

Celso Gonçalves Camilo Junior

*Sistema Multiagente de Negociação
Automatizada para Comércio Eletrônico*

Goiânia – GO

Abril / 2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Celso Gonçalves Camilo Junior

*Sistema Multiagente de Negociação
Automatizada para Comércio Eletrônico*

Dissertação apresentada à Coordenação do Mestrado em Engenharia Elétrica e de Computação da Universidade Federal de Goiás para a obtenção do título de Mestre em Engenharia Elétrica e de Computação. Área de concentração: Engenharia de Computação. Linha de Pesquisa: Planejamento e Computação Aplicada.

Orientador:

Prof. Dr. Reinaldo Gonçalves Nogueira

Co-orientador:

Prof. Dr. Cássio Dener Noronha Vinhal

MESTRADO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
ESCOLA DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS

Goiânia – GO

Abril / 2005

*Dedico esta dissertação a meus pais,
cujo exemplo de honestidade e trabalho
tem sido um norteador para a minha vida,
meus irmãos, que juntos aos meus pais
e familiares engrandecem a família,
e para minha noiva, que demonstrou ser uma mulher sábia
quando criou um ambiente de paz e muito amor,
necessário para a boa conduta deste trabalho.*

Agradecimentos

Dedico meus sinceros agradecimentos para:

- o professor doutor Prof. Dr. Reinaldo Gonçalves Nogueira, pela orientação, incentivo e dedicação;
- o professor doutor Prof. Dr. Cássio Dener Noronha Vinhal, pela co-orientação;
- os colegas dos laboratórios PIRENEUS e NEPE, pelo incentivo e auxílio;
- o professor doutor Prof. Dr. Weber Martins, pela atenção e permissão para conduzir aulas, de uma de suas turmas, da Graduação de Engenharia Elétrica e de Computação;
- os colegas da Universidade Católica de Goiás, que tanto me incentivaram e torceram para o sucesso deste trabalho;
- os funcionários do Mestrado em Engenharia Elétrica e de Computação/UFG, Sr. João e Sra. Dulce, pela atenção dispensada;
- todos os professores do Mestrado em Engenharia Elétrica e de Computação/UFG, pelo auxílio;
- meus familiares, pelo suporte dado durante o trabalho;
- minha noiva, pela dedicação;
- todos os colegas do Mestrado em Engenharia Elétrica e de Computação da UFG;
- todos aqueles que, diretamente ou indiretamente, contribuíram, com críticas e sugestões, para o sucesso deste trabalho.

*“...Nem vos façais chamar de mestre,
porque só tendes um Mestre, o Cristo.
O maior dentre vós será vosso servo.
Aquele que se exaltar será humilhado
e aquele que se humilhar será exaltado...”*

Mateus 23:10,11,12

Resumo

Analisando os *softwares* existentes para comércio eletrônico, identifica-se uma demanda por *softwares* mais inteligentes. Deseja-se que estes *softwares* sejam capazes de executar ações de forma mais autônoma com acréscimo de eficácia e eficiência. O campo de negociação automatizada para agentes de *software* tem contribuído com estudos para viabilizar as interações entre agentes autônomos com interesses concorrentes ou complementares.

Este trabalho, através da aplicação da negociação nas transações comerciais virtuais, propõe um novo protocolo de negociação automatizada para agentes. Este protocolo, que tem como referência o protocolo citado por Farantin em 1998, tem o objetivo de melhorar as utilidades dos participantes, fazendo com que compradores e vendedores tenham mais retorno nas negociações em que obtiveram acordo.

A aplicação desenvolvida, denominada SMANACE (Sistema Multiagente de Negociação Automatizada para Comércio Eletrônico), para provimento de simulações e comparações entre o protocolo proposto e o protocolo referenciado por Farantin em 1998, permite que utilizadores deleguem intenções de compra ou venda para os agentes de *software*. Estes agentes negociaram de forma autônoma, guiados pelos protocolos e um modelo de negociação que é multilateral (várias negociações simultâneas) e multidimensional (vários atributos como preço, prazo de entrega e forma de pagamento).

Foram criados dez cenários para as simulações, sendo quatro cenários sem concorrência e outros seis com concorrência entre os agentes analisados. Como resultado, obteve-se acréscimo na eficiência para compradores e vendedores, quando aplicado o protocolo proposto. Nota-se, também, um melhor desempenho deste protocolo quando aplicado em ambientes com concorrência. Com isso, pode-se concluir que o protocolo proposto é útil academicamente, pois contribui com a evolução da negociação automatizada para agentes de *software*; e mercadologicamente, pois diminui os gastos envolvidos em uma negociação e aumenta os retornos das transações.

Abstract

A survey of electronic commerce software currently available reveals a demand for more intelligent programs. These are characterized by being capable of operating autonomously, increasing their effectiveness and efficiency. The related field of automated brokering, using software agents, investigates the issue of interactions among autonomous agents with competing or complementary interests.

The present work proposes a new automated brokering agent protocol, for negotiation in virtual commerce transactions. The proposed protocol, based on a protocol created by Farantin in 1998, increases the participating parties' utility, leading to increased yields for both sellers and buyers from successful negotiations.

An application, called SMANACE (*Sistema Multiagente de Negociação Automatizada para Comércio Eletrônico*), was developed to run simulations comparing the proposed protocol to Farantin's. SMANACE allows users to delegate buy/sell options to software agents. These agents autonomously pursue negotiations, guided by the aforementioned protocols and a brokerage model that is multi-lateral (many simultaneous negotiations) and multi-dimensional (using multiple attributes such as price, delivery time and payment methods).

The simulations employed ten scenarios, of which six allowed for competition among agents. The results obtained indicate an efficiency increase in both sellers and buyers, with the proposed protocol, when compared to Farantin's. The best results shown by the proposed protocol occurred in the competitive scenarios. Therefore, the proposed agent negotiation protocol gives a useful contribution to electronic commerce research and presents a viable real-world option, since it cuts negotiation costs and increases transaction yields.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas **17**

Introdução **19**

1 Inteligência Artificial **21**

1.1 Sistemas que agem como Humanos 22

1.2 Sistemas que pensam como Humanos 22

1.3 Sistemas que pensam racionalmente 23

1.4 Sistemas que agem racionalmente 23

1.5 Fundamentos da IA 23

1.6 Breve Histórico 25

1.7 Estado da Arte da Inteligência Artificial 26

2 Inteligência Artificial Distribuída **28**

2.1 Resolução Distribuída de Problemas (RDP) 29

2.2 Sistemas Multiagentes 30

2.3 Simulação Multiagente 31

2.3.1 Análise de Sociedades Artificiais 31

2.3.2 Análise de Sistemas Sociais 32

2.3.2.1 Abordagem Fundacional 32

2.3.2.2	Abordagem Representacional	32
2.3.3	Prototipagem para Resolução	32
2.4	Resolução Multiagente	33
3	Sistemas Multiagentes	34
3.1	Definição de Agente	34
3.2	Ambientes dos Agentes	37
3.3	Definição dos Sistemas Multiagentes	38
3.4	Por que usar Sistemas Multiagentes	39
3.5	Características dos Ambientes Multiagentes	42
3.6	Comunicação entre Agentes	43
3.6.1	Características da Comunicação e Significado das Mensagens	44
3.6.2	Os Níveis de Comunicação	46
3.6.3	Tipos de Mensagens	46
3.6.3.1	A Teoria <i>Speech-Act</i>	47
3.6.4	Linguagens de Comunicação	48
3.7	Linguagem, <i>framework</i> e especificações para agentes	49
3.7.1	FIPA	49
3.7.2	JADE	53
3.7.2.1	A Arquitetura da Plataforma de Agente JADE	55
4	Comércio Eletrônico e Negociação em SMA	58
4.1	Comércio Eletrônico	58
4.1.1	Classificação do Comércio Eletrônico	59
4.1.2	Modelos do Comércio Eletrônico	62
4.1.2.1	O Modelo CBB	63
4.1.2.2	O Modelo BBT	65

4.1.3	O Mercado Brasileiro	67
4.2	Negociação	69
4.2.1	A Classificação da Negociação	69
4.2.2	Características e Parâmetros do Espaço de Negociação	70
4.2.2.1	Cardinalidade da Negociação	70
4.2.2.2	Características dos Agentes	70
4.2.2.3	Ambiente da Negociação	71
4.2.2.4	Parâmetros de Eventos	72
4.2.2.5	Parâmetros de Informação	72
4.2.2.6	Parâmetro de Alocação	72
5	A Aplicação SMANACE	73
5.1	O Mercado	73
5.2	O Modelo de Negociação	73
5.3	Os Protocolos de Negociação	75
5.3.1	O Protocolo referenciado por Farantin (1998)	75
5.3.2	O Protocolo CAMILO	77
5.3.3	A Evolução do Protocolo CAMILO	78
5.4	Táticas e Estratégias	81
5.4.1	Dependente do Tempo	82
5.5	A Plataforma e a Implementação	84
5.5.1	A Plataforma	84
5.5.2	A Implementação	85
5.5.2.1	A Função de Depreciação	86
5.5.2.2	A Função de Acréscimo	87
5.6	Simulações e Resultados	88
5.6.1	Comparando os Protocolos em Ambiente sem Concorrência	89

5.6.1.1	Primeira Comparação	90
	Primeiro Cenário:	90
	Segundo Cenário:	91
	Análise dos Resultados (Cenário 1 X Cenário 2):	92
5.6.1.2	Segunda Comparação	93
	Primeiro Cenário:	93
	Segundo Cenário:	95
	Análise dos Resultados (Cenário 1 X Cenário 2):	96
5.6.2	Comparando os Protocolos em Ambiente de Concorrência	97
5.6.2.1	Primeiro Cenário	97
	Configurando os Agentes:	97
	Resultado:	98
5.6.2.2	Segundo Cenário	98
	Configurando os Agentes:	99
	Resultado:	100
5.6.2.3	Terceiro Cenário	101
	Configurando os Agentes:	101
	Resultado:	102
5.6.2.4	Quarto Cenário	102
	Configurando os Agentes:	103
	Resultado:	104
5.6.2.5	Quinto Cenário	105
	Configurando os Agentes:	105
	Resultado:	106
5.6.2.6	Sexto Cenário	107
	Configurando os Agentes:	107

Resultado:	108
6 Considerações Finais	110
6.1 Conclusões	110
6.2 Sugestões para Trabalhos Futuros	111
Anexos	112
Anexo A - Início de uma Simulação	112
Anexo B - A Negociação	113
Anexo C - Resultado de uma Simulação	114
Anexo D - Exemplo das Trocas de Mensagens	115
Referências Bibliográficas	116

Lista de Figuras

2.1	Proposta de nova classificação para a IAD, segundo Marietto et al. (2002).	29
2.2	A grande área RDP (SICHMAN; ALVARES, 1997).	30
2.3	Fluxo operacional de uma simulação, segundo Marietto et al. (2002).	31
3.1	Taxonomia biológica (MANGAN, 2001).	37
3.2	Estrutura de um Sistema Multiagente (REIS, 2003).	40
3.3	Agente com habilidade de comunicação (REIS, 2003).	44
3.4	KQML - um protocolo de comunicação para agentes e programas (WEISS, 1999).	48
3.5	Partes da especificação FIPA2002.	50
3.6	Divisões da especificação FIPA <i>Agent Communication</i> .	50
3.7	Divisões da especificação FIPA Agente de Transporte da Mensagem.	51
3.8	A mensagem (FIPA, 2002).	52
3.9	Preparando a mensagem para o transporte (FIPA, 2002).	53
3.10	Duas Mensagens de Transporte para o mesmo agente (FIPA, 2002).	54
3.11	Criptografando o cabeçalho da Mensagem (FIPA, 2002).	55
3.12	Arquitetura de uma plataforma JADE.	56
4.1	Relação de 1:1.	61
4.2	Relação de 1:N.	61
4.3	Relação de M:N.	61
4.4	Relação de 1:B:N.	62
4.5	Relação de M:B:B:N.	62
4.6	Modelo CBB (VETTER; PITSCH, 2000).	63

4.7	Modelo BBT (HE; JENNINGS; LEUNG, 2003).	66
5.1	Protocolo Farantin para compradores e vendedores.	76
5.2	Exemplo de interação entre dois agentes.	77
5.3	Protocolo CAMILO para agentes compradores (Versão 1).	79
5.4	Protocolo CAMILO para agentes vendedores (Versão 1).	80
5.5	Protocolo CAMILO para agentes compradores.	81
5.6	Protocolo CAMILO para agentes vendedores.	82
5.7	Relação de $\alpha(t)$ com o $\frac{t}{t_{max}}$ na família polinomial (FARANTIN, 1998).	84
5.8	Relação de $\alpha(t)$ com o $\frac{t}{t_{max}}$ na família exponencial (FARANTIN, 1998).	85
5.9	Primeiro cenário da primeira comparação.	91
5.10	Segundo cenário da primeira comparação.	93
5.11	Primeiro cenário da segunda comparação.	94
5.12	Segundo cenário da segunda comparação.	96
5.13	Gráfico referente ao primeiro cenário.	99
5.14	Gráfico referente ao segundo cenário.	100
5.15	Gráfico referente ao terceiro cenário.	103
5.16	Gráfico referente ao quarto cenário.	104
5.17	Gráfico referente ao quinto cenário.	107
5.18	Gráfico referente ao sexto cenário.	109

Lista de Tabelas

1	Grupos de definições da Inteligência Artificial.	21
2	Propriedade de agentes (MANGAN, 2001).	34
3	Propriedade X Conceitos (MANGAN, 2001).	36
4	Características do Ambiente Multiagente (WEISS, 1999).	42
5	Características do Ambiente pelo Agente (WEISS, 1999).	43
6	Agentes e suas Capacidades (WEISS, 1999).	47
7	Tipos de mensagens provindas da <i>Speech-Act</i> (WEISS, 1999).	47
8	Técnicas de Filtragem para a “Procura do Produto” no comércio eletrônico (HE; JENNINGS; LEUNG, 2003).	64
9	Comparação entre as técnicas de “Procura do Produto” (HE; JENNINGS; LEUNG, 2003).	64
10	Continuação da Comparação entre as técnicas de “Procura do Produto” (HE; JENNINGS; LEUNG, 2003).	64
11	Agentes Mediadores X CBB (FLOREA, 2002).	65
12	Configuração do Agente Comprador 11c1.	90
13	Configuração do Agente Vendedor 11v1.	90
14	Configuração do Agente Vendedor 11v2.	91
15	Configuração do Agente Comprador 12c1.	92
16	Configuração do Agente Vendedor 12v1	92
17	Configuração do Agente Vendedor 12v2	92
18	Configuração do Agente Comprador 21c1	93
19	Configuração do Agente Comprador 21c2	94

20	Configuração do Agente Vendedor 21v1	94
21	Configuração do Agente Comprador 22c1.	95
22	Configuração do Agente Comprador 22c2.	95
23	Configuração do Agente Vendedor 22v1.	95
24	Configuração do Agente Comprador 1c1.	97
25	Configuração do Agente Comprador 1c2.	97
26	Configuração do Agente Vendedor 1v1.	98
27	Configuração do Agente Vendedor 1v2.	98
28	Configuração do Agente Comprador 2c1.	99
29	Configuração do Agente Comprador 2c2.	99
30	Configuração do Agente Vendedor 2v1.	100
31	Configuração do Agente Vendedor 2v2.	100
32	Configuração do Agente Comprador 3c1.	101
33	Configuração do Agente Comprador 3c2.	101
34	Configuração do Agente Vendedor 3v1.	102
35	Configuração do Agente Vendedor 3v2.	102
36	Configuração do Agente Comprador 4c1.	103
37	Configuração do Agente Comprador 4c2.	103
38	Configuração do Agente Vendedor 4v1.	104
39	Configuração do Agente Vendedor 4v2.	104
40	Configuração do Agente Comprador 5c1.	105
41	Configuração do Agente Comprador 5c2.	106
42	Configuração do Agente Vendedor 5v1.	106
43	Configuração do Agente Vendedor 5v2.	106
44	Configuração do Agente Comprador 6c1.	107
45	Configuração do Agente Comprador 6c2.	108

46	Configuração do Agente Vendedor 6v1.	108
47	Configuração do Agente Vendedor 6v2.	108

Lista de Siglas

ACC: *Agent Communication Channel* (Canal de Comunicação para Agente).

ACL: *Agent Communication Language* (Linguagem de Comunicação para Agente).

AMS: *Agent Management System* (Sistema gestor de agente).

B2B: *business-to-business* (Empresa para empresa).

B2C: *business-to-consumer* (Empresa para consumidor).

BBT: *Business to Business Transaction* (Trasacção entre empresas).

C2C: *consumer-to-consumer* (Consumidor para consumidor).

CBB: *Consumer Buying Behavior* (Modelo de comportamento do comprador).

DARPA: *Defense Advanced Research Projects Agency*.

DF: *Directory Facilitator* (Diretório facilitador).

FIPA: *Foundation for Intelligent Physical Agents*.

GUID: *Globally Unique Identifier* (Identificador único e global).

IA: Inteligência Artificial.

IAD: Inteligência Artificial Distribuída.

JADE: *Java Agent Development Framework*.

JVM: *Java Virtual Machines* (Máquina virtual java).

KIF: *Knowledge Interchange Format*.

KQML: *Knowledge and Query Manipulation Language*.

KSE: *Knowledge Sharing Effort*.

RDP: Resolução Distribuída de Problemas.

RMA: *Remote Monitoring Agent* (Monitoramento remoto do Agente).

RMI: *Remote Method Invocation* (Chamada remota de métodos).

SMA: Sistema Multiagente.

SMANACE: Sistema Multiagente de Negociação Automatizada para Comércio Eletrônico.

Introdução

Com a necessidade de melhorias dos *softwares* existentes, tornando-os mais inteligentes, autônomos, reduzindo ou anulando a interferência humana, é que a área de Sistemas Multiagentes (SMA), inserida na Inteligência Artificial Distribuída (IAD), vem se motivando.

Analisando os vários ambientes, concorrentes ou cooperativos, em que os Sistemas Multiagentes propõem automação, pode-se identificar um elemento em comum, a comunicação entre as partes. Esta interação é fundamental para a sobrevivência e eficiência do sistema, pois possibilita a troca de informações ou intenções entre os agentes.

Quando a interação tem o objetivo de distribuir as tarefas (ambientes cooperativos) ou de sincronizar objetivos (ambientes concorrentes), recebe o nome de negociação. Das várias aplicações existentes, que utilizam da negociação, uma das mais dependentes é a transação comercial, pois as partes envolvidas defendem posições, na maioria das vezes, concorrentes e necessitam de mecanismo para ajustar interesses e alcançar o acordo. As transações comerciais são representadas no mundo virtual pelo Comércio Eletrônico e é por isso que este trabalho optou por tê-lo como aplicação para simulações e comprovações das inovações sugeridas no campo da negociação automatizada para agentes de *software*.

Reconhecendo a importância desta linha de pesquisa, o foco principal deste trabalho foi propor um novo protocolo de negociação automatizada para agentes em aplicações de Comércio Eletrônico. Este protocolo, que teve como referência o protocolo referenciado por Farantin (1998), tem o objetivo de melhorar as utilidades dos participantes, fazendo com que compradores e vendedores tenham mais retorno nas negociações em que obtiveram acordo. Para a execução das simulações e as comparações, foi desenvolvido o *software* SMANACE (Sistema Multiagente de Negociação Automatizada para Comércio Eletrônico). Com ele os usuários podem delegar intenção de compra ou venda para os agentes de *software* que negociam de forma autônoma segundo os protocolos e um modelo de negociação multilateral (várias negociações simultâneas) e multi-dimensão (vários atributos como preço, prazo de entrega e forma de pagamento).

Este trabalho discorre sobre alguns temas como: Inteligência Artificial, Sistema Mul-

tiagente, Comércio Eletrônico e Negociação Automatizada. Estes temas estão divididos em capítulos sendo:

- Capítulo 2: Conceito, História e Estado da Arte da Inteligência Artificial
- Capítulo 3: Conceito, Divisões e Subdivisões da Inteligência Artificial Distribuída.
- Capítulo 4: Definições e características de agentes e as sociedades de agentes, também chamadas de Sistemas Multiagentes.
- Capítulo 5: Uma visão do Comércio Eletrônico, desde as classificações até o mercado Brasileiro. Está presente neste capítulo um fundamentação teórica da ferramenta fundamental para as interações entre agentes, a negociação.
- Capítulo 6: Por último, e não menos importante, a aplicação SMANACE que demonstra os resultados dos desafios propostos através dos vários cenários criados.
- Capítulo 7: Apresenta as considerações finais mais importantes e direciona os trabalhos futuros.

1 *Inteligência Artificial*

Há milhares de anos, tenta-se entender como o homem pensa. Estudos e mais estudos são desenvolvidos para que se tenha um conhecimento mais profundo sobre a forma de pensar e a inteligência humana. A Inteligência Artificial (IA) usufrui destes estudos para desenvolver outros novos, que por sua vez terão não só características teóricas, mas também práticas. Ou seja, a Inteligência Artificial não só estuda a inteligência como também tenta implementar esta inteligência (capacidade de raciocínio, comunicação de linguagem natural, aprendizado, etc.).

A Inteligência Artificial é uma ciência muito recente, que desde o início, após a Segunda Guerra Mundial, vem gerando estudos com grande valia para a sociedade de uma forma geral. Talvez por ser recente, não existe uma concordância no que diz respeito à definição de Inteligência Artificial.

Analisando algumas definições e opiniões, chegou-se a um agrupamento de idéias como mostra a Tabela 1.

Tabela 1: Grupos de definições da Inteligência Artificial.

	Humano	Racional
Pensar	Sistemas que pensam como humanos	Sistemas que pensam racionalmente
Agir	Sistemas que agem como humanos	Sistemas que agem racionalmente

As definições fazem parte de duas principais dimensões. A primeira dimensão é a do pensamento e raciocínio e contempla as definições de dois grupos: sistemas que pensam como humanos e sistemas que pensam racionalmente. A outra dimensão, focada no comportamento, contempla as definições de outros dois grupos: sistemas que atuam como humanos e sistemas que atuam racionalmente (RUSSELL; NORVIG, 2002).

1.1 Sistemas que agem como Humanos

Esta linha está basicamente fundamentada pelo Teste de Turing (RUSSELL; NORVIG, 2002), que foi proposto por Alan Turing em 1950. O teste foi projetado para dizer se uma máquina é ou não inteligente. O teste Consiste em separar, em salas distintas, um interrogador humano de candidatos que poderiam ser humanos ou máquinas, sendo que a única forma de comunicação entre eles é através da escrita. Caso o interrogador não consiga distinguir se está falando com uma máquina ou com um humano, então a máquina passa no teste e é considerada inteligente. Para passar neste teste a máquina precisaria de algumas habilidades como:

- Processamento de linguagem natural: para garantir uma boa comunicação em um idioma;
- Representação de conhecimento: para armazenar o que se sabe;
- Raciocínio automatizado: para utilizar de informações armazenadas e ser capaz de formular novas conclusões;
- Aprendizado de máquina: o que garantiria melhor adaptação às novas circunstâncias, além de identificar padrões e extrapolar.

Estas habilidades aumentam no “Teste de Turing Total”, onde as percepções visuais e físicas são acrescentadas. Portanto, faz-se necessário o acréscimo de mais duas disciplinas:

- Visão computacional: para perceber objetos;
- Robótica: para manipular objetos e movimentar-se.

1.2 Sistemas que pensam como Humanos

Para modelar sistemas que pensam como humanos é necessário saber como os humanos pensam. Após se ter uma teoria suficientemente precisa da mente, é possível expressar a teoria em um programa de computador. Para esta difícil tarefa, a interdisciplinar “Ciência Cognitiva” une modelos computacionais da IA com técnicas experimentais da psicologia. O GPS (*General Problem Solver*), desenvolvido por Allan Newell e Herbert Simon em 1961 (RUSSELL; NORVIG, 2002), é um exemplo de aplicação que não se preocupa somente em resolver problemas, mas se concentra na comparação das etapas de raciocínio. Ou seja, o ideal seria fazer uma máquina que seguisse os mesmos passos dos humanos.

1.3 Sistemas que pensam racionalmente

Esta abordagem também estuda o pensamento, porém, não focada no raciocínio humano. O foco é o “pensamento correto”, isto é, processos de raciocínio irrefutáveis. O primeiro a tentar codificar este pensamento foi o filósofo grego Aristóteles, que através dos seus silogismos, forneceu padrões para estruturas de argumentos que sempre resultavam em conclusões corretas ao receberem premissas corretas. Aristóteles, com o estudo da lei do pensamento, deu início ao campo chamado lógica.

1.4 Sistemas que agem racionalmente

Segundo Weiss (1999), esta é a abordagem mais completa, pois utiliza de vários mecanismos para alcançar a racionalidade, que não somente as “leis do pensamento”, além de ser mais acessível ao desenvolvimento científico, em relação às estratégias baseadas em comportamento ou pensamento humano, por ter padrão de racionalidade definido com clareza.

A racionalidade perfeita não é algo viável quando se trata de ambientes complexos, pois as demandas computacionais são demasiadamente elevadas.

1.5 Fundamentos da IA

Algumas disciplinas foram e são importantes para o desenvolvimento da IA, pois contribuíram e contribuem com idéias, pontos de vista e técnicas (RUSSELL; NORVIG, 2002).

As provas da ajuda destas disciplinas são algumas respostas a diversas perguntas. A Filosofia contribui com a IA desde 428 a.C. e ajudou a responder algumas perguntas como:

- Regras formais podem ser usadas para obter conclusões válidas?
- Como a mente se desenvolve a partir de um cérebro físico?
- De onde vem o conhecimento?
- Como o conhecimento conduz à ação?

A Matemática, desde 800 d.C. até a atualidade, forneceu as ferramentas para manipular declarações de certeza lógica, bem como declarações incertas e probabilísticas, além de ajudar a responder algumas perguntas como:

- Quais são as regras formais para obter conclusões válidas?
- O que pode ser computado?
- Como se raciocina com informações incertas?

A Economia, que formulou o problema da tomada de decisão maximizando os resultados esperados, também ajudou a responder algumas perguntas (de 1776 até a atualidade) como:

- Como se deve tomar decisões para maximizar o lucro?
- Como se deve fazer isso quando outros não podem nos acompanhar?
- Como se deve fazer isso quando o lucro pode estar distante do futuro?

A Neurociência, estudo do sistema nervoso, em particular do cérebro, também ajuda a IA desde 1861 até a atualidade. Ajudando a entender como o cérebro processa informações, a Neurociência contribuiu muito para a evolução das redes neurais.

Como os seres humanos e os animais pensam e agem? Esta pergunta é um exemplo da conexão da Psicologia e a IA, pois com a visão da psicologia cognitiva, de que os seres humanos e animais são máquinas de processamento de informações, a IA tenta estudar o dispositivo responsável por esse processamento, o cérebro, e os comportamentos para poder tirar conhecimentos.

A Engenharia de Computadores (de 1940 até a atualidade) tem um papel fundamental de tornar viáveis os experimentos, fornecendo recursos computacionais suficientes para aplicações de IA que exigem muito.

As Teorias de Controle e Cibernética (de 1948 até atualidade) ajudam a colocar os resultados das aplicações de IA em mundo real, construindo robôs capazes de atuar no nosso dia-a-dia. A autonomia dos agentes é outra área de atuação desta disciplina de interesse da IA.

A Linguística contribuiu com a IA desde 1957, e também ajudou a responder perguntas como:

- Como a linguagem se relaciona ao pensamento?

1.6 Breve Histórico

Quando se fala em histórico da Inteligência Artificial, uma das mais lembradas referências é Russell e Norvig (2002), pela objetividade. Seguem pontos importantes desta referência.

É uma ciência nova que teve sua gestação entre os anos de 1943 e 1955, sendo que o primeiro trabalho reconhecido foi o de Warren McCulloch e Water Pitts, que tinham como parâmetro o cérebro humano. No entanto, somente no verão de 1956 surge o nome Inteligência Artificial, após um *workshop* entre grandes nomes como: John McCarthy, Arthur Samuel (IBM), Ray Solomonoff e Oliver Selfridge (MIT).

De 1952 a 1969 houve um entusiasmo em torno da área, que até então não era uma ciência, trazendo uma boa expansão e interesses de pessoas que conheciam as aplicações demonstradas na época. O entusiasmo foi tão grande, que muitos exageraram, iniciando projetos ambiciosos, como foi o caso de Newell e Simon com o *General Problem Solver* (GPS), que procurava imitar o protocolo de resolução de problemas dos humanos. Tanta euforia causou no período seguinte, de 1966 até 1973, doses de realidade que diminuiriam bastante o entusiasmo e as grandes ambições. Problemas como a mal sucedida tradução de Russo para o Inglês, que não foi solucionado devido a ambigüidades e falta de conhecimento do contexto aplicado ao texto a ser traduzido, trouxeram para os pesquisadores uma boa amostra das dificuldades que existiam e as que viriam a existir.

Surgem então projetos mais reais que já não visualizavam a resolução de todos os problemas, mas sim de problemas específicos, diminuindo assim a complexidade. Foi nesta época também, de 1969 a 1979, que surgiram, para solucionar alguns problemas, os sistemas baseados em conhecimento. Um bom exemplo, e o primeiro, foi o DENDRAL, um programa desenvolvido em Stanford para resolver o problema de inferência da estrutura molecular, tendo como base informações providas de um espectômetro de massa.

Em 1980 a IA começa a ser utilizada na indústria, promovendo cortes de despesas e otimização de tarefas. Foi sem dúvida uma entrada triunfal com o R1, o primeiro sistema especialista (*expert system*) de sucesso que iniciou suas atividades na Digital Equipment Corporation, proporcionando uma economia de 40 milhões de dólares por ano. Sem dúvida a indústria é um dos maiores consumidores desta tecnologia.

Após um período de abandono, na década de 70, a rede neural, um campo da Inteligência Artificial, teve retorno em 1986 pelas mãos de alguns cientistas como: John Hopfield, David Rumelhart, Geoff Hinton e principalmente por quatro grupos diferentes, que nos anos 80 reinventaram o algoritmo de aprendizagem *back-propagation*, produzido em 1969 por Bryson e Ho.

Em 1987 a IA torna-se uma ciência; um processo que veio se desenvolvendo, fazendo da área um campo de pesquisa com metodologias e conteúdos próprios. Definitivamente foi um processo conturbado, onde a ciência buscava a sua identidade; às vezes cometendo excessos na tentativa de separá-la da computação, todavia chegando ao equilíbrio satisfatório de complemento entre as ciências existentes.

A partir de 1995 surge, impulsionado pelo progresso da resolução de subproblemas da IA, um movimento com interesse nos agentes inteligentes. Um dos mais importantes ambientes para os agentes inteligentes é a *internet*. E aproveitando este “caldo de cultura”, a IA é elemento fundamental nas mais diversas ferramentas disponíveis na *internet*; como é o caso dos motores de busca e sistemas de recomendações, além de outros .

1.7 Estado da Arte da Inteligência Artificial

Como está a IA? Qual o nível de evolução? O que ela é capaz de fazer?

Para responder estas perguntas, nada melhor do que as provas concretas da evolução da IA e suas aplicações. Na prática, é possível perceber o quanto evoluíram e quanto faltam evoluir a IA e a IAD (Inteligência Artificial Distribuída).

Apesar da IA atuar em muitos campos e sub-campos, seguem algumas aplicações citadas em Russell e Norvig (2002).

A NASA colocou para operar, a 100 milhões de milhas da Terra, um programa chamado *Remote Agent*, capaz de fazer planejamento autônomo e agendamento, ou seja, o programa é capaz de automatizar o planejamento do programa que controla as operações da nave espacial. O programa gera planos e monitora a nave para certificar que os mesmos foram executados.

Outra aplicação, já conhecida, é o *Deep Blue* da IBM, que foi o primeiro programa de computador a ganhar uma partida de xadrez de um campeão mundial de xadrez (Garry Kasparov). Após uma das partidas, Kasparov disse: “Eu senti um novo tipo de Inteligência”.

ALVINN é uma aplicação de IA, classificada como sistema de visão computacional, capaz de guiar um carro. A aplicação foi executada nas estradas dos Estados Unidos da América e teve resultados considerados surpreendentes. De um total de 2.850 milhas, o *software* foi capaz de controlar em 98% do tempo, ficando apenas 2% de controle humano. Câmeras transmitiam imagens das rodovias e o *software* determinava a direção, baseado em experiência de treinamentos anteriores.

HipNav é o sistema utilizado em microcirurgia, que usa a técnica de visão computacional para criar um modelo tridimensional de partes anatômicas internas, além de usar controles robóticos para guiar as incisões nos pacientes (RUSSELL; NORVIG, 2002).

Mais recentemente algumas experiências vêm dando resultado. É o caso do CogVis (*Cognitive Vision*) que foi desenvolvido por um consórcio (LEEDS, 2005) chefiado pela Universidade de Leeds e formado pelas instituições:

- Instituto de Tecnologia Royal (Royal Institute of Technology - KTH), Stockholm, Suécia;
- Universidade de Hamburg (Hamburg University - CSL), Hamburg, Alemanha;
- Instituto para biologia cirbenética Max-Planck (MPIK), Tuebingen, Alemanha;
- Universidade de Leeds, Leeds, Reino Unido;
- Universidade de Estudos de Genova (Departamento de Informática - DIST), Genova, Itália;
- Instituto Federal Suíço de Tecnologia (ETH), Zurich, Suíça;
- Universidade de Ljubljana, Ljubljana, Eslovênia.

O sistema foi capaz de aprender regras de jogos infantis, utilizando programação indutiva lógica, observando partidas entre humanos. Após a fase de treinamento o sistema foi capaz de jogar sem interferência humana (LEEDS, 2005; SBPC, 2005).

2 Inteligência Artificial Distribuída

Em meados de 1970, a Inteligência Artificial Distribuída (IAD) evoluiu e diversificou-se rapidamente. Hoje em dia é um campo de pesquisa já estabelecido e muito promissor que concentra idéias de muitas disciplinas como:

- Inteligência Artificial;
- Ciências da Computação;
- Sociologia;
- Economia;
- Ciência da Organização e Gerência;
- Filosofia.

Segundo Weiss (1999), a Inteligência Artificial Distribuída é o estudo, construção e aplicação de Sistemas Multiagentes, de sistemas com graus de interação em que os agentes (definido no item 3.1) perseguem um conjunto de objetivos ou fazem um conjunto de tarefas.

Para Demazeau e Müller (1990), a Inteligência Artificial Distribuída é solução colaborativa de problemas globais por um grupo distribuído de entidades. Assim sendo, estas entidades, que podem estar física ou geograficamente dispersas, devem compartilhar as informações a fim de atingir um objetivo global.

Outro item muito discutido é a forma de divisão da Inteligência Artificial Distribuída. Para Sichman e Alvares (1997), a IAD é dividida em duas grandes áreas: Resolução Distribuída de Problemas (RDP) e Sistemas Multiagentes (SMA). Mas, no entanto, com o avanço da IAD novos modelos estão sendo propostos, como pode-se identificar em Marietto et al. (2002) (ver Figura 2.1).

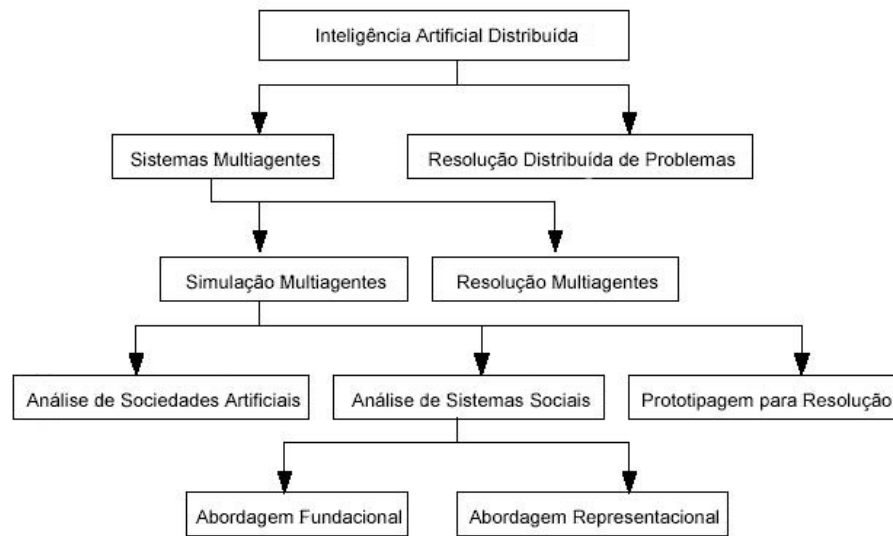


Figura 2.1: Proposta de nova classificação para a IAD, segundo Marietto et al. (2002).

Para Marietto et al. (2002), a forma geral em que foi dividida a IAD não mais retrata a atual diversidade de trabalhos de pesquisa e desenvolvimento em IAD, podendo-se hoje encontrar outras sub-áreas razoavelmente estruturadas e consolidadas, como por exemplo a Simulação Baseada em Multiagente.

Serão apresentadas, nas seções seguintes, as duas grandes áreas da IAD (Resolução Distribuída de Problemas e Sistemas Multiagentes), relatadas em Sichman e Alvares (1997), e as subdivisões propostas no Marietto et al. (2002).

2.1 Resolução Distribuída de Problemas (RDP)

Nesta grande área, os agentes são direcionados para resolver um problema em particular, dentro de uma concepção fechada do domínio a que se refere o problema. Ou seja, os agentes são projetados para resolver apenas um tipo específico de problema, e mesmo que haja similaridade com outros problemas, não podem ser utilizados para resolver estes outros. O projetista define o número de agentes, que não varia, e dá a cada agente uma visão específica e incompleta do problema. Desta forma, para a resolução de um problema os agentes devem cooperar entre si, compartilhando conhecimento sobre o problema e sobre o processo de obter uma solução.

A abordagem utilizada pela Resolução Distribuída de Problemas é a *top-down* (descendentes), ou seja, o projetista primeiramente realiza uma análise do problema a ser resolvido e, então, identifica os agentes necessários para atuarem conforme o esquema

estruturado para a solução desse problema. Desta forma, a tarefa de resolução será decomposta entre os vários agentes, buscando melhorar o processamento do sistema através da execução paralela. Fica claro que grande parte do raciocínio sobre a solução é inserido no sistema pelo projetista, o que segundo Marietto et al. (2002), normalmente leva ao desenvolvimento de controles geralmente hierárquicos e centralizados.

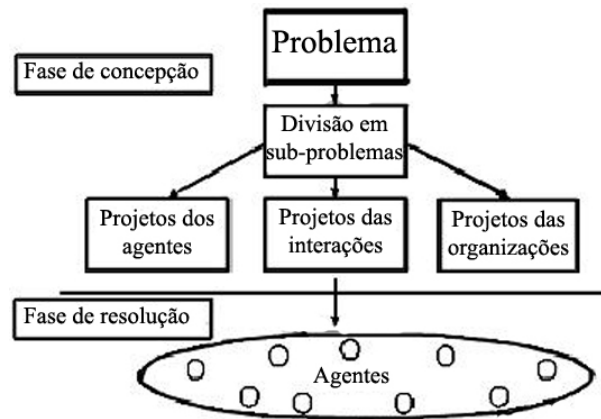


Figura 2.2: A grande área RDP (SICHMAN; ALVARES, 1997).

2.2 Sistemas Multiagentes

Ao contrário do RDP, os Sistemas Multiagentes têm um foco na estruturação do agente e não na estruturação do problema, o que dá uma flexibilidade para os agentes que são capazes de resolver mais de um problema. Isto leva a uma característica de *button-up* (ascendente) no desenvolvimento do sistema, pois o projetista preocupa-se em desenvolver arquiteturas de agentes que interajam de forma autônoma e social.

É notável que, em relação a RDP, há um aumento significativo da complexidade no desenvolvimento de Sistemas Multiagentes que disponibilizam suporte a modificações ambientais. A complexidade aumenta pois os sistemas precisam ter, além de outras características, agentes autônomos que detenham mecanismos de comunicação bem organizados para prover a interação.

Este será assunto do próximo capítulo, onde serão discutidas algumas características dos Sistemas Multiagentes.

2.3 Simulação Multiagente

Formada pela intersecção das áreas de IAD e Simulação Computacional, esta classe tem como objetivo o estudo da simulação e do modelo simulado.

A simulação ocorre quando há uma transposição de um sistema alvo para um modelo conceitual equivalente que será codificado em um modelo computacional.

Este processo tem como ponto fundamental a garantia de que o modelo computacional desenvolvido é o mais fiel possível do sistema alvo. Para que a fidelidade aconteça dois processos são executados: verificação e validação, como mostra a Figura 2.3.

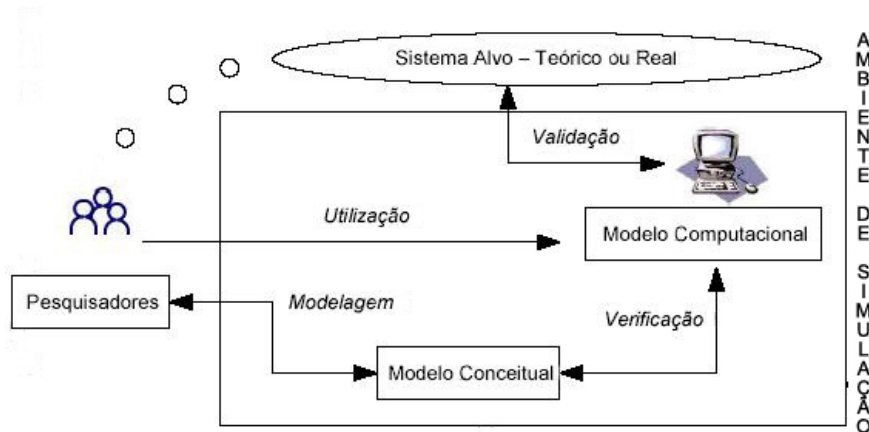


Figura 2.3: Fluxo operacional de uma simulação, segundo Marietto et al. (2002).

O processo de verificação garante que o modelo conceitual foi transcrito corretamente para o modelo computacional. Já o processo de validação visa assegurar que o modelo computacional representa o sistema alvo com aceitável grau de fidelidade.

Esta classe Simulação de Multiagente é delimitada em três classes interdependentes: Análise de Sociedades Artificiais, Análise de Sistemas Sociais e Prototipagem para Resolução. Nas subseções seguintes dar-se-á uma breve descrição para cada uma das três classes acima, segundo Marietto et al. (2002).

2.3.1 Análise de Sociedades Artificiais

Nesta classe o sistema alvo não se restringe a sistemas reais de sociedades e de ambientes físicos, o que permite ao pesquisador explorar qualquer relação física, social, psicológica e econômica conhecida e adotada.

Apesar de não se restringir a sistemas reais, as aplicações desta classe são desenvol-

vidas com o propósito de responder questões específicas, bem como explorar fenômenos particulares.

2.3.2 Análise de Sistemas Sociais

A estruturação desta classe provém da interação entre Ciências Naturais, Sociais e Simulação Multiagente. Esta estrutura é utilizada para modelar fenômenos sociais em geral, desde que sejam existentes ou que haja provas da sua existência. A abstração das simulações da classe de Análise de Sistemas Sociais segue duas direções: Abordagem Fundacional e Abordagem Representacional.

2.3.2.1 Abordagem Fundacional

Nesta direção as simulações estão voltadas para uma abordagem mais formal e abstrata que procura construir e representar fundamentos da teoria social. Os pesquisadores desta área podem atuar em duas dimensões:

- propor novas estruturas ou alternativas para sistemas sociais, verificando a viabilidade de sua existência e funcionamento;
- obter um melhor entendimento dos fundamentos sociais, antropológicos, psicológicos, etc., que subsidiam e direcionam sistemas sociais.

2.3.2.2 Abordagem Representacional

Esta direção não aborda construção de estruturas teóricas de sistemas sociais, mas as utiliza para modelar processos sociais e institucionais. Alguns destes processos podem ser identificados como fenômenos sociais, como por exemplo o desenvolvimento de características culturais ou efeitos de mudanças em estruturas organizacionais.

2.3.3 Prototipagem para Resolução

Nesta classe os sistemas alvo são sistemas reais em que os agentes interagem com ambientes também reais. Os objetivos principais destes sistemas são eminentemente práticos.

Portanto, os sistemas desta classe podem ser aplicados para:

- avaliar os sistemas modelados para posterior aplicação real;

- treinar futuros usuários do sistema, suporte à tomada de decisão;
- enfatizar a predição de comportamentos e de ações futuras.

2.4 Resolução Multiagente

Esta classe dedica-se a desenvolver aplicações baseadas em formalismos de SMA (definido na Seção 3.3), bem como para o estudo da infra-estrutura teórica-operacional destes sistemas. Ou seja, tem uma abordagem mais computacional dos aspectos sociais dos sistemas computacionais.

3 *Sistemas Multiagentes*

Os Sistemas Multiagentes (SMA) estão voltados para aplicações onde o foco são sistemas que utilizam agentes em sociedade. Ou seja, que haja interações entre agentes autônomos cientes da própria existência e da existência de outros agentes. É um campo novo, pois teve início nos anos 80. Porém, só em meados dos anos 90, com o crescimento das investigações e, por consequência, aumento nas publicações (revistas, livros), ganhou uma notoriedade.

Mas antes de discutir sobre os Sistemas Multiagentes, é necessário discutir sobre a unidade destes sistemas, o agente.

3.1 Definição de Agente

Apesar da importância que este elemento representa, não existe um consenso, na comunidade científica, sobre o significado de um agente.

As diferenças, na maioria das vezes, estão nas funcionalidades que os agentes implementam, e no contexto no qual são aplicados. O que para alguns é uma funcionalidade indispensável para que um objeto seja considerado agente, para outros é uma funcionalidade desejável mas não essencial. Para se tentar entender a teoria defendida por cada pesquisador, primeiro serão verificadas as principais características que um agente pode ter (ver Tabela 2).

Tabela 2: Propriedade de agentes (MANGAN, 2001).

Propriedade	Outros Nomes	Significado
reativo	<i>sensing and acting</i>	duas definições mais usadas: (a) responde às mudanças no ambiente após um período de tempo adequado. (b) toma ações sem consultar seu estado interno.

autônomo		possui controle sobre suas próprias ações.
orientado a objetos	pró-ativo com propósito	não age simplesmente em resposta ao ambiente, mas o faz a fim de atingir um determinado objeto.
temporalmente contínuo		é um processo que executa continuamente.
comunicativo	sociável	comunica-se com outros agentes, talvez incluindo pessoas.
aprendiz	adaptativo	muda seu comportamento baseando-se em suas experiências prévias.
móvel		possui a habilidade de transportar a si mesmo de uma máquina para outra.
flexível		suas ações não são rigidamente estabelecidas.
persistente	operação contínua	esta propriedade serve para caracterizar o ciclo de vida do agente e devem ter associadas a esta os estados possíveis do agente.
cooperativo		agentes podem ignorar, cooperar ou competir com outros agentes. Cooperação ocorre quando os agentes combinam seus esforços para alcançar um objetivo comum.
personagem		possui uma “personalidade”.

O conceito sobre agente é motivo de muito debates no meio acadêmico. A consequência destes debates são alguns conceitos:

- Russell e Norvig (RUSSELL; NORVIG, 2002): um agente é qualquer coisa que possa perceber o ambiente, através de sensores, e agir no ambiente, através de efetadores;
- Wooldridge e Jennings (MANGAN, 2001): sistemas autônomos, reativos, pró-ativos e com habilidades sociais;
- Brustoloni (MANGAN, 2001): agentes autônomos são sistemas autônomos com um propósito, ou seja, que tenha um objetivo a ser efetuado;
- *Oxford Dictionary of Computing* (MANGAN, 2001): o agente é um sistema autônomo que recebe informação do seu ambiente, processa e efetua uma ação no ambiente. Agentes de *softwares* atuam em ambientes simbólicos, percebendo e agindo de forma também simbólica. Um robô é um exemplo de agente que percebe o ambiente físico, através de sensores, e age através de efetadores;

- Franklin e Graesser (MANGAN, 2001): um agente autônomo é um sistema situado dentro do ambiente que o mesmo percebe e atua sempre buscando cumprir sua agenda.

De posse destas principais características dos agentes, Mangan (2001) montou uma tabela (Tabela 3) para diferenciar as várias definições, tendo como base alguns dos conceitos sobre agente.

Tabela 3: Propriedade X Conceitos (MANGAN, 2001).

Propriedades	Russell e Norvig	Wooldrige e Jennings	Brustoloni	Oxford Dictionary	Franklin e Graesser
reatividade	#	X	#	#	#
autonomia		X	X	X	X
orientação a objetivos		X	X		#
continuidade temporal					
comunicabilidade		X			
aprendizado					
mobilidade					
flexibilidade					
personalidade					
percepção	X				X
distribuição		X			

X: utiliza o conceito com esta denominação na definição.

#: utiliza conceito sem usar o termo ou usando sinônimo.

Além das definições apresentadas, existe outra importante definição feita por Weiss (1999): Agente é um sistema de computador que está situado em algum ambiente e que é capaz de ações autônomas, neste ambiente, a fim de atingir o seu objetivo.

Outra definição importante é a de agente inteligente, que também é uma definição sem consenso por envolver duas definições sem consenso: a de agentes, como foi dissertado anteriormente, e a definição de inteligência, que é motivo de pesquisa e muita discussão de várias ciências.

Uma definição muito utilizada é a de Weiss (1999): o agente é inteligente quando ele

é capaz de flexibilizar o seu comportamento autônomo. E para Weiss, esta flexibilização tem três significados:

- **Reatividade:** capacidade de perceber o ambiente e responder a tempo às mudanças ocorridas na tentativa de satisfazer os objetivos traçados.
- **Pró-Atividade:** exercer um comportamento direcionado ao objetivo, tomando a iniciativa para satisfazer os objetivos traçados.
- **Habilidade social:** Capacidade de interagir com outros agentes (e possivelmente humanos) para satisfazer ou atingir os objetivos traçados.

Uma maneira eficaz de classificar um agente é através de uma taxonomia. Uma taxonomia muito usada na literatura é a Biológica de Franklin e Graesser, que apesar de algumas críticas, permite uma visualização ampla do conceito de agente, mostrando que os agentes não se restringem a sistemas de computadores.

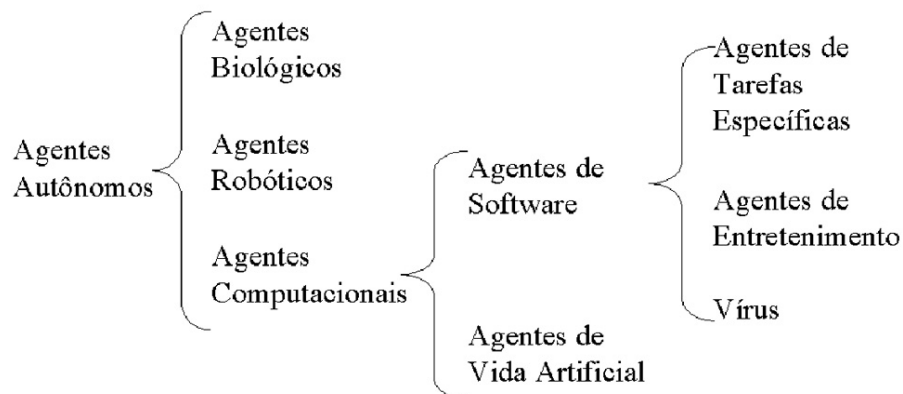


Figura 3.1: Taxonomia biológica (MANGAN, 2001).

3.2 Ambientes dos Agentes

Um outro elemento muito importante é o ambiente em que os agentes são aplicados. É baseado nestes ambientes que os agentes irão reagir e tentar alcançar os seus objetivos. Encontra-se em Russell e Norvig (2002) uma forma de classificar estes ambientes:

- **Totalmente observável *versus* parcialmente observável:** um ambiente é totalmente observável se o ambiente possibilitar aos sensores dos agentes, acesso ao estado completo do ambiente;

- Determinístico *versus* Estocástico: um ambiente é determinístico se o próximo estado é completamente determinado pelo estado corrente e pelas ações selecionadas pelo agente;
- Episódico *versus* Seqüencial: em ambiente episódico, a experiência do agente é dividida em episódios. Cada episódio consiste de uma percepção seguida de uma ação;
- Estático *versus* Dinâmico: se o ambiente pode modificar enquanto o agente atua, então ele é dinâmico;
- Discreto *versus* Contínuo: para um número limitado, distinto e claramente definido de percepções e ações, diz-se que o ambiente é discreto;
- Mono-Agente *versus* Multiagente: ambientes Multiagentes possuem mais de um agente atuando.

3.3 Definição dos Sistemas Multiagentes

Algumas considerações são feitas por Sichman e Alvares (1997) e Demazeau (1995) para os Sistemas Multiagentes. Rocha, Elmo e Alves (2003) citam estas considerações:

- devem ser capazes de decompor as tarefas baseando-se no conhecimento que eles possuem de si próprios e dos outros agentes;
- por serem autônomos, podem possuir metas próprias e decidir o que fazer, a qualquer momento;
- possuem capacidade para resolver seus problemas e os problemas que surgirem no ambiente;
- podem entrar e sair do ambiente a qualquer momento. Portanto, em um SMA os agentes devem ser capazes de modificar o conhecimento que possuem dos outros agentes do ambiente;
- devem ser capazes de reconhecer modificações no ambiente quando estas ocorrerem, alterando sua representação interna do ambiente.

Sichman e Alvares (1997) fazem mais algumas considerações:

- os agentes são concebidos independentemente de um problema particular a ser resolvido. O projeto de um agente deve resultar numa entidade capaz de realizar um determinado processamento, e não numa entidade capaz de realizar este processamento exclusivamente no contexto de uma aplicação alvo particular;
- a concepção das interações também é realizada independentemente de uma aplicação alvo particular. Busca-se desenvolver protocolos de interação genéricos, que possam ser reutilizados em várias aplicações similares;
- a mesma filosofia anterior pode ser estendida ao projeto das organizações. Normalmente, distinguem-se as funcionalidades necessárias a uma resolução particular dos agentes que irão efetivamente implementar tais organizações;
- durante a fase de resolução, os agentes utilizam suas representações locais dos protocolos de interação e das organizações para raciocinar e agir. Deste modo, não existe um controle global do sistema, e este é implementado de forma totalmente descentralizada nos agentes.

Para Reis (2003), um Sistema Multiagente é um sistema computacional em que dois ou mais agentes, homogêneos ou heterogêneos, trabalham ou interagem em conjunto, assincronamente, para atingir objetivos ou executar determinadas tarefas.

As investigações e as implementações dos SMA estão focadas no desenvolvimento de padrões e modelos computacionais para construir, analisar, descrever e implementar as formas de interação e comunicação de agentes em sociedades reduzidas ou de elevada dimensão (REIS, 2003). Na Figura 3.2 encontra-se a estrutura de um SMA.

3.4 Por que usar Sistemas Multiagentes

Apesar da maior eficiência das soluções centralizadas, o interesse em sistemas de agentes distribuídos se deve principalmente ao fato da facilidade de entendimento e desenvolvimento dos mesmos especialmente quando o problema é “naturalmente” distribuído, ou seja, tem informações distribuídas (WEISS, 1999). Um exemplo desta “naturalidade” é uma aplicação de marcação distribuída de reuniões, onde os agentes autônomos encontram-se realmente geograficamente distribuídos, mantendo assim, a privacidade das informações.

Sistemas grandes e de alta complexidade (*large-scale systems*) são o foco também

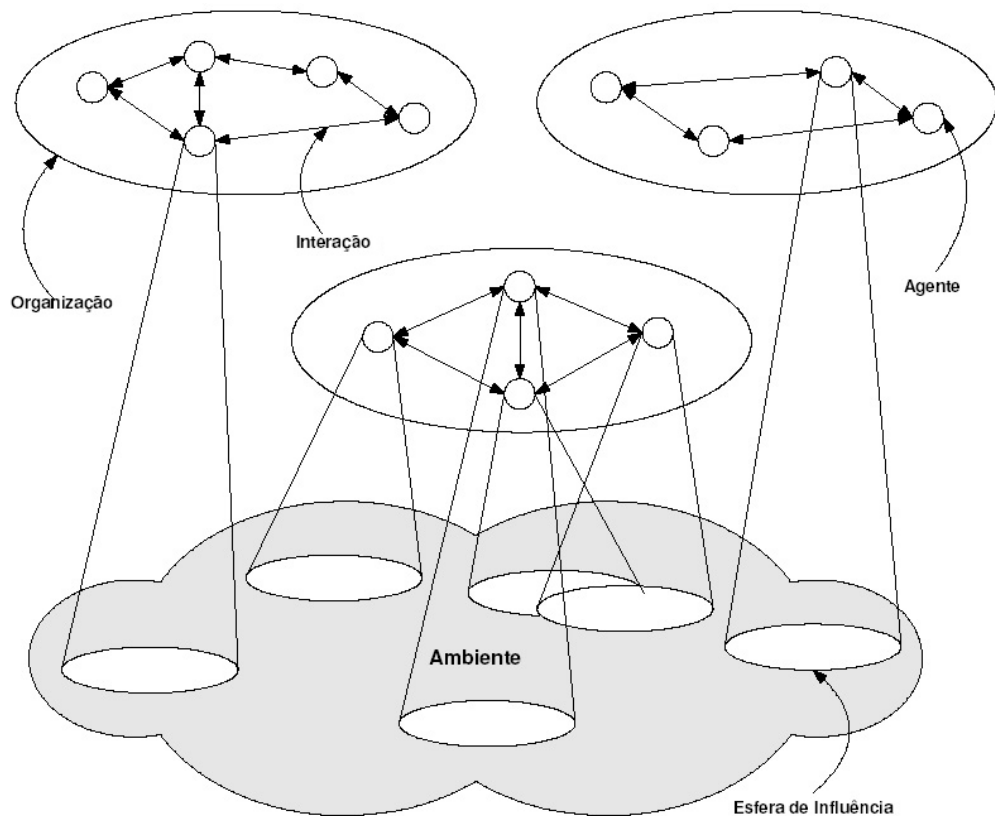


Figura 3.2: Estrutura de um Sistema Multiagente (REIS, 2003).

dos SMA. Para lidar com estes sistemas, a IAD utiliza quatro técnicas: modularização, distribuição, abstração e inteligência (WEISS, 1999).

Reis (2003) cita mais razões para o uso de SMA:

- A dimensão do problema ser extremamente elevada para poder ser resolvido por um único agente monolítico;
- Permitir a interconexão e interoperação de vários sistemas cuja manutenção de código já não é possível;
- Providenciar uma solução natural para problemas geográfica ou funcionalmente distribuídos;
- Fornecer soluções para problemas em que as informações necessárias para a resolução encontram-se distribuídas;
- Permitir uma interface cooperativa homem-máquina mais natural em que ambos funcionam como agentes no sistema;
- Oferecer uma maior clareza e simplicidade conceitual de projeto.

De acordo com o problema e o domínio em que estes se encontram, ter-se-ão diferentes vantagens para aplicar os Sistemas Multiagentes. No entanto, algumas vantagens foram levantadas por Stone e Veloso, em 1996, e citadas por Reis (2003) :

- O próprio domínio do problema o exige, por exemplo devido à distribuição espacial dos intervenientes;
- O paralelismo, atribuindo diferentes tarefas a diferentes agentes de forma que a execução seja mais rápida;
- A robustez, pois utilizam-se diferentes agentes, não existindo, desta forma, um ponto único de falha no sistema;
- A escalabilidade, permitindo o aumento dos agentes intervenientes num determinado sistema aberto;
- A simplificação das tarefas individuais de programação, dividindo o problema global em vários subproblemas;
- O estudo da inteligência individual e do comportamento social, pois os SMA permitem a interoperacionalidade entre os agentes;
- A manutenção da privacidade da informação e conhecimentos individuais de cada agente.

Os SMA apresentam diversos benefícios na resolução de problemas de IA. Em sistemas onde a informação é “naturalmente” distribuída, há uma maior rentabilidade de recursos (REIS, 2003):

- Resolução mais rápida de problemas devido ao processamento concorrente;
- Diminuição da comunicação devido ao processamento estar localizado junto à fonte de informação e a comunicação ser realizada em alto nível;
- Aumento da flexibilidade e escalabilidade resultantes da possibilidade de interconexão de múltiplos sistemas com arquiteturas distintas;
- Aumento da viabilidade devido à inexistência de um ponto único propício a falhas;
- Aumento da capacidade de resposta devido aos sensores, sistemas de processamento e efetadores estarem localizados em conjunto, no interior dos agentes;

- Facilidade no desenvolvimento de sistemas devido à modularidade resultante da decomposição dos problemas e da decomposição dos sistemas em agentes semi-autônomos.

3.5 Características dos Ambientes Multiagentes

Para prover todos os benefícios citados na seção anterior, o agente precisa de um ambiente com infra-estrutura adequada, que irá proporcionar meios para a interação e comunicação.

Weiss (1999) afirma que os ambientes de Sistemas Multiagentes devem ter as seguintes características :

- Promova uma infra-estrutura, especificando comunicações e protocolos de interação;
- Seja aberto e com um *design* não centralizado;
- Tenha agentes autônomos e distribuídos que possam ter interesses próprios ou ser cooperativos.

O ambiente de execução de um SMA pode ter algumas propriedades que aumentam a lista de possíveis características dos ambientes multiagentes (ver Tabela 4).

Tabela 4: Características do Ambiente Multiagente (WEISS, 1999).

Propriedade	Faixa de valores
Autonomia de Projeto	Plataforma/Protocolo de Interação Linguagem/Arquitetura Interna
Infraestrutura de Comunicação	Memória Compartilhada (<i>blackboard</i>) ou Baseado em Mensagem Orientado a conexão ou não orientado a conexão Ponto a Ponto, <i>Multicast</i> ou <i>Broadcast</i>
Diretório de Serviço	Páginas brancas, Páginas amarelas
Protocolo de Mensagem	KQML HTTP e HTML OLE, CORBA e DCOM

Serviços de Mediação	Baseado em Ontologias ? Transações ?
Serviços de Segurança	Tempo/Autenticação
Suporte Operacional	Arquivando/Redundância /Restauração/Contabilização

A Tabela 5 lista algumas propriedades chaves nos ambientes que estão relacionadas com os agentes habitantes.

Tabela 5: Características do Ambiente pelo Agente (WEISS, 1999).

Propriedade	Definição
Conhecido (<i>Knowable</i>)	O agente conhece o ambiente
Previsível (<i>Predictable</i>)	O ambiente pode ser prognosticado pelo agente
Controlável (<i>Controllable</i>)	O agente pode modificar o ambiente
Histórico (<i>Historical</i>)	Estados futuros dependentes de estados passados ou corrente
Tempo Real (<i>Real-Time</i>)	O ambiente pode mudar enquanto os agentes executam

3.6 Comunicação entre Agentes

A definição do *Oxford Dictionary of Computing*, citada na Seção 3.1, considera o agente como uma entidade com habilidade de perceber, processar e agir. Porém, esta definição fica incompleta para agentes participantes de SMA, pois é necessário que o agente tenha habilidades sociais para ser parte integrante nos SMA, que por definição são essencialmente sociais.

A comunicação é uma das habilidades sociais necessárias a um agente de SMA. É responsável pela “ponte” entre os agentes, proporcionando a troca de informação ou negociação de interesses. Reis (2003) demonstra um agente (Figura 3.3) com capacidade de comunicação. Foi inserido na arquitetura do agente um módulo de comunicação que está diretamente ligado ao módulo central (Módulo Inteligente). Através do módulo de comunicação, o módulo inteligente controla o envio e recebimento das mensagens, ou seja, tem acesso às mensagens recebidas e define que mensagens a enviar.

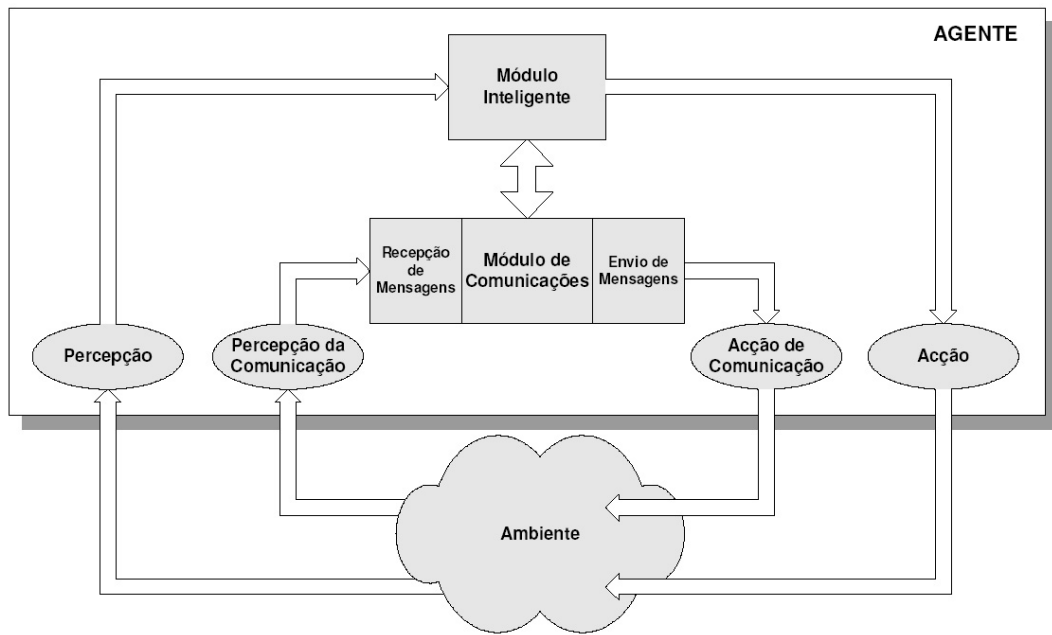


Figura 3.3: Agente com habilidade de comunicação (REIS, 2003).

3.6.1 Características da Comunicação e Significado das Mensagens

A comunicação tem por objetivo a partilha de conhecimento e a coordenação de atividades entre agentes. No entanto, para alcançar estes objetivos os agentes precisam compartilhar três aspectos de estudo formal da comunicação (WEISS, 1999):

- **Sintaxe:** parte da gramática que estuda a disposição das palavras na frase e a das frases no discurso (AURÉLIO, 1977);
- **Semântica:** estudo das mudanças ou translações sofridas, no tempo e no espaço, pelo significado das palavras (AURÉLIO, 1977);
- **Pragmática:** conjunto de regras de ação e fórmulas de interpretação dos símbolos utilizados na comunicação (REIS, 2003).

Um termo muito importante na comunicação de SMA é o significado. Formado pela combinação da semântica e pragmática, o significado orienta o agente a entender e a ser entendido. Por isso é necessário considerar as diferentes dimensões dos significados que são associados à comunicação (WEISS, 1999):

- **Descritiva versus Prescritiva:** algumas mensagens descrevem fenômenos, enquanto outras prescrevem comportamentos. A descrição é importante na comunicação hu-

mana, porém, difícil de ser implementada por agentes. Desta forma, os agentes trocam informações, na maioria das vezes, sobre atividades e comportamentos;

- Significado Personalizado *versus* Convencional: a interpretação que um agente possui sobre uma mensagem pode não ser a de significado convencional (a mesma de outros agentes). Nesta situação, os Sistemas Multiagentes devem optar pelo significado convencional, principalmente quando o ambiente do sistema for aberto (agentes entram e saem a qualquer instante);
- Significado Subjetivo *versus* Objetivo: similar ao significado convencional, onde o significado é interpretado externamente ao agente, a objetividade produz mensagens com efeito explícito no ambiente;
- Perspectiva do Remetente *versus* Destinatário *versus* Sociedade: independentemente do significado convencional ou objetivo da mensagem, esta pode ser expressa segundo o ponto de vista do remetente, destinatário ou dos agentes restantes no ambiente;
- Semântica *versus* Pragmática: a pragmática da comunicação concentra-se em como os agentes usam a comunicação. Isto inclui considerações do estado e do ambiente do agente, que estão externas à sintaxe e semântica da comunicação;
- Contextualidade: as mensagens não devem ser interpretadas isoladamente, mas sim considerando o estado do agente, o estado presente do ambiente e o histórico do ambiente (como o ambiente chegou ao atual estado). As interpretações são diretamente afetadas pela ação dos agentes e pelas mensagens enviadas;
- Cobertura: linguagens pequenas são mais flexíveis e mais facilmente utilizáveis. Porém, precisam ser suficientemente grandes para atender às necessidades de expressão dos agentes;
- Identidade: o significado da comunicação é dependente da identidade, das regras em que o agente está envolvido e como o agente envolvido é especificado. A mensagem pode ser enviada a um agente particular ou para qualquer agente que satisfaça o critério especificado;
- Cardinalidade: a mensagem enviada a um agente específico pode ter interpretação diferenciada se enviada a todos os agentes.

3.6.2 Os Níveis de Comunicação

Weiss (1999) cita três níveis de especificação nos protocolos de comunicação:

- Nível mais baixo: Especifica o método de interconexão;
- Nível médio: Especifica o formato ou sintaxe da informação a ser transferida;
- Nível mais alto: Especifica o significado ou semântica (refere-se não somente ao conteúdo da mensagem, mas também ao tipo) da informação a ser transferida.

Existem protocolos de comunicação binários, envolvendo um emissor e um receptor, e n-ários, onde existe um emissor e vários receptores (também chamado de *broadcast* ou *multicast*). A estrutura de informação do protocolo de comunicação é especificada por cinco campos: emissor, receptor, linguagem no protocolo, função de codificação e de decodificação e ações a serem tomadas pelos receptores (WEISS, 1999).

3.6.3 Tipos de Mensagens

Independentemente de outras capacidades funcionais, os agentes precisam ser capazes de comunicar. O interesse de comunicação dos agentes, entre si, levam-nos para o diálogo. O agente pode assumir o papel de ativo, passivo ou ambos no diálogo, permitindo funcionar como *master*, *slave* ou *peer* respectivamente. Existem dois tipos básicos de mensagens: afirmação e interrogação; e quatro tipos de agentes, quanto ao papel desempenhado na comunicação: agente básico, passivo, ativo e *peer* (Tabela 6). O agente básico tem a capacidade de receber as informações de fontes externas que as enviam por meio de uma afirmação. Caso o agente assuma o papel passivo, terá que adicionar a capacidade de aceitar as interrogações de fontes externas e respondê-las por meio de afirmações. Se o agente for ativo no diálogo, terá que fazer interrogações e afirmações, o que potencializará para controlar outros agentes. Por fim, o agente pode funcionar como *peer*, assumindo o papel de ativo em um determinado instante e passivo em outro. Para tal, é necessária a capacidade de fazer e receber as afirmações e interrogações.

Tabela 6: Agentes e suas Capacidades (WEISS, 1999).

	Agente Básico	Agente Passivo	Agente Ativo	Agente <i>Peer</i>
Recebe Afirmação	*	*	*	*
Recebe Perguntas		*		*
Envia Afirmação		*	*	*
Envia Perguntas			*	*

Weiss (1999) lista outros tipos de mensagens derivadas do trabalho em *Speech-Act* (Tabela 7).

Tabela 7: Tipos de mensagens provindas da *Speech-Act* (WEISS, 1999).

Ações de Comunicação	Força de Elocução	Resultado Esperado
Afirmação	Informar	Aceitação
Pergunta	Questionar	Resposta
Resposta	Informar	Aceitação
Pedido	Pedir	
Explicação	Informar	Acordo
Comando	Pedir	
Permissão	Informar	Aceitação
Recusa	Informar	Aceitação
Oferta	Informar	Aceitação
Aceitar		
Acordar		
Proposta	Informar	Oferta
Confirmação		
Retração		
Negação		

3.6.3.1 A Teoria *Speech-Act*

A teoria *Speech-Act* analisa a comunicação humana, que é usada como base para o desenvolvimento da comunicação de agentes computacionais, e a caracteriza como uma linguagem de ação, ou seja, de pedidos, sugestões, comprometimento e respostas. A teoria tem três aspectos (WEISS, 1999):

- Locução: é a pronúncia física do emissor;
- Elocução: sentido atribuído à locução;
- Perlocução: ação resultante da locução.

Speech-Act é uma teoria que ajuda a definir o tipo da mensagem usando o conceito de força da elocução, ou seja, relaciona a semântica da comunicação com a ação, através das performativas, não deixando dúvidas ao receptor. Este relacionamento simplifica o desenvolvimento de agentes computacionais.

3.6.4 Linguagens de Comunicação

Financiado pelo DARPA (*Defense Advanced Research Projects Agency, E.U.A*), foi fundado no início dos anos 90 o KSE (*Knowledge Sharing Effort*) para o desenvolvimento de protocolos de troca e representação de informação entre sistemas de informação autônomos. Os resultados gerados foram dois produtos principais (REIS, 2003):

- A linguagem KQML (*Knowledge and Query Manipulation Language*): é uma linguagem externa para comunicações entre agentes. Define um invólucro para formatar mensagens que determinam o significado locutório da mensagem. A KQML não está preocupada com o conteúdo da mensagem mas sim com a caracterização da informação necessária à compreensão desse conteúdo. A Figura 3.4 mostra que a KQML é uma linguagem de comunicação entre entidades heterogêneas;

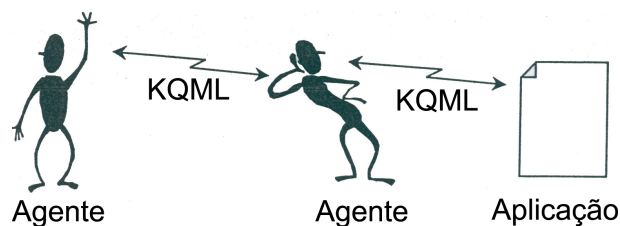


Figura 3.4: KQML - um protocolo de comunicação para agentes e programas (WEISS, 1999).

- O formato KIF (*Knowledge Interchange Format*): uma linguagem criada explicitamente para representar o conhecimento sobre um domínio de discurso específico. O formato foi desenvolvido primeiramente como forma de definir o conteúdo de mensagens expressas em KQML.

Existem, além de KQML e KIF, várias outras linguagens de comunicação em Sistemas Multiagentes. Entre as mais utilizadas destaca-se a FIPA ACL (*Agent Communication Language*).

Desde 1995 a FIPA (*Foundation for Intelligent Physical Agents*) gera padrões para Sistemas Multiagentes, e um dos pontos principais da especificação é a linguagem ACL, que é resultante do KQML, que primariamente é uma linguagem de comunicação externa e não obriga a utilização de qualquer linguagem específica para o conteúdo. Mais informações da linguagem FIPA ACL são fornecidas na Seção 3.7.1.

3.7 Linguagem, *framework* e especificações para agentes

As especificações mais conhecidas e utilizadas são produzidas pela FIPA. Seguindo a padronização FIPA, surgiram várias ferramentas que implementam tal padronização. Uma dessas ferramentas é a *framework* JADE (*Java Agent Development Framework*). Sem dúvida nenhuma o JADE hoje é um dos *frameworks* mais utilizados no mundo devido a sua robustez (modularizado) e facilidade.

3.7.1 FIPA

A FIPA é uma associação de companhias internacionais sem fins lucrativos que, juntas, tentam produzir especificações de tecnologias genéricas para agente. FIPA não é apenas uma tecnologia para uma aplicação, mas sim tecnologias genéricas para aplicações diferentes. FIPA é baseada em duas suposições principais:

- A primeira determina que o momento de alcançar o consenso e de terminar o padrão não deve ser longo e, principalmente, não deve agir como um freio no progresso;
- A segunda determina que somente o comportamento externo de componentes do sistema deve ser especificado, deixando detalhes da execução e arquiteturas internas aos desenvolvedores dos agentes.

Os primeiros documentos emitidos pela FIPA, chamados de FIPA97, especificam as regras que permitem que uma sociedade de agentes relacione-se e seja controlada de forma eficaz.

Durante os anos foram sendo desenvolvidas novas especificações (FIPA98, FIPA2000), sempre visando a inter-operabilidade e reutilização, que incrementavam mais funcionalidades em nível de abstração para que a comunicação, que é o coração de um sistema com agentes, fosse melhor e mais flexível. Apesar de várias especificações, há uma tentativa de manter compatibilidade entre as versões, a qual não é total mais existe.

A últimas especificações emitidas pela FIPA foram do grupo FIPA2002, que comparadas com as anteriores, trouxeram ganhos e principalmente maior definição em alto nível (abstrata). O grupo de especificações FIPA2002 é dividido, pelo assunto, em cinco partes (Figura 3.5):

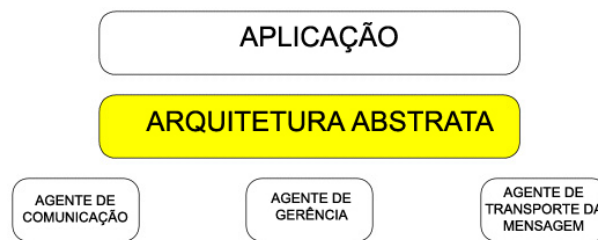


Figura 3.5: Partes da especificação FIPA2002.

- Especificação FIPA Aplicação (*Applications*): são exemplos de áreas em que os agentes FIPA podem ser aplicados. Especifica os serviços e as ontologias de domínios particulares;
- Especificação FIPA Arquitetura Abstrata (*Abstract Architecture*): tratam das entidades abstratas que são requeridas para construir serviços dos agentes e seus ambientes;
- Especificação FIPA Agente de Comunicação (Figura 3.6): tratam das mensagens da língua de comunicação do agente (ACL), dos protocolos da interação, das trocas das mensagens e teoria baseada no ato de discurso (*Speech-Act*);



Figura 3.6: Divisões da especificação FIPA *Agent Communication*.

- FIPA Agente de Gerência (*Agent Management*): tratam do controle e da gerência dos agentes dentro e através das plataformas do agente;
- FIPA Agente de transporte da mensagem (*Agent Message Transport*) (Figura 3.7): tratam do transporte e da representação das mensagens através dos protocolos diferentes do transporte da rede, incluindo ambientes *wireless*.

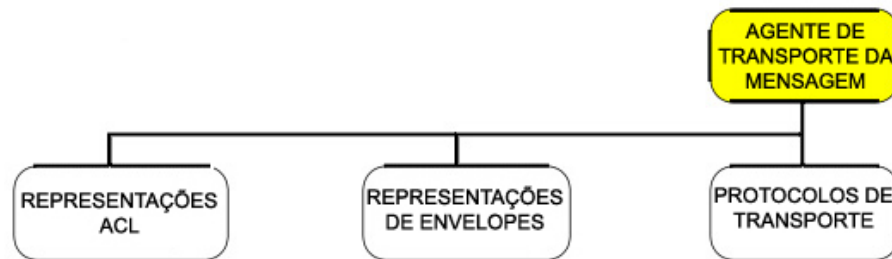


Figura 3.7: Divisões da especificação FIPA Agente de Transporte da Mensagem.

Dentro da Especificação Abstrata da Arquitetura FIPA (FIPA, 2002) pode-se encontrar uma noção (*overview*) da arquitetura, que por sua vez é composta de agentes e serviços. Entre estes elementos estão :

- Serviço de diretórios para Agentes;
- Serviço de diretórios para Serviços;
- Mensagens de Agentes;
- Enviar Mensagem a outros Agentes;
- Validação de Mensagem e Criptografia;
- Provendo Interoperabilidade.

No serviço de diretórios para Agentes, pode-se identificar como regra básica prover um lugar onde os agentes registrarão suas descrições como uma das entradas no diretório (*agent-directory-entries*) e onde outros agentes podem pesquisar dentro deste diretório outros agentes para interagir.

No momento do registro o agente preenche dois valores que são requeridos pelo serviço. Uma é o Nome do Agente (*Agent-name*), que irá identificar o agente, por isso o valor terá que ser global e único.

A principal função do serviço de diretórios para Serviços é prover um centro de informações consistente, dos serviços disponibilizados pelos agentes. Operacionalmente, o agente registra os possíveis serviços que gostaria de disponibilizar através de um agente, agente de serviços (*agent-directory-service*), que será usado nas pesquisas de agentes ou serviços que necessitam de uma funcionalidade a mais. São requisitadas 3 entradas de valores para cadastrar o serviços. São elas: Nome do Serviço (*Service-name*) (Global e único); Tipo do Serviço(*Service-type*); Localização do Serviço(*Service-locator*). Não se pode esquecer que podem ser implementados novos valores para adequar à necessidade.

A comunicação entre agentes é dada pela troca de mensagens, e esta troca tem que ser eficiente para que haja comunicação. Dentro do FIPA estas mensagens de comunicação foram divididas em três aspectos, que são: a estrutura da mensagem, a representação da mensagem e o transporte desta mensagem.

A estrutura da mensagem (Figura 3.8) é definida no FIPA ACL como contendo um cabeçalho, onde pode-se identificar uma linguagem de conteúdo (*content-language*), ontologia, o remetente e o(s) destinatário(s). Estes últimos, são expressados pelo NOME AGENTE, que é global e único. Na falta de destinatário se faz *broadcast*. E por fim a própria mensagem que pode ser recursiva e ter outra mensagem.

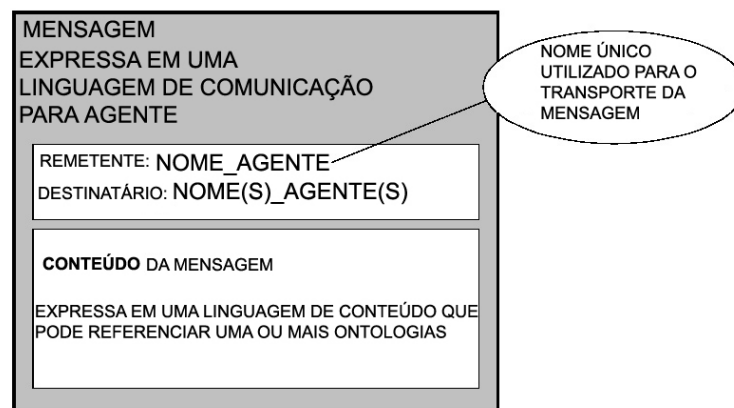


Figura 3.8: A mensagem (FIPA, 2002).

Para enviar esta mensagem, é necessária uma rotina de transporte que é responsável por preparar a mensagem, dando a ela uma cabeçalho após uma codificação prévia. Este cabeçalho é colocado de acordo com o meio de transporte da mensagem. Após isto, ela é incluída na estrutura final da mensagem que será enviada. A Estrutura de Transporte da Message é por si um cabeçalho do envelope, este envelope tem como valores a descrição do transporte, o remetente e o destinatário. O envelope pode conter informações extras, como por exemplo, Representação de Codificação responsável por definir o tipo de codificação

utilizada na mensagem. A Figura 3.9 mostra os encapsulamentos.

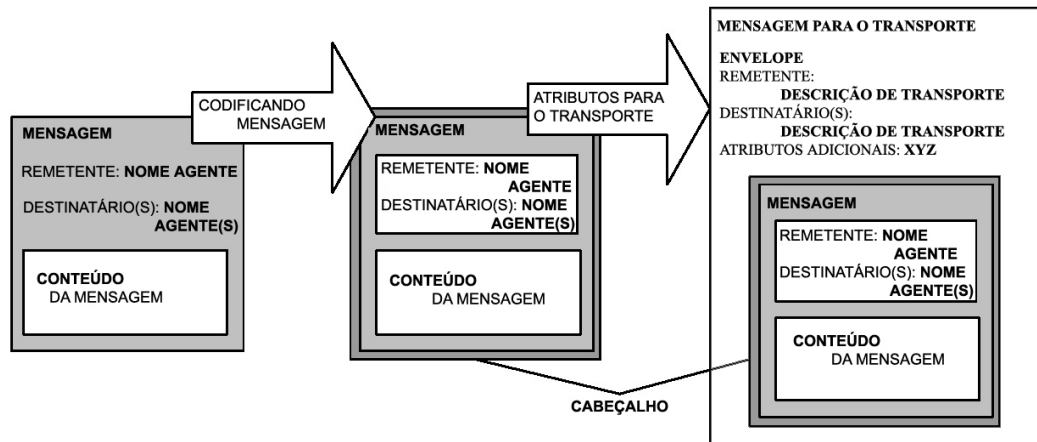


Figura 3.9: Preparando a mensagem para o transporte (FIPA, 2002).

Um exemplo de mensagens utilizadas na comunicação entre agentes está na Figura 3.10 e na Figura 3.11 um exemplo de criptografia.

Toda estas definições vem ajudando o padrão a se tornar um dos mais usados. Baseado neste padrão foi desenvolvido o JADE, que será o assunto da Seção 3.7.2.

3.7.2 JADE

Existem alguns *frameworks* para o desenvolvimento de agentes como: FIPA-OS (FIPA, 2003), JADE (JADE, 2003), Zeus Project (PROJECT, 2004) e Aglets (AGLETS, 2004). Será dado um destaque maior no JADE, por existir uma grande comunidade de colaboradores e por ser uma das mais utilizadas.

JADE é um *framework* criada para facilitar a criação de aplicações com agentes. O *framework* segue os padrões FIPA e dá suporte a multi-agentes inteligentes.

O objetivo do JADE é simplificar o desenvolvimento, disponibilizando ferramentas de suporte à depuração e às fases de desenvolvimento. A plataforma de agente pode ser distribuída entre máquinas (que podem ter sistemas operacionais diferentes) e a configuração pode ser controlada de forma remota. Em tempo de execução, o controlador pode transferir um agente de uma máquina para outra quando necessário. Para disponibilizar estas ferramentas, o JADE traz com si um requisito, o pacote *Java Run Time* versão 1.2 ou superior (JADE, 2003).

Para atingir seu objetivo de facilitar o desenvolvimento, o JADE disponibiliza ferramentas para o desenvolvedor do agente:

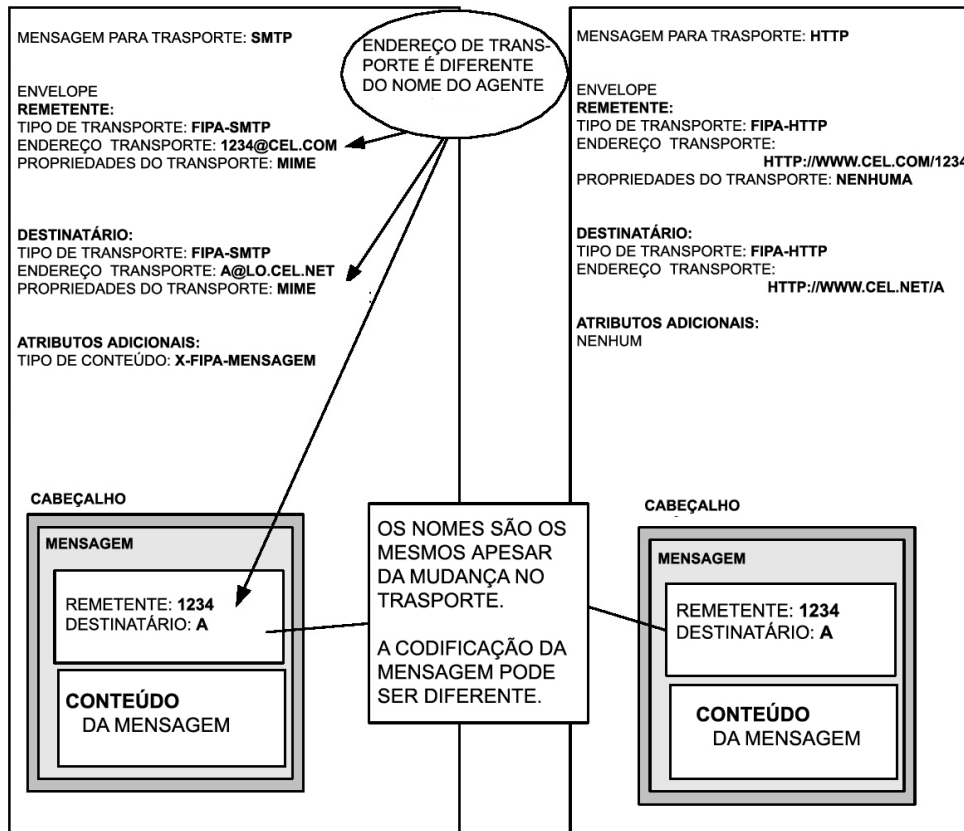


Figura 3.10: Duas Mensagens de Transporte para o mesmo agente (FIPA, 2002).

- FIPA - *Compliant Agent Platform*, que inclui o AMS (*Agent Management System*), o DF (*Directory Facilitator*) e o ACC (*Agent Communication Channel*). Todas estes agentes são ativados automaticamente na inicialização da plataforma;
- Plataforma distribuída, o que permite que ela esteja dividida em vários computadores (desde que não haja um *firewall* entre eles que impeça esta comunicação);
- A quantidade de DFs pode ser definida em tempo de execução, possibilitando assim que seja implementado uma aplicação multi-domínio;
- Uma interface para facilitar o registro de serviços providos pelos agentes a um ou mais domínios;
- Um mecanismo de transporte, além de uma interface para envio e recebimento de mensagens para agentes;
- Protocolos para conectar diferentes plataformas de agentes;
- O transporte da mensagem (ACL), quando feito dentro da plataforma, é codificada como um objeto Java, e caso o emissor e receptor não estejam na mesma plataforma,

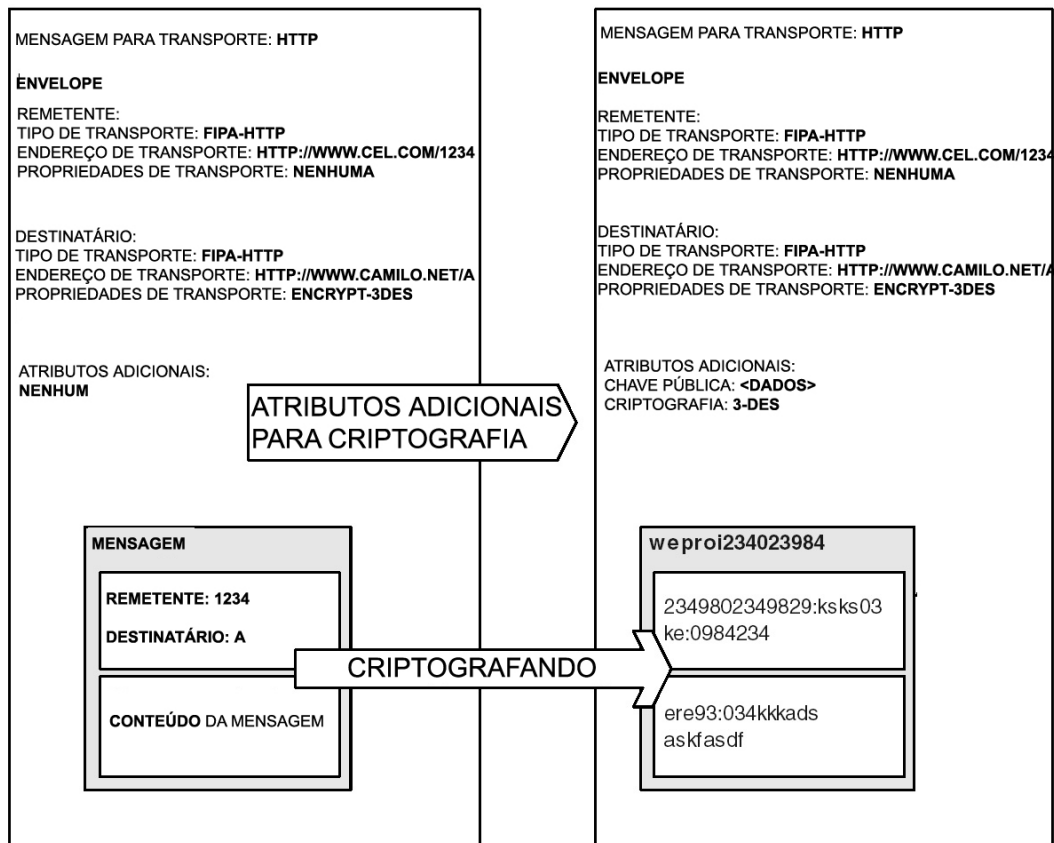


Figura 3.11: Criptografando o cabeçalho da Mensagem (FIPA, 2002).

é feita uma montagem automática de um vetor de caracteres (*string*) FIPA;

- Bibliotecas para interações do modelo FIPA;
- Registro automático dos agentes com o AMS;
- *Name-service*, que irá dar ao agente um GUID (*Globally Unique Identifier*) responsável pela identificação pela plataforma;
- Uma interface gráfica para controlar e monitorar os agentes e a plataforma.

3.7.2.1 A Arquitetura da Plataforma de Agente JADE

O JADE, seguindo a especificação da FIPA, implementa agentes para o controle da plataforma. Dentre os agentes estão o ACC, o AMS e o DF. Toda comunicação é feita através de mensagens utilizando a linguagem FIPA ACL.

A arquitetura é baseada na JVM (*Java Virtual Machines*) e a transferência de mensagem no Java RMI (*Remote Method Invocation*).

Cada JVM é um *container* básico de agentes que provê um ambiente para execução de agentes e permite a execução concorrente entre agentes numa mesma máquina.

Na Figura 3.12 poderá ser observada a Plataforma de Agente que tem um conjunto de *containers* de agentes.

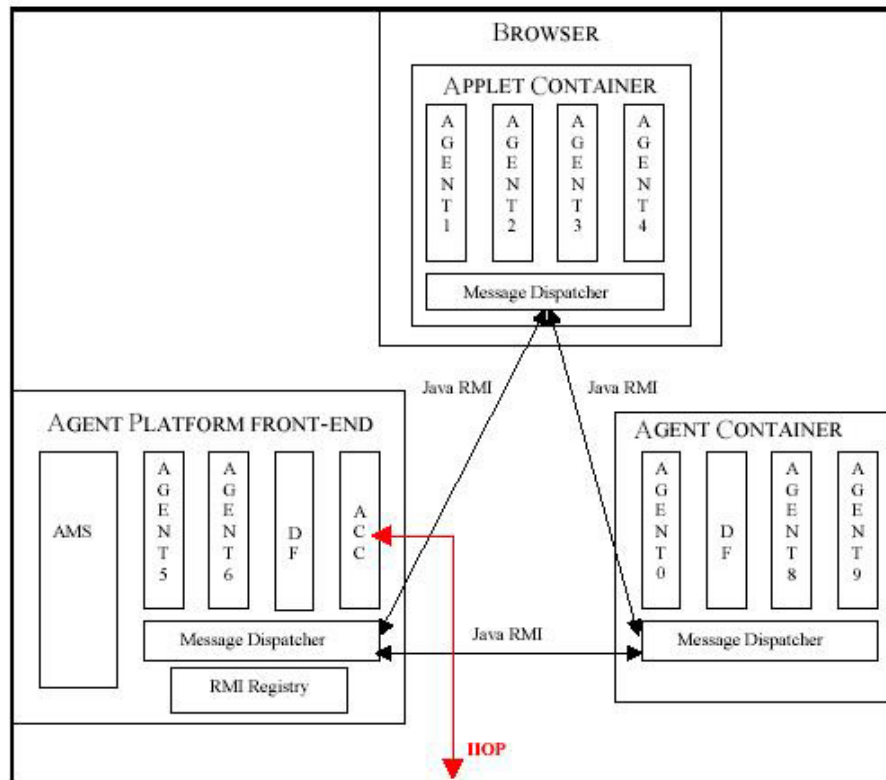


Figura 3.12: Arquitetura de uma plataforma JADE.

Os agentes *containers* podem estar espalhados na rede, mas para isso ele precisa ser um objeto RMI servidor que gerencia um conjunto de agentes locais. Ele controla o ciclo de vida dos agentes criando, suspendendo, liberando e matando os mesmos. Este container também faz controle da comunicação dos agentes locais.

A interface gráfica que controla os agentes remotos tem uma implementação através do RMA (*Remote Monitoring Agent*). É através deste agente que é possível, por exemplo, parar ou reinicializar um agente remoto. Toda comunicação entre os agentes e esta interface e toda a comunicação entre a interface e o AMS é feita através da ACL.

Outra implementação que chama a atenção é a forma com que o JADE trata as mensagens para que haja o menor consumo de recursos no momento da troca de mensagens. Os cabeçalhos de controle (*overheads*) dependem da localização do destinatário e o status do cache, como descrito a seguir:

- mesmo Recipiente(*Same Container*): neste ambiente a comunicação se dá entre agentes que estão no mesmo container e ela é feita sem instanciar controles remotos. É feito um *clone* do objeto ACL;
- mesma Plataforma e diferente Recipiente(*Same Agent Platform, different container, cache hit*): neste cenário os agentes estão na mesma plataforma, no entanto em containers diferentes e com a *cache hit*. Para isso serão feitas duas chamadas no RMI, uma para atualizar a *cache* e outra para enviar a mensagem;
- Plataforma JADE diferente(*Different platform*): neste cenário os agentes estão em plataformas diferentes. É feita uma chamada direta ao ACC gerando um custo de dupla transformação. A primeira é feita do objeto Java para a *String* Java e a segunda da *String* Java para a *stream* IIOP. Estas transformações são feitas na origem e desfeitas no destino;
- Plataforma, não JADE, diferente(*Different platform*): este cenário é semelhante ao anterior.

4 *Comércio Eletrônico e Negociação em SMA*

Impulsionado pelo grandioso desenvolvimento da *internet*, surgiu o comércio eletrônico (*E-commerce*) com as categorias mais usadas: *business-to-consumer* (B2C) e *business-to-business* (B2B). No entanto, a maioria dos sistemas atuais de comércio eletrônico são pouco mais do que catálogos eletrônicos, que impõe ao usuário um produto com termos e condições predeterminadas e inflexíveis. Na tentativa de aumentar a utilidade do comércio eletrônico, surgiu uma nova geração de sistemas, a qual vem sendo aprimorada, baseada em negociação automatizada. Neste capítulo serão apresentados alguns conceitos, características e modelos importantes para esta nova geração de sistemas de comércio eletrônico que utilizam de Sistemas Multiagentes.

4.1 Comércio Eletrônico

O encontro de duas entidades: por um lado os clientes, anunciando a intenção de compra e do outro os vendedores, anunciando a intenção de venda, definem a forma do comércio. Com as melhorias das tecnologias da informação e comunicação, que possibilitaram a criação do comércio eletrônico, as transações comerciais entre clientes e vendedores torna-se mais fáceis e ágeis.

Encontra-se na literatura definições mais amplas de comércio eletrônico: segundo a *Electronic Commerce Association*, citado em He, Jennings e Leung (2003), o comércio eletrônico abrange qualquer forma de negócio, transação administrativa ou troca de informação que é executada usando qualquer tecnologia da informação e comunicação. Comparado ao comércio tradicional, o comércio eletrônico tem muitas vantagens como: informações atualizadas, vinte e quatro horas de serviços a um custo baixo, além da possibilidade de implantação do preço dinâmico (aumento de rentabilidade) e automatização dos processos de negociação e transação.

Há cada vez mais empresas com *sites* na *internet*, tentando satisfazer e conquis-

tar mais clientes. No entanto, a maioria destes *sites* são apenas catálogos eletrônicos ou classificados, que possibilitam ao usuário pesquisar por produtos e, em alguns *sites*, após encontrá-los pagá-los. Estes *softwares* de comércio eletrônico, também chamados de catálogos *on-line*, são classificados como sistemas da primeira geração. Ou seja, sistemas onde os servidores (vendedores), conectados na *internet*, fornecem uma lista de produtos com regras de preço, pagamento, entrega e outras, bem definidas para os que possuem navegador(*browser*) conectados à *internet*, permitindo a navegação pelo catálogo para escolha do produto (LOMUSCIO; WOOLDRIGE; JENNINGS, 2001).

Existem sistemas de primeira geração, chamados de assistentes de compra, que aumentaram as funcionalidades e implementaram um ambiente que possibilita o cliente (comprador) a escolher a forma de pagamento, após ter escolhido o produto. No entanto, esta escolha ainda é restrita, pois as opções são pré-definidas pelo fornecedor do produto.

Um outro tipo de assistente de compra disponibiliza a funcionalidade de escolha do produto ao cliente, ou seja, o agente tenta automatizar o processo de escolha do produto fornecendo ao cliente o produto que ele precisa.

Apesar das melhorias nos sistemas considerados de primeira geração, um problema chave, o foco em apenas um aspecto da transação: o preço, permaneceu. No entanto, a solução deste problema (parametrização por parte do cliente de outros aspectos como tempo de entrega), juntamente com a negociação automatizada, que reduz o tempo de negociação e aumenta o número de negociações por fração de tempo, fazem dos sistemas de segunda geração ferramentas importantes no aumento de rendimentos para as empresas (LOMUSCIO; WOOLDRIGE; JENNINGS, 2001).

4.1.1 Classificação do Comércio Eletrônico

Analisando os mercados, onde a tecnologia de agentes juntamente com a automação da negociação são aplicados, Vetter e Pitsch (2000) classificam-os segundo quatro critérios principais: tipos de participantes, número de agentes, tipo de produto e tipo de alocação.

De acordo com os tipos de participantes, Vetter e Pitsch (2000) considera três classificações para os mercados:

- *business-to-consumer* (B2C) : o consumidor (pessoa física) é obrigado a aceitar o método de pagamento estipulado pela empresa vendedora do produto;
- *business-to-business* (B2B) : existe uma parceria entre as empresas (cliente e ven-

dedor), permitindo assim, uma maior flexibilidade nos termos das negociações;

- *consumer-to-consumer* (C2C) : as transações são feitas entre duas entidades (pessoa física) e têm uma maior variedade de opções para escolha na forma de pagamento do que o B2C, porém, menor que o B2B. Por exemplo, pessoa física não aceita cartões de créditos.

Em He, Jennings e Leung (2003), baseado em Turban et al. (1999), a classificação quanto ao tipo de participantes é maior, pois sendo mais abrangente, considera seis classificações:

- *business-to-business* (B2B): refere-se a transações onde ambos, vendedores e compradores, são empresas. É o tipo de mercado que movimenta a maior quantidade de dinheiro, além de ser apontado como o mais promissor mercado para os próximos anos;
- *business-to-consumer* (B2C): refere-se a transações feitas por um consumidor (pessoa física) que conduz a transações pelo *website* da empresa (vendedora);
- *consumer-to-consumer* (C2C): modelo em que duas pessoas físicas comercializam diretamente;
- *consumer-to-business* (C2B): a pessoa física usa da *internet* para vender produtos ou serviços para as empresas;
- não comercial : envolve entidades não comerciais, como instituições acadêmicas ou governamentais. Utiliza-se do comércio eletrônico para reduzir gastos e prover serviços;
- negociação interna(*intrabusiness*): toda atividade da empresa é feita eletronicamente, como troca de produtos, serviços ou informações entre unidades ou indivíduos.

O mercado pode ser classificado também quanto ao número de agentes envolvidos nas transações. São três categorias:

- Um para um 1:1 (*one-to-one*): existe um vendedor e um consumidor (Figura 4.1);
- Um para muitos 1:N (*one-to-many*): um vendedor para vários consumidores ou um consumidor para vários vendedores (Figura 4.2);

- Muitos para Muitos M:N (*many-to-many*): muitos vendedores relacionam com muitos consumidores (Figura 4.3).



Figura 4.1: Relação de 1:1.

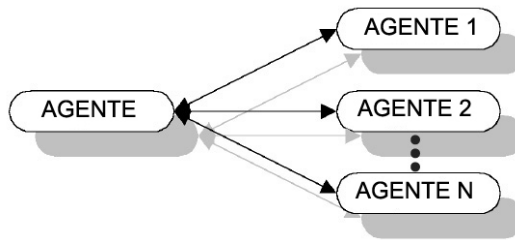


Figura 4.2: Relação de 1:N.

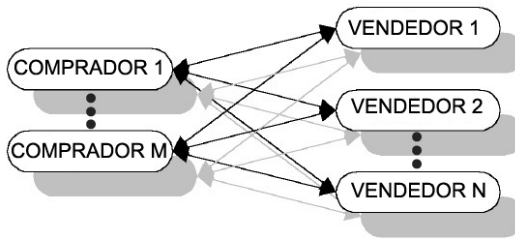


Figura 4.3: Relação de M:N.

A categoria 1:1 é a mais simples das relações apresentadas, e a mais utilizada, tanto no mercado eletrônico quanto no convencional. A categoria 1:N é usada para maximizar o retorno para o agente de papel único na transação. Um exemplo desta categoria é o leilão. Na categoria M:N, cada negociação pode fazer parte de uma transação complexa. Esta transação pode ser executada em paralelo com muitas outras transações, algumas das quais podem estar competindo pelos recursos ou complementando-os.

Além das configurações de mercado citadas, existem duas outras configurações que inserem a figura do intermediário (representado por B) no mercado: a primeira é a 1:B:N (Figura 4.4), onde o intermediário coleta ordens do consumidor e as tenta cumprir junto aos vendedores; e a segunda é a M:B:B:N (Figura 4.5), onde os dois intermediários (um representando os consumidores e o outro os vendedores) comunicam para tentar cumprir as ordens.

Outra classificação que pode ser feita é quanto ao tipo de produto, ou seja, se é um produto físico ou virtual. Um exemplo de produto virtual é a informação. Estes

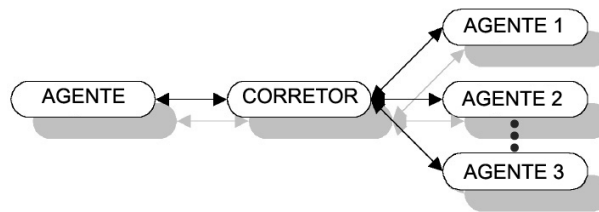


Figura 4.4: Relação de 1:B:N.

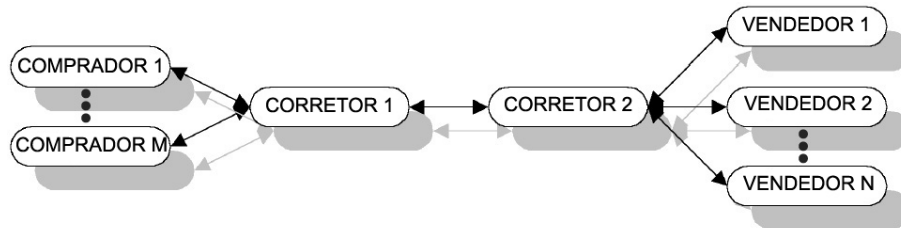


Figura 4.5: Relação de M:B:B:N.

produtos são considerados os melhores produtos para o comércio eletrônico por serem de fácil movimentação. Já produtos físicos, especialmente alimentos perecíveis, tendem a ter uma faixa de mercado mais nacional, em alguns casos continental. Apesar de que muitos produtos físicos são comercializados internacionalmente.

Além da divisão entre físico e virtual outras sub-divisões podem ser feitas quanto ao valor de venda e a descrição do produto (ontologia), categorizando os mercados para as padronizações e, assim, ter-se maior retorno das ferramentas customizadas para cada ambiente.

Por fim, o mercado pode ser classificado pelo tipo de alocação utilizada. A forma mais tradicional desta classificação é a negociação, que pode ser longa e tediosa se as partes envolvidas são humanos, pois tendem a incluir termos subjetivos como preferências culturais, orgulhosas e sentimentais. Já os agentes computacionais seguem regras para maximizar o retorno do usuário, sem serem subjetivos.

4.1.2 Modelos do Comércio Eletrônico

Os modelos auxiliam no entendimento do funcionamento de qualquer aplicação. No entanto, modelar comportamento, que é o caso do comércio eletrônico, é uma tarefa árdua, devido a dificuldade de se encontrar padrões. Os modelos apresentados neste trabalho tentam modelar dois tipos de comércio eletrônico o B2C (*Business to consumer*) e o B2B (*Business to business*). Para o primeiro o modelo apresentado será o CBB (*Consumer Buying Behavior*) e para o segundo será o BBT (*Business to Business Transaction*).

4.1.2.1 O Modelo CBB

Analisando o comportamento dos agentes no B2C, foi criado o modelo CBB (Figura 4.6), que divide em seis os estágios da comercialização dos produtos em mercado eletrônico.

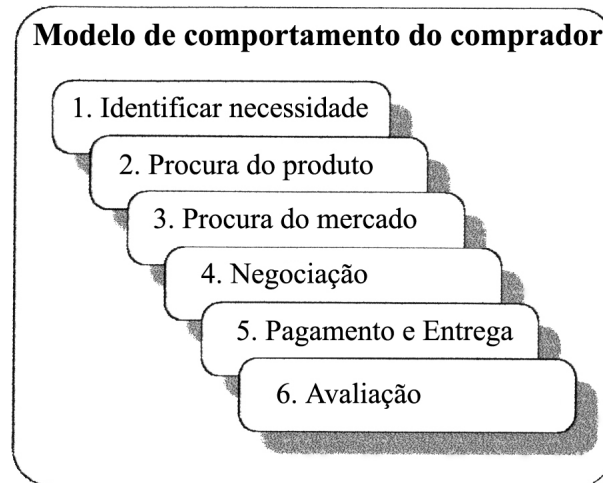


Figura 4.6: Modelo CBB (VETTER; PITSCH, 2000).

O primeiro estágio é chamado de “Identificação de necessidade” e incorpora todas as funcionalidades de informar ao cliente sobre os produtos ou serviços disponíveis no mercado que corresponde ao perfil do mesmo. O segundo estágio é formado pela “Procura do Produto” que satisfaça às necessidades levantadas. As técnicas mais usadas neste estágio são: “Filtragem baseada em características”, “Filtragem colaborativa” e “Filtragem baseada em requisitos”. Na Tabela 8 estão alguns comércios eletrônicos existentes e respectivas técnicas aplicadas. A Tabela 9 mostra as definições e comparações entre as técnicas de filtragem do segundo estágio, chamado de “Procura do Produto”.

O terceiro estágio, chamado de “Procura do mercado”, é o momento da escolha de um mercado apropriado para comercializar os produtos ou serviços selecionados. Algumas ferramentas de procura do mercado (*BargainFinder*, *Priceline*, *Buscapé*) focalizam na busca pelo mercado que oferece o menor preço. No entanto, o preço pode não ser o único fator determinante na escolha de um mercado. Fatores como tempo de entrega ou brindes podem ser levados em consideração. Um exemplo deste mercado é o *Frictionless* (2004).

Tabela 8: Técnicas de Filtragem para a “Procura do Produto” no comércio eletrônico (HE; JENNINGS; LEUNG, 2003).

	Características	Colaborativa	Requisitos
Amazon	X		X
eBAY	X		X
CDNOW	X	X	
Yahoo shopping	X		X
Net Perceptions		X	

Tabela 9: Comparação entre as técnicas de “Procura do Produto” (HE; JENNINGS; LEUNG, 2003).

	Características	Colaborativa
Quando usar a técnica	As necessidades do usuário é conhecida	As necessidades do usuário não são conhecidas
É requerido	Característica do produto	Perfil do usuário
Interação com usuário	Média	Pouca
Retorno de resultados	Produtos que satisfaça as características requeridas	Sugestão de produtos para a compra
Apropriado para	A maioria dos produtos	Livros, CDs, etc.

Tabela 10: Continuação da Comparação entre as técnicas de “Procura do Produto” (HE; JENNINGS; LEUNG, 2003).

	<i>Requisitos</i>
Quando usar a técnica	Algumas idéias do usuário são conhecidas
É requerido	Condições de satisfação para o produto
Interação com usuário	Média
Retorno de resultados	Produtos que satisfaça restrições particulares
Apropriado para	A maioria dos produtos

Com o produto e o mercado selecionados, inicializa-se o quarto estágio, chamado de “Negociação”, onde haverá a negociação entre os agentes envolvidos para determinar os termos do contrato. Para chegar a um contrato (acordo), os agentes negociam utilizando estratégias sofisticadas, baseadas em regras e funções de utilidade definidas individualmente.

O “Pagamento e entrega” é o quinto estágio onde serão determinadas a forma de pagamento (cartão de crédito, boleto bancário, etc.) e de entrega (avião/5 dias, navio/20 dias, eletronicamente/*on-line*, etc.).

Por fim, o sexto estágio, chamado “Avaliação”, será provido pelo vendedor, é o serviço de pós-venda, tendo assim, informações sobre a satisfação ou decepção do consumidor. Estas informações são de grande valia para o vendedor, pois permitem a calibragem dos aspectos como: preço, prazo de entrega, garantia, qualidade e outros que são influenciadores em um contrato de serviços ou produtos (GUTTMAN; MOUKAS; MAES, 1998).

Na Tabela 11 são apresentados alguns mercados eletrônicos e as respectivas funções, implementadas segundo os estágios do modelo CBB.

Tabela 11: Agentes Mediadores X CBB (FLOREA, 2002).

	Persona Logic	Bargain Finder	Jango	Kasbah	Auction Bot	T@T	Fish Market
Identificação de Necessidade	X		X				
Procura do Produto	X		X			X	
Procura do Mercado		X	X			X	
Negociação				X	X	X	X

Foi sugerido em He, Jennings e Leung (2003) um acréscimo nos estágios do CBB. Ele seria inserido entre o segundo e o terceiro estágios do CBB tradicional, com o nome de “Formação de cooperativa para compra”, e seria destinado à tarefa de procurar agentes com o mesmo interesse de compra para, juntos, fazerem um único pedido em quantidade maior, ganhando em poder de barganha (desconto por volume).

4.1.2.2 O Modelo BBT

Para prover um canal facilitador mais eficiente e ágil entre o complexo relacionamento entre as organizações, surgiu o mercado eletrônico B2B. O modelo BBT (*Business to business transaction*) tenta padronizar a forma de interação deste tipo de comércio eletrônico (B2B), identificando as regras que envolve os vários estágios das transações.

O modelo BBT (Figura 4.7) é formado por seis estágios: “Formação de parceria”, “Intermediação”, “Negociação”, “Elaboração do contrato”, “Execução do contrato” e “Avaliação do serviço”, sendo que, os mais implementados pelos agentes, devido às complexas características relacionadas à tomada de decisão, procura e acordo, são: “Formação de parceria”, “Intermediação” e “Negociação”, definidos a seguir (HE; JENNINGS; LEUNG, 2003).

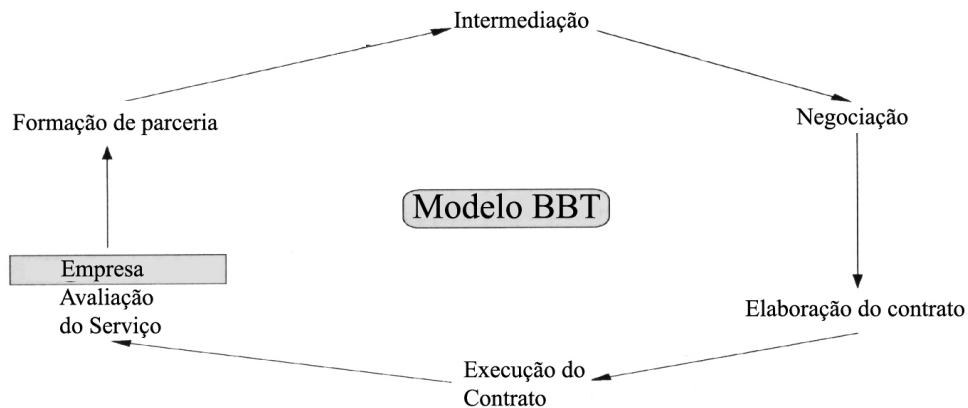


Figura 4.7: Modelo BBT (HE; JENNINGS; LEUNG, 2003).

Graças ao avanço na tecnologia da informação, hoje é possível para uma empresa procurar por parceiros em todo o mundo. Com isso, as empresas podem ser mais ágeis e competitivas, com a possibilidade de formarem empresas virtuais. Esta busca por parceiros é considerada o primeiro estágio chamado “Formação de parceria”.

No segundo estágio, chamado de “Intermediação”, há o encontro entre o vendedor, que oferece o produto ou serviço, e o comprador, que precisa do produto ou serviço. Para o vendedor, é o momento de propagar os seus produtos e procurar compradores em potencial. Já os compradores buscam o vendedor apropriado para atender às suas necessidades.

Com o aumento da pluralidade de ofertas sobre produtos/serviços e a diversidade das necessidades de compra entre as empresas, aumenta a necessidade da figura do intermediador entre a entidade que compra e a que vende. O intermediador, também chamado de corretor, responsável pelo encontro das empresas, vem sendo informatizado para agilizar e tornar mais eficaz este encontro. Os corretores podem ser agentes que implementam as funções de recuperar e processar informação, manter um repositório próprio de informações sobre o usuário, traçar perfil do usuário, monitorar itens de interesse do usuário, filtrar informações, prever as exigências do usuário, intermediar negociações entre consumidores e fornecedores e colaborar com outros corretores.

Após o encontro entre vendedores e consumidores, passa-se para o próximo estágio, chamado Negociação. Na negociação as partes irão tentar chegar a um acordo para fechar o negócio, ou seja, estabelecer um contrato. A negociação no B2B (BBT) é mais complexa se comparada ao B2C (CBB), pois envolve maior volume, transações repetidas e termos de contratos mais complexos.

Os três estágios já apresentados, “Formação de parceria”, “Intermediação” e “Negociação”, são os mais implementados pelos agentes, pois envolvem funções (tomada de decisão, procura, acordos) que são especialidades dos agentes eletrônicos. No entanto, os três estágios restantes, “Elaboração do contrato”, “Execução do contrato” e “Avaliação”, não são comumente implementados com agentes eletrônicos por enfrentarem barreiras técnicas, culturais e até mesmo legais, e por isso não serão explorados.

4.1.3 O Mercado Brasileiro

O mercado eletrônico é sucesso no mundo desde o seu surgimento. Este sucesso vem aumentando a cada ano e deve continuar em ritmo acelerado, pois existe um espaço grande para o crescimento.

No Brasil não é diferente, e a cada ano o volume de negócios aumenta de forma imprecionante. Pode-se perceber este crescimento através das várias pesquisas feitas em diferentes tipos de mercado.

Segundo a 6^a pesquisa da Fundação Getúlio Vargas-FAESP (ALBERTIN, 2004) sobre comércio eletrônico no mercado brasileiro, os valores movimentados no B2C e B2B eletrônico das empresas que atuam no mercado brasileiro atingiram cerca de 16,3 bilhões de dólares em 2003. Deste total, 11,8 bilhões referem-se a B2B eletrônico e 4,5 bilhões a B2C eletrônico, representando 4,94% de todo o mercado B2B e 2,08% de todo o mercado B2C.

Analisando estes dados e comparando-os aos de 2002, onde o B2B eletrônico movimentou 5,7 bilhões de dólares e o B2C eletrônico 1,9 bilhões de dólares, pode-se retirar duas informações importantes: a primeira, que o movimento de 2003 aumentou 2,07 vezes no B2B eletrônico e 2,36 vezes no B2C eletrônico; e a segunda, que o volume movimentado nos mercados eletrônicos em relação ao montante total ainda é pequeno, e por isso o espaço para crescimento futuro é grande.

Outros dados desta mesma pesquisa, coletados entre 435 empresas de vários setores econômicos, ramos e portes que operam no mercado brasileiro, são relevantes: a indústria

foi a que mais utilizou do B2B eletrônico, 5,37% do total de transações B2B, seguida do ramo de serviços (4,87%) e comércio (3,97%). Entre os consumidores (B2C), a *internet* foi mais utilizada pela indústria (2,50%), contra 1,42% dos serviços e 1,09% do comércio.

Com relação aos meios de pagamento, os mais utilizados pelos consumidores *on-line* foram: o cartão de crédito com 33%, seguido pelo *smart card* (10%), o *e-check* (9%) e o *e-cash* (8%).

Outra pesquisa (INFO100, 2003), publicada pela revista INFO, também mostra que o comércio eletrônico expande rapidamente no Brasil. Segundo a pesquisa, os 100 maiores nomes do comércio eletrônico movimentaram 394 bilhões de reais em 2003, mais que o triplo do volume registrado pela mesma revista no ano anterior. Os negócios das 43 maiores empresas de B2B aumentaram 60% com relação à pesquisa anterior. No B2C, a confiança do consumidor aumentou, elevando em 78% os negócios das 40 maiores no B2C, em relação a 2002. O maior crescimento apontado pela pesquisa, foi o setor financeiro, com um aumento de 180% no volume movimentado no ano de 2003 comparado a 2002.

A 5ª edição da pesquisa INFO 100 (INFO100, 2003), publicou as 100 maiores empresas em volume de dinheiro movimentado no comércio eletrônico brasileiro, das quais pode-se citar as 10 melhores do B2c, B2B e setor financeiro:

- B2C: a primeira foi a General Motors seguida de Fiat, Ford, Telefônica São Paulo, Visanet, Americanas.com, Gol Linhas Aéreas, Submarino, Magazine Luiza e Grupo Pão de Açúcar.
- B2B: em primeiro ficou a Ford seguida de General Motors, Genexis, Tv Globo, Ticket Serviços, Petróleo Ipiranga, Toyota, Basf, Salutia e Goodyear.
- Setor Financeiro: em primeiro está o Bradesco, seguido do Banco Itaú, Banco do Brasil, Unibanco, Caixa Econômica Federal, Grupo Santander Banespa, Ágora Senior, BankBoston, ABN AMRO Real e Porto Seguro.

Graças a este crescimento vertiginoso, o Brasil foi considerado o 35º melhor país para comércio eletrônico segundo o relatório sobre comércio eletrônico elaborado pela *The Economist Intelligence Unit Limited* (ECONOMIST; IBM, 2004) (centro de estudo do mesmo grupo da revista *The Economist*) e a *IBM Corporation*.

4.2 Negociação

Um dos fatores mais importantes de um Sistema Multiagente é a capacidade de interação social dos agentes, ou seja, o relacionamento entre os mesmos.

Os agentes relacionam-se de várias formas, porém, quando se trata de objetivos, os agentes podem ir de completamente cooperativos a antagônicos. Qualquer que seja o relacionamento quanto ao objetivo, a negociação exerce um papel fundamental. Para os cooperados, a negociação é responsável por distribuir as tarefas e necessidades. Já para os antagônicos, ela desempenha a função de conciliação, para que as partes cedam e encontrem a satisfação. Portanto, o objetivo final de um processo de negociação é o acordo. Seja antagônica ou cooperada, a negociação é uma ferramenta para uma decisão em comum.

Em uma negociação automatizada, que é o foco deste trabalho, os agentes autônomos devem preocupar-se com três componentes principais: o protocolo, que define as regras do relacionamento; as dimensões que estão sendo negociadas (preço, data da entrega, forma de pagamento) e a estratégia, que define as funções e seus pesos que um agente aplicará para conseguir, dado um protocolo, atingir o seu objetivo na negociação. Todos estes componentes serão tratados neste trabalho.

O campo da negociação automatizada é bastante atraente e diversificado, quanto às definições. Por isso várias pesquisas foram feitas, resultando em diferentes paradigmas. Um dos paradigmas mais utilizado é o leilão, que consiste de regras pré-definidas, grande simplicidade e quase sempre envolve uma única dimensão a negociar: o preço. Outro paradigma é a teoria dos jogos, que só pode ser aplicada em contexto com informações e racionalidade completa. Em contrapartida, existe o paradigma de agentes inteligentes, que é mais flexível e mais adequado para ambientes abertos e dinâmicos (ROCHA; OLIVEIRA, 2000). Por isso, parece o mais adequado e é o escolhido para este trabalho.

4.2.1 A Classificação da Negociação

Em um contexto de transações comerciais existem dois tipos de negociação (GUTTMAN; MAES, 1998):

- Competitiva ou Distribuída: pode ser definida como o processo de tomada de decisão, para resolução de conflitos, envolvendo as partes em torno de um objetivo único e mutuamente exclusivo.

- Cooperativa ou Integrada: definida como o processo de tomada de decisão envolvendo as partes em torno de vários objetivos interdependentes, mas não mutuamente exclusivos.

4.2.2 Características e Parâmetros do Espaço de Negociação

Para projetar e implementar um cenário de negociação apropriado, é necessário observar o contexto com suas particularidades e identificar as características e parâmetros de uma negociação que se adeque. Segundo Lomuscio, Wooldrige e Jennings (2001), uma negociação tem as seguintes características e parâmetros: cardinalidade da negociação, características dos agentes envolvidos nas transações, características do objeto a ser negociado e o ambiente, parâmetros de eventos, parâmetros da informação e parâmetros de alocação.

4.2.2.1 Cardinalidade da Negociação

Existem dois tipos de cardinalidade em uma negociação. A primeira é em relação ao domínio e a segunda em relação às interações.

Quanto ao domínio, podem-se identificar negociações com um único atributo ou múltiplos atributos, sendo que os atributos são as dimensões que integram uma negociação. Alguns exemplos de atributos são: preço, prazo de entrega e forma de pagamento.

Já quanto às interações, a classificação é dada segundo o número de agentes que participam da negociação. Ou seja, na forma *One-to-One* (Um-para-Um) a negociação é feita somente entre dois agentes que interagem, sem interferência de outros agentes. Na forma *Many-to-One* (muitos-para-um) existem muitos agentes que negociam com um único oponente. Por fim, *Many-to-Many* (muitos-para-muitos), onde muitos agentes negociam com muitos outros oponentes. É importante ressaltar que a cardinalidade de interações não depende de quantos agentes existem no sistema, e sim quantos agentes participam das interações em uma negociação.

4.2.2.2 Características dos Agentes

Considerando cada agente como uma entidade computacional em um processo de negociação que, dotado de capacidade de identificar suas preferências, é capaz de fazer uma avaliação e posterior escolha de diferentes negócios, podem-se identificar algumas dependências para um agente participar da negociação:

- Regra: identifica o tipo de comportamento e identidade de um agente. Por exemplo, podem ser compradores ou vendedores em uma negociação ou compradores e vendedores dependendo da negociação;
- Racionalidade ou Sensatez: pode ser dividida entre perfeita ou limitada. Na perfeita é permitido ao agente usar de recursos computacionais, em um tempo constante, de forma arbitrária e sem limites, o que a torna não viável. No entanto, a limitada é mais realista e considera um consumo computacional menor;
- Conhecimento: faz diferença na escolha de uma estratégia, ou seja, dependendo da forma que as informações são distribuídas, os agentes podem escolher diferentes estratégias. O conhecimento em questão refere-se aos produtos a serem negociados e a valorização destes quanto à função utilidade do oponente;
- Comprometimento: vários tipos de comprometimentos podem estar presentes. Por exemplo, após fazer uma oferta, o agente necessariamente é obrigado a parar negociações com outros negociadores sobre o mesmo produto, até receber o aceite ou contra-oferta. Outro exemplo, é obrigar o agente vendedor a fazer uma reserva do produto em negociação até o término da mesma e, após, liberar ou faturar o produto;
- Comportamento Social: o agente pode agir de forma individualista ou de forma cooperada. Este posicionamento pode ser alterado de acordo com os interesses do agente;
- Estratégia das interações (lances): a estratégia é a responsável pela tomada de decisão. Conceitualmente independente de outros parâmetros, na prática é razoável pensar na escolha de uma estratégia como algo relacionado com o comprometimento, o conhecimento, a racionalidade e o comportamento social de um agente oponente.

4.2.2.3 Ambiente da Negociação

O ambiente pode ir de estático a dinâmico. No estático as variáveis de ambiente (por exemplo, tempo limite) são constantes, e no dinâmico elas alteram com o passar do tempo.

Existem vários graus de quanto um ambiente é estático ou dinâmico. Tudo depende de como se comportam as variáveis, ou seja, pode-se ter ambientes com algumas variáveis estáticas e outras dinâmicas, sendo assim, um ambiente nem totalmente estático nem totalmente dinâmico.

4.2.2.4 Parâmetros de Eventos

Um protocolo de negociação é influenciado pela forma pela qual as ofertas e outros eventos que ocorrem durante a negociação são regulados. Na especificação de um protocolo, alguns eventos são considerados:

- Validação dos lances: definir os critérios para validação dos lances. Para ser válido, o lance deve estar no tempo aceitável e satisfazer algumas restrições. Um exemplo é o leilão inglês, em que os lances só podem ser feitos quando o leiloeiro autorizar, e sempre com o valor maior que o lance anterior.
- Visibilidade dos lances: é relevante para casos que envolvem *Many-to-One* (muitos-para-um) ou *Many-to-Many* (muitos-para-muitos). Existem três tipos de visibilidade: privada, aberta e conjunta. Na privada, a mensagem é passada entre o comprador e o vendedor diretamente sem intermediários e interferências. Já no aberto, chamado de *broadcast*, todos agentes de uma negociação têm acesso ao lance feito. Por último a conjunta, onde um grupo restrito de agentes recebe as mensagens.
- Agendamento de cotações: o número e a frequência de pedidos de cotações podem afetar o desempenho das negociações, pois se um agente tem que negociar com uma quantidade grande de agentes, automaticamente terá que dividir seus recursos computacionais.

4.2.2.5 Parâmetros de Informação

Antes e durante uma negociação, informações podem ser trocadas entre os agentes, através de mensagens, na tentativa de agilizar o processo de negociação e, com isso, reduzir também o consumo computacional. Estas mensagens podem ser informações que ajudam compradores e vendedores a chegarem ao acordo e diminuir o ruído (informações não entendidas) entre os mesmos.

4.2.2.6 Parâmetro de Alocação

Pode ser aplicada nos cenários *Many-to-Many* (muitos-para-muitos) e *Many-to-One* (muitos-para-um). A alocação determina o vencedor quando mais de um agente mostrou interesse, através de ofertas, no produto. O parâmetro de alocação é estudado e usado em cenário de leilão, por isso é de grande interesse da teoria de leilão.

5 *A Aplicação SMANACE*

O produto final deste trabalho foi um sistema (protótipo), com o qual os usuários criam seus agentes e os colocam no mercado eletrônico para defender seus interesses e alcançar os objetivos. O usuário pode optar por criar agentes vendedores ou compradores que, de forma autônoma, irão negociar para obter acordos sobre os serviços que procuram ou oferecem.

A aplicação disponibiliza duas etapas do modelo CBB (Seção 4.1.2.1):

- Seleção do mercado/vendedor: o comprador procura por um vendedor que possa oferecer o serviço desejado.
- Negociação: o processo de negociação para atingir o acordo, uma vez que compradores e vendedores estão com interesses opostos.

5.1 O Mercado

O mercado nesta aplicação tem o papel de centralizador dos encontros, ou seja, é no mercado que acontece a seleção do vendedor pelo comprador. Sendo assim, pode-se chamar o mercado de páginas amarelas, pois todo comprador que deseja encontrar um vendedor recorre ao mercado para descobrir quais são os vendedores que tem o produto para atendê-lo.

5.2 O Modelo de Negociação

O modelo que serviu de base para a aplicação foi o modelo de negociação orientado ao serviço (*service-oriented negotiation model*) (FARANTIN, 1998). Neste modelo, os papéis estão a princípio em conflito, ou seja, compradores querendo preço mais baixo, menores prazos de entrega e maior prazo para pagamento e, do outro lado, vendedores buscando maior preço para venda, maior prazo de entrega e menor prazo para pagamento.

Este modelo é caracterizado por ser multilateral e multidimensional (ver Seção 4.2.2.1). A aplicação implementou estas características da seguinte forma:

- Multilateral: os agentes, vendedores e compradores, têm a capacidade de negociar simultaneamente com vários oponentes. Desta forma, um comprador pode, ao mesmo instante, negociar com vários vendedores sobre o mesmo produto, e fazer comparativos possibilitando a descoberta da melhor oferta. Já o vendedor pode usar da multi-lateralidade como mecanismo de defesa, pois estará observando as ofertas de vários interessados em suas mercadorias.
- Multi-dimensão: a negociação ocorre sobre três dimensões: preço, prazo de entrega e forma de pagamento. Ao contrário de muitas aplicações de comércio eletrônico que analisam somente o fator preço, a aplicação SMANACE permite ampliar as dimensões proporcionando maior satisfação e poder de negociação, o que facilita o acordo. Desta forma, uma proposta consiste da valoração destas três dimensões.

Nem sempre os interesses sobre uma das dimensões são opostos. Nestes casos recomenda-se retirar a dimensão da negociação, pois esta tende ao valor máximo, uma vez que ambos os lados desejam a maximização (FARANTIN, 1998).

O processo de negociação ocorre com a troca de ofertas e contra-ofertas, formando o fluxo de negociação. Este fluxo pode ser representado como uma seqüência finita de tamanho n na forma $(x_{a \rightarrow b}^{t_1})$ onde $t_1, t_2, t_3, \dots, t_n \in Time$ (Tempo) e $a, b \in AGENTES$ (conjunto de agentes). As gerações das ofertas e contra-ofertas são feitas através de combinações lineares de funções chamadas táticas, que usam critérios como tempo, recursos e comportamento para gerar valor para cada dimensão (preço, prazo de entrega, forma de pagamento). Cada dimensão recebe um peso (w_j^i) para diferenciar a importância entre elas.

A combinação das táticas gera a estratégia, ou seja, dentre as várias táticas aplicáveis pode-se atribuir pesos a cada uma delas e montar uma estratégia diferente, que irá guiar a tomada de decisão quanto à formação de ofertas.

Em uma negociação com n dimensões, para cada dimensão $j \in \{1, 2, \dots, n\}$, cada agente i define:

- Uma gama de valores aceitáveis $[min_j^i, max_j^i]$;
- Uma função de pontuação (função utilidade) $V_j^i : [min_j^i, max_j^i] \rightarrow [0, 1]$;

- Um peso w_j^i ($\sum_{1 \leq j \leq n} w_j^i = 1$, para todo $i \in AGENTES$).

A função utilidade é responsável pela avaliação das propostas. Quanto maior o valor de retorno da função, melhor é a proposta para aquele agente. A função utilidade de um agente i que avalia uma proposta x , pode ser definida como:

$$V^i(x) = \sum_{1 \leq j \leq n} w_j^i V_j^i(x_j) \quad (5.1)$$

Os valores podem ser positivos ou negativos. Neste último caso, a proposta analisada tem valores de uma ou mais dimensões fora da faixa aceitável definida para o agente. Pode-se definir a função utilidade de um agente i para uma dimensão j como mostrado nas equações 5.2 e 5.3.

$$V_j^i(x_j) = \frac{x_j - \min_j^i}{\max_j^i - \min_j^i} \quad (5.2)$$

ou

$$V_j^i(x_j) = 1 - \left(\frac{x_j - \min_j^i}{\max_j^i - \min_j^i} \right) \quad (5.3)$$

Percebe-se que a utilização de uma função ou da outra depende do objetivo de maximizar (Equação 5.2) ou minimizar (Equação 5.3) o valor da dimensão.

5.3 Os Protocolos de Negociação

O protocolo define as “regras do jogo”, ou seja, como deve proceder um agente após ter definido seu tipo. Uma vez definido como comprador, o agente segue o protocolo de negociação para agir de forma correta (maximizar utilidade) no processo de negociação.

Neste trabalho são implementados dois protocolos. O primeiro é o protocolo relatado em Farantin (1998), e o segundo um protocolo criado para este trabalho.

5.3.1 O Protocolo referenciado por Farantin (1998)

Cria-se um agente com objetivo de compra ou de venda que terá um tempo máximo (t_{max}) para alcançar o objetivo. Portanto, um agente negocia com outro até que chegue a um acordo ou alcance o tempo limite ($t_n = t_{max}$) ou ultrapasse o tempo limite ($t_n > t_{max}$).

Nesta implementação, antes de iniciar a negociação o agente identifica um possível interessado e envia a ele um pedido de cotação. Caso o vendedor tenha o produto no estoque, faz uma proposta, iniciando, assim, o fluxo de negociação. Ao receber uma proposta do fluxo, o agente a analisa, através da função de interpretação, e toma a decisão de qual ação executar.

Dado um agente a e sua função de pontuação V^a em um tempo t' , uma oferta $x_{b \rightarrow a}^t$ enviada no tempo $t < t'$ é interpretada pela função I^a :

$$I^a(t', x_{b \rightarrow a}^t) = \begin{cases} \text{rejeitar} & \text{Se } t' > t_{max}^a \\ \text{aceitar} & \text{Se } V^a(x_{b \rightarrow a}^t) \geq V^a(x_{a \rightarrow b}^{t'}) \\ \text{enviar} & x_{a \rightarrow b}^{t'} \text{ outros casos} \end{cases} \quad (5.4)$$

sendo que, $x_{a \rightarrow b}^{t'}$ é a contraproposta gerada por a para o agente b em um tempo t' .

Na Figura 5.1 estão mostrados os passos possíveis (função de interpretação) do agente após receber uma proposta.

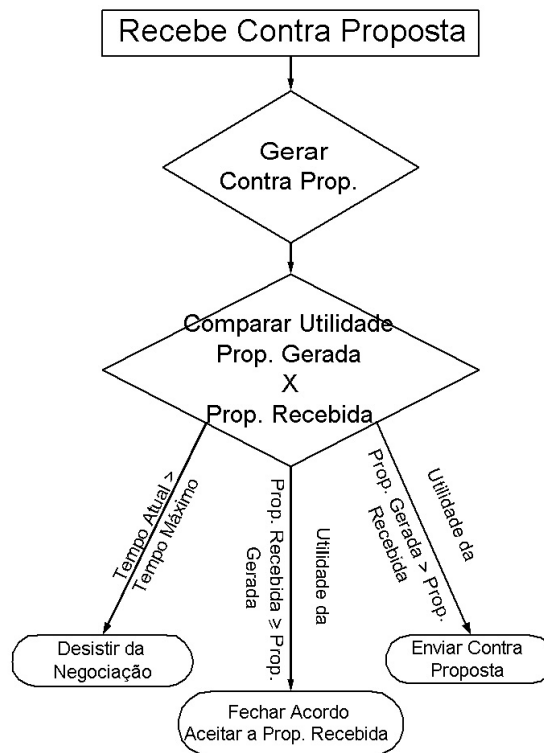


Figura 5.1: Protocolo Farantin para compradores e vendedores.

Segundo Farantin (1998), quando a negociação termina, o fluxo de negociação tem

como seu último elemento um {aceitar} ou {rejeitar}. No entanto, para esta aplicação foi levado em consideração o fato de existirem vários agentes participando da negociação e, portanto, faz-se necessária uma confirmação do agente receptor do {aceitar}, como mostra a Figura 5.2, para evitar problemas causados pela possível desistência do mesmo.

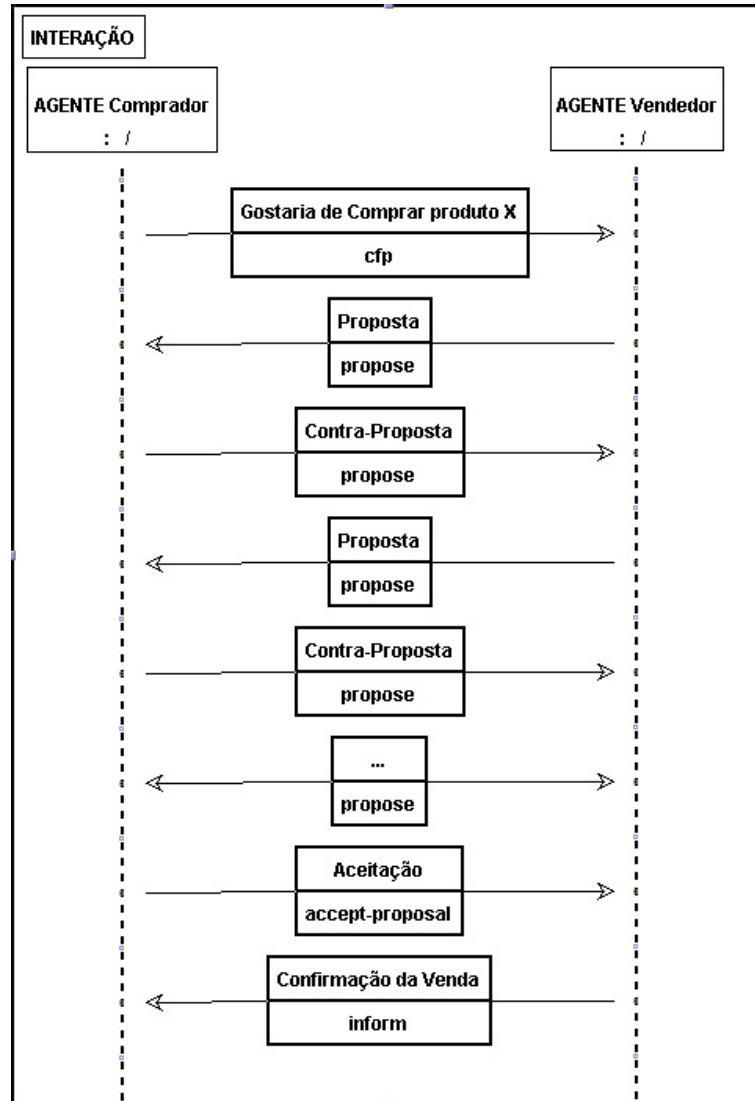


Figura 5.2: Exemplo de interação entre dois agentes.

5.3.2 O Protocolo CAMILO

No protocolo desenvolvido neste trabalho, doravante denominado protocolo CAMILO, o agente, no momento da criação, é tipificado como sendo vendedor ou comprador e define os intervalos (valor máximo e valor mínimo) das dimensões preço, prazo de entrega e forma de pagamento.

A definição de um tempo máximo para a negociação (t_{max}) é feita por parte do

comprador que detém um poder maior na negociação. O comprador também define o momento de parada na negociação, ou seja, dentro de um fluxo de negociação é o comprador que define o {fechar acordo} ou {cancelar negociação}. Desta forma, ambos, comprador e vendedor, analisam as propostas, mas somente o comprador detém o poder de fechar a negociação, restando ao vendedor decidir apenas pela entrada ou não na transação. A implementação deste maior poder operacional ao comprador é um dos pontos que diferencia este protocolo do anterior. A intenção em aumentar este poder é aproximar o protocolo do contexto real (mundo real) de uma transação comercial.

Outro ponto diferente entre os dois protocolos é que o protocolo CAMILO aproveita todo o tempo disponível para negociar, ou seja, se o comprador receber uma proposta boa e o tempo corrente (t_n) for menor que o t_{max} , ele não irá fechar o acordo, pois ele acredita que no tempo restante é possível receber melhores propostas. Para isso o vendedor tem que sustentar qualquer proposta feita até o fim da negociação, pois o comprador pode aceitar uma proposta feita em um instante anterior (por exemplo t_{n-4}) ao corrente (t_n).

Um outro ponto pode ser identificado como sendo diferente entre os protocolos apresentados. O protocolo CAMILO, prevendo a possibilidade de existir mais de um oponente na negociação (negociação muitos-para-muitos), utiliza uma variável de armazenamento para guardar a melhor proposta recebida entre os oponentes. Isso possibilita ao agente enviar uma proposta recebida de um oponente para outro, caso a mesma seja a melhor proposta recebida e seja melhor que a contra-proposta gerada pelo agente. Com isso, o agente cria uma concorrência entre os oponentes.

Após ser tipificado, definidos os intervalos de valores aceitáveis das dimensões, definidos o t_{max} e o produto a ser comprado, o usuário da aplicação dá autonomia ao agente comprador que irá entrar no mercado procurando um vendedor para negociar. Após encontrar vendedores, o comprador faz um pedido de cotação para cada vendedor, que pode ser respondido com as informações sobre o produto (e.g. preço, prazo de entrega e forma de pagamento) ou um informativo de não participação na negociação.

5.3.3 A Evolução do Protocolo CAMILO

Tomando como base o protocolo refenciado por Farantin (1998) e almejando o aprimoramento, surgiu a primeira versão do protocolo CAMILO para Comprador (Figura 5.3) e Vendedor (Figura 5.4).

Com o grande aumento da complexidade em relação ao protocolo base, foi necessário

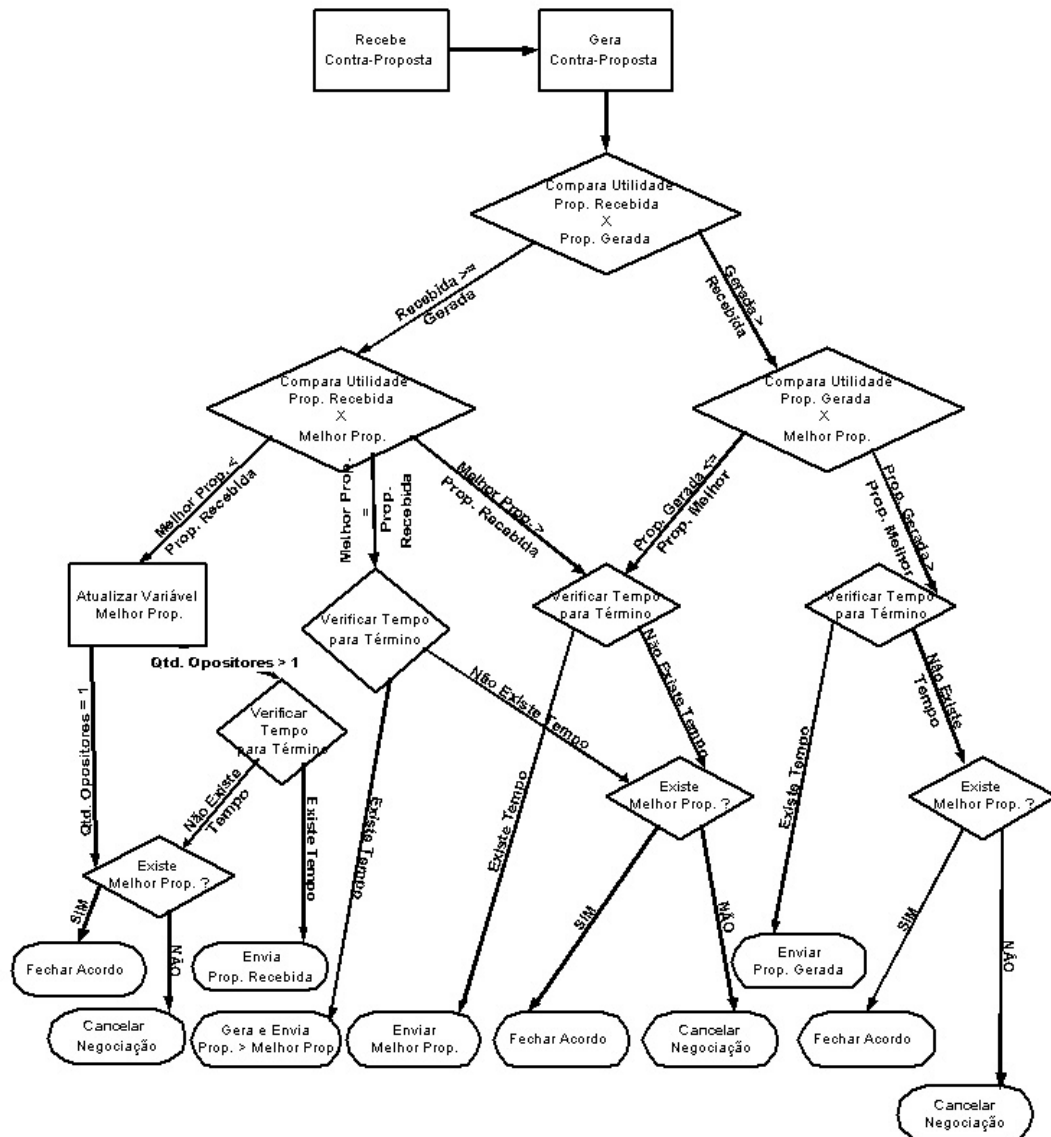


Figura 5.3: Protocolo CAMILO para agentes compradores (Versão 1).

a implementação do algoritmo para executar simulações e viabilizar uma análise ainda mais criteriosa. Durante as simulações foram identificados três necessidades de alteração do protocolo CAMILO.

A primeira alteração foi feita no protocolo CAMILO Comprador, onde foi necessário alterar o retorno quando a utilidade da proposta recebida de um vendedor for menor que a utilidade da melhor proposta já recebida. O retorno, nestes casos, deixa de ser a melhor proposta para ser uma proposta criada para ser melhor do que a melhor. Esta alteração foi necessária para inibir a possibilidade que o comprador passe uma proposta para o vendedor igual à recebida de outro vendedor, evitando assim que o vendedor atinja

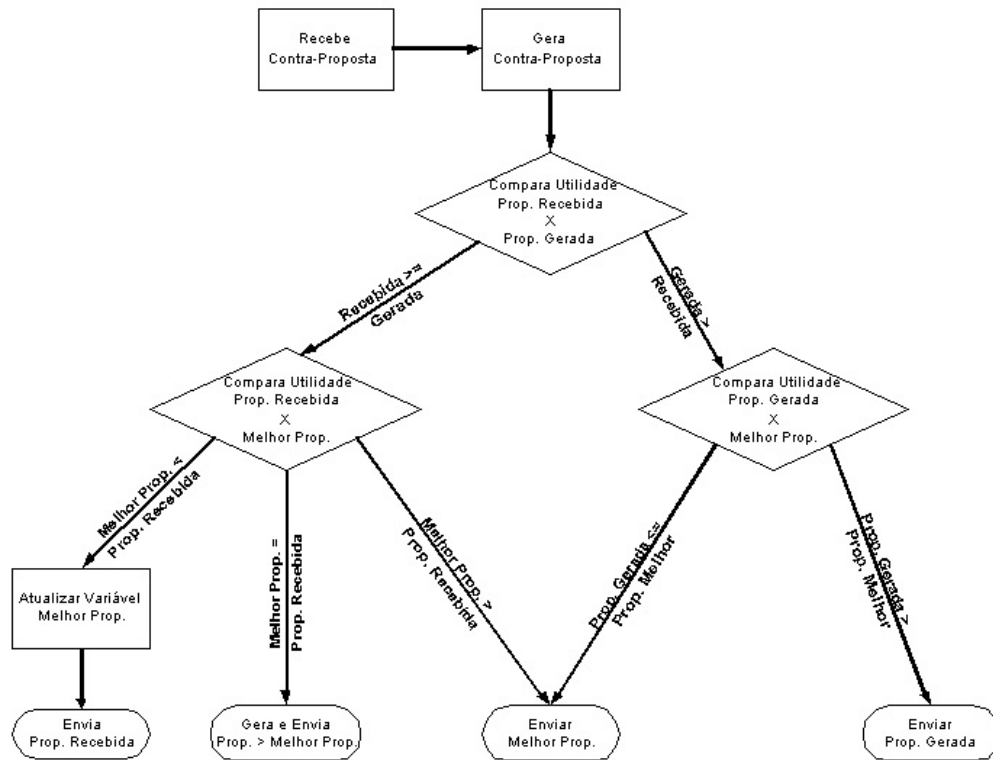


Figura 5.4: Protocolo CAMILO para agentes vendedores (Versão 1).

a utilidade pedida pelo comprador e mesmo assim não ganhe a concorrência.

A segunda alteração foi feita no protocolo CAMILO Vendedor, onde a melhor proposta foi substituída pela melhor proposta depreciada. A depreciação da melhor proposta é necessária para evitar que o vendedor torne-se rígido e não ceda quando receber uma melhor proposta.

A terceira alteração, também efetuada no protocolo CAMILO Vendedor e provocada pela segunda alteração, elimina a geração e envio de uma proposta com utilidade maior que a melhor proposta recebida.

Após as alterações, a versão final dos protocolos que foram implementados foram: para Comprador (Figura 5.5) e para Vendedor (Figura 5.6).

Sendo assim, o comprador que recebe alguma proposta, das cotações ou contra-proposta enviadas, ele irá analisá-la, segundo o diagrama de interpretação definido na Figura 5.5. E o vendedor que recebe alguma proposta, das propostas ou contra-propostas enviadas, ele irá analisá-la, segundo o diagrama de interpretação definido na Figura 5.6.

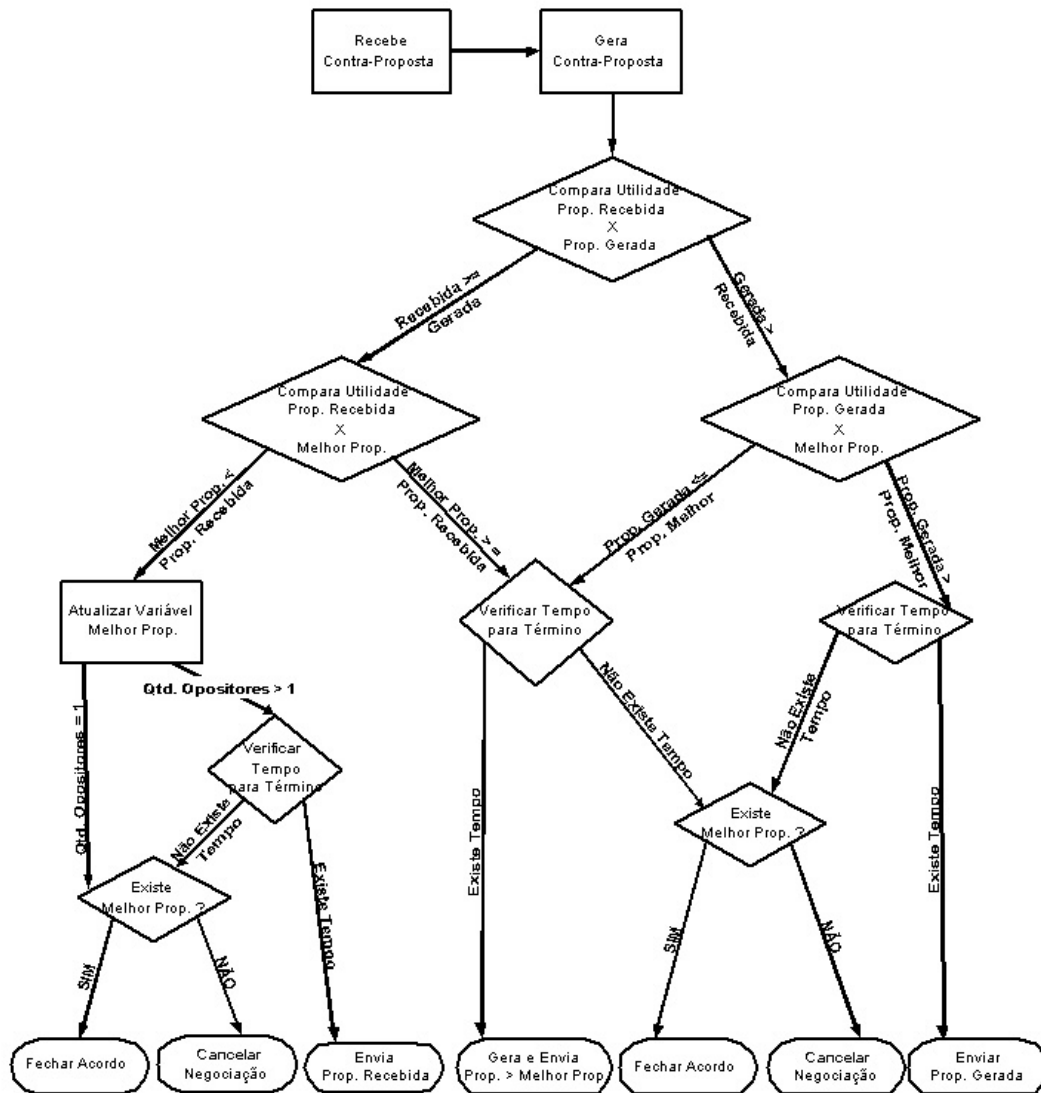


Figura 5.5: Protocolo CAMILO para agentes compradores.

5.4 Táticas e Estratégias

Para gerar uma proposta ou contra-proposta, o agente é orientado (guiado) por uma estratégia (plano de ação), que por sua vez é uma combinação linear das táticas. Um exemplo de estratégia seria a combinação linear de duas táticas ($T1$ e $T2$): $2T1 + 3T2$. Já as táticas são funções que usam, como critérios tempo, recurso ou comportamento para determinar valores para cada dimensão de uma proposta. Portanto, existem três grandes famílias de táticas: dependente do tempo, dependente de recurso e dependente do comportamento (FARANTIN, 1998).

- Dependente do Tempo: há uma variação das propostas de acordo com o tempo que

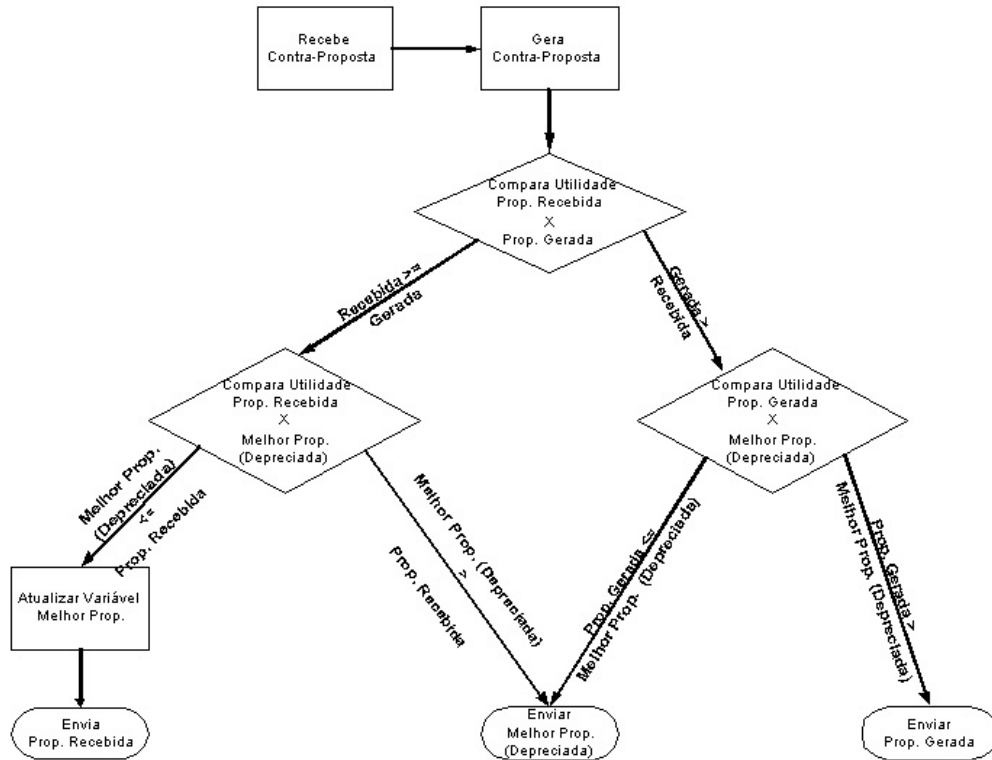


Figura 5.6: Protocolo CAMILO para agentes vendedores.

os agentes dispõem para negociar. À medida que decresce o tempo restante, mais se concede.

- Dependente de recurso: as propostas oscilam com base na disponibilidade de um determinado recurso que não o tempo.
- Dependente do comportamento: tenta-se imitar o comportamento do oponente. A imitação pode ter várias intensidades, ou seja, imitação proporcional, imitação absoluta e imitação proporcional média.

Das famílias existentes, a aplicação implementa uma tática dependente do tempo de concessão polinomial.

5.4.1 Dependente do Tempo

Nesta grande família de táticas, há predominância do fator tempo, ou seja, é a quantidade de tempo restante ($t_{max}^i - t_n$) que determina os valores das dimensões a serem propostas.

O valor inicial é determinado pela soma entre o menor valor aceitável das dimensões e a multiplicação entre uma constante K_j^i , sendo i o agente e j a dimensão, e o intervalo de valores aceitáveis ($max_j^i - min_j^i$). Portanto, a equação do valor inicial é a 5.5 com a substituição da função $\alpha_j^i(t_n)$ pela constante K_j^i .

Para determinar uma proposta do agente i para o agente h , no intervalo de tempo $0 \leq t_n \leq t_{max}^i$ para qualquer tática de família dependente do tempo, deve-se usar a Equação 5.5:

$$x_{i \rightarrow h}^{t_n}[j] = \begin{cases} min_j^i + \alpha_j^i(t_n)(max_j^i - min_j^i) & \text{Se } V_j^i \text{ está decrescendo} \\ min_j^i + (1 - \alpha_j^i(t_n))(max_j^i - min_j^i) & \text{Se } V_j^i \text{ está aumentando} \end{cases} \quad (5.5)$$

Sendo que V_j^i é a função utilidade da dimensão j no agente i e α_j^i é o resultado de uma das táticas.

O valor de α_j^i é definido pelas condições $0 \leq \alpha_j^i(t_n) \leq 1$, o $\alpha_j^i(0) = K_j^i$ e $\alpha_j^i(t_{max}^i) = 1$. Desta forma os valores das dimensões irão oscilar entre o valor inicial, determinando pela constante K_j^i , até o valor de reserva (max_j^i).

Existem dois tipos de táticas já definidas (FARANTIN, 1998): polinomial (Equação 5.7) e exponencial (Equação 5.6). Ambas são parametrizadas por um valor $\beta \in \mathfrak{R}^+$ que determina o grau de convexidade da curva. Para valores altos de β , a função polinomial concede mais rapidamente no começo do que a exponencial (Figura 5.7). Para valores pequenos de β , a função exponencial espera mais para conceder do que a polinomial (Figura 5.8).

$$\text{Exponencial} : \alpha_j^i(t_n) = e^{(1 - \frac{\min(t_n, t_{max}^i)}{t_{max}^i})^\beta \log K_j^i} \quad (5.6)$$

$$\text{Polinomial} : \alpha_j^i(t_n) = K_j^i + (1 - K_j^i) \left(\frac{\min(t_n, t_{max}^i)}{t_{max}^i} \right)^{\frac{1}{\beta}} \quad (5.7)$$

Portanto, existe uma variedade muito grande de táticas, pois para cada valor de β é possível formar uma tática diferente. Para valores de $\beta > 1$, as táticas são classificadas como concedentes e para valores de $\beta < 1$, são classificadas como não concedentes.

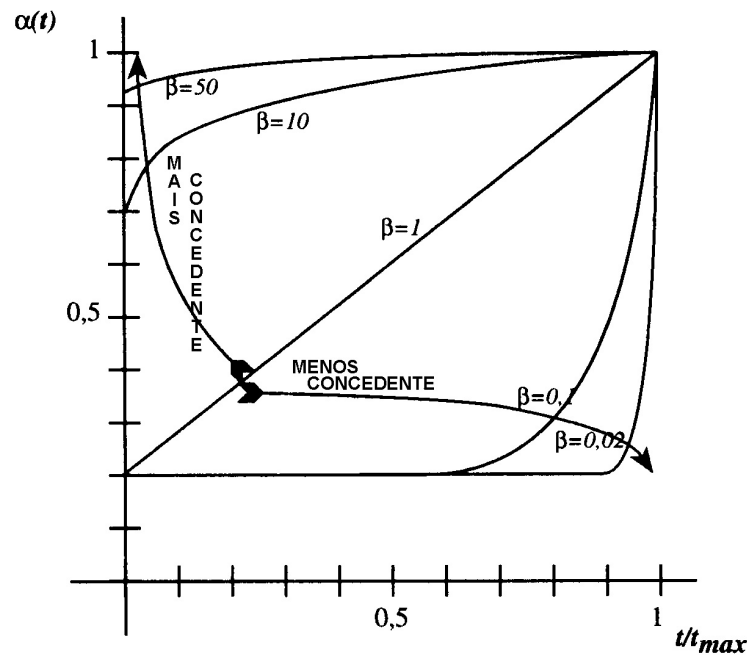


Figura 5.7: Relação de $\alpha(t)$ com o $\frac{t}{t_{max}}$ na família polinomial (FARANTIN, 1998).

5.5 A Plataforma e a Implementação

Após definir toda parte teórica da aplicação, nesta seção, será mostrada a forma como foram implementadas as trocas de mensagens e a estrutura base que sustentou o desenvolvimento do SMANACE (Sistema Multiagente de Negociação Autônoma para o Comércio Eletrônico).

5.5.1 A Plataforma

Foi feita uma busca por uma plataforma que facilitasse o desenvolvimento do sistema. Esta facilitação envolve principalmente a parte social que traria um ganho de tempo, pois reduziria o código, e de qualidade, pois a plataforma já estaria exaustivamente testada.

Dentre as plataformas pesquisadas, a que mais supriu as necessidades foi a JADE (Seção 3.7.2). Como esta é uma plataforma que segue a FIPA (Seção 3.7.1), a linguagem usada foi a ACL. Uma das facilidades do JADE é a existência de agentes já criados que fazem funções de páginas amarelas e páginas brancas, ou seja, para a aplicação que necessita de um ponto centralizador que sirva de encontro para os agentes, encontra-se nestes agentes características que facilitam a construção deste ambiente chamado mercado. Com as páginas-amarelas os vendedores podem oferecer seus serviços e os compradores podem encontrar os fornecedores para determinados serviços. Com as páginas brancas

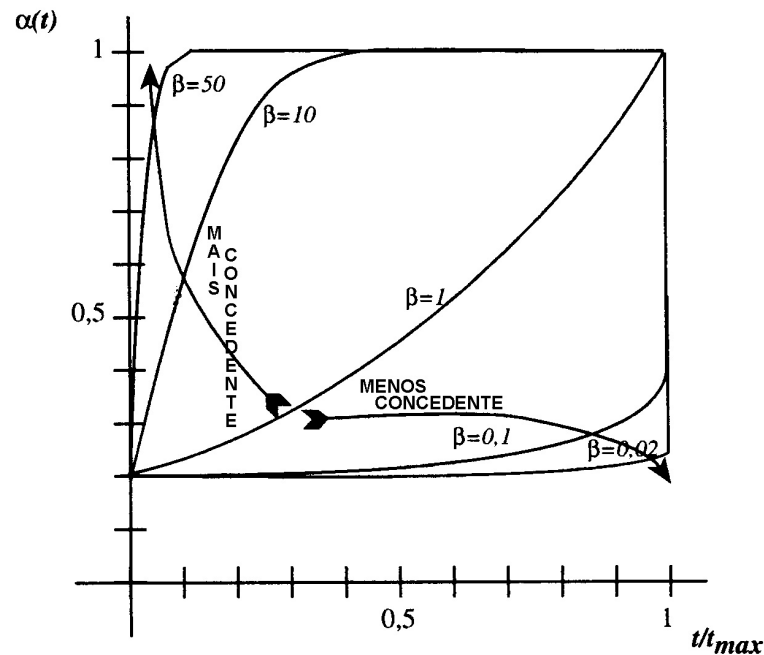


Figura 5.8: Relação de $\alpha(t)$ com o $\frac{t}{t_{max}}$ na família exponencial (FARANTIN, 1998).

os compradores e vendedores podem encontrar o endereço do agente para a troca de mensagens.

5.5.2 A Implementação

Como a plataforma escolhida (JADE) é implementada em Java, o sistema SMANACE foi desenvolvido também todo em Java.

O sistema é composto de dez classes principais. Duas destas, “Agente Comprador NP.java” e “AgenteVendedor.java”, são classes estendidas da classe *Agent* da plataforma JADE e responsáveis por determinar a seqüência de comportamento dos agentes compradores e vendedores no mercado.

Uma outra classe, chamada “CadastraEstoque.java”, é a responsável por cadastrar os estoques dos vendedores, ou seja, a classe é executada e recebe como parâmetros o nome do produto, preço máximo, preço mínimo, importância da dimensão em relação às outras dimensões e a constante K_j^i . Todos estes parâmetros, exceto o nome do produto, são recebidos para três dimensões: preço, forma de pagamento e prazo de entrega. Para armazenar estes valores em arquivos e melhor estruturar a aplicação, foram criadas duas classes, “ItemEstoque.java” e “Estoque.java”, que contêm a estrutura necessária como: dados do item de estoque (preço, forma de pagamento e prazo de entrega) e dados de estoque (quantidade do item no estoque).

A classe “CadastraPerfilComprador.java”, após executada, parametriza as dimensões dos compradores como: preço, forma de pagamento e prazo de entrega. São preenchidos para cada dimensão, o intervalo de valores aceitáveis, a constante K_j^i e a importância de cada dimensão em relação às outras. Para armazenar estas informações, foi criada uma classe chamada “Perfil.java”, que contém as propriedades (dimensões). Após valorar cada propriedade, a estrutura é armazenada em um arquivo.

“Pedido.java” é a classe instanciada quando o comprador quer fazer um pedido de proposta. Nesta classe é possível informar o emissor do pedido, o identificador único do pedido (posteriormente será o identificador único da negociação), a matriz que diz ao vendedor qual é o produto a ser negociado e o t_{max} (tempo limite), além das dimensões que o comprador gostaria de negociar. Após preenchido, este objeto irá para o vendedor como conteúdo de uma mensagem “cfp” (Pedido de proposta).

Outra importante classe é a “Proposta.java”. Com ela, ambos, compradores e vendedores, expressam uma oferta para o produto em negociação. Após instanciar a classe, os agentes informam o número da negociação, o emissor da proposta, o t_{max} e a matriz de valor para cada dimensão em negociação.

A classe “TáticaTempo.java” é utilizada para determinar valores para as dimensões, usando como base na família de táticas em relação ao tempo (polinomial).

Duas funções, entre tantas, merecem destaque pela importância no protocolo CAMILO. A primeira, chamada de Função Depreciadora, está presente somente na classe “AgenteVendedor.java”. Já a segunda, chamada Função de Acréscimo, está presente na classe “AgenteVendedor.java” e “AgenteCompradorNP.java”.

5.5.2.1 A Função de Depreciação

Disponibilizada somente para os agentes vendedores que utilizam o protocolo CAMILO, a função é responsável por manter uma depreciação da utilidade do vendedor, evitando, assim, que o vendedor se prejudique fixando um valor que não corresponde às expectativas dos compradores.

A função é requisitada quando acontece uma estabilização por parte do vendedor, ou seja, quando o comprador faz uma oferta, considerada a melhor oferta, o vendedor considera que chegou a um preço de mercado e, portanto, não é necessário ceder muito. Com a utilização da função, o vendedor evita ser explorado pelos compradores, pois não cede muito, e ao mesmo tempo não pára de ceder, evitando perda de venda por pequenas

faixas de valores.

Para calcular os valores das dimensões da proposta depreciada, a função recebe como parâmetro a melhor proposta que será depreciada. Após o parâmetro ser recebido, é calculado, nos casos das dimensões preço e prazo de entrega, o valor mínimo (Equação 5.8) e nos caso da dimensão forma de pagamento, o valor máximo (Equação 5.9). O valor máximo aceitável, nos casos das dimensões preço e prazo de entrega, e o valor mínimo aceitável, no caso da dimensão forma de pagamento, é o valor da dimensão extraída da melhor proposta passada como parâmetro.

$$\text{Valor Mínimo} = \text{VlrMinAc} + ((\text{VlrDimMP} - \text{VlrMinAc}) * \text{PorcProx}) \quad (5.8)$$

$$\text{Valor Máximo} = \text{VlrMaxAc} - ((\text{VlrMaxAc} - \text{VlrDimMP}) * \text{PorcProx}) \quad (5.9)$$

sendo que: VlrMinAc = Valor mínimo aceitável (inicialmente configurado), VlrMaxAc = Valor máximo aceitável (inicialmente configurado), VlrDimMP = Valor da dimensão na melhor proposta recebida como parâmetro e PorcProx = porcentagem de proximidade da melhor proposta oferecida, ou seja, quanto maior for o valor da PorcProx , mais próximo o valor mínimo ou valor máximo ficará da melhor proposta e, portanto, menor será a depreciação.

Após a definição dos valores máximos e mínimos, a função instancia a classe de táticas e chama outra função que irá calcular o valor da dimensão. A função da tática é a mesma utilizada para gerar as propostas, mas a função de depreciação passa parâmetros diferentes, como os valores máximos e mínimos e o tempo inicial. O tempo inicial é o momento de entrada da melhor proposta. Com isso, o resultado é uma proposta pouco depreciada em função do tempo.

5.5.2.2 A Função de Acréscimo

Disponibilizada somente para os agentes compradores que utilizam o protocolo CAMILO, a função é responsável por montar uma nova proposta, melhor em utilidade para o comprador. Com isso, o comprador consegue explorar melhor a faixa de valores aceitáveis

do vendedor.

A função é requisitada quando o comprador já tem uma melhor proposta e precisa melhorá-la para enviá-la a outro vendedor.

Para calcular os valores das dimensões da melhor proposta melhorada, a função recebe como parâmetro a melhor proposta que será acrescida. Após receber o parâmetro, calcula o valor para todas as dimensões presentes na negociação.

$$\text{Valor da dimensão} = \text{VlrDimMP} - (\text{VlrDimMP} * (\text{ImpRelDim} * \text{MelhorPorcento})) \quad (5.10)$$

$$\text{Valor da dimensão} = \text{VlrDimMP} + (\text{VlrDimMP} * (\text{ImpRelDim} * \text{MelhorPorcento})) \quad (5.11)$$

sendo que: VlrDimMP = Valor da dimensão na melhor proposta recebida como parâmetro, ImpRelDim = Importância relativa da dimensão (inicialmente configurado) e MelhorPorcento = A porcentagem desejável de melhora na utilidade, lembrando que a porcentagem MelhorPorcento não é direta, ou seja, depende da importância (ImpRelDim) dada à dimensão inicialmente.

A Equação 5.10 é utilizada para fazer o cálculo de todas as dimensões que precisam ser reduzidas, ou seja, dimensões preço e prazo de entrega, que quanto menores melhor a utilidade do comprador. Já a Equação 5.11 é utilizada para fazer o cálculo de dimensões que precisam ser acrescidas, por exemplo a dimensão forma de pagamento, que quanto maior melhor a utilidade do comprador. Após a definição dos valores de todas as dimensões, a proposta é enviada para o vendedor.

5.6 Simulações e Resultados

Para que fosse possível avaliar o desempenho do protocolo CAMILO, foi implementado um protótipo de um sistema de negociação automatizado para comércio eletrônico, chamado SMANACE. Com esse protótipo foram executadas diversas experiências que possibilitaram, após análise progressiva dos resultados, melhorias no protocolo CAMILO e um comparativo deste protocolo e o referenciado por Farantin (1998).

Todas as configurações de negociação, como táticas de negociação e parâmetros das dimensões de negociação (pesos, gama de valores aceitáveis e função de pontuação), são privadas. Portanto, os agentes começam a negociar sem saber se é possível obter um acordo.

Durante os testes realizados, tentou-se criar ambientes favoráveis à obtenção de acordos, ou seja, os agentes foram configurados para que o produto requisitado fosse o mesmo oferecido e o intervalo de valores tivesse intersecção. As negociações executadas decorrem sobre três dimensões: preço, prazo de entrega e forma de pagamento (em dias). Com três dimensões é possível demonstrar a negociação multidimensional com n dimensões, uma vez que o aumento de dimensão, na estrutura utilizada, não altera os resultados.

Todos os testes realizados tiveram, como parâmetros de entrada, variáveis que alimentavam os agentes com informações sobre o perfil do usuário que o mesmo representava. As variáveis utilizadas foram: Nome do agente, Valor Mínimo, Valor Máximo, Importância Relacional (ver Seção 5.2) e constante K_j^i (ver Seção 5.4.1).

5.6.1 Comparando os Protocolos em Ambiente sem Concorrência

Foram feitas duas comparações para analisar o quanto o protocolo CAMILO favorece o comprador e o vendedor, se comparado ao protocolo referenciado por Farantin (1998).

Para a primeira comparação dois cenários foram construídos. No primeiro, o comprador e os dois vendedores executavam o protocolo referenciado por Farantin (1998), e no segundo, o comprador executava o protocolo CAMILO e os vendedores o protocolo referenciado por Farantin (1998). Comparando os resultados, é possível identificar qual é o protocolo mais favorável para o comprador.

Para a segunda comparação, outros dois cenários foram construídos. No primeiro, os dois compradores e o vendedor executavam o protocolo referenciado por Farantin (1998), e no segundo, o vendedor executava o protocolo CAMILO e os dois compradores o protocolo referenciado por Farantin (1998). Comparando os resultados, é possível identificar qual é o protocolo mais favorável para o vendedor.

Ambos os cenários montados terão como resultados valores retirados dos respectivos gráficos (gerados a partir do arquivo montado pelo simulador) ou da interface de execução do simulador (exemplificado nos Anexos A, B, C e D). Valores de utilidade e tempo, de cada cenário, poderão ser observados pelo gráfico, já os valores finais das dimensões (como

preço, prazo de entrega e forma de pagamento) poderão ser observados pela interface de execução do simulador, como mostrado no Anexo C.

Para facilitar a leitura dos cenários, foi adotada a seguinte nomenclatura para os agentes: <nComparação> <nCenário> <intenção> <n>, onde <nComparação> é o número da comparação, <nCenário> é o número do cenário, <intenção> representa a intenção de compra (c - comprador) ou venda (v - vendedor) e <n> é o número de ordem do comprador ou vendedor. Por exemplo, o agente 11c2 é o segundo comprador do cenário 1 da comparação 1.

5.6.1.1 Primeira Comparação

Primeiro Cenário: No primeiro cenário, um comprador utilizando o protocolo referenciado por Farantin (1998) e valor de $\beta=1$, negocia com dois vendedores que também utilizam o protocolo referenciado por Farantin (1998) e valor de $\beta=1$. O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante. Portanto, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 12: Configuração do Agente Comprador 11c1.

Agente: 11c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 13: Configuração do Agente Vendedor 11v1.

Agente: 11v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 14: Configuração do Agente Vendedor 11v2.

Agente: 11v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

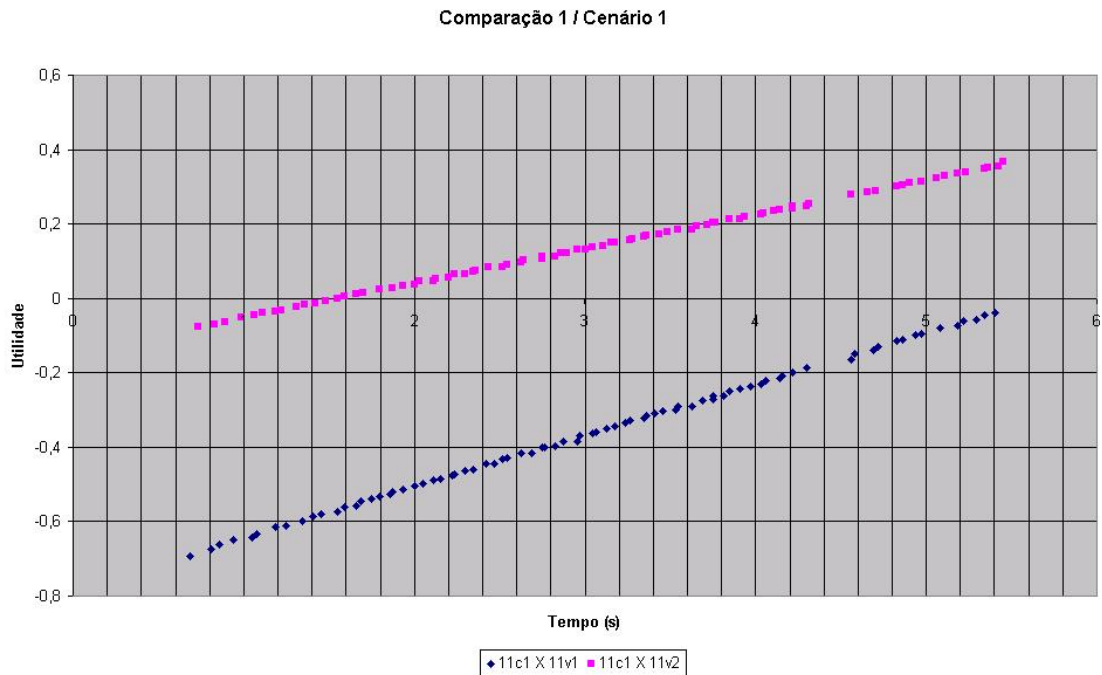


Figura 5.9: Primeiro cenário da primeira comparação.

Resultados: Após a execução (Figura 5.9) o comprador 11c1 irá comprar o produto do vendedor 11v2 a um preço de 105,35, pagará em 40,98 dias e receberá o produto em 13,04 dias. A utilidade final do comprador é de 0,3662 e o tempo gasto foi de 5,45 segundos.

Segundo Cenário: No segundo cenário, um comprador utilizando o protocolo CAMILO e valor de $\beta=1$, negocia com dois vendedores que utilizam o protocolo referenciado por Farantin (1998) e valor de $\beta=1$. O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 15: Configuração do Agente Comprador 12c1.

Agente: 12c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 16: Configuração do Agente Vendedor 12v1

Agente: 12v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 17: Configuração do Agente Vendedor 12v2

Agente: 12v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

Resultados: Após a execução (Figura 5.10) o comprador 12c1 irá comprar o produto do vendedor 12v2 a um preço de 90,4, pagará em 59,5 dias e receberá o produto em 5,35 dias. A utilidade final do comprador é de 0,7674 e o tempo gasto foi de 10 segundos.

Análise dos Resultados (Cenário 1 X Cenário 2): Analisando os dois cenários, é possível identificar o benefício de usar o protocolo CAMILO. O comprador no cenário 1 utilizou o protocolo referenciado por Farantin (1998) e obteve uma utilidade de 0,3662, enquanto no cenário 2 o comprador, utilizando o protocolo CAMILO, negociou com os mesmos vendedores e obteve uma utilidade de 0,7674. Sendo assim, o comprador teve um grande ganho de 0,4012 na utilidade.

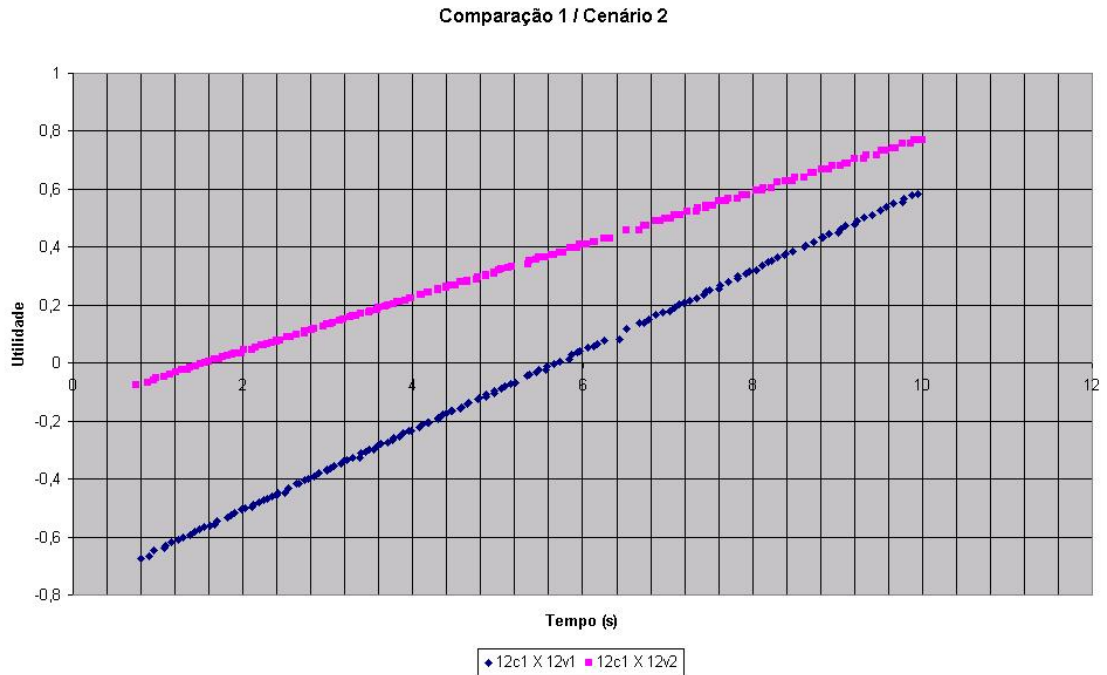


Figura 5.10: Segundo cenário da primeira comparação.

5.6.1.2 Segunda Comparação

Primeiro Cenário: No primeiro cenário, um vendedor utilizando o protocolo referenciado por Farantin (1998) e valor de $\beta=1$, negocia com dois compradores que também utilizam o protocolo referenciado por Farantin (1998) e valor de $\beta=1$. O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 18: Configuração do Agente Comprador 21c1

Agente: 21c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 19: Configuração do Agente Comprador 21c2

Agente: 21c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 20: Configuração do Agente Vendedor 21v1

Agente: 21v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

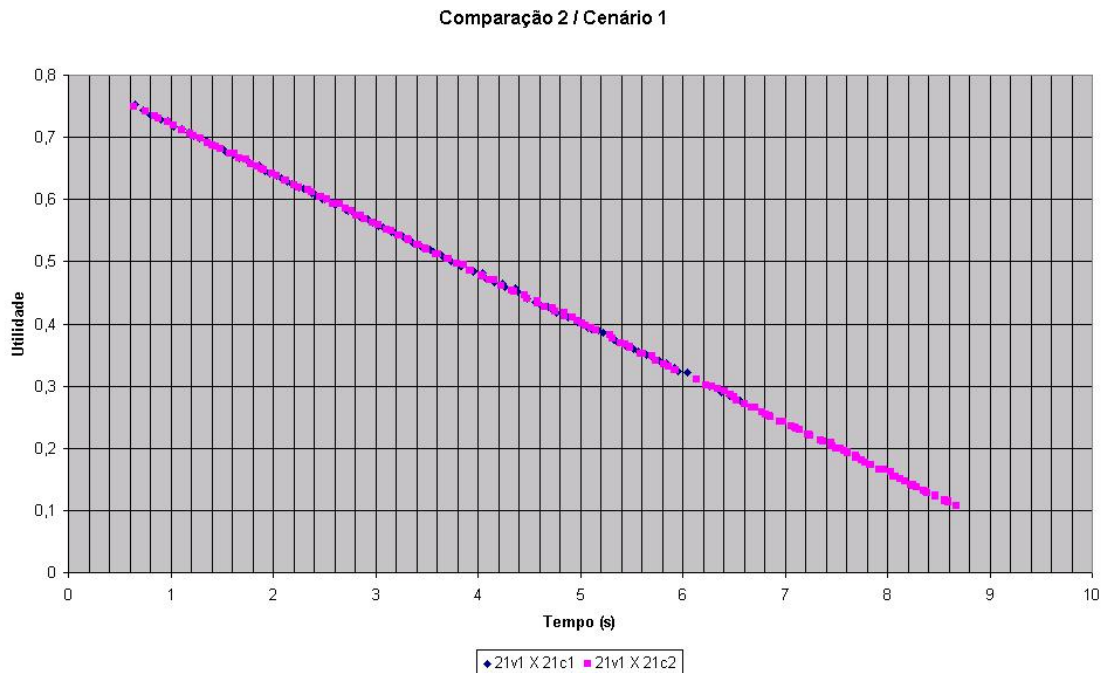


Figura 5.11: Primeiro cenário da segunda comparação.

Resultados: Após a execução (Figura 5.11) o vendedor 21v1 irá vender o produto para o comprador 21c1 a um preço 109,00, que pagará em 38,24 dias e receberá o produto em 14,77 dias. A utilidade final do vendedor na negociação com o 21c1 é de 0,2748 e o tempo gasto foi de 6,57 segundos.

Na negociação com o 21c2 o vendedor 21v1 irá vender o produto a um preço de 96,81, receberá o pagamento em 33,18 dias e entregará o produto em 13,61 dias. A utilidade do vendedor nesta negociação foi de 0,1077 e o tempo gasto foi de 8,67 segundos.

Segundo Cenário: No segundo cenário, um vendedor utilizando o protocolo CAMILO e valor de $\beta=1$, negocia com dois compradores que utilizam o protocolo referenciado por Farantin (1998) e valor de $\beta=1$. O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 21: Configuração do Agente Comprador 22c1.

Agente: 22c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 22: Configuração do Agente Comprador 22c2.

Agente: 22c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 23: Configuração do Agente Vendedor 22v1.

Agente: 22v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

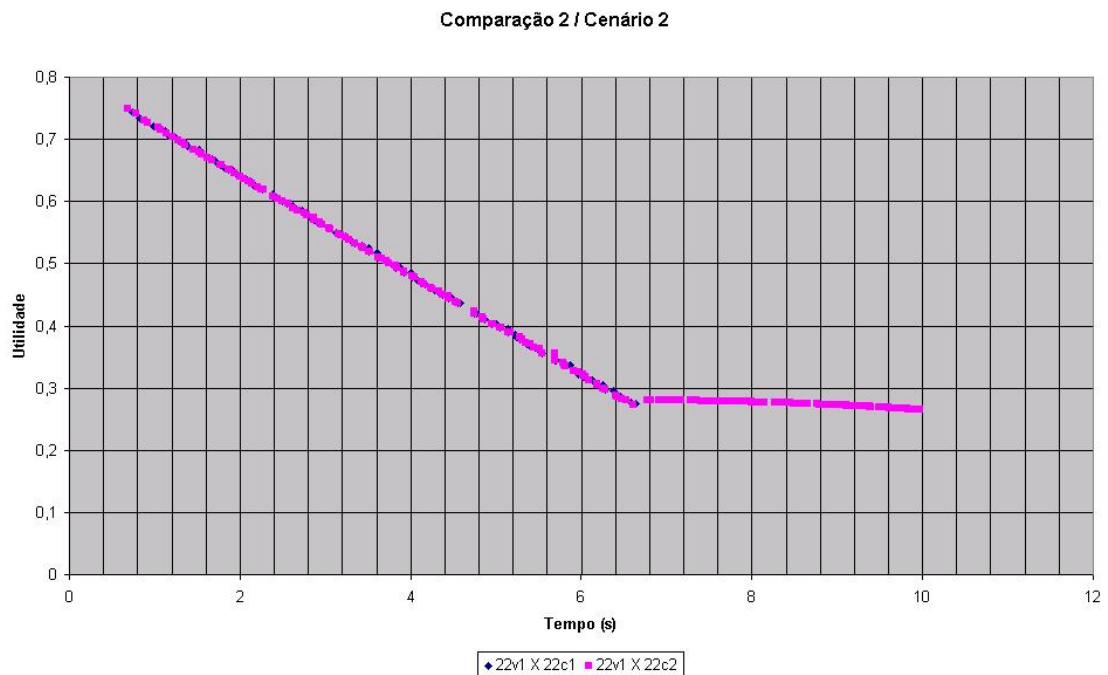


Figura 5.12: Segundo cenário da segunda comparação.

Resultados: Após a execução (Figura 5.12) o vendedor 22v1 irá vender o produto para o comprador 22c1 a um preço de 109,04, que pagará em 38,21 dias e receberá o produto em 14,79 dias. A utilidade final do vendedor na negociação com o 22c1 é de 0,2756 e o tempo gasto foi de 6,64 segundos.

Na negociação com o 22c2, o vendedor 22v1 não irá vender o produto, pois não chegaram em um acordo.

Análise dos Resultados (Cenário 1 X Cenário 2): Analisando os dois cenários, é possível identificar o benefício de usar o protocolo CAMILO. O vendedor no cenário 1 utilizou o protocolo referenciado por Farantin (1998) e obteve uma utilidade de 0,2748, enquanto no cenário 2 o vendedor, utilizando o protocolo CAMILO, negociou com os mesmos compradores e obteve uma utilidade de 0,2756. Apesar do ganho muito pequeno, de 0,0008 na utilidade, o que mais favorece a utilização do protocolo CAMILO é o fato de que o vendedor, no cenário 2, após fechar acordo com o 22c1, não cede muito a outros compradores, pois entende que chegou a um preço razoável de mercado. Com isso o vendedor não é muito explorado.

5.6.2 Comparando os Protocolos em Ambiente de Concorrência

Foram criados alguns cenários que possibilitaram a análise dos agentes, implementando os protocolos, quando colocados em ambientes de concorrência.

Para facilitar a leitura dos cenários, foi adotada a seguinte nomenclatura para os agentes: $\langle n \text{Cenário} \rangle \langle \text{intenção} \rangle \langle n \rangle$, onde $\langle n \text{Cenário} \rangle$ é o número do cenário, $\langle \text{intenção} \rangle$ representa a intenção de compra (c - comprador) ou venda (v - vendedor) e $\langle n \rangle$ é o número de ordem de comprador ou vendedor. Por exemplo, o agente 1c2 é o segundo comprador do cenário 1.

5.6.2.1 Primeiro Cenário

Para o cenário 1 foram executados 4 agentes, sendo 2 compradores (1c1,1c2), com protocolo referenciado por Farantin (1998), e 2 vendedores (1v1,1v2), também com o protocolo referenciado por Farantin (1998). O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário agentes não entram durante o processo de negociação.

Configurando os Agentes:

Tabela 24: Configuração do Agente Comprador 1c1.

Agente: 1c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 25: Configuração do Agente Comprador 1c2.

Agente: 1c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 26: Configuração do Agente Vendedor 1v1.

Agente: 1v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 27: Configuração do Agente Vendedor 1v2.

Agente: 1v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

Resultado: Após a execução (Figura 5.13) o vendedor 1v2 irá vender o produto para o comprador 1c1 a um preço de 105,25, que pagará em 41,06 dias e receberá o produto em 12,99 dias. A utilidade final do comprador na negociação é de 0,3687 e o tempo gasto foi de 5,51 segundos. A utilidade do vendedor 1v2 foi de 0,3657.

Na negociação com o 1c2 o vendedor 1v2 irá vender o produto a um preço de 96,89, receberá o pagamento em 51,37 dias e entregará o produto em 11,03 dias. A utilidade do vendedor 1v2 nesta negociação foi de 0,1724 e o tempo gasto foi de 7,8 segundos. A utilidade do comprador 1c2 foi de 0,1844.

O vendedor 1v1 não conseguiu fechar acordo com os compradores, pois os intervalos de valores, referentes às dimensões, estavam mais longe do requerido pelos compradores.

5.6.2.2 Segundo Cenário

Para o cenário 2 foram executados 4 agentes, sendo 1 comprador (2c1) com protocolo referenciado por Farantin (1998), 1 comprador (2c2) com o protocolo CAMILO e 2 vendedores (2v1,2v2) com o protocolo referenciado por Farantin (1998). O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo

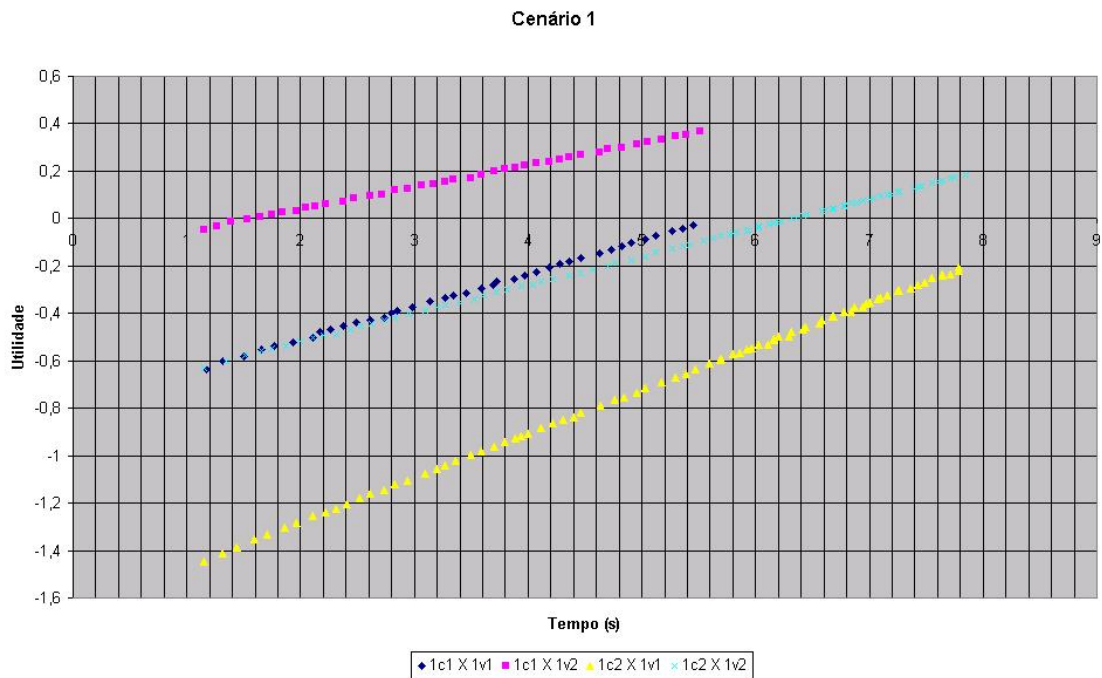


Figura 5.13: Gráfico referente ao primeiro cenário.

instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 28: Configuração do Agente Comprador 2c1.

Agente: 2c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 29: Configuração do Agente Comprador 2c2.

Agente: 2c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 30: Configuração do Agente Vendedor 2v1.

Agente: 2v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 31: Configuração do Agente Vendedor 2v2.

Agente: 2v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

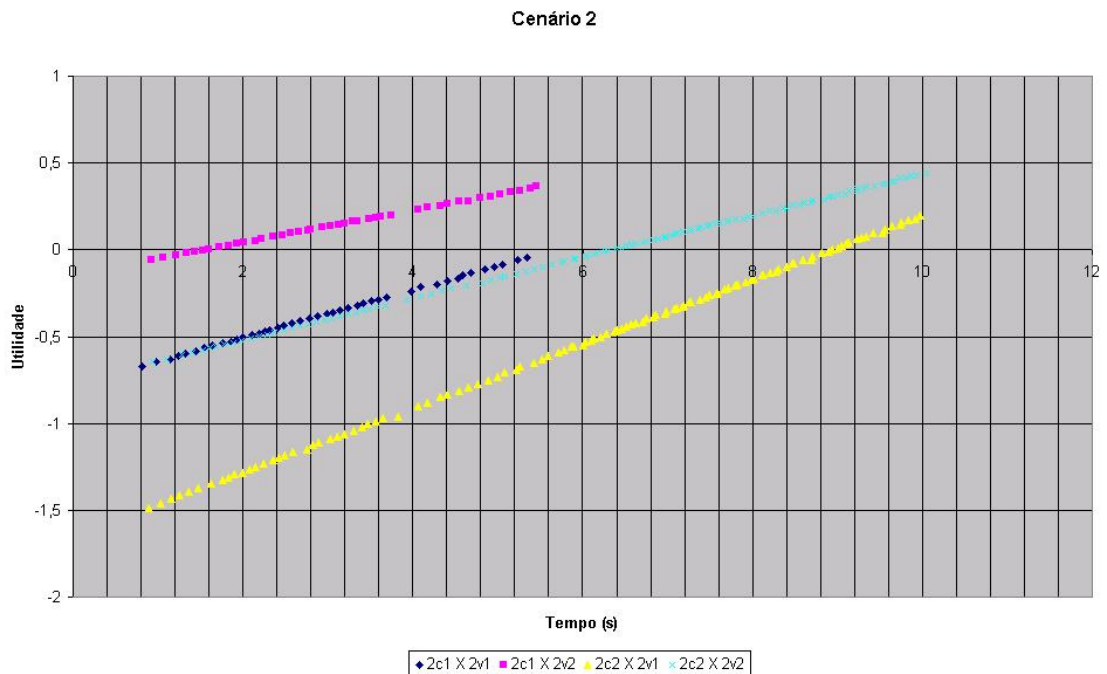


Figura 5.14: Gráfico referente ao segundo cenário.

Resultado: Após a execução (Figura 5.14) o vendedor 2v2 irá vender o produto para o comprador 2c1 a um preço de 105,24, que pagará em 41,06 dias e receberá o produto

em 12,99 dias. A utilidade final do comprador na negociação é de 0,3688 e o tempo gasto foi de 5,45 segundos. A utilidade do vendedor 2v2 foi de 0,3656.

Na negociação com o 2c2 o vendedor 2v2 irá vender o produto a um preço de 90,10, receberá o pagamento em 59,87 dias e entregará o produto em 5,08 dias. A utilidade do vendedor 2v2 nesta negociação foi de 0,0025 e o tempo gasto foi de 10 segundos. A utilidade do comprador 2c2 foi de 0,4397.

O vendedor 2v1 não conseguiu fechar acordo com os compradores, pois os intervalos de valores, referentes às dimensões, estavam mais longe do requerido pelos compradores.

5.6.2.3 Terceiro Cenário

Para o cenário 3 foram executados 4 agentes, sendo 2 compradores (3c1,3c2) com o protocolo CAMILO e 2 vendedores (3v1,3v2) com o protocolo referenciado por Farantin (1998). O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não entram agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 32: Configuração do Agente Comprador 3c1.

Agente: 3c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 33: Configuração do Agente Comprador 3c2.

Agente: 3c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 34: Configuração do Agente Vendedor 3v1.

Agente: 3v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 35: Configuração do Agente Vendedor 3v2.

Agente: 3v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

Resultado: Após a execução (Figura 5.15) o vendedor 3v2 irá vender o produto para o comprador 3c1 a um preço de 90,19, que pagará em 59,75 dias e receberá o produto em 5,17 dias. A utilidade final do comprador na negociação é de 0,7732 e o tempo gasto foi de 10 segundos. A utilidade do vendedor 3v2 foi de 0,0049.

Na negociação com o 3c2 o vendedor 3v2 irá vender o produto a um preço de 90,49, receberá o pagamento em 59,37 dias e entregará o produto em 5,43 dias. A utilidade do vendedor 3v2 nesta negociação foi de 0,0124 e o tempo gasto foi de 10 segundos. A utilidade do comprador 3c2 foi de 0,4248.

O vendedor 3v1 não conseguiu fechar acordo com os compradores, pois os intervalos de valores, referentes às dimensões, estavam mais longe do requerido pelos compradores.

5.6.2.4 Quarto Cenário

Para o cenário 4 foram executados 4 agentes, sendo 2 compradores (4c1,4c2) com protocolo referenciado por Farantin (1998) e 2 vendedores (4v1,4v2) com o protocolo CAMILO. O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo

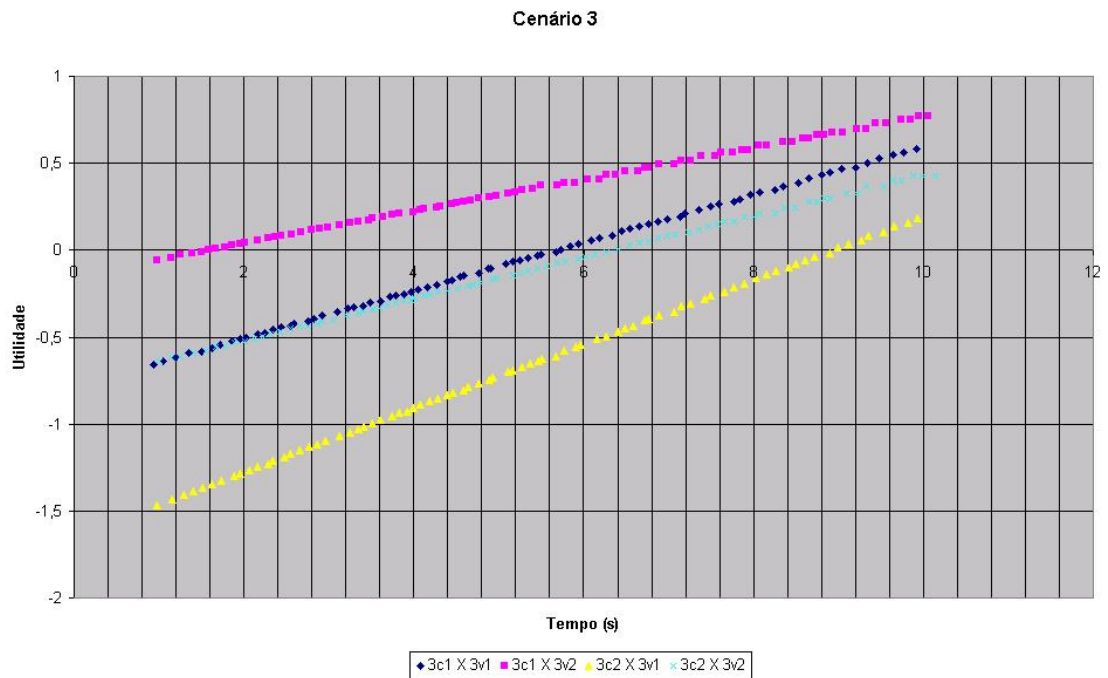


Figura 5.15: Gráfico referente ao terceiro cenário.

instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 36: Configuração do Agente Comprador 4c1.

Agente: 4c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 37: Configuração do Agente Comprador 4c2.

Agente: 4c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 38: Configuração do Agente Vendedor 4v1.

Agente: 4v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 39: Configuração do Agente Vendedor 4v2.

Agente: 4v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

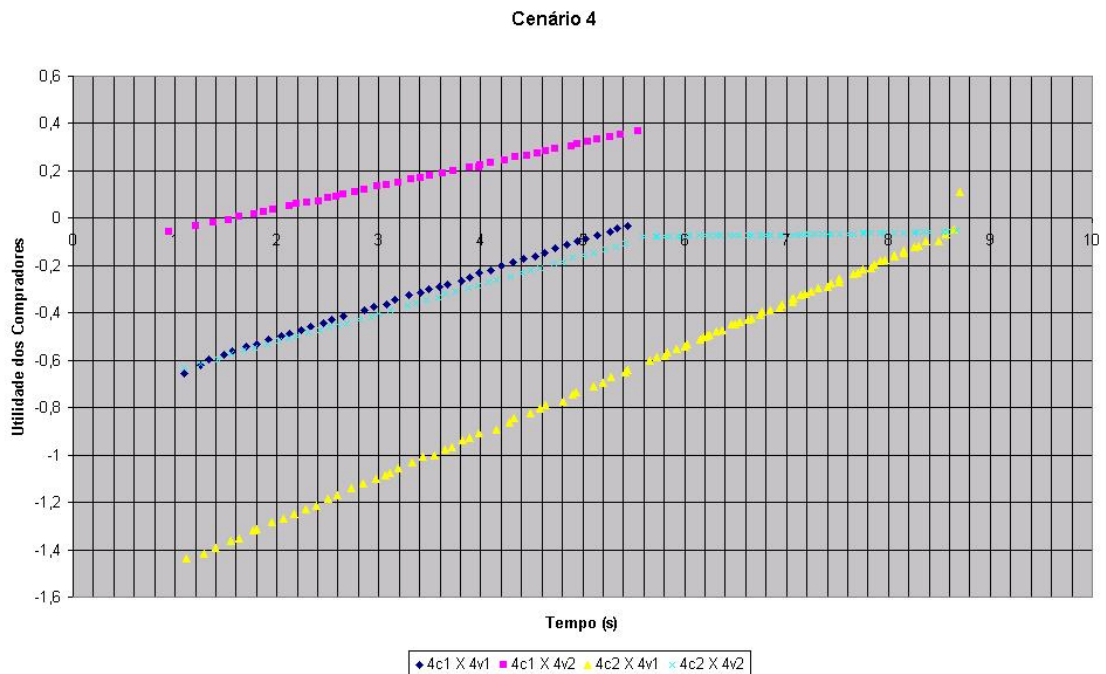


Figura 5.16: Gráfico referente ao quarto cenário.

Resultado: Após a execução (Figura 5.16) o vendedor 4v2 irá vender o produto para o comprador 4c1 a um preço de 105,2, que pagará em 41,1 dias e receberá o produto em

12,97 dias. A utilidade final do comprador na negociação é de 0,3699 e o tempo gasto foi de 5,54 segundos. A utilidade do vendedor 4v2 foi de 0,3645.

Na negociação com o 4c2 o vendedor 4v1 irá vender o produto a um preço de 96,73, receberá o pagamento em 33,26 dias e entregará o produto em 13,58 dias. A utilidade do vendedor 4v1 nesta negociação foi de 0,1064 e o tempo gasto foi de 8,70 segundos. A utilidade do comprador 4c2 foi de 0,1087.

O vendedor 4v2 quando vende seu produto, para o comprador 4c1, desacelera o processo de ceder para o comprador 4c2, pois entende que chegou em um preço de mercado. Com isso, o vendedor 4v1 que até então não conseguiu vender é obrigado a ceder e vende o produto para o 4c2.

5.6.2.5 Quinto Cenário

Para o cenário 5 foram executados 4 agentes, sendo 1 comprador (5c1) com protocolo referenciado por Farantin (1998), 1 comprador (5c2) com o protocolo CAMILO e 2 vendedores (5v1,5v2) com o protocolo CAMILO. O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 40: Configuração do Agente Comprador 5c1.

Agente: 5c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 41: Configuração do Agente Comprador 5c2.

Agente: 5c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 42: Configuração do Agente Vendedor 5v1.

Agente: 5v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 43: Configuração do Agente Vendedor 5v2.

Agente: 5v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

Resultado: Após a execução (Figura 5.17) o vendedor 5v2 irá vender o produto para o comprador 5c1 a um preço de 105,40, que pagará em 40,94 dias e receberá o produto em 13,06 dias. A utilidade final do comprador na negociação é de 0,3649 e o tempo gasto foi de 5,61 segundos. A utilidade do vendedor 5v2 foi de 0,3691.

Na negociação com o 5c2 o vendedor 5v1 irá vender o produto a um preço de 95,53, receberá o pagamento em 35,31 dias e entregará o produto em 13,55 dias. A utilidade do vendedor 5v1 nesta negociação foi de 0,0842 e o tempo gasto foi de 10 segundos. A utilidade do comprador 5c2 foi de 0,1478.

O vendedor 5v2 quando vende seu produto, para o comprador 5c1, desacelera o processo de ceder para o comprador 5c2, pois entende que chegou em um preço de mercado. Com isso, o vendedor 5v1 que até então não conseguiu vender para ninguém é obrigado a ceder e vende o produto para o 5c2.

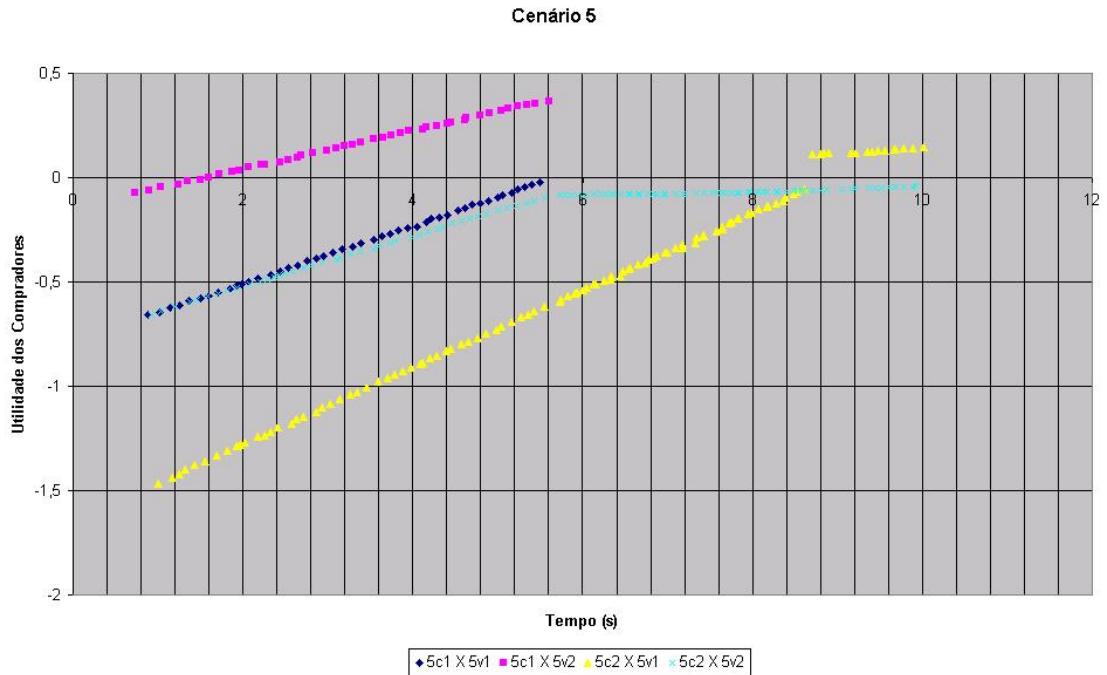


Figura 5.17: Gráfico referente ao quinto cenário.

5.6.2.6 Sexto Cenário

Para o cenário 6 foram executados 4 agentes, sendo 2 compradores (6c1,6c2) com protocolo CAMILO e 2 vendedores (6v1,6v2) com o protocolo CAMILO. O valor de β é igual a 1 para todos.

O tempo de negociação máximo é de 10 segundos e a tática usada, por todos os agentes, é em função do tempo e polinomial. Os agentes são inicializados no mesmo instante, ou seja, neste cenário não existe a entrada de agentes durante o processo de negociação.

Configurando os Agentes:

Tabela 44: Configuração do Agente Comprador 6c1.

Agente: 6c1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	80	120	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	1	20	0.1	0.2

Tabela 45: Configuração do Agente Comprador 6c2.

Agente: 6c2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	70	100	0.8	0.2
Forma de Pagamento	30	60	0.1	0.2
Prazo de Entrega	2	15	0.1	0.2

Tabela 46: Configuração do Agente Vendedor 6v1.

Agente: 6v1	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	145	0.8	0.2
Forma de Pagamento	1	45	0.1	0.2
Prazo de Entrega	30	120	0.1	0.2

Tabela 47: Configuração do Agente Vendedor 6v2.

Agente: 6v2	Valor Mínimo	Valor Máximo	Importância Relacional	Constante K_j^i
Preço	90	130	0.8	0.2
Forma de Pagamento	10	60	0.1	0.2
Prazo de Entrega	5	40	0.1	0.2

Resultado: Após a execução (Figura 5.18) o vendedor 6v1 irá vender o produto para o comprador 6c1 a um preço de 95,91, que pagará em 34,69 dias e receberá o produto em 13,58 dias. A utilidade final do comprador na negociação é de 0,5310 e o tempo gasto foi de 10 segundos. A utilidade do vendedor 6v1 foi de 0,0912.

Na negociação com o 6c2 o vendedor 6v1 irá vender o produto a um preço de 95,32, receberá o pagamento em 35,72 dias e entregará o produto em 13,58 dias. A utilidade do vendedor 6v1 nesta negociação foi de 0,0803 e o tempo gasto foi de 10 segundos. A utilidade do comprador 6c2 foi de 0,1545.

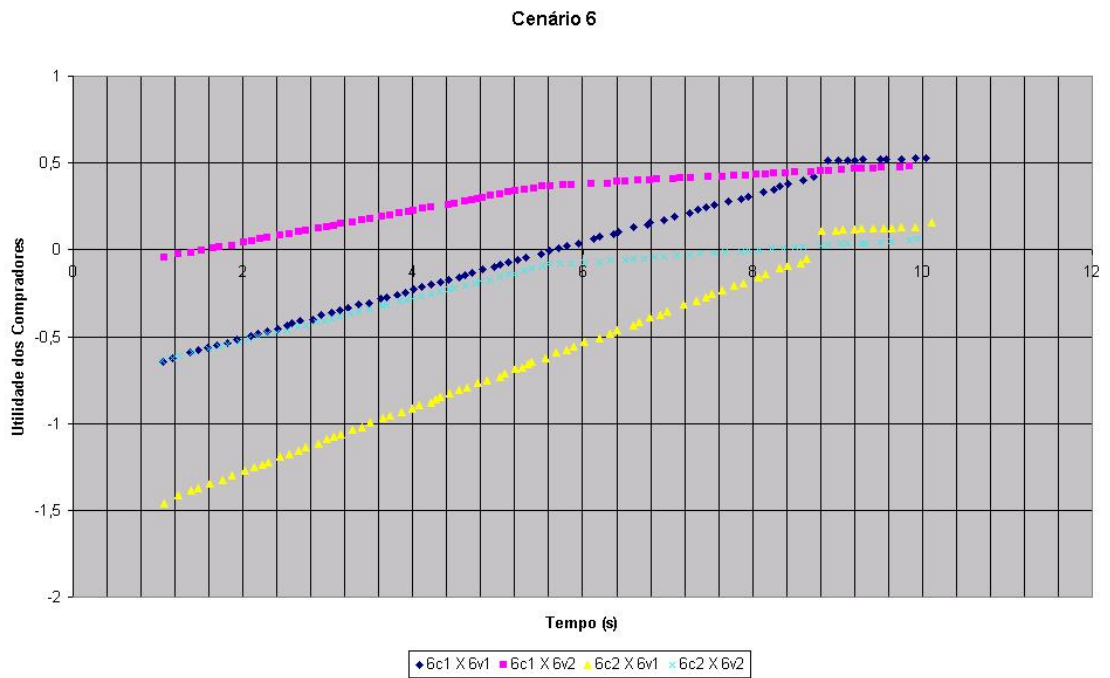


Figura 5.18: Gráfico referente ao sexto cenário.

6 *Considerações Finais*

6.1 Conclusões

O campo de negociação automatizada para agentes de *softwares* é, sem dúvida, um campo muito importante que fomenta muitas aplicações, como por exemplo, o Comércio Eletrônico, o qual cresceu em progressão geométrica nos últimos anos e tem perspectivas de mais crescimento para os anos seguintes. Isso porque alia a necessidade de compra/venda (transação comercial) com a facilidade e comodidade da *internet*.

Inspirado pelo ambiente fértil, formado pelas áreas de Comércio Eletrônico e negociação automatizada para agentes de *softwares*, este trabalho empenhou-se em abastecer o mercado e incrementar os estudos científicos de uma novidade que pudesse trazer benefícios. Com este propósito, foi desenvolvido o SMANACE (Sistema Multiagente de Negociação Automatizada para Comércio Eletrônico), que teve como principais objetivos o desenvolvimento de um protocolo, denominado CAMILO, para negociação automatizada e a comparação entre o protocolo proposto e o protocolo referenciado por Farantin (1998).

O processo de desenvolvimento do protocolo CAMILO foi constituído por uma fase inicial de análise das possíveis negociações, entre uma entidade compradora e outra vendedora, uma segunda fase de definição e modelagem das regras e por último a implementação. Após a primeira versão do protocolo, foram feitos alguns testes e constatou-se a necessidade de melhorias que acarretou uma nova análise, modelagem e implementação. Alterações importantes na proposta de retorno, criação de uma taxa de depreciação e outras decorrentes destas foram implementadas, a partir de necessidades detectadas no decorrer dos testes realizados. Deste processo de desenvolvimento, descrito com maiores detalhes na seção 5.3.3, surgiu a versão final usada nas simulações e comparações.

Deste processo surgiu a versão final usada nas simulações e comparações.

Com o intuito de analisar a utilidade e comparar o desempenho do protocolo CAMILO em relação ao referenciado por Farantin (1998), foram criados dez cenários. Em quatro

destes o agente analisado não tinha no mercado nenhum concorrente, enquanto nos outros seis inseriu-se a concorrência. Após a execução dos cenários, detalhados na seção 5.6.1, foi possível constatar que o protocolo CAMILO tem vantagens, nos cenários criados, em relação ao referenciado por Farantin (1998).

Os bons resultados obtidos contribuem para que o campo de pesquisa se torne ainda mais atrativo.

6.2 Sugestões para Trabalhos Futuros

Durante o desenvolvimento deste trabalho, deparou-se com algumas possibilidades que viabilizariam a continuidade de estudos nesta área, os quais poderiam ter como base esta dissertação.

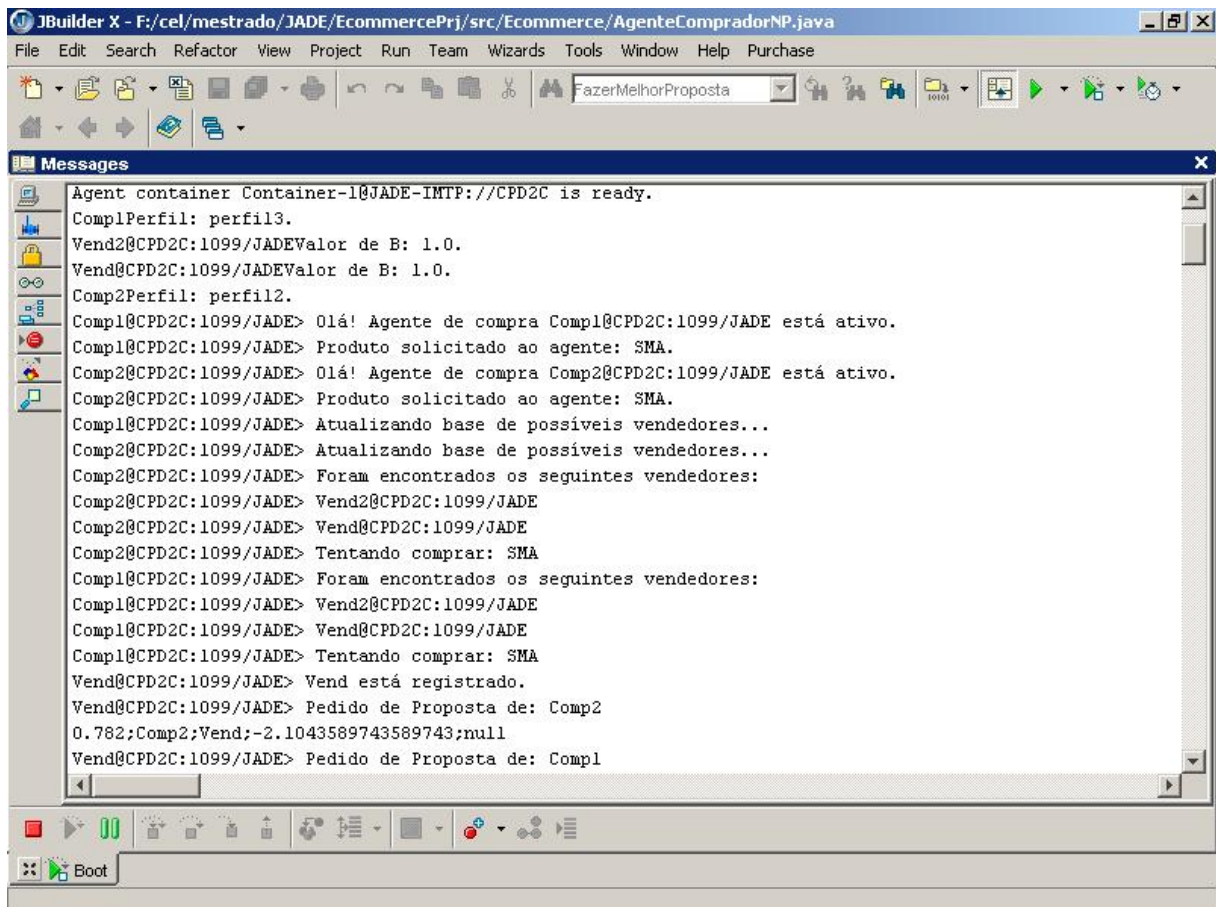
Como sugestão, verificou-se a possibilidade de exploração da negociação multidimensional com o diferencial de utilizar dimensões qualitativas ou dependentes entre si.

Outro ponto possível de investigação seria a utilização do protocolo CAMILO em diferentes cenários ou diferentes aplicações.

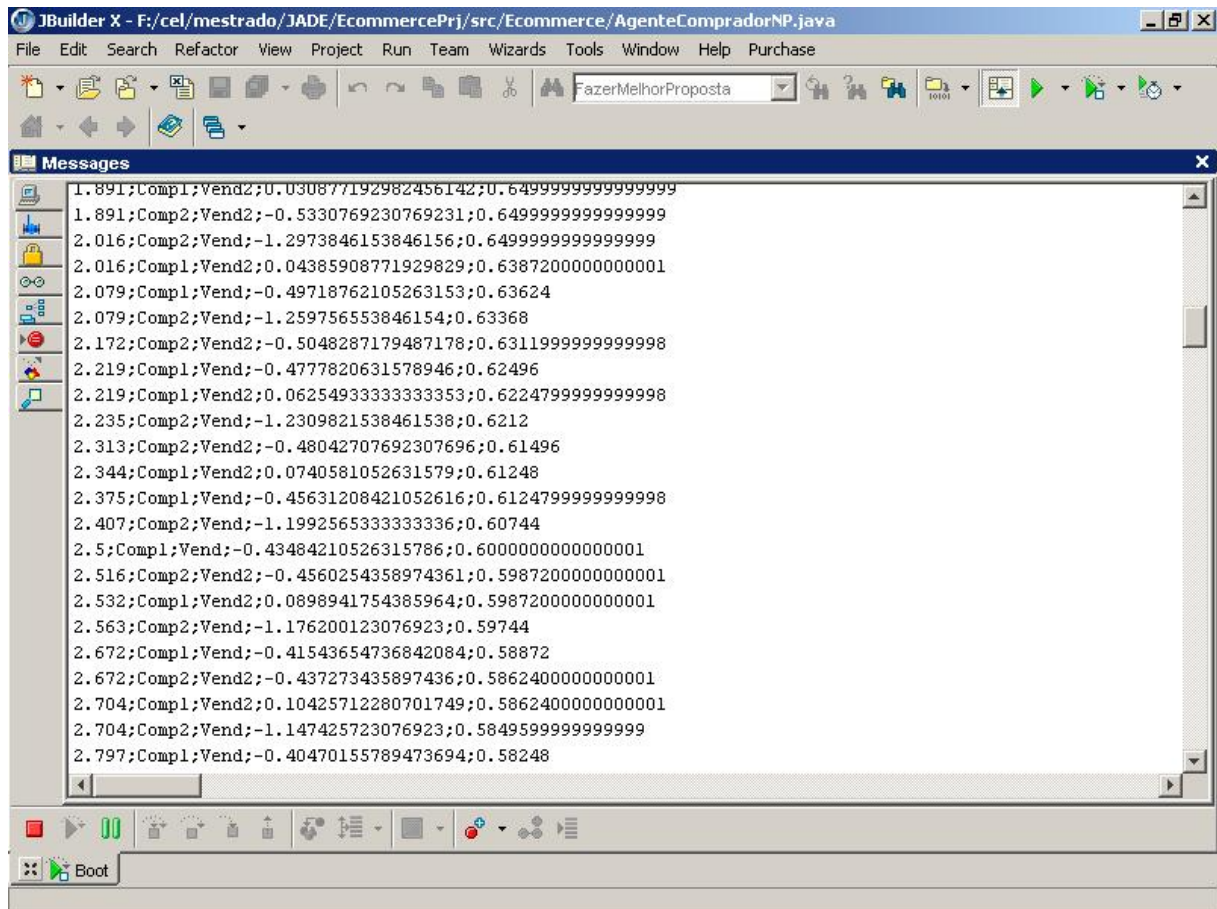
Por fim, a utilização de algoritmos de aprendizagem para montar estratégias, principalmente aprendizagem por reforço, também poderiam trazer melhorias no desempenho dos agentes, fato este que justificaria uma investigação mais profunda.

Anexos

Anexo A - Início de uma Simulação



Anexo B - A Negociação



Anexo C - Resultado de uma Simulação

```

JBuilder X - F:/cel/mestrado/JADE/EcommercePrj/src/Ecommerce/AgenteCompradorNP.java
File Edit Search Refactor View Project Run Team Wizards Tools Window Help Purchase
FazerMelhorProposta
Messages
9.407;Comp1;Vend;0.5162845851313143;0.10192596891059413
9.485;Comp2;Vend;0.12119586522147094;0.09947013252083911
9.485;Comp2;Vend2;0.030673293896040864;0.2842049370785121
9.5;Comp1;Vend;0.5202791230801795;0.09906185550209079
9.907;Comp1;Vend2;0.4690207932524173;0.276252527542581
9907@Vend>Emissor:( agent-identifier :name Comp2@CPD2C:1099/JADE )
9.907;Comp2;Vend;0.12119586522147094;0.09947013252083911
9969@Vend>Emissor:( agent-identifier :name Comp1@CPD2C:1099/JADE )
10.032;Comp2;Vend;0.12119586522147094;0.09947013252083911
10.063;Comp1;Vend;0.5202791230801795;0.09906185550209079
Comp2@CPD2C:1099/JADE>Prop Vencedora: 96.35644438329899/33.87431463283323/13.554400000000001.De:( agent-i
Comp1@CPD2C:1099/JADE>Prop Vencedora: 96.33440324547544/33.91289323901177/13.554400000000001.De:( agent-i
Comp2@CPD2C:1099/JADE>Prop Vencedora Utilidade: 0.12119586522147094/Existe Tempo:false/Tempo Restante:-63
Comp1@CPD2C:1099/JADE>Prop Vencedora Utilidade: 0.5202791230801795/Existe Tempo:false/Tempo Restante:-63
Vend@CPD2C:1099/JADE> Ordem de Compra: Comp2
Vend@CPD2C:1099/JADE> PED1112963000341Comp2 vendido para o agente Comp2@CPD2C:1099/JADE
Comp2@CPD2C:1099/JADE> Compra de SMA/ efetuada com sucesso. Vendedor Vend@CPD2C:1099/JADE
Comp2@CPD2C:1099/JADE> Agente de Compra Comp2@CPD2C:1099/JADE finalizado.
Vend@CPD2C:1099/JADE> Ordem de Compra: Comp1
Vend@CPD2C:1099/JADE> PED1112963000341Comp1 vendido para o agente Comp1@CPD2C:1099/JADE
Comp1@CPD2C:1099/JADE> Compra de SMA/ efetuada com sucesso. Vendedor Vend@CPD2C:1099/JADE
Comp1@CPD2C:1099/JADE> Agente de Compra Comp1@CPD2C:1099/JADE finalizado.
Boot

```

Anexo D - Exemplo das Trocas de Mensagens

The screenshot displays a network sniffer interface titled "sniffer3@CPD2C:1099/JADE - Sniffer Agent". The main window shows a sequence of 21 messages between four entities: Other, Vend, Vend2, Comp1, and Comp2. The messages are all of type "PROPOSE" and include details like "(ade 076 mp2)", "(ade 310 mp1)", etc. The ACL Message detail panel on the right shows the following information:

- Sender:** Vend2@CPD2C:1099/JADE
- Receivers:** Comp2@CPD2C:1099/JADE
- Reply-to:** (empty)
- Communicative act:** propose
- Content:** A large block of escaped characters, including "xsrjade.util.leap.Properties", "thresholdxp?", and "6t34.688t22.71840000000003t00.50624ppppp".
- Language:** (empty)
- Encoding:** (empty)
- Ontology:** (empty)
- Protocol:** Null
- Conversation-id:** book-trade
- In-reply-to:** PED1112964636951Comp2
- Reply-with:** @CPD2C:1099/JADE1112964640623
- Reply-by:** View
- User Properties:** (empty)

At the bottom of the sniffer window, a status bar reads: "Message:6 PROPOSE (cid=trade rw=40623 irt=Comp2 proto=null onto=nul". An "OK" button is visible at the bottom right of the ACL Message panel.

Referências Bibliográficas

AGLETS. *The Aglets Portal*. 2004. Disponível em: <<http://aglets.sourceforge.net/>>. Acesso em: 14 fev. 2004.

ALBERTIN, A. L. 6a. pesquisa fgv-eaesp de comércio eletrônico no mercado brasileiro. Março 2004. Resumo. Disponível em: <http://www.fgvsp.br/academico/estudos/cia/Pesquisa_5CE.PDF>. Acesso em: 22 jan. 2005.

AURÉLIO, B. d. H. F. *Minidicionário da Língua Portuguesa*. Primeira edição. [S.l.]: Editora Nova Fronteira, 1977.

DEMAZEAU; MÜLLER. Decentralized artificial intelligence. In: *European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge, England*. [S.l.]: Elsevier Science Publishers, 1990. p. 3–13.

DEMAZEAU, Y. From interactions to collective behaviour agent-based systems. In: *Proc. of the 1st. European Conference on Cognitive Science, St. Malo, France*. [S.l.: s.n.], 1995.

ECONOMIST, I. U.; IBM, I. f. B. V. The 2004 e-readiness rankings. 2004. Resumo. Disponível em: <http://graphics.eiu.com/files/ad_pdfs/ERR2004.pdf>. Acesso em: 21 jun. 2005.

FARANTIN, P. *Negotiation Among Groups of Autonomous Computational Agents*. Tese (Doutorado) — Department of Electronic Engineering-Queen Mary, University of London, Junho 1998.

FIPA. Fipa abstract architecture specification. 2002.

FIPA. *O framework FIPA*. 2003. Disponível em: <<http://fipa-os.sourceforge.net/>>. Acesso em: 17 mai. 2003.

FLOREA, A. M. 2002. Computer Science-WPI (Worcester Polytechnic Institute). Disponível em: <http://www.cs.wpi.edu/~cs525m/s02/Lecture_12.ppt>. Acesso em: 11 abr. 2005.

FRICITIONLESS, C. *Homepage do Friction Sourcing*. 2004. Disponível em: <<http://www.frictionless.com>>. Acesso em: 25 jun. 2004.

GUTTMAN, R. H.; MAES, P. Cooperative versus competitive multi-agent negotiations in retail electronic commerce. In: *Proceedings of the Second International Workshop on Cooperative Information Agents II, Learning, Mobility and Electronic Commerce for Information Discovery on the Internet*. [S.l.]: Springer-Verlag, 1998. p. 135–147. ISBN 3-540-64676-0.

- GUTTMAN, R. H.; MOUKAS, A. G.; MAES, P. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review*, v. 13:3, Junho 1998.
- HE, M.; JENNINGS, N. R.; LEUNG, H.-F. On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, v. 15, n. 4, Julho/Agosto 2003.
- INFO100. *Homepage da revista INFO*. 2003. Quinta edição da pesquisa INFO 100. Disponível em: <<http://info.abril.com.br/>>. Acesso em: 12 dez. 2003.
- JADE. *Homepage do Projeto JADE*. 2003. Disponível em: <<http://sharon.cselt.it/projects/jade/>>. Acesso em: 25 mai. 2003.
- LEEDS, U. of. 2005. Desenvolvido pelo Vision Group do School of Computing. Disponível em: <<http://www.comp.leeds.ac.uk/vision/cogvis/>>. Acesso em: 16 nov. 2005.
- LOMUSCIO, A. R.; WOOLDRIGE, M.; JENNINGS, N. R. A classification scheme for negotiation in electronic commerce. *Lecture Notes in Computer Science*, v. 1991, p. 19+, 2001.
- MANGAN, P. K. V. Implementação de sistemas multiagentes. COPPE/Sistemas - UFRJ, p. 1–33, Novembro 2001.
- MARIETTO, M. B. et al. Proposta de uma nova classificação para a Área de inteligência artificial distribuída: Focos de interesse e sistemas computacionais. In: *Proc. 8th. Ibero-American Conference on Artificial Intelligence (IBERAMIA'02), Open Discussion Track, Sevilha, Espanha*. [S.l.: s.n.], 2002. p. 229–238.
- PROJECT, Z. *Zeus open source*. 2004. Disponível em: <<http://www.btexact.bt.com/technologies/trials?doc=42926&page=1>>. Acesso em: 23 jun. 2004.
- REIS, L. P. *Coordenação em Sistemas Multi-Agentes: Aplicações na Gestão Universitária e Futebol Robótico*. Tese (Doutorado) — FEUP, Julho 2003. Capítulo 3.
- ROCHA, A. de R.; ELMO, M. d. S. J.; ALVES, J. C. Introdução aos agentes inteligentes e aos sistemas multiagente. UFLA - Universidade Federal de Lavras, DCC - Departamento de Ciência da Computação, p. 1–14, Abril 2003.
- ROCHA, A. P.; OLIVEIRA, E. Adaptive multi-issue negotiation protocol for electronic commerce. The Fifth International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000), Abril 2000.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence*. Second edition. [S.l.]: MIT, 2002.
- SBPC. 01 2005. *Jornal da Ciência*, órgão da Sociedade Brasileira para o Progresso da Ciência. Disponível em: <<http://www.jornaldaciencia.org.br/Detail.jsp?id=25009>>. Acesso em: 28 jan. 2005.
- SICHMAN, J. S.; ALVARES, L. O. Introdução aos sistemas multiagentes. In: *Anais da Jornada de Atualização em Informática - JAI 97*. [S.l.]: Sociedade Brasileira de Computação, SBC, 1997.

TURBAN, E. et al. *Electronic Commerce: A Managerial Perspective*. Primeira edição. [S.l.]: Prentice Hall, 1999.

VETTER, M.; PITSCH, S. Using autonomous agents to expand business models in electronic commerce. *International Journal of E-Business Strategy Management*, v. 1, 2000.

WEISS, G. *Multiagent Systems*. [S.l.]: MIT, 1999.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)