qualquer alteração registrada na aplicação, tal como alteração em alguma regra do negócio ou na interface com o usuário é repassada de forma direta aos usuários do sistema, uma vez que o código da aplicação se encontra centralizado no servidor de aplicações, de onde é requisitada pelo cliente.

Consegue-se desta forma obter um ganho de performance e escalabilidade significativos no sistema como um todo, uma vez que ambas as três camadas dividem responsabilidades específicas e bem definidas.

Figura 31 – Arquitetura cliente/servidor – Three Tier



A arquitetura Three-Tier se apresenta como uma alternativa interessante para a implantação do modelo aqui proposto, devido às características de autonomia e comunicabilidade dos agentes e da modularidade do modelo, permitindo ampla comunicação entre os elementos, situados inclusive em vários equipamentos, desde que remotamente conectados através de algum canal de comunicação.

Neste sentido, no modelo proposto, a primeira camada representa a camada cliente, o agente de interface do usuário e do responsável, atuando num navegador, normalmente referenciado como 'browser'. Esse agente se comunica com a base de conhecimentos, através do servidor de aplicações, requisitando solicitações e enviando informações e resultados das solicitações aos clientes.

Na camada intermediária, onde encontra-se o servidor de aplicações, estão armazenados os agentes responsáveis pelo processamento das solicitações, os agentes coordenadores, agentes executores e agentes de busca. É de responsabilidade dessa camada, o gerenciamento do fluxo dos dados bem como as regras que regem os negócios na sociedade dos agentes.

A última e terceira camada é representada pelas bases de dados e base de conhecimentos, registrando e servindo às aplicações com as informações solicitadas pelos elementos da camada anterior.

Abstract

The automation of the activities carried through for the man is a constant in our days, aiming at to improve the way of these activities are executed.

Objectifying to assist in this process of automation, through the use of the technology of agents, the indirect management and techniques of artificial intelligence, this work considers an architecture of specialized agents ally the application of the theory of the indirect management, for bed the automation of jobs. In this architecture it will be possible to identify the jobs to be given, as well as the way as they will be given, using the features of the society of agents in this model.

This architecture contemplates four elements: Base of knowledge, blackboard architecture, databases and a society of agents. This society of agents through its agents specialized in the fetching of information, interfaces, coordination and execution of tasks uses of the others elements of the architecture to provide the jobs with the maximum automatized form as possible.

This architecture still considers the use of an architecture of execution distributed in three layers, providing flexibility and use of the features for the execution of the jobs.

In this work, the architecture was applied having as purpose the management automatized of services for a secretary of education of the high school, in view of the need and the increasing demand for automation of the processes to the attendance of its customers.