

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**AGENTES INTELIGENTES PARA O PROCESSO DE
NEGOCIAÇÃO EM UM AMBIENTE DE COMÉRCIO
ELETRÔNICO**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À
UNIVERSIDADE REGIONAL DE BLUMENAU PARA A OBTENÇÃO
DOS CRÉDITOS DE DISCIPLINA COM NOME EQUIVALENTE NO
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

LUIS HENRIQUE BOGO

BLUMENAU, JANEIRO DE 2000.

AGENTES INTELIGENTES PARA O PROCESSO DE NEGOCIAÇÃO EM UM AMBIENTE DE COMÉRCIO ELETRÔNICO

LUIS HENRIQUE BOGO

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIO PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Paulo de Tarso Mendes Luna — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

1.1.1.1 BANCA EXAMINADORA

Prof. Paulo de Tarso Mendes Luna

Prof. Roberto Heinzle

Prof. Alejandro Rodriguez Martins

AGRADECIMENTOS

Agradeço aos professores do curso de Bacharelado em Ciências da Computação da Universidade Regional de Blumenau pela sua ajuda prestada no decorrer do curso, principalmente ao professor Paulo de Tarso Mendes Luna por ter me orientado neste trabalho final.

Agradeço aos vários amigos que fiz em Blumenau durante esta grandiosa fase da minha vida que foi este curso de graduação, principalmente ao Jorge, Luiz, Raphael, Roque, Evandro, Heleno e outros que estiveram mais presentes neste período.

Agradeço também aos amigos em Florianópolis, Jordan, Nilson, Cloves, professor Alejandro Martins e outros por estarem dando o apoio necessário nessa etapa final.

Agradeço a várias pessoas que sempre estiveram ao meu lado como minha irmã Raquel, meu cunhado Rodrigo, meus amigos Erasmo, Sandro, Rogério, Nedigard, entre outros.

Agradeço principalmente aos meus pais Irio e Mercedes por terem proporcionado condições e terem dado todo o apoio sempre que necessitei.

SUMÁRIO

1 INTRODUÇÃO	3
1.1 OBJETIVOS	5
1.2 RELEVÂNCIA	5
1.3 ORGANIZAÇÃO DO TEXTO.....	6
2 INTELIGÊNCIA ARTIFICIAL	7
2.1 INTRODUÇÃO	7
2.2 CONCEITOS	7
2.3 INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA (IAD)	8
2.3.1 Sistemas Multiagentes (Mas).....	10
2.3.2 Resolução De Problemas Distribuídos (Dps)	11
2.4 SISTEMAS ESPECIALISTAS.....	12
3 AGENTES INTELIGENTES	15
3.1 CONCEITOS	15
3.2 AGENTES AUTÔNOMOS	16
3.3 PROPRIEDADES DOS AGENTES.....	18
3.4 CLASSIFICAÇÃO DE AGENTES.....	19
3.4.1 Quanto ao Nível de Inteligência	20
3.4.2 Quanto a Tarefa que Executam.....	20
3.4.3 Quanto À Mobilidade	21
3.4.4 Quanto À Ênfase Dada A Alguns Atributos Primários	22
3.5 ARQUITETURA DOS AGENTES	22
3.5.1 Arquiteturas Cognitivas Ou Deliberativas	23
3.5.2 Arquiteturas Reativas.....	23
3.5.3 Arquiteturas híbridas	23
3.6 COMUNICAÇÃO	24
3.6.1 Sistemas de Quadro-Negro	24
3.6.2 Passagem de Mensagens.....	25
3.7 LINGUAGEM DE COMUNICAÇÃO ENTRE AGENTES	26
3.8 APLICAÇÕES DE AGENTES.....	27

3.8.1 Busca e Filtragem de Informações.....	28
3.8.2 <i>NewsWatcher</i>	28
3.8.3 Aconselhamento	28
3.8.4 Entretenimento.....	28
3.8.5 <i>Shopping Agents</i>	28
3.8.6 <i>Customer Service Agents</i> ou <i>Chatter Bots</i>	29
3.8.7 <i>Stock Agents</i>	29
3.8.8 Computação Móvel.....	29
3.8.9 Controle de Processos Industriais.....	30
3.8.10 Gerenciamento de redes e sistemas.....	30
4 NEGOCIAÇÃO	31
4.1 CONCEITOS.....	31
4.2 PROBLEMAS.....	31
4.3 ABORDAGENS.....	32
4.4 REGRAS DE NEGÓCIO.....	33
5 COMÉRCIO ELETRÔNICO	35
5.1 HISTÓRIA.....	35
5.2 CONCEITOS.....	35
5.3 VANTAGENS/DESVANTAGENS	36
5.4 TENDÊNCIAS.....	40
5.5 CATEGORIAS	45
5.6 AGENTES APLICADOS EM COMÉRCIO ELETRÔNICO.....	47
5.6.1 Introdução.....	47
5.6.2 Agentes e Negociação	48
5.6.2.1 Ebay's Auctionbot.....	48
5.6.2.2 Mit Media Lab's Kasbah.....	49
6 PROTÓTIPO	50
6.1 PROJETO.....	50
6.1.1 CONSULTE	50
6.1.2 Processo.....	51
6.1.3 Tecnologia Utilizada	56
6.1.4 Agentes Aplicados Ao Sistema.....	57
6.1.5 Metodologia De Desenvolvimento	58
6.1.5.1 Modelo De Objetos	58
6.1.5.2 Modelo Funcional	60
6.1.5.3 Modelo Dinâmico	61
6.2 IMPLEMENTAÇÃO	62

7 CONCLUSÕES E RECOMENDAÇÕES	75
8 ANEXOS.....	76
ANEXO 1:	76
9 BIBLIOGRAFIA.....	87

LISTA DE FIGURAS

Figura 1: Principais acontecimentos relacionados a IA.....	8
Figura 2: Origem da Inteligência Artificial Distribuída	9
Figura 3: Arquitetura de um Sistema Especialista.....	13
Figura 4: Definição de agentes.....	16
Figura 5: Categorias de agentes inteligentes	17
Figura 6: Sistema de Inteligência Artificial Tradicional.....	18
Figura 7: Agente Autônomo.....	18
Figura 8: Classificação de Agentes quanto a ênfase dada a alguns atributos primários ...	22
Figura 9: Estrutura de Quadro-Negro.....	25
Figura 10: Princípio da Passagem de Mensagem.....	25
Figura 11: Tipos de modelos de comércio eletrônico.....	46
Figura 12: Visualização do sistema CONSULTE	51
Figura 13: Funcionamento básico do protótipo	52
Figura 14: Cadastro de itens de negociação	53
Figura 15: Cadastro de produtos	54
Figura 16: Tela de cadastro de regras referente ao fluxo de caixa	56
Figura 17: Modelo de objetos dos agentes	59
Figura 18: Modelo funcional dos agentes	61
Figura 19: Modelo dinâmico dos agentes	60
Figura 20: Homepage inicial.....	62
Figura 21: Produtos encontrados na base de dados	63
Figura 22: Produtos na cesta de compras.....	64
Figura 23: Homepage de cálculo de preço inicial para negociação.....	65
Figura 24: Tela inicial do processo de negociação	66
Figura 25: Agente informa que proposta não pode ser aceita.....	67
Figura 26: Contra-proposta realizada por agentes	68
Figura 27: Agentes aceitam a proposta.....	69
Figura 28: Proposta sendo feita com fluxo de caixa abaixo do ponto médio	70
Figura 29: Agentes fazem uma contra-proposta avaliando o fluxo de caixa	71
Figura 30: Preço maior sendo proposto.....	72
Figura 31: Agentes informam sobre preço acima do especificado	73
Figura 32: Compra finalizada com sucesso.....	74

LISTA DE TABELAS

Tabela 1: Propriedades dos agentes	19
Tabela 2: Número de usuários da Internet/Brasil (Fonte: http://www.intermanagers.com.br)	44
Tabela 3: Comércio Eletrônico no Brasil (Fonte: http://www.intermanagers.com.br)	45
Tabela 4: Forma de pagamento necessária em relação ao desconto pedido	53
Tabela 5: Preço do concorrente do(s) produto(s) que estão sendo negociados	54
Tabela 6: Desconto/Acréscimo aplicado sobre forma de pagamento	64

RESUMO

Esta proposta visa ilustrar a potencialidade da utilização de agentes inteligentes que realizem o processo de negociação em sistemas de comércio eletrônico para micro e pequenas empresas. A viabilidade da implementação dos agentes inteligentes é mostrada por meio de um protótipo de um sistema de comércio eletrônico. Este protótipo é integrado com o sistema de administração financeira CONSULTE.

A funcionalidade dos agentes inteligentes é determinar o preço ideal para o produto de acordo com a situação financeira da empresa, assim como a forma de pagamento ideal para o momento.

ABSTRACT

This proposal intends to illustrate the potentiality of the intelligent agents' use that accomplish the negotiation process in e-commerce systems for small companies. The viability of the intelligent agents implementation is shown through a prototype of a e-commerce system. This prototype is integrated with a financial administration system called CONSULTE.

The intelligent agents' functionality is to determine the ideal price for the product in agreement with the financial situation of the company, as well as the payment ideal form for the moment.

1 Introdução

Segundo [FAR1998], a Internet é considerada um dos maiores fenômenos do século XX, e tem feito com que as pessoas e organizações tenham que repensar o modo com que executam suas tarefas diárias.

Para [SIE1997], na maioria das indústrias, o comércio eletrônico fatalmente implicará na redefinição de como elas criam novos produtos e colocam no mercado, como elas conseguem clientes para seus produtos e conquistam esses clientes, e como elas se comprometem com consumidores finais, fornecedores e sociedade. Uma das tecnologias que terá um importante papel nesse novo contexto é a Inteligência Artificial, em especial, por meio da tecnologia de agentes inteligentes.

[BIT1998] afirma que o objetivo central da IA (*Inteligência Artificial*) é simultaneamente teórico – a criação de teorias e modelos para a capacidade cognitiva – e prático – a implementação de sistemas computacionais baseados nesses modelos. IAD (*Inteligência Artificial Distribuída*) é definida por [RIC1994] como aquele que é composto por um conjunto de módulos separados (em geral chamados de *agentes*, já que cada módulo assume o papel de uma entidade de solução de problemas) e por um conjunto de caminhos de comunicação entre eles. [BRE1998] faz uma analogia da IAD com as soluções de problemas realizadas por humanos. Segundo ele, IAD faz uso de um aglomerado de especialistas que se utilizam cooperação para resolver problemas complexos, que não poderiam ser resolvidos por uma pessoa individualmente.

Segundo [FLE1996], agentes são programas que travam diálogos, negociam e coordenam transferências de informações. Já segundo [HEI95] define agente como “alguém ou alguma coisa que atua como um representante de outrem, com o propósito de desempenhar ações que são benéficas para a parte representada”. Agentes se diferenciam de outras aplicações por apresentarem mobilidade, autonomia e habilidade de interagir independentemente da presença do usuário. [SOU1996] caracteriza agentes como “sistemas computacionais residentes em ambientes dinâmicos complexos, os quais percebem e atuam autonomamente, e ao fazê-lo realizam um conjunto de objetivos e tarefas para os quais foram

designados”. Por fim, [PAL1998] define agentes como “uma entidade real ou abstrata que é capaz de agir sobre ela mesma e sobre seu ambiente, que dispõe de uma representação parcial deste ambiente, que, em um universo multiagente, pode comunicar-se com outros agentes, e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes”.

No estudo de agentes inteligentes, a tendência atual contempla o desenvolvimento de aplicações modestas sobre domínios restritos. No presente momento a pesquisa vem sendo realizada sobre agentes isolados, tais como agentes de *mail*, agentes de *newse* agentes de pesquisa. Mas, segundo [PAL1998] os primeiros passos rumo à construção de aplicações integradas onde os agentes isolados desempenham o papel de peças elementares vem sendo desenvolvidos. A expectativa é que esta seja a principal tendência para os próximos dois a três anos.

Muitos estudos já foram realizados visando a utilização de agentes inteligentes no comércio eletrônico, mas alguns, como [FAR1998] defendem a principal funcionalidade de um agente seria a procura pelo menor preço, o que implica na questão da negociação.

A negociação foi uma área muito estudada por teóricos durante décadas. Segundo [ZEN1996], o estudo da utilização de agentes autônomos em ambientes eletrônicos para os processos de negociação é uma área em crescimento.

Segundo [BEA1996], a negociação dentro do comércio eletrônico é definida como um processo em que duas ou mais partes, multilateralmente barganham recursos para um ganho múltiplo, usando ferramentas e técnicas de comércio eletrônico. [BEA1996] ainda define negociação eletrônica como uma negociação realizada por computadores interligados, sem a intervenção humana no processo. Os sistemas multiagentes atuais estão mudando o significado das negociações para vários caminhos. Enquanto um dos modos descreve os subproblemas e recursos para a negociação, o outro modo descreve uma negociação direta entre dois agentes. Enquanto o protocolo de negociação provê regras que possibilitam as duas formas de negociação, sua estratégia depende da implementação do próprio agente [BRE1998].

Conforme [FAR1998], atualmente os agentes utilizados em ambientes de comércio eletrônico podem ser divididos em dois grupos, os agentes que procuram produtos e reportam

apenas o preço final e os agentes que realizam o processo de negociação. O grande problema de vários ambientes de comércio eletrônico no Brasil, principalmente de médio e pequeno porte, é que os mesmos resumem-se a mostrar aos clientes apenas o preço final do produto, sem disponibilizar uma possibilidade de negociação, que é chave fundamental para uma grande parte das vendas realizadas.

O presente trabalho visa discutir a utilização de técnicas de Inteligência Artificial, mais especificamente agentes inteligentes, no processo de negociação dentro do comércio eletrônico, assim como a implementação destes agentes e sua aplicação em um sistema de comércio eletrônico. Na especificação dos agentes, será utilizada a metodologia *Object Modeling Technique (OMT)*, definida por [BRE96] e na implementação será utilizado a linguagem de programação *java*.

1.1 OBJETIVOS

O objetivo principal deste trabalho é o desenvolvimento de agentes inteligentes que realizem o processo de negociação em um sistema de comércio eletrônico.

Os objetivos secundários do trabalho são:

- a) estudo da integração entre as tecnologias de agentes inteligentes e sistemas de comércio eletrônico;

1.2 RELEVÂNCIA

A tecnologia de agentes inteligentes, indiscutivelmente, vem trazendo grandes contribuições para os usuários da rede Internet. Com o grande aumento do volume de transações em ambientes de comércio eletrônico, torna-se necessário o estudo de técnicas que automatizem o processo de negociação. Segundo [FAR1998], futuramente, dentro de um ambiente comercial, os consumidores que fazem uso de agentes terão seus próprios assistentes de compra e de pesquisa de tal maneira que conheçam sua personalidade, preferências, objetivos e informações desejadas.

O estudo da tecnologia de agentes inteligentes inseridos neste processo está em uma fase

inicial, mas mostra-se muito viável para atingir-se bons resultados, pois permite mobilidade e principalmente a capacidade de aplicar as regras necessárias para uma negociação.

1.3 ORGANIZAÇÃO DO TEXTO

O trabalho foi desenvolvido observando várias etapas. No capítulo 1 é demonstrada a origem deste trabalho, bem como seus objetivos e a organização aqui apresentada.

No capítulo 2 apresenta-se um estudo resumido sobre a *Inteligência Artificial (IA)*, apresentando maior ênfase a área de sistemas especialistas.

O capítulo 3 aprofunda-se na área da IA denominada de agentes inteligentes, que é o foco deste trabalho. É apresentada uma visão mais abrangente das características de um agente inteligentes, linguagens de comunicação e as principais aplicações de agentes na atualidade.

O capítulo 4 apresenta uma introdução, alguns conceitos e as abordagens que uma negociação pode abranger.

No capítulo 5 é apresentada uma visão sobre comércio eletrônico na atualidade, assim como suas tendências.

O capítulo 6 detalha os principais pontos do protótipo desenvolvido, como a especificação dos agentes, tecnologia utilizada, testes e validação.

No capítulo 7 é apresentada a conclusão sobre a viabilidade de utilizar-se agente inteligentes em ambientes de comércio eletrônico e sugestões para trabalhos futuros.

2 INTELIGÊNCIA ARTIFICIAL

2.1 INTRODUÇÃO

Neste capítulo, é apresentada uma introdução a *Inteligência Artificial (IA)*, destacando os principais conceitos da mesma e aprofundando na subárea denominada de *Inteligência Artificial Distribuída (IAD)*, na qual se enquadram os agentes inteligentes.

2.2 CONCEITOS

Desde a década de 40, procura-se emular o comportamento inteligente humano através de máquinas. Com a popularização e o rápido avanço tecnológico dos computadores, esta tarefa tem se tornado cada vez mais acessível. O termo inteligência artificial (IA) surgiu em 1956, quando um pequeno grupo de pesquisadores, que participavam de um seminário de verão em Dartmouth College (EUA), discutiam como o computador poderia simular o raciocínio humano. Abaixo segue algumas definições do termo. Não há uma definição única de o que é IA.

O que hoje chamamos de IA é um ramo da ciência da computação ao mesmo tempo recente (oficialmente a IA nasceu em 1956) e muito antigo, pois a IA foi construída a partir de idéias filosóficas, científicas e tecnológicas herdadas de outras ciências, algumas tão antigas quanto a lógica, com seus 23 séculos [BIT98]. Já segundo [BIG98], a ciência de IA possui aproximadamente 40 anos, sendo que seu início ocorreu em uma conferência ocorrida no ano de 1958 em Dartmouth.

[BIT98] afirma que o objetivo central da IA é simultaneamente teórico – a criação de teorias e modelos para a capacidade cognitiva – e prático – a implementação de sistemas computacionais baseados nesses modelos.

Já [RIC94] elaborou uma definição mais simples de IA, segundo o autor IA é o estudo de como fazer os computadores realizarem coisas, que no momento, as pessoas fazem melhor.

Apesar de sua definição como campo de estudo ocorrer somente em 1956, algumas áreas foram vitais para o seu surgimento, pois tecnologias inteligentes sempre foram objetos

de pesquisa e desenvolvimento desde o início da ciência da computação. Na figura 1 é apresentada uma visualização dos principais acontecimentos relacionados com a IA.

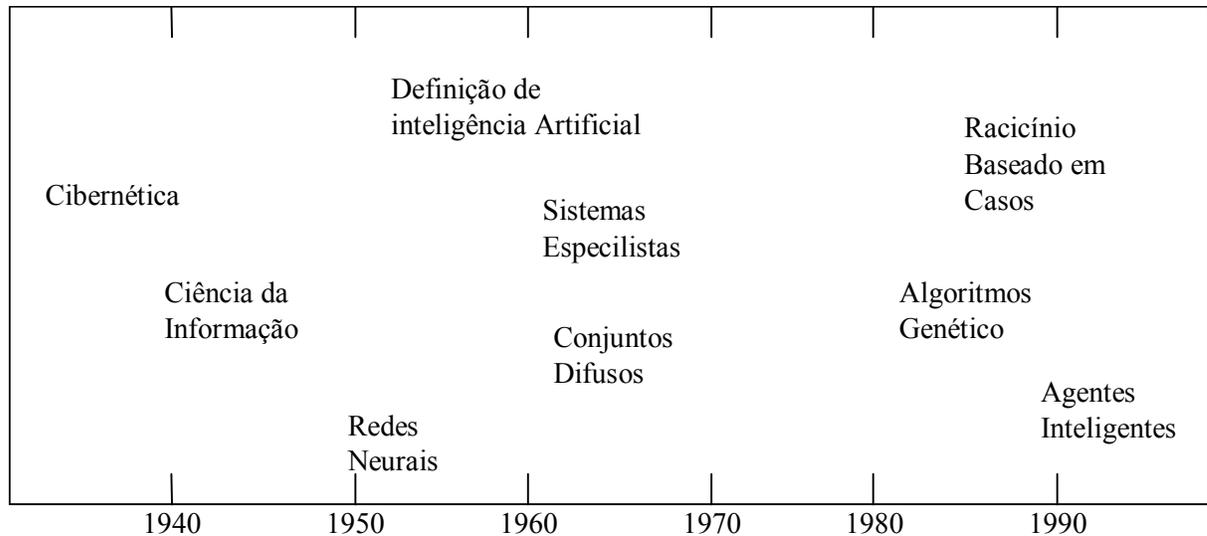


Figura 1: Principais acontecimentos relacionados a IA

2.3 INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA (IAD)

Nas abordagens Clássicas de Inteligência Artificial (IA), a ênfase da inteligência é baseada em um comportamento humano individual e o foco de atenção volta-se à representação de conhecimento e métodos de inferência. Já a Inteligência Artificial Distribuída (IAD), é baseada em comportamento social e sua ênfase é para cooperações, interações e para o fluxo de conhecimento entre unidades distintas. IAD é definida por [RIC94] como aquele que é composto por um conjunto de módulos separados (em geral chamados de *agentes*, já que cada módulo assume o papel de uma entidade de solução de problemas) e por um conjunto de caminhos de comunicação entre eles. Uma definição mais simples é dada por [GIE1998] que define IAD como a intersecção da Computação Distribuída e da Inteligência Artificial, conforme figura 2.



Figura 2: Origem da Inteligência Artificial Distribuída

Segundo [BIT98], IAD estuda o conhecimento e as técnicas de raciocínio que podem ser necessárias ou úteis para que agentes computacionais participem de sociedade de agentes.

Pode-se destacar várias vantagens dos sistemas distribuídos em relação aos sistemas monolíticos, como as seguintes descritas por [RIC94]:

- a) **Modularidade do Sistema:** É mais fácil criar e manter uma coleção de módulos quase independentes do que um módulo imenso;
- b) **Eficiência:** Nem todos os conhecimentos são necessários para todas as tarefas. Quando o modularizamos, é ganho a possibilidade de focalizar os esforços do sistema de solução de problemas do modo mais compensador possível;
- c) **Arquiteturas Rápidas:** Quando os resolvidores de problemas começam a ficar complexos, eles passam a precisar de mais e mais ciclos. Apesar de as máquinas estarem cada vez mais rápidas, a real aceleração vem não de um único processador com uma enorme memória associada a ele, mas de um agrupamento de processadores menores, cada um com memória própria. Os sistemas distribuídos têm melhores condições de explorar essas arquiteturas;
- d) **Raciocínio Heterogêneo:** As técnicas de solução de problemas e os formalismos de representação de conhecimento que mais se prestam para trabalhar em uma parte de um problema podem não ser os melhores para trabalhar em outra parte;
- e) **Múltiplas Perspectivas:** O conhecimento exigido para solucionar um problema pode não estar na cabeça de uma única pessoa. É muito difícil juntar diversas pessoas para criar uma única base de conhecimento coerente, e às vezes essa

reunião é até mesmo impossível, porque os modelos de cada uma delas tem sobre o domínio em questão são inconsistentes;

- f) **Problemas Distribuídos:** Alguns problemas são inerentemente distribuídos. Por exemplo, pode haver dados diferentes disponíveis em cada uma das várias localizações físicas distintas;
- g) **Confiabilidade:** Se um problema está distribuído entre agentes de sistemas diferentes, a solução pode continuar mesmo que um sistema apresente alguma falha.

[BRE98] faz uma analogia da IAD com as soluções de problemas realizadas por humanos. Segundo ele, IAD faz uso de um aglomerado de *experts* que se utilizam cooperação para resolver problemas complexos, que não poderiam ser resolvidos por uma pessoa individualmente.

2.3.1 SISTEMAS MULTIAGENTES (MAS)

Segundo [FAR98], nos sistemas MAS, os agentes são a parte central que devem cooperar e trocar conhecimento para obter a solução de problemas antes desconhecidos, ou seja, o projetista não volta sua atenção para um problema específico, mas para um domínio específico. Assim o problema não é necessariamente específico e o processo de coordenação das ações dos agentes ocorre em tempo de execução.

A forma menos distribuída do raciocínio distribuído é aquela em que um único agente [RIC94]:

1. Decompõe o objetivo em sub-objetivos;
2. Atribui as submetas aos vários outros agentes.

Este tipo de raciocínio em geral é chamado de planejamento multiagente. A primeira etapa, decomposição do problema, é essencialmente a mesma de sistemas de planejamento com um único agente. O ideal seria que a decomposição resultasse em um conjunto de subproblemas mutuamente independentes.

Uma vez produzida a decomposição, o subproblema precisa ser alocado aos agentes disponíveis para execução. Neste ponto, o planejamento distribuído difere do planejamento com um único agente nos seguintes aspectos importantes:

- a) Se os agentes escravos não forem todos idênticos, o agente-mestre precisará ter acesso aos modelos dos recursos dos vários escravos. Esses modelos permitem a alocação de tarefas aos agentes com melhores condições para executá-las;
- b) Mesmo que todos os agentes escravos sejam idênticos, o mestre precisa fazer um balanço da carga para assegurar que o objetivo global seja concluído o mais rápido possível;
- c) Uma vez distribuídas as tarefas, é necessária uma sincronização entre os agentes, caso elas não sejam completamente independentes. No planejamento com um único agente, as dependências em geral são trabalhadas no momento da criação do plano. Em um sistema com vários agentes, nem sempre é possível fazer isso, já que um esquema tão estático estaria fadado ao fracasso, caso os vários agentes demorassem um tempo imprevisível para executar suas tarefas.

2.3.2 RESOLUÇÃO DE PROBLEMAS DISTRIBUÍDOS (DPS)

Segundo [FAR98], a DPS se caracteriza por ter como objetivo principal o problema a ser resolvido, ou seja, o centro de tudo está na solução de um problema particular. Assim, toda a arquitetura de um sistema DPS é projetada para resolver esse tipo de problema e não um problema genérico qualquer.

De acordo com [RIC94], neste tipo de sistema não há um agente controlador central, por causa disto não é possível fazer nenhuma suposição sobre como os vários agentes irão se comportar. Usando essas idéias, podem ser definidas técnicas que um agente isolado pode usar para atingir seus objetivos, levando em consideração o que provavelmente acontecerá como resultado do que outros agentes do seu ambiente possam a vir fazer. As vezes, eles estão trabalhando em outros objetivos, que podem ser competitivos, ou simplesmente ortogonais. Duas classes de abordagens podem ser consideradas a este problema:

- a) Planejamento com comunicação;
- b) Planejamento sem comunicação.

A primeira abordagem é aquela em que os agentes podem comunicar-se livremente entre si durante a solução do problema. Neste caso, os agentes podem criar seus próprios planos, que são compostos por ações de solução de problemas e de ações de comunicação. As vezes, as ações de comunicação são endereçadas a um agente específico, que supostamente é

capaz de satisfazer uma solicitação (solicitação de informações ou para que uma outra tarefa seja executada). Em outros sistemas, os agentes não sabem explicitamente uns dos outros. Eles podem anunciar em uma estrutura de memória compartilhada, que certamente será lida por todos os outros.

A segunda abordagem é aquela em que se assume que não pode haver comunicação entre os agentes. Essa restrição pode parecer muito séria, mas é útil considerá-la, tanto porque ela, às vezes, surge em situações extremas (talvez porque os agentes estão isolados geograficamente), quanto porque elas sempre surgem nos níveis de pequena granularidade em que o custo da comunicação constante pode acabar dominando o custo da própria solução do problema.

Em geral, os agentes representam seu conhecimento sob a forma de regras, atuando como mini-sistemas especialistas autônomos. Na seção a seguir, são apresentados os conceitos básicos referentes aos sistemas especialistas.

2.4 SISTEMAS ESPECIALISTAS

O desenvolvimento dos Sistemas Especialistas (SE) começou de maneira tímida, nos anos cinquenta, com programas de jogos de xadrez e programas de resolução automática de teoremas. Durante os anos sessenta teve um grande impulso com o projeto DENDRAL, uma encomenda da NASA para a Universidade de Stanford. Este sistema tinha por objetivo desenvolver um programa para analisar a composição química do solo marciano, com base em dados de análises de espectrofotometria de massa [DUR1996]. [DUR1996] define este projeto como o início da era dos SE. Desde então, esta técnica tem sido aplicada em diversas áreas, destacando-se medicina, engenharia, química e economia; nas mais diferentes tarefas, tais como: interpretação, planejamento e projeto, consultoria, diagnóstico, monitoramento, controle e simulação.

Os SE tem sido definidos de muitas maneiras. Porém, todas definições compartilham a mesma idéia de que são sistemas artificiais usados para emular a *expertise* de especialistas na resolução de problemas.

Conforme [BIT1998], os sistemas especialistas (SE) atuais apresentam três módulos principais: uma *base de regras*, uma *memória de trabalho* e um *motor de inferência* (ver figura 3). A base de regras e a memória de trabalho formam a chamada *base de conhecimento* do SE, onde está representado o conhecimento sobre o domínio. O motor de inferência é o mecanismo de controle do sistema que avalia e aplica as regras de acordo com as informações da memória de trabalho.

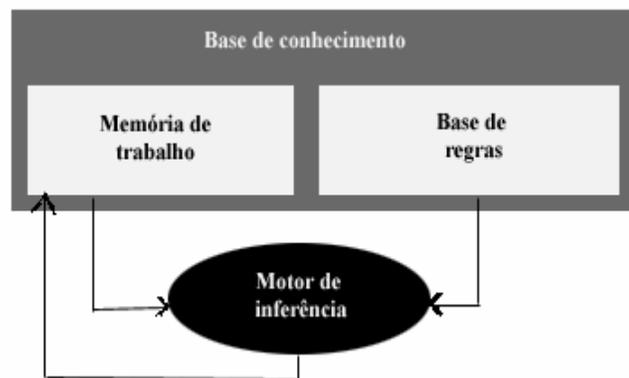


Figura 3: **Arquitetura de um Sistema Especialista**

A memória de trabalho, no modelo generalizado pode conter qualquer tipo de estrutura de dados. Mais do que estruturas de dados, as memórias de trabalho de SE's devem respeitar a forma como as regras estão representadas na base de regras, isto é, uma linguagem formal e uma descrição matemática de seu significado.

A base de regras passa a conter condições que representam “perguntas” à representação de conhecimento da memória de trabalho. Estas perguntas podem ser de diferentes tipos, mas em geral envolvem variáveis a serem instanciadas e, eventualmente, algum tipo de inferência. A sintaxe das regras varia de acordo com o sistema e pode ser bastante flexível e próxima da linguagem natural. Como exemplo, pode-se usar as regras de produção, que são uma forma de representar o conhecimento.

Em geral uma regra é representada por conjuntos de condições, conclusões e ações. Se as condições são verdadeiras, a regra é disparada e as ações são desencadeadas. Ou seja, uma regra é uma forma de conhecimento procedural, associando uma informação a alguma ação (Durkin, 1994).

As regras consistem de duas partes: a primeira, chamada de antecedente, ou premissa,

ou condição, ou parte IF (se); e a segunda, chamada de conseqüente, ou conclusão, ou ação, ou parte THEN (então). Abaixo é apresentada a estrutura geral de uma regra:

IF a bateria do carro está sem carga THEN o caro não irá pegar

As regras podem ainda possuir conectivos lógicos ligando seus antecedentes. Estes podem ser os conectivos AND e OR, como no exemplo abaixo:

IF o carro não pega AND tem gasolina THEN verifique o sistema elétrico

Uma regra pode conter também uma sentença ELSE, que será verificada a partir do momento que um ou mais antecedentes forem falsos. Por exemplo:

IF hoje é Domingo AND e o céu está limpo AND ensolarado THEN vou a praia ELSE vou ao shopping

Na sentença acima se não for Domingo ou o céu não estiver limpo e ensolarado, devo ir ao shopping.

O motor de inferência controla a atividade do sistema. Essa atividade acontece em ciclos, cada ciclo consistindo em três fases:

1. Correspondência de dados, onde as regras que satisfazem a descrição da situação atual são selecionadas;
2. Resolução de conflitos, onde as regras que serão realmente executadas são escolhidas dentre as regras que foram selecionadas na primeira fase, e ordenadas;
3. Ação, a execução propriamente dita das regras.

A chave para o desempenho de um SE está no conhecimento armazenado em suas regras e em sua memória de trabalho. Este conhecimento deve ser obtido junto a um especialista humano do domínio e representado de acordo com as regras formais definidas para a codificação de regras no SE em questão.

3 AGENTES INTELIGENTES

3.1 CONCEITOS

Durante os anos 80, a comunidade de Inteligência Artificial, desencorajada pela falta de progresso, após 30 anos de pesquisa em projetos de sistema, começou a explorar novas áreas onde sistemas de AI pudessem ter um domínio mais dinâmico. Ao invés de olhar para resultados simulados, simbólicos em mundos artificiais, começaram a explorar as possibilidades de interações complexas com o mundo físico, através de um mecanismo denominado agentes.

O advento de agentes inteligentes origina muita discussão sobre o que são exatamente, e de como eles diferem de programas em geral. Com isso, diferentes pesquisadores propõem diversas definições para esta área de Inteligência Artificial. Essas definições vão desde o nível elementar até o nível mais elaborado. Além disso, estes pesquisadores criaram também novos termos para referenciar os agentes inteligentes, tais como: interfaces inteligentes, interfaces adaptativas, *knowbots*, *softbots*, *taskbots*, entre outras. Verificou-se que cada autor procura enfatizar as características desejáveis de um agente de acordo com a aplicação por ele desenvolvida [MOD2000]. Na comunidade de IA não existe uma definição única de agentes inteligentes, sendo assim várias são as definições correntes:

- a) Chama-se agente uma entidade real ou abstrata que é capaz de agir sobre ela mesma e sobre seu ambiente, que dispõe de uma representação parcial deste ambiente, que, em um universo multiagente, pode comunicar-se com outros agentes, e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes [FER91].
- b) Agentes são sistemas computacionais residentes em ambientes dinâmicos complexos, os quais percebem e atuam autonomamente, e ao fazê-lo realizam um conjunto de objetivos e tarefas para os quais foram designados [SOU96].
- c) Agentes são programas que travam diálogos, negociam e coordenam transferências

de informações [FLE96].

Atualmente, a definição mais aceita descreve um agente como sendo um programa de software que auxilia o usuário na realização de alguma tarefa ou atividade. Agente é a palavra utilizada para designar uma entidade inteligente e autônoma. A palavra autônoma, neste caso, significa que cada agente possui sua própria existência, a qual não é dependente da existência de outros agentes. A figura 4 demonstra esta definição:

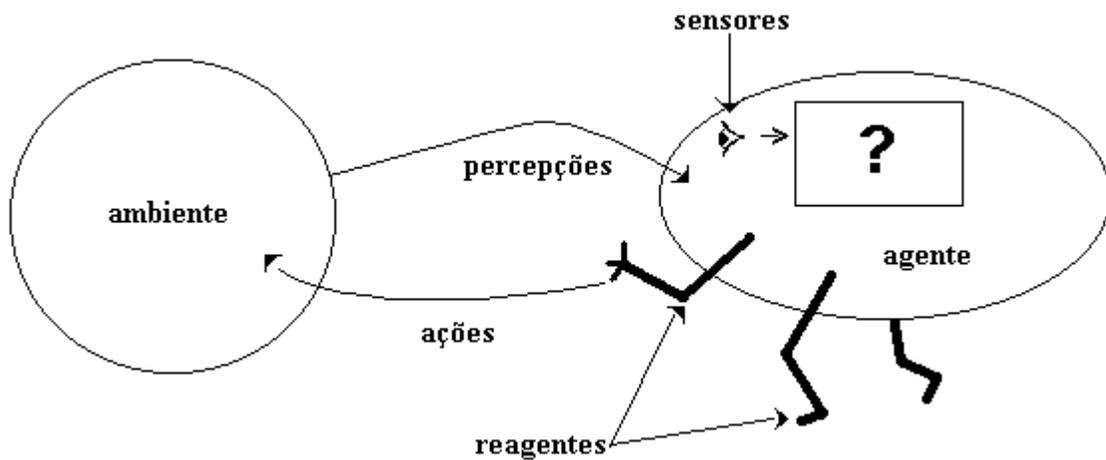


Figura 4: Definição de agentes

3.2 AGENTES AUTÔNOMOS

Os agentes autônomos são capazes de interagir independentemente e efetivamente com seus ambientes, por meio de seus próprios sensores ou atuadores, para concluir alguma tarefa dada ou gerada automaticamente. Assim humanos e muitos animais podem, neste sentido, ser vistos como agentes autônomos. [GIE1998]

Em um nível mais alto, os agentes podem ser distinguidos em três categorias: agentes humanos, agentes de hardware e agentes de software. [BRE1998]

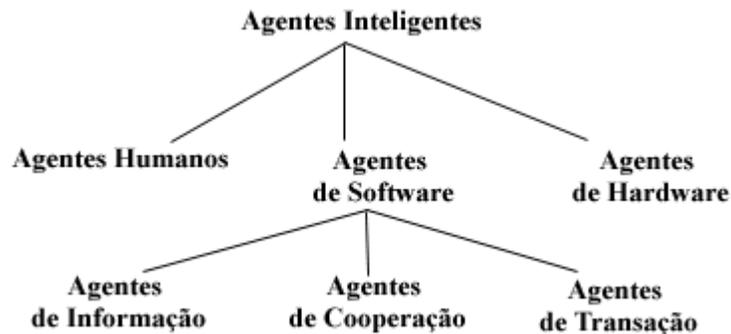


Figura 5: **Categorias de agentes inteligentes**

Os agentes de softwares podem ser mais facilmente entendidos se comparados com agentes humanos ou agentes robôs. Um agente pode ser visto como algo que observa o ambiente através de sensores e age nesse ambiente através de atuadores. Comparando, um agente humano possui olhos, ouvidos e outros órgãos para “sensoriar” o ambiente, assim como mãos, pernas, bocas e outras partes do corpo para agirem nesse ambiente. Por outro lado um agente robô substitui esses elementos por câmeras e outros sensores para observar o ambiente e tem vários motores como atuadores. Um agente de software possui bits codificados (código) para a percepção e ação.

Os agentes autônomos diferem de Sistemas de Inteligência Artificial tradicionais (ou sistemas de computador) nos seguintes aspectos [GIE1998] :

- Os sistemas de **Inteligência Artificial tradicional** tipicamente necessitam de um usuário que observa o ambiente (o problema) e descreve as partes relevantes dele para o computador. O resultado do processamento (execução das tarefas) do computador é interpretado pelo usuário que então executa a ação requerida. Assim o sistema pode ser visto como uma ferramenta, estendendo suas capacidades (mental) do usuário, mas é inútil sobre si próprio, pois ele é dependente das capacidades básicas (mental e física) do usuário, conforme figura 6.

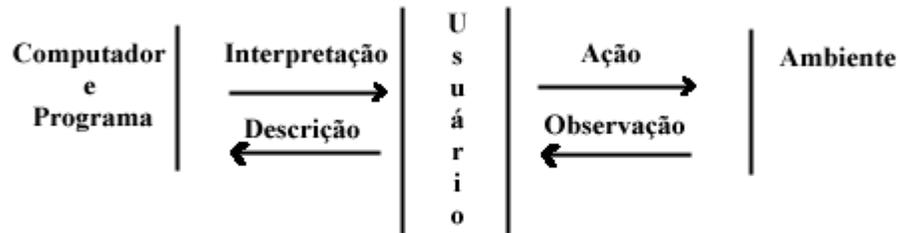


Figura 6: Sistema de Inteligência Artificial Tradicional

- Um **agente autônomo**, ao contrário dos sistemas de inteligência artificial tradicional, pode observar o sistema por si próprio e tornar essas observações em descrições de futuros processamentos. Entretanto, ele pode também por si próprio, interpretar o resultado de seu processamento e então executar a ação apropriada.



Figura 7: Agente Autônomo

3.3 PROPRIEDADES DOS AGENTES

O comportamento dos agentes dentro de uma sociedade é definido por suas propriedades. [FAR98] define algumas dessas propriedades:

PROPRIEDADE	DESCRIÇÃO
Autonomia	Capacidade de tomar ações conduzindo para o término de algumas tarefas e objetivos, sem a interferência do usuário final.
Aprendizagem	Capacidade de alterar o seu comportamento baseado em experiências prévias.
Comunicabilidade	Capacidade dos agentes de se comunicar com outros agentes e pessoas.
Comportamento Adaptativo	Capacidade do agente de modificar seu comportamento em função de experiências anteriores.
Confiabilidade	Capacidade do agente de demonstrar veracidade e benevolência nas informações e ações realizadas em nome do usuário.
Cooperatividade	É a capacidade de os agentes inteligentes trabalharem juntos alcançando o objetivo geral através da ajuda mútua.
Degradação Gradual	É a capacidade do agente de ainda completar a execução de uma tarefa mesmo se alguma anomalia no sistema esteja ocorrendo
Discurso	Propriedade onde o usuário e agente concordam sobre o que deve ser feito e por quem deve ser feito.
Flexibilidade	Habilidade dos agentes em escolher que ações e em que sequências deve tomá-las em resposta a um evento do ambiente.
Inteligência	Propriedade de um agente que o habilita de negociar efetivamente com ambigüidades.
Mobilidade	É a capacidade de transportar-se de uma máquina para outra.
Persistência	Capacidade do agente de manter um estado interno conciso através do tempo.
Personalização	Capacidade do agente de aprender com o usuário e adaptar suas ações de acordo com ele.
Planejamento	Habilidade de sintetizar e escolher entre diferentes opções de ações desejadas para atingir os objetivos.
Pró-atividade ou Orientação ao Objetivo	Não respondem apenas ao ambiente, mas perseguem um objetivo.
Reatividade	Respondem de forma oportuna às mudanças no ambiente.
Representatividade	Capacidade do agente de representar o usuário através de suas ações.
Responsabilidade	Descrição intencional da natureza de uma tarefa agendada.
Sociabilidade	Interagir com outros agentes através de algum tipo de linguagem de comunicação.
Veracidade	Suposição de que um agente não comunicará informações falsas.

Tabela 1: Propriedades dos agentes

O conjunto de propriedades que um agente possui define o comportamento do agente. Como diferentes agentes podem possuir diferentes conjuntos de propriedades, tem-se então agentes com comportamentos diferentes, os quais são classificados de diferentes maneiras.

3.4 CLASSIFICAÇÃO DE AGENTES

Conforme [GIE1998], os agentes podem ser classificados de diversas formas:

quanto ao nível de inteligência, quanto a tarefa que executam, quanto a mobilidade e quanto a ênfase dada a alguns atributos primários.

3.4.1 QUANTO AO NÍVEL DE INTELIGÊNCIA

As aplicações com agentes apresentam diferentes níveis de inteligência, podendo ser classificadas nos seguintes níveis:

- a) **Baixo:** Neste nível, os *softwares* agentes desempenham tarefas rotineiras, disparadas por eventos externos. Estes agentes executam redes de regras complexas, não se adaptam as mudanças e não demonstram oportunismo com o passar do tempo;
- b) **Médio:** Estes agentes utilizam uma base de conhecimento para desenvolver raciocínio em eventos monitorados. Podem adaptar-se à mudanças de condições na base de conhecimento e manipular novas condições, porém, normalmente não demonstram oportunismo; e
- c) **Alto:** Neste nível de inteligência, os *softwares* de agentes utilizam tanto aprendizado quanto raciocínio na base de conhecimento. Aprendem com o comportamento do usuário, podem adaptar-se as mudanças e demonstram oportunismo com o passar do tempo.

3.4.2 QUANTO A TAREFA QUE EXECUTAM

Os agentes podem ser classificados quanto a tarefa que executam da seguinte forma:

- a) **Gopher:** são considerados os agentes que executam as tarefas mais simples, baseando-se em suposição e regras pré-especificadas. Por exemplo, o agente pode avisar o usuário que ele possui uma reunião marcada para sexta-feira as 2:00hs;
- b) **Prestadores de serviço:** são agentes que executam tarefas de alto nível, bem definidas, quando requisitadas pelo usuário. Estes agentes podem organizar uma reunião (negociar datas e horários da reunião com os participantes); e
- c) **Pró-Ativo / Preditivo:** são agentes que executam as tarefas mais complexas, eles podem pesquisar informações ou executar tarefas para o usuário sem serem requisitadas, sempre que isso for julgado apropriado. Por exemplo, um agente pode monitorar novos grupos sobre a *Internet* e retornar discussões que ele acredita serem de interesse do usuário.

- d) **Conselheiros:** oferece ajuda e treinamento. Ensina os passos iniciais para usar um determinado sistema. Fornece suporte contínuo, observando todas as ações do usuário, as quais ele pode interceptar e pedir confirmação. Pode ser consultado para mostrar como executar uma atividade particular, ou então, sugerir métodos alternativos e mais rápidos para executá-la.
- e) **Guia:** ajuda a navegação em bancos de dados e hipermídia. Classifica, recupera e filtra grandes quantidades de informações, apresentando somente os dados relevantes e importantes aos usuários, no formato personalizado. Fornece caminhos apropriados para o usuário navegar pelo banco de dados, e auxilia-o caso se sinta "perdido".
- f) **Empregados:** executa as atividades tediosas ou repetitivas. Atividades são executadas imediatamente, e algum tipo de feedback pode ser fornecido tanto pelo usuário como pelo próprio agente.
- g) **Representantes:** trabalha na ausência do usuário. De certa forma, seria parecido ao agente Empregado citado anteriormente, exceto pelo fato de que as atividades não precisam ser imediatamente executadas ou então, são executadas somente após eventos específicos. Por exemplo, pode fazer backup de arquivos de madrugada ou fazer pedidos de compras, caso algum produto atinja o limite mínimo no estoque.
- h) **Comunicadores:** trabalha com outros usuários e seus agentes, para assim, conseguir executar as atividades as quais é incumbido. Pode, por exemplo, organizar reuniões, reunindo recursos e pessoas. Ou então, pode reunir um grupo de agentes para que assim possam executar uma atividade mais complexa.

3.4.3 QUANTO À MOBILIDADE

Os agentes podem ser classificados de acordo com sua capacidade de se mover na rede em:

- a) **Agentes estáticos:** são aqueles que não podem mover-se através da rede, ou seja, eles são fixos em um determinado local.
- b) **Agentes móveis:** são agentes que possuem a habilidade para mover-se através da rede.

3.4.4 QUANTO À ÊNFASE DADA A ALGUNS ATRIBUTOS PRIMÁRIOS

[GIE1998] define que os agentes podem ser classificados levando em conta a ênfase dada para alguns atributos primários e considerados ideais (cooperação, aprendizado e autonomia) em: agentes inteligentes, agentes de aprendizado, agentes de interface e agentes colaborativos, conforme figura 8.

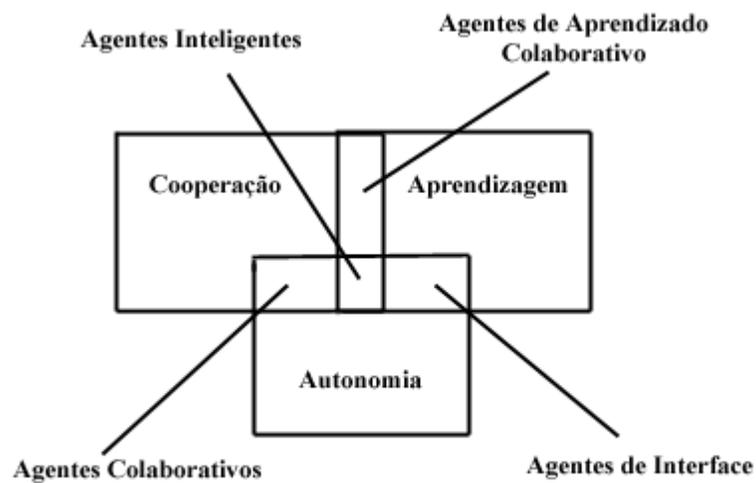


Figura 8: **Classificação de Agentes quanto a ênfase dada a alguns atributos primários**

É bom ressaltar que estas distinções não são definitivas. Por exemplo, os agentes deliberativos, possuem mais ênfase sobre as propriedades de cooperação e autonomia do que sobre aprendizado, isto não quer dizer que agentes colaborativos nunca aprendem.

3.5 ARQUITETURA DOS AGENTES

As propriedades, sua estrutura e a maneira de como os módulos que compõe o sistema é que definem a arquitetura de um sistema. [FAR98] separa as arquiteturas em três tipos principais:

3.5.1 ARQUITETURAS COGNITIVAS OU DELIBERATIVAS

Essa arquitetura se caracteriza pela complexidade dos agentes, onde estes apresentam mecanismos de inferência de decisão robustos, inferências sofisticadas e alto grau de intenção no comportamento, ou seja, esta arquitetura possui um modelo simbólico do mundo, representado explicitamente, onde as decisões são efetuadas através de raciocínio lógico.

Alguns exemplos:

- a) **Arquiteturas funcionais:** Arquitetura deliberativa onde os agentes são formados por módulos funcionais que são necessários para a operação adequada do sistema.
- b) **Arquiteturas baseadas em estados mentais:** Adota uma perspectiva psicológica para a definição da estrutura dos agentes. Os agentes possuem estados mentais a respeito do domínio em que se encontram, crenças, capacidade de escolhas e compromisso.

3.5.2 ARQUITETURAS REATIVAS

Uma arquitetura reativa não inclui nenhum tipo de modelo do mundo simbólico e não utiliza raciocínio simbólico complexo, somente reage a ações que ocorrem no ambiente.

Exemplo:

- a) **Arquiteturas de suposição (*Subsumption*):** Possui uma hierarquia de níveis de comportamento tarefa-ação, onde cada comportamento compete com outros para ter controle sobre um robô. Essa arquitetura surgiu da necessidade de um mecanismo para controle de robôs móveis e autônomos.

3.5.3 ARQUITETURAS HÍBRIDAS

A arquitetura híbrida mistura componentes das arquiteturas deliberativas e reativas com o objetivo de torná-la mais adequada e funcional para a construção dos agentes. Neste caso, os agentes são dotados de um comportamento reativo com relação aos eventos que ocorrem no ambiente e comportamento deliberativo onde existe uma definição simbólica do mundo para a tomada de decisões.

Exemplos:

- a) **Arquiteturas *TouringMachines*:** A camada reativa é implementada com um conjunto de regras de situação-ação, onde a partir de eventos que ocorrem no ambiente executam-se ações potenciais. A camada de planejamento constrói planos e seleciona

ações para executar, a fim de alcançar os objetivos do agente. Por fim a camada de modelagem contém os modelos ou representações dos estados cognitivos de outras entidades no ambiente dos agentes.

- b) **InteRRaP**: arquitetura de camadas onde o nível de abstração de cada camada é maior em relação a camada anterior. Possui cinco componentes básicos: interface que implementa facilidades como ação, percepção e comunicação, componentes de comportamento, que implementa comportamento reativo e o conhecimento do agente, componentes de planejamento, visando encontrar planos para resolver algum problema que o agente possua, componentes de cooperação que são mecanismos para projetar planos conjuntos e estratégias de comunicação visando a criação de planos que atendam metas globais e a base de conhecimento do agente composta por ações primitivas, padrões de comportamento, metas, planos, conhecimento dos outros agentes e procedimentos de cooperação.

As arquiteturas híbridas serão provavelmente as mais utilizadas, tendo em vista que esta tem recebido maiores esforços de pesquisadores. Tais arquiteturas mostram-se mais adequadas por aproveitarem as vantagens de ambas arquiteturas deliberativas e reativas. Contudo é certo que tais arquiteturas e as atuais pesquisas demonstram dificuldades para representarem comportamentos abstratos, na qual são realizadas com certa facilidade por humanos.

3.6 COMUNICAÇÃO

A comunicação entre os agentes pode ser feita de várias maneiras. Eles podem conversar diretamente um com o outro, falando a mesma linguagem. Ou podem conversar utilizando-se de um interprete ou facilitador, sendo que estes saberão qual agente pode conversar com qual [BIG98].

3.6.1 SISTEMAS DE QUADRO-NEGRO

[FAR98] define quadro-negro como uma estrutura de dados geralmente persistente onde existe uma divisão em regiões ou níveis, visando facilitar a busca de informações. Nesta

estrutura, todas as interações entre os agentes se dão através dessa estrutura onde os mesmos escrevem e lêem suas mensagens no quadro quando necessário. Pode-se dizer que um quadro-negro é uma memória de compartilhamento global onde existe uma quantidade de informações e conhecimento usados para escrita e leitura pelos agentes.

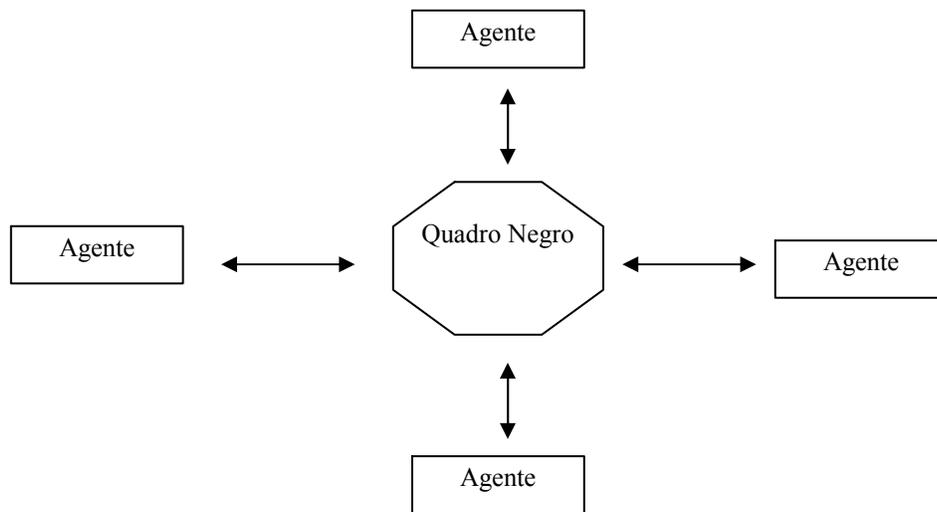


Figura 9: Estrutura de Quadro-Negro

3.6.2 PASSAGEM DE MENSAGENS

Na comunicação utilizando-se de troca de mensagens as mensagens são trocadas diretamente entre os agentes [BRE98].

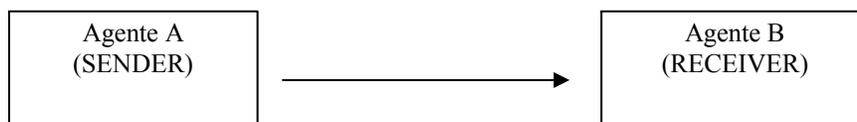


Figura 10: Princípio da Passagem de Mensagem

Para que este processo funcione corretamente é necessário que os agentes saibam os nomes e endereços uns dos outros para que as mensagens sejam devidamente endereçadas. Além disso, é necessário um protocolo bem definido para a comunicação. Segundo [FAR98],

esse método é mais eficiente no sentido de obter as mensagens em tempo hábil, mas por outro lado, sua implementação é dificultada pelo crescimento exponencial das diferentes mensagens trocadas pelos agentes que compõe a sociedade.

3.7 LINGUAGEM DE COMUNICAÇÃO ENTRE AGENTES

Como há uma grande variedade de software no mercado, criados por diversas pessoas, dos mais diferentes países, credos e cultura, há uma enorme diferença entre a forma e estrutura de seus programas. Como os agentes se propõem a comunicar entre os aplicativos do usuário, sejam quais forem, há uma necessidade de criar-se um padrão de comunicação. É necessário, portanto, determinar um padrão para comunicação entre agentes, para depois de criado e testado, o mesmo passe a ser usado também nas diversas aplicações. Criar um padrão não é uma tarefa simples. Há a necessidade de ter-se um grupo de desenvolvimento. Este grupo tem que levar em consideração a heterogeneidade dos programas, escritos em diferentes momentos, por pessoas distintas, em diferentes linguagens, com interfaces também diferentes. Para não ficar restrito aos problemas decorrentes da criação de um padrão, discute-se a seguir um exemplo de tentativa de definição de padrão.

O grupo DARPA *Knowledge Sharing Effort*, usa a abordagem declarativa, que é baseada na idéia de que a comunicação pode ser modelada com a troca de afirmações declarativas, sendo, porém, ao mesmo tempo compactada e suficientemente expressiva para comunicar uma grande quantidade de tipos de informações. Este grupo definiu uma Linguagem de Comunicação de Agentes (ACL), que satisfaz estas necessidades. A ACL pode ser melhor imaginada como sendo constituída de três partes - seu vocabulário, uma linguagem interna chamada KIF (*knowledge Interchange Format*) e uma outra linguagem chamada KQML(*Knowledge Query and Manipulation Language*). Uma mensagem ACL é uma expressão KQML na qual os argumentos são termos ou sentenças em KIF, formadas por palavras encontradas no vocabulário da ACL. O vocabulário da ACL é armazenado em um dicionário com as palavras apropriadas às áreas comuns das aplicações. Cada palavra tem notações formais (em KIF) e descrição em inglês. É um dicionário aberto, permitindo

acréscimo de palavras nas áreas já existentes ou então de novas áreas que possam surgir. Tem-se então uma linguagem padrão de comunicação entre agentes [FIN1993].

Mas como ficam os programas já escritos, ou seja, as aplicações que o usuário tem e que gostaria que agentes lhe auxiliassem? Temos, então, algumas abordagens: Transdutor, Empacotador e Reescrita.

Na abordagem Transdutor a aplicação envia a mensagem, o transdutor a transforma em ACL e passa ao agente. Quando da resposta, o transdutor converte na linguagem da aplicação. Tem-se alguns problemas: tempo de resposta (a tradução de ida e volta), a base de conhecimento do transdutor (quanto mais souber, mais aplicações poderá traduzir, e maior espaço ocupará). Mas a vantagem de funcionar em qualquer aplicativo. Na Empacotador, insere-se código dentro da aplicação para permitir a comunicação. Isto permite ao empacotador examinar diretamente as estruturas de dados do programa e mudá-las. Torna a comunicação mais eficiente, por não ter que traduzir e diminuir a comunicação serial, além de permitir que programas que antes não podiam se comunicar, agora tem esta facilidade. Porém, necessita que o código original esteja disponível. Finalmente, na Reescrita é a atitude mais drástica, reescrever o programa todo. Obviamente aumenta em muito a eficiência e capacidade do programa, muito além das anteriores, mas altera de tal forma o código, que praticamente o programa 'ganha' uma nova versão.

3.8 APLICAÇÕES DE AGENTES

Agentes são atualmente a base para o desenvolvimento de uma grande variedade de aplicações, que vão desde as mais simples, como filtro de e-mail, até sistemas complexos e críticos, como o controle de tráfego aéreo. Esta seção introduz algumas áreas de desenvolvimento em que a tecnologia de agentes tem sido aplicada com sucesso. As principais áreas, atualmente, em evidência são descritas brevemente a seguir:

3.8.1 BUSCA E FILTRAGEM DE INFORMAÇÕES

O rápido aumento do volume de informações traz consigo a necessidade de ferramentas (agentes) eficientes e inteligentes para a busca das mesmas [BRE98]. Navegadores como o Netscape, catálogos de pesquisa como o Yahoo (<http://www.yahoo.com>) e servidores de pesquisa como o Altavista (<http://www.altavista.com>) são exemplos claros desta área.

3.8.2 *NEWSWATCHER*

Nesse caso, os agentes recuperam apenas as mensagens de listas que atendem as necessidades do usuário. Alguns exemplos destes agentes são o PointCast Network (<http://www.pointcast.com>) e o After Dark OnLine (<http://www.afterdark.com>).

3.8.3 ACONSELHAMENTO

Nesta área os agentes observam passos dos usuários que utilizam o navegador, aprendem suas ações e dão conselhos e informações para trabalhos futuros [BRE98]. Conseqüentemente, o usuário do *browser* ou sistema pode concentrar-se nas tarefas importantes, deixando o trabalho repetitivo para o agente. O Web Browser Intelligence da IBM (<http://www.networking.ibm.com/wbi/wbisoft.htm>) é um exemplo.

3.8.4 ENTRETENIMENTO

A principal aplicação na área de entretenimento é dar suporte ao usuário, de acordo com suas preferências, na escolha de sua atividade de recreação. É utilizado em áreas como compras on-line, para programação de cinema, televisão e músicas. Algumas aplicações são o NetRadio (<http://www.netradio.net>) e o FireFly (<http://www.firefly.com>).

3.8.5 *SHOPPING AGENTS*

São agentes que procuram algum produto desejado em loja on-line e retornam uma lista de preços, normalmente esta lista já contém *links* para os *sites* das lojas. Alguns agentes podem oferecer também *links* para produtos similares. Alguns exemplos de *shopping agents* são: ACSES da Muenchhoff & Jans GmbH Inc. (<http://www.acses.com>); Fido the Shopping Doggie da Continuum Software (<http://www.shopfido.com>); Jango 2.0 da Excite (<http://jango.com>); O RoboShopper International (<http://www.roboshopper.com>) e o ShopFind da Viaweb (<http://shoopfind.com>).

3.8.6 CUSTOMER SERVICE AGENTS OU CHATTER BOTS

Estes agentes são aplicados em sistemas de *help desks* com o objetivo de agilizar a busca de informações, trazendo uma resposta efetiva ao problema apresentado pelo usuário. Normalmente estes agentes fazem a busca em base de dados em *cd-rom* e existem alguns que procuram informações na Internet. Alguns exemplos de *Chatter Bots* são: Alice, da Eastport Internet Associates (<http://alice.eecs.lehigh.edu:1991>); o Shallow Red da Neuromedia (<http://www.neurostudios.com/html/ShallowRedFrame.html>); Barry DeFacto da FringWare (<http://www.fringeware.com/bot/barry.html>) e o Millie (<http://millie.y2klinks.com/html/MillieFrame.htm>)

3.8.7 STOCK AGENTS

Estes agentes são similares aos *chatter bots*, entretanto os *Stock Agents* ou *Stock Bots* procuram na internet informações relacionadas à área financeira. Um exemplo típico é o de um usuário que gasta tempo verificando num *site* de aplicações financeiras se houve alguma mudança nos índices de sua aplicação ou de seu portfólio. Um *stock bot*, poderia verificar constantemente *sites* de aplicações financeiras e a medida que houver alguma alteração nos índices da aplicação ou portfólios de interesse do usuário, pode então enviar um e-mail para o mesmo. Há agentes que não alertam somente para mudanças de interesse da aplicação do usuário, mas também sobre novas aplicações rentáveis que surgem no mercado. Alguns exemplos destes agentes são: Investor Agent da AgentSoft (<http://agentsoft.com/demos/investor>); All Seeing EYE da StreetEYE (<http://streeteye.com/cgi-bin/allseeingeye.cgi>); o Stock Agent Pro da Insanely Great Software (<http://www.igsnet.com/stockagent.html>); o StockVue da AlphaCONNECT (<http://stockvue.com/download.asp>); e o FinanceWise da Financial Engineering (<http://www.financewise.com/>).

3.8.8 COMPUTAÇÃO MÓVEL

A migração dos sistemas isolados para os sistemas baseados em rede já é realidade há algum tempo. Esta evolução gerou novas necessidades para os usuários: os usuários precisam de mobilidade, acessando os recursos de uma rede a partir de qualquer lugar, independente das

limitações de largura de banda e volatilidade da rede. Os agentes inteligentes, que neste caso residem na rede e não no computador pessoal do usuário, podem atender a esses requisitos, encarregando-se das solicitações do usuário. Os agentes podem processar as informações na fonte, retornando para o usuário somente os dados importantes, evitando o congestionamento da rede com grandes quantidades de dados não processados.

3.8.9 CONTROLE DE PROCESSOS INDUSTRIAIS

Segundo [JEN1998], *“o controle de processos é uma área natural para a aplicação de agentes inteligentes e sistemas multi-agentes, já que os controladores são, por natureza, sistemas autônomos e reativos”*. O maior exemplo desta aplicação é uma plataforma denominada ARCHON, destinada à construção de sistemas multi-agentes. Agentes, desenvolvidos sobre o ARCHON, são aplicados em sistemas complexos, como transporte de energia e aceleração de partículas.

3.8.10 GERENCIAMENTO DE REDES E SISTEMAS

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

4 NEGOCIAÇÃO

4.1 CONCEITOS

A negociação no contexto do comércio eletrônico é um processo em que duas ou mais partes multilateralmente barganham recursos para um ganho mútuo, usando técnicas e ferramentas do comércio eletrônico [BEA96].

Segundo esta perspectiva, o uso de correio eletrônico entre duas ou mais pessoas afim de negociarem um determinado produto não se caracteriza um processo de negociação automática dentro do comércio eletrônico [FAR1998]. Mas se dois ou mais agentes de maneira autônoma negociarem esse mesmo produto eletronicamente, para apresentar a seus usuários o resultado da negociação, pode-se dizer que houve uma negociação automática.

Os sistemas multi-agentes atuais estão mudando o significado das negociações de várias maneiras. Enquanto uma corrente de pesquisa busca descrever os subproblemas e recursos para a negociação, outra trata da negociação direta entre dois agentes. Enquanto o protocolo de negociação provê regras que possibilitam as duas formas de negociação, sua estratégia depende da implementação do próprio agente [BRE98].

Embora o criador do agente possa prover diferentes graus de complexidade para suas capacidades de negociação, é preciso tomar cuidado para o protocolo e estratégia sejam compatíveis, ou seja, a estratégia selecionada possa ser executada com o protocolo disponível.

4.2 PROBLEMAS

Dois são os principais problemas da negociação [BEA96]: problema da negociação cooperativa e o problema da negociação competitiva.

A negociação cooperativa defende que todas as partes ou agentes estão trabalhando

para um mesmo objetivo [FAR98]. Geralmente, pode-se encontrar essa situação quando diferentes departamentos de uma organização estão negociando recursos internos. Nesse caso, uma parte pode ceder recursos para outra se houver lucro para a corporação como um todo. No entanto, existe uma falha no processo de negociação cooperativa com relação a vulnerabilidade para a estratégia de um possível oponente, ou seja, existe uma grande perda da parte que quer cooperar para a outra parte que possui uma estratégia de competição. Desta forma, se um agente tenta cooperar com um outro que somente compete, o primeiro perderá sempre pois cederá recursos para o segundo sem um possível retorno.

Na negociação não cooperativa ou competitiva, as partes assumem algum conflito de interesses. O grau de incompatibilidade nos objetivos e os métodos para lidar com os conflitos, determinam as estratégias a serem adotadas pelos agentes em diferentes circunstâncias. Esse tipo de negociação se identifica como um jogo de soma zero, ou seja, o ganho de um jogador é a perda de outro. Um exemplo é a negociação do preço de um produto entre um comprador e um vendedor, onde as duas partes possuem objetivos antagônicos: o comprador quer o menor preço e o vendedor persegue o maior [FAR98].

4.3 ABORDAGENS

As diferentes maneiras e ambientes onde os processos de negociação ocorrem é que vão definir a abordagem a ser usada. As seguintes abordagens merecem destaque [BEA1996]:

- a) Fatores Humanos:** nesse caso a negociação é feita por humanos e concentram-se em métodos, técnicas, ferramentas e estratégias para negociar satisfatoriamente. O foco principal é como gerenciar os fatores humanos como orgulho, ego e cultura dentro do processo de negociação. A negociação através do comércio eletrônico remove os fatores humanos em favor de um processo puro, ou seja, sem intervenção humana.
- b) Economia e Teoria de Jogos:** essas abordagens tratam a negociação através de uma perspectiva mais matemática. A abordagem econômica para a negociação cobre mecanismos e métodos para tratar o preço dos produtos e se concentra principalmente na determinação de preços. Já a Teoria de Jogos recomenda determinadas ações para um jogador enquanto considera a estratégia e os valores pagos pelo oponente. Existem

alguns problemas na aplicação da teoria de jogos no comércio eletrônico, quais sejam, a necessidade de perfeita informação para a tomada de decisões e a perfeita racionalidade das partes no sentido de identificar qual a ação que o oponente tomará.

c) Ciências da Computação e Agentes Inteligentes: Os sistemas multi-agentes projetados para o comércio eletrônico podem incorporar várias características, tais como regras de negócio (*vide tópico 4.4*), acesso à base de dados dos fornecedores e facilidade de encontrar fornecedores e clientes. Os agentes presentes na negociação podem ser de dois tipos: os que aprendem com a interação com outros agentes e os que não aprendem. No primeiro caso, os agentes aprendem estratégias de negociação através de ferramentas da Inteligência Artificial Tradicional, como por exemplo, redes neurais. No último caso, eles não tem a habilidade de aprender durante a negociação e dessa forma fazem uso de regras do ambiente que são delineadas pelos usuários e armazenadas em uma grande memória, onde para cada situação que ocorra no ambiente existe uma ação específica.

4.4 REGRAS DE NEGÓCIO

O processamento de regras de negócio é uma tecnologia que surgiu nos anos 80, com os então denominados sistemas especialistas, aplicações da tecnologia de Inteligência Artificial. Regras de Negócio são estruturas da forma "Se-Então-Senão" que permitem representar conhecimento, podendo ser armazenadas em bases de regras (também chamadas bases de conhecimento) da mesma forma que informações são armazenadas em registros de bases de dados. Sua utilização permite criar soluções em que o conhecimento é representado em estruturas separadas da aplicação, facilitando a manutenção e incorporação de novas regras. As regras de negócio são processadas por um algoritmo (engine) denominado motor de inferência [FAR1998].

Utilizando-se este tipo de tecnologia, experiências adquiridas pelos seus melhores gerentes podem ser traduzidas em bases de regras de negócio e processadas pelo motor de inferência, que as encadeia de acordo com o comportamento de cada usuário. As regras de negócio detectam as necessidades e comportamento de seus clientes, enviando e-mails

customizados, prevendo necessidades, auxiliam na resolução de problemas simples e encaminham automaticamente a resolução de problemas complexos.

5 COMÉRCIO ELETRÔNICO

5.1 HISTÓRIA

Quando a Internet foi criada, era exclusiva das forças armadas dos Estados Unidos e somente depois começou a ser usada por civis. Sua utilização com fins comerciais, começou a surgir quando sua área de atuação foi ampliada em produtos e fronteiras. Os bancos foram os pioneiros a usar a internet em larga escala com fins comerciais. Inicialmente, operando em redes entre si, e posteriormente ampliando a sua clientela. Logo depois vieram as lojas "virtuais", os supermercados, as indústrias, etc. Até esse momento, e principalmente enquanto ele encontrava-se praticamente circunscritos aos E.U.A., os riscos eram calculados, já que o cliente sabia com que instituto estava operando. Tratavam-se em sua maioria de típicas relações de consumo como pólos definidos, Bancos, Operadoras de Cartão de Crédito e grandes comércios [ALB1999].

Com seu direito de "*comon law*" e forte influência da securitização, muitas dessas relações estão lastreadas em contratos de seguro e responsabilidade aquilina. A partir daí ocorreu, o que convencionou-se chamar de – a terceira onda – desse comércio, que é sua popularização e disseminação. Essa fase que se encontra em franca expansão, e que é a maior delas, caracteriza-se pelo comércio entre desconhecidos, sejam entre pessoas físicas ou jurídicas, ou ambas, e que necessitarão, em muitos casos, de contratos, típicos do comércio. Hoje como uma amostra dessas realidades tem-se, por exemplo, a possibilidade de alugar-se e até comprar um imóvel numa cidade distante, comprar um projeto arquitetônico, um software, uma música, etc.

5.2 CONCEITOS

Segundo [ALB99], o comércio eletrônico (CE) é a finalização de toda a cadeia de valor dos processos de negócio num ambiente eletrônico, por meio da aplicação intensa de tecnologias de comunicação e informação, atendendo aos objetivos de negócio. Os processos

podem ser realizados de forma completa ou parcial, incluindo as transações negócio-a-negócio (*bussiness to bussiness*), negócio-a-consumidor (*bussiness to consumer*) e intra-organizacional, numa estrutura predominantemente pública de fácil e livre acesso e baixo custo. Já [CAM97] define que CE inclui qualquer negócio transacionado eletronicamente, em que essas transações ocorrem entre dois parceiros de negócio ou entre um negócio e seus clientes.

Dependendo para quem se pergunte, CE pode ter definições diferentes:

- a) De uma perspectiva de *comunicações*, o CE é a entrega de informações, produtos/serviços, ou pagamentos por meio de linhas de telefone, redes de computadores ou qualquer outro meio eletrônico;
- b) De uma perspectiva de *processo de negócio*, o CE é a aplicação de tecnologias e para a automação de transações de negócios e fluxos de dados;
- c) De uma perspectiva de *serviços*, o CE é uma ferramenta que endereça o desejo das empresas, consumidores e gerência para cortar custos de serviços, enquanto melhora a qualidade das mercadorias e aumenta a velocidade da entrega do serviço;
- d) De uma perspectiva *on-line*, o CE provê a capacidade de comprar e vender produtos e informações na Internet e outros serviços on-line.

Resumindo, o CE pode ser definido como a compra e a venda de informações, produtos e serviços por meio de redes de computadores.

5.3 VANTAGENS/DESVANTAGENS

De acordo com [Jean Paul JACOB 1998 *apus* AMO1999], através da Internet a relação entre consumidor e empresa com certeza será modificada, mas as lojas físicas como existem hoje continuarão a existir. O que a informática fará é "estreitar o contato entre empresas e consumidores". Realmente, hoje a Internet esta sendo mais usada para contatos entre empresas e fornecedores, mas o autor afirma ainda que no futuro "uma empresa poderá buscar informações detalhadas sobre quem são os seus clientes e mandar o material promocional exclusivo para os interessados em seus produtos".

Na visão de [LOPES 1998 *apud* AMO1998], a Internet oferece vantagens e benefícios que residem na própria maneira como a Internet foi idealizada e na maneira como é utilizada:

- a) **Conveniência** - O usuário pode acessar de qualquer lugar que tenha um micro conectado, as informações que deseja e realizar operações. Um exemplo é o da empresa de cosméticos Natura (<http://www.natura.com.br>), onde as pessoas que distribuem os produtos da empresa podem encomendar cosméticos usando a Internet, ao invés dos meios tradicionais como fax e telefone;
- b) **Variedade** - As empresas podem oferecer uma grande variedade de produtos, já que não precisam manter um estoque físico. Bons exemplos são a livraria Amazon (<http://www.amazon.com>) com uma disposição de três milhões de livros aos consumidores e a loja virtual de discos CDNOW (<http://www.cdnow.com>) com uma disponibilidade de 250.000 Cd's;
- c) **Fidelidade** - Através da criação de "consumidores virtuais", as pessoas podem se conectar a um ambiente virtual onde o usuário tem acesso a informações e podem trocar experiências entre si. Esta é uma das melhores maneiras para se criar fidelidade por parte dos consumidores, já que a venda passa, teoricamente a um segundo plano;
- d) **Custo** - O custo de manter uma presença virtual é mais barato se comparado a uma presença real, ou seja, a uma presença tradicional. Um estabelecimento físico necessita de um local, a construção, a mão-de-obra, luz, água, etc. Na Internet estes custos são bastante reduzidos, ou mesmo não existem;
- e) **Preço final** - O preço final do produto teoricamente deveria ser menor se comparado a um estabelecimento físico. Porém, pesquisas apontam que só em 12% das lojas virtuais isto se verifica. Tal fato se deve, porque muitos fabricantes temem deteriorar seu relacionamento com seus fornecedores;
- f) **Serviço ao consumidor** - Por meio da Internet, pode-se prestar uma excelente assistência ao consumidor. No *site* da Natura, por exemplo, a distribuidora pode obter as mais diversas informações como: pedidos anteriores, produtos disponíveis e características, limites e lançamentos. Tudo isto com respostas imediatas;

g) **Fim do atravessador** - A Internet abre espaço a desintermediação. Um bom exemplo é o das agências de turismo. Estas comercializam "algo que o comprador não precisa ver, tocar ou experimentar antes de comprar". E, justamente por isso corre perigo, porque o caminho entre produtor e consumidor é incrivelmente facilitado pela Internet. De acordo com Nicholas NEGROPONTE (1997), pessoas que hoje trabalham como intermediárias, descobrirão que vão se tornar supérfluas. A *interface* dos sistemas de reserva de passagem tornou-se quase utilizável, dando margens para que os compradores e vendedores de passagens aéreas possam chegar a retirar os intermediários. O que sobra para as agências de turismo, se o usuário pode sozinho escolher o destino, fazer comparações de trajetos e preços e comprar um bilhete aéreo via Internet? Só em 1997, uma agência de 25 milhões de dólares em San Francisco, estima que perdeu de 10% a 15% das suas vendas para a Internet (HOF, Robert D., MCWILLIAMS, Gary, SAVERI, Gabrielle, 1998). Da mesma forma que o atendimento bancário automatizado, as máquinas ATM possibilitam ao usuário um atendimento muito melhor, fazendo com que não seja necessário se deslocar até as agências bancárias dentro dos horários rigidamente estabelecidos. Só em 1997 os bancos brasileiros investiram 2 bilhões em tecnologia da informação (CAVALCANTI, 1997). A saída para as agências de viagens esta em oferecer algo mais através da Internet como uma "espécie de consultoria de turismo". Isto significa "personalização", a força única mais poderosa do comércio no mundo digital.

Na verdade, várias formas de Comércio Eletrônico vêm sendo utilizadas ao longo dos anos: O EDI e as redes privadas de serviços financeiros (SWIFT, por exemplo) são alguns exemplos. Os Sistemas de Comércio Eletrônico atualmente, no entanto, apresentam algumas limitações que necessitam ser endereçadas:

a) **Soluções Parciais**: Os atuais sistemas de Comércio Eletrônico automatizam apenas parte de um processo. Por exemplo, apesar do pedido e distribuição de um produto como software ou revista eletrônica poderem ocorrer de forma imediata e simultânea, o processo contábil, transferência efetiva de fundos tendem a durar alguns dias. Uma

solução de Comércio Eletrônico totalmente integrada facilita o controle sobre os fluxos de caixa, estocagem, etc.

- b) **Requerimentos Rígidos:** A primeira geração de Sistemas de Comércio Eletrônico utilizava linhas de comunicação dedicadas e protocolos altamente estruturados. Para se operar em EDI, por exemplo, definições rígidas têm de ser estabelecidos entre as partes com relação à estrutura e significado dos dados transmitidos. Os tempos envolvidos e custos resultantes em função desta rigidez criam barreiras para a disseminação dos sistemas de Comércio Eletrônico.
- c) **Acesso Limitado:** Nos atuais sistemas de Comércio Eletrônico, o usuário não pode se comunicar e interagir com vendedores de uma forma direta e livre. Os Agentes Inteligentes e outros mecanismos vão aproximar cada vez mais os compradores dos *sites* de Comércio Eletrônico.
- d) **Interoperabilidade Limitada:** Uma infraestrutura de Comércio Eletrônico interoperante facilitaria transações privadas, reduzindo a necessidade de intermediários a menos que eles sejam provedores de algum tipo de valor agregado, como financiamentos. Esta infraestrutura possibilita o aumento da oferta de serviços e de consumidores.
- e) **Segurança Insuficiente:** Apesar de todos os recursos existentes voltados para criptografia, autenticação e certificação, ainda não existe efetivamente uma "Moeda da Internet" adotada por todos. Meios de pagamento como cheques, são usados com limitação nas compras on-line. A segurança em transações eletrônicas é realmente um dos aspectos de maior preocupação de todos os envolvidos com o Comércio Eletrônico.

Muitos dos novos serviços de Comércio Eletrônico consistem nas fundações necessárias para que se possam atingir níveis de segurança e credibilidade, através da incorporação de características como:

- a) Autenticação em redes públicas;

- b) Certificação de informações e transações;
- c) Seguros de transações on-line;
- d) Serviços de Corretagem.

À medida que várias iniciativas vão se desenvolvendo, *standards* vão sendo sedimentados, os recursos vão se tornando cada vez mais sofisticados, e a nova forma de comércio vai sendo cada vez mais aceita pela população.

5.4 TENDÊNCIAS

Segundo [GIE1998], a abertura da economia juntamente com o desenvolvimento da *Internet* e outras tecnologias, fizeram com que o número de empresas concorrentes aumentasse significativamente, provocando a queda dos preços e o aumento da qualidade dos produtos e serviços oferecidos. De acordo com Stüpp Schneider, presidente da Confederação Nacional dos Dirigentes Lojistas (CNDL), o futuro pertence as grandes redes e grandes marcas, e às lojas pequenas que tenham como diferencial o preço e o atendimento e ao mercado de vendas eletrônicas. Stüpp salienta ainda que as novas oportunidades de mercado apontam para uma tendência de crescimento nas vendas não tradicionais (como *Internet*, *TV shop* e malas diretas). Conforme Adkisson Tapscott, gerente de operação e desenvolvimento da Nordstrom, o segredo para o sucesso no futuro é “estar onde os clientes querem estar” [ALB1999].

Estima-se que a receita gerada no mundo devida ao comércio eletrônico dê um salto de 8 bilhões de dólares em 1996 para 327 bilhões de dólares em 2002 [ALB1999]. Com este crescimento, as soluções de Comércio Eletrônico vêm se modernizando e incorporando recursos sofisticados, tornando cada vez mais eficiente e lucrativo o relacionamento comercial entre clientes e fornecedores através da *Internet*. Atualmente, vários paradigmas vêm sendo revistos e modificados nesta área. Alguns dos benefícios que vêm sendo obtidos em função da utilização das novas soluções são os seguintes:

- a) ***One-to-One Marketing***: Os Sistemas de Comércio Eletrônico passam a incorporar Regras de Negócio voltadas para a determinação do perfil dos clientes, promoções e

produtos complementares. Através das técnicas de *One-to-One* Marketing pode-se personalizar totalmente as sessões de consulta de clientes a *sites* de Comércio Eletrônico, maximizando as possibilidades de venda e oferecendo um tratamento totalmente personalizado;

- b) **Produção “*Build to order*” e “*Mass Customization*”**: Os produtos oferecidos em *sites* de Comércio Eletrônico passam a ser produzidos seguindo exatamente a especificação do cliente. Através da utilização de Regras de Negócio voltadas para a configuração de produtos, os sistemas podem guiar o usuário durante todo o processo de configuração, possibilitando a criação de produtos totalmente personalizados;
- c) ***Customer Care***: “Cuidar” bem do cliente, antecipando-se com relação a suas necessidades também é um dos desafios dos Sistemas de Comércio Eletrônico. Regras de Negócio que automatizam a condução do relacionamento com o cliente através da emissão inteligente de e-mails;
- d) **Integração da Cadeia de Fornecimento (*Supply Chain Management*)**: A integração entre os elementos de toda a cadeia de fornecimento (cliente, *sites* de *e-Commerce*, fornecedores, terceiros) passa a ser muito maior em relação aos métodos tradicionais. Sistemas que integram toda a cadeia de fornecimento consistem em uma nova filosofia de negócios;
- e) **Fabricação “*Just-in-time*”**: Os produtos passam a poder ser produzidos somente em função de pedidos específicos. Com isso consegue-se uma diminuição nos níveis de estoque. Em indústrias, a tendência é que Sistemas de Comércio Eletrônico sejam conectados a softwares de ERP para que se possa viabilizar a fabricação just-in-time em função de pedidos oriundos da Internet;
- f) **Agentes Inteligentes e Regras de Negócio**: As regras de negócio, contidas em Sistemas de Comércio Eletrônico, devem poder ser rapidamente modificadas, uma vez que novos produtos e promoções sempre estarão sendo dinamicamente incluídos. Para tal, é extremamente desejável que as regras de negócio estejam separadas da lógica da aplicação. Dessa forma, a manutenção e atualização das mesmas podem ser facilmente realizadas. As Regras de Negócio são encapsuladas em agentes inteligentes

programados em Java, responsáveis por atividades como determinar o perfil do cliente, auxiliar na configuração de produtos, sugerir promoções e produtos complementares, etc;

- g) **Servidores *Web* com Conteúdo Dinâmico:** Em função do grande volume de informação, que deve ser constantemente atualizado, um dos requisitos dos Sistemas de Comércio Eletrônico é que o conteúdo das páginas HTML seja modificado dinamicamente. Dentre os Servidores *Web* que suportam a construção de páginas com conteúdo dinâmico destacam-se ferramentas como Domino Merchant, da Lotus, utilizado em projetos de Sistemas de Comércio Eletrônico. Pode-se também utilizar aplicações Java Server-Side para criar *sites* de conteúdo dinâmico;
- h) **Sistemas de *Middleware* e Tecnologia CORBA:** Para tornar mais simples a integração entre os sistemas já existentes na Organização residentes em diferentes plataformas, softwares de *middleware* que funcionam com o conceito de subscrição / publicação disparadas por eventos, são extremamente úteis na implementação dos novos Sistemas de Comércio Eletrônico. Suporte à tecnologia CORBA para integração entre as plataformas é umas tendências que se concretiza no que se refere à integração entre os ambientes;
- i) **Modelo “*Thin client*”:** Uma vez que existe a tendência de que os Sistemas de Comércio Eletrônico possam ser também acessados a partir de quiosques, é interessante utilizar uma arquitetura baseada no conceito de “*thin-client*”, as páginas dos *sites* sejam carregadas no *browsers* do usuário;
- j) **Arquitetura dos Novos Sistemas de Comércio Eletrônico:** A nova geração de Sistemas de Comércio Eletrônico deve ser dotada de uma arquitetura modular que possibilite a implementação de soluções escaláveis, que possam começar com um tamanho reduzido e ser gradativamente expandidas. Os novos sistemas devem contemplar aspectos como:
 - Seleção dinâmica de itens;
 - Determinação de requisitos através de perguntas;
 - Associação da necessidade do cliente ao item que melhor se aplica;

- Recomendação automática de itens;
- Controle da Oferta de Itens;
- Priorização da Oferta de itens;
- Sugestão de itens alternativos caso o item desejado pelo cliente não se encontre disponível;
- Configuração automatizada de produtos e serviços;
- Seleção dinâmica de opções com base em regras de configuração;
- Integração a aplicações de Serviço a Clientes para identificar problemas relacionados à configuração;
- Determinação automatizada de preços e descontos através de regras de negócios;
- Estabelecimento de preços e políticas de descontos através de uma base integrada de políticas;
- Possibilidade de customizar programas de descontos de forma única para cada usuário;
- Integração com Sistemas existentes voltados para o input de pedidos.

Além disso, a nova geração de Sistemas de Comércio Eletrônico deve ser dotada de uma arquitetura modular que possibilite a implementação de soluções escaláveis, que possam começar com um tamanho reduzido e serem gradativamente expandidas. Neste ambiente fica caracterizada a possibilidade de implementação de algumas funções que podem agregar muito valor às Soluções de *E-Commerce*:

- a) Determinação automatizada de preços e descontos através de regras de negócios;
- b) Possibilidade de customização de programas de descontos de forma única para cada

usuário;

- c) Integração com sistemas transacionais existentes;
- d) Determinação de requisitos através de perguntas estratégicas;
- e) Associação da necessidade do cliente aos produtos que melhor se aplicam;
- f) Recomendação automática de itens;
- g) Priorização da Oferta de itens;
- h) Sugestão de itens alternativos caso o item desejado pelo cliente não se encontre disponível;
- i) Configuração automatizada de produtos e serviços;
- j) Seleção dinâmica de opções com base em regras de configuração.

Conforme [BLU2000] o número de usuários da *Internet* e os números referentes ao comércio eletrônico crescem rapidamente.

Ano	PCs (milhões)	Usuários de Internet (milhões)	Assinaturas de Internet (milhões)
1997	4,05	1,19	0,51
1998	4,92	2,74	1,405
1999	5,73	3,83	2,027
2000	6,65	4,99	3,042
2001	7,72	6,52	4,191
2002	8,98	7,79	5,346
2003	10,1	9,03	6,412

Tabela 2: Número de usuários da Internet/Brasil (Fonte: <http://www.intermanagers.com.br/>)

Ano	Compradores através de comércio eletrônico (milhares)	Gasto anual por comprador (milhares)
1997	87	0,398
1998	337	0,531
1999	537	0,462
2000	841	0,674
2001	1286	0,873
2002	1837	1,127
2003	2415	1,43

Tabela 3: Comércio Eletrônico no Brasil (Fonte: <http://www.intermanagers.com.br/>)

5.5 CATEGORIAS

Temos basicamente quatro tipos de comércio eletrônico:

- a) Negócio-Negócio;
- b) Negócio-Consumidor;
- c) Negócio-Administração Pública;
- d) Consumidor-Administração Pública.

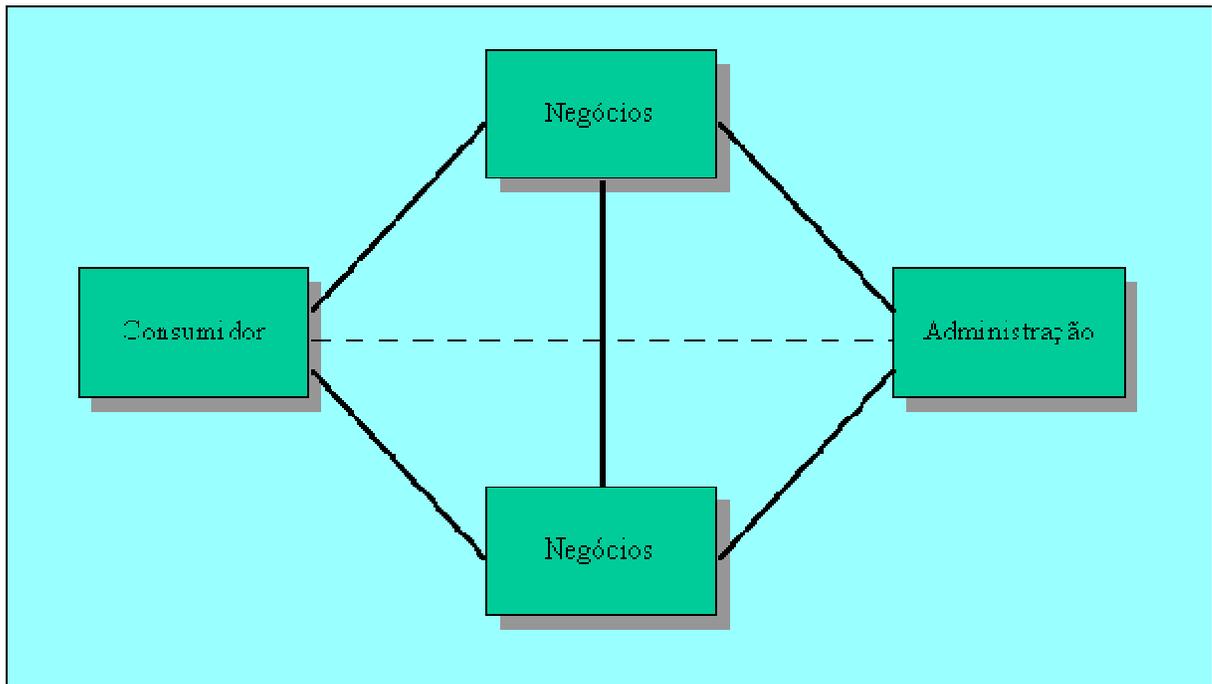


Figura 11: **Tipos de modelos de comércio eletrônico**

Um exemplo na categoria negócio-negócio seria uma companhia que usa uma rede para solicitar aos seus fornecedores, receber pedidos e fazer pagamentos. Esta categoria de comércio eletrônico já tem sido feito há muitos anos, notadamente pelo uso de EDI sobre uma rede privada ou não.

A categoria negócio-consumidor equivale em grande parte ao varejo eletrônico. Esta categoria tem tido um crescimento enorme com o advento da WWW. Existem shoppings eletrônicos por toda internet oferecendo de tudo, desde bolos e vinhos a computadores e carros.

A categoria negócio-administração pública cobre todas as transações entre companhias e organizações governamentais. Esta categoria está na infância mas pode expandir-se rapidamente à medida que os governos usarem suas próprias operações para despertar a atenção e o crescimento do comércio eletrônico. Nos Estados Unidos já é possível cadastrar-se como fornecedor de certos produtos ao governo via internet e todo o processo de compras também é eletrônico.

A categoria consumidor-administração pública também está em crescimento. No

Brasil, tem-se exemplos muito significativos: a receita federal, Detran, Ministério do Trabalho, correios, entre outros. Sempre no sentido de atender ao cidadão comum. Entretanto, pouco, ou quase nada, no sentido de atender aos fornecedores.

5.6 AGENTES APLICADOS EM COMÉRCIO ELETRÔNICO

5.6.1 INTRODUÇÃO

Apesar de seu grande potencial para o comércio, entretenimento e a aprendizagem, os internautas estão limitados em vários aspectos, isto porque grande parte da Web é estritamente um ambiente *self-service*, caracterizando-se por uma intervenção somente mecânica do usuário. Máquinas “sem sentimentos”, com métodos e interações rígidas podem frustrar as pessoas e desmotivar um grande número de transações.

Os visitantes de hoje de um *site* de comércio eletrônico são clientes em potencial. Eles vieram procurar informação, comprar bens ou participar de atividades em grupo. Muitas empresas já procuram restabelecer o contato tradicional com o cliente colocando atendentes para ajudar o usuário por telefone ou conexões Internet usando programas de “bate-papo”. Embora essa aproximação seja importantíssima, é cara, muito trabalhosa e, geralmente, não satisfaz completamente o cliente. Segundo [HER1996], “*o comércio eletrônico é uma área em expansão, impulsionada pela popularidade da Internet*”. Os compradores precisam encontrar os fornecedores dos produtos e serviços que necessitam, precisam encontrar informações sobre produtos que atendam aos seus requisitos, e, após a compra, freqüentemente necessitam de algum tipo de suporte. Os fornecedores por sua vez, necessitam de meios para alcançar o maior número possível de compradores com seus produtos. Agentes podem ser aplicados para auxiliar tanto compradores quanto fornecedores, realizando compras para o usuário, buscando informações técnicas sobre produtos, promovendo a venda de produtos, solucionando problemas pós-venda para os clientes, negociando o melhor preço e forma de pagamento, etc.

A alternativa, neste caso, é criar-se agentes inteligentes para interagir com o usuário e consumidores, utilizando mensagens, sons, diálogos e animação. Como o sistema (agentes), fica disponível durante 24 horas, cria-se um ambiente ilimitado, personalizado, prático e

agradável para o visitante.

Neste contexto, personagens interativos representam uma classe bastante útil de agentes inteligentes. Como outros agentes, eles tem programas inteligentes que assimilam informações, realizam operações internas e cognitivas e controlam ações executadas no ambiente. Operam com autonomia tomando suas próprias decisões em tempo real. Os personagens precisam de recursos gráficos, áudio, texto e outras mídias para representar a variedade lingüística e comportamentos gestuais. Possuem ainda, personalidade, comportamento social e conhecimento de fatos diários, o que os tornam agradáveis, como um atendente humano.

No Brasil, essa área de pesquisa ainda foi pouco explorada, mas em empresas do exterior, como a Extempo (<http://www.extempo.com/>) essa área já faz parte de seus negócios. Um exemplo de agente criado pela empresa é o Papai Noel, que é um agente adaptado para falar com crianças e adulto, interagindo de acordo com as perspectivas dos visitantes.

5.6.2 AGENTES E NEGOCIAÇÃO

Conforme [MAE1998], é durante o estágio de negociação que preços e outros termos da transação são definidos. No varejo, os clientes estão acostumados com preços fixos, já que este é o processo usado há muito tempo. O benefício da negociação dinâmica é que em vez de o proprietário pôr um preço fixo, é possível propor um preço que seja o mais adequado para o momento. Em muitos tipos de negociação que acontecem no mundo físico a presença física geralmente é obrigatória, como em leilões domiciliares. Esses problemas desaparecem no mundo digital. Por exemplo, OnSale (<http://www.onsale.com/>) e eBay's AuctionWeb (<http://www.ebay.com/aw>) são dois populares sites de venda de produtos de “segunda mão” não necessitam da presença física dos participantes para a negociação se realizar. Por outro lado, necessitam que os clientes criem suas estratégias de negociação. É neste ponto que a tecnologia dos agentes entra em ação. A seguir, cada uma dessas aplicações é apresentada brevemente.

5.6.2.1 Ebay's Auctionbot

Este é um servidor de leilões para a Internet de propósito geral feito na Universidade de Michigan. O usuário do AuctionBot pode criar novos leilões escolhendo padrões a partir de

uma lista de leilões pré-definidos e então especifica alguns parâmetros como (tempo de duração, número máximo de participantes, etc). A partir deste ponto, vendedores e compradores apresentam ofertas até chegarem ao fechamento da compra. O que difere o AuctionBot dos outros leilões é que ele possui uma interface de programação (API) para o vendedor criar seus próprios agentes de negociação.

5.6.2.2 Mit Media Lab's Kasbah

Este é um sistema de transação consumidor-consumidor que utiliza vários agentes para o processo. O usuário que deseja vender ou comprar algum produto cria um agente, ensina a este algumas estratégias e o envia a uma feira de agentes. Este agente procura por possíveis compradores/vendedores para/do o produto e negocia com estes. Nessa negociação ele pergunta preços iniciais e maiores, data de pagamento e possíveis restrições a compra.

Este sistema incorpora um mecanismo de confiança e reputação chamado “*Better Bussiness Bureau*”. Assim que a transação é completada, ambas as partes criam uma taxa de o quanto outras partes envolvidas na negociação são compatíveis (semelhança nas condições do produto, transações completas e incompletas, etc). O sistema usa essas taxas para determinar com quais agentes ele deverá negociar na próxima transação efetuada.

6 PROTÓTIPO

6.1 PROJETO

Para ilustrar a aplicabilidade de agentes inteligentes no processo de negociação de um sistema de comércio eletrônico foi implementado um protótipo de site de vendas de computadores. O site foi implantado como sendo o módulo de comércio eletrônico do sistema CONSULTE.

6.1.1 CONSULTE

De acordo com [MOD2000], O Sistema Consulte possui as funções de um sistema de informação tradicional, agregadas a ferramentas para o controle automatizado da situação financeira da empresa. Ou seja, possui propriedades de um sistema de informação, caracterizado pela entrada de dados, seu processamento e saída de dados, complementados por módulos de análise automática (onde estão empregadas as técnicas de inteligência artificial). Na figura 12 é apresentada a visualização do sistema implementado.

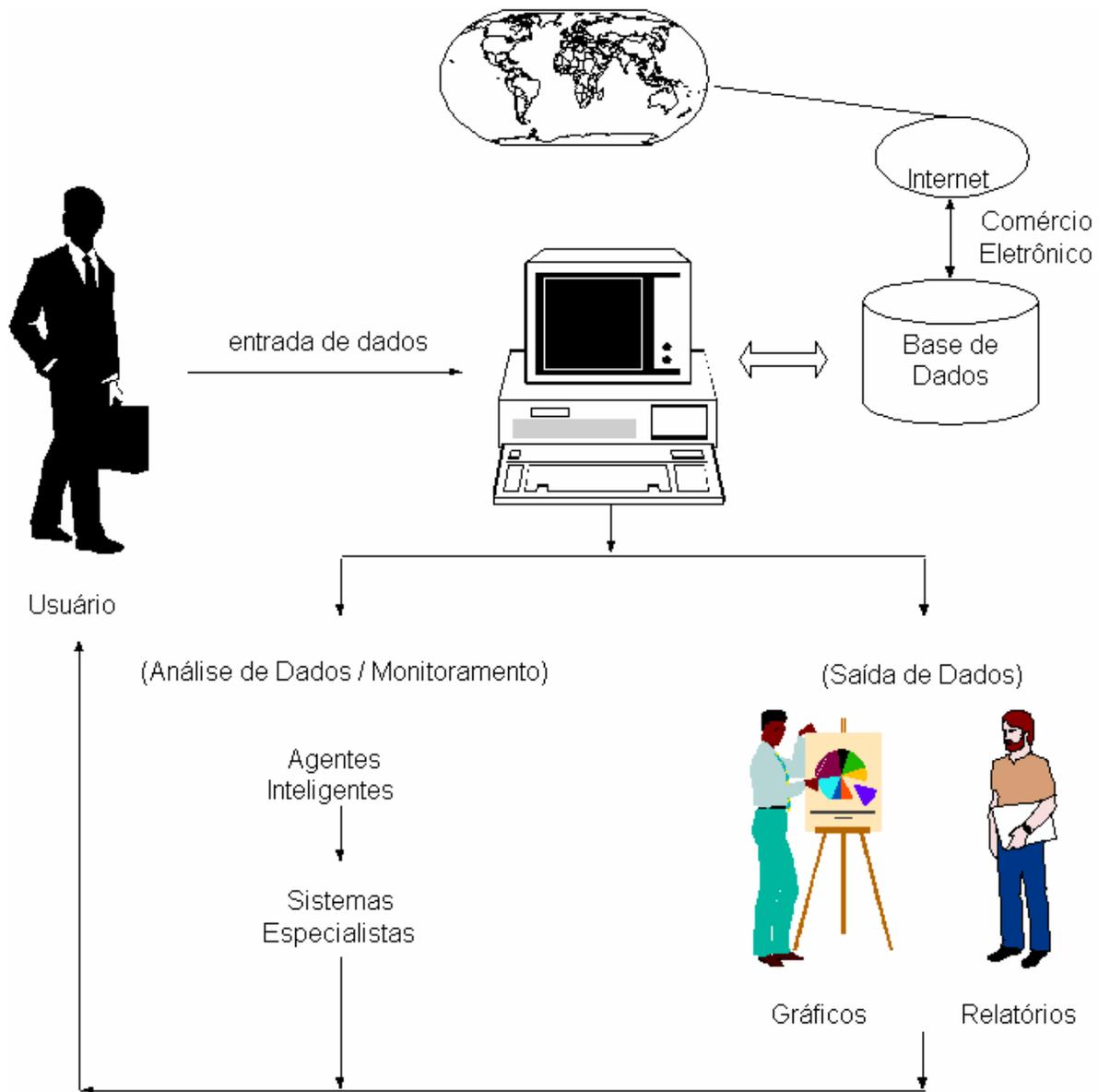


Figura 12: Visualização do sistema CONSULTE

Como saídas do sistema Consulte, citam-se os relatórios de: monitoramento, fluxo e orçamento de caixa, contas a pagar e contas a receber. Referente a monitoramento automático, é possível destacar o monitoramento de metas, de orçamento e fluxo de caixa, bem como a verificação das contas a pagar. Cabe ressaltar que a verificação de contas a pagar é realizada toda vez que o usuário acessa o sistema Consulte.

6.1.2 PROCESSO

O protótipo implementa as propriedades básicas de um *site* de comércio eletrônico como, pesquisa por produtos, cesta de compras, pagamento, entre outros. Os agentes inteligentes realizam o processo de negociação de preço entre o cliente que fez as compras e o proprietário da loja virtual, que pré-define regras para esta negociação, conforme figura 13.

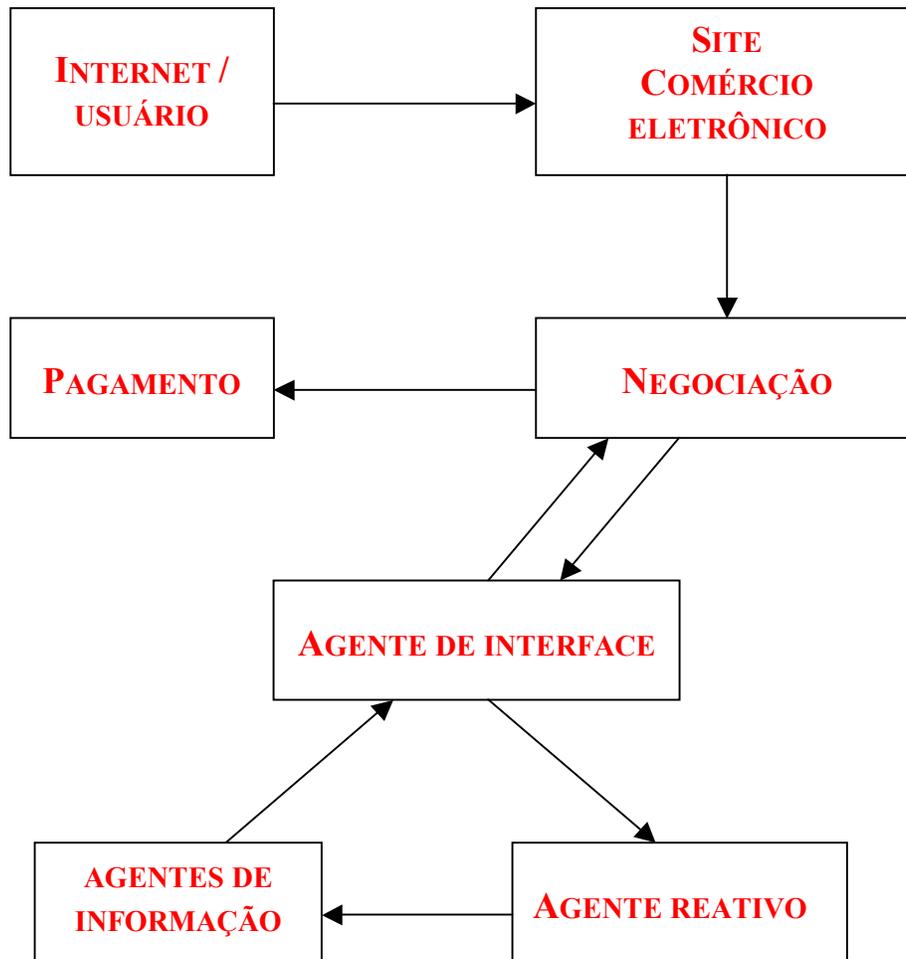


Figura 13: Funcionamento básico do protótipo

Os agentes avaliam três pontos no momento da negociação com o cliente: tabela de negociação, preço de concorrente e fluxo de caixa da empresa:

Tabela de negociação: Foi-se criado uma tabela de negociação que pré-define regras a serem seguidas pelos agentes no processo de negociação. Esta tabela foi criada a baseando-se em pesquisas realizadas em várias empresas da região, observando as regras (estratégias) de negociação aplicadas. Um modelo pode ser visto na tabela 4, onde se é definido um percentual inicia e final de descontos pedidos pelo cliente e a forma de pagamento necessária para esse percentual pedido.

Percentual Inicial (%)	Percentual Final (%)	Forma de Pagamento (mínima)
1	5	Qualquer forma de pagamento
5	15	60 dias
15	35	30 dias
35	50	À vista
50	...	Desconto não aceito

Tabela 4: Forma de pagamento necessária em relação ao desconto pedido

Tabelas semelhantes a esta são usadas com frequência por um grande número de empresas no momento de vender algum produto ou serviço. O cadastro das regras de negociação é feito em uma *homepage* de cadastro, conforme [figura 14].

The screenshot shows a Netscape browser window titled "TCC - Tela de cadastro de item de negociação - Netscape". The address bar shows the file path "file:///C:/Meus documentos/cadNegociacao.html". The main content area displays the form "Cadastro de item de tabela de negociação" with the following fields and controls:

- Percentual inicial: (%)
- Percentual final: (%)
- Forma de Pagamento: (dropdown menu)
- Buttons: "Finaliza cadastro" and "Limpar campos"

Figura 14: Cadastro de itens de negociação

Preço do concorrente: Outro ponto que vários negociadores avaliam na hora de vender algum produto ou serviço é o preço dos concorrentes. Não é difícil ouvir supermercados, lojas e outros estabelecimentos fazer grande marketing onde dizem que *cobrem* qualquer proposta de concorrentes. Para isto foi-se criado uma um campo na tabela de produtos que diz qual é o preço do concorrente para o produto que está sendo oferecido, conforme tabela 4.

Descrição do Produto	Preço da Loja	Preço do Concorrente
Notebook	3000,00	3500,00
Scanner	400,00	400,00
Processador	200,00	150,00
Produto n...	Valor loja n...	Valor concorrente n...

Tabela 5: Preço do concorrente do(s) produto(s) que estão sendo negociados

Assim como no cadastro de itens de negociação, o cadastro da tabela de produtos é feito a partir de uma *homepage* de cadastro, conforme [figura 15].

The image shows a Netscape browser window with the following elements:

- Browser Title:** Netscape
- Menu Bar:** File, Edit, View, Go, Communicator, Help
- Navigation Bar:** Back, Forward, Reload, Home, Search, Guide, Print, Security, Stop
- Address Bar:** Location: http://localhost/cadProdutos.htm
- Toolbar:** Instant Message, WebMail, Contact, People, Yellow Pages, Download, Channels
- Page Title:** Cadastro de Produtos
- Form Fields:**
 - Código:
 - Descrição:
 - Preço (à vista):
 - Preço (Concorrente):
- Buttons:** Finaliza cadastro, Limpar campos
- Status Bar:** Document: Done

Figura 15: Cadastro de produtos

Fluxo de Caixa da Empresa: Denomina-se Fluxo de Caixa de uma empresa ao conjunto de ingressos¹ e desembolsos² de numerário ao longo de um período determinado. O Fluxo de Caixa consiste na representação dinâmica da situação financeira de uma empresa, considerando todas as fontes de recursos e todas as aplicações em itens do ativo (ZDANOWICZ 1998 *apud* JUL1999). Já (FERREIRA 1998 *apud* MOD2000), diz que o Fluxo de Caixa é uma das mais importantes ferramentas de análise da circulação do dinheiro na empresa.

O principal objetivo do fluxo de caixa, a possibilidade de visualização das atividades desenvolvidas pela empresa, bem como as operações financeiras realizadas diariamente (ZDANOWICZ 1998 *apud* JOR1999). Além desse, pode-se determinar outros objetivos do fluxo de caixa, tais como:

- a) Permitir o planejamento dos desembolsos de acordo com as disponibilidades de caixa;
- b) Avaliar quanto de recursos próprios a empresa possui em um determinado período, permitindo determinar antecipadamente alternativas de investimento, ou linhas de crédito a serem obtidas junto às instituições financeiras;
- c) Utilizar racionalmente e eficientemente o disponível;
- d) Financiar necessidades sazonais da empresa;
- e) Providenciar recursos para projetos de expansão e modernização da empresa;
- f) Fixar o nível de caixa, com relação ao capital de giro.

No caso do protótipo desenvolvido, os agentes avaliam o saldo do fluxo de caixa na hora de realizar uma proposta para o cliente do *site* de comércio eletrônico. O cadastro é realizado na *homepage* de cadastro, conforme figura 16:

¹ Compõem-se de todas as entradas de caixa (vendas à vista, vendas a prazo, recebimentos normais, recebimentos com atraso, entre outros) a serem transportadas ao fluxo de caixa.

² São todas as saídas de caixa (compras à vista, compras a prazo, despesas operacionais, entre outras) a serem transportadas para o fluxo de caixa.

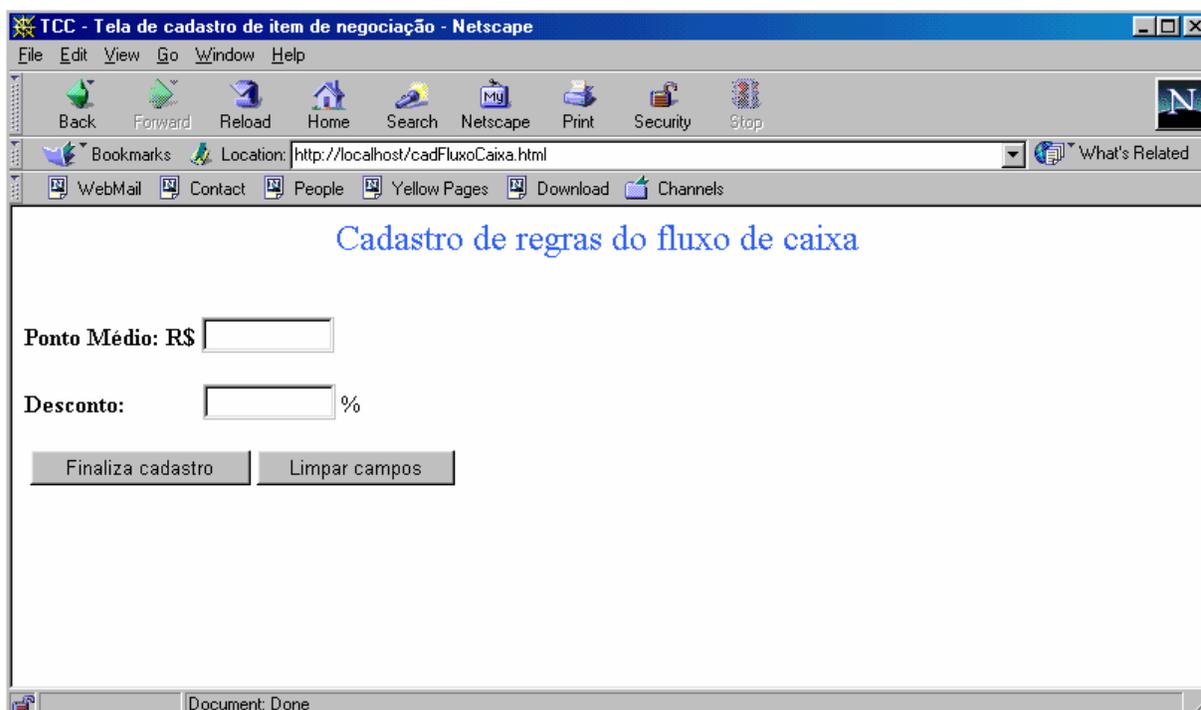


Figura 16: Tela de cadastro de regras referente ao fluxo de caixa

As regras cadastradas são o Ponto Médio e o percentual de desconto. O ponto médio correspondente ao valor que é o limite entre o saldo bom e ruim. Por exemplo, se o ponto médio cadastrado for de R\$ 1.000,00 e o fluxo de caixa apresentar um saldo de 500,00 a situação financeira da empresa não é boa. Os agentes avaliam a situação da forma descrita acima.

A lógica aplicada é a seguinte: se o fluxo de caixa estiver abaixo do ponto médio, geralmente é necessário de entrada de dinheiro com urgência para normalizar o fluxo. Caso, no processo de negociação, o cliente esteja tentando realizar uma compra a prazo, os agentes propõem descontos maiores para tentar realizar a compra à vista. Caso o fluxo esteja acima do ponto médio a negociação ocorre normalmente. O valor do fluxo de caixa será atualizado diariamente na base de dados do *site* de comércio eletrônico por agentes que atuam no sistema CONSULTE.

6.1.3 TECNOLOGIA UTILIZADA

A tecnologia utilizada para o desenvolvimento do site de comércio eletrônico foi

servlets. A tecnologia de *servlets* utiliza-se da linguagem de programação *Java* na parte do servidor e de *HTML* para o retorno dos dados na parte cliente. Esta tecnologia foi a escolhida por ter apresentado melhor portabilidade em relação a outras tecnologias como *CGI* ou *Java* na parte cliente e servidor, além de ser uma tendência apontada por algumas empresas que trabalham no ramo de comércio eletrônico atualmente. O banco de dados (BD) utilizado para o cadastro de produtos, clientes e outras tabelas relativas ao *site* foi o MySQL (<http://www.mysql.com/>). Este BD foi o escolhido porque apresenta desempenho significativo com até 300000 registros e sua utilização é gratuita, além de possui os *drivers* necessários para o acesso utilizando a linguagem *Java*.

6.1.4 AGENTES APLICADOS AO SISTEMA

Visto que o objetivo principal deste trabalho é a utilização de agentes especificamente no processo de negociação dentro do comércio eletrônico, foi-se definido seis (6) agentes para realizar o processo:

- a) **Agente de cálculo do preço inicial** (*agCalculoPreco*): Este agente faz o cálculo do preço inicial que o vendedor (empresa responsável pelo *site* de vendas) vai fazer ao cliente (consumidor visitando o *site*). Este agente age depois de definido os produtos a serem negociados (cesta de compras);
- b) **Agente avaliador da proposta** (*agAvaliadorProposta*): Este agente avalia a proposta feita pelo cliente e indica se a proposta pode ser aceita, pode ser negociada ou não pode ser aceita. Esta avaliação é feita levando-se em conta o preço dos concorrentes (uma propriedade consiste em sempre baixar o preço da concorrência, ou não) e verificando a porcentagem de desconto pedido em relação ao total de produtos comprados (tabela de descontos que podem ser aplicados);
- c) **Agente de busca de preço do concorrente** (*agConcorrente*): Este agente busca o preço que o concorrente faz para os produtos que estão sendo negociados;
- d) **Agente que verifica desconto pedido** (*agVerificaDesconto*): Este agente verifica se a porcentagem de desconto pedido pelo cliente pode ser aceita em relação ao total da compra realizada;
- e) **Agente de controle** (*agControle*): Funciona como um agente de interface. Monta a

tela para cliente verificar o resultado da negociação;

- f) **Agente que busca a contra proposta realizada** (*agContraProposta*): Este agente buscará a contra-proposta realizada pelo comerciante em relação a proposta feita pelo cliente;
- g) **Agente que verifica a situação do fluxo de caixa da empresa** (*agVerificaFluxoCaixa*): Este agente verifica se a situação do fluxo de caixa está alto ou baixo, para decidir se é necessário tentar adiantar o pagamento de algumas vendas.

6.1.5 METODOLOGIA DE DESENVOLVIMENTO

Segundo [BRE98], os clássicos modelos e métodos de desenvolvimento de sistemas não podem ser transferidos, sem modificações, para um sistema orientado a agentes, isto porque estes exibem várias diferenças básicas, incluindo conceitos e objetivos na metodologia. [BRE98] ainda defende que sistemas de agentes podem ser ligados a sistemas orientados a objetos. Os objetos (os agentes) consistem em atributos e métodos e comunicam-se uns com os outros invocando métodos e enviando mensagens, além de poder usar de outros conceitos clássicos da orientação a objetos, como encapsulamento de dados, agregação e herança. Já uma análise estrutura apresenta vários limites para a modelagem de sistemas de agentes.

Todos os métodos permitem que o processo de análise e design seja feito com vários tipos de modelos gráficos que visam simplificar a concepção e a estrutura do sistema. A metodologia utilizada para o desenvolvimento desse protótipo é a *Object Modeling Technique (OMT)*, que se utiliza três modelos: o modelo de objetos, o modelo dinâmico e o modelo funcional.

6.1.5.1 Modelo De Objetos

O modelo de objetos incorpora a estrutura estática de um sistema mostrando os objetos pertencentes a esse sistema, os relacionamentos entre esses objetos, os atributos e as operações que caracterizam cada classe de objetos [RUM94]. O modelo de objetos é o mais importante dos três modelos. É atribuída maior importância à construção de sistemas em torno de objetos do que em relação à funcionalidade, porque os modelos baseados em objetos correspondem mais aproximadamente ao mundo real, e são, em consequência, mais adaptáveis às

modificações.

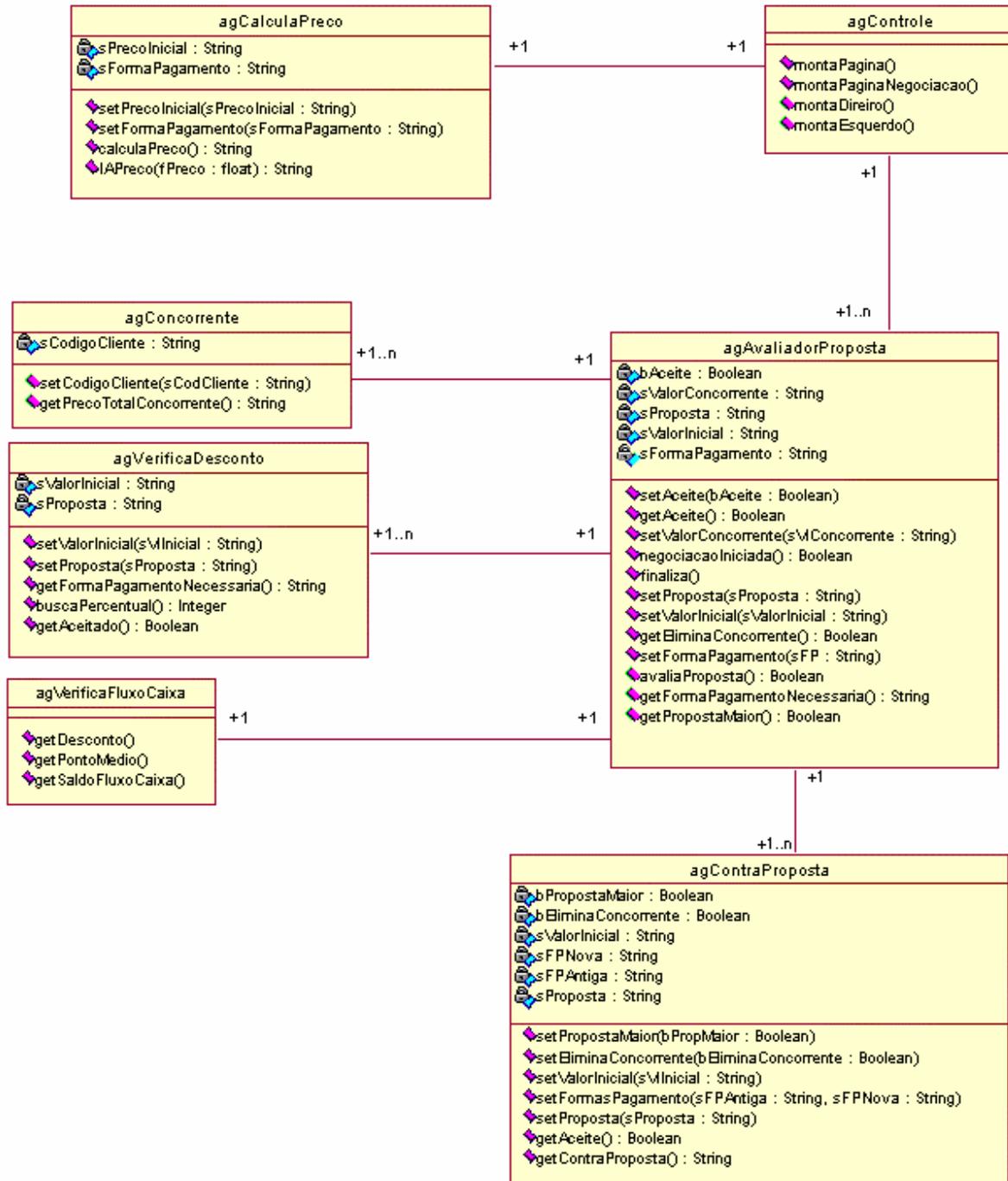


Figura 17: Modelo de objetos dos agentes

6.1.5.2 Modelo Funcional

O modelo funcional especifica os resultados de um processamento sem especificar quando ou como eles são processados. Ele especifica o significado das operações do modelo de objetos e as ações do modelo dinâmico, bem com quaisquer restrições no modelo de objetos [RUM94]. Este modelo é o terceiro ponto de apoio da trípole a modelagem, juntamente com o modelo de objetos e o modelo dinâmico.

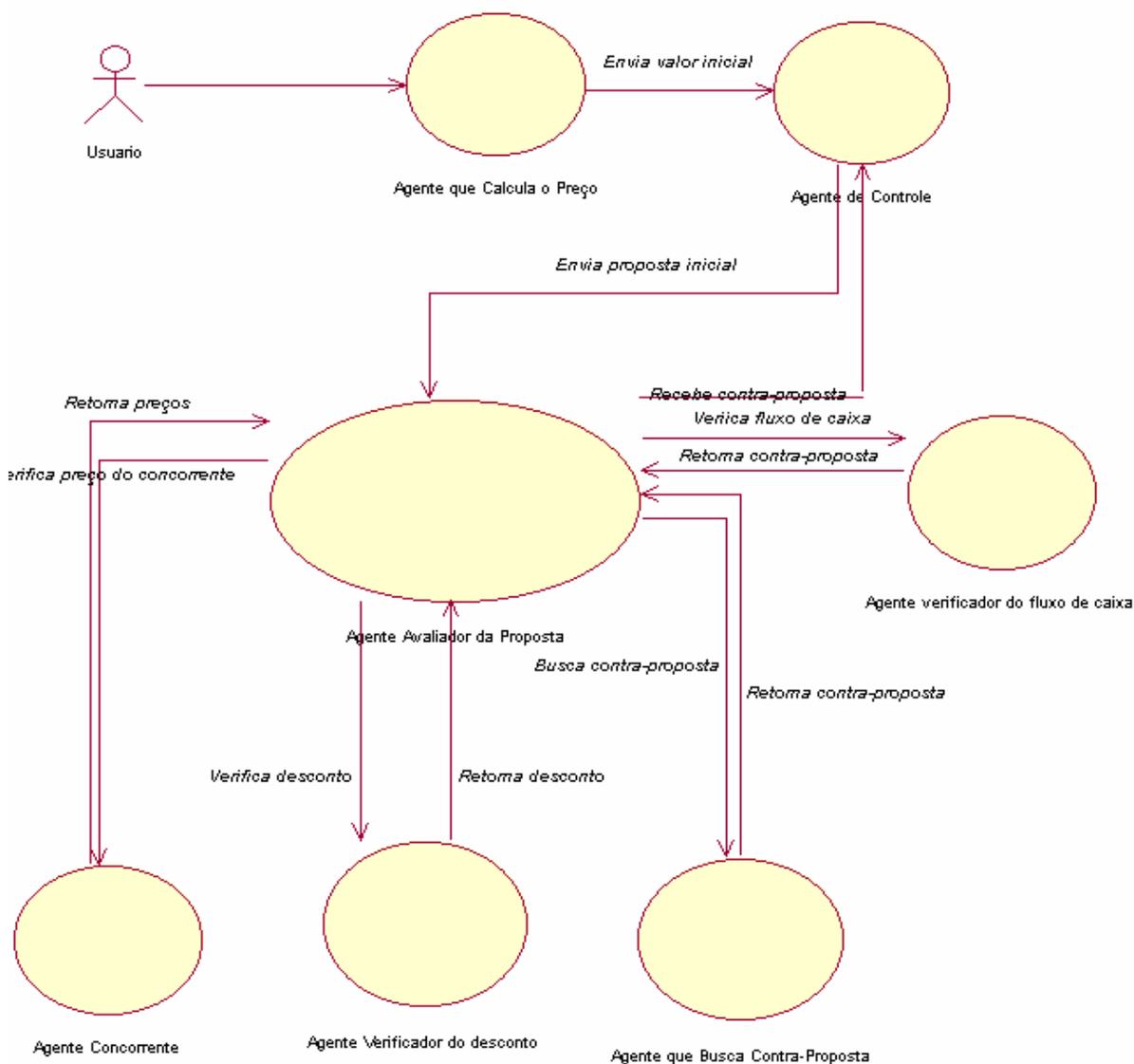


Figura 19: Modelo funcional dos agentes

6.1.5.3 Modelo Dinâmico

Os relacionamentos vinculados ao tempo são difíceis de entender. Um sistema pode ser entendido melhor quando se examina em primeiro lugar sua estrutura estática, isto é, a estrutura de seus objetos e os relacionamentos existentes entre elas em um determinado instante de tempo [RUM94]. O modelo dinâmico especifica esses aspectos relativos a tempo.

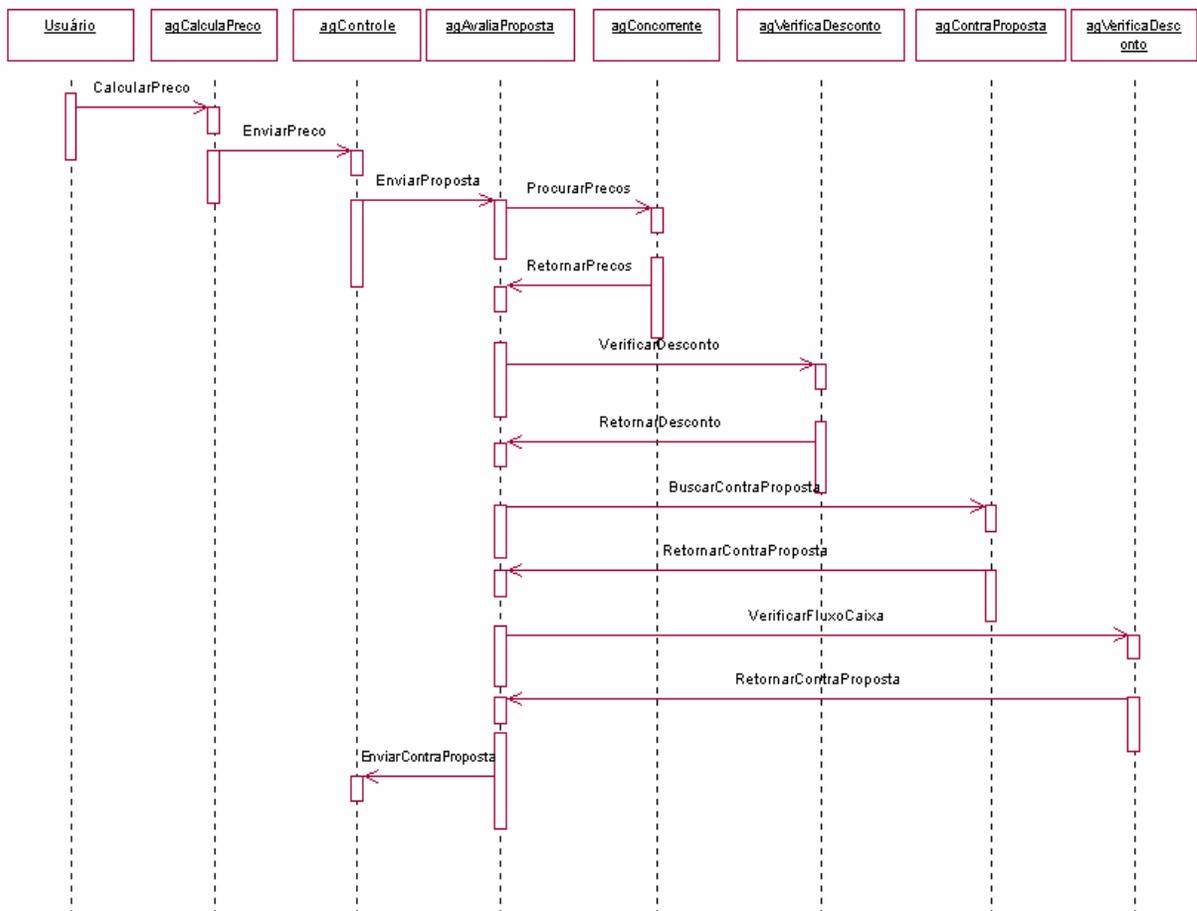


Figura 18: Modelo dinâmico dos agentes

6.2 IMPLEMENTAÇÃO

Como já dito anteriormente o protótipo do *site* possui grande parte das funcionalidades de um *site* normalmente encontrado na Internet. O usuário que acessa o protótipo inicialmente acessa uma *homepage* para a procura de produtos [figura 20].

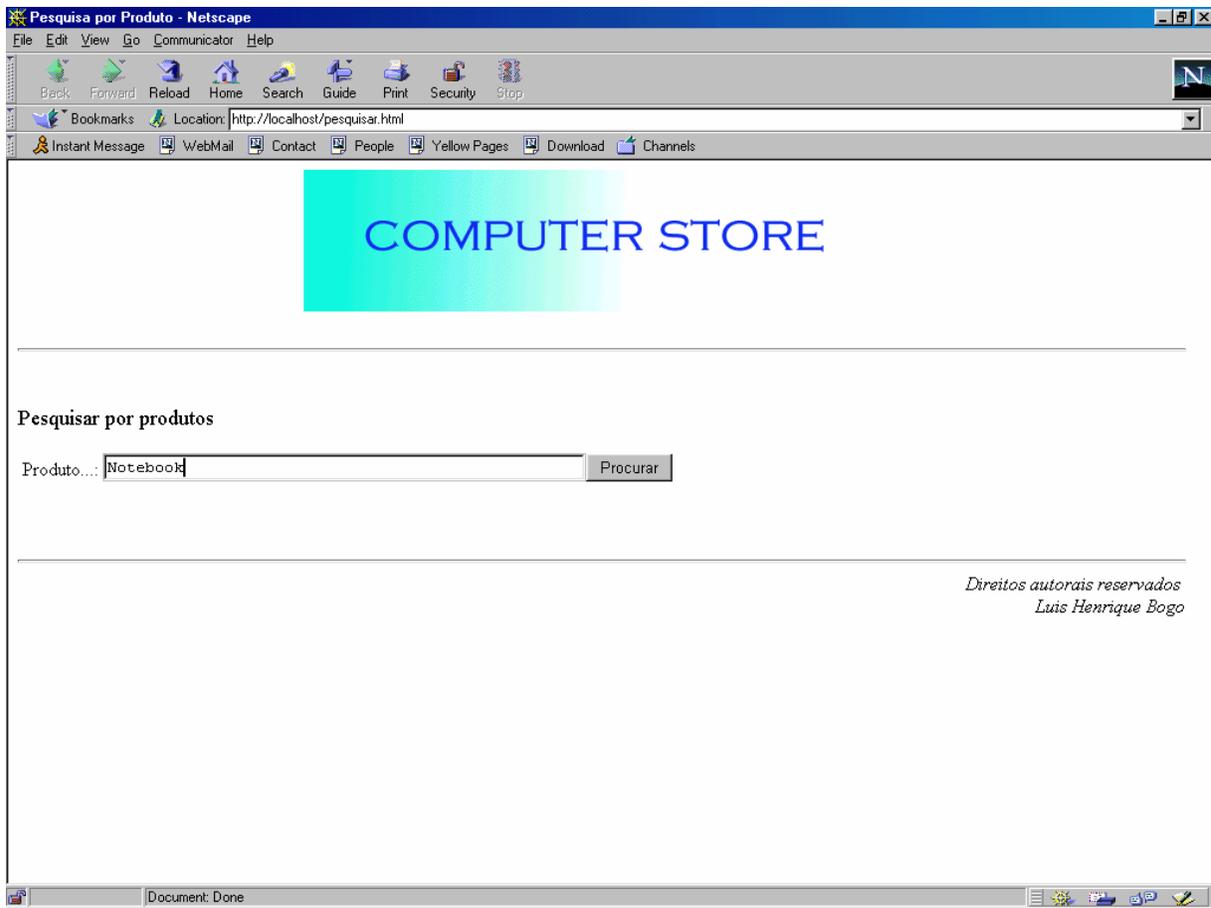


Figura 20: Homepage inicial

O produto procurado é digitado na caixa de texto e os produtos encontrados na base de dados são mostrados para o cliente [figura 21].

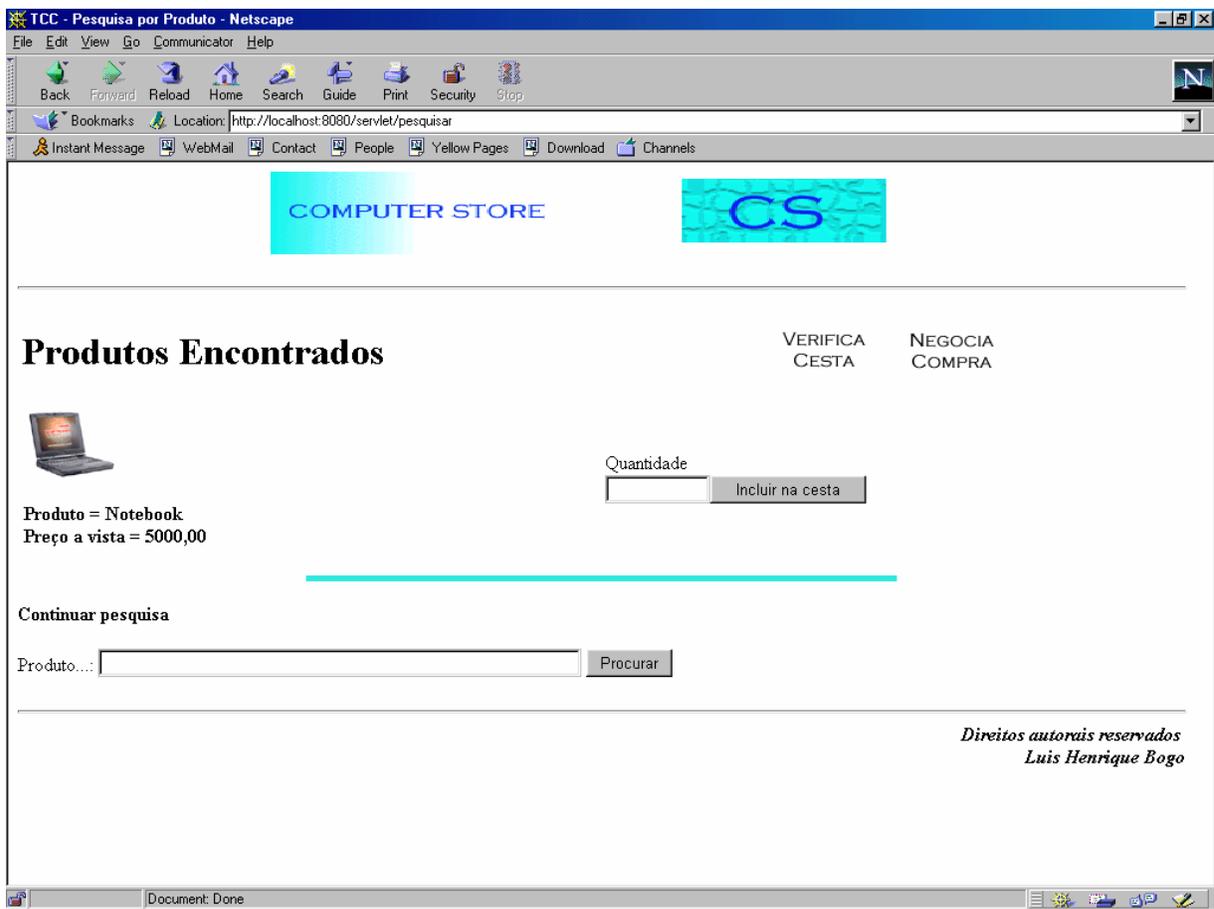


Figura 21: Produtos encontrados na base de dados

No momento que o cliente decide comprar algum produto, este é incluído na cesta de compras. Em qualquer momento do processo de compra o cliente pode verificar na cesta de compras quais os produtos comprados até o momento [figura 22].

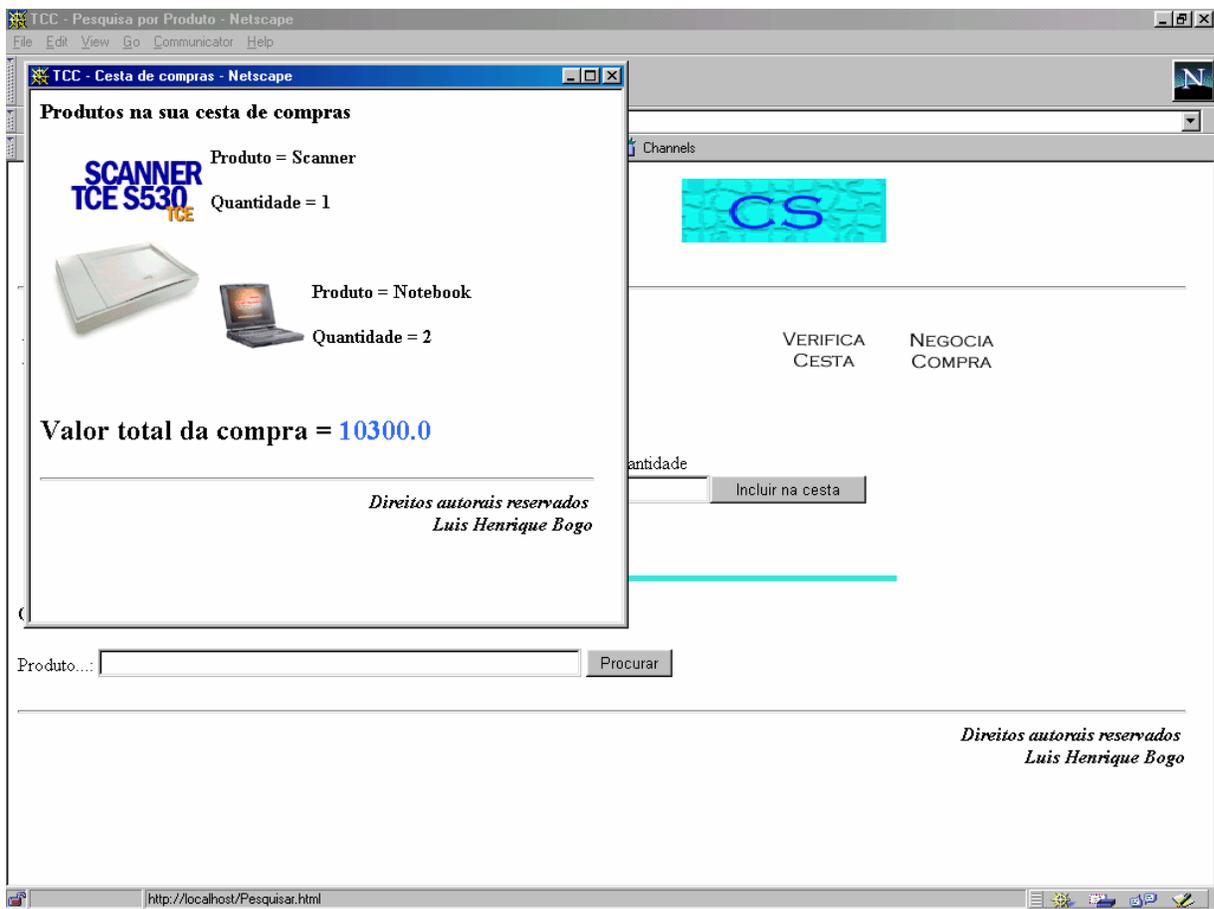


Figura 22: Produtos na cesta de compras

O processo de negociação começa a partir do momento que o usuário finalizar as compras, que podem ser de um ou mais tipo(s) de produto(s). O primeiro agente a entrar em ação é o agCalculoPreco que faz uma pesquisa em uma tabela de descontos [tabela 5] relativos ao prazo de pagamento requerido.

Forma de Pagamento escolhida	Desconto/Acréscimo aplicado
Á vista	5% (desconto)
30 dias	2% (Acréscimo)
60 dias	5% (Acréscimo)
1+1 (30 dias)	8% (Acréscimo)

Tabela 6: Desconto/Acréscimo aplicado sobre forma de pagamento

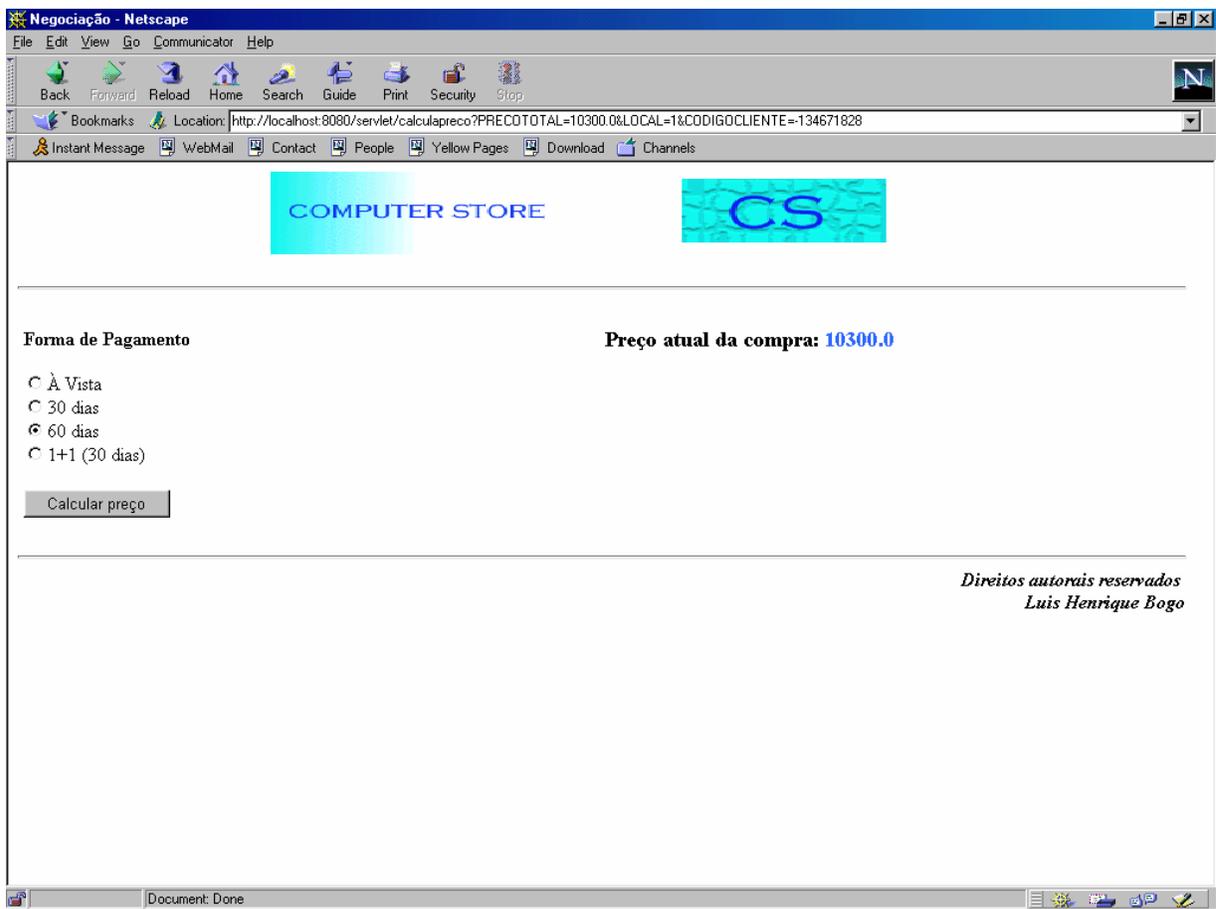


Figura 23: Homepage de cálculo de preço inicial para negociação

Após a escolha de uma forma de pagamento aparecerá a *homepage* que irá permitir o início da negociação (figura 24).

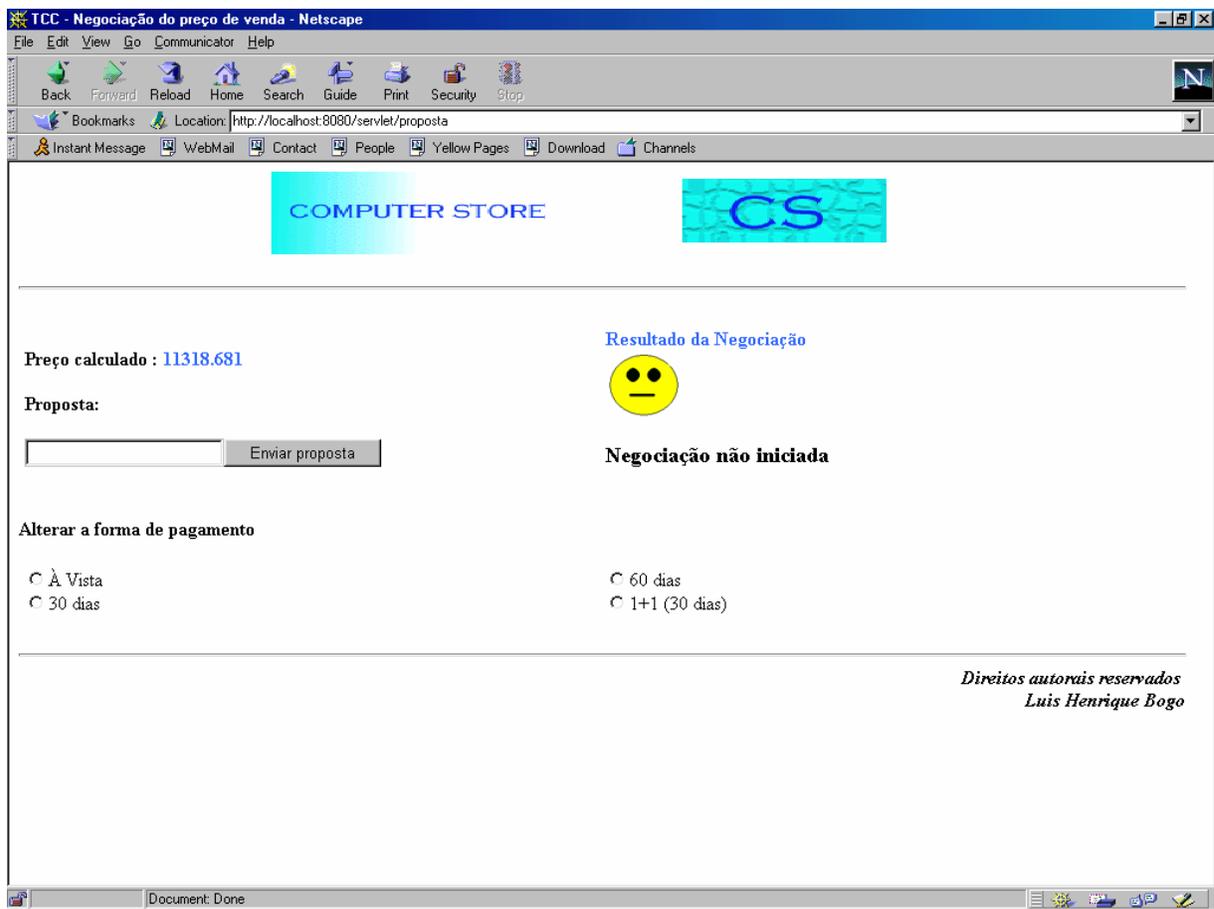


Figura 24: Tela inicial do processo de negociação

A negociação do preço inicial ainda não foi iniciada. Este estágio fica claro verificando o estado do personagem de negociação. A partir deste momento o usuário poderá oferecer novas propostas de pagamento a compra, alterando valor e forma de pagamento e estas propostas serão avaliadas pelos agentes inteligentes (avaliando tabelas 3 e 4 e propriedade referente a preço dos concorrentes) até que se chegue ao final da negociação, com sucesso ou não.

Para melhor exemplificar o funcionamento dos agentes, será demonstrado a negociação desta compra. O preço inicial calculado pelos agentes foi de R\$ 11.318,681. Inicialmente é feita uma proposta de R\$ 1.000,00. Esta proposta não será aceita visto que o percentual de desconto é maior que 50%, percentual este que foi definido como o maior percentual de desconto que poderá ser aceito.

O agente informará ao usuário que a proposta não pode ser aceita e dará as possíveis soluções para o problema [figura 25].

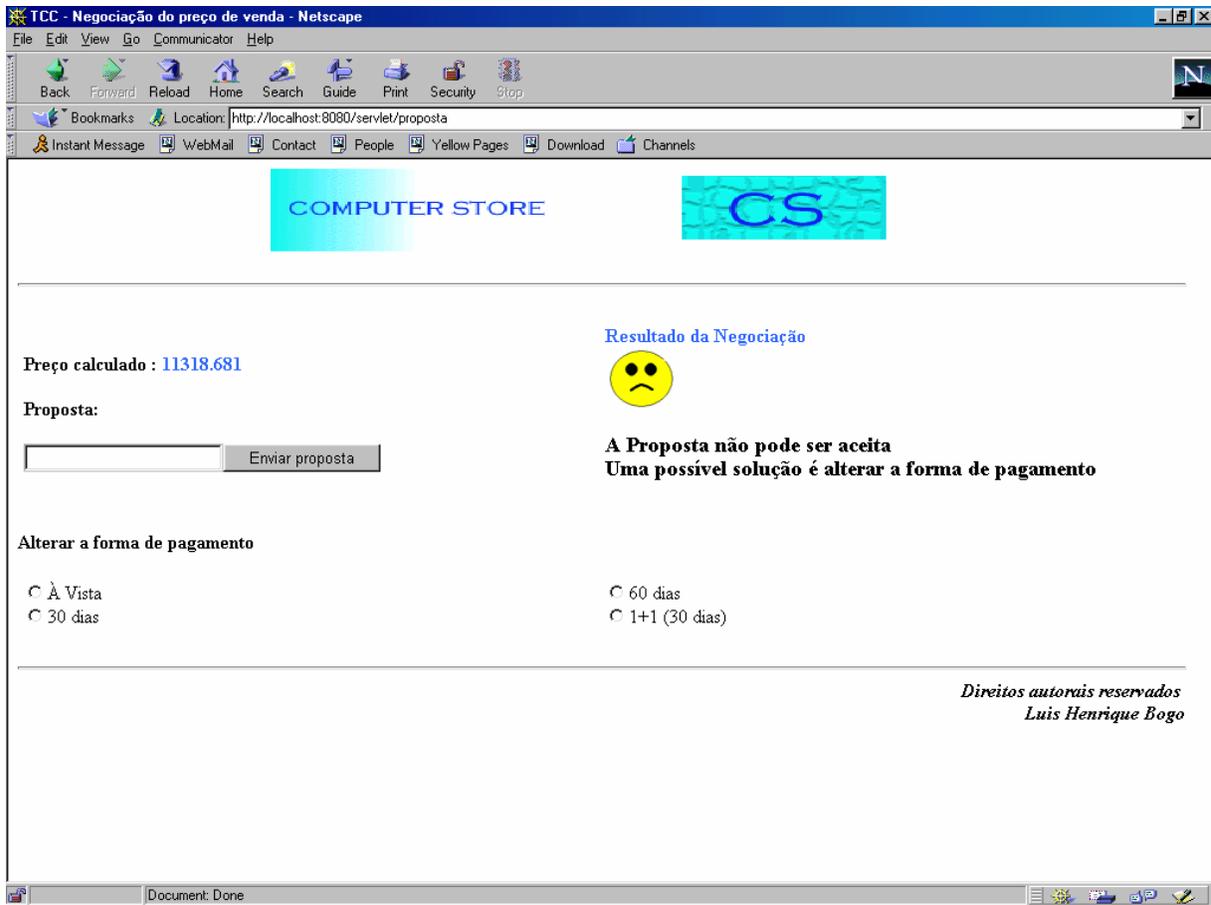


Figura 25: Agente informa que proposta não pode ser aceita

Suponha agora uma proposta de R\$ 8.000,00 com forma de pagamento de 1+1 (30dias). Os agentes verificarão as regras que foram cadastradas na base de conhecimento. Estas regras serão verificadas usando conceitos de sistema especialista, ou seja, através de regras IF – THEN – ELSE, conforme exemplo:

Se ((DescontoPedido Entre [5..15]) E (FormaPagamento <= 60 dias)) então
Proposta Aceita
Senão
Proposta não aceita

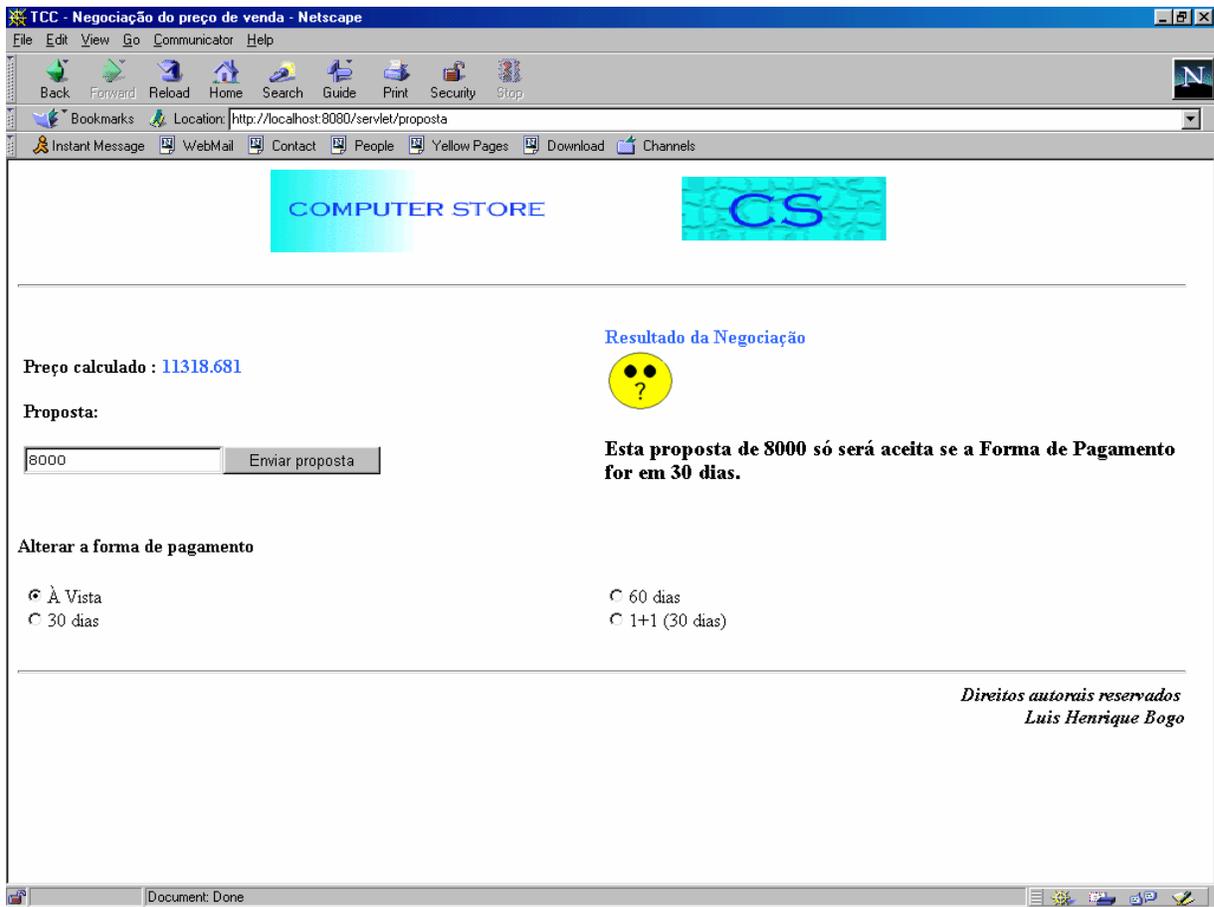


Figura 26: Contra-proposta realizada por agentes

Os agentes avaliam as regras cadastradas e a partir destas regras montam uma contra-proposta que é apresentada ao cliente [figura 26]. A contra-proposta diz que o desconto pedido só poderá ser aceito se a forma de pagamento de 30 dias, ou seja, menor que a proposta. Agora aceitaremos a contra-proposta feita pelos agentes e pode ser constatado que a proposta foi aceita [figura 27].

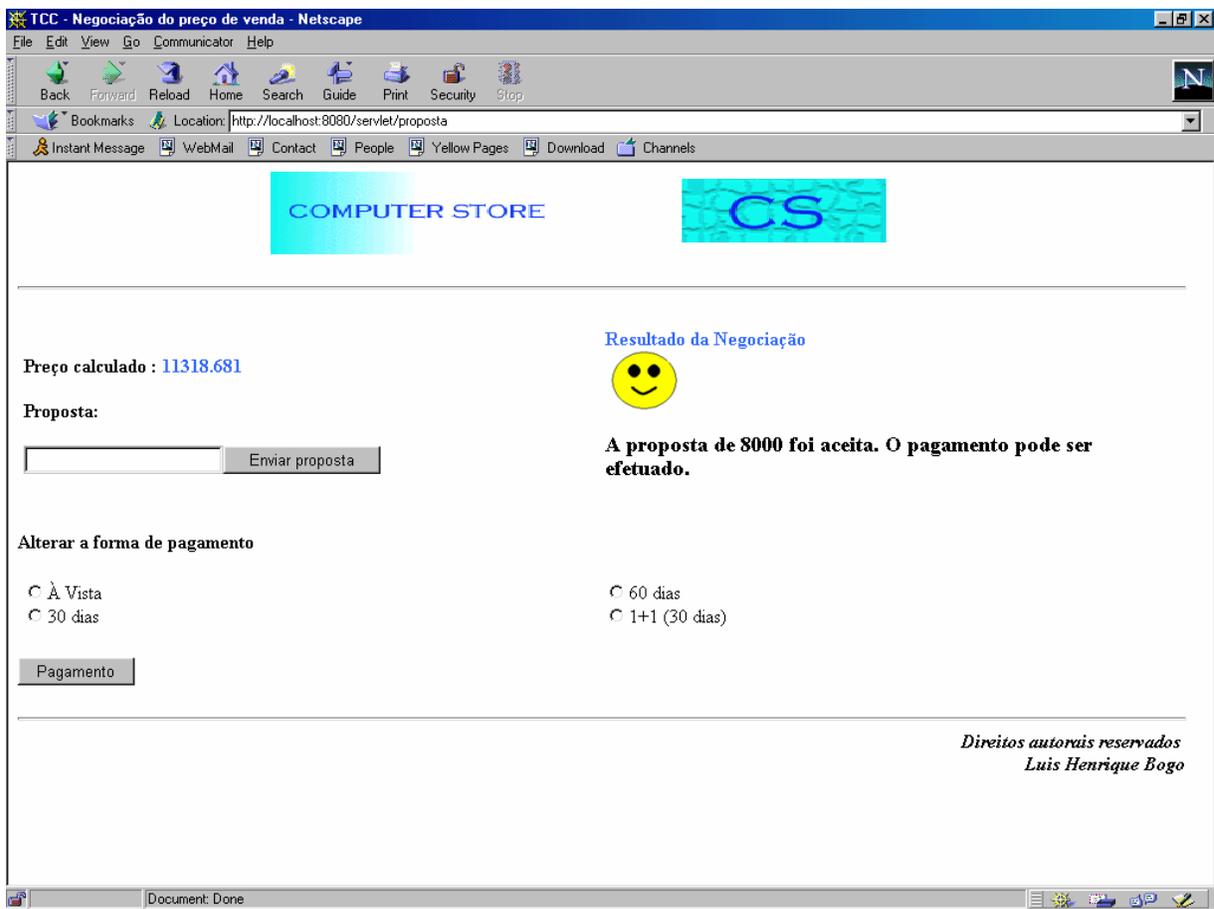


Figura 27: Agentes aceitam a proposta

Na situação descrita acima, o processo foi realizado com o fluxo de caixa acima do ponto médio. Quando o fluxo de caixa estiver baixo os agentes realizarão a negociação de uma maneira diferente. No momento em que a proposta é aceita, se a forma de pagamento utilizada não for a vista, os agentes aceitam a proposta realizada, mas ainda fazem uma contra-proposta (abaixando o valor proposto, mas propondo a venda à vista) para tentar fazer com que o dinheiro entre em caixa no mesmo dia, para normalizar o fluxo de caixa [figura 28 e 29].

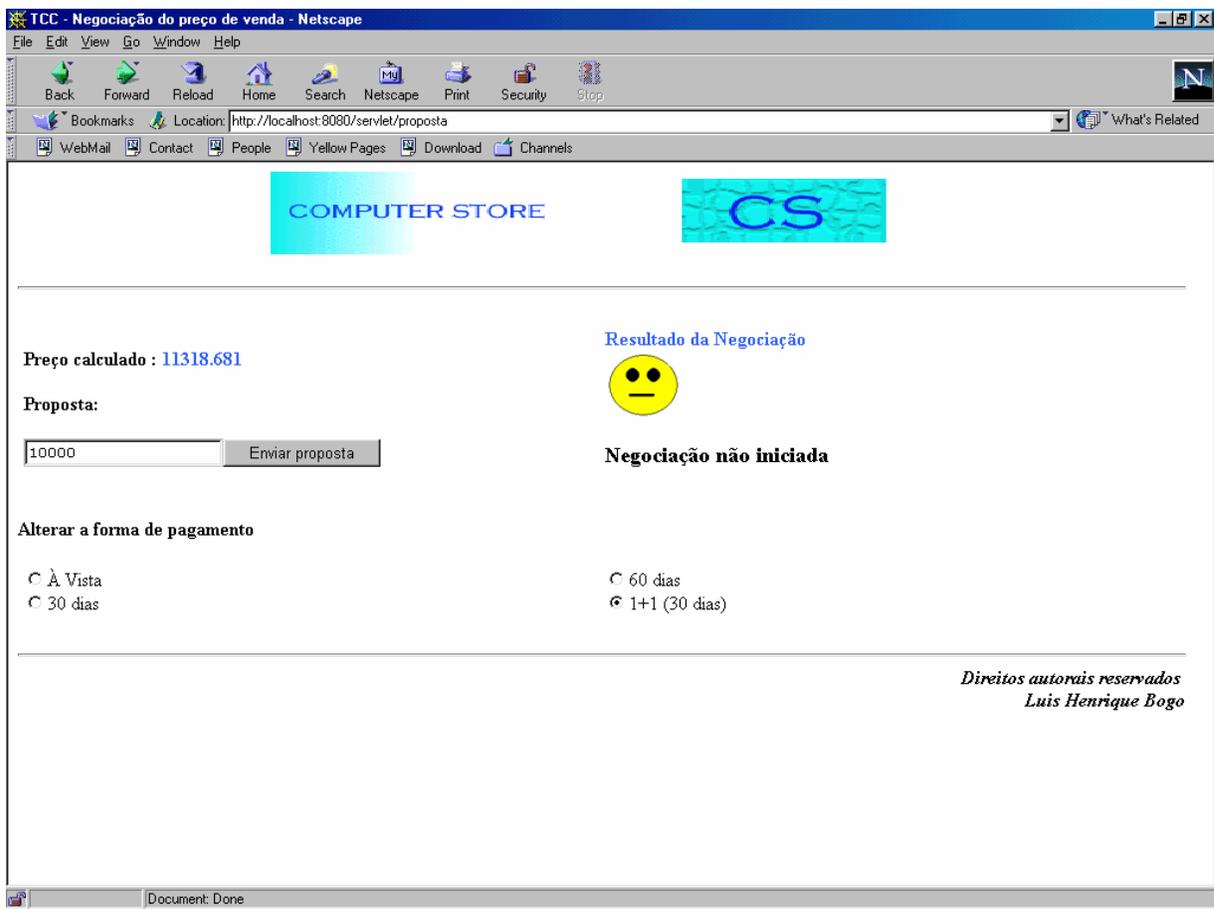


Figura 28: Proposta sendo feita com fluxo de caixa abaixo do ponto médio

Os agentes verificarão que a proposta esta sendo feita é em 1+1 (30 dias) e como o caixa necessita de entrada de dinheiro imediata, os agentes farão uma contra-proposta com pagamento à vista.

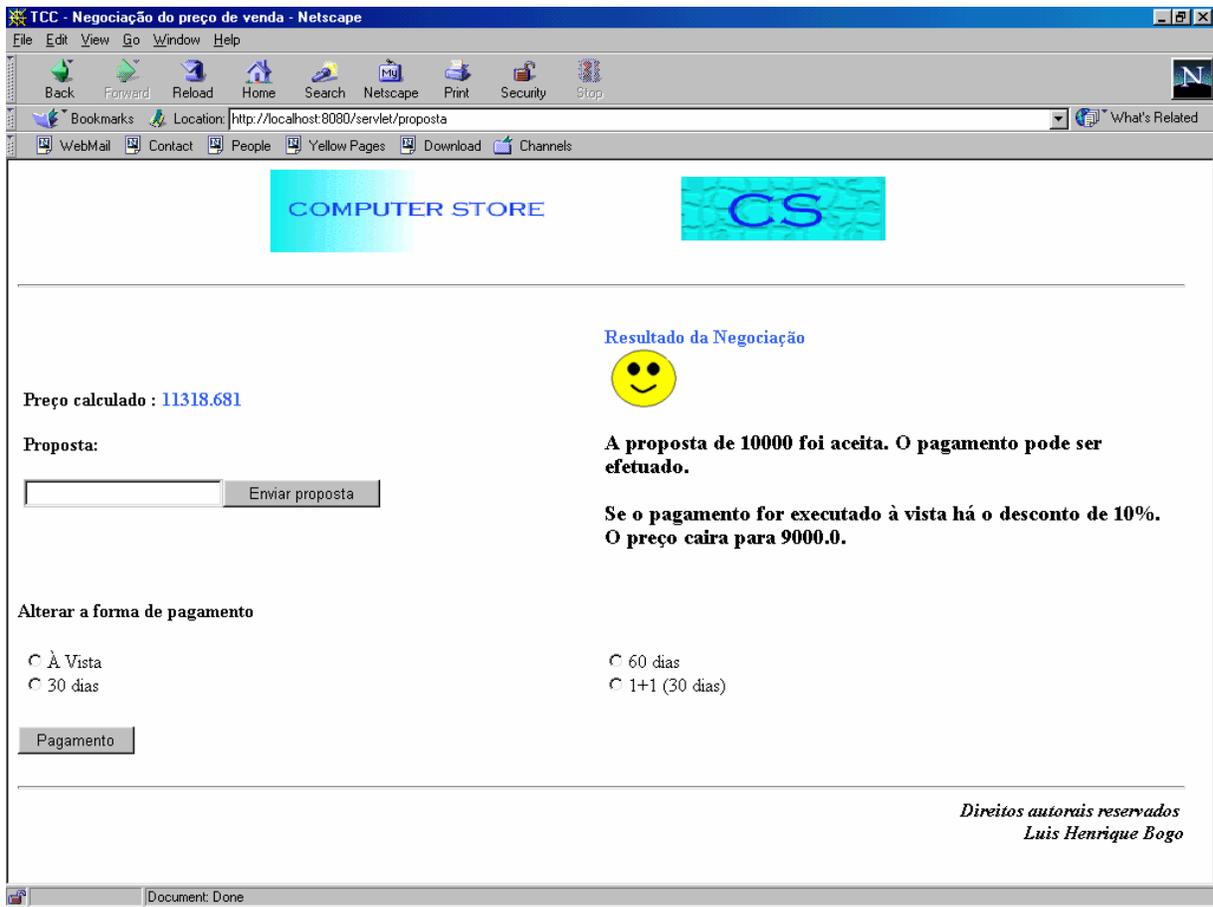


Figura 29: Agentes fazem uma contra-proposta avaliando o fluxo de caixa

Outra situação que pode acontecer é do cliente oferecer um valor maior que o proposto pelos agentes da loja [figura 30].

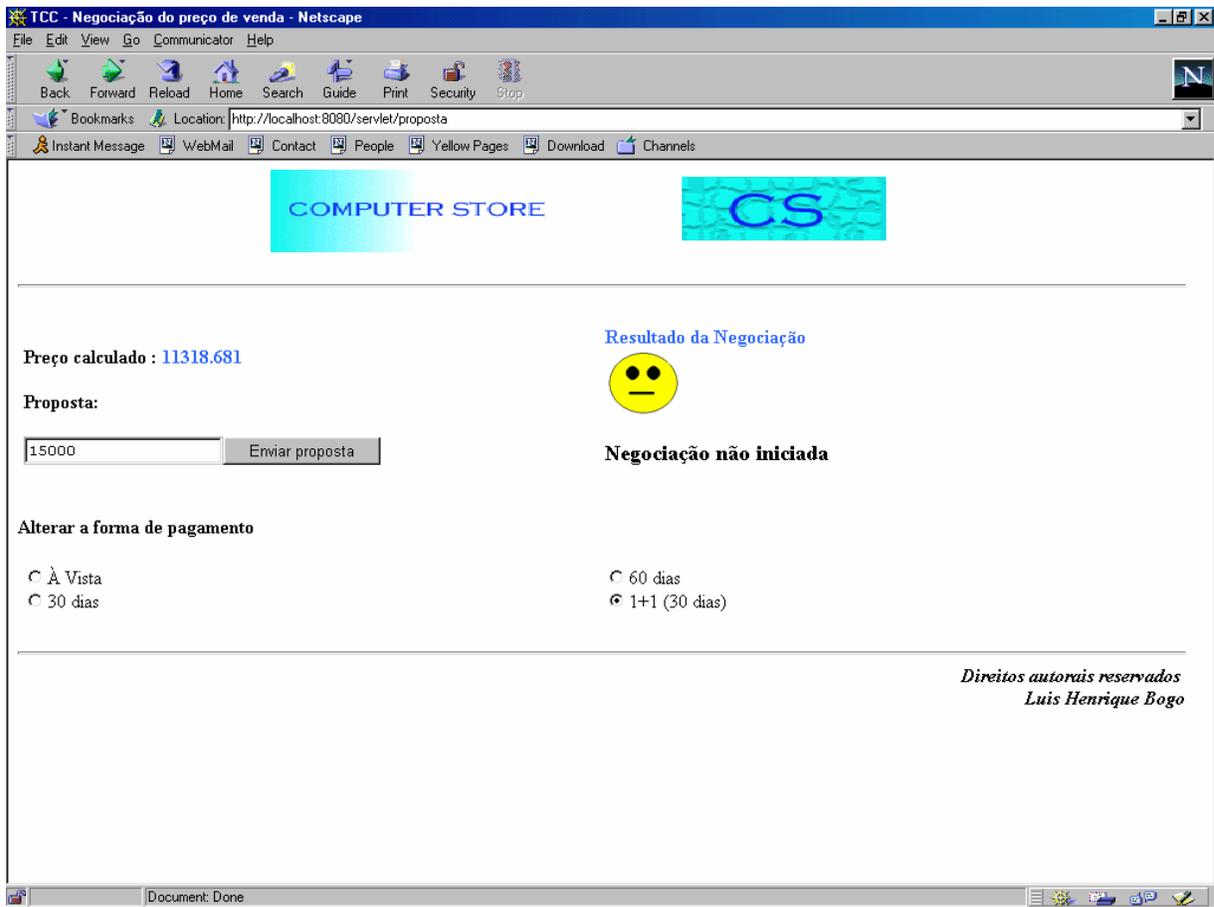


Figura 30: Preço maior sendo proposto

Os agentes verificam que o preço proposto é maior que o da tabela e informam isto ao cliente, mas permitem que a compra seja finalizada se for isso mesmo que o cliente deseje [figura 31].

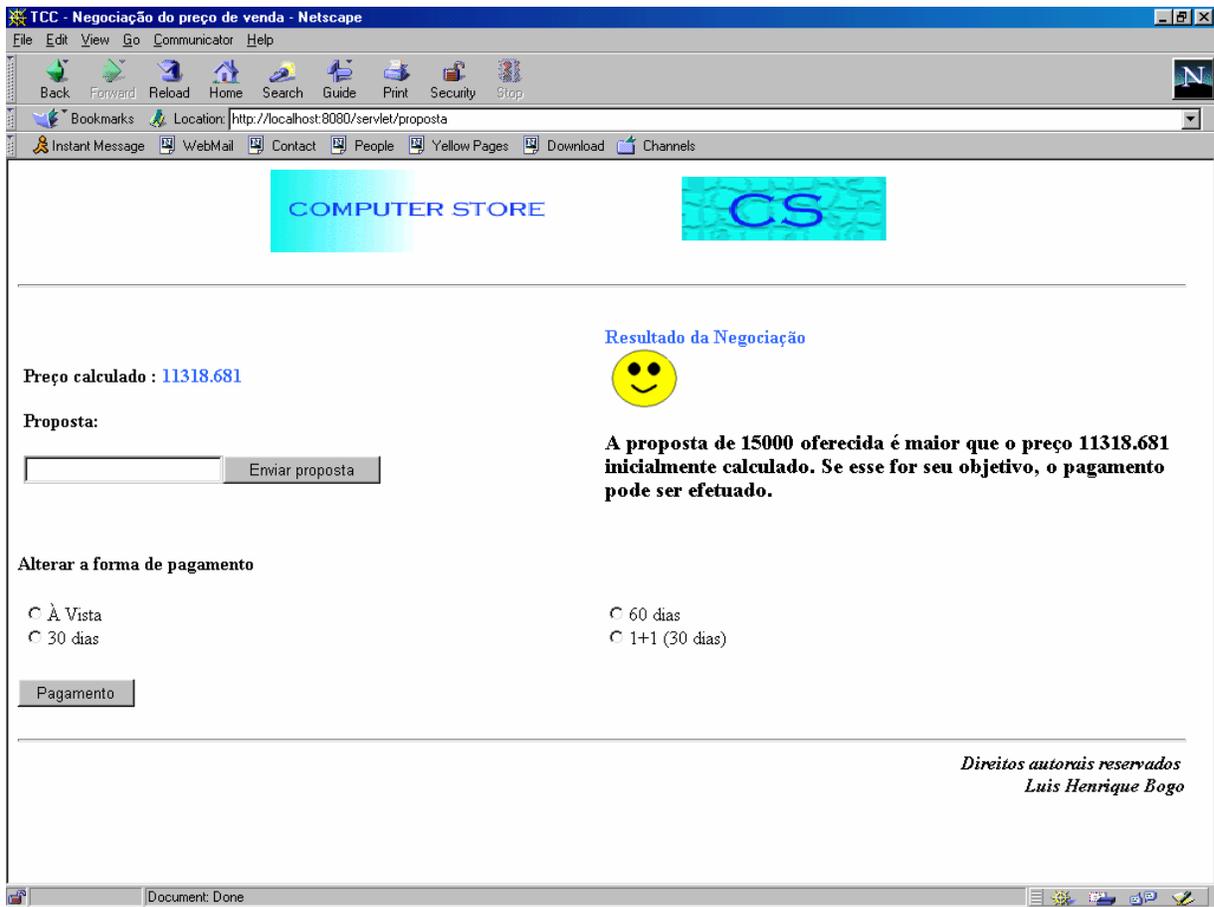


Figura 31: Agentes informam sobre preço acima do especificado

Após a negociação realizada entre agentes e cliente ter sido finalizada, o protótipo permite que a compra seja finalizada [figura 32].



Figura 32: Compra finalizada com sucesso

7 CONCLUSÕES e RECOMENDAÇÕES

Durante a pesquisa pode-se visualizar que muitas iniciativas estão sendo feitas (principalmente no exterior) para a aplicação de agentes inteligentes e outras técnicas de inteligência artificial no comércio eletrônico. Isto em parte se explica pelo fato da competição nesta área que cresce rapidamente o que faz com que proprietários de sites tenham a necessidade de ter diferenciais competitivos, que podem ser obtidos pela aplicação de técnicas inteligentes para aumentar as vendas.

Tais informações permitem concluir sobre a importância da pesquisa aplicada nesta área. Em particular, que a utilização dos agentes inteligentes no negócios on-line parece ser uma tendência irreversível.

A tecnologia de agentes inteligentes, unida com a tecnologia dos sistemas especialistas mostrou-se propícia para este fim, já que como no mundo real, as técnicas de venda são, principalmente, baseadas em regras bem definidas, definidas como regras de negócio. Essas regras necessitam ser aplicadas, porque, o preço de um produto não pode ser fixo, ele muda de acordo com a necessidade da empresa no momento. O protótipo apresentado ilustra de forma adequada a potencialidade e viabilidade imediata do emprego dessa tecnologia.

Para trabalhos futuros será interessante realizar estudos para a aplicação de redes neurais (RN) e raciocínio baseado em casos (RBC) em sistemas baseados em agentes inteligentes. Com isso os agentes podem aprender durante uma negociação, podendo assim adequar-se a novas tendências de mercado automaticamente, sem a intervenção humana.

8 ANEXOS

ANEXO 1:

Código fonte dos agentes aplicados ao sistema.

```
/**
 * Este agente avaliará a proposta feita pelo cliente
 */
class agAvaliadorProposta
{
    String sProposta, sQuantidade, sValorInicial, sFormaPagamento;
    String sFormaPagNova, sCodigoCliente, sValorTotalConc;
    String url = "jdbc:mysql://localhost:1114/Versao2";
    Connection con;
    ResultSet rs;
    Statement stmt;
    Properties p = new Properties();
    ServletOutputStream saida;

    boolean bAceite; //Váriavel que irá definir se será aceito
    boolean bPropostaMaior = false;
    boolean bEliminaConcorrente = false;

    agAvaliadorProposta(ServletOutputStream saidax, String sCodigoCli)
    {
        try
        {
            sCodigoCliente = sCodigoCli;
            saida = saidax;
            p.put("user", "borg");
            Class.forName("com.imaginary.sql.mysql.MysqlDriver");
            con = DriverManager.getConnection(url, "borg", " ");

            sValorInicial = new String();
            sFormaPagamento = new String();
            sProposta = new String();
            sQuantidade = new String();
        }
        catch (Exception ex)
        {}
    }

    boolean getPropostaMaior()
    {
        return bPropostaMaior;
    }

    void setAceite(boolean bAc)
    {
        bAceite = bAc;
    }

    boolean getAceite()
    {
        return bAceite;
    }

    void setValorConcorrente(String sValor)
    {
        sValorTotalConc = sValor;
    }

    boolean negociacaoIniciada()

```

```

    {
        if (sProposta.equalsIgnoreCase("0"))
            return false;
        else
            return true;
    }

    void finaliza()
    {
        try
        {
            String sSQL = "delete from PesquisaCliente";
            stmt = con.createStatement();
            rs = stmt.executeQuery(sSQL);
        } catch (Exception ex) {}
    }

    void setProposta(String sProp)
    {
        sProposta = sProp;
    }

    void setValorInicial(String sVI)
    {
        sValorInicial = sVI;
    }

    //Verifica no arquivo de configuração se é para igualar o
    //preço do concorrente
    boolean getEliminaConcorrente()
    {
        try
        {
            File arquivo = new File("concorrencia.conf");
            LineNumberReader inLin = new LineNumberReader(new FileReader(arquivo));
            BufferedReader in = new BufferedReader(inLin);

            String sLinha = new String();
            while ((sLinha = in.readLine()) != null)
            {
                String sCaracter = new String();
                sCaracter = sLinha.substring(0,1);
                if (sCaracter.equals("#"))
                {
                    continue;
                }
                else
                {
                    if (sCaracter.equals("0"))
                        return false;
                    else
                    {
                        if (sCaracter.equals("1"))
                        {
                            bEliminaConcorrente = true;
                            return true;
                        }
                    }
                }
            }
            return false;
        }
        catch (Exception e)
        {
            return false;
        }
    }

    void setFormaPagamento(String sFP)
    {
        if (sFP == null)
            sFP = "4";
        sFormaPagamento = sFP;
    }

```

```

boolean avaliaProposta()
{
    int iCont=0;
    int iPercentual = 0;
    bPropostaMaior = false;
    try
    {
        /******
        //Passo 1: Ver se proposta do cliente for maior que preço feito*/
        /******

        float iValorProposta = Float.parseFloat(sProposta);
        float iValorInicial = Float.parseFloat(sValorInicial);
        if (iValorProposta > iValorInicial)
        {
            bPropostaMaior = true;
            return true;
        }

        /******
        //Passo 2: Verifica preco dos concorrentes*/
        /******
        agConcorrente agConc = new agConcorrente();
        agConc.setCodigoCliente(sCodigoCliente);
        setValorConcorrente(agConc.getPrecoTotalConcorrente());

        //Apesar da proposta ser fora das regras, o sistema aceita a
        //proposta para igualar o preço do concorrente.
        if (getEliminaConcorrente())
        {
            float iProp = Float.parseFloat(sProposta);
            float iConc = Float.parseFloat(sValorTotalConc);
            if (iProp >= iConc)
                return true;
        }

        agVerificaDesconto agDesconto = new agVerificaDesconto();
        agDesconto.setProposta(sProposta);
        agDesconto.setValorInicial(sValorInicial);
        sFormaPagNova = agDesconto.getFormaPagamentoNecessaria();

        if (agDesconto.getAceitado())
            return true; //proposta aceita

        return false;

    } catch (Exception sqle) {}

    return false;
}

String getFormaPagtoNecessaria()
{
    return sFormaPagNova;
}

/******
// Este agente calcula o preço em função da forma de pagamento
/******
class agCalculoPreco
{
    String sFormaPagto, sPrecoInicial;
    String url = "jdbc:mysql://localhost:1114/Versao2";
    Connection con;
    ResultSet rs,rsAux;
    Statement stmt;
    Properties p = new Properties();

```

```

agCalculoPreco()
{
    try
    {
        p.put("user", "borg");
        Class.forName("com.imaginary.sql.mssql.MssqlDriver");
        con = DriverManager.getConnection(url,"borg"," ");
    }
    catch (Exception e)
    {}
}

void setPrecoInicial(String sPrecoIni)
{
    sPrecoInicial = sPrecoIni;
}

void setFormaPagto(String sForma)
{
    sFormaPagto = sForma;
}

String calculaPreco()
{
    String PrecoCalculado = new String();
    float fPrecoInicial = 0;
    fPrecoInicial = Float.parseFloat(sPrecoInicial);
    System.out.println("Preco = "+fPrecoInicial);
    PrecoCalculado = IAPreco(fPrecoInicial);

    return PrecoCalculado;
}

//*****
// Calcula o preço basendo-se na tabela de descontos
//*****
String IAPreco(float iPrecoInicial)
{
    ResultSet rsNew;
    String sSQL, sRetorno;
    int iPrecoFinal=0, iDescontoQtde=0, iQtde=0;
    float ft=0, fDescontoForma=0;

    System.out.println("FP = "+sFormaPagto);
    sSQL = "Select * from FormaPagtoDesconto where Codigo = "+sFormaPagto;
    try
    desconto nas tabelas //busca
    {
        stmt = con.createStatement();
        rsNew = stmt.executeQuery(sSQL);
        while (rsNew.next())
            fDescontoForma = rsNew.getFloat("Desconto");
    } catch (Exception ex)
    {}

    ft = iPrecoInicial;
    System.out.println("Preco = "+ft);
    float fPercentual = fDescontoForma / 100; //calcula o desconto
    System.out.println("Percentual = "+fPercentual);
    ft = ft / fPercentual;
    System.out.println("Preco = "+ft);

    Float Auxiliar = new Float(ft);
    sRetorno = Auxiliar.toString();
    System.out.println("Retorno = "+sRetorno);
    return sRetorno;
}
}

//*****/

```

```

// Este agente busca o preço do concorrente
//*****
class agConcorrente
{
    String sCodigoCliente;
    String url = "jdbc:mysql://localhost:1114/Versao2";
    Connection con;
    ResultSet rs;
    Statement stmt;
    Properties p = new Properties();

    agConcorrente()
    {
        try
        {
            sCodigoCliente = "0";
            p.put("user", "borg");
            Class.forName("com.imaginary.sql.mysql.MysqlDriver");
            con = DriverManager.getConnection(url,"borg"," ");
        }
        catch (Exception e)
        {}
    }

    void setCodigoCliente(String sCodigo)
    {
        sCodigoCliente = sCodigo;
    }

    String getPrecoTotalConcorrente()
    {
        try
        {
            Vector vProd = new Vector();
            Vector vQtde = new Vector();
            String sSQL = "Select * from cesta where codcliente = "+sCodigoCliente;
            stmt = con.createStatement();
            rs = stmt.executeQuery(sSQL);
            while (rs.next())
            {
                vProd.add(rs.getString("produto"));
                vQtde.add(rs.getString("qtde"));
            }

            //Busca preço de concorrentes e soma o total
            stmt = con.createStatement();
            float fPrecoTotConc = 0;
            for (int i=0; i < vProd.size(); i++)
            {
                sSQL = "Select * from Produto where Descricao =
" +(String)vProd.get(i)+" ";
                rs = stmt.executeQuery(sSQL);
                while (rs.next())
                {
                    String sPC = new String(rs.getString("PrecoConc"));
                    sPC = sPC.replace(',','.');
                    Float ft = new Float(sPC);
                    Float fQtde = new Float((String) vQtde.get(i));
                    fPrecoTotConc += ft.floatValue() * fQtde.floatValue();
                }
            }

            Float fAux = new Float(fPrecoTotConc);
            return fAux.toString();
        }
        catch (Exception e)
        {
            System.out.println("Erro buscando valor dos concorrentes");
        }
        return "0";
    }
}

```

```

//*****
// Este agente verifica se o desconto pedido pode ser aceito
//*****
class agVerificaDesconto
{
    String sValorInicial, sProposta;
    int iCont;
    String url = "jdbc:mysql://localhost:1114/Versao2";
    Connection con;
    ResultSet rs;
    Statement stmt;
    Properties p = new Properties();

    agVerificaDesconto()
    {
        try
        {
            iCont = 0;
            p.put("user", "borg");
            Class.forName("com.imaginary.sql.mysql.MysqlDriver");
            con = DriverManager.getConnection(url,"borg"," ");
        }
        catch (Exception e)
        {}
    }

    void setValorInicial(String sVlInicial)
    {
        sValorInicial = sVlInicial;
    }

    void setProposta(String sProp)
    {
        sProposta = sProp;
    }

    String getFormaPagamentoNecessaria()
    {
        String sFormaPagNova = new String();
        try
        {
            int iPercentual = BuscaPercentual();
            String sSQL = "Select * from Negociacao where PercIni <= "+iPercentual+" and
PercFin >= "+iPercentual;

            stmt = con.createStatement();
            rs = stmt.executeQuery(sSQL);
            if (iPercentual < 1)
            {
                if (sFormaPagNova == null)
                    sFormaPagNova = "4";
                return sFormaPagNova;
            }
            while (rs.next())
            {
                sFormaPagNova = rs.getString("FP");
                if (sFormaPagNova == null)
                    sFormaPagNova = "4";
                iCont++;
            }
            return sFormaPagNova;
        }
        catch (Exception e)
        {
            System.out.println("Problema buscando forma de pagamento necessária");
        }

        return sFormaPagNova;
    }
}

//*****

```

```

//Calcula percentual de desconto pedido pelo cliente na proposta*
//*****
int BuscaPercentual()
{
    try
    {
        float fValorIni, fProposta, fPer=0;

        fValorIni = Float.parseFloat(sValorInicial);
        fProposta = Float.parseFloat(sProposta);

        if (fProposta < fValorIni)
        {
            //calcula diferença percentual da proposta
            fPer = (fProposta * 100) / fValorIni;
            fPer = 100 - fPer;
            Float fRet = new Float(fPer);
            int iRet = fRet.intValue();
            return iRet;
        }
        else
            return 0;
    }
    catch (Exception e)
    {
        System.out.println("Erro buscando percentual");
    }

    return 0;
}

boolean getAceitado()
{
    if (iCont > 0)
        return true;

    return false;
}
}

class agContraProposta
{
    boolean bPropostaMaior, bEliminaConcorrente, bAceite;
    String sFormaPagtoNova, sFormaPagtoAntiga, sProposta, sValorInicial;

    agContraProposta()
    {
        bPropostaMaior = false;
        bEliminaConcorrente = false;
        bAceite = false;
    }

    void setPropostaMaior(boolean bPropMaior)
    {
        bPropostaMaior = bPropMaior;
    }

    void setEliminaConcorrente(boolean bElimConc)
    {
        bEliminaConcorrente = bElimConc;
    }

    void setValorInicial(String sValIni)
    {
        sValorInicial = sValIni;
    }

    void setFormasPagamento(String sFormaNova, String sFormaAntiga)
    {
        sFormaPagtoNova = sFormaNova;
        sFormaPagtoAntiga = sFormaAntiga;
    }
}

```

```

void setProposta(String sProp)
{
    sProposta = sProp;
}

boolean getAceite()
{
    return bAceite;
}

String getContraProposta()
{
    String sContraProposta = new String();
    try
    {
        if (!bPropostaMaior)
        {
            if (!bEliminaConcorrente)
            {
                if (sFormaPagtoNova.equals("0"))
                {
                    sContraProposta = "A proposta de "+sProposta+" foi aceita";
                    bAceite = true;
                }
                else
                {
                    int iAux,iAux2;
                    iAux = Integer.parseInt(sFormaPagtoAntiga);
                    iAux2 = Integer.parseInt(sFormaPagtoNova);

                    //Se FP proposta for maior que a das FP regras
                    if (iAux > iAux2)
                    {
                        String sAuxFP = new String();

                        if (sFormaPagtoNova == null)
                        {
                            sContraProposta = "Problemas durante a negociação";
                            return sContraProposta;
                        }

                        if (sFormaPagtoNova.equals("1"))
                            sAuxFP = "À Vista";
                        else
                            if (sFormaPagtoNova.equals("2"))
                                sAuxFP = "em 30 dias";
                            else
                                if (sFormaPagtoNova.equals("3"))
                                    sAuxFP = "em 60 dias";
                                else
                                    if (sFormaPagtoNova.equals("4"))
                                        sAuxFP = "1+1 vezes (30 dias)";

                        sContraProposta = "Esta proposta de "+sProposta+" só será aceita
se a Forma de Pagamento for "+sAuxFP+".";
                    }
                    else
                    {
                        String sFC = retornaPropostaFluxoCaixa();
                        bAceite = true;
                        sContraProposta = "A proposta de "+sProposta+" foi aceita. O
pagamento pode ser efetuado.<p>";
                        sContraProposta += sFC;
                    }
                }
            }
        }
        else
        {
            String sFC = retornaPropostaFluxoCaixa();
            sContraProposta = "A proposta de "+sProposta+" foi aceita! O pagamento
pode ser efetuado.<p>";
        }
    }
}

```

```

        sContraProposta += sFC;
        bAceite = true;
    }
}
else
{
    sContraProposta = "A proposta de "+sProposta+" oferecida é maior que o
preço "+sValorInicial+" inicialmente calculado. Se esse for seu objetivo, o
pagamento pode ser efetuado.";
    bAceite = true;
}
} catch (Exception e) {}

return sContraProposta;
}

String retornaPropostaFluxoCaixa()
{
    agVerificaFluxoCaixa agFC = new agVerificaFluxoCaixa();
    String sFC = new String();
    String sPM = new String();
    String sPropostaFC = new String();

    sFC = agFC.getSaldoFluxoCaixa();
    sPM = agFC.getPontoMedio();
    sPM = sPM.substring(4);

    float fSaldo = Float.parseFloat(sFC);
    float fPontoMedio = Float.parseFloat(sPM);

    //Saldo do fluxo de caixa está baixo
    //Fazer contra-proposta baseando-se neste ponto
    if (fSaldo < fPontoMedio)
    {
        if (!sFormaPagtoNova.equals("1")) //Pagamento não é à vista
        {
            /*Abaixando o preço para a compra ser a vista*/
            String sDesconto = new String();
            sDesconto = agFC.getDesconto();
            sDesconto = sDesconto.substring(4);
            float fDesconto = Float.parseFloat(sDesconto);

            float fProposta = Float.parseFloat(sProposta);
            System.out.println(fProposta);

            float fAux = fDesconto / 100;
            float fAux2 = fAux * fProposta;
            fProposta = fProposta - fAux2; //Valor da nova proposta
            Float fPropAux = new Float(fProposta);

            sPropostaFC = " Se o pagamento for executado à vista há o desconto de
"+sDesconto+"%. O preço cairá para "+fPropAux.toString()+".";
            return sPropostaFC;
        }
    }

    return sPropostaFC;
}
}

//*****
// Este agente calcula o preço em função da forma de pagamento
//*****
class agVerificaFluxoCaixa
{
    agVerificaFluxoCaixa()
    {
    }

    String getDesconto()

```

```

{
    try
    {
        File arquivo = new File("PontoMedio.conf");
        LineNumberReader inLin = new LineNumberReader(new FileReader(arquivo));
        BufferedReader in = new BufferedReader(inLin);

        String sLinha = new String();
        while ((sLinha = in.readLine()) != null)
        {
            String sCaracter = new String();
            sCaracter = sLinha.substring(0,1);
            if (sCaracter.equals("#"))
                continue;
            else
                if (sCaracter.equals("B"))
                    return sLinha;
        }
    }
    catch (Exception e){
        return "";
    }
    return "";
}

String getPontoMedio()
{
    try
    {
        File arquivo = new File("PontoMedio.conf");
        LineNumberReader inLin = new LineNumberReader(new FileReader(arquivo));
        BufferedReader in = new BufferedReader(inLin);

        String sLinha = new String();
        while ((sLinha = in.readLine()) != null)
        {
            String sCaracter = new String();
            sCaracter = sLinha.substring(0,1);
            if (sCaracter.equals("#"))
                continue;
            else
                if (sCaracter.equals("A"))
                    return sLinha;
        }
    }
    catch (Exception e){
        return "";
    }
    return "";
}

String getSaldoFluxoCaixa()
{
    try
    {
        File arquivo = new File("FluxoCaixa.conf");
        LineNumberReader inLin = new LineNumberReader(new FileReader(arquivo));
        BufferedReader in = new BufferedReader(inLin);

        String sLinha = new String();
        while ((sLinha = in.readLine()) != null)
        {
            String sCaracter = new String();
            sCaracter = sLinha.substring(0,1);
            if (sCaracter.equals("#"))
                continue;
            else
                return sLinha;
        }
    }
    catch (Exception e){
        return "";
    }
}

```

```
    }  
    return "";  
  }  
}
```

9 BIBLIOGRAFIA

- [ALB1999] ALBERTIN, Alberto Luiz. **Comércio Eletrônico. Modelo, Aspectos e Contribuições de sua Aplicação.** São Palo: Atlas, 1999. 220p.
- [AMO1999] AMORIM, Giana Magali de. **Estratégias para difusão de um ambiente virtual para comércio eletrônico via Internet. Um estudo de caso – CIMM: Centro de Informação Metal Mecânica.** Florianópolis, 1999. 117p. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.
- [BAR1999] BARTH, Carlos Alberto. **Multi-Agentes no Auxílio à Utilização de Conhecimento Distribuído.** Blumenau, 1999. 69p. Monografia (Bacharelado) Curso de Ciências da Computação, Universidade Regional de Blumenau.
- [BEA1996] BEAM, Carrie.; SEVEG, Arie. **Automated Negotiations: A Survey of the State of the Art.** Berkley: Fischer Center for Information Technology & Management – University of California, 1998.
- [BIG1998] BIGUS, Joseph P.; BIGUS, Jennifer. **Constructing Intelligent Agents whit Java.** New York: Wiley Computer Publishing, 1998. 379p.
- [BIT1998] BITTENCOURT, Guilherme. **Inteligência Artificial – Ferramentas e Teorias.** Florianópolis: Editora da UFSC, 1998. 362p.
- [BLU2000] **Blue Earth Language Solutions.** [OnLine] Disponível na internet via WWW. URL: <http://www.blueearth.net/en/resources/index.html> April, 2000.
- [BRE1998] BRENNER, Walter.; ZARNEKOW, Rüdiger.; WITTIG, Hartmut. **Intelligent Software Agents – Foundations and Applications.** New York: Springer-Verlag, 1998. 326p.

- [DUR1996] DURKIN, John. **Expert Systems: A view of the field**. IEEE Expert, April 1996. 644p.
- [FAR1998] FARACO, Rafael Avila. **Uma Arquitetura de Agentes Para Negociação Dentro do Domínio do Comércio Eletrônico**. Florianópolis, 1998, 100p. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.
- [FER1991] FERBER, J.; GASSER, L. **Intelligence artificielle distribuée**, 1991. Tutorial Notes of the 11^a Conference on Expert Systems and their Applications, Avignon'91, France.
- [FIN1993] FININ, Tim, et all. **DRAFT. Specification of the KQML Agent-Communication Language**, 1993. The DARPA Knowledge Sharing Initiative External Interfaces Working Group, Stanford, USA.
- [FLE1996] FLEISCHHAUER, Luciana I. A. **Uso da Tecnologia de Agentes na Integração da Programação da Produção**. Florianópolis, 1996, 68p. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.
- [GIE1998] GIESE, Luiz F. **Estrutura de Agentes para o Processo de Compra e Venda utilizando Tomada de Decisão Difusa**. Florianópolis, 1998, 114p. Dissertação (Mestrado) - Programa de Pós-Graduação em Ciências da Computação, Universidade Federal de Santa Catarina.
- [HEI1995] HEILMAN, Kathrin.; KIHANYA, Dan. **Intelligent Agents: A Technology and Business Application Analysis**. [OnLine] Disponível na internet via WWW. URL: <http://haas.berkeley.edu/~heilmann/agents/index.html>. November, 1995.
- [JEN1998] JENNINGS, Nicolas R.; WOOLDRIDGE, Michael. **Agent Technology: Foundations, Applications and Markets**. Springer-Verlag, 1998. 325p.

- [JUL1999] JULIANI, Jordan Paulesky. **Sistema Inteligente de Monitoramento Orçamentário de Fluxo de Caixa**. São José, 1999, 113p. Monografia (Bacharelado) – Curso de Ciências da Computação, Universidade do Vale do Itajaí.
- [MAE1998] MAES, Pattie.; GUTMANN, Robert H.; MOUKAS, Alexandros V. **Agents that Buy and Sell: Transforming Commerce as we Know It**. Communications of the ACM: MIT, 1998.
- [MOD2000] MODRO, Nilson Ribeiro. **Sistema Inteligente de Monitoramento e Gerenciamento Financeiro para Micro e Pequenas Empresas**. Florianópolis, 2000, 98p. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.
- [NOR1996] NORMAN, Timothy J.; JENNINGS, Nick R.; FARATIN, Peyman. Designing and Implementing a Multi-Agent Architecture for Business Process Management. In: **Intelligent Agents III. Agents Theories, Architectures, and Languages**, pg – 261-275, 1996.
- [PAL1998] PALAZZO, Luiz A. M.; CASTILHO, José M. V. **Agentes de Informação Inteligentes**. Porto Alegre, 1998, 93p. Dissertação (Mestrado) - Programa de Pós-Graduação em Informática, Escola de Informática.
- [PEN1996] PETRIE, Charles. What is an Agent?. In: **Intelligent Agents III. Agents Theories, Architectures, and Languages**, pag – 41-43, 1996.
- [RIC1994] RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. São Paulo: Makron Books, 1994. 503p.
- [SIE1997] SIERRA, Carles.; JENNINGS, Nick R.; NORIEGA, Pablo. A Framework for Argumentation-Based Negotiation. In: **Intelligent Agents IV. Agents Theories, Architectures, and Languages**, pg – 177-192, 1997.

[SOU1996] SOUZA, Eliane Moreira de Sá. **Uma estrutura de agentes para Assessoria na Internet**. Florianópolis, 1996, 118p. Dissertação (mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.

[ZEN1996] ZENG, Dajun; SYCARA, Katia. How Can na Agent Learn to Negotiate? In: **Intelligent Agents III. Agents Theories, Architectures, and Languages**, pag – 233-244, 1996.