UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Aprendizagem de Máquina em um Ambiente para Negociações Automatizadas

Rômulo Nunes de Oliveira

Campina Grande, PB, Brasil
Março de 2006

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Aprendizagem de Máquina em um Ambiente para Negociações Automatizadas

Rômulo Nunes de Oliveira

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação (MSc).

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Modelos Computacionais e Cognitivos

Evandro de Barros Costa
Orientador
Herman Martins Gomes
Orientador

Campina Grande, PB, Brasil
Março de 2006

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

O148a Oliveira, Rômulo Nunes de
2006 Aprendizagem de Máquina em um Ambiente para Negociações Automatizadas – Campina Grande, 2006
123.: il.

Inclui bibliografia
Dissertação (Mestrado em Informática) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientadores: Evandro de Barros Costa e Herman Martins Gomes

1– Aprendizagem de Máquina 2– Negociação Automatizada 3– Inteligência Artificial 4– Redes Neurais I-Título

CDU 681.3.06

Resumo

A automatização do processo de negociação é um dos tópicos importantes e desafiadores no contexto da pesquisa em IA aplicada ao comércio eletrônico. Particularmente, um dos problemas relacionados à concepção de agentes de software negociadores diz respeito a dotá-los da capacidade de aprender e se adaptar ao dinamismo normalmente requerido no processo de negociação.

Neste trabalho, propõe-se um ambiente para construção de agentes inteligentes dotados das capacidades de negociar e aprender. Foram realizados dois experimentos para demonstrar a factibilidade da nossa proposta. Os experimentos mostraram a capacidade de aprendizagem dos agentes, segundo as técnicas empregadas: Redes Neurais, Aprendizagem por Reforço (Q-learning), Raciocínio Baseado em Regras e Raciocínio Baseado em Casos.

Abstract

Automated negotiation is a hot research topic in AI applied to e-commerce. Particularly, one of the problems concerning the design of negotiation agents is to enclow them with learning and adaptation capabilities in order to face the dynamism often required by negotiation processes.

This dissertation proposes an environment for building intelligent agents capable of negotiating and learning. Two experiments were done in order to demonstrate the feasibility of the proposed environment. The experiments show that the agents learned during the interaction process according to the used techniques: Neural networks, Q-learning, Rule-based Reasoning and Case-based Reasoning.

Agradecimentos

A Deus, por ter me ajudado e me dado a capacidade para realizar este trabalho. A Ele também, pela saúde de minha mãe, que recuperou-se de uma doença grave no coração em 2005.

À minha família, em especial a minha mãe e meu pai, por terem me apoiado em todo o período em que estive em Campina Grande.

Aos professores Evandro e Herman, que orientaram este trabalho com paciência e dedicação.

Ao Alan e ao Ig, por terem colaborado na implementação do sistema e ajudado com discussões e sugestões sobre o projeto.

Aos amigos da equipe de louvor da Igreja Presbiteriana em Maceió e em Campina Grande, pelo prazer que tenho ao louvar a Deus junto a eles, e por suas vidas dedicadas ao serviço de Deus. “Mas a hora vem, e já chegou...” (Jo 4:23).

Aos colegas do Mestrado e do LabPesquisas-UFAL, pelos incentivos e momentos de descontração.

A Aninha, Vera e Zeneide, que sempre se mostraram dispostas e empenhadas nas suas responsabilidades.

Aos membros da banca examinadora, pelas críticas e sugestões que contribuíram para o enriquecimento deste trabalho.

A todos meus amigos em Campina Grande e em Maceió, que sempre me ajudaram com palavras de incentivo e apoio.

Aos colegas radioamadores pela companhia durante minhas constantes viagens.

A todos que fazem a COPIN.

A CAPES, que apoiou financeiramente este trabalho.

A todos que diretamente ou indiretamente contribuíram para a conclusão deste trabalho.

Conteúdo

1 Introdução **1**

1.1 Motivação e Problemática 1

1.2 Objetivos e Relevância do Trabalho 2

1.3 Organização desta Dissertação 3

2 Fundamentação Teórica **4**

2.1 Comércio Eletrônico Mediado por Agentes 4

2.1.1 Negociações no Comércio Eletrônico 5

2.1.2 Agentes e Sistemas Multiagentes 7

2.1.3 Agentes Atuando como Negociadores 11

2.2 Aprendizagem de Máquina Aplicada à Negociação 13

2.2.1 Caracterização da Aprendizagem em Agentes 13

2.2.2 Redes Neurais 14

2.2.3 Raciocínio Baseado em Casos (RBC) 20

2.2.4 Aprendizagem por Reforço 22

2.2.5 Aprendizagem na Negociação 25

3 Trabalhos Relacionados **27**

3.1 Ambiente de Negociação 27

3.1.1 O Sistema Kasbah 28

3.1.2 Wreper 29

3.1.3 Framework por Fatima 30

3.1.4 Framework por Bartolini 31

3.1.5 Modelo Proposto por Gustavo de Paula 31

3.1.6 Modelo Proposto por Pinho Júnior 35
3.2 Aprendizagem e Modelo de Decisão 40
3.2.1 Utilizando RBC 41
3.2.2 Utilizando Q-learning 42
3.2.3 Utilizando Redes Neurais 44
3.3 Síntese dos Trabalhos Revisados 45

4 Um Ambiente para Negociações Automatizadas 47
4.1 Descrição do Ambiente 47
4.1.1 Variáveis de Decisão 48
4.1.2 Entidades Participantes 48
4.1.3 Protocolo da Negociação 50
4.2 Arcabouço do Negociador 53
4.2.1 Agente Mediador (AM) 54
4.2.2 Agente Gerenciador das Negociações (AGN) 56
4.2.3 Agente Executor (AE) 58
4.2.4 Agente Gerador de Informação (AGI) 59
4.3 Modelo de Aprendizagem 59
4.3.1 Tomada de Decisão $\times$ Aprendizagem 60
4.3.2 Facilidade na Utilização das Técnicas 61
4.4 Discussão Comparativa do Trabalho 64
4.4.1 Modelo Para Negociações Automatizadas 64
4.4.2 Ferramenta de Auxílio Didático 65
4.4.3 Conclusão 65

5 Estudos de Caso 66
5.1 Negociação com Pacotes Turísticos 66
5.1.1 Definição dos Negociadores 67
5.1.2 Descrição e Resultados do Experimento 73
5.2 Negociação em Pregão Eletrônico 75
5.2.1 Descrição do Pregão Eletrônico 75
5.2.2 Negociadores 78

5.2.3 Descrição e Resultados do Experimento 82

6 Considerações Finais **83**

6.1 Trabalhos Futuros 84

A Classificações no Comércio Eletrônico **96**

A.1 Tipo de Negócio 96

A.2 Modalidades de Comércio Eletrônico 97

A.3 Tipos de Comércio Eletrônico 101

B Plataforma JADE **103**

B.1 Arquitetura 103

B.2 Agentes JADE 105

B.3 Comportamentos 105

B.4 Troca de Mensagens 106

B.5 Pacote *jade.tools* 109

C TAC - *Trading Agent Competition* **111**

C.1 Introdução à Teoria de Leilões 111

C.1.1 Considerações Básicas 111

C.1.2 Os Tipos Básicos de Leilões 112

C.1.3 Estratégias de Atuação 113

C.2 O Ambiente TAC 114

C.2.1 Estudo do Ambiente TAC 114

C.2.2 O *Agentware* 119

C.2.3 O *Dummy Agent* 120

D Conjuntos de Regras para o Fornecedor 1 **122**

Lista de Figuras

2.1 Agente interagindo com o ambiente. 7
2.2 Representação de um sistema multiagente 9
2.3 O comércio eletrônico visto como um SMA *(Sistema Multiagente)* 12
2.4 Similaridade entre neurônios biológico e artificial 15
2.5 (a) Coleção linearmente separável (b) Coleção linearmente inseparável . . . 16
2.6 Uma rede neural multicamada 16
2.7 Treinamento para séries temporais 18
2.8 Estrutura básica de uma Rede Neural SOM 18
2.9 Treinamento: neurônio vencedor e seus vizinhos no tempo 18
2.10 Ciclo de solução de problemas em um sistema de RBC 21

3.1 Modelagem do Objeto da Negociação 32

4.1 Entidades no ambiente de negociação 49
4.2 Diagrama de seqüência AUML inicial 52
4.3 Troca de propostas entre SMAs Negociadores 53
4.4 Arquitetura do SMA Negociador 59
4.5 Tomada de decisão vista como uma função 60
4.6 Processo de aprendizagem visto como uma função 61

5.1 SMA Negociador 1 - Diagrama de colaboração 67
5.2 Função para o cálculo de $\alpha(t)$ [FSJ98] 68
5.3 SMA Negociador 2 - Diagrama de colaboração 71
5.4 Comparação da convergência entre os cenários 74
5.5 Acordos por ciclo de interação 74

5.6 Pregão Eletrônico 76
5.7 Sequência do Pregão Eletrônico 77

A.1 Tipos de negócio no comércio eletrônico. 97

B.1 Modelo de referência da FIPA para plataformas de agentes 104
B.2 Arquitetura JADE 105

Índice de Abreviaturas e Siglas

ACL – Agent-Communication Language, linguagem de comunicação de agentes

API – Application Programming Interface, interface de programação para aplicações

AR – Aprendizagem por Reforço

B2B – *Business to Business*, transações entre empresas

B2C – *Business to Consumer*, transações entre cliente e empresa

C2C – *Consume to Consumer*, transações entre consumidores finais

FIPA – *Foundation for Intelligent Physical Agents*, entidade responsável pela padronização em sistemas baseados em agentes.

IAD – Inteligência Artificial Distribuída

IAM – Inteligência Artificial Monolítica

IP – *Internet Protocol*

JADE – *Java Agent Development Framework*, arcabouço para desenvolvimento de sistemas multiagente

Jess – *Java Expert System Shell*. Máquina de inferência para regras de produção desenvolvida para linguagem Java.

MDP – *Markov Decision Process*, Processo Decisório de Markov

NBS – Negociação bilateral seqüencial

RBC – Raciocínio Baseado em Casos

RBR – Raciocínio Baseado em Regras

RN – Rede Neural

SMA – Sistema Multiagente

SOM – *Self-Organizing Map* ou *Mapa Auto-Organizável*, é um tipo de Rede Neural

TAC – *Trading Agent Competition*, competição anual de agentes negociadores em leilões.

URL – *Universal Resource Locator*, é o endereço de um recurso disponível numa rede.

XML – *Extensible Markup Language*, linguagem de marcação capaz de descrever diversos tipos de dados

Capítulo 1

Introdução

Através da WWW (*World Wide Web*) um novo mundo de opções passou a existir como, por exemplo, a possibilidade de realizar transações comerciais sem precisar sair de casa ou da empresa. Atualmente, os consumidores já podem , por exemplo, decidir que produto comprar através da Internet, identificar seus vendedores preferidos a partir de sua experiência nas compras, efetuar pagamentos, dentre outras facilidades. Assim, o comércio eletrônico pode ser definido como "o conjunto de transações comerciais que utilizam tecnologia da informação como apoio na efetivação de tais transações" [HJL03] ou como "o processo de compra e venda na Internet que frequentemente é realizado com utilização de softwares específicos" [Fil04]. Abstratamente, o foco principal deste trabalho está na contribuição que a Inteligência Artificial (IA) pode oferecer na modelagem computacional nesses processos de negociação.

Este capítulo está subdividido em três seções. Na Seção 1.1 estão as motivações para a pesquisa e elaboração deste trabalho. Na Seção 1.2 apresenta-se os objetivos da dissertação e, finalmente, na Seção 1.3 descreve-se a estrutura completa deste trabalho.

1.1 Motivação e Problemática

A automatização do processo de negociação é interessante, tendo em vista que ela busca resolver os problemas na escolha de propostas e métodos de venda de forma a aumentar os percentuais de vendas concretizadas e a qualidade da venda, o que sugere estudos na tentativa de uma ampliação para as possibilidades do comércio eletrônico. Associado a isso, é

também desejável que o ambiente de negociação eletrônico possua características presentes no comércio praticado no mundo real, tais como, flexibilidade nas propostas de acordo, possibilidade de sugestão de produtos, negociação do tempo de entrega, prazo de pagamento etc. Ainda em relação ao mundo real, observa-se algo que pode ser entendido como "experiência da compra", onde por exemplo, a tendência natural humana é dar menor importância para negociações que tenham características semelhantes a outras de uma negociação anterior em que não se obteve o sucesso esperado. Isso é um problema, onde uma possível solução é através da IA, requerendo o uso de suas técnicas para solucionar problemas de otimização, aprendizagem e raciocínio lógico. Para comércio eletrônico, por exemplo, existem técnicas específicas que podem ser utilizadas para aquisição da experiência de compra, assim como para a tomada de decisão e aprendizagem no ambiente de negociação, o que inclui a elaboração de diretivas para o rumo da negociação. De um ponto de vista mais específico, o atual estado da arte da pesquisa no tema revela preocupações focalizadas em apoiar decisões no curso da negociação, assim como mecanismos de aprendizagem automática.

1.2 Objetivos e Relevância do Trabalho

O objetivo geral desta dissertação é a proposição de um ambiente de suporte à negociação em *e-commerce*[1] dotado da capacidade de aprender. O trabalho abrange o estudo de técnicas de aprendizagem automática para dar suporte aos agentes de negociação. São abordadas as técnicas de aprendizagem automática em IA, identificando alternativas para tornar as negociações mais confiáveis e aumentar as chances dos agentes autônomos realizarem bons negócios para as partes envolvidas. Assim, o foco principal da pesquisa é a aprendizagem para dar suporte ao processo de negociação. Além disso, pretende-se realizar experimentos para demonstrar a factibilidade do ambiente proposto focalizando a capacidade de aprendizagem dos agentes negociadores.

A negociação eletrônica e automatizada por agentes é algo que já vem sendo trabalhada e apresenta alguns pontos fracos a serem explorados. A relevância deste trabalho está na melhoria do suporte à decisão e aprendizagem. Existe a necessidade de um ambiente em que se possa realmente usar e aplicar os agentes de negociação: um arcabouço para apoiar a negoci-

[1] O mesmo que "comércio eletrônico", ou ainda "mercado virtual".

ação. Aperfeiçoar os estudos anteriores e disponibilizar esse ambiente como arcabouço para negociação, onde futuros estudos poderão usá-lo para testar novas teorias e métodos. Existe também a possibilidade de sua utilização no mercado, aplicado diretamente como solução em empresas que têm complexidade logística ou quer aperfeiçoar seu sistema de vendas. Este ambiente para negociações, associado às ferramentas disponibilizadas para construção de agentes negociadores inteligentes também é de grande valor no meio acadêmico, onde disciplinas que abordam temas da Inteligência Artificial podem adotá-lo para experiências didáticas entre os alunos.

1.3 Organização desta Dissertação

Este trabalho está organizado em 6 capítulos. O Capítulo 2 compreende a descrição do comércio eletrônico e sistemas multiagente, que são base para discussão sobre a atuação dos agentes como negociadores no comércio eletrônico. Além disso, neste capítulo comenta-se o tema aprendizagem de máquina e como este pode ser utilizado em um ambiente de negociação. No Capítulo 3 apresenta-se uma classificação de negociadores autônomos, frutos de pesquisas anteriores e uma discussão sobre suas implementações, sob ponto de vista das técnicas, abordagens e etapas do processo de compra/venda que utilizam. Neste mesmo capítulo são apresentados os trabalhos relacionados à aprendizagem de máquina e tomada de decisão, que ilustram a eficiência das técnicas da Inteligência Artificial (IA) em diversos contextos. No Capítulo 4 descreve-se um arcabouço para construção de agentes inteligentes para o comércio eletrônico, utilizando-se dos recursos da IA. Ainda neste capítulo, existe uma discussão comparativa com os trabalhos correlatos, onde são identificadas as contribuições da presente dissertação. No Capítulo 5 mostra-se um estudo de caso, onde está demonstrado a utilização da ferramenta e a influência da aprendizagem na negociação. Finalmente, no Capítulo 6 apresenta-se as conclusões finais, contribuições da dissertação e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo contém o embasamento teórico para compreensão do presente trabalho. Ele está divido em duas partes: na primeira, descreve-se um pouco sobre a história do comércio eletrônico. Além disso, discute-se o tema agentes e sistemas multiagentes, definindo-se tais termos e a forma na qual os agentes podem atuar no comércio eletrônico. A segunda parte enfoca o tópico aprendizagem de máquina, onde se descreve como ela está caracterizada nos agentes e quais as principais técnicas de aprendizagem utilizadas.

2.1 Comércio Eletrônico Mediado por Agentes

A necessidade de negociar surgiu há muito tempo, na época do escambo, onde não existia moeda e as mercadorias eram trocadas entre si, sem equivalência de valor [Ban06]. Essa necessidade de negociar, de trocar mercadorias, nos acompanhou até hoje. Após séculos de prática, o mercado evoluiu a ponto de duas pessoas distantes geograficamente conseguirem fechar um negócio com total controle, incluindo garantias de entrega, qualidade, pagamento etc.

Nos dias de hoje, a grande concorrência proporcionada pelo avanço da tecnologia, fez com que as empresas adotassem a estratégia do “bom relacionamento” com seus clientes. Assim, o que antes era visto totalmente como um jogo competitivo passou a ser tratado como um jogo colaborativo, onde empresa-cliente se relacionam buscando acordos benéficos para ambos [Mel05]. Para a empresa, o preço para manter esse relacionamento íntimo com seus clientes é alto, porque, agora, ela tem que se investir em conhecer as preferências individuais

dos clientes, investir em conforto e segurança. Considerando que em grandes empresas o fluxo de negociações é alto devido ao alto número de clientes, surge o problema de demanda para garantir esse relacionamento personalizado com seus clientes.

A Inteligência Artificial (IA) possui técnicas que podem ajudar na modelagem computacional de processos de negociação em geral, no intuito de aproximar o modelo computacional do modelo real. A IA é uma subárea da Ciência da Computação que compreende o projeto de sistemas computacionais que exibam características associadas à inteligência quando inerente ao comportamento humano; inclui o estudo das faculdades mentais através do uso de modelos computacionais [TXF95; Win88]. Nesse sentido, a Inteligência Artificial procura imitar o comportamento e características humanas em um sistema computacional, podendo ser utilizada na resolução de problemas cognitivos em diversos contextos. Quanto a localização espacial das soluções, a IA possui as abordagens: a Monolítica (IAM) e a Distribuída (IAD). A IAM envolve sistemas simples sem modularidade, como é o caso de *sistemas especialistas*. O funcionamento da IAD depende de um determinado conjunto de partes (ou módulos) para resolver de modo cooperativo um determinado problema. Sua modularidade para encontrar soluções de problemas está diretamente ligada ao conceito de agentes, que será definido na Subseção 2.1.2.

Em seguida será explanado sobre o advento da tecnologia como auxiliadora da prática do comércio; o que são agentes e sistemas multiagente; e como esses agentes contribuem no comércio eletrônico.

2.1.1 Negociações no Comércio Eletrônico

Desde o seu surgimento nas últimas décadas do século passado, o comércio eletrônico vem crescendo cada vez mais em diferentes formas na rede mundial de computadores, ou Internet. A forma mais comum de realizar comércio na Internet é através das lojas virtuais, que podem ser acessadas pelas suas URLs[1] e possuem um modelo de interação semelhante a um supermercado como será detalhado a seguir.

A facilidade de acesso às informações é a característica responsável pelo rápido crescimento do mercado [Fil04]. Tanto empresas como consumidores podem ampliar seu nicho de mercado, seja a empresa interagindo com mais fornecedores/colaboradores ou com mais

[1] *Universal Resource Locator*, é o endereço de um recurso disponível numa rede.

clientes, seja os clientes interagindo com mais empresas. Essa facilidade no acesso às informações, muitas vezes através de sites de busca ou ferramentas especializadas, é a razão pela qual os parceiros de negócio se encontram facilmente para negociar.

As empresas também se utilizam da Internet como grande responsável pela diminuição de seus custos. Procurar novos parceiros, recursos de tecnologia é uma tarefa significativamente mais fácil em relação ao modo do comércio real, onde na maioria das vezes recorre-se ao catálogo impresso, panfleto de propaganda ou representante.

Por outro lado, o público consumidor também é beneficiado com essa facilidade na obtenção de informações com o aumento da concorrência entre as empresas. Desta forma, as empresas devem inserir diferenciais para que possam ter evidência nesta concorrência. Por exemplo, informações disponíveis sobre os clientes (consumidores ou outras empresas) podem ser utilizadas para personalização da negociação com cada cliente; pode-se permitir a negociação de outros termos da compra, além do preço (prazo de pagamento, tempo de entrega, garantia)[Jun02; Fil04].

O comércio eletrônico pode ser definido como qualquer forma de negociação em que as partes interagem eletronicamente, caracterizado pela utilização das tecnologias da informação e comunicação para condução dessas transações comerciais [Pau01; WC99]. As atividades fundamentais para a conclusão de uma negociação também necessitam recursos da tecnologia da informação em conjunto com a Internet para sua execução, como por exemplo, a busca de produtos, identificação de fornecedores, elaboração de pedidos, efetivação do pagamento, dentre outras. Apesar disso, esse novo modelo para negociação mostra apenas transformação dentro de uma empresa isoladamente. Benefícios maiores também podem ser alcançados com a integração da cadeia produtiva, de tal forma que as ofertas e demandas de uma empresa possam ser atendidas com maior agilidade e possibilitando a redução de custos [Alm05].

Historicamente, a utilização da tecnologia em negociações começou na década de 70, quando os bancos utilizaram redes privadas para suas transações. Em 80, na França, os negociadores passaram a trocar mensagens através das linhas telefônicas em um sistema chamado *Minitel*. Outras tecnologias surgiram e se desenvolveram para lidar com essas informações e documentos, em que, por exemplo, a criptografia obteve um vasto campo de aplicação. Então, um conceito mais adequado sobre o comércio eletrônico é a troca de informações por

meios eletrônicos que facilitam e possibilitam as transações comerciais. Assim, o comércio eletrônico praticado hoje é o resultado da evolução desses modelos antigos, pesquisas e desenvolvimento de tecnologias [Fil04]. Mais detalhes sobre classificação, tipos, aplicações, podem ser consultadas no Apêndice A.

2.1.2 Agentes e Sistemas Multiagentes

Agentes, são entidades que podem ser vistas como percebendo um ambiente por meio de sensores e atuando no mesmo por meio de atuadores [RN04]. Essa definição pode ser ilustrada através da Figura 2.1, e de acordo com ela, um agente depende de três fatores: o ambiente a sua volta, seus sensores e seus atuadores.

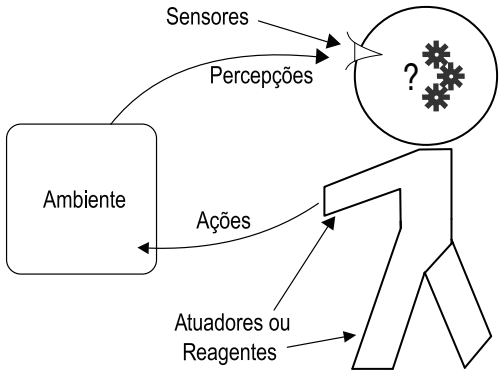

Figura 2.1: Agente interagindo com o ambiente.

Uma definição mais formal sobre agentes é encontrada em [WJ95], onde são apresentadas duas noções conceituais. A primeira, e mais abstrata, diz que um agente deve possuir algumas das propriedades abaixo:

Autonomia - capacidade de desempenhar funções de forma autônoma, sem intervenção de terceiros.

Capacidades sociais - capacidade de interagir com o mundo que o rodeia através da interpretação e geração de mensagens numa linguagem que seja entendida em toda sociedade onde o agente interage.

Reatividade - capacidade de reagir aos estímulos externos desencadeando ações de resposta.

Proatividade - capacidade de desencadear ações mesmo sem a ocorrência de estímulos exteriores que os desencadeiem. São ações relacionadas ao cumprimento de metas aspiradas.

O segundo conceito, aplicado normalmente a sistemas que, além de possuírem as características anteriores, são implementados e idealizados com origem em características encontradas nos seres humanos.

Conhecimento - capacidade de criar e/ou aperfeiçoar um modelo do mundo que rodeia o agente. O conhecimento é representado através de regras e fatos.

Crenças - valores em que o agente acredita, embora não possua provas da sua autenticidade ou validade

Intenções - metas que o agente planeja atingir no futuro

Obrigações - compromissos que o agente deve cumprir

Além dessas, que foram mapeadas do comportamento humano, ainda pode-se incluir outras características na definição de agente:

Mobilidade - capacidade do agente se movimentar de máquina para máquina através da rede.

Veracidade - característica que garante que o agente não comunica propositadamente falsa informação.

Benevolência - característica que diz que o agente se comporta sempre de forma a satisfazer os objetivos dos que o rodeiam

Racionalidade - capacidade do agente de imitar o comportamento racional, de modo a obter sempre o máximo benefício dos seus atos.

Como se pode perceber, a definição dada por Russell [RN04] é bem mais abstrata do que a última, mas, como o próprio Russell diz, “a idéia de agente deve ser utilizada como uma ferramenta de análise de sistemas e não como uma função rígida que divida o mundo em agentes e não-agentes”. Assim, para o entendimento desta dissertação, basta assumir que

agentes são softwares independentes, autônomos, sociais, capazes de reagir a estímulos e perceber mudanças no ambiente.

Quando se fala em sistemas multiagentes, normalmente refere-se a um conjunto de agentes, em caráter colaborativo ou competitivo, inseridos em um determinado contexto. Um sistema multiagente permite que grandes problemas possam ser divididos em outros menores, embora seja exigida uma complexidade maior na solução [Fon01]. Um SMA pode ser fechado ou aberto, dependendo da flexibilidade do sistema a ponto de aceitar a entrada e saída de agentes no sistema.

Para ilustração do poder de simplificação que um sistema multiagente pode trazer a um problema complexo, na Figura 2.2 está representado um sistema multiagente formado por seis agentes. Perceba que existe interação entre eles, e que cada agente age em subespaços no ambiente. Isso significa que um determinado problema foi decomposto para vários agentes, onde esses agentes podem estar fortemente acoplados[2] ou não [Jen99] e cada um está preparado para resolver sua própria parcela do problema maior.

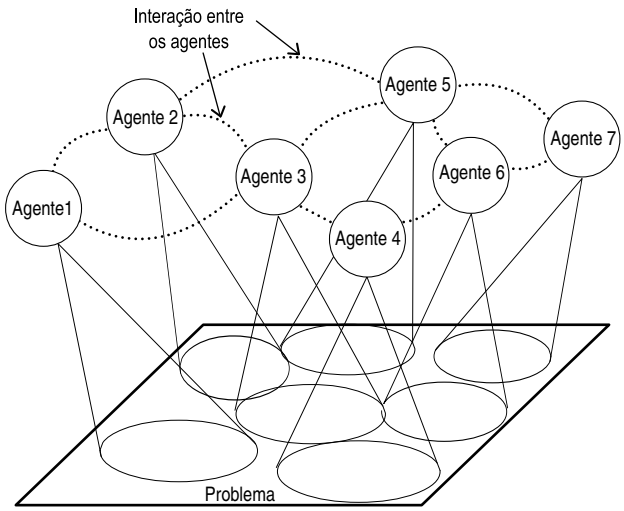

Figura 2.2: Representação de um sistema multiagente

Comunicação

No processo de desenvolvimento de um software baseado em agentes, uma das primeiras preocupações é a de como os agentes irão interagir. Uma interação caracteriza uma relação

[2]O acoplamento é o nível de inter-dependência entre os módulos de um programa de computador.

dinâmica entre dois ou mais agentes, durante a qual estes estão em contato, seja diretamente, seja por intermédio de outros agentes ou através do ambiente que eles dividem. Essa comunicação deverá ser conduzida baseada em algumas normas, pois se trata de comunicação entre máquinas. Por isso, juntamente com um protocolo que proporcione a comunicação física, se faz necessário aplicar regras para que essa comunicação exista coerentemente [Fon01]. No caso do comércio eletrônico, cada agente possui um protocolo de negociação para que exista a coerência na comunicação e o negociador possa saber como dialogar, em qual momento, e com quem.

Em um ambiente compartilhado, agentes comunicam-se buscando atingir seus objetivos. Em uma sociedade, as ações dos agentes devem ser coordenadas e regidas, possivelmente por protocolos, seja para cooperarem, onde os agentes irão interagir em conjunto para a execução do objetivo comum do sistema, seja para competir, onde agentes com intenções diferentes, negociarão a fim da obtenção de seus objetivos. Sendo assim, a comunicação entre tais agentes mostra-se um fator crucial para um bom funcionamento de um sistema complexo. Para que haja a cooperação/competição entre estas entidades de software, é necessário um mecanismo de comunicação, sendo este composto por uma linguagem e por um meio de comunicação que a transmite.

Os protocolos de comunicação entre agentes se baseiam na área de conhecimento como filosofia da linguagem, especialmente a teoria dos atos e da fala [BVlFM01]. Estes protocolos são utilizados para definirem como será a comunicação entre os agentes, os atos que devem ser realizados em uma determinada situação.

Em cenários onde exista a possibilidade da posterior integração de novos subsistemas (cenários evolutivos), o uso de uma linguagem compartilhada para interação entre os agentes provê maior flexibilidade, pois todos sistemas utilizarão uma linguagem comum, evitando que o crescimento do número de subsistemas implique na compreensão de novas definições sintáticas e semânticas por parte dos sistemas existentes [Alm04]. Exemplos de tais linguagens são FIPA-ACL[FIP06], KQML [FLM97].

Compartilhamento de conhecimento

Em um sistema orientado à agentes, para que um grupo de agentes compartilhe um mesmo vocabulário ao fazerem asserções e perguntas sobre um determinado domínio, é necessário

que haja um consenso em relação ao significado e representação do conhecimento do domínio entre os agentes, para evitar ambigüidades.

Pode-se obter este consenso através de duas maneiras [Alm04]:

- Através da difusão do conhecimento de cada subsistema aos outros: esta solução obriga que cada subsistema compreenda os conceitos de cada um dos outros subsistemas. Logo, esta prática mostra-se impraticável em cenários com a existência de vários subsistemas;

- Através da definição de um conhecimento compartilhado por todos os subsistemas: nesta solução, a utilização de ontologias mostra-se evidente, provendo um consenso entre os agentes de forma compartilhada. Uma ontologia pode fornecer um vocabulário comum para que os agentes se comuniquem de acordo com o que foi modelado.

A utilização de ontologias para a especificação dos conceitos compartilhados pelos subsistemas, facilita sobremaneira a integração dos mesmos, tornando possível que novos subsistemas sejam integrados desde que respeitem os conceitos definidos pela ontologia [BVlFM01].

2.1.3 Agentes Atuando como Negociadores

No contexto da negociação, as partes envolvidas no negócio podem ser representadas por Agentes, e assim obter o encapsulamento de comportamentos e conhecimentos, que devem ser independentes. A Figura 2.3 mostra uma aplicação da definição de Russell [RN04] no âmbito do comércio eletrônico em uma negociação bilateral: de um lado o comprador e do outro o vendedor, ambos podem ser vistos isoladamente como uma entidade definida por Russell, representada na Figura 2.1. Observe interação dos negociadores entre si e no ambiente, através ações e percepções.

É conhecida como *negociação bilateral* a interação entre duas entidades, inseridas em um contexto pré-definido, com o objetivo de alcançar um acordo [Deb03]. Uma negociação bilateral pode ser entre cliente-empresa[3] (C2B), empresa-empresa[3] (B2B) ou entre consumidores[3] (C2C). Por sua vez, *negociações multilaterais*, onde existem um maior

[3]Consultar o Apêndice A para mais detalhes

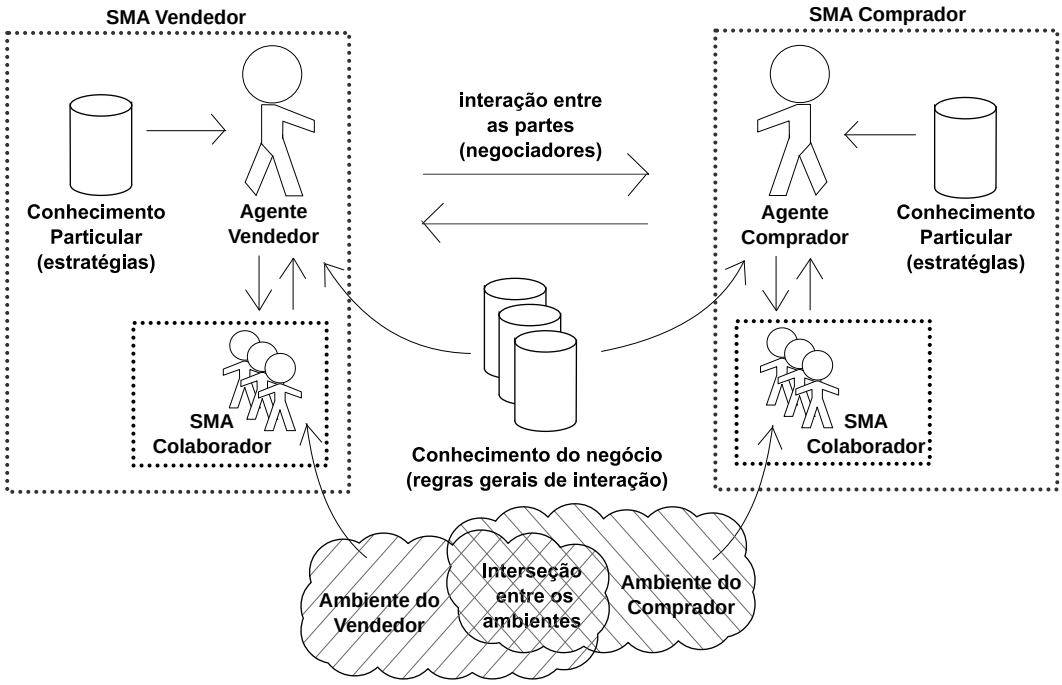

Figura 2.3: O comércio eletrônico visto como um SMA *(Sistema Multiagente)*

número de entidades, podem ser vistas, simplificadamente, como múltiplas negociações bilaterais ocorrendo paralelamente [KL02]. Logo, a Figura 2.3 abrange e resume bem a comunicação de quaisquer dois agentes em negociação. Essa composição com múltiplas negociações bilaterais é apenas ilustrativa, pois existe uma diferença entre múltiplas negociações bilaterais e uma negociação multilateral: nesta última, o protocolo de negociação considera a existência de mais de duas entidades envolvidas no jogo, considerando tudo como uma única negociação [KL02].

Negociação automatizada em comércio eletrônico diz respeito a um jogo de proposta e contra-propostas por parte dos jogadores (agentes), que desempenham papéis de compradores e vendedores, buscando alcançar uma situação de acordo sobre os objetos, a qual é alvo da negociação [Fil04; Deb03]. A análise de fatores que influenciam indiretamente o negócio, por exemplo, tendências e valores de mercado, taxas do governo e outros, são importantes no processo de negociação no sentido da adoção de um caráter concessivo ou ganancioso, de forma a otimizar o lucro e não acarretar prejuízos ao negócio.

2.2 Aprendizagem de Máquina Aplicada à Negociação

Em princípio, o termo "aprendizagem" refere-se ao conhecimento adquirido e retido na memória por experiência, esforço próprio, observação ou estudo [Fer99]. Da mesma forma, no contexto da Ciência da Computação, aprendizagem também se refere à aquisição de conhecimento, mas, em forma de dados interpretados, novas regras, ou mesmo na representação de uma experiência positiva ou negativa, que são aproveitados em situações futuras. Com isso, apresenta-se como grande interesse dotar os chamados agentes inteligentes de capacidades de auto-adaptação e aprendizagem, permitindo-lhes modificar seu comportamento em função da experiência adquirida e da percepção de alterações no ambiente [Fon01].

2.2.1 Caracterização da Aprendizagem em Agentes

Segundo José Fonseca [Fon01], o processo de aprendizagem pode ser classificado em 5 categorias:

- por implantação: quando o conhecimento é adquirido por uma entidade ou especialista externo, sem que o próprio agente intervenha nesse processo.
- por transformação: quando a aprendizagem se dá internamente no agente, com a adaptação do conhecimento externo recebido.
- por exemplos e prática: quando a aprendizagem é adquirida através de exemplos ou experiência prática, sejam eles positivos ou negativos.
- por analogia: quando são usados casos para representar um conhecimento, e por analogia encontra-se soluções para novas situações.
- por descoberta: quando o agente aprende por experiências ou testes, gerando e testando hipóteses.

Quanto à validação do conhecimento adquirido, existem algumas formas para se confirmar a aprendizagem, que podem estar no tipo de treinamento ou na característica do algoritmo de aprendizagem [Fon01]:

- aprendizagem supervisionada: o conhecimento que o agente deve aprender é indicado por alguma entidade externa, que funciona como um professor. O agente então deve adquirir esse conhecimento o mais próximo possível dessa indicação. Normalmente são fornecidos exemplos completos (dados de entrada, saída, comportamento esperado ou significado), que são usados no algoritmo de aprendizagem.

- aprendizagem por reforço: o reforço é a informação sobre a qualidade de uma ação desempenhada pelo agente. Essa informação influencia futuras decisões na utilização da mesma ação. Então, a aprendizagem por reforço acontece através de críticas positivas e negativas sobre uma ação relacionada a um problema.

- aprendizagem não-supervisionada: para esse tipo de aprendizagem não se fornecem exemplos completos, o agente aprende sobre relações e características no domínio através da auto-organização dos dados de entrada.

Para cada problema a ser resolvido, ou conhecimento a ser adquirido, existe um modelo de aprendizagem mais adequado. Logo, em uma comunidade multiagente, é perfeitamente possível coexistirem agentes adotando métodos e técnicas de aprendizagem diferentes. Dentre os diversos estudos e pesquisas sobre aprendizagem, algumas das técnicas mais conhecidas na IA são: Redes Neurais, RBC e Q-learning.

2.2.2 Redes Neurais

O cérebro humano é capaz de processar informações de forma rápida e consideravelmente eficiente em relação a qualquer computador digital existente. Enquanto, por exemplo, a tarefa de identificação e reconhecimento de faces de pessoas em uma cena é uma tarefa difícil e muito longa para ser feita por um computador, o nosso cérebro pode fazê-la em aproximadamente 100-200ms [Hay01]. No modelo biológico humano, os neurônios são células responsáveis pela propagação dos impulsos nervosos. Nosso sistema nervoso é composto por, aproximadamente, 10 bilhões de neurônios [TXF95]. Um neurônio recebe estímulos e os propaga para o próximo neurônio, através do *axônio*, se os estímulos recebidos nos *dendritos* foram suficientes para isso. Na tentativa de aproveitar a idéia da capacidade do modelo biológico humano, surgiram as Redes Neurais Artificiais que são constituídas por unidades

de processamento simples. Inspirados em neurônios biológicos, os neurônios artificiais possuem sinais de entrada, cada um associado e influenciado por seu peso sináptico. Esses sinais são somados e aplicados a uma função de ativação, e assim, a saída do neurônio é ativada ou não [MP43; Hay01]. A Figura 2.4 ilustra a inspiração biológica na concepção do modelo artificial do neurônio, em que $x_1, ..., x_n$ são as entradas da rede, e os valores $w_1, ..., w_n$ são os pesos sinápticos associados à cada valor x da entrada.

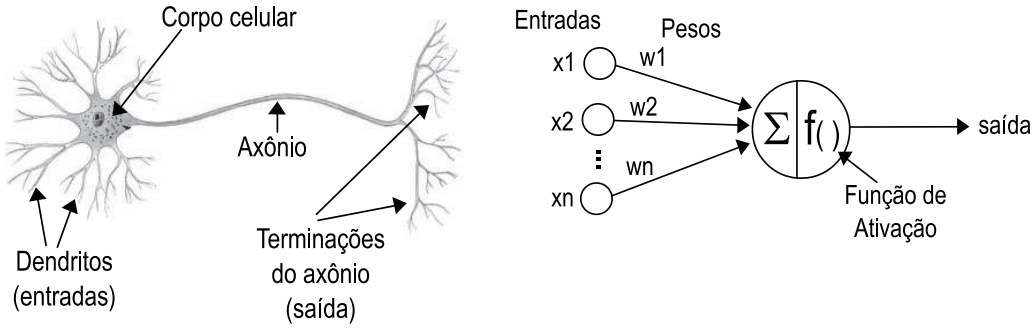

Figura 2.4: Similaridade entre neurônios biológico e artificial

Características Básicas

Um neurônio individualmente é capaz de resolver problemas linearmente separáveis. Nesse caso, só é possível se as duas coleções envolvidas ocuparem regiões no espaço (no caso $\mathbb{R}^n$) de tal forma que possibilite a passagem de um hiperplano separando-as. Para o caso $\mathbb{R}^2$, uma reta faz a separação dos conjuntos [Kov02]. A Figura 2.5 ilustra as situações de problemas linearmente separáveis e inseparáveis. Para que seja possível utilizar uma rede neural para resolver problemas linearmente inseparáveis é necessário agrupar neurônios em múltiplas camadas [Kov02]: camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. A importância de camadas intermediárias se dá quando o número de neurônios na camada de entrada é grande. A utilização de camadas intermediárias torna a rede capaz de aprender informações complexas (ou de ordem elevada) [Hay01]. Um exemplo de uma rede neural com neurônios agrupados em camadas pode ser visto na Figura 2.6.

O conhecimento da rede está representado nas ligações sinápticas, através dos pesos associados a elas. Dessa forma, existe um peso influenciando cada ligação entre os neurônios, e, para uma rede como no exemplo da Figura 2.6, temos 18 ligações sinápticas e, portanto,

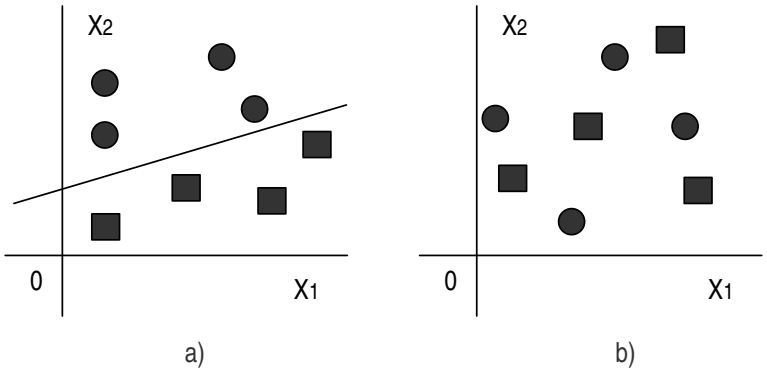

Figura 2.5: (a) Coleção linearmente separável (b) Coleção linearmente inseparável

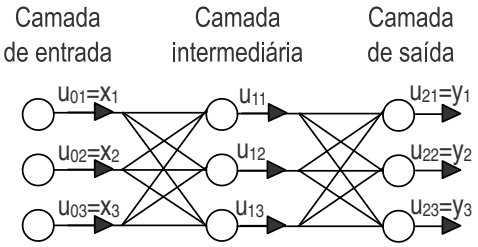

Figura 2.6: Uma rede neural multicamada

18 pesos sinápticos. Os valores de saída de cada neurônio U_{ij}, em cada ligação, são influenciados pelos pesos sinápticos antes de atingirem o próximo neurônio na camada seguinte [Hay01]. Quando uma rede neural possui todos os seus neurônios de uma camada ligados a todos os neurônios da próxima camada – isso para toda a rede – ela é chamada de "*fully connected*" (totalmente conectada) [TXF95]. A Figura 2.6 também ilustra uma rede deste tipo.

Existe também um modelo de rede neural que é realimentada com sua saída, chamada de rede recorrente (ou rede com realimentação, rede com ciclos ou rede com *feedback*). O objetivo da realimentação é fazer a rede convergir e se estabilizar em algum instante t, após algumas iterações. Este tipo de rede é extremamente importante quando se quer trabalhar com *memórias associativas*[4]. Um exemplo de redes recorrentes é a rede de Hopfield. Redes diretas (*feedforward*) são aquelas que não possuem ciclos nas suas ligações sinápticas, como por exemplo, a rede da Figura 2.6 [ABO00].

[4]Trata-se o reconhecimento de padrões, onde um novo caso apresentado é associado a um dos padrões previamente treinados.

Treinamento e Aprendizagem

Uma rede neural tem que ser devidamente treinada para ajustes nos pesos sinápticos de cada neurônio de forma individual, e esse treinamento pode ser supervisionado ou não-supervisionado. O treinamento supervisionado consiste basicamente em se ajustar os pesos sinápticos, diante apresentação de conjuntos de treinamento (*informações de entrada* $\times$ *saída desejada*) sobre o conhecimento que se deseja aprender. Assim, a cada entrada apresentada, os pesos dos neurônios são ajustados gradativamente. O objetivo é que, depois de treinada, a rede possa identificar padrões semelhantes aos padrões fornecidos no treinamento e assim, por generalização, classificar uma nova entrada como pertencente ao conjunto usado no treinamento ou não [ABO00]. O treinamento não-supervisionado deixa a rede livre para organizar os casos apresentados conforme a proximidade de identificação de um neurônio. Para cada padrão apresentado à rede, um neurônio será identificado com essa entrada e o seu peso sináptico é ajustado para a melhor identificação desta entrada. Os pesos dos neurônios vizinhos também são ajustados, porém com uma intensidade menor, assim garantindo, na organização da rede, que os padrões semelhantes estejam próximos [ABO00]. O processo de treinamento de uma rede neural é sempre necessário. Assim, se exige um tempo extra quando na utilização de tais métodos, destinados à aprendizagem. Existe a necessidade de se prever mudanças no domínio do problema que com o passar do tempo pode sofrer alterações. Por exemplo, em um sistema de identificação de padrões, podem surgir novos padrões, sendo necessário um novo treinamento da rede. Então, conforme a rede necessite de mudanças, será necessário treiná-la novamente.

As redes neurais também podem ser utilizadas para a previsão de séries temporais, seja para séries lineares ou não-lineares. Para isso, no treinamento, utiliza-se como entrada da rede os n primeiros termos da série, e na saída o termo $n+1$, como objetivo da aprendizagem. Assim os pesos sinápticos são ajustados para aprender sobre a série temporal de um domínio, que é representado por um conjunto de treinamento com amostragens da série. A Figura 2.7 exemplifica o treinamento de uma rede neural para previsão de séries temporais. Os termos iniciais a, b e c são submetidos a entrada da rede, e saída d' é comparada a saída desejada d para os ajustes de pesos.

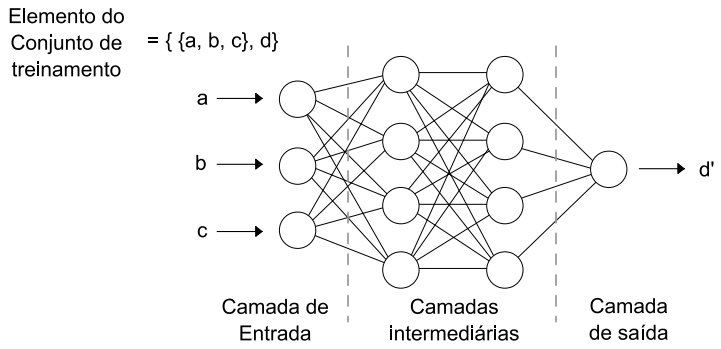

Figura 2.7: Treinamento para séries temporais

Redes SOM

Um outro modelo de Rede Neural, conhecido como Mapa Auto-Organizável, ou Redes SOM (*Self-Organizing Map*), utiliza o treinamento não-supervisionado e competitivo[5] para classificar dados automaticamente. Para fins didáticos, a camada de saída de uma rede SOM pode ser representada como uma matriz $m \times n$. Cada um desses neurônios na matriz possui uma ligação sináptica para cada neurônio de entrada. Assim, se por exemplo tiver 3 neurônios na camada de entrada e 25 na camada de saída, então existirão 75 ligações entre essas camadas. Esse exemplo pode ser visualizado através da Figura 2.8.

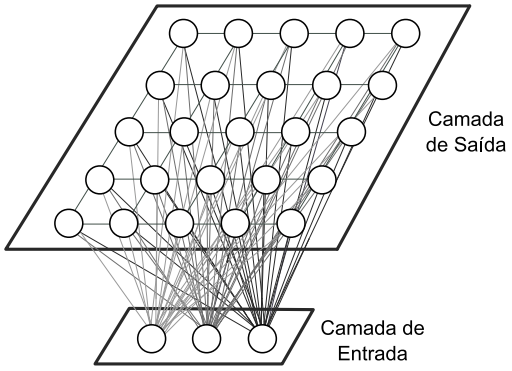

Figura 2.8: Estrutura básica de uma Rede Neural SOM

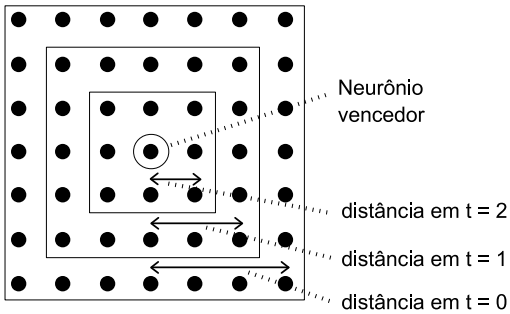

Figura 2.9: Treinamento: neurônio vencedor e seus vizinhos no tempo

O treinamento da Rede SOM segue a seguinte ordem:

1. Os pesos sinápticos das conexões entre a camada de entrada e saída são iniciados aleatoriamente e possuem valores diferentes entre eles.

[5]Os neurônios literalmente competem, onde o vencedor tem seus pesos ajustados, bem como o mesmo ocorre com uma vizinhança ao seu redor.

2. Para cada padrão apresentado na camada de entrada, apenas um neurônio será vencedor, que receberá um incentivo por isso, para que fique mais semelhante ao padrão de entrada. O neurônio vencedor v é aquele menor distância ao padrão de entrada, medida através da Distância Euclidiana[6] [Hay01; ABO00]:

3. Os neurônios vizinhos também recebem uma recompensa, porém, com menor valor, conforme o grau de vizinhança diminui ou conforme o tempo aumenta (como mostra a Figura 2.9).

4. No final do treinamento, a rede convergiu para um estado onde o domínio apresentado no treinamento está organizado e representado na camada de saída.

A Rede SOM é utilizada para organizar dados onde não existe uma metodologia classificatória definida, ou quando o conjunto de atributos envolvidos na classificação é muito grande. Embora tenha esse poder de convergência para um estado que representa a organização do domínio, sua organização não possui semântica direta, sendo necessário um trabalho adicional para isso.

Aplicabilidade da Rede Neural

As redes neurais são geralmente aplicadas em tarefas de classificação e previsão com base em séries temporais. São utilizadas quando o domínio e solução preenchem os requisitos abaixo:

- Não se pode traduzir facilmente por funções matemáticas o padrão a ser aprendido. Após treinada, uma rede pode trabalhar com classificação, agrupamento, previsão de séries temporais, com um nível de erros aceitável.

- Existe um conjunto de treinamento que representa bem o domínio do problema. O desempenho da rede não depende apenas do procedimento de treinamento, mas também do conjunto utilizado para isso.

- Em geral, não se deseja uma semântica para o conhecimento representando, apenas sua aplicação. Após treinada, o conhecimento aprendido pela rede está distribuído nos

[6] A distância euclidiana é a raiz quadrada da soma dos quadrados das diferenças de valores para cada variável n em $\mathbb{R}^n$.

pesos sinápticos em cada neurônio de forma que não se pode facilmente visualizar o conhecimento representado, dando-lhe uma semântica.

2.2.3 Raciocínio Baseado em Casos (RBC)

Raciocínio baseado em casos ou RBC [RS89; Abe01; Cun98] tenta resolver o problema da representação e processamento de conhecimento tendo como experiência *casos* semelhantes anteriores [Abe01]. A motivação para o RBC vem do raciocínio por analogia usado para resolver problemas, que é construído também a partir de experiências vividas ou outros conhecimentos adquiridos [TXF95; Abe01]. Imagine a situação seguinte: em uma loja de sapatos existem dois vendedores com o mesmo conhecimento básico sobre os fabricantes, valores, descontos, enfim, tudo sobre a loja e os produtos à venda. Só que um dos vendedores começou o seu trabalho no ramo há 2 meses e o outro já trabalha há 5 anos. Nesse caso, é fácil prever que o vendedor mais antigo, pela sua experiência com vendas de sapatos, pela sua vivência cotidiana, vende melhor e realiza mais negociações bem-sucedidas do que o vendedor novato. Da mesma forma, um agente negociador (Capítulo 4.1.2), lida com situações em vendas que, muitas vezes, são parecidas com outras já vivenciadas onde se obteve sucesso ou não. A idéia central do RBC é, portanto, ajustar soluções que foram usadas para resolver antigas situações e utilizá-las na resolução de situações novas [ABO00; RS89; Abe01].

Um *caso* é a representação completa de um problema-solução associado à uma medida de *utilidade*[7] que representa a eficácia da solução. O problema, por sua vez, é descrito através de atributos e valores. Assim, o conjunto de casos é indexado por seus atributos, e a pesquisa pelo caso semelhante considera todos os atributos e seus graus de importância [WW03; Abe01].

Etapas de um Sistema RBC

O ciclo de funcionamento de um sistema de RBC é apresentado na Figura 2.10 e suas etapas são expostas a seguir.

Resumidamente, os passos de solução podem ser descritos como segue [Abe01; Cun98]:

[7] O termo *utilidade*, utilizado frequentemente no presente trabalho, representa quantativamente a eficácia/qualidade da solução aplicada ao problema.

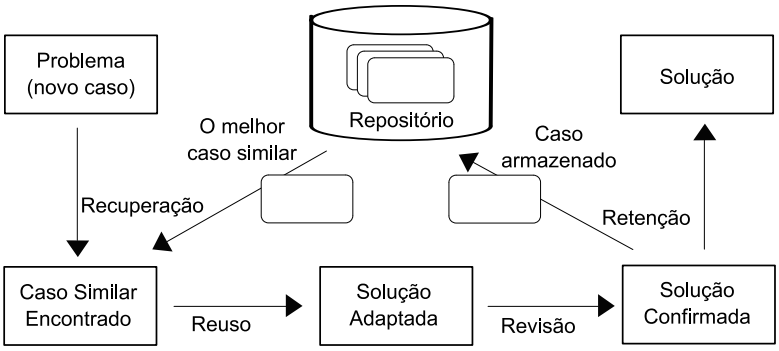

Figura 2.10: Ciclo de solução de problemas em um sistema de RBC

- Informações sobre o novo caso: o usuário ou o operador irá informar um conjunto de atributos com informações sobre o problema, e o sistema representará esse novo caso no padrão de consulta adotado;

- Recuperação e avaliação: um conjunto de casos semelhantes é selecionado na base de casos. Esses casos são comparados apenas em relação aos atributos mais relevantes e trazidos para a memória principal para uma comparação mais completa. Para a medida de similaridade (utilizada na recuperação dos casos semelhantes) pode-se utilizar funções discretas, Distância Euclidiana, *Lógica Fuzzy*[8], entre outros;

- Logo em seguida, cada um dos casos candidatos é comparado ao novo problema, utilizando-se um algoritmo de vizinhança, que mede a diferença de valores de cada um dos atributos de ambos os casos. O caso que obtiver a menor soma das diferenças é escolhido como o melhor caso, ou seja, o caso recuperado que for mais próximo ao novo caso é o eleito;

- A solução associada ao caso é avaliada e, se necessário, a solução é adaptada para compensar as diferenças entre os casos;

- O novo caso assim é gerado e também armazenado na base de casos com a solução adotada, quando então diz-se que o sistema aprendeu.

[8]Ao contrário da lógica clássica, que reconhece apenas o falso ou verdadeiro, a *Lógica Fuzzy* reconhece diversos valores, assegurando que a verdade é uma questão de ponto de vista ou de graduação. Isso permite a manipulação informações imprecisas, como conceitos de pouco, alto, bom, muito quente, frio etc [MBCJ05].

Aplicabilidade do Algoritmo

O RBC pode ser utilizado em vários contextos a exemplo dos que se enquadrem nos tópicos que seguem:

- A essência do funcionamento do RBC está na aplicação de uma experiência antiga como solução de uma nova experiência. Logo, deduz-se que exista um conhecimento mínimo sobre o domínio do problema, mapeado através de exemplos.
- Deve existir um senso de métrica em cada atributo para calcular a semelhança. Ex.: distância Euclidiana, função fuzzy, linear ou discreta.
- Dois casos semelhantes devem ocupar realmente lugares próximos no espaço do domínio, ou seja: considere $A = \{a_1, a_2, a_3, ..., a_n\}$ e $B = \{b_1, b_2, b_3, ..., b_n\}$ dois casos de uma base de casos, com seus respectivos conjuntos de atributos. Dizemos que os casos são tão semelhantes quanto $\Sigma \mid a_i - b_i \mid$ é próximo de 0.

Observe também que, para existir aprendizagem, é necessário que o domínio possua um número de estados não totalmente representados na base de casos. De outra forma, o sistema não poderá adicionar novas experiências em sua base.

2.2.4 Aprendizagem por Reforço

A aprendizagem por reforço é utilizada para aprender sobre a relação estado-ação em um domínio. Assim, seu objetivo é assimilar a melhor ação aplicada a um estado inicial que maximiza as expectativas para chegar a um estado final/objetivo. Esta técnica é adotada quando não existe uma definição ou estudo bem formado sobre as ações a serem tomadas conforme cada estado do ambiente. O processo de aprendizagem consiste no ajuste contínuo do reforço, em forma de recompensa ou punição, que uma ação recebe conforme mensuração de sua eficácia. A aprendizagem por reforço é garantida segundo o modelo de decisão de Markov [Lit94; BRC04; Tes04].

Processos Decisórios de Markov

Conforme [MR03], “um ambiente satisfaz a propriedade de Markov se o seu estado resume o passado de forma compacta e pode-se predizer qual o próximo estado e recompensa es-

perados, dados o estado e ação atuais". A técnica de Aprendizagem por Reforço (AR) que satisfaz a propriedade de Markov é chamada de Processo Decisório de Markov (MDP - *Markov Decision Process*), e é definido por um conjunto (S, A, P, R), onde temos:

- S um conjunto finito de estados do sistema;
- A: conjunto finito de ações;
- $P : S \times A$: função de transição que mapeia os pares estado-ação em uma distribuição de probabilidades sobre o conjunto de estados;
- $R : S \times A$: é a função de retorno ou recompensa pela escolha de uma determinada ação $a \in A$ no estado $s \in S$.

Assim, o estado s e a ação a atuais informam sobre o próximo estado s' de acordo com a probabilidade $P(s'|s, a)$ e a recompensa associada $r(s, a)$. Intuitivamente, um processo é Markoviano se e somente se o estado de transições depende apenas do estado corrente do sistema e é independente de todos estados anteriores [NJ05].

Q-learning

O algoritmo Q-learning [WD92] é um MDP, o que implica num algoritmo que necessita de constante atualização através de recompensas esperadas, $Q(s, a)$. A cada interação com o ambiente, os valores de Q são atualizados de acordo com a Equação 2.1.

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r + \gamma\, \epsilon Q(s')\right], \tag{2.1}$$

onde γ é o valor de desconto utilizado para garantir o número finito para valores de Q e α é a constante de aprendizagem, onde $0 \leq \gamma < 1$ e $0 < \alpha \leq 1$. Após a execução da ação α, o algoritmo saiu do estado s e está no estado s', onde deve receber uma recompensa r conforme o resultado obtido. Em s' é feita uma busca pela ação a' que apresente o maior valor de retorno esperado, isso é representado por $\epsilon Q(s') = max_{a'} Q(s', a')$.

Caso a ação α' seja escolhida como a próxima a ser executada, existe uma probabilidade maior do algoritmo cair em *máximos locais* [9]. Segundo [FR99], para evitar este problema,

[9] Na aprendizagem por reforço, usa-se esse termo para apontar uma estabilização na escolha da ação a aplicada ao estado s. Matematicamente, diz-se estar em um *máximo local* quando para todo x próximo ao ponto $(max, f(max))$, tem-se $f(x) < f(max)$.

deve-se escolher uma ação que não é necessariamente igual a α', onde uma boa estratégia poderia ser escolher, em 70% dos casos, a ação que retorne o valor máximo e nos outros 30% escolher α' de forma aleatória.

Aplicabilidade do Algoritmo

Mapeando o algoritmo Q-learning para o domínio do comércio eletrônico, o conjunto (S, A, P, R) pode ser definido como:

- S é um conjunto composto por pares de propostas e contra-propostas, representando o estado da negociação;
- A: conjunto finito de estratégias;
- $P : S \times A \rightarrow \Pi(s)$: função de transição de estado representada pelo valor probabilidade, mostrando as melhores estratégias para serem escolhidas;
- $R : S \times A$: é descrito como um valor utilidade, definido pela similaridade dos atributos, mapeados como uma função de recompensa.

Assim, o Algoritmo pode ser escrito como na Tabela 2.1.

Inicialize $Q(s, a)$
Repita:
Visite o estado S
Selecione uma estratégia $a \in A$
Execute a ação a
Receber o reforço $r(s, a)$ e observar o próximo estado s'.
Atualizar os valores de $Q_t(s, a)$ de acordo com:
$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma\epsilon Q(s') - Q(s, a)\right]$
Atualizar o estado $s \leftarrow s'$
Até que algum critério de parada seja alcançado,

Tabela 2.1: Q-learning aplicado ao comércio eletrônico

2.2.5 Aprendizagem na Negociação

No inicio desta seção foi discutida a importância dos sistemas dinâmicos e o fato de ser desejável agregar aprendizagem e inteligência aos mesmos, pois o mundo se modifica constantemente e uma boa ação hoje pode não ser considerada amanhã. Considerando uma negociação como um jogo, envolvendo propostas e contra-propostas [Pau01; Fil04], a aprendizagem pode ser aplicada tanto na aquisição de conhecimento sobre a estratégia/ação a ser tomada quanto na aquisição de dados relevantes na tomada de decisão. Na negociação automatizada, os jogadores, que desempenham papéis de compradores e vendedores, buscam alcançar uma situação de acordo sobre os objetos, a qual é alvo da negociação [Fil04; PRR01]. A análise de fatores externos ao negociador, ou seja, tendências e valores de mercado, taxas do governo e outros, são importantes no processo de negociação no sentido da adoção de um caráter concessivo ou ganancioso de forma a otimizar o lucro e não ter prejuízos na negociação [FWJ04].

No dia-a-dia, na prática natural do comércio, também pode-se observar o que é chamado intuitivamente de *experiência do negociador*. A tendência natural humana é dar menor importância para negociações que tenham características semelhantes a outras de uma negociação anterior onde não se obteve o sucesso esperado. Como mostrado neste capítulo, a IA se propõe a resolver situações dessa natureza com a abordagem Raciocínio Baseado em Casos (RBC), onde uma base de dados (casos) é consultada de forma a se propor caminhos para negociação ou propostas que tenham grandes chances de sucesso. A tarefa para a escolha de boas estratégias também pode ser investigada através de uma solução baseada em Redes Neurais, onde o maior problema seria encontrar uma boa base de dados para o treinamento.

Alguns pontos onde a aprendizagem deve contribuir já podem ser identificados, são eles:

- Seleção da melhor estratégia de negociação: onde deve-se escolher uma ação, considerando o estado atual da negociação. Para essa seleção, pode-se utilizar qualquer um dos modelos descritos anteriormente (Redes Neurais, RBC, Aprendizagem por Reforço), sendo que apenas a Aprendizagem por Reforço, não necessita de um histórico de casos prévio.

- Determinação do perfil do usuário: item essencial quando se deseja tratar o cliente de forma individualizada. Uma Rede Neural SOM pode fazer a classificação dos usuários

em perfis.

- Elaboração de propostas mais lucrativas e ao mesmo tempo interessantes para o cliente: considerando tanto dados diretamente ligados à empresa, como também fatores externos à negociação (taxas de frete, variação do dólar, previsão do tempo etc).

Capítulo 3

Trabalhos Relacionados

Na tentativa de aproximar o modelo eletrônico ao real, alguns trabalhos têm contribuído em vários aspectos na negociação automatizada, por exemplo, na aprendizagem, tomada de decisão, arquitetura/modelo de ambientes de negociação. Como o presente trabalho envolve a construção de um ambiente para negociação e tal ambiente oferece também mecanismos para aprendizagem dos negociadores, este capítulo está dividido em trabalhos relacionados ao ambiente de negociação e relacionados à aprendizagem e modelo de decisão.

3.1 Ambiente de Negociação

A questão da negociação automatizada, de um modo geral, tem sido abordada na literatura, já fornecendo uma significativa evolução de resultados de pesquisa. Em particular, podem-se destacar como um dos aspectos dessa evolução, as seguintes classes de Sistemas para o comércio eletrônico com suporte à negociação automatizada:

a) sistemas com apenas um único atributo negociável pelos agentes: o preço;

b) sistemas envolvendo mais de um atributo e utilizando heurísticas para atingir melhores resultados.

Com as características do item *b)* acima surgiram sistemas para o comércio eletrônico com comportamentos[1] que os aproximam de vendedores reais, oferecendo também produtos

[1]Refere-se às propriedades encontradas em um agente da IA, descritas na Subseção 2.1.2

alternativos e impondo propostas finais (*ultimatos*). Em seguida, evoluíram no suporte à tomada de decisão, e no tratamento individualizado para cada cliente, considerando suas características pessoais.

Durante a exposição dos trabalhos relacionados a seguir, os mesmos estarão sendo enquadrados nas características acima. Os dois últimos trabalhos revisados ([Pau01; Jun02]) estarão recebendo uma atenção especial, pois foram tomados como ponto de partida e inspiração para esta dissertação.

3.1.1 O Sistema Kasbah

Com as características de *a)* foi concebido o sistema multiagente Kasbah [CM96], que é um sistema multiagente no qual os usuários publicam suas ofertas e demandas através de agentes, e estes negociam apenas o preço do produto. Neste sistema, agentes fazem o papel de compradores ou vendedores que anunciam suas demandas ou ofertas. O usuário precisa também definir: a estratégia de negociação, que é traduzida por uma função; os limites máximo e mínimo do preço; e o tempo máximo da negociação.

O modelo do Kasbah é simples, existindo uma troca de propostas que são geradas sob influência do tempo, e existe uma função de avaliação que calcula a utilidade de cada proposta gerada ou recebida. Assim, as ações para um negociador que recebe uma proposta, são:

- sair da negociação: quando o tempo estipulado para o término da negociação chega ao fim;
- aceitar a proposta ou rejeitar/enviar contra-proposta: o negociador sempre tenta gerar uma contraproposta com uma utilidade melhor, e, quando não consegue, aceita a proposta recebida.

O sistema Kasbah teve sua importância como um dos pioneiros a trabalhar com comércio eletrônico, porém não representava bem uma negociação real, que é bem mais complexa. O único atributo negociável é o preço e a elaboração de propostas só depende do tempo.

3.1.2 Wreper

O sistema Wreper [Far00], que é uma extensão do Kasbah, propõe um modelo mais elaborado de negociação, e pode ser retratado pelo item *b)* na linha de evolução dos sistemas. São considerados na negociação mais de um atributo e cada proposta é avaliada pela função U_p^i que informa o ganho do agente, avaliada no intervalo [0,1], onde: i representa o agente negociador e p uma proposta na negociação. A Equação 3.1 mostra a função U_p^i em detalhes:

$$U_p^i\left(P_t^i = \{a_1 = v_1, a_2 = v_2, ..., a_n = v_n\}\right) = \sum_{j=1}^{n} wa_j^i \; V_{a_j}^i\left(v_j\right) \tag{3.1}$$

em que: P_t^i é uma proposta de n atributos a_i com respectivos valores v_i, tendo sido enviada pelo agente i no tempo t; w é o peso que traduz a importância do atributo na negociação; $V_{a_j}^i$ é a função que calcula a utilidade individual de cada atributo; e cada atributo a considerado no acordo está associado aos seus respectivos intervalos de valores aceitáveis e intervalo de valores negociáveis [Pau01]. Faratin propõe também estratégias de negociação heurísticas que permitem ao agente tomar suas decisões com base em conhecimentos do ambiente, assim introduzindo o conceito de estratégia de negociação. Por exemplo, o comportamento concessivo do oponente, o tempo gasto para negociar ou os recursos utilizados na negociação, são levados em consideração na estratégia [FSJ98; Far00]. As ações permitidas para cada agente são as mesmas do Kasbah. A inovação de Faratin é com o conceito de estratégia de negociação. Uma estratégia pode ser definida como uma combinação de táticas de geração de propostas. As táticas de geração de propostas para Faratin podem ser de três diferentes famílias:

1. Dependentes do tempo, que levam em consideração o tempo gasto na negociação para adquirir comportamento mais concessivo ou ambicioso.

2. Dependentes do recurso, que analisam as limitações existentes de recursos (número de oponentes existentes, quantidade de dinheiro, etc). Funcionam como uma tática dependente do tempo sendo que o recurso considerado não é o tempo.

3. Imitativas, nas quais o agente negociador, depois que obter uma quantidade razoável de informações sobre o seu oponente, pode imitá-lo. Ou seja, caso o oponente ceda, o agente cede, e caso o oponente não ceda, ele também não cede [Pau01].

As duas dificuldades que podem ser apontadas neste trabalho são apresentadas a seguir. (1) Rigidez da estrutura das preferências para negociação. Pode ser necessário fazer ajustes

nas preferências de negociação definidas antecipadamente. Os intervalos definidos podem ser totalmente incompatíveis ou muito grandes para serem utilizados nas propostas. Não existe nenhum meio ou estratégia de alterar esses espaços de acordo dinamicamente. (2) A tomada de decisão não considera informações extras sobre o oponente. A personalização de uma negociação poderia ser conseguida se o modelo considerasse informações importantes como um perfil do usuário ou histórico de negociações. Com base nessas informações do usuário, pode-se conseguir melhores acordos, oferecer condições personalizadas, e assim aumentar a satisfação dos mesmos.

3.1.3 Framework por Fatima

Fatima et al. [FWJ04] propõem um ambiente onde os agentes negociadores criam suas propostas utilizando um modelo probabilístico associado à uma função de decisão (Equação 3.2) para as estratégias. Neste trabalho pode-se negociar com mais de uma mercadoria ou serviço ao mesmo tempo, porém apenas o atributo *preço* é considerado nos testes e a aprendizagem vem da estatística em dados históricos. Assim, a função de decisão A^s para o caso de duas negociações simultâneas (porém, mesmo vendedor e mesmo comprador), sobre os itens X e Y, está representada na Equação 3.2, onde: a notação $X^t_{b\to s}$ representa o *preço*, sobre X, oferecido pelo agente b para o agente s no tempo t; e a função utilidade U^s é a mesma definida na Equação 3.1.

$$A^s(t, X^t_{b\to s}, Y^t_{b\to s}) = \begin{cases} Sair & se\ t > T^s, \\ Aceitar & se\ U^s(X^t_{b\to s}) \geq U^s(X^{t'}_{s\to b}), \\ Aceitar & se\ U^s(Y^t_{b\to s}) \geq U^s(Y^{t'}_{s\to b}), \\ ofertar X^{t'}_{s\to b} & se X^t_{b\to s}\ não\ foi\ aceito, \\ ofertar Y^{t'}_{s\to b} & se Y^t_{b\to s}\ não\ foi\ aceito. \end{cases} \tag{3.2}$$

O seu trabalho apresenta um modelo de negociação bem detalhado, com um protocolo semelhante ao de Faratin [Far00; FSJ98], que considera o tempo na negociação. Da mesma forma, as estratégias vêm da idéia de *táticas dependentes do tempo* de Faratim, que podem ser ajustadas para gerar propostas em um comportamento mais concessivo, ganancioso ou intermediário. A tentativa de encontrar a melhor estratégia para negociação vem de uma exaustiva análise da função utilidade [FSJ02; KFSBY03] em simulações e da conseqüente estatística associada.

3.1.4 Framework por Bartolini

No trabalho de Bartolini et al. [BPJ04; BPJ02] está definido um framework para negociação automatizada, utilizando o JADE[2] [JAD], com o propósito de especificar e gerenciar as regras que regem uma negociação. Para isto, a tecnologia Jess[3] [FH] foi associada independentemente em cada agente, que utilizará um conjunto de regras pré-definidas para admitir novos participantes, validar propostas, para o protocolo da aplicação, atualizar o *status*(ou estado) dos participantes, formar de propostas e acordos, e gerenciar o ciclo de negociação.

Quanto à tomada de decisão e aprendizagem, este trabalho utiliza regras no processo de negociação, que são inseridas no modelo por especialistas externos e não por algum processo de aprendizagem. Assim, este trabalho é classificado como do tipo *b)*, possuindo uma estrutura bastante estável pela utilização integrada de JADE e Jess.

3.1.5 Modelo Proposto por Gustavo de Paula

O trabalho de Gustavo de Paula e outros [PRR01] propõe um modelo matemático para representar uma negociação bilateral entre agentes inteligentes. Basicamente, esse modelo contribui no cenário de negociação automatizada em comércio eletrônico em dois aspectos: na modelagem de sua proposta e em mecanismos de tomada de decisão. Ainda nesta subseção será mostrado o modelo de representação do produto negociável, o modelo de avaliação de propostas, o mecanismo de tomada de decisão, e um apanhado sobre o modelo de tomada de decisão.

Modelagem do Produto

Para que se possa projetar um agente negociador, a primeira preocupação é se determinar qual o modelo de produto que este agente deverá entender durante o processo de negociação. Dessa forma, [PRR01] propõe que um produto seja especificado sob duas óticas: Atributos de produto (C), que são os atributos que representam as características que especificam um determinado produto; e os atributos de acordo (A), que são aqueles atributos que representam

[2] *Java Agent Development Framework* - Arcabouço para desenvolvimento de sistemas multiagente que segue as normas internacionais da FIPA [FIP06]. Mais detalhes no Apêndice B.

[3] *Java Expert System Shell* - Máquina de inferência para regras de produção, que também possui um ambiente editor. Foi desenvolvido para a linguagem Java.

sob quais condições o produto será barganhado. Assim, o objeto da negociação é definido por $O = A' \bigcup C'$, conforme está representando na Figura 3.1.

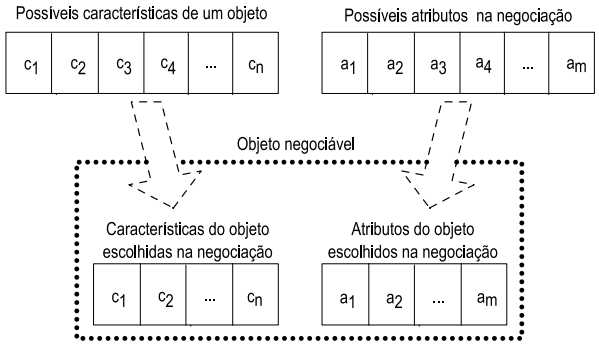

Figura 3.1: Modelagem do Objeto da Negociação

Tomada de Decisão

O modelo de tomada de decisão proposto por de Paula [PRR01] permite ao agente, ao receber uma proposta, tomar uma das seguintes decisões:

- *Enviar Ultimato:* A decisão de enviar um ultimato depende do tempo transcorrido atual da negociação, representado por *t*. Se *t* superar o tempo máximo de negociação, que é uma medida particular do agente, então o agente tomará a decisão de enviar um ultimato, como está representado na Equação 3.3.

$$EnviarUltimato\,(t, t_{max}) = \begin{cases} verdadeiro,\ se\ (t > t_{max}) \wedge ultima_acao \neq ultimato \\ falso,\ caso\ contrário \end{cases} \tag{3.3}$$

- *Enviar Produto Alternativo:* Depende da taxa de convergência das propostas enviadas. Se ela está acima de um limiar, um produto alternativo deve ser enviado. A Equação 3.4 representa formalmente o modo como é tomada essa decisão.

$$EnviarAlternativa\left(t, \overrightarrow{P}^{v}_{t-1}, \overrightarrow{P}^{c}_{t}\right) = \begin{cases} verdadeiro,\ se\ \left[E_c(P^{v}_{t-j}) - E_c(P^{v}_{t-1})\right]/j > minTax \\ \wedge ultima_acao \neq ultimato \\ falso,\ caso\ contrário \end{cases} \tag{3.4}$$

em que t representa a rodada de negociação atual; P^{v}_{t-j} representa um vetor com as propostas trocadas pelo agente v até a rodada $t-1$, e P^{c}_{t} um vetor com as propostas feitas pelo agente c até a rodada t.

- *Enviar Aceitação:* A decisão de enviar uma aceitação depende do *status*(ou estado) da proposta recebida: caso seja um *status* informando um ultimato, basta que a avaliação da proposta recebida esteja numa escala aceitável e que a última ação do agente não tenha sido enviar um ultimato; com o *status* de proposta normal, é avaliado se a proposta recebida possui uma utilidade maior que a próxima proposta que o agente deverá enviar. As Equações 3.5 e 3.6 representam matematicamente as funções que expressam as situações de proposta normal e proposta de ultimato.

$$EnviarAceitacao(P^v_{t-1}, P^c_t) = \begin{cases} verdadeiro,\ se\ \left[E_c(P^v_{t-1}) \geq E_c(P^c_t)\right] \wedge ultima_acao \neq ultimato \\ falso,\ caso\ contrário \end{cases} \tag{3.5}$$

$$EnviarAceitacaoUltimato(t, P^v_{t-1}) = \begin{cases} verdadeiro,\ se\ \left[E_c(P^v_{t-i}) > -\infty\right] \wedge ultima_acao\ ultimato \\ falso,\ caso\ contrário \end{cases} \tag{3.6}$$

- *Enviando contra-proposta:* A ação de enviar uma contra-proposta, acontece se as ações de *Enviar Ultimato, Enviar Produto Alternativo* e *Enviar Aceitação* não foram adequadas e ainda não tiver recebido um ultimato. A equação 3.7 representa formalmente essa situação.

$$EnviarContraProposta(t, \overrightarrow{P}^v_{t-1}, \overrightarrow{P}^c_t) = \begin{cases} verdadeiro,\ se \neg \left(EnviarAlternativa\left(t, \overrightarrow{P}^v_{t-1}, \overrightarrow{P}^c_t\right)\right) \wedge \\ \neg\left(EnviarUltimato(t, t_{max})\right) \wedge \\ \neg\left(EnviarAceitacao(P^v_{t-1}, P^c_t)\right) \wedge \\ ultima_acao \neq ultimato \wedge \\ falso,\ caso\ contrário \end{cases} \tag{3.7}$$

- *Sair da negociação:* A última ação possível é a de sair da negociação, que acontece se o agente já tiver enviado um ultimato ou estiver recebendo um ultimato. A Equação 3.8 representa formalmente o mecanismo de tomada de decisão.

$$EnviarSair(t, P^v_{t-1}) = \begin{cases} verdadeiro,\ se \neg \left(EnviarAceitacaoUltimato(t, P^v_{t-1})\right) \\ \wedge ultima_acao = ultimato \\ falso,\ caso\ contrário \end{cases} \tag{3.8}$$

Comentários

O Modelo proposto por [PRR01] é bastante amadurecido e já aproxima bastante a idéia de se automatizar negociações no cenário do comércio eletrônico. Por exemplo, já é possível atribuir a agentes negociadores a capacidade de tomar decisões de forma autônoma. Além

disso, a contribuição com o aumento da capacidade de barganha, para o negociador, mostra-se um grande avanço.

Por outro lado, alguns aspectos cognitivos muito importantes, relacionados à atividade de negociar, ainda não foram contemplados. O primeiro é a capacidade de *auto-reflexão*: no final de cada negociação, é importante que um negociador estime o quanto o seu modo de interação com o seu oponente foi eficaz, avaliando os pontos positivos e negativos de sua interação. Outro aspecto importante é a *empatia*, onde se deve tentar entender as reais necessidades ou preferências de seu oponente, com o objetivo de diminuir as divergências existentes entre eles. Por último, adicionar ao agente a capacidade de considerar várias negociações em paralelo, conseguindo dessa forma melhorar suas ofertas como conseqüência do barateamento do frete, custos com os fornecedores, planejamento de compras etc.

Assim, com o objetivo de aprimorar tais características nos agentes propostos por de Paula [PRR01], algumas questões foram levantadas com respeito ao mecanismo de inferência adotado naquele trabalho:

- *Enviar Produto Alternativo:* A decisão de enviar um produto alternativo é um aspecto que poderia ter uma conotação mais individualizada. O que acontece é que a decisão de enviar um produto alternativo não leva em consideração aspectos do oponente. Em outras palavras, não existe um mecanismo para sugerir um outro produto na negociação atual, segundo algum critério que implique maiores chances da aceitação da proposta.

  Outro aspecto a ser considerado é o fato que a substituição do produto em questão na negociação está restrita a apenas produtos do mesmo tipo, mas, com especificações diferentes. Em algumas negociações, poderia ser interessante substituir um produto que está sendo negociado por um outro que atendesse melhor aos interesses dos negociadores.

- *Avaliação de Propostas:* O modo de avaliação das propostas tem um problema sutil, que é a obrigação de vincular uma função de avaliação a um, e somente um, atributo da negociação. Isso pode não ser interessante, pois, um critério de avaliação pode estar vinculado a mais de um atributo, e um mesmo atributo pode estar vinculado a mais de um critério de avaliação.

- *Negociações Paralelas:* Uma possibilidade interessante que o modelo por de Paula não oferece é a possibilidade do agente negociador poder negociar com vários oponentes ao mesmo tempo, possibilitando dessa forma, ter uma visão mais realista das expectativas do mercado em relação ao produto em questão. Fazendo, dessa forma, com que as propostas de negociação sejam conduzidas segundo as leis de oferta e procura.

- *Aprendizagem de Máquina:* O modelo de tomada de decisão, baseado em funções pré-ajustadas poderá ser substituído por um modelo dinâmico onde as estratégias e decisões tomadas durante a negociação se ajustem adequadamente, por exemplo, através da utilização da Inteligência Artificial. Com isso a tomada de decisão acompanharia o dinamismo do ambiente de negociação, se adaptando conforme suas tendências.

3.1.6 Modelo Proposto por Pinho Júnior

O modelo proposto por Pinho Júnior [Jun02] é uma extensão do modelo por de Paula [PRR01], que adicionalmente capacita aos seus agentes negociadores negociar bilateralmente de forma seqüencial com vários agentes negociadores oponentes.

Para que isso possa acontecer, foi adicionado um conceito novo em relação aos possíveis estados da negociação, que é o *compromisso de acordo*. Com esse novo *status*, os agentes negociadores que estejam negociando, têm a possibilidade de manter o acordo encontrado sob o estado *compromissado*, antes de realmente efetivar o acordo. O *status* compromissado ficará por um período de tempo definido entre eles, e é durante esse período de tempo que os agentes negociadores tentarão barganhar com outros agentes negociadores oponentes, tentando obter um acordo com uma utilidade maior.

Compromisso de Compra

Negociação bilateral seqüencial (NBS) não é o mesmo que negociação multilateral, aqui é considerada uma negociação entre duas entidades por vez e o vendedor pode "dar sua palavra de compromisso" por um tempo determinado. Formalmente, uma negociação em estado *compromissado* n_c, em uma série N de negociações bilaterais, pode ser identificada em uma posição anterior à da negociação atual n_a: $N = \{n_1, ..., n_c, ..., n_a, ..., n_n\}$. Inicialmente,

[Jun02] propõe a seguinte expressão matemática, mostrada na Equação 3.9, que representa a forma como o agente negociador toma a decisão de efetivar ou não um compromisso. Essa decisão é tomada pouco antes do final do período de compromisso, caso o mesmo agente não tenha obtido um acordo melhor com outros negociadores. O Termo *mimNeg* representa o número mínimo de negociações que o agente negociador deverá enfrentar antes de desistir completamente de negociar (o modo como esse valor é obtido será esclarecido posteriormente). Por sua vez, *c* representa o índice da negociação que o agente negociador esta enfrentando atualmente.

$$fabandon(c, minNeg) = \begin{cases} Se\ (c < minNeg),\ retorna\ falso \\ caso\ contrário,\ retorna\ verdadeiro \end{cases} \tag{3.9}$$

O mecanismo de avaliação do valor mínimo de negociações, que o agente negociador deverá passar antes de abandonar o processo seqüencial de negociação, representado pelo termo *minNeg*, dependerá de três condições:

1. A avaliação atual do acordo em compromisso, que é representado por V_Ω.

2. O valor mínimo de utilidade ótima, ou seja, o valor que indica o quanto o agente negociador está satisfeito com a negociação, sem precisar negociar com outros agentes. Este valor é representado por ***commThresold***, onde $commThresold \in [0, 1]$.

3. E por último, a personalidade do agente em relação ao quanto este é favorável a riscos nas negociações, ou o quanto este agente tentará forçar a tentar uma negociação com um valor de utilidade mais expressivo. Essa personalidade de quanto o agente é favorável a riscos é expresso através do termo ***abandonVenture***, onde $0 < abandonVenture < 1$ representa um agente com personalidade mais precavida em relação aos riscos e $abandonVenture > 1$ representa um agente com personalidade mais audaciosa na busca por melhores resultados.

A Equação 3.10 representa a fórmula ***fminNeg*** que define o cálculo de ***minNeg***.

$$fminNeg(V_{\hat{U}}) = \begin{cases} Se\ (commThreshold = 0) \vee \left(\frac{V_\Omega}{commThreshold} > 1\right),\ retorna\ zero \\ caso\ contrário,\ retorna\ totalNeg\left(1 - \frac{V_\Omega}{commThreshold}\right)^{\frac{1}{abandonVenture}} \end{cases} \tag{3.10}$$

Tomada de Decisão

O agente negociador possui mecanismos para a tomada de decisão, principalmente para avaliar se uma proposta recebida por seu oponente, deve ser aceita ou não. Essa avaliação deve ser feita considerando o contexto atual do negociador, e nesse caso existem duas situações: a primeira é quando o agente ainda não tem um acordo em compromisso, onde se deve avaliar conforme os valores de cada atributo da proposta; e a outra é quando o negociador já possui um acordo em compromisso, onde também usa um mecanismo de comparação.

Primeiramente, [Jun02] define um mecanismo para representar a expectativa do agente em melhorar o seu acordo, dado que o agente esteja com um acordo em compromisso. Essa expectativa é representada pela letra grega Φ, e é calculada segundo a Equação 3.11.

$$\Phi = V_\Omega \; X \; (1 + \lambda) \tag{3.11}$$

Esta expressão expressão leva em consideração dois fatores:

1. o valor da avaliação do acordo em compromisso atual, representado por V_Ω;
2. e de um fator, representado pela letra grega λ, que indica o percentual de acréscimo de ganho para a negociação atual, chamado de ganho esperado.

O ganho esperado, representado pela expressão matemática 3.12, por sua vez, é calculado através de três fatores, descritos abaixo:

$$fgain(V_\Omega) = refGain \; \times \; (1 - V_\Omega)^{\frac{1}{bargainVenture}} \tag{3.12}$$

- O valor da avaliação do acordo em compromisso atual, representado por V_Ω.
- Um fator de referência para o calculo do valor esperado, chamado de ganho de referência, representado pelo termo *refGain*.
- É um índice que representa o quanto o agente espera receber novas e melhores propostas. Este índice está representado no termo *bargainVenture*, onde ($0 < bargainVenture < 1$) indica um agente que possui uma personalidade mais conciliadora em relação as negociações, e ($bargainVenture > 1$) torna o agente mais agressivo em relação as suas negociações.

Dessa forma, o agente negociador poderá calcular a aceitação das propostas através da Equação 3.13, dado que este agente possua um acordo em compromisso. Para os casos onde a proposta recebida for rejeitada, é verificado se a próxima ação é o envio de um ultimato ou de outra proposta.

$$bargainInterpretOffer(P_r = \begin{cases} Se\ timeToBargain() \wedge (V_{P_r} \geq \Phi)\ ,\ aceita(P_r) \\ Se\ \neg timeToBargain() \wedge (V_{P_r} < \Phi)\ ,\ ultimato(P_{barganha}) \\ caso\ contrário,\ oferta\ (P_{táticas}) \end{cases} \tag{3.13}$$

1. Se for momento de barganhar (decisão inferida pela função *timeToBargain()*, que será discutida ainda nesta subseção), e a avaliação da proposta recebida for maior ou igual a avaliação esperada, então a proposta será aceita.

2. Se ainda não estiver no momento de barganhar ou se a proposta recebida (V_{P_r}) tiver um ganho menor que o ganho absoluto esperado (ϕ), então um ultimato com a proposta de barganha será enviada.

3. E em outro caso, uma proposta de oferta baseada em funções definidas no modelo por Gustavo de Paula (Subseção 3.1.5), será enviada.

Caso o agente negociador ainda não tenha um acordo em compromisso, então seu mecanismo de tomada de decisão será semelhante ao definido no modelo por de Paula, expresso na Equação 3.14.

$$notBargainInterpretOffer(P_r) = \begin{cases} Se\ \left(V_{P_r} \geq V_{P_{táticas}}\right),\ aceita(P_r) \\ Se\ (t \geq tmax) \wedge (V_{P_r} < P_{táticas})\ ,\ ultimato(P_{tática}) \\ caso\ contrário,\ oferta(P_{táticas}) \end{cases} \tag{3.14}$$

1. Caso a proposta recebida tenha uma utilidade ou ganho maior que a próxima proposta gerada para ser enviada, então a proposta é aceita.

2. Caso o tempo da negociação tenha finalizado e avaliação da proposta recebida seja menor que a avaliação da próxima proposta a ser enviada, então será enviado um ultimato com a próxima proposta a ser enviada.

3. Em outro caso, a próxima proposta a ser enviada será ofertada.

Por outro lado, caso a proposta recebida seja na forma de um ultimato, as funções de avaliação de tomada de decisão não serão as representas pelas Equações 3.13 e 3.14, mas sim,

novas funções específicas para propostas do tipo ultimato. Estas funções são representadas pelas Equações 3.15 e 3.16.

Em propostas recebidas como ultimato, a única possibilidade do agente negociador que a recebeu é a de aceitar ou rejeitar a proposta. Dado que o agente já possui um acordo em compromisso, a expressão 3.15 representa o seu mecanismo de tomada de decisão:

$$bargainInterpretUltimatum(P_r) = \begin{cases} Se\ V_{P_r} \geq \Phi,\ \ aceita(P_r) \\ caso\ contrário,\ \ rejeita(P_r) \end{cases} \tag{3.15}$$

- Caso a proposta recebida como ultimato tenha uma utilidade maior que o ganho absoluto esperado (ϕ), a proposta será aceita.

- Caso contrário, a proposta será rejeitada.

Se no momento que o agente recebe uma proposta no formato ultimato, e este agente ainda não tiver nenhum acordo em compromisso, a expressão 3.16 será utilizada como mecanismo de tomada de decisão, que de forma simples, tomará uma seguinte decisões:

$$notBargainInterpretUltimatum(P_r) = \begin{cases} Se\ V_{P_r} \geq 0,\ aceita(P_r) \\ caso\ contrário,\ rejeita(P_r) \end{cases} \tag{3.16}$$

1. Se a proposta recebida possui uma utilidade maior que zero, então, a proposta será aceita. Uma utilidade maior que zero representa que o valor recebido esta dentro do intervalo de valores aceitáveis pelo agente negociador, conforme será visto adiante.

2. Caso contrário, o agente rejeitará a proposta.

O mecanismo de avaliação das propostas geradas ou recebidas por um agente negociador é bastante simples, tendo em vista que o modelo dos agentes inclui apenas o preço na avaliação. Este modelo está representado na Equação 3.17. Vale ressaltar que β é um valor que representa o modo de como o agente negociador está avaliando suas propostas, onde: se $\beta > 1$, o agente avaliará propostas de forma mais complacente e vice-versa.

$$v(x) = \begin{cases} Se\ min \leq x \leq max,\ [\frac{(max-x)}{(max-min)}]^{\frac{1}{\beta}} \\ Se\ x > max, -\infty \\ Sex < min, 1 \end{cases} \tag{3.17}$$

De forma paralela, segue o mecanismo de geração de proposta de ultimato, proposto por [Jun02] e representado pela Equação 3.18.

$$x = max - \left(\Phi^{\beta} X(max - min)\right) \tag{3.18}$$

E por último, a definição da função *timeToBargain()* que determina qual o momento em que um agente deve enviar uma proposta de barganha ou ultimato. Como mostra a Equação 3.19, as decisões são inferidas da seguinte forma:

1. Quando a utilidade da proposta gerada a ser enviada for menor que a utilidade definida pelo termo ganho esperado (ϕ) ou o tempo da negociação esteja acabando, será momento de enviar um ultimato;

2. Caso contrário, não será momento de enviar um ultimato.

$$timeToBargain() = \begin{cases} Se(V_{P_{táticas}} \leq \Phi) \vee (t \geq max),\ verdadeiro \\ caso\ contrário,\ falso \end{cases} \tag{3.19}$$

Comentários

Na prática, uma pesquisa de compra é feita de forma seqüencial, onde são comparados os preços, loja por loja. Agora, com o modelo trazido por Pinho Júnior [Jun02], tem-se esse aspecto traduzido para o enriquecimento do comércio eletrônico.

Resgatanto os comentários sobre o modelo por Gustavo de Paula (Subseção 3.1.5), existe a necessidade de aspectos cognitivos para suportar a barganha, a tomada de decisão, mudanças de estratégias, e outros relacionados ao andamento da negociação. Em outras palavras, ainda existe uma negociação baseada em modelos matemáticos e probabilísticos, em que o negociador não tem um modelo para dinâmico para tomada de decisões, assim como não possui um modelo para aprendizagem e uso da sua experiência em futuras negociações.

3.2 Aprendizagem e Modelo de Decisão

Quanto às técnicas utilizadas em sistemas inteligentes para o aprendizado e tomada de decisão, abordadas no Capítulo 2, estão relacionados a seguir alguns trabalhos e pesquisas considerados mais afins com esta dissertação, os quais foram organizados conforme as técnicas de aprendizagem utilizadas.

Devedzic [Dev01] faz um apanhado geral sobre o gerenciamento da aprendizagem, características de sistemas inteligentes e aponta várias áreas que podem ser enriquecidas por

agentes inteligentes e aprendizagem de máquina. Ele cita um trabalho em e-commerce onde os agentes trabalham no monitoramento dos assinantes de um site utilizando *lógica fuzzy*[4]. Assim, através da aprendizagem, o site oferece anúncios com grandes chances de interesse aos seus leitores, individualmente.

Um outro trabalho relacionado é o *framework* TAC (Trading Agent Competition) [TAC01] é um fórum internacional que tem por objetivo fomentar pesquisas de alta qualidade na área de negociação com leilões entre agentes autônomos. Para isso, é disponibilizado gratuitamente um *framework*, onde agentes desenvolvidos por equipes em diversas partes do mundo competem entre si, simulando a maioria das situações reais desse tipo de negociação. As estratégias para os leilões existentes devem ser completamente implementadas pela equipe que implementa o negociador, pois não existem componentes auxiliares que encapsulam técnicas conhecidas. Como não existe nenhum incentivo à utilização de uma técnica ou outra, diversas áreas de estudo têm utilizado a ferramenta para testes e pesquisas (por exemplo, a estatística, RBC, Q-learning [Sar05]). Mais detalhes sobre o TAC podem ser acompanhados no Apêndice C.

3.2.1 Utilizando RBC

Em [CBS+01; CBS+00] o sistema WEBSELL utiliza RBC em um sistema de recomendação inteligente que trabalha dentro de um perfil de cliente selecionado pelo sistema. No WEBSELL existe um módulo chamado de "Repositório de Conhecimento", onde estão representados os perfis pré-determinados, modelo dos produtos e outras informações para o funcionamento do sistema. A comunicação com o usuário é através da Internet, assim como entre os módulos internos do sistema que fazem uso de XML. Não existe um modelo de negociação, mas o RBC está presente na recomendação de produtos [Cun98]. De forma análoga, como tecnologia para ferramentas de apoio à decisão, o RBC trabalha recomendando soluções baseadas em casos de sucesso passados [IOOY01; Kas01].

Em [Nak03] existe uma preocupação nos riscos financeiros proporcionados pelas fraudes

[4]Ao contrário da lógica clássica, que reconhece apenas o falso ou verdadeiro, a *Lógica Fuzzy* reconhece diversos valores, assegurando que a verdade é uma questão de ponto de vista ou de graduação. Isso permite a manipulação informações imprecisas, como conceitos de pouco, alto, bom, muito quente, frio etc [MBCJ05].

e negociadores mal intencionados. Tal preocupação foi inspiração para um sistema que avalia e classifica consumidores confiáveis e não-confiáveis. O sistema *Credit risk information Service* utiliza o RBC para comparar os dados fornecidos pelo negociador, juntamente com informações físicas coletadas (Ex.: Endereço IP, padrões no preenchimento dos dados), na tentativa de encontrar casos anteriores que o classifique como confiável ou não.

Este último trabalho, em [Nak03], aborda uma questão que normalmente não é considerada nos modelos existentes para negociações automatizadas. Os trabalhos citados na Seção 3.1 consideram que os negociadores são honestos tanto nas informações trocadas nas propostas, como no cumprimento do acordo.

O RBC é também muito utilizado como método de aprendizagem em sistemas multiagentes para futebol de robôs, como em [Ste04; WB03; ALN+03]. As ações são escolhidas conforme análise de situações anteriores, onde o estado atual (posição e comportamento dos jogadores) é chave de entrada para o RBC indicar a melhor ação para o caso.

Em [SA04] e [lVSCFM+04] o RBC é utilizado em projetos de *Helpdesk*[5], melhorando a qualidade dos sistemas de ajuda eletrônica aos usuários. Basicamente a função do RBC é procurar questionamentos anteriores, através de palavras chaves e análise semântica das expressões, e oferece-los como resposta, segundo um limiar de similaridade à questão atual. As contribuições destes trabalhos estão nas soluções para identificação dos casos similares, que envolve a elaboração de um dicionário de expressões/palavras sinônimas e a identificação dos termos importantes na pergunta do usuário.

3.2.2 Utilizando Q-learning

Em [Tes04], Tesauro estende o algoritmo Q-learning e mostra em sua pesquisa que o algoritmo trás resultados significantes em Sistemas Multiagentes. O seu trabalho demonstra a convergência do algoritmo e mostra a avaliação das simulações com seus agentes aprendendo estratégias em um jogo simples. O algoritmo de Tesauro, Hyper-Q learning, utiliza-se da regras bayesianas para otimizar a convergência do algoritmo.

Em [SC95], Sandholm and Crites discutem o uso de Q-learning e através da repetição o jogador aprende sobre seu oponente. Simulações foram feitas onde o jogador consegue

[5]Espaço reservado aos clientes para consulta e assistência.

aprender a jogar da mesma forma que seu oponente, mas apresenta dificuldades diante outros jogadores que também aprendem.

Em [FR99], Gedson e Roseli utilizaram Q-learning para otimizar a navegação de um robô. Os sensores espalhados nas extremidades do robô fornecem informações para o algoritmo aprender se a ação escolhida (ir pra frente, esquerda, pra trás) foi adequada ou não. Observou-se que o algoritmo Q-learning conseguiu aprender de forma satisfatória, conforme o ajuste nos índices de recompensa ou punição oferecidos em cada ação.

Em negociações, a aprendizagem por reforço pode também ser observada em [Fon01], onde Manoel em sua tese utiliza Q-learning para que agentes negociadores se adaptassem automaticamente na gestão da construção civil. Embora o trabalho tenha o objetivo de apresentar um protocolo para coligações em sistemas multiagentes, foi necessário a construção de um modelo inteligente para testes, onde a aprendizagem por reforço contribuiu para a escolha da proposta mais adequada em cada momento. Segundo testes no trabalho [Fon01], o Q-learning se adaptou bem ao problema fornecendo resultados significativos, mas Manoel [Fon01] deixa claro que existe a necessidade de uma comparação do Q-learning com outras técnicas de aprendizagem.

A aprendizagem por reforço também foi aplicada à solução de um jogo lógico conhecido como “Dilema do prisioneiro” (ou *Prisoner’s Dilemma*) [SC95]. O Jogo é bastante simples, onde dois amigos (representados por agentes) são presos por um crime leve, mas a polícia sabe que um deles cometeu um crime pesado. Os dois têm que escolher entre “denunciar o amigo” ou “ficar calado”. As regras são:

- Se apenas um denunciar, este fica livre (pela cooperação) e o outro fica preso por 5 anos;
- Se os dois se denunciarem, os dois pegam três anos de prisão (a policia conclui que os dois cometeram o crime pesado);
- Se os dois ficarem calados, os dois ficam um ano na cadeia (pelo crime leve).

Aplicando a aprendizagem por reforço, várias vezes, fazendo que os agentes aprendessem após muitas interações. O algoritmo Q-learning sugere que inicialmente eles optaram pela ação mais lucrativa, mas sempre eram penalizados severamente, o que fez com que adotassem a outra opção como mais adequada. O trabalho de [SC95] não se resume a isso,

ele também testou várias combinações para punição e incentivo no algoritmo de aprendizagem, comparando-as e classificando-as conforme o desempenho apresentado. Mais tarde, em [San02], o mesmo autor propôs um ambiente protótipo para negociações em leilões, mas não utilizou aprendizagem por reforço no modelo.

3.2.3 Utilizando Redes Neurais

Utilizando-se de redes neurais, o trabalho de [MR04] utiliza os sonares do robô, em locais pré-organizados, para treina-lo na identificação e elaboração de mapas para navegação, que são criados conforme os obstáculos são reconhecidos no ambiente. A navegação do robô também utiliza-se de Q-learning para evitar colisões e para escolha de melhores rotas.

Em [LKK04; Lag02] um projeto utiliza Redes Neurais SOM (*Self-Organizing Maps*) para classificar livros por assuntos. Trata-se de um sistema web que organiza documentos pelos seus temas e assuntos, facilitando a recuperação de todo material correlato em uma pesquisa acadêmica.

Soluções de controle para estruturas complexas, onde a calibração ou conjunto de ajustes ideais para funcionamento são dependem de muitos fatores, por exemplo, no controle de caldeiras de uma usina, ajuste de uma antena espacial , controle de pressão e bombeamento de petróleo ou mesmo no controle de processos em usinas [BR00; Sou00]. Em negociações por meios eletrônicos, existe um problema quanto à identificação de fraudes, que por considerar diversos fatores é considerado é considerado complexo. Não existe uma fórmula definida que separe a negociação fraudulenta da honesta. Semelhante ao sistema que identifica situações de fraude utilizando RBC, o trabalho em [Tre04] propõe a utilização de Redes Neurais para identificação de situações de risco. O treinamento da Rede Neural considera diversas informações, como hora, CEP, localização geográfica, endereço IP, dados do comprador, dados do cartão de crédito, valor da compra, entre outras. Após treinada, a Rede Neural consegue identificar situações de risco rapidamente. Mas, para que possa "aprender" novos padrões de fraude, deve ser treinada com freqüência.

Exitem muitos trabalhos utilizando Redes Neurais em negociações e alguns estão catalogados em [VLV99]. Um deles [Opr02], é destacado por incluir em sua estratégia o mapeamento do negociador oponente. Nesse trabalho, apenas o preço é considerado durante a negociação e uma Rede Neural é utilizada para previsão de séries temporais. Para este caso,

onde se tem apenas o preço como atributo, as propostas enviadas por cada jogador podem ser vistas isoladamente como séries temporais, já que é previsto que os jogadores cedam com o tempo (Tanto o vendedor diminuindo o seu preço, como o comprador aumentando sua oferta). Dessa forma, são informadas as últimas 3 propostas do negociador oponente para entrada da Rede Neural, e esta informa qual o provável valor na próxima proposta. Essa informação é utilizada para a escolha da melhor estratégia conforme o momento da negociação. Para o treinamento da Rede, são utilizados históricos com seqüencias de propostas recebidas em diversas negociações.

Já no trabalho desenvolvido por Sardinha [Sar05], é mostrado um sistema multiagente desenvolvido para negociação na competição anual TAC (*Trading Agent Competition*), que utiliza diversas técnicas da IA em leilões competitivos, dentre as quais as Redes Neurais. A Rede Neural foi utilizada com Aprendizagem por Reforço para encontrar lances ótimos para o leilão. Mais tarde, a Rede Neural foi substituída para evitar a necessidade de treinamento, uma vez que na competição o agente sempre está muito ocupado e não pode ser parado. Como a utilização da Rede Neural foi abandonada, o trabalho não entra em detalhes quanto à utilização híbrida com Aprendizagem por Reforço, mas o texto sugere que a Rede Neural esteve presente para apontar a ação mais apropriada para estado do leilão.

3.3 Síntese dos Trabalhos Revisados

Os trabalhos comentados nesta última seção mostram o poder de aprendizagem e adaptação das técnicas de aprendizagem nas diversas situações, o que justifica suas utilizações no âmbito do comércio eletrônico. A possibilidade de utilização de cada técnica de aprendizagem, aplicadas à situações particulares da negociação, será vista no Capítulo 4 em detalhes.

Sobre os trabalho diretamente relacionados à negociação (Seção 3.1), a Tabela 3.3 mostra uma síntese desses estudos realizados anteriormente na linha do comércio eletrônico e suas atuações na negociação: antes de forma mais distante do modelo de uma negociação real e, mais recentemente, com um modelo mais aperfeiçoado que se aproxima da realidade. Na última etapa alcançada tem-se um ambiente de negociação que, além de trabalhar com vendas e até mesmo poder oferecer produtos alternativos ao consumidor, também se preocupa com a personalização da negociação, assim, oferecendo mais vantagens, descontos ou itens

estratégicos através do estudo do histórico do cliente, por exemplo. O próximo passo a ser tomado é dar suporte ao conhecimento na extração de características do comportamento humano e de informações do ambiente para o agente vendedor. Esse suporte à negociação será feito utilizando-se das técnicas de Redes Neurais, Q-learning e RBC inseridas nos agentes de suporte à venda.

	Kasbah (1996)	Wreper (2000)	Modelo por Gustavo de Paula(2001)	Modelo por Orlando Pinho Jr. (2002)
Representação da Proposta	Único atributo(preço)	Múltiplos atributos	Múltiplos atributos	Múltiplos atributos
Avaliação da Proposta	Com base no preço	Combinação ponderada dos atributos	Idem	Idem
Possíveis Ações	Aceitar, sair, rejeitar e gerar contraproposta	idem	Idem + ultimato + produto alternativo	Idem
Tomada de Decisão	Compara propostas recebidas e geradas	idem	Idem	Idem + suporte a decisão por agentes
Contra Proposta	Dependente de tempo	Idem + comportamento e recurso	Idem	Idem + histórico pessoal
Ultimato	Não	não	sim	sim
Produtos alternativos	Não	não	sim	sim
Aprendizagem	Não	não	não	não

Tabela 3.1: Quadro comparativo entre os principais modelos de negociação revisados

Capítulo 4

Um Ambiente para Negociações Automatizadas

Nesse capítulo é proposta uma nova solução para negociação automatizada entre agentes, tendo em vista o suporte à aprendizagem. Esse modelo surgiu com o objetivo de aproximar os trabalhos existentes, relacionados à negociações automatizadas, à realidade existente na negociação entre os seres humanos. Junto à aprendizagem, através da IA buscou-se adicionar aspectos cognitivos robustos com o objetivo de aumentar as chances do negociador em conseguir um acordo, com um bom nível de satisfação e no menor tempo de interação possível.

Na Seção 4.1 definimos o modelo do ambiente que nossos agentes irão interagir, na Seção 4.2 definimos o modelo de tomada de decisão do agente e por último, na Seção 4.3, definimos o nosso modelo abstrato de aprendizagem.

4.1 Descrição do Ambiente

O ambiente proposto neste trabalho visa possibilitar aos agentes condições de realizar negociações automatizadas de forma aberta e multi-lateral. Como o ambiente é aberto, em qualquer momento os agentes negociadores poderão entrar ou sair da sociedade, como também poderão estar negociando bilateralmente com mais de um oponente. Isso possibilita a esses agentes um poder maior barganha, dado que terão uma melhor visão do que está acontecendo no mercado.

4.1.1 Variáveis de Decisão

As *variáveis de decisão* V servem para descrever sob quais condições um determinado acordo poderá ser negociado. Por exemplo, poderia ser interessante descrever uma negociação de automóveis com as seguintes variáveis de decisão: preço, marca, modelo, ano fabricação e quilometragem. Em outro contexto, onde só existem veículos novos, não é mais interessante negociar sob o ano de fabricação e quilometragem, mas sim, com opcionais para o veículo.

Todas as variáveis de decisão são interpretadas da mesma maneira, tanto aquelas que representam as características dos produtos, como as que servem para mapear as questões envolvidas na negociação. Isso é importante para simplificar o mecanismo de avaliação ou interpretação de propostas. Na Equação 4.1, está representado um conjunto V de variáveis de decisão.

$$V = \{v_1, v_2, ..., v_n\} \tag{4.1}$$

em que v_i representa a i-ésima variável de decisão possível em um domínio de negociação. Cada variável de decisão, tem o seu domínio particular de possíveis valores, representado por um conjunto de valores discretos ou contínuos:

- As variáveis chamadas de contínuas devem possuir valores no domínio dos números reais, conforme descrito na Equação 4.2 [OSB+06].

$$v_{i_{continua}} = [valor_{minino}, valor_{maximo}] \tag{4.2}$$

- Por outro lado, as variáveis de decisão que possuem um conjunto de valores discretos em seu domínio, são representadas segundo a Equação 4.3, em que $valor_i$ representa o i-ésimo valor possível de m valores possíveis no total.

$$v_{i_{discreta}} = \{valor_1, valor_2, ..., valor_m\} \tag{4.3}$$

4.1.2 Entidades Participantes

No modelo apresentado neste capítulo, uma sociedade para negociação possui três tipos de entidades protagonistas, que são entidades necessárias para que uma negociação possa exis-

tir, e podem ser identificados como SMA vendedores, SMA compradores ou agente catálogo. Essa definição pode ser acompanhada na Figura 4.1, e mais detalhes sobre o conteúdo interno de cada SMA Negociador (Agentes internos mostrados na Figura 4.1) serão fornecidos na Seção 4.2.

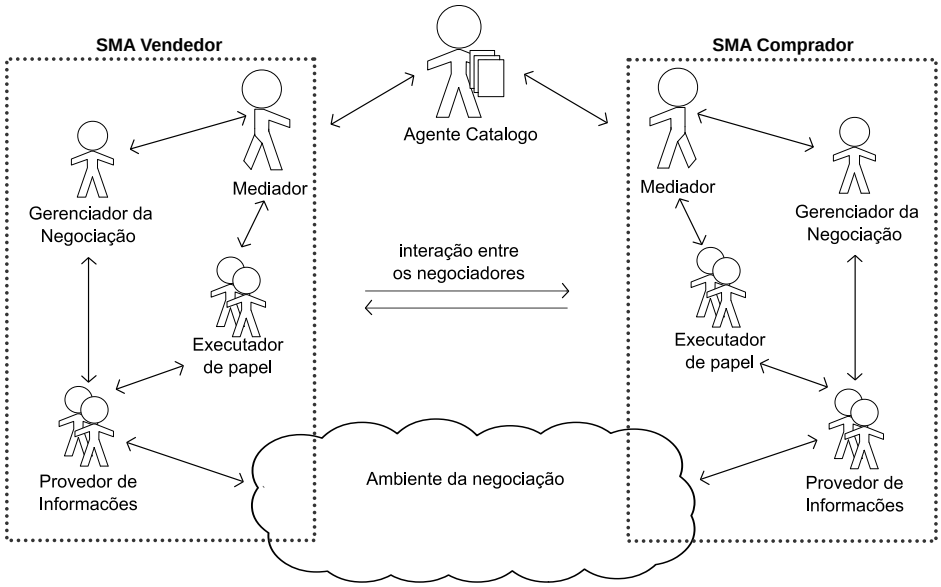

Figura 4.1: Entidades no ambiente de negociação

Os compradores e vendedores possuem características similares, tanto no sentido das ações que eles podem tomar (ex.: mudança de comportamento relacionado à estratégia de negociação), como nas possíveis combinações de técnicas para o modelo de decisão. Por causa dessa pequena diferença, essas entidades são chamadas de *SMA negociadores*, que possuem papéis distintos para vendedor e comprador:

- No papel de vendedor, o SMA tem o dever de esperar, receber e responder à requisições de compra dos compradores. Em suas negociações, um vendedor, pode ser dinâmico a ponto de barganhar utilizando sua experiência, seu poder financeiro, estoque, ou mesmo considerar outras negociações em paralelo.

- No papel de comprador, o SMA deverá ser capaz de procurar vendedores junto ao agente catálogo, além de utilizar suas estratégias para conseguir menores preços e melhores condições de compra.

O Agente Catálogo é único dentro da sociedade, e tem por objetivo fazer a intermediação inicial entre dois SMAs negociadores. Ao entrar na sociedade, o vendedor deve informar

ao agente catálogo sobre quais domínios ele é capaz de negociar. Por outro lado, o agente comprador se utiliza do agente catálogo, solicitando a ele a lista de vendedores para um tipo específico de negócio que este deseja negociar.

4.1.3 Protocolo da Negociação

O protocolo de negociação foi adaptado de [BPJ04; Jun02; Pau01], e expandindo idéias desses trabalhos com o objetivo de proporcionar uma negociação realística possível. Com isso, é proposto um modelo onde as interações entre os negociadores possam ser mais expressivas, aumentando conseqüentemente o poder de barganha em uma negociação. Esse protocolo, que define a interação entre os agentes protagonistas, será detalhado em cada uma das fases a seguir.

Encontro Entre Negociadores

Inicialização do Vendedor: Antes de que um SMA vendedor comece de fato a negociar, será necessário que este faça um anúncio de sua disposição e quais seus domínios de comércio. Nesta última frase, está implícita a idéia de que é possível um SMA negociador suportar negociação em vários domínios, então, um mesmo vendedor deverá internamente dar suporte a esses domínios de forma independente. Veremos como isso é possível na Seção 4.2. O anúncio dos domínios de comércio do SMA Vendedor é feito em um catálogo do tipo "páginas amarelas", que pode ser consultado por qualquer agente no ambiente (O conceito de "páginas amarelas" é melhor explicado no Apêndice B).

Solicitação de Vendedores: Antes que qualquer comprador possa iniciar uma troca de mensagens com um vendedor, é necessário que ele conheça o endereço físico de cada um dos agentes vendedores. Isso é conseguido com uma solicitação feita ao agente catálogo.

Solicitar Negociação: Para que se possa iniciar uma negociação entre dois negociadores é necessário que o SMA comprador solicite ao SMA vendedor uma negociação. No nosso modelo, um SMA comprador pode solicitar uma negociação a um ou vários vendedores paralelamente. Essa solicitação deve incluir dados referentes ao cliente e ao objeto que ele deseja negociar. O passo seguinte é o vendedor responder à esta solicitação, informando

se poderá negociar ou não. Caso a resposta seja positiva, será necessário enviar também uma identificação da negociação (código único no ambiente) que será utilizada por ambos na negociação. Após o recebimento da identificação da negociação, os dois agentes entram na fase de *pré-negociação*.

Pré-Negociação

Aqui, os dois agentes entram em consenso sobre quais variáveis de decisão vão utilizar (Figura 4.2), o que inclui a identificação das variáveis de decisão que possuam valores fixos (tenha que ter um valor pré-ajustado).

Essa pré-negociação acontece em três iterações de troca de mensagens. Na primeira, o agente vendedor envia o seu escopo de variáveis de decisão ($V'_{vendedor}$) que este, por algum motivo, deseja negociar. O escopo de variáveis de decisão $V'_{vendedor}$ é subconjunto de V, que representa todas variáveis possíveis para o domínio do negócio. $V'_{vendedor}$ é representado pela Equação 4.4, onde, $V' \subset V$, ou seja, cada v'_i representa uma variável de decisão no domínio (V) considerada na negociação.

$$V'_{vendedor} = \{v'_1, v'_2, ..., v'_i\} \tag{4.4}$$

Já a resposta do comprador é dada por $V'_{comprador}$, onde $V'_{comprador} \subset V'_{vendedor}$. Assim, $V'_{comprador}$ representa a intersecção das variáveis de decisão que os negociadores desejam utilizar durante a negociação.

Negociação

Na negociação, os dois agentes negociadores buscam um acordo que satisfaça à ambição de ambos. Isso é feito através de um jogo, com a troca de propostas entre os SMAs. Na Equação 4.5 é definido o nosso modelo de proposta P, onde:

- $Id_{proposta}$: representa a identificação da proposta;
- $Id_{negociacao}$: representa a identificação da negociação;
- $Id_{rementente}$: representa a identificação do negociador que enviou a proposta;
- $Id_{destinatário}$: representa a identificação do SMA destinatário;

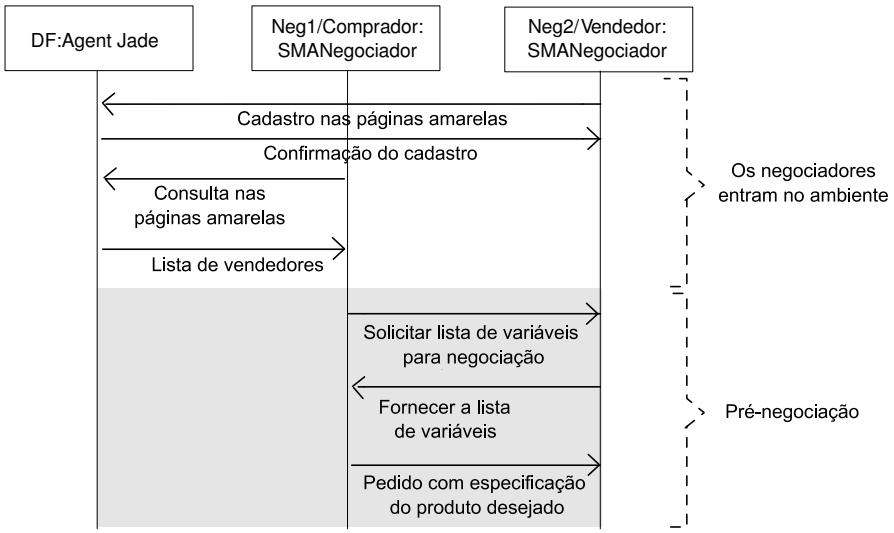

Figura 4.2: Diagrama de seqüência AUML inicial

- $TipoProposta$: *flag* que a principio indica apenas se trata-se de um *ultimato*[1] ou não.;
- $ValoresDaProposta$: representa um conjunto de valores atribuídos a cada variável do escopo da negociação em questão, mostrado na Equação 4.6. Os possíveis valores para cada variável, podem ser discretos ou contínuos como foi definido na Subseção 4.1.1.

$$P_{=}\{Id_{proposta}, Id_{negociacao}, Id_{rementente}, Id_{destinatário}, TipoProposta, ValoresDaProposta\} \quad (4.5)$$

$$ValoresDaProposta = \{v'_1 = valor_atribuido_1, v'_2 = valor_atribuido_2, ..., v'_n = valor_atribuido_n\} \quad (4.6)$$

A negociação é sempre iniciada pelo vendedor, com o envio da primeira proposta. Após isso, o mesmo espera receber uma resposta do comprador com a sua avaliação da proposta (A_P). Caso o agente comprador não tenha aceito essa primeira proposta do agente vendedor, então, será a vez do agente comprador enviar um proposta, e cabe agora ao agente vendedor enviar uma avaliação (Figura 4.3). Esse ciclo termina quando um acordo é estabelecido ou quando um dos negociadores sai da negociação.

O formato dessa avaliação está definido na Equação 4.7, onde, $Id_{proposta}$ representa a identificação da proposta que está sendo avaliada e $avaliacao$ indica qual a avaliação do

[1] *Ultimato* no contexto do comércio eletrônico automatizado, refere-se a última proposta enviada pelo negociador antes de abandonar a negociação.

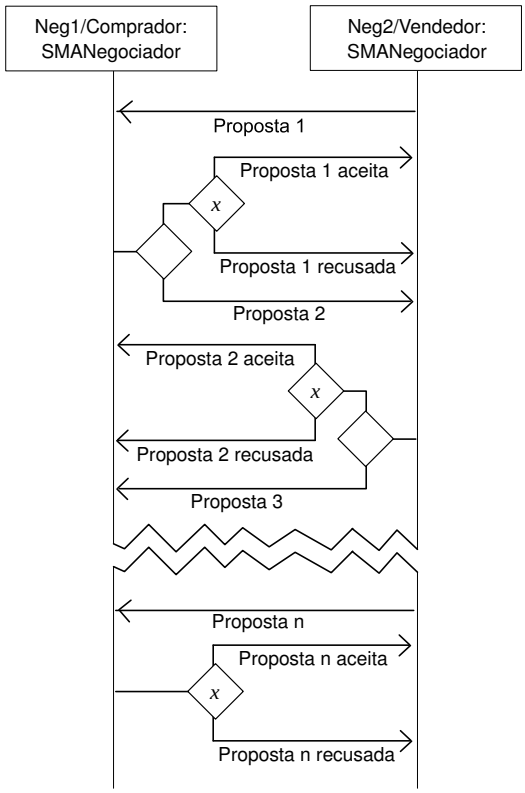

Figura 4.3: Troca de propostas entre SMAs Negociadores

agente sobre a proposta. Essa avaliação pode ser uma das seguintes:

$$A_P = \{Id_{proposta}, avaliacao\} \tag{4.7}$$

- $Aceitar$: Representa uma resposta positiva em relação a proposta recebida, em outras palavras, indica o final da negociação competitiva com um acordo estabelecido.
- $Rejeitar$: Indica que o agente não aceitou a proposta, e que o mesmo vai enviar uma contra-proposta.
- $Cancelar$: Indica que o agente não aceitou a proposta e desistiu da negociação.

4.2 Arcabouço do Negociador

O SMA negociador descrito na Seção 4.1, pode ser implementado utilizando o arcabouço para negociação multiagente que é proposto nesta seção. O arcabouço para construção dos SMAs Negociadores possui as seguintes características:

- Oferece o controle de várias negociações em vários domínios: O arcabouço oferece um mecanismo de administração para cada uma das negociações em que o agente está participando;

- Possui, internamente, uma estrutura de troca de propostas entre os SMAs negociadores, com o propósito de não passar ao desenvolvedor a responsabilidade de ter que implementar o protocolo de comunicação entre os agentes. Este poderá ser sobrescrito[2] se for o caso.

- Fornece meios de acoplamento para os mecanismos de aprendizagem na negociação, favorecendo um ajuste ou aperfeiçoamento das estratégias utilizadas nas negociações.

O arcabouço fornece facilidades para o usuário desenvolvedor, como por exemplo, no incremento do poder de barganha do SMA Negociador através das técnicas de aprendizagem. De forma abstrata, a única tarefa que o usuário deve implementar é como responder à questões requisitadas por agentes externos.

Cada SMA Negociador possui 4 tipos de agentes que o compõe: *Agente Mediador*, *Agente Gerenciador das Negociações*, *Agente Executor* e *Agente Gerador de Informação*. A Figura 4.1, na página 49, ilustra a disposição desses agentes no SMA, que serão detalhados em seguida.

4.2.1 Agente Mediador (AM)

O agente mediador é o responsável pela troca de mensagens entre os agentes internos do SMA Negociador e os SMA's Negociadores Oponentes. É ele que recebe, por exemplo, uma proposta, e a repassa para o agente interno responsável por tratá-la, e vice-versa.

Do ponto de vista do usuário desenvolvedor, essa tarefa é executada automaticamente, sendo apenas necessário que ele forneça um conjunto de informações de inicialização para o agente mediador ($S_{mediador}$), conforme representado na Equação 4.8.

$$S_{mediador} = \{\vec{E}, G, \vec{D}\} \tag{4.8}$$

[2]Na Engenharia de Software, a sobrescrição de métodos de classes é conhecida por *Override*

em que $\vec{E}$ representa a lista de identificações dos agentes que executam os papéis nos domínios (esta informação poderá ser obtida em *runtime*, através do *agente catálogo*[3]); G representa a identificação do gerenciador de negociação; e $\vec{D}$ representa um vetor com os domínios que o SMA Negociador poderá negociar.

O agente mediador trabalha com os seguintes tipos de mensagem:

- *Mensagem de Confirmação de Solicitação de Negociação*: Essa é uma mensagem que dá uma resposta positiva ou negativa a uma *Mensagem de Solicitação de Negociação*. Observando pela perspectiva de um SMA Comprador, esta mensagem sempre será enviada pelo SMA Vendedor que recebeu a solicitação.

- *Mensagem de Solicitação de Negociação*: Uma mensagem de solicitação para uma negociação é enviada ao SMA Negociador quando este se inscreve na sociedade como um comprador, conforme o protocolo descrito na Subseção 4.1.3. No caso contrário, quando o SMA está atuando como vendedor, é ele quem vai receber este tipo de mensagem.

- *Mensagem de Definição do Escopo de Negociação*: Esse tipo de mensagem representa uma resposta à *Mensagem de Proposta de Escopo de Negociação*, do SMA Comprador ao SMA Vendedor. O conteúdo da mensagem contém o escopo final da negociação, definido por $V'_{comprador}$ na Subseção 4.1.3.

- *Mensagem de Proposta de Escopo de Negociação*: As mensagens de proposta de um escopo de negociação serão sempre enviadas por um SMA Negociador que estiver atuando como vendedor, e será sempre recebida pelo SMA Comprador. O conteúdo desta mensagem corresponde ao conjunto de variáveis de decisão representados por $V'_{vendedor}$ na Equação 4.4, Subseção 4.1.3.

- *Mensagem de Avaliação de Proposta*: Igualmente às *Mensagens de Proposta*, as *Mensagens de Avaliação de Proposta* também podem ser enviadas por qualquer tipo de agente (SMA Vendedor ou SMA Comprador), e conseqüentemente, recebidas por qualquer tipo de agente. É uma resposta a uma *Mensagem de Proposta* e tem o seu conteúdo definido pela Equação 4.7, na Subseção 4.1.3.

[3]Consultar Subseção 4.1.2, página 48, para mais detalhes.

- *Mensagem de Proposta*: A Mensagem de proposta pode ser enviada por qualquer tipo de agente atuando em qualquer domínio, SMA Comprador ou SMA Vendedor. Conseqüentemente, ela também pode ser recebida por qualquer tipo de agente em qualquer domínio. O conteúdo desta mensagem foi definido na Subseção 4.1.3, na Equação 4.5.

4.2.2 Agente Gerenciador das Negociações (AGN)

O *agente gerenciador de negociação* é o agente responsável por administrar o ciclo de vida de cada uma das negociações existentes, para garantir que seja seguido o protocolo de negociação definido na Subseção 4.1.3. Também é de sua responsabilidade a orientação do *Agente Executor* (descrito na subseção em seguinte) sobre qual estratégia adotar na negociação. Se o SMA Negociador está atuando como comprador, será atribuído a ele também a tarefa de iniciar o ciclo de vida de uma negociação.

O gerenciamento é realizado com o auxílio de um conjunto de informações iniciais

$$I_{AGN} = \{t, \vec{a}\} \tag{4.9}$$

onde: t indica o intervalo de tempo que o agente aguarda antes de buscar atualizações (coletadas de fontes internas ou externas ao SMA Negociador); e $\vec{a}$ representa os analisadores[4] considerados em cada domínio, os quais deverão ser desenvolvidos pelos usuários do SMA Negociador.

A representação de uma negociação (N), para o *agente gerenciador de negociação*, é descrita por

$$N = \{Id_N, Id_{oponente}, tipo_papel, status, Id_V, Id_{perfil_oponente}\} \tag{4.10}$$

onde:

- Id_N: representa a identificação da negociação em questão;
- $Id_{oponente}$: representa a identificação do negociador oponente;
- $tipo_papel$: representa o tipo de papel que o SMA Negociador exerce naquela negociação (comprador ou vendedor);

[4] Abstratamente, são funções que indicam "o que eu devo atualizar e como".

- $status$: representa o status da negociação no momento corrente, esse status pode ser ajustado pelo agente gerenciador de negociações no intervalo de tempo definido em t.
- Id_V: representa o domínio que a negociação esta referenciando;
- $Id_{perfil_oponente}$: representa a identificação do tipo de perfil que está referenciando o oponente, que deverá ser determinada através do *agente gerador de informação*.

Os possíveis valores para $status$ e sua forma de interpretação são:

- *Negociação de Proposta*: Esse status é atribuído somente a SMA Negociadores que estão atuando como Vendedores. Indica que uma negociação que foi solicitada por um determinado SMA Comprador, e o SMA Vendedor deve enviar uma resposta caso queira iniciar uma negociação.
- *Aguardando Escopo da Negociação Vendedor*: Esse status é atribuído somente a SMA Negociadores que estão atuando como comprador, e representa dependência de uma resposta do SMA Vendedor (contendo $V'_{vendedor}$)[5] para que a negociação possa dar andamento.
- *Enviar Escopo Negociação Final*: Esse status é atribuído somente a SMA Compradores, que devem enviar o escopo final para a negociação ($V'_{comprador}$, onde $V'_{comprador} \subseteq V'_{vendedor}$)[5] , caso continue intencionado a negociar.
- *Aguardando Escopo da Negociação Final*: Esse status é atribuído somente SMA Vendedores. Aqui o SMA Vendedor está esperando uma resposta do SMA Comprador ($V'_{comprador}$)[5], sobre o escopo final da negociação que eles estarão negociando.
- *Enviar Proposta*: Indica que o SMA Negociador (comprador ou vendedor) deve enviar uma proposta ao seu oponente. O primeiro SMA Negociador a atribuir esse status é o SMA Vendedor, que tem a responsabilidade de enviar a primeira proposta depois de ter recebido o escopo da negociação final do SMA Comprador.
- *Aguardando Avaliação Proposta*: Esse status indica que o SMA Negociador esta esperando uma avaliação de uma proposta enviada ao seu oponente.

[5]Consultar Subseção 4.1.3 para mais detalhes.

- *Aguardando Proposta*: Indica que o SMA Negociador recebeu uma resposta negativa em relação a sua última proposta enviada, e portanto está esperando que seu oponente envie uma outra proposta. A exceção se dá quando o SMA Comprador envia o escopo final da negociação e fica esperando o SMA Vendedor enviar a primeira proposta da negociação.

- *Enviar Avaliação*: Esse status indica que o SMA Negociador precisa enviar uma avaliação de uma proposta que recebeu de seu oponente.

4.2.3 Agente Executor (AE)

Existe um *agente executor* para cada papel em um domínio de negociação. Assim, se por exemplo, o SMA Negociador trabalha no domínio de calçados e CDs, vendendo e comprando ao mesmo tempo, devem existir quatro agentes executores, um comprador e um vendedor para cada domínio. O agente executor tem um papel fundamental dentro do SMA Negociador, pois é ele quem tem a responsabilidade de executar uma tarefa que esteja definida no status (do *Agente Gerenciador das Negociações*) de cada uma das negociações relacionadas ao seu domínio de negociação.

Em seguida, será mostrado como um *agente executor* realiza suas tarefas segundo o status de cada negociação (considerando que esteja no seu respectivo domínio de atendimento):

- Quando identifica o status *Negociação Proposta*, e nenhum escopo de negociação ($V'_{vendedor}$) foi enviado ao SMA Comprador: então, o agente executor enviará uma mensagem, contendo ($V'_{vendedor}$), para que o *agente mediador* possa enviar ao SMA Comprador;

- Quando identifica o status *Enviar Escopo Negociação Final*, e nenhuma mensagem de escopo de negociação final referente a essa negociação foi enviada: então o agente enviará um escopo $V'_{comprador}$ ao *agente mediador*, para que este envie a resposta ao SMA Vendedor;

- Quando identifica o status *Enviar Proposta*, e não existe proposta postada referente ao momento da negociação corrente: então o *agente executor* irá elaborar uma proposta para ser enviada pelo *agente mediador*.

- Quando identifica o status *Enviar Avaliação* e ainda não foi postada uma resposta referente à última proposta recebida: então o *agente executor* irá avaliar e enviar essa resposta ao *agente mediador*, que providenciará que seja entregue.

4.2.4 Agente Gerador de Informação (AGI)

Este agente é optativo no SMA Negociador, mas quando utilizado oferece vantagem à tomada de decisão por ser capaz de perceber o ambiente e prover informações relevantes aos agentes executores. Em princípio, apenas o *agente gerador de informação* enxerga o ambiente externo ao SMA Negociador.

4.3 Modelo de Aprendizagem

A arquitetura do SMA negociador está representada na Figura 4.4, onde estão destacadas (com bordas mais grossas) as classes relacionadas à aprendizagem e tomada de decisão. O objetivo aqui é mostrar a estrutura do arcabouço para criação de SMA Negociadores inteligentes, que poderão fazer uso de quaisquer técnicas da IA disponíveis.

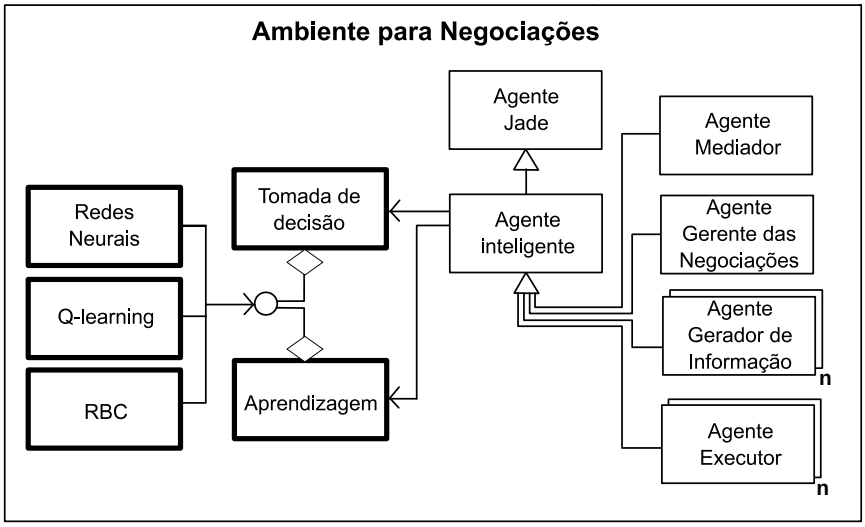

Figura 4.4: Arquitetura do SMA Negociador

4.3.1 Tomada de Decisão × Aprendizagem

No modelo proposto, técnicas de Inteligência Artificial estão presentes para duas finalidades bem distintas:

Tomada de decisão: na escolha da melhor estratégia para cada situação, na elaboração de propostas interessantes, e até mesmo na escolha da melhor combinação de técnicas de negociação. A tomada de decisão então está presente como uma ferramenta disponível para os agentes da classe *Agente Inteligente*, que poderá se utilizar de qualquer técnica relacionada à tomada de decisão e aprendizagem: RBC, Raciocínio Baseado em Regras, Rede Neural, Aprendizagem por reforço etc.

Vendo a tomada de decisão como um mecanismo (Figura 4.5) que considera fatos relacionados ao problema e toma uma decisão (ou fornece um indicativo para uma ação), pode-se considerar a utilização de qualquer técnica da IA, sendo que algumas necessitam de um treinamento prévio, outras necessitam de uma amostragem inicial, outras de um conjunto de regras de decisão. Esse fomento para o comportamento dessa "função de decisão" deverá ser fornecido à parte, antes da sua utilização.

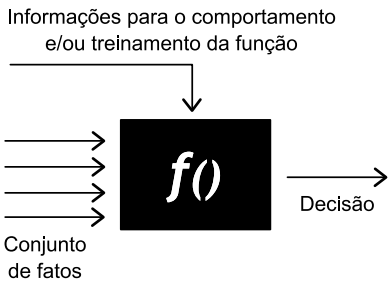

Figura 4.5: Tomada de decisão vista como uma função

Aprendizagem: é o que assegura a inserção de novas informações à base de dados do agente. Essa aprendizagem pode acontecer de duas formas: com o tempo, à medida que o agente recebe *feedbacks* relacionados as suas ações no ambiente; ou de forma reservada, quando acontece após o acúmulo de muitas informações/exemplos sobre o comportamento a ser aprendido. A aprendizagem pode ser vista como um processo de ajuste de informações já conhecidas ou como a aquisição de uma nova informação (ou regra).

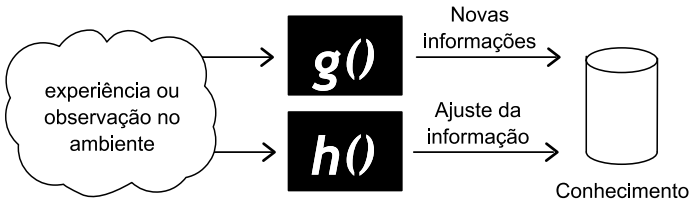

Figura 4.6: Processo de aprendizagem visto como uma função

Para facilitar o entendimento, a aprendizagem é tratada isoladamente, como se o processo de aprendizagem alimentasse uma base de dados que futuramente serão utilizados na tomada de decisão, mas não acontece sempre assim: o processo de aprendizagem pode também estar diretamente ligado à tomada de decisão, por exemplo com o Q-learning e RBC, onde se faz obrigatório a utilização do módulo de aprendizagem após cada tomada de decisão.

4.3.2 Facilidade na Utilização das Técnicas

Como a classe *Agente Inteligente* "conhece" todas as técnicas para tomada de decisão e aprendizagem, todos os seus descendentes também herdam essas características. Em cada um dos agentes que compõem o SMA Negociador foram identificados pontos que merecem soluções baseadas na IA. Assim, em cada agente existem soluções simples para tomada de decisão e aprendizagem, que podem ser sobrescritas conforme exista necessidade. O objetivo aqui é fornecer um SMA Negociador básico, apto a trabalhar, bastando apenas uma pequena configuração, e toda infraestrutura para aperfeiçoa-lo de acordo com as necessidades do meio. Dessa forma, além de um sistema voltado para o comércio, configura-se também um ambiente de estudos para as técnicas de IA.

Existem diferenças entre as técnicas disponíveis, não somente quanto à teoria ou filosofia da ciência que a suporta, mas também nos seus pré-requisitos e configuração para utilização.

Tomada de decisão : Para a tomada de decisão, pode-se utilizar o *Raciocínio Baseado em Regras*, através de uma máquina de inferência. Para o presente trabalho sugere-se o Jess [FH], por ser uma ferramenta finalizada e por sua compatibilidade com o projeto. Para utilização de uma técnica baseada em regras como modelo de decisão, é preciso preencher as seguintes condições:

1. Ter a experiência/conhecimento mapeado em regras do tipo "*SE condição ENTÃO ação*", que deverão estar disponíveis à máquina de inferência;
2. Buscar *fatos* no ambiente, relacionados às regras, que servirão de "gatilho" para o processo de inferência;
3. Entender a semântica no resultado do processo de inferência, que deverá ser traduzido para uma ação correspondente.

Ainda para a tomada de decisão, existe o Q-learning, onde também é exigido um conhecimento prévio sobre os pares estado-ação no domínio do problema, além de uma função ou metodologia de avaliação para a ação no estado. A vantagem dessa técnica é que ela está diretamente relacionada a um processo de aprendizagem (no seu processo de avaliação), possibilitando uma tomada de decisão que obedece à dinâmica do ambiente. Para utilização do Q-learning, se faz necessário:

1. O conhecimento definido dos estados do ambiente e todas as possíveis ações associadas a eles;
2. Uma metodologia para avaliação, que através de recompensas e punições será responsável pela aprendizagem da técnica;
3. Saber discretizar os estados do ambiente para entrada do algoritmo, assim como entender a semântica na saída do mesmo.

Uma Rede Neural também pode ser utilizada na tomada de decisões. Para isso, é fundamental a existência de um conjunto de exemplos sobre o conhecimento que se quer representar, para o treinamento da rede.

Assim, se por exemplo, deseja-se tomar decisão sobre o melhor dia para ir à praia, é necessário um conjunto de treinamento que contenham os fatores que influenciam decisão associados a decisão tomada. Um dado nesse conjunto de treinamento poderia ser a tupla:

```
( dia ensolarado & crianças de férias & hoje é sábado & o
carro está funcionando, dia para praia )
```

cujos atributos podem ser discretos ou não. A primeira parte da tupla (`dia ensolarado & crianças de férias & hoje é sábado & o carro`

`está funcionando`) seria a entrada da rede, indicando portanto 4 neurônios de camada de entrada. A segunda parte da tupla (`dia para praia`), a saída desejada, que deverá ser utilizada para correção dos pesos dos neurônios na fase de treinamento.

Aprendizagem : A característica "aprender", como comentado nesta seção, refere-se a aquisição de conhecimento em forma de regras ou casos, ou o ajuste de funções ou regras de forma dinâmica. Assim, a aprendizagem, em cada técnica da Inteligência Artificial (IA), pode ser identificada:

- no RBC a aprendizagem está na aquisição do novo caso. Aqui a preocupação é seleciona-lo segundo a sua relevância como exemplo, observando a existências de casos semelhantes na base de casos ou não. Deve-se atentar para a essência do próprio algoritmo que analisa a semelhança dos casos, aplicando medidas de similaridade, um a um. Assim, uma base de casos muito cheia influenciaria na performance do algoritmo [OSB+06];
- no Q-learning a aprendizagem é constante a cada incentivo ou punição para as ações tomadas. Uma atenção especial deve ser dada para os coeficientes de incentivo e punição, para que o modelo não desaprenda com uma avaliação fora dos padrões;
- numa Rede Neural a aprendizagem está no ajuste de pesos e relaciona-se com a qualidade do conjunto de treinamento. A aprendizagem da Rede Neural acontece em um momento anterior a sua utilização. Deve-se tomar cuidado na escolha do coeficiente que representa o erro médio aceitável no treinamento, para não especializa-la[6] demais (*overfitting*) [OSB+06];
- com RBR[7] a aprendizagem acontece por meio de novas regras adicionadas na base de conhecimento, que podem ser incluídas automaticamente, por meio de um software inteligente, ou manualmente, através de um administrador externo. Na aprendizagem, deve-se atentar para não tornar a base de conhecimento inconsistente pela inserção de regras contraditórias. Segundo Castillo [CGH97],

[6] Quando o treinamento é intenso a ponto de tirar seu poder de generalização na classificação de dados da mesma classe.

[7] Raciocínio Baseado em Regras.

existe um módulo idealizado especialmente controle de coerência, que traz uma indesejável perda de performance ao sistema.

4.4 Discussão Comparativa do Trabalho

Este arcabouço, proposto nesta dissertação, pode ser comparado a seus trabalhos relacionados (Capítulo 3), particularmente por duas perspectivas: segundo o caráter evolutivo do negociador para as negociações automatizadas; e segundo sua função de ferramenta didática para disciplinas relacionadas à IA.

4.4.1 Modelo Para Negociações Automatizadas

Refletindo sobre os modelos mostrados na Seção 3.1, percebe-se que já existem modelos bem consolidados que tratam sobre o protocolo de negociação, os *modelos de negociação*[8], que dão suporte: a negociações com vários atributos, ultimato, comparação de acordos finalizados (através do estado de compromisso), modelo de decisão por análise estatística. Todas essas contribuições, fruto da evolução de várias pesquisas enfocando pontos específicos da negociação, em sua essência foram incorporadas nesta dissertação, adicionando-se a questão da aprendizagem de máquina.

Nos modelos baseados em funções determinísticas e estatísticas, a manipulação do conhecimento, contido nas informações em histórico, dá-se através de cálculos de probabilidades, estimativas, geração de funções de interpolação etc. Por sua vez, inteligência artificial acrescenta o dinamismo do comportamento de um vendedor real, seja na tomada de decisão empírica, baseada na experiência, julgamento ou intuição do negociador, como na elaboração de propostas que se adequam ao estado do negociador no ambiente. O modelo proposto nesta dissertação acrescenta ao agente negociador características de aprendizagem e tomada de decisão, baseadas nos conceitos e técnicas da IA.

[8]Referem-se aos tipos de negociação existentes, por exemplo: shoppings virtuais, leilões, vitrines virtuais. Mais detalhes no Apêndice A.3

4.4.2 Ferramenta de Auxílio Didático

O propósito do *Trading Agent Competition* (TAC) é incentivar a pesquisa e o desenvolvimento de sistemas para comércio automatizado em mercados eletrônicos. Utilizado mundialmente, é uma ferramenta didática que motiva os estudantes e pesquisadores a buscar novas estratégias vencedoras. O caráter acadêmico e didático também pode ser encontrado no modelo proposto nesta dissertação, onde estudantes das disciplinas relacionadas à IA podem testar seus conhecimentos, elaborando vendedores autônomos capazes de raciocinar e aprender (Mais detalhes sobre o TAC estão dispostos no Apêndice C).

4.4.3 Conclusão

Seja para construção de agentes negociadores inteligentes ou para estudo do comportamento das técnicas clássicas da Inteligência Artificial, o arcabouço aqui proposto tem o objetivo de facilitar essas implementações. Embora possam ser modificadas, as funcionalidades para as entidades atuantes no SMA já foram identificadas, o que também simplifica a construção de negociadores autônomos (principalmente quando em caráter didático).

Embora existam modelos bem fundamentados, que tratam a negociação como um todo, não foi identificado em nenhum deles a possibilidade da combinação de várias técnicas para aprendizagem de máquina, o que, com o advento desta dissertação, cria a necessidade de experimentos específicos para identificar a melhor técnica para cada característica inteligente no negociador.

Capítulo 5

Estudos de Caso

Nesse capítulo estão descritos dois estudos de caso utilizando o modelo apresentado no Capítulo 4. O primeiro estudo de caso é em um ambiente de negociação de pacotes turísticos, em que um dos negociadores, utilizando técnicas de IA, deve aprender durante as negociações. O segundo estudo de caso está no domínio dos leilões, em que agentes disputam em um pregão eletrônico, mas apenas um deles possui um modelo de aprendizagem definido.

5.1 Negociação com Pacotes Turísticos

Aplicando o arcabouço desenvolvido nesta dissertação ao domínio do Turismo, mais especificamente na venda de pacotes turísticos, foram elaborados dois negociadores diferentes: o primeiro, atua sempre como cliente, tendo um modelo baseado em uma função matemática (definida adiante) para a elaboração das propostas de negociação e instanciado com características baseadas em algumas regras de produção; o segundo negociador, atuando como vendedor, deverá implicitamente aprender sobre as regras utilizadas para elaboração das instâncias do primeiro negociador, e para isso utilizará algumas técnicas da IA, descritas ao longo desta seção.

Foi admitido, como simplificação para os experimentos, que cada pacote formado por variáveis de decisão do sistema (visto na Subseção 4.1.1) é constituído por *P = {Local, Transporte Ida, Transporte Volta, Estadia, Tempo Estadia, Valor}*, em que: *Local* representa o destino da viagem; *Transporte Ida* e *Transporte Volta* representam os meios de transporte para ida e volta, respectivamente; *Estadia* representa a qualidade/tipo do local de aco-

modação durante a estadia na cidade; *Tempo Estadia*, a duração da viagem no destino; e *Valor* representa o custo financeiro total para o pacote turístico.

5.1.1 Definição dos Negociadores

Foram definidos dois negociadores para o experimento no ambiente. O primeiro SMA negociador não aprende e trabalha gerando propostas com dados matemáticos e regras definidas a priori. Para o segundo SMA, seus agentes internos utilizam algumas das técnicas de aprendizagem apresentadas na Seção 2.2. Maiores detalhes sobre cada um desses agentes são apresentados a seguir:

Negociador 1

Seguindo o diagrama de colaboração sugerido na Figura 5.1, os agentes colaboradores assumirão as seguintes características:

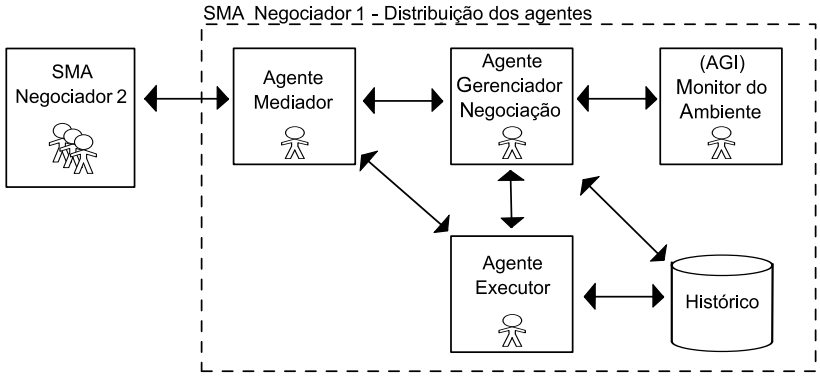

Figura 5.1: SMA Negociador 1 - Diagrama de colaboração

Agente Executor: elabora propostas segundo a metodologia adotada pelo *agente gerenciador da negociação* e baseado no tempo. A função polinomial, utilizada para o cálculo do valor de um atributo x na elaboração de uma proposta do agente a para o agente b, é dada por Faratin [FSJ98] e está representada na Equação 5.1.

$$x^t_{a\to b}[j] = \begin{cases} min^a_j + \alpha^a_j(t)\left(max^a_j - min^a_j\right) & Se\ V^a_j\ decresce \\ min^a_j + \left(1 - \alpha^a_j(t)\right)\left(max^a_j - min^a_j\right) & Se\ V^a_j\ cresce \end{cases} \tag{5.1}$$

onde α_j^a é a subfunção (Equação 5.2) que depende do tempo e sua imagem está no intervalo $[0, 1]$. V_j^a é a variação positiva ou negativa dos valores do atributo, que representa, por exemplo, se o agente está contra ou a favor do preço.

$$\alpha_j^a(t) = \kappa_j^a + \left(1 - \kappa_j^a\right)\left(\frac{min(t, t_{max})}{t_{max}}\right)^{\frac{1}{\beta}} \tag{5.2}$$

O coeficiente β (onde $\beta \in \mathbb{R}^+$), presente na Equação 5.2, representa um número infinito de estratégias e será determinado pelo Agente Gerenciador da Negociação.

Para avaliar as propostas recebidas, este agente utiliza a função utilidade definida em [FSJ02; KFSBY03], expressa pela Equação 3.1 (página 29). Se a uma nova proposta elaborada tiver utilidade inferior a proposta recebida, esta última é aceita.

Agente Gerenciador da Negociação: informará sempre a estratégia adequada para o momento na negociação, fornecendo valores para o índice β da Equação 5.2. Esse índice β pode ser visto como um representante da estratégia do negociador na elaboração das propostas, onde: valores próximos a zero caracterizam um negociador que não cede no inicio da negociação, mas se torna flexível quando o tempo da negociação está próximo ao fim; valores de β maiores que 1, caracterizam um negociador que cede mais em suas propostas durante o inicio da negociação do que no fim [FSJ98]. A Figura 5.2 ilustra influência do índice β para o valor de $\alpha(t)$ no tempo.

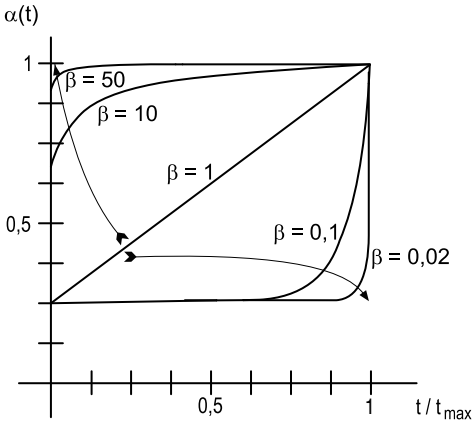

Figura 5.2: Função para o cálculo de $\alpha(t)$ [FSJ98]

O índice β é dado segundo algumas regras fixas de produção que dependem do tempo restante para negociação (*TempoRestante*), do tempo médio de uma interação (*Tem-*

poMedio), informações do *agente monitor do ambiente* e se recebeu um *ultimato* ou não. Como as instâncias deste SMA devem se comportar de forma diferente segundo suas características, para cada par *profissão-motivo da viagem* foram elaboradas regras diferentes. Um exemplo é o subconjunto das regras definidas para o par *professor-trabalho*:

- SE ultimato ENTAO Beta = Beta / 10
- SE (TempoRestante / TempoMedio > 5) ENTÃO Beta = 10
- SE (TempoRestante / TempoMedio <= 5) E (TempoRestante / TempoMedio >= 2) ENTÃO Beta = 2
- SE (TempoRestante / TempoMedio < 2) ENTÃO Ultimato
- SE (TempoRestante / TempoMedio < 2) ENTÃO Beta = 0,02

Os possíveis valores para *profissão* e *motivo da viagem* serão expostos ainda nesta seção.

Agente monitor do ambiente: É uma instância do AGI (Agente Gerador de Informação) que observa as mudanças dos preços e promoções de viagens, hotéis e cotação do dólar. Sua tarefa é buscar, a exemplo de *WebServices*[1], por esses valores no mercado e no ambiente, que para o nosso experimento, trata-se do acesso a valores fornecidos por um servidor único. Para cada informação valores são elaborados segundo uma função senoidal que utiliza o tempo como entrada, e devem ser ajustadas com os limites máximo e mínimo para o valor da variável.

$$valor = (\max \sin \tau - \min \sin \tau)/2 - \min \tag{5.3}$$

As variáveis de decisão e características particulares das instâncias do Negociador 1, que são geradas aleatoriamente, variam conforme a Tabela 5.1.

A instanciação dinâmica para este negociador obedece a algumas regras, as quais deverão ser implicitamente aprendidas ou mapeadas pelo Negociador 2 (Objetivo da simulação). Assim, a definição do perfil do negociador e suas preferências na negociação são feitas quando

[1]São componentes aplicacionais programáveis, em que a idéia genérica é a de um conjunto de operações/métodos que podem ser acessados remotamente.

Características e Preferências	Possíveis Variações
Profissão	professor, médico, engenheiro, advogado
Motivo da Viagem	férias, visita, trabalho
Limite Disponível (R$)	1000, 5000, 10000, 20000
Prazo de Pagamento	entre 1 e 12 meses
Dias para Início da Viagem	entre 1 e 120 dias
Transporte Ida	trem, ônibus, avião, navio
Transporte Volta	trem, ônibus, avião, navio
Estadia	pousada, hotel 3 estrelas, hotel 4 estrelas, hotel 5 estrelas

Tabela 5.1: Opções para os campos da proposta para o Negociador 1

o mesmo é criado, permanecendo fixas durante o ciclo de vida da instância. As regras utilizadas na criação do negociador não foram projetadas para serem mutuamente excludentes, podendo uma mesma instância do negociador obedecer a mais de uma. Essas regras estão descritas a seguir:

1. Se *MotivoViagem = [trabalho]* Então *Estadia = [hotel3Estrelas, hotel4Estrelas] e TransporteIda = [avião, onibus] e TransporteVolta = [avião, onibus] e DiasInicioViagem = [2–15]*
2. Se *MotivoViagem = [Ferias]* Então *Estadia = [pousada, hotel5Estrelas] e TransporteIda = [navio, avião, trem] e DiasInicioViagem = [30–90]*
3. Se *Profissao = [advogado, professor]* Então *LimiteDisponivel = [5000] e TransporteIda = [aviao, onibus] e DiasInicioViagem = [5–30]*
4. Se *Profissao = [medico, engenheiro] e MotivoViagem = [visita]* Então *LimiteDisponivel = [5000,10000,20000] e Estadia = [pousada, hotel simples] e TransporteIda = [avião]*
5. Se *Profissao = [medico, engenheiro] e MotivoViagem = [ferias]* Então *LimiteDisponivel = [10000,20000] e Estadia = [hotel4Estrelas, hotel5Estrelas] e TransporteIda = [navio, avião, trem] e TransporteVolta = [navio, avião, trem]*

Negociador 2

Seguindo o diagrama de colaboração sugerido na Figura 5.3, os agentes colaboradores assumirão as seguintes características:

Agente Gerenciador da Negociação: informará sempre a estratégia com maior chance de sucesso, ou mais adequada para o momento na negociação, fornecendo valores para o índice β da Equação 5.2, assim como o Negociador 1. Porém, aqui será utilizado o algoritmo Q-learning (visto na Subseção 2.2.4, página 22) para aprender sobre o melhor

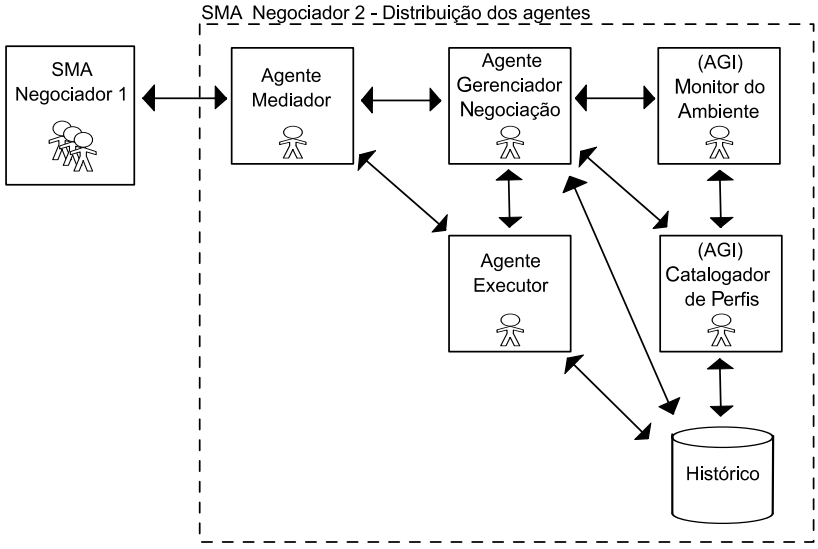

Figura 5.3: SMA Negociador 2 - Diagrama de colaboração

valor de β (agora discretizado e não mais envolvendo todos os valores de $\mathbb{R}^+$) associado ao estado atual da negociação (tempo restante, tempo da negociação, informações dos agentes *Monitor do Ambiente* e *Catalogador de Perfis*).

Para o experimento, os valores de recompensa e punição para o algoritmo Q-learning têm, respectivamente, uma proporção de 3 para 1. A justificativa para isso é que deseja-se desincentivar uma ação (punição) a uma velocidade menor (3 vezes menor) do que incentiva-la, para que o algoritmo não desaprenda facilmente conforme as perturbações no comportamento do ambiente. Na literatura [Tes04; Fon01; FR99], geralmente são adotados valores menores para a punição no algoritmo, porém a relação proposta foi intuitiva e caberiam aqui testes exaustivos para a escolha da melhor configuração, mas isto foge ao escopo desta dissertação.

Agente Monitor do Ambiente: instância do AGI (Agente Gerador de Informação) que observa as mudanças dos preços e promoções de viagens, hotéis e cotação do dólar, da mesma forma que no Negociador 1.

Agente Catalogador de Perfis: outra instância do AGI que agrupa os usuários considerando vários aspectos utilizando Redes Neurais para a aprendizagem (como visto na Subseção 2.2.2), com 16 neurônios na camada de saída e 5 na camada de entrada. A escolha desses números foi arbitrária, uma vez que não é objetivo do experimento a

descoberta da melhor configuração possível, interessando apenas a caracterização do "aprender". Para o treinamento da rede neural do tipo SOM foram utilizados os valores de identificação na instanciação do agente (informações estas fornecidas através do *DF Agent*), o que inclui as características *profissão* e *motivo da viagem,* em conjunto com informações de acordos relacionados em histórico (apenas o preço, estadia e transporte). Após a aprendizagem este agente deve enviar uma informação de classificação para o Agente Gerenciador da Negociação, que a utiliza como variável que define o estado da negociação. Como será visto na subseção seguinte, foi utilizado um conjunto de treinamento com 2000 casos. O critério de parada para o treinamento foi cumprimento de 100 ciclos de treinamento com o mesmo conjunto.

Agente Executor: Elabora propostas segundo a metodologia adotada pelo agente de estratégias, também conforme a função de Faratin [FSJ98], vista na Equação 5.1 . São elaboradas várias propostas através de pequenas variações (entre 1% e 10%) do coeficiente β recebido. Porém, algumas dessas propostas vêm do algoritmo de RBC (visto na Subseção 2.2.3, página 20), que toma como base a experiência em histórico, utilizando algumas características (β & profissão & tempo restante & índice de variação das propostas vindas do negociador) que definem o estado do ambiente como chave de busca. A escolha da proposta para o envio, dentre todas, é feita através de testes de similaridade conforme a função da Equação 5.4, definida por [OSBC05]. Para avaliar as propostas recebidas, este agente utiliza a função utilidade definida em [FSJ02; KFSBY03], expressa pela Equação 3.1 (página 29). Se a uma nova proposta elaborada tiver utilidade inferior a proposta recebida, esta última é aceita.

A função que calcula o similaridade entre dois atributos r_i, s_i é dada por: $sim(r_i, s_i) = 1 - |h_i(r_i) - h_i(s_i)| \in [0, 1]$, onde $h_i()$ é a função de similaridade para o i-ésimo atributo que retorna valores no intervalo $[0, 1]$. Agora, a função $SIM()$, que calcula similaridade entre propostas, é definida como:

$$SIM(R, S) = \sum(w_i * sim(r_i, s_i)) \tag{5.4}$$

onde w_i é o peso pré-ajustado para cada atributo da negociação e $0 \leq SIM(R, S) \leq 1$.

5.1.2 Descrição e Resultados do Experimento

Com os negociadores descritos acima, inicialmente foram realizadas 2000 negociações, com instâncias diferentes do Negociador 1 e utilizando o mesmo Negociador 2 (para aquisição de experiência). Em um segundo momento, após o treinamento do negociador 2, mais 500 negociações para avaliação. Em todas as 2500 negociações, o tempo limite para fechamento do negócio foi de 1 min, e ocorreu uma por vez. Os testes foram realizados com 2 computadores, ambos com a seguinte configuração: AMD Athlon 2.08Ghz, 512RAM, sistema operacional Windows XP, em uma rede de barramento 10 Mbps (*MegaBits Per Second*). Os resultados são discutidos em seguida.

A priori, comparando os 2000 primeiros negociadores, criados aleatoriamente, com as 5 regras utilizadas na criação do Negociador 1, foi identificado que: 12% dos negociadores criados enquadram-se na primeira regra; 9%, na segunda; 22%, na terceira; 7%, na quarta; e, 11%, na última regra (Os outros 39% não se enquadraram em nenhuma regra).

O resultado, derivado principalmente da análise das últimas 500 negociações (após o período de treinamento), pode ser resumido na Tabela 5.2. Essa tabela mostra a diminuição do tempo médio por negociação no segundo momento, que pode ser vista como uma contribuição da aprendizagem, mesmo com o tempo extra previsto para elaboração das propostas e algoritmos de tomada de decisão.

	Momento 1	Momento 2
Número médio de iterações por negociação	4,8	3.4
Tempo médio por negociação	24283ms	18217ms
Negociações finalizadas pelo limite de tempo	29	7
Negociações concluídas com oponentes $\in$ regra 1	92%	94%
Negociações concluídas com oponentes $\in$ regra 2	95%	100%
Negociações concluídas com oponentes $\in$ regra 3	100%	100%
Negociações concluídas com oponentes $\in$ regra 4	84%	93%
Negociações concluídas com oponentes $\in$ regra 5	88%	97%

Tabela 5.2: Resumo dos resultados dos testes

Como mostra a Tabela 5.2, o tempo médio de negociação diminuiu no momento 2, mostrando conseqüentemente que a convergência para um acordo foi mais rápida. Na Figura 5.4, o gráfico da média das utilidades no tempo, ilustra esse ganho na convergência.

Na Figura 5.5, é mostrada uma aproximação média do número de acordos realizados por

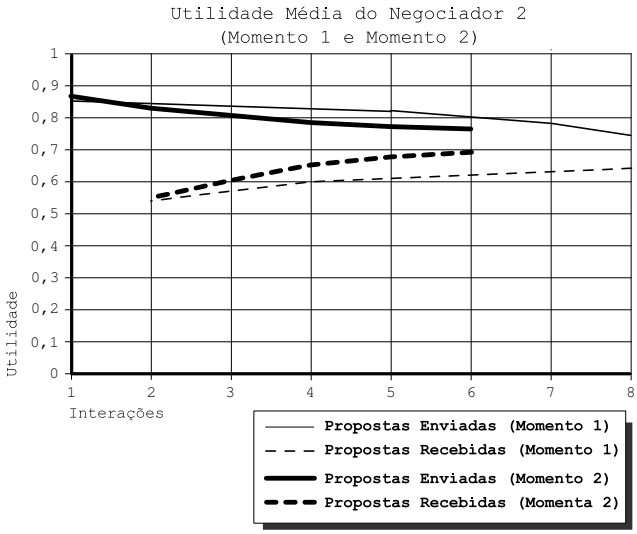

Figura 5.4: Comparação da convergência entre os cenários

interação. A linha pontilhada representa o primeiro momento do experimento, onde a concentração dos acordos fechados está na quinta interação. O segundo momento, representado pela linha tracejada, tem a maioria dos acordos realizados na terceira interação.

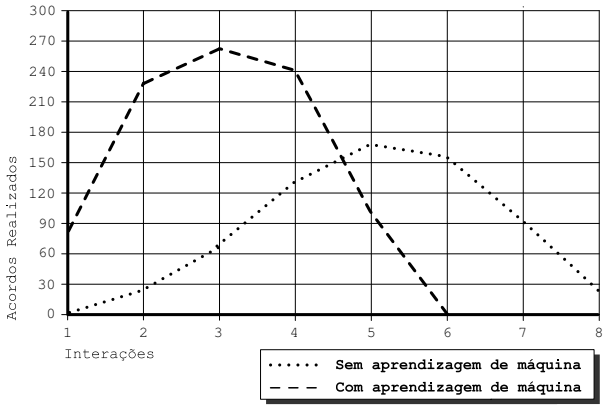

Figura 5.5: Acordos por ciclo de interação

As diferenças entre as duas partes da experiência trouxe uma aproximação para um negociador inteligente através do Negociador 2. Como o Negociador 1 não possuía um modelo de aprendizagem definido, e sempre eram tomadas novas instâncias dele a cada negociação, não foi observado nenhuma evolução. O objetivo da experiência, que era mostrar o arcabouço e a aprendizagem influenciando significativamente o processo da negociação, foi alcançado. Uma pesquisa exaustiva à parte deveria ser feita, tanto para o ajuste das técnicas utilizadas, como para compará-las com outras, mas foge do escopo do experimento. Considerando

as alterações proporcionadas, mesmo com um modelo de aprendizagem passível de aprimoramentos, pode-se considerar que as técnicas da IA contribuíram significativamente no experimento e portanto podem atuar também em um cenário real.

Esse experimento também ilustra a idéia da utilização do arcabouço como ferramenta didática para estudos na Inteligência Artificial. Como sugestão para isso, pode-se pedir que cada um dos aprendizes construa um agente negociador e que se faça uma comparação prática para (ou competição) testa-los entre si.

5.2 Negociação em Pregão Eletrônico

Algumas empresas e, em especial, os órgãos da administração pública[2] quando fazem contratos[3] com terceiros, costumam se submeter ao procedimento administrativo prévio de licitação. As empresas públicas o fazem conforme a legislação em vigor. Desta forma, licitação pode ser definida como sendo um procedimento administrativo destinado a selecionar, entre outras, a proposta mais vantajosa para a administração de uma empresa.

As empresas e administrações públicas possuem seus processos de compra regulamentados pela Lei no 8.666/93, também chamada de Lei de Licitações. Esta lei estabelece as situações em que as compras realizadas por entidades governamentais são ou não realizadas através de processos licitatório. Em sua redação, no artigo 22, a lei 8.666/93 estabelece, dentre outras normas gerais, cinco possíveis modalidades de licitação: Concorrência, Tomada de Preços, Carta Convite, Concurso, Leilão e Pregão. Este último foi escolhido como um dos estudos de caso deste trabalho.

5.2.1 Descrição do Pregão Eletrônico

Esta simulação, no processo de negociação, está na modalidade de licitação pregão. Sendo assim, dois tipos de SMA negociadores foram identificados para esta modalidade de licitação: o agente pregoeiro e os agentes fornecedores. O pregoeiro faz o papel de representante do governo no processo de compra, é ele quem determina qual o melhor fornecedor

[2]No âmbito Federal, Estadual, e Municipal e demais entidades controladas direta ou indiretamente pela União, Estados, Distrito Federal e Municípios

[3]Contratos de qualquer natureza: obras, serviços, compras, alienações, permissões e locações

para fazer a compra, onde, o único critério de avaliação considerado foi o preço. A Figura 5.6 representa a descrição dos participantes em uma negociação em pregão eletrônico, cuja seqüência de negociação, representada na Figura 5.7, está detalhada a seguir:

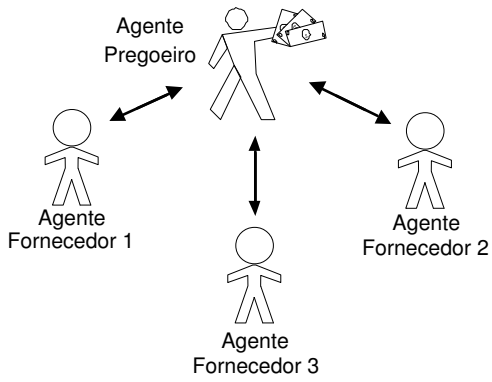

Figura 5.6: Pregão Eletrônico

1. **Resgatar lista de Fornecedores.** Em um primeiro momento, o pregoeiro solicita ao *DF Agent*[4] a lista de possíveis fornecedores existentes no domínio de aplicação do pregoeiro. Cabendo a cada agente fornecedor fazer o seu próprio cadastro junto ao agente *DF Agent*.

2. **Envio do Edital.** Depois de estar de posse dos possíveis fornecedores, o agente pregoeiro comunica a todos os fornecedores existentes, através do envio de cópias do Edital, que está pronto para receber os envelopes relacionadas ao respectivo Edital. Para o escopo deste trabalho, as únicas informações contidas no Edital são o tempo em que o agente pregoeiro estará disponível a receber envelopes[5] e a descrição do objeto da negociação (como na Subseção 4.1.1).

3. **Envio de Envelopes.** Os fornecedores enviam as suas respectivas propostas iniciais (envelopes) para o pregoeiro, contendo o preço inicial. O fornecedor que não enviar seu envelope proposta dentro do prazo especificado no Edital, estará automaticamente eliminado do processo de licitação.

[4]Agente do Jade que fornece o serviço de páginas amarelas

[5]Envelopes são caracterizados por propostas lacradas e enviadas por um fornecedor, de forma que apenas o pregoeiro tem o conhecimento da existência deste.

4. **Fechamento de Período de Recebimento de Envelope.** O pregoeiro escolhe, dentre os vários fornecedores que enviaram seus envelopes, apenas os três que possuem os menores preços. Em seguida, é enviado a cada um dos agentes fornecedores contemplados, um aviso indicando o inicio do pregão. Aos demais, o pregoeiro envia-lhes uma mensagem comunicando que não foram selecionados. Os fornecedores também são informados sobre qual foi a menor proposta recebida.

5. **Pregão.** Quando o pregão é iniciado, os três agentes fornecedores que participam do pregão enviam suas propostas. Paralelamente, o pregoeiro vai selecionando um fornecedor como vencedor e informando aos demais participantes sobre o lance vencedor.

6. **Finalização.** Em cumprimento do Edital, pregão é finalizado quando chega ao término do tempo pré-definido pelo pregoeiro. Neste momento, o pregoeiro comunica qual o fornecedor vencedor e qual o valor da proposta vencedora.

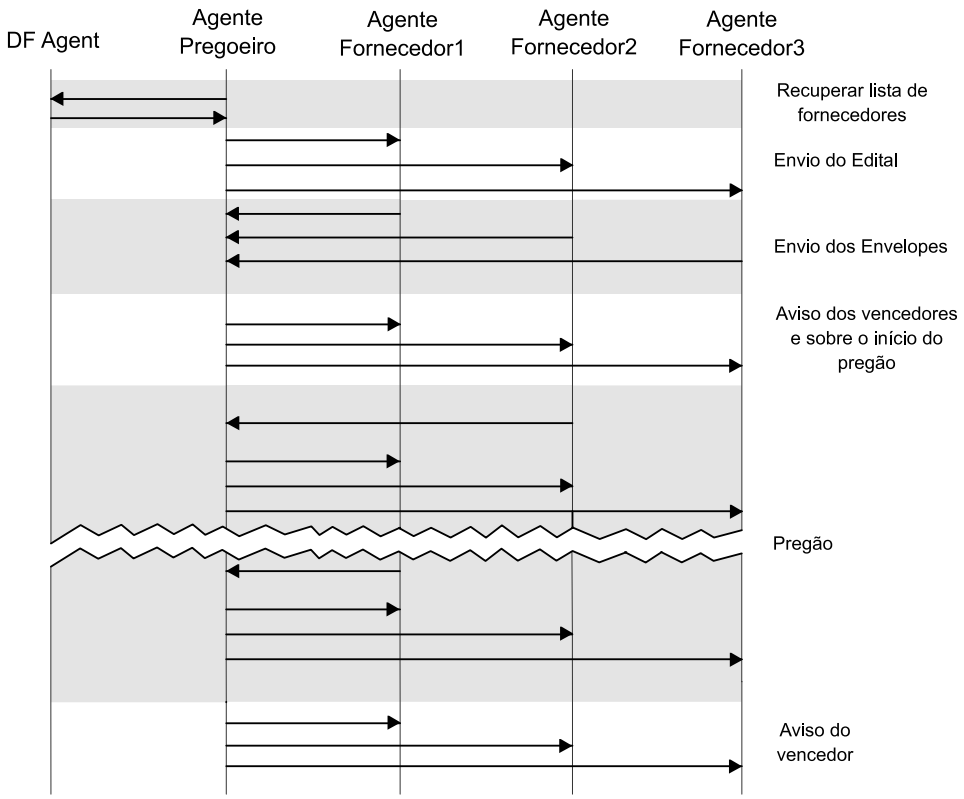

Figura 5.7: Sequência do Pregão Eletrônico

5.2.2 Negociadores

Os negociadores foram implementados a partir do arcabouço apresentado no Capítulo 4. Para este experimento, existem três tipos de negociadores. Um deles é o leiloeiro que se comportará como representante do bem leiloado (vendedor). Os participantes ativos do leilão (os que oferecem os lances) podem ser de dois tipos diferentes, um deles se utiliza de técnicas de aprendizagem.

SMA Pregoeiro

Os agentes que compõem o SMA Pregoeiro são instâncias dos agentes mediador e gerenciador de estratégia, todos definidos no no Capítulo 4. As particularidades sobre cada um são discutidas abaixo:

Agente Gerenciador da Negociação (AGN): Controla o protocolo de negociação, que difere um pouco do protocolo para negociação descrito na Subseção 4.1.3. Neste tipo de negociação existem duas etapas: uma para a seleção dos concorrentes no leilão; e outra para o leilão propriamente dito.

Na primeira etapa, o Agente Gerenciador da Negociação estipula o tempo máximo para recebimento dos envelopes. Assim, o *TempoEdital* corresponde ao valor em milisegundos que o pregoeiro espera, após enviar o Edital, pelo recebimento das propostas iniciais dos negociadores.

Após verificar quais os três menores valores de propostas recebidas (até então secretas), o leiloeiro divulga quem continua na negociação e quem não. Dá-se início a segunda etapa do leilão com o tempo *TempoPregao* (em milisegundos), onde os negociadores do pregão tentarão em competição dar o seu melhor lance (que corresponde ao menor preço). Para segurança, o pregoeiro só deve aceitar propostas abaixo de um valor de reserva definido por *PrecoMaximo*.

Agente Mediador (AM): Conforme definido na Subseção 4.2.1, é o agente responsável pela comunicação e interpretação das mensagens. Este agente sofreu alguns ajustes para se adequar ao tipo de negociação proposto neste experimento.

SMA Fornecedor 1

Para este negociador, instanciado do arcabouço da Seção 4.2, não serão atribuídas características de aprendizagem e raciocínio lógico. Suas decisões serão tomadas como descrito:

Agente Executor: elabora propostas segundo a metodologia adotada pelo *agente gerenciador da negociação* e baseado no tempo. Uma função polinomial, utilizada para o cálculo do valor do atributo $preco$ na elaboração de uma proposta do agente a (Fornecedor) para o agente b (Pregoeiro), foi derivada da função de Faratin [FSJ98] e está representada na Equação 5.5.

$$preco_{a \to b}[j] = min_j^a + \alpha_j^a(t)\left(max_j^a - min_j^a\right) \tag{5.5}$$

onde α_j^a é a subfunção (Equação 5.6) que depende do tempo e sua imagem está no intervalo $[0, 1]$.

$$\alpha_j^a(t) = \kappa_j^a + \left(1 - \kappa_j^a\right)\left(\frac{min(t, t_{max})}{t_{max}}\right)^{\frac{1}{\beta}} \tag{5.6}$$

O coeficiente β (onde $\beta \in \mathbb{R}^+$), presente na Equação 5.6, representa um número infinito de estratégias e será determinado pelo Agente Gerenciador da Negociação.

Agente Gerenciador da Negociação (AGN): Conta com um ajuste prévio sobre o menor preço possível para a oferta, através de *PrecoMinimo* Da mesma forma que no SMA pregoeiro, este agente controla o protocolo de negociação, que difere um pouco do protocolo para negociação descrito na Subseção 4.1.3.

Este agente informa sempre a estratégia com maior chance de sucesso, ou mais adequada para o momento na negociação, fornecendo valores para o índice β da Equação 5.6. O índice β representa a estratégia de negociação do SMA negociador, onde: valores próximos a zero caracterizam um negociador que não cede no inicio da negociação, mas se torna flexível quando o tempo da negociação está próximo ao fim; valores de β maiores que 1, caracterizam um negociador que cede durante o inicio da negociação, e cada vez menos até o fim [FSJ98]. A Figura 5.2 (página 68) ilustra influência do índice β para o valor de $\alpha(t)$ no tempo.

O índice β é dado segundo algumas regras fixas de produção que dependem do tempo restante para o pregão (*TempoRestante*), do tempo médio de resposta dos outros fornecedores (*TempoMedio1* e *TempoMedio2*). Como as instâncias deste SMA devem se comportar de forma diferente, existem 10 conjuntos diferentes de regras que definem esse comportamento. Um exemplo é o conjunto das regras que definem um negociador que espera um último momento para lances maiores (Os demais conjuntos de regras utilizados estão dispostos no Apêncice D).

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 5) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 5) ENTÃO Beta = 0,1
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 1

SMA Fornecedor 2

Para este negociador, também instanciado do arcabouço da Seção 4.2, está atribuído um modelo de aprendizagem por reforço para elaboração da nova proposta que conta com a ajuda da previsão das próximas propostas dos seus oponentes, com Redes Neurais.

Agente Gerenciador da Negociação (AGN): Da mesma forma que no SMA pregoeiro, este agente controla o protocolo de negociação, que difere um pouco do protocolo para negociação descrito na Subseção 4.1.3. Como no SMA pregoeiro, também existe um ajuste sobre o menor preço possível para a oferta, expresso na variável *PrecoMinimo*, realizado antes negociação.

Este agente ainda decide qual a melhor estratégia, através do índice β, que influencia a elaboração de propostas. A estratégia é definida utilizado um algoritmo de aprendizagem por reforço (visto na Subseção 2.2.4, página 22) para aprender sobre o melhor valor de β (agora discretizado e não mais envolvendo todos os valores de $\mathbb{R}^+$) associado ao estado atual da negociação (tempo restante e decorrido do pregão, e informações sobre os oponentes).

O valor inicial para β é 1, e permanece até a primeira metade do tempo total da negociação e uma análise dos outros negociadores. Da mesma forma que na simulação anterior, os índices para recompensa e punição do algoritmo Q-learning são arbitrários,

isso porque seria necessário um estudo mais profundo sobre a situação/problema para ajustá-los. Foram escolhidos valores proporcionais a 10 e 1, respectivamente, para recompensa e punição. Apenas é punida a estratégia final (a que não consegue vencer o leilão), e nos outros momentos os valores para recompensa e punição são iguais a zero.

Agente Executor: elabora propostas segundo a metodologia adotada pelo *agente gerenciador da negociação* e baseado no tempo. Assim como no *SMA Fornecedor 1*, uma função polinomial foi utilizada para o cálculo do valor do atributo $preco$ na elaboração de uma proposta do agente a (Fornecedor) para o agente b (Pregoeiro), que derivada da função de Faratin [FSJ98] e está representada na Equação 5.5. O coeficiente β (onde $\beta \in \mathbb{R}^+$), presente na Equação 5.6, representa um número finito de estratégias e será determinado pelo Agente Gerenciador da Negociação.

Agente Previsão de Valores: instância do AGI (Agente Gerador de Informação) que infere sobre os valores das próximas propostas dos negociadores. Para isto, foi considerado que os decrementos das seqüências de valores nas propostas são séries temporais, e portanto podem ser mapeados por uma Rede Neural (como visto na Subseção 2.2.2). A Rede Neural possui 3 neurônios de entrada, 1 neurônio na camada de saída e duas camadas intermediárias com 4 neurônios cada. Mais uma vez, cabe o comentário justificando que os números de neurônios para essas camadas da rede e o número de camadas intermediárias foram arbitrários, uma vez que não foi objetivo do experimento o ajuste ideal da rede, e sim apenas a caracterização da aprendizagem. Para o treinamento da rede neural, foi utilizado um conjunto de negociações ocorridas em um primeiro momento na simulação. O tamanho do conjunto de treinamento foi de 2000 itens (obtidos das 1000 negociações com duas instâncias do Fornecedor 1 em cada, como será visto na subseção seguinte) e a taxa de tolerância de erro estipulada para o treinamento foi de $0,05$, que para atingi-la necessitou-se de cerca de 350 ciclos de treinamento.

5.2.3 Descrição e Resultados do Experimento

Com os negociadores descritos acima, inicialmente foram realizadas 1000 leilões de pregão eletrônico, sempre com duas instâncias diferentes do Fornecedor 1 e utilizando uma mesma instância do Fornecedor 2, para aquisição de experiência. Em um segundo momento, após o treinamento do negociador 2, mais 500 negociações para avaliação. Em todas as 1500 negociações, os tempos limite para fechamento do Edital e pregão foram de 15seg e 2min, respectivamente, e ocorreu um pregão por vez. Os testes foram realizados com 2 computadores, ambos com a seguinte configuração: AMD Athlon 2.08Ghz, 512RAM, sistema operacional Windows XP, em uma rede de barramento 10 Mbps (*MegaBits Per Second*). Os SMAs ficaram distribuídos da seguinte forma: Pregoeiro e uma das instâncias do Fornecedor 1, em uma máquina, e a outra instância do Fornecedor 1 e o Fornecedor 2, na segunda máquina. Os resultados são discutidos em seguida.

De forma geral, o Fornecedor 2 obteve sucesso, vencendo na primeira etapa do experimento (fase de aprendizagem inicial, sem a rede neural e com o Q-learning se adaptando) cerca de 45% dos pregões. Na segunda etapa, venceu cerca de 65% das negociações, onde o Q-learning continuou se ajustando pelo adicionamento das previsões das propostas através da Rede Neural.

A tática utilizada pelos fornecedores, instâncias do Fornecedor 1, não utiliza a constante postagem de lances no primeiro momento da negociação, que se intensifica apenas quando o tempo restante permite poucos lances. Com isso, a aprendizagem do *Agente Previsão de Valores* ficou prejudicada pela baixa amostragem, motivo pelo qual a rede possui apenas 3 neurônios na camada de entrada.

Considerando os índices de ajuste para valores de recompensa e punição do Q-learning deficientes, o algoritmo conseguiu mapear os 10 comportamentos diferentes nas instâncias do SMA Fornecedor 1. A Rede Neural também conseguiu uma aproximação razoável, com um erro médio de aproximadamente 20% nas suas previsões de valores, o que pode ser atribuído à baixa qualidade na amostragem ou ainda a um número baixo de neurônios nas camadas da rede.

Capítulo 6

Considerações Finais

Nesta dissertação foram apresentados os resultados originados de uma pesquisa envolvida com a aprendizagem na negociação automatizada. A partir dos estudos sobre trabalhos relacionados com este tema, foi proposta uma solução que provê suporte à tomada de decisão destes agentes no processo de negociação.

O modelo de negociação apresentado mostra novas características sobre a aprendizagem e tomada de decisão em relação ao estado-da-arte do tema em pauta, as quais buscam enriquecer o processo de negociação. Desta forma, novas opções foram introduzidas na negociação, visando torná-la mais próxima daquela praticada no comércio tradicional e possibilitando o aumento da satisfação dos usuários (consumidores e fornecedores) com o seu resultado final.

O modelo de suporte à tomada de decisão apresenta uma visão diferenciada da negociação automatizada, ao considerar os dados presentes no histórico e o próprio ambiente do negociador em busca de prover informações relevantes para a tomada de decisão na negociação. Esta visão é um avanço significativo em relação às demais pesquisas neste tema, e os principais benefícios trazidos pela presente proposta são:

- possibilidade da tomada de decisão mais robusta por parte dos agentes negociadores, seja desempenhando o papel de cliente, seja como de fornecedor;
- maior aproximação entre a tomada de decisão praticada na negociação automatizada e a negociação no mundo real;

- possibilidade da personalização das ofertas que um negociador emite ao outro, inclusive com também tratando-o de forma diferenciada como negociador;
- busca de maior satisfação com a negociação, pela possibilidade de acordos mais interessantes para ambas as partes;

A partir dos modelos de negociação e suporte à tomada de decisão propostos, uma plataforma multiagente também foi apresentada com o propósito de ser um arcabouço para elaboração de sistemas multiagente negociadores, que inclui um conjunto das principais técnicas para aprendizagem e tomada de decisão.

A partir da infra-estrutura definida com o arcabouço, um negociador é então capaz de definir seu conhecimento, modelo de aprendizagem e tomada de decisão sobre a negociação de produtos, de forma que esse conhecimento possa ser recebido/adquirido e utilizado na negociação. Foi discutido também, a exemplo do TAC, que é plausível a utilização do produto deste trabalho em sala de aula, como ferramenta didática auxiliadora nas disciplinas relacionadas à Inteligência Artificial. Enfim, como discutido ao final de cada experimento no Capítulo 5, ficou demonstrado a capacidade de aprendizagem dos agentes.

6.1 Trabalhos Futuros

Vários trabalhos futuros podem ser apontados para o enriquecimento deste trabalho e da negociação automatizada. Um primeiro trabalho, considerado fundamental, seria um esforço em conjunto com outras disciplinas, como administração e economia, para o desenvolvimento do conhecimento sobre a negociação de produtos. Esse conhecimento poderia ser geral o bastante, de forma a ser utilizado em diversos domínios de negócios, ao mesmo tempo em que também poderia permitir o negociador fazer pequenos ajustes, a fim de se encaixar melhor em suas necessidades. Um ambiente que facilite a construção e edição desses negociadores, assim como também para edição do conhecimento sobre a negociação o que pode ser apontado como outro trabalho futuro.

A partir do momento que se possa contar com um conhecimento real sobre a negociação, uma avaliação sobre as técnicas mais adequadas para diversos pontos na negociação, incluindo ajustes e inclusão de novos métodos de aprendizagem, são importantes no sentido

de se obter de um negociador bem mais eficiente. Como o sistema verifica as condições do ambiente e pela importância em não se comprometer o modelo de aprendizagem e tomada de decisão do negociador, um tratamento sobre a consistência e qualidade dessas informações também pode ser apontado como um trabalho futuro.

A extensão/adaptação do arcabouço para outros fins, além da negociação automatizada, também é um trabalho futuro que se mostra interessante e promissor. Um primeiro caso seria a sua utilização para fazer simulações. Por exemplo, um negociador poderia inserir dados fictícios sobre o ambiente, e observar o impacto desses dados em suas negociações. Outrossim, novas estratégias para negociação também poderiam ser testadas através de simulações no sistema.

Como a proposta da plataforma é oferecer um suporte de decisão e aprendizagem, especificamente para os agentes negociadores, isto poderia ser ampliado, de forma a oferecer suporte de decisão para os funcionários de uma empresa. Também, de uma forma mais simples, a ferramenta também poderia prover entidades de apoio a decisão, por exemplo, para prover alertas a uma empresa quando determinadas condições no ambiente fossem encontradas.

Bibliografia

[Abe01] Mara Abel. *Estudo da Perícia em Petrografia Sedimentar e sua Importância para a Engenharia de Conhecimento*. (tese de doutorado), Universidade Federal do Rio Grande do Sul, 2001.

[ABO00] Fernando M. Azevedo, Lourdes M. Brasil, and Roberto Célio L. Oliveira. *Redes Neurais com Aplicações em Controles e em Sistemas Especialistas*. Bookstore, 2 edition, 2000.

[Alm04] Hyggo Oliveira Almeida. Compor - desenvolvimento de software para sistemas multiagentes. Master's thesis, Universidade Federal de Campina Grande, FEV 2004.

[Alm05] Rodrigo Rebouças Almeida. *Uma Arquitetura de Software para Arranjos Produtivos Locais*. (dissertação de mestrado), Universidade Federal de Campina Grande, 2005.

[ALN+03] Mazda Ahmadi, Abolfazl Keighobadi Lamjiri, Mayssam M. Nevisi, Jafar Habibi, and Kambiz Badie. Using a two-layered case-based reasoning for prediction in soccer coach. In *MLMTA*, pages 181–185, 2003.

[Ban06] Origem e evolução da moeda. Banco Central do Brasil, fev 2006. http://www.bcb.gov.br/?origemoeda.

[BPJ02] C Bartolini, C Preist, and N R Jennings. Architecting for reuse: A software framework for automated negotiation. In *Third International Workshop on Agent Oriented Software Engineering*, pages 87–98, 2002.

[BPJ04] Claudio Bartolini, Chris Preist, and Nicholas R. Jennings. A Software Framework for Automated Negotiation. In *Software Engineering for Large-Scale Multi-Agent Systems*, volume 3390 of *Lecture Notes in Computer Science*, pages 213–235. Springer-Verlag, 2004.

[BR00] Steffen Brückner and Stephan Rudolph. Neural networks applied to smart structure control. In *SPIE Conference On Application and Science of Computational Intelligence III*, volume 1, Orlando - Florida, April 2000.

[BRC04] Reinaldo A. C. Bianchi, Carlos H. C. Ribeiro, and Anna H. Reali Costa. Heuristically accelerated q-learning: A new approach to speed up reinforcement learning. In Ana L. C. Bazzan and Sofiane Labidi, editors, *17th Brazilian Symposium on Artificial Intelligence*, volume 3171 of *Lecture Notes in Computer Science*, pages 245–254. Springer, 2004.

[BVlFM01] Rafael Heitor Bordini, Renata Vieira, and Álvaro Freitas Moreira. Fundamentos de sistemas multiagentes. In *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001)*, volume 2, pages 3–41, 2001.

[Cas03] P. A. L. Castro. *Uma Infra-Estrtutura para Agentes Arrematantes em Múltiplos Leilões Simultâneos*. Dissertação de mestrado, USP, São Paulo, 2003.

[CBS+00] Pádraig Cunningham, Ralph Bergmann, Sascha Schmitt, Ralph Traphöner, Sean Breen, and Barry Smyth. Websell: Intelligent sales assistants for the world wide web. (Relatório Técnico) SC42, Trinity College Dublin, Ireland, November 2000.

[CBS+01] Pádraig Cunningham, Ralph Bergmann, Sascha Schmitt, Ralph Traphöner, Sean Breen, and Barry Smyth. Intelligent support for online sales: The websell experience. In R. Weber and C.G. Wangenheim, editors, *Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning (ICCBR)*. Springer-Verlag, 2001.

[CGH97] Enrique Catillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, 1997.

[CM96] Anthony Chavez and Pattie Maes. Kasbah: An Agent Marketplace for Buying and Selling Goods. In *First International Conference on the Pratical Application of Intelligent Agents and Multi-Agent Technology*, pages 75–90, London, UK, 1996.

[Cun98] Padraig Cunningham. CBR: Strengths and weaknesses. In *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 2, pages 517–524. Springer-Verlag, 1998.

[Deb03] John K. Debenham. An eNegotiation Framework. In *Australian Conference on Artificial Intelligence*, pages 833–846, 2003.

[Dev01] Vladan Devedzic. Knowledge modeling - state of the art. In *Integrated Computer-Aided Engineering 8*, volume 8, 2001.

[Dua05] E. W. Duarte. *Um agente atuante em leilões simultâneos visando a aquisição de itens interdependentes*. Monografia de graduação, PUC-Rio, 2005.

[Far00] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. (phd thesis), University of London, London, 2000.

[Fer99] Aurélio Buarque Holanda Ferreira. *Novo dicionário Aurélio - Século XXI*. Editora Nova Fronteira, 3 edition, 1999.

[FH] Ernest Friedman-Hill. Sandia National Laboratories - Jess Web Site, http://herzberg.ca.sandia.gov - acessado em 20/01/2006.

[Fil04] Ricardo Rubens G. Nunes Filho. *Um Modelo de Sistema Multi-Agente para Negociação Automatizada no Comércio Eletrônico: Foco na Empresa.* (dissertação de mestrado), Universidade Federal de Campina Grande, 2004.

[FIP06] FIPA. Foundation for intelligent physical agents. http://www.fipa.org, 2006.

[FLM97] T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. In J. Bradshaw, editor, *Software Agents*. MIT Press, Cambridge, MA, 1997.

[Fon01] José Manuel Matos Ribeiro Fonseca. *Protocolos de Negociação com Coligações em Sistemas Multi-agente*. (tese de doutorado), Universidade Nova de Lisboa, 2001.

[FR99] Gelson Faria and Roseli Francelin Romero. Explorando o potencial de algoritmos de aprendizado com reforço em robês móveis. In *IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais*, volume 1, pages 237–242, July 1999.

[FSJ98] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation Decision Functions for Autonomous Agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[FSJ02] P. Faratin, C. Sierra, and N. R. Jennings. Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

[FWJ04] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An Agenda Based Framework for Multi-issues Negotiation. *Artificial Intelligence*, 152(1):1–45, January 2004.

[GLS01] A. C. B. Garcia, A. Lopes, and C. B. Seidel. AGILE: Intelligent Agents to Assits Electronic Auction. In *International Conference on Enterprise Information Systems - ICEIS (2)*, pages 1174–1178, 2001.

[Hay01] Simon Haykin. *Redes Neurais: Princípios e Prática*. Bookman, 2 edition, 2001.

[HJL03] Minghua He, Nicholas R. Jennings, and Ho-Fung Leung. On Agent-Mediated Electronic Commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003, July/August 2003.

[IOOY01] Hiroya Inakoshi, Seishi Okamoto, Yuiko Ohta, and Nobuhiro Yugami. Effective decision support for product configuration by using cbr. In *International Conference on Case-Based Reasoning: Workshop CBR in E-Commerce*, Vancouver BC, July 2001.

[JAD] JADE Web Site, http://jade.tilab.com/ - acessado em 02/02/2006.

[Jen99] Nicholas R. Jennings. Agent-Oriented Software Engineering. In Francisco J. Garijo and Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, pages 1–7. Springer-Verlag: Heidelberg, Germany, 30– 2 1999.

[JMCN02] Wagner Meira Jr, Cristina Duarte Murta, Sérgio Vale Aguiar Campos, and Dorgival Olavo Guedes Neto. *Sistemas de Comércio Eletrônico: Projeto e Desenvolvimento*. Editora Campus, 2002.

[Jun02] Orlando C. Pinho Junior. *Negociações Bilaterais Sequenciais Mediadas por Agentes*. (dissertação de mestrado), Universidade Federal de Pernambuco, 2002.

[Kas01] Daniel Santos Kaster. *Combinando bancos de dados e raciocinio baseado em casos para apoio a decisão em planejamento ambiental*. (dissertação de mestrado), Universidade Estadual de Campinas. Instituto de Computação, 2001.

[KFSBY03] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam. Protocols for Negotiating Complex Contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

[KL02] K. Kurbel and I. Loutchko. Multi-agent negotiation under time constraints on an agent-based marketplace for personnel acquistion. In *In Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB2002)*, pages 566–579, Erfurt, Germany, October 2002.

[Kle99] P. Klemperer. Auction Theory: A guide to the literature. *Journal of Economic Surveys*, 13:86–227, 1999.

[Kov02] Zsolt L. Kovács. *Redes neurais artificiais: fundamentos e aplicações*. Editora Livraria da Física, São Paulo, 3 edition, 2002.

[Lag02] Krista Lagus. Text retrieval using self-organized document maps. In *Neural Processing Letters*, volume 15, pages 21–29, January 2002.

[Lit94] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *11th International Conference on Machine Learning (ICML-94)*, pages 157–163, 1994.

[LKK04] Krista Lagus, Samuel Kaski, and Teuvo Kohonen. Mining massive document collections by the websom method. *Information Sciences*, 163(1-3):135–156, 2004.

[Luc04] Edimara Mezzomo Luciano. *Consolidação de Componentes de Modelos de Negócios para o Comércio Eletrônivo de Produtos Virtuais*. (tese de doutorado), Universidade Federal do Rio Grande do Sul, Porto Alegre, Novembro 2004.

[lVSCFM+04] Álvaro Vinicius S. Coêlho, Edilson Ferneda, Agenor Martins, Marcelo Alves Barros, and Flavius Luz Gorgônio. Help desk inteligente em gestão do conhecimento: Um tratamento integrador de paradigmas. Programa Excelência Gerencial do Exército Brasileiro (PEG-EB), 2004.

[MBCJ05] Antônio M.Silveira, Alfredo B.Furtado, Roberto C.L.Oliveira, and Carlos T.Costa Jr. Identificação de abordagens administrativas: Um ensaio com lógica fuzzy. *INFOCOMP Journal of Computer Science*, 4(1):36–45, 2005.

[Mel05] José Carlos M. F. Mello. *Negociação Baseada em Estratégia*. Atlas, São Paulo, 2 edition, 2005.

[MP43] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(1):115–133, 1943.

[MR03] Sildomar Takahashi Monteiro and C. H. C. Ribeiro. Aprendizagem da navegação em robôs móveis a partir de mapas obtidos autonomamente. In *Encontro Nacional de Inteligencia Artificial - XXIII Congresso da Sociedade Brasileira de Computação*, volume 1, pages 152–152, Campinas, 2003.

[MR04] Sildomar T. Monteiro and Carlos H. C. Ribeiro. Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel. *Revista de Controle & Automação*, 15(3):320–338, Julho 2004.

[Nak03] Mitsuaki Nakasumi. Credit risk management system on e-commerce: case based reasoning approach. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 438–449, New York, NY, USA, 2003. ACM Press.

[NJ05] Vidya Narayanan and Nicholas R. Jennings. An adaptive bilateral negotiation model for e-commerce settings. In *Seventh IEEE International Conference on E-Commerce Technology*, pages 34–41, 2005.

[Opr02] Mihaela Oprea. An Adaptive Negotiation Model for Agent-Based Electronic Commerce. In *Studies in Informatics and Control*, volume 11, pages 271–279, 2002.

[OSB+06] Rômulo Nunes Oliveira, Alan Pedro Silva, Ig Bitencourt, Herman Gomes, and Evandro Barros Costa. A multiagent based framework for supporting learning in adaptive automated negotiation. In *accepted for publication at the 8th International Conference on Enterprise Information Systems - ICEIS 2006*, Paphos - Cyprus, May 2006.

[OSBC05] Rômulo Nunes Oliveira, Alan Pedro Silva, Ig Bitencourt, and Evandro Barros Costa. Um ambiente multi-agente adaptável para negociações no e-commerce. In *10º Congresso Estadual de Informática e Telecomunicações - SUCESU - (anais no formato digital) ISSN 1897-4499*, Cuiabá-MT, Novembro 2005.

[Pau01] Gustavo E. Paula. *Modelo de Negociação Bilateral para Comércio Eletrônico*. (dissertação de mestrado), Universidade Federal de Pernambuco, 2001.

[PRR01] Gustavo E. Paula, F. Ramos, and G. Ramalho. Bilateral Negotiation Model for Agent-Mediated Electronic Commerce. In *Third Workshop on Agent-Mediated Electronic Commerce*, volume 2003 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2001.

[RN04] Stuart Russell and Peter Norving. *Inteligência Artifical*. Editora Campus, 2 edition, 2004. Tradução da segunda edição por Vandenberg D. de Souza.

[RS89] C. K. Riesbeck and R. C. Schank. *Inside Case-Based Reasoning*. Erlbaum, Hillsdale, NJ, 1989.

[SA04] Jaine José Silva and Mara Abel. Help desk com sistema rbc para as gerências de aplicativos do banco do brasil. Trabalho de Conclusão (especialização) - Universidade Federal do Rio Grande do Sul. Curso de Especialização em Desenvolvimento, Segurança e Qualidade na Internet, Porto Alegre - RS, 2004.

[San02] Tuomas Sandholm. emediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002.

[Sar05] José Alberto Rodrigues Pereira Sardinha. *MAS-School e ASYNC: Um Método e um Framework para Construção de Agentes Inteligentes*. (tese de doutorado), Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

[SC95] Tuomas W. Sandholm and Robert H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems, Special Issue on the Prisoner's Dilemma*, (37, 147-166), 1995.

[Sou00] Nelson Souza. *Controladores Lógicos Difusos: Uma Alternativa para o Controle de Processos Industriais Críticos*. (dissertação de mestrado), Universidade Federal de Santa Catarina, Florianópolis-SC, 2000.

[Ste04] Timo Steffens. Adapting similarity-measures to agent-types in opponent-modelling. In Mathias Bauer, Piotr Gmytrasiewicz, Gal A. Kaminka, and

David V. Pynadath, editors, *Workshop on Modeling Other Agents from Observations at AAMAS 2004*, pages 125–128, 2004.

[TAC01] Trading Agent Competition, Fevereiro 2001. http://www.sics.se/tac - acessado em 06/01/2006.

[Tes04] Gerald Tesauro. Extending q-learning to general adaptive multi-agent systems. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 215–250. MIT Press, Cambridge, MA, 2004.

[Tre04] Maurício Trezub. Guia de prevenção à fraudes e-commerce para lojistas. http://www.ciashop.com.br - acessado em 10/02/2006, 2004.

[TXF95] Malcon A. Tafner, Marcos Xerez, and Ilson W. Rodrigues Filho. *Redes Neurais Artificiais: Introdução e Princípios de Neurocomputação*. EKO, 1995.

[VLV99] A. Vellido, P.J.G. Lisboa, and J. Vaughan. Neural networks in business: a survey of applications (1992-1998). *Expert Systems with Applications*, 17:51–70, 1999.

[WB03] Jan Wendler and Joscha Bach. Recognizing and predicting agent behavior with case based reasoning. In *RoboCup*, pages 729–738, 2003.

[WC99] J. Christopher Westland and Theodore H. K. Clark. *Global Electronic Commerce: Theory and Case Studies*. MIT Press, Cambridge, Massachusetts, 1999.

[WD92] Christopher J. C. H. Watkins and Peter Dayan. Technical note q-learning. *Machine Learning*, 8:279–292, 1992.

[Win88] Patrick Henry Winston. *Inteligência Artificial*. LTC-Livros Técnicos e Científicos, 1988.

[WJ95] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[WW03] Christiane Gresse Von Wangenheim and Aldo Von Wangenheim. *Raciocínio Baseado em Casos*. Manole, Barueri, SP, 2003.

Apêndice A

Classificações no Comércio Eletrônico

O conhecimento de algumas classificações e termos básicos no comércio eletrônico são interessantes para melhor compreensão deste trabalho. Neste apêndice estão relacionados: os tipos de negociações e suas siglas freqüentemente usadas na área; as modalidades de comércio eletrônico; e os tipos de comércio eletrônico mais conhecidos, destinados à consumidores finais (clientes).

A.1 Tipo de Negócio

O comércio eletrônico, do ponto de vista das entidades envolvidas no negócio e segundo o uso dos meios eletrônicos para a condução das transações, podem ser identificados como [JMCN02; Luc04]:

- **B2B** (*Business to Business*): é o comércio eletrônico realizado entre duas ou mais empresas. O B2B agiliza todo o processo comercial entre uma empresa e suas empresas-clientes, desde a troca de informações sobre os produtos, preços e condições de pagamento, até a realização do pedido e os acertos sobre a entrega.

- **B2C/C2B** (*Business to Consumer* ou *Consumer to Business*): são transações comerciais que envolvem diretamente empresas e consumidores, como por exemplo em um contrato de compra e venda eletrônico entre um consumidor e uma loja ou shopping virtual.

- **C2C** (*Consume to Consumer*): são transações entre consumidores finais, como sites de leilões ou tocas, classificados on-line e outros negócios particulares.

- **B2G/G2B** (*Business to Government* ou *Government to Business*): trata-se de transações comerciais que possuem empresas e governo como partes contratantes, e podem ser reconhecidas em compras de determinado equipamento para uso em repartições públicas.

- **G2C/C2G** (*Government to Consumer* ou *Consumer to Government*): São transações envolvendo governo e consumidores finais, como pagamentos de impostos, serviços e outras atividades prestadas pelo governo via rede eletrônica.

- **G2G** (*Government to Government*): Transações entre governos de níveis diferentes (federal, estadual ou municipal) ou entre países diferentes, por meio eletrônico (prestação de serviços ou transações comerciais de importação e exportação).

Sendo que as três últimas definições (G2B, G2C e G2G) podem ser vistas como especializações do B2B ou B2C, se considerarmos o governo como uma empresa estatal [Luc04]. A Figura A.1 resume bem esta Seção.

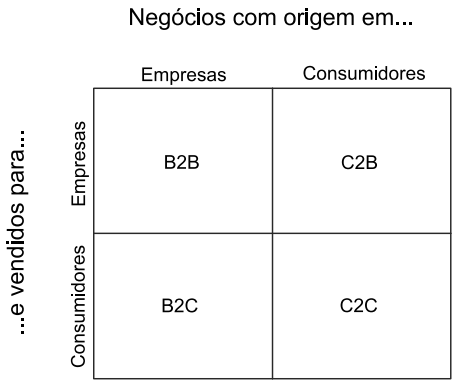

Figura A.1: Tipos de negócio no comércio eletrônico.

A.2 Modalidades de Comércio Eletrônico

Ainda segundo [JMCN02], as modalidades de Comércio Eletrônico são determinadas por dois critérios: natureza do bem (produto) e natureza da transação.

Natureza do produto: Indica qual o tipo do produto comercializado, que é classificado como real ou virtual. Produtos reais são produtos já comercializados tradicionalmente, tais como livros e CDs, enquanto produtos virtuais são todos aqueles produtos digitais cuja entrega é normalmente realizada através da Internet. A distinção entre produtos reais e virtuais é relevante no sentido que a natureza do produto determina a forma de interação do servidor de comércio eletrônico com provedores de serviços virtuais ou sistemas de logística. Por exemplo, uma loja que vende livros virtuais deve se preocupar com servidores de arquivo e banda para tráfego de dados, enquanto uma loja que vende livros reais (impressos) deve dar atenção para diversas editoras, sistema de entrega, estoque etc.

Natureza da transação: Indica a modalidade de comercialização adotada. Distinguimos nominalmente dois tipos de transação: *aquisição de bens* e a *prestação de serviços*. A aquisição de bens se caracteriza pela mudança da propriedade do produto em caráter permanente, enquanto a prestação de serviços é a execução de um serviço por tempo determinado, sem que haja transferência de propriedade. Ambos os casos oferecem desafios operacionais que precisam ser analisados. No caso da aquisição de produtos, a logística de entrega e a cobrança são questões fundamentais ao bom funcionamento do servidor. A modalidade de prestação de serviços, além de um sistema de cobranças, deve também prover mecanismos para o acompanhamento da execução dos serviços, que podem ser bastante complexos.

Combinando estas definições, temos quatro modalidades possíveis de comércio eletrônico: aquisição de produtos reais, aquisição de produtos virtuais, prestação de serviços reais e prestação de serviços virtuais. Essas modalidades são discutidas a seguir.

A aquisição de produtos reais é a aplicação mais tradicional do comércio eletrônico, principalmente em negócios orientados a consumidores (B2C). Do ponto de vista do cliente, o processo eletrônico se diferencia do processo tradicional em pelo menos dois aspectos: a forma de interação e a integração com recursos físicos. A forma de interação mais tradicional em serviços de comércio eletrônico é a *World Wide Web*, que até o momento oferece apenas recursos audiovisuais, mas, por exemplo, em Arranjos Produtivos Locais [Alm05] o uso de outra tecnologia dedicada pode ser mais adequado, garantindo uma conexão contínua para o

processo. A integração com recursos físicos é uma necessidade inerente à propria natureza do produto, como no caso do processo de produção real do bem, da forma de pagamento e da logística de entrega do produto.

Podem-se diferenciar três fatores que influenciam a integração com sistemas de logística. O primeiro fator é a reduzida granulação dos produtos vendidos, ou seja, os pedidos tendem a conter um menor número de itens, o que pode ser explicado pela maior facilidade de aquisição de bens pelos clientes, não sendo necessário agrupar muitos produtos a cada compra. Essa reduzida granulação impões uma maior demanda sobre a infra-estrutura de logística como um todo, podendo resultar na saturação de capacidade em alguns de seus componentes.

O segundo fator é o prazo de entrega, que pode ser lento em certos casos ou épocas. Para que o tempo no prazo de entrega seja reduzido, é necessário ampliar a infra-estrutura de forma a suportar os picos de demanda. A impossibilidade de prever a quantidade de pedidos que um produto pode sofrer é um dos maiores desafios de logística. Existem cálculos estatísticos considerando muitos fatores (época do ano, tempo do pedido no estoque, validade do produto, capital de giro disponível), mas as técnicas da IA (ver Subseção 2.2) podem também resolver questões de previsibilidade.

Aquisição de produtos virtuais, os recursos atuais de tecnologia da informação permitiram a criação de uma série de modalidades de comercialização baseadas na venda de produtos virtuais.

A comercialização de produtos virtuais também levanta questões interessantes: a primeira delas é com relação à disposição de informações a respeito de um produto sem que o produto em si esteja disponível; um outro aspecto nos quais produtos virtuais diferem fundamentalmente de produtos reais é com relação aos modelos de custo e de cobrança, que por serem mais baratos deram margem a outras formas de pagamento. Pelo custo baixo, a própria cobrança é um problema, uma vez que os custos de cobrança nas formas de pagamento tradicionais (por exemplo, cartão de crédito) facilmente superariam o custo do próprio produto comercializado, inviabilizando o modelo como um todo. Nesse caso, novas formas

de cobrança, como *micropagamentos*[1] e *dinheiro eletrônico*[2], têm sido desenvolvidas para permitir que tais transações tenham um custo compatível.

Um aspecto final na cobrança de bens digitais é a garantia de entrega dos bens, ou seja, a cobrança só deve ocorrer se o produto for efetivamente entregue ao usuário. O problema nesse caso é que a tecnologia da Internet não provê recursos muito elaborados para verificar a conclusão de transações envolvendo produtos virtuais, o que expõe a necessidade de desenvolvimento de novas metodologias.

A prestação de serviços reais refere-se a sistemas de comércio eletrônico para a automação na contratação e prestação de serviços, que se assemelham aos métodos usados para aquisição de produtos reais. Neste caso, entretanto, as restrições logísticas são mais severas e devem ser avaliadas cuidadosamente para fins de automação.

A manutenção dessas informações é mais complexa que nos sistemas de aquisição porque há uma grande variedade de eventos possíveis e que podem ser gerados não apenas a partir da Internet, mas também através do telefone ou pessoalmente. Independente da natureza da ocorrência ou do meio utilizado, todas essas informações devem estar sempre atualizadas, sob pena de uma inconsistência nas informações do sistema.

Um bom exemplo de prestação de serviços reais, são os oferecidos por servidores para páginas web, onde o serviço contratado envolve a cessão temporária de espaço físico nos servidores dentre outros.

A prestação de serviços virtuais é a aplicação mais inovadora em termos de exploração dos recursos de tecnologia da informação. O fornecimento de vídeo sob demanda é um exemplo de prestação de serviços virtuais que introduz novos requisitos para a implementação de servidores para esta modalidade de comércio eletrônico. Um dos desafios relaciona-se a verificação e manutenção da qualidade dos serviços prestados, diretamente ligado ao planejamento de capacidades físicas. Deve-se sempre ter em mente que serviços de provimento contínuo têm duração indeterminada e a qualidade do serviço oferecido pode se deteriorar com o aumento do número de locações em curso, principalmente quando os recursos físicos (servidores para os serviços) se aproximam do seu limite.

[1] São quantias mínimas sobre produtos ou serviços que são cobradas em conjunto após um consumo mínimo ou período.

[2] Ou *eletronic cash*, refere-se a transações efetuadas eletronicamente com o propósito de transferir fundos de uma parte para outra, facilitando as operações de câmbio.

Assim como as demais modalidades vistas nesta seção, a prestação de serviços virtuais abre inúmeras possibilidades do ponto de vista de personalização, desde o auxílio aos usuários para encontrar os serviços desejados até o estabelecimento de serviços adaptativos, cujo funcionamento se altera de acordo com a preferência e a reatividade dos usuários.

A.3 Tipos de Comércio Eletrônico

O *varejo eletrônico* (*ou e-tailing*), conforme o próprio significado sugere, é a venda direta de produtos para consumidores finais, via vitrines eletrônicas ou shoppings virtuais, geralmente no formato de um catalogo eletrônico e/ou leilões [Fon01].

- **Vitrines eletrônicas:** São geralmente extensões das lojas reais, embora existam outras empresas que trabalham apenas com o nicho encontrado na *Web*. Existem dois tipos de vitrines:
  - *vitrines Especializadas* - vendem um ou alguns produtos do mesmo gênero (por exemplo: flores, CDs);
  - *Vitrines Gerais* - vendem diversos produtos. Os bens mais apresentados são computadores e acessórios, roupas e calçados, brinquedos e alimentos. Os serviços mais comuns on-line incluem serviços de viagens, ações e títulos, banco eletrônico, seguros e busca de emprego.

- **Shoppings Virtuais:** Também chamados de *cybermall* ou *e-mall*, trata-se de um conjunto de lojas individualizadas em um endereço na Internet. A idéia básica de um shopping virtual é a mesma de um shopping real: fornecer um local único de compras que ofereça muitos produtos e serviços. Assim, como os fornecedores alocados em um shopping real, um fornecedor em um *e-mall* renuncia a certa dose de independência. Uma das práticas adotadas localmente, por exemplo, é oferecer um único sistema de pagamento para site todo, facilitando assim o processo de compra para o cliente.

- **Leilão:** é um mecanismo do mercado pelo qual os vendedores fazem ofertas e os compradores dão lances em seqüência. Os leilões são caracterizados pela natureza

competitiva, por meio da qual se alcança o preço final. A Internet fornece uma infraestrutura para realizar leilões a baixo custo, e com muito mais vendedores e compradores envolvidos. Existem diversos tipos de leilões, cada um com suas motivações e procedimentos. Os leilões se dividem em dois tipos principais, *leilões diretos* e *leilões reversos*.

- **Leilões diretos:** Os leilões diretos são utilizados principalmente como um canal de vendas, onde um vendedor leiloa um ou mais itens para os possíveis compradores. Os modelos mais comuns para leilões diretos são:
  - *Leilões Ingleses* - Os compradores fazem lances em um item por vez. O preço dos lances vai aumentando. O maior lance ganha (so o preço for o único critério).
  - *Leilões Yankees* - São semelhantes aos leilões ingleses, mas vários itens idênticos são oferecidos. Você pode dar lances em qualquer número de itens. Os preços dos lances vão subindo.
  - *Leilões Holandeses* - Geralmente, oferecem diversos itens idênticos. Os preços começam altos e vão caindo à medida que o relógio do leilão via correndo até que uma quantia específica seja apresentada. O primeiro a dar esse lance ganha.
- **Leilões Reversos:** Nos leilões reversos existe um comprador, que deseja adquirir um produto ou serviço. Os fornecedores são convidados a apresentar suas ofertas, o fornecedor que apresentar a menos oferta, ganha.
- **Trocas:** Estão relacionadas aos leilões, uma troca de bens oui serviços sem intermediação monetária.

Apêndice B

Plataforma JADE

O JADE (*Java Agent Development Framework*) é uma plataforma *open source* para desenvolvimento de sistemas multiagentes distribuídos, possui licença LGPL (*Library General Public License*), é implementado totalmente em Java e segue as especificações propostas pela FIPA[FIP06](*Foundation for Intelligent Physical Agents*)[1]. Por essas características, o JADE foi escolhido para o projeto, além de ser uma plataforma de fácil acesso e largamente utilizada no meio acadêmico.

B.1 Arquitetura

Um modelo abstrato para uma plataforma de agentes pode ser visto na Figura B.1,onde alguns itens importantes, definidos pela FIPA, estão identificados:

- *Agent Management System (AMS)*: responsável por controlar todo o acesso e uso de uma instância da plataforma. Conhecido como "serviço de páginas brancas", existe apenas um AMS para cada instância da plataforma. Além disso, é esse agente que gerencia o ciclo de vida de todos os agentes existentes. No Serviço de páginas brancas, que está relacionado ao serviço de endereçamento, cada agente recebe uma identificação (*AID*) única no ambiente.

- *Directory Facilitator (DF)*: esse agente fornece um serviço de "páginas amarelas", que é a catalogação dos serviços oferecidos pelos agentes na plataforma.

[1]Entidade responsável pela padronização em sistemas baseados em agentes.

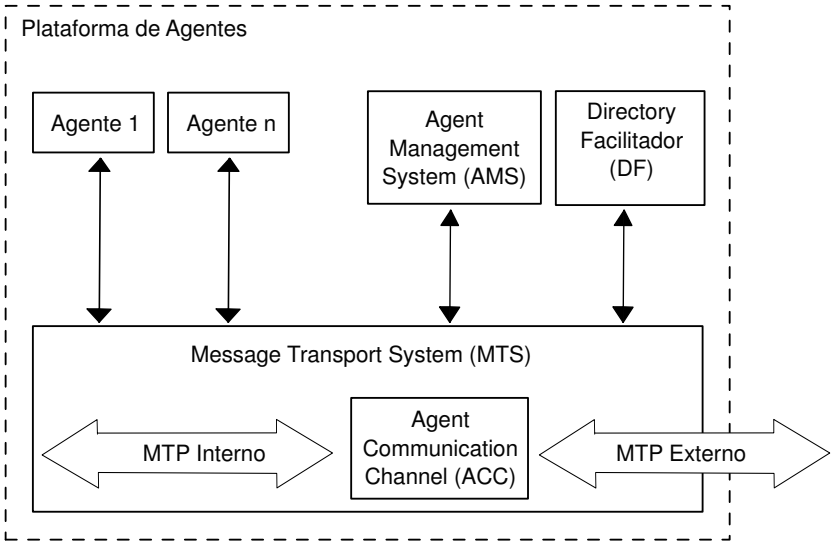

Figura B.1: Modelo de referência da FIPA para plataformas de agentes

- *Message Transport System (MTS)*: componente para o transporte das mensagens entre os agentes, onde se pode encontrar:
  - *Agent Communication Channel (ACC)*: provê a comunicação entre os agentes, e é responsável pelas as devidas traduções em trocas de mensagens entre agentes distribuídos e localizados, em ambientes heterogêneos.
  - *Message Transport Protocol (MTP)*: que utiliza-se do RMI[2] para toca de mensagens entre as máquinas virtuais Java (com IIOP[3] ou HTTP[4] quando entre plataformas diferentes)

O JADE utiliza-se dessa mesma arquitetura especificada pela FIPA, e um exemplo de um sistema multiagente em JADE pode ser visto na Figura B.2. Cada *container* é uma máquina virtual Java diferente, e agrupa um conjunto de agentes. As máquinas virtuais podem estar em hospedeiros (*hosts*) diferentes e heterogêneos, graças ao serviço do *ACC*.

[2] *Remote Method Invocation* (Java RMI)
[3] *Internet Inter ORB Protocol*
[4] *Hypertext Transfer Protocol*

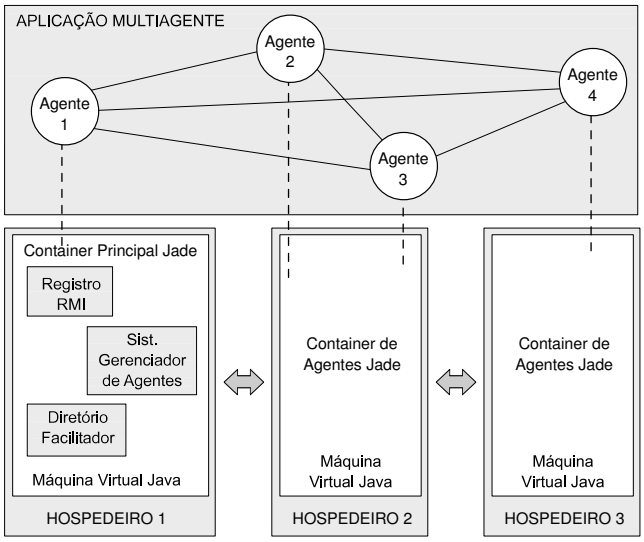

Figura B.2: Arquitetura JADE

B.2 Agentes JADE

A Plataforma JADE oferece um arcabouço para construção de agentes, onde a tarefa de trocar mensagens, registro nas páginas amarelas, endereçamento dos agentes estão disponíveis para os usuários do arcabouço. Para que um agente JADE possa ser construído, basta que o usuário do arcabouço implemente a classe abstrata *jade.core.Agent*, que possui um único método abstrato chamado *setup()*, conforme pode mostra o código a seguir.

```
1  public class AgenteJADE extends Agent{
2      /*
3       * Metodo Setup() sobrescrito da Classe jade.core.Agent
4       * */
5      public void setup(){
6          // Implementacao
7      }
8  }
```

B.3 Comportamentos

Cada agente JADE pode executar mais de uma tarefa, as quais podem ser mapeadas em comportamentos. O arcabouço faz todo o controle da execução dos comportamentos, que podem ser executados concorrentemente. O Comportamento mais simples pode ser herdado da classe *jade.core.behaviours.Behaviour*, onde é necessário a implementação de dois méto-

dos: *action()* e *done()*. Quando um comportamento é atribuído a um agente, o mesmo solicita ao comportamento a execução do método *action()*, que se repete enquanto o método *done()* estiver retornando *false*.

```
1 public class Comportamento extends Behaviour{
2     public void action(){
3         //Implementacao
4     }
5
6     public boolean done(){
7         //Implementacao
8     }
9 }
```

B.4 Troca de Mensagens

Para que dois agentes possam trocar mensagens, é necessário que cada um deles conheça o endereço JADE um do outro, que está abstraída na classe *AID*. Para que isso aconteça de forma dinâmica, pelo menos um dos agentes deverá ser cadastrado nas páginas amarelas (fornecido pelo agente *Directory Facilitator*), descrevendo quais seus serviços. No código seguinte, é criado um agente que tem por objetivo receber mensagens. Observe que para isso, logo quando ele é criado, cadastra-se nas páginas amarelas descrevendo seu serviço como *ReceberMensagem*, e com o nome *AgenteRecebendoMensagem* (linha 3).

```
1  public class AgenteRecebendoMensagem extends Agent{
2      public void setup(){
3          this.cadastroNasPaginasAmarelas("ReceberMensagem","AgenteRecebendoMensagem");
4          this.addBehaviour(new RecebedorDeMensagem(this));
5      }
6      public boolean cadastroNasPaginasAmarelas(String tipoDoServico, String nomeDoServico){
7          DFAgentDescription descritorDeServicosDeUmAgente = new DFAgentDescription();
8          ServiceDescription descricaoDeUmServico = new ServiceDescription();
9          descricaoDeUmServico.setName(nomeDoServico);
10         descricaoDeUmServico.setType(tipoDoServico);
11         descritorDeServicosDeUmAgente.addServices(descricaoDeUmServico);
12         try{
13             DFService.register(this,descritorDeServicosDeUmAgente);
14             return true;
15         }catch(FIPAException e){
16             return false;
17         }
```

```
18      }
19  }
```

Esse cadastro é executado pelo método *cadastroNasPaginasAmarelas(String tipoDoServico, String nomeDoServico)*, que executa os seguintes passos:

1. Na linha 7, é criado uma instância de *DFAgentDescription*. Essa classe representa uma descrição de um agente para as páginas amarelas.

2. Na linha 8, foi criado uma instância de *ServiceDescription*. Essa classe representa uma abstração de um serviço nas páginas amarelas. Nas linhas 9 e 10, foram atribuídas a essa instância, o nome e o tipo de serviço desejado.

3. Na linha 11, a descrição do serviço (*descricaoDeUmServico*) foi atribuída à instância da classe *DFAgentDescription*, chamada *descritorDeServicosDeUmAgente*.

4. Na linha 13, o serviço descrito em *descritorDeServicosDeUmAgente* foi registrado em *DFService*. A classe *DFService* é uma abstração do agente páginas amarelas (*Directory Facilitator*).

Ainda em relação ao código anterior, na linha 4, um comportamento é adicionado ao agente, chamado *RecebedorDeMensagem*, através do método *addBehaviour(Behaviour behaviour)* da classe *jade.core.Agent*. A implementação desse comportamento *RecebedorDeMensagem* segue em seguida.

```
1  public class RecebedorDeMensagem extends TickerBehaviour{
2      Agent agent;
3      public RecebedorDeMensagem(Agent agent){
4              super(agent,1000);
5              this.agent = agent;
6      }
7      public void onTick(){
8          ACLMessage receivedMessage = agent.receive();
9          if (receivedMessage != null){
10             //Tratar a mensagem
11         }
12     }
13 }
```

O comportamento *RecebedorDeMensagem* foi estendido da classe *TickerBehaviour*, cuja qual possui um único método abstrato chamado *onTick()*, que deverá ser implementado.

Esse método é executado a cada período de tempo definido em seu construtor. Neste caso específico (no código anterior), a cada um segundo (1000 milisegundos) o comportamento *RecebedorDeMensagem* executa as seguintes tarefas:

1. Na linha 8, o comportamento solicita ao seu agente que envie a última mensagem recebida, através do método *agent.receive()*.

2. Na linha 9, o comportamento verifica se a mensagem recebida não é nula. Não sendo, ele tratará a mensagem a sua maneira.

Por último, segue o código do agente *AgenteEnviandoMensagem*, que está enviando uma mensagem para o agente *AgenteRecebendoMensagem*, que executa essa tarefa segundo os seguintes passos:

1. Na linha 3, o agente *AgenteEnviandoMensagem* procura nas páginas amarelas quem está recebendo mensagens com o nome *AgenteRecebendoMensagem* e do tipo *ReceberMensagem*, através do método *getAIDsDasPaginasAmarelas*, que será detalhado adiante.

2. Na linha 5, uma mensagem no formato JADE é representada por uma instância da classe *ACLMessage*, para ser posteriormente enviada ao agente *AgenteRecebendoMensagem*.

3. Na linha 7, o agente *AgenteEnviandoMensagem* atribui o endereço do remetente à mensagem, nesse caso, o endereço do agente *AgenteRecebendoMensagem*.

4. Na linha 8, a mensagem recebe como conteúdo a String contendo *"Conteúdo da Mensagem"*.

5. Na linha 9, o agente *AgenteEnviandoMensagem* envia a mensagem, através do método *send* disponibilizado pela classe *Agent*.

```
1 public class AgenteEnviandoMensagem extends Agent{
2     public void setup(){
3         AID[] aid = this.getAIDsDasPaginasAmarelas
4                         ("ReceberMensagem","AgenteRecebendoMensagem");
5         ACLMessage message = new ACLMessage();
```

```
6           if (aid[0] != null){
7               message.addReceiver(aid[0]);
8               message.setContent("úContedo da Mensagem");
9               this.send(message);
10          }
11      }
12      public AID[] getAIDsDasPaginasAmarelas(String tipoDoServico,
13                                                  String nomeDoServico){
14          DFAgentDescription descritorDeServicosDeUmAgente =
15                                                  new DFAgentDescription();
16          ServiceDescription descricaoDeUmServico = new ServiceDescription();
17          descricaoDeUmServico.setName(nomeDoServico);
18          descricaoDeUmServico.setType(tipoDoServico);
19          descritorDeServicosDeUmAgente.addServices(descricaoDeUmServico);
20          try{
21              DFAgentDescription results[] =
22                      DFService.search(this,descritorDeServicosDeUmAgente);
23              AID aid[] = new AID[results.length];
24              for (int i = 0; i < results.length; i++) {
25                  aid[i] = results[i].getName();
26              }
27              return aid;
28          }catch(FIPAException e){
29              return null;
30          }
31      }
32  }
```

O método *getAIDsDasPaginasAmarelas* tem a função de resgatar das páginas amarelas os agentes que possuam um determinado *tipo de serviço* e um determinado *nome de serviço*, esse método é bastante similar ao método *cadastroNasPaginasAmarelas(String tipoDoServico, String nomeDoServico)* criado para o agente *AgenteRecebendoMensagem*. A grande diferença está nas linhas 21 e 22, que através do método *DFService.search*, faz uma busca pelos agentes que fornecem um determinado serviço. Este método retorna um Array de AID's, onde cada AID representa um endereço JADE.

B.5 Pacote *jade.tools*

O pacote jade.tools contém algumas ferramentas úteis que facilitam a administração e implementação dos agentes na plataforma.

Remote Monitoring Agent (RMA): console responsável pela administração e controle da plataforma. O JADE mantém coerência entre os RMAs através de envio de *multicasting*[5] entre eles.

Dummy Agent: ferramenta de monitoramento e *debug* que possui funções típicas tais como enviar, receber e armazenar mensagens ACL (*Agent Communication Language*).

Sniffer Agent: ferramenta utilizada para *debug* que rastreia e salva em arquivo a comunicação entre agentes. É capaz de interceptar as mensagens ACL em trânsito e exibir uma anotação gráfica muito semelhante ao UML[6], que é de grande utilidade para depuração da sociedade de agentes.

Introspector Agent: ferramenta que permite monitorar o ciclo de vida dos agentes, suas mensagens ACL trocadas e os behaviours em execução.

SocketProxy Agent: agente simples que age como uma porta bidirecional entre a plataforma e uma conexão TCP/IP. Este agente é útil para manipular *firewalls* ou prover interações entre a plataforma e *applets java*.

DF GUI: Modo gráfico para gerenciar a base de conhecimento do serviço de "páginas amarelas".

[5] Transmissão de dados de um computador central a vários outros numa rede

[6] *Unified Modeling Language*

Apêndice C

TAC - *Trading Agent Competition*

Esse apêndice irá apresentar o ambiente TAC, seus conceitos, funcionamento e regras. Também será visto um pouco sobre história da competição e o que ela objetiva, além de descrever o *framework* utilizado para a criação dos agentes. Como o TAC relaciona-se diretamente à teoria de leilões, a primeira seção deste apêndice será dedicada a uma introdução ao assunto, objetivando a melhor compreensão do *framework*.

C.1 Introdução à Teoria de Leilões

Teoria de Leilões [Kle99] dispõe-se a estudar tanto leilões como suas aplicações. Diversas razões práticas, teóricas e até empíricas têm aumentado a importância da Teoria de Leilões, não apenas pela diversidade na natureza das negociações, mas também pelo valor transacionado por elas. Os leilões são hoje peças-chaves utilizados por exemplo, tanto para a venda de sucatas como na concessão de áreas para exploração de petróleo e gás num dado país.

C.1.1 Considerações Básicas

Serão apresentados em seguida, alguns conceitos básicos e termos importantes para o melhor entendimento da Teoria de Leilões.

Leiloeiro: "Condutor" do leilão em si. Normalmente uma pessoa ou entidade que não esteja vendendo ou comprando dentro do leilão.

Arrematante: Pessoa ou entidade fundamental na dinâmica do leilão, pois é responsável pelas variações de preço no item, sendo leiloado através dos lances que efetua.

Preço Reservado: Valor máximo que o arrematante se dispõe a pagar pelo item em leilão. Esse dado é normalmente mantido em segredo por parte do arrematante, visto que é fundamental a sua intenção de pagar o menor preço possível pelo bem sendo leiloado.

C.1.2 Os Tipos Básicos de Leilões

Os leilões podem seguir os mais diversos comportamentos [Kle99], por exemplo, iniciar a partir de um preço mínimo e ir aumentando gradativamente o valor dos lances, ou irem reduzindo o valor a ser pago por um bem inicialmente ofertado de maneira bastante alta. Os leilões podem também possuir um momento de fechamento bem definido ou ainda não terem leiloeiro e arrematantes em seus papéis fixos.

Inglês

É o leilão mais comum dentre os demais. Seu funcionamento baseia-se em lances, sempre com valores ascendentes por parte de seus arrematantes. Estes, dão *lances abertos*[1] ou secretos, apenas apresentados caso seja o melhor lance entre os demais. Conhecido também por *leilão oral*, o leilão inglês é a escolha padrão quando se pretende leiloar antigüidades, obras de arte etc.

Holandês

E intuitivamente o contrário do leilão inglês. Aqui o preço é inicialmente publicado num patamar bastante elevado. O leiloeiro vai gradativamente baixando esse preço até conseguir algum arrematante disposto a pagar o dado valor, ficando com o item em leilão. Recebeu tal denominação por ser bastante utilizado em leilões de flores na Holanda.

Lances Fechados e Primeiro Preço

Envelopes lacrados são entregues pelos arrematantes contendo os preços que eles se dispõem a pagar pelo item em leilão. Não é permitido que um arrematante participe com mais de uma

[1] Lance aberto é feito publicamente, onde o valor do lance é conhecido por todos imediatamente.

oferta. Todos os envelopes são abertos em conjunto e o bem será vendido pelo maior lance. Essa modalidade de leilão foi utilizada em diversas privatizações de estatais no Brasil e também é muito usada em leilões imobiliários.

Vickrey

Similar ao anterior, diferenciando-se apenas no valor a ser escolhido para pagar o lance. Aqui, em vez da primeira é a segunda oferta que arremata o bem. Ou seja, o comprador que der o maior lance para o valor ofertado arremata o bem pelo lance imediatamente inferior ao seu. Idealizado por William Vickrey [Kle99], o leilão Vickrey é mais estudado do que mesmo utilizado, devido suas características teóricas.

Duplo

Não possui uma definição clara entre leiloeiro e arrematante. Nele todos podem tanto comprar como vender, o que acontece por exemplo com os corretores nas bolsas de valores. Os fechamentos ocorrem a toda hora e os bens podem passar de mão em mão várias vezes, dentro de um mesmo pregão, por exemplo. Essas características fazem dele um leilão com fechamento contínuo.

C.1.3 Estratégias de Atuação

A fundamentação teórica necessária às estratégias de atuação em leilões pode ser extraída da Teoria dos Jogos, inclusive, sistemas como o AgILE [GLS01] fazem bastante uso dessa característica. Em seguida dois conceitos fundamentais para a Teoria dos Jogos são apresentados [Cas03].

Estratégia Dominante: É a estratégia que consegue agregar mais utilidade para o jogador, ou seja, independente dos lances dos outros jogadores, arrematantes do leilão essa estratégia atenderá da melhor forma possível os anseios do jogador. Também recebe o nome de *completamente dominante*.

A utilidade para um arrematante pode ser calculada em função do preço reservado, preço pago pelo bem leiloado e a probabilidade de adquiri-lo. Isso não exclui que arrematantes do leilão considerem outros fatores, como por exemplo, o tempo gasto

durante a negociação de um item. Assim, a função de utilidade que será maximizada precisará ser considerada na formulação da estratégia [Cas03].

Estratégia Dominada: É aquela onde sempre há uma estratégia melhor a ser seguida, independente do conjunto de lances dos outros jogadores [Cas03].

Exemplificando: para leilões ingleses a estratégia dominante é dar lances pouco acima dos lances correntes e tentar assim ou vencer o leilão ou acabar por atingir o valor de reserva; Com um leilão de Vickrey a estratégia dominante é diferente e consiste em dar um lance exatamente igual ao preço reservado [Kle99].

C.2 O Ambiente TAC

O TAC [TAC01] é um fórum internacional que tem por objetivo fomentar pesquisas de alta qualidade na área de negociação entre agentes autônomos. Para isso, ele disponibiliza gratuitamente um *framework*, onde agentes desenvolvidos por times de diversas partes do mundo são postos a competir no servidor do TAC simulando a maioria das situações reais desse tipo de negociação.

Organizado inicialmente por Michael Wellman, o TAC aborda um cenário onde a venda de pacotes de viagem é negociada entre agentes — *TAC Classic*. Atualmente o TAC também dispõe mais uma modalidade, *TAC SCM*: essa nova competição tem agora, computadores como objeto de venda. Apenas o *TAC Classic* será abordado aqui.

C.2.1 Estudo do Ambiente TAC

A modalidade clássica do TAC também pode ser considerada um jogo. Cada partida desse jogo é chamada de uma instância do jogo e cada instância do jogo dura 9 minutos. Cada agente participante do jogo é um agente de viagens, com o objetivo principal de montar pacotes de viagem (de TACtown a Tampa, durante um período 5 dias) pelo menor preço possível. Cada agente está agindo em nome de oito clientes, que expressam suas preferências para vários aspectos da viagem. O segundo objetivo do agente de viagens é maximizar a satisfação total de seus clientes, expressa por uma função de utilidade que será descrita adiante [Dua05].

Os pacotes de viagem consistem em:

- Um vôo de ida para Tampa;
- Um vôo de volta para TACtown;
- Reservas de hotel;
- Bilhetes para alguns entretenimentos:
  - Alligator wrestling;
  - Amusement park;
  - Museum.

Neste ambiente existem interdependências óbvias, porque o viajante necessita de um hotel para cada noite entre a chegada e a partida do seu vôo, e pode ir a eventos de entretenimento somente durante esse intervalo de dias. Além disso, os clientes têm preferências individuais pelos dias que estarão em Tampa, pelo tipo de hotel, e por quais entretenimentos eles querem. Todos os três tipos de bens (vôos, quartos de hotéis, entretenimentos) são negociados em leilões separados com diferentes regras.

Conceitos

De forma geral, os principais conceitos a serem observados no ambiente estão relacionados a leilões:

***Good* (Bem):** É o item que está sendo leiloado. O qual o leiloeiro deseja vender e o arrematante deseja comprar.

***Bid* (Lance)** É o que expressa concretamente o desejo do arrematante em comprar, ou o desejo do leiloeiro em vender. Por exemplo, um arrematante informa que deseja comprar duas cadeiras pelo preço de $50 cada uma. Isto é o seu lance.

Lance de venda: É o preço oferecido pelo leiloeiro para vender seu bem.

Lance de compra: É o preço oferecido pelo arrematante para comprar um bem.

***Asked Price* (Preço de compra):** É o preço mínimo que um arrematante deve oferecer para conseguir comprar o bem pretendido. Normalmente é o valor do menor lance de venda do leilão.

***Bid Price* (Preço de venda):** É também o preço máximo que um leiloeiro pode pedir para conseguir vender um bem. Isto é, o valor do maior lance de compra do leilão.

***Quote* (Cotação):** No ambiente do TAC, o significado da palavra cotação varia nos diferentes tipos de leilão. A cotação de um leilão contém os preços vigentes no leilão, que podem ser somente o preço de compra (no caso dos leilões com único vendedor) ou o preço de venda e o preço de compra (no caso dos leilões duplos).

Funcionamento

No TAC exitem 3 tipos de negociações, que serão descritas a seguir:

Passagens Aéreas: Existe apenas uma companhia aérea que oferece vôos entre TACtown e Tampa, chamada TACAir. A companhia oferece apenas um vôo em cada sentido por dia, mas a quantidade de lugares nesses vôos é ilimitada. Como todos os clientes devem ficar ao menos um dia em Tampa, a TACAir não oferece vôos de volta no primeiro dia, nem vôos de ida no último dia, o quinto dia.

Hotéis: Como os clientes precisam de hotéis apenas desde a noite em que chegam até a noite anterior a sua partida, nenhum hotel está disponível, nem é necessário, no último dia. Deve-se lembrar também que um cliente não pode trocar de hotel durante a viagem.

Existem dois hotéis em Tampa: o Tampa Towers (TT) e o Shoreline Shanties (SS). Como o primeiro hotel é mais limpo, mais confortável, mais conveniente, e fica em um lugar mais agradável, espera-se que ele vá custar mais caro que o segundo. Os quartos dos hotéis são negociados em leilões ascendentes. Existe um leilão para cada combinação de hotel e noite. Em cada leilão são leiloados 16 quartos. Apenas os hotéis podem vender quartos, isto é, os agentes de viagem não podem revender quartos uma vez comprados. Não há lance mínimo para nenhum dos tipos de hotéis, a diferença de preço só aparece por causa da preferência dos clientes pelo melhor hotel.

***Tickets* de Entretenimento:** Existe um total de 8 *tickets* de entretenimento disponíveis para cada tipo de evento em cada dia. No começo de uma instância do jogo, cada agente recebe 12 desses *tickets*, divididos da seguinte maneira:

- 4 *tickets* de um único tipo qualquer no dia 1 ou no dia 4;
- 4 *tickets* de um único tipo qualquer no dia 2 ou no dia 3;
- 2 *tickets* de um único tipo qualquer (diferente dos acima sorteados) no dia 1 ou no dia 4;
- 2 *tickets* de um único tipo qualquer (diferente dos acima sorteados) no dia 2 ou no dia 3.

Os agentes devem então trocar (comprar e/ou vender) os *tickets* que lhes sobram ou faltam em um leilão duplo, com mecanismo de fechamento contínuo. Existe um leilão para cada combinação de evento e dia. Do mesmo modo como acontece com os hotéis, um cliente não pode usar um *ticket* de entretenimento no último dia de sua viagem.

Regras

Todos os leilões funcionam de acordo com o seguinte protocolo de alto nível:

- Um agente submete um lance para o leilão;
- A cotação do leilão é atualizada, indicando os preços correntes.

As regras para os diferentes tipos de leilões especificam quando ou sobre quais condições o leilão irá combinar os lances e gravar as transações, isto é, especificam-se mecanismos de fechamento de cada leilão.

Formato dos Lances: No ambiente do TAC, um lance é representado por uma *string* com um formato específico. Ela representa concretamente a vontade do agente de comprar ou vender bens em um leilão.

Leilões de Passagens Aéreas: Neste tipo de leilão os agentes podem submeter apenas lances de compra. Somente o vendedor da TACAir pode submeter lances de venda. As cotações são atualizadas imediatamente em resposta a novos lances. A cotação neste

leilão contém apenas o lance de venda corrente publicado pela TACAir. Assim como nos leilões duplos, o mecanismo de fechamento desse tipo de leilão é por evento, no momento da apresentação de um lance, que neste caso só pode ser um lance de compra.

Leilões de Hotel: Depois de passado um minuto do jogo, o primeiro leilão de hotel, escolhido randomicamente, fecha. Outro leilão, novamente escolhido aleatoriamente, fecha a cada minuto seguinte do jogo, até o oitavo minuto, quando o último leilão é fechado. Os agentes não sabem com antecedência qual leilão irá fechar em qual minuto. Os lances nesses leilões são combinados apenas uma vez, no momento do fechamento e as cotações são atualizadas uma vez por minuto.

Leilões de Tickets de Entretenimento: Nesse leilão os agentes podem submeter tanto lances de compra quanto de venda, mas esses lances não permitem que o agente venda bens para si mesmo. As cotações são atualizadas imediatamente sempre que novos lances são feitos. A cotação neste leilão é composta pelo preço de compra e pelo preço de venda. O preço de venda é o preço do maior lance de compra em espera. O preço de compra é o preço do menor lance de venda em espera.

Função de Utilidade dos Clientes: Dentro da modalidade clássica do TAC, os clientes possuem preferências por:

- Uma data de chegada em Tampa;
- Uma data de partida de Tampa. As preferências por datas de chegada e partida são escolhidas aleatoriamente para cada cliente, de modo que cada par de dias de chegada e partida são igualmente possíveis de ocorrerem.
- Ficar no melhor hotel, pelo qual o cliente oferece um bônus. Os valores dos bônus oferecidos pelo melhor hotel são escolhidos para cada cliente aleatória e uniformemente numa faixa entre $50 e $150;
- Diferentes entretenimentos, pelos quais eles oferecem diferentes bônus para cada tipo. Os valores dos bônus oferecidos pelos diferentes tipos de entretenimentos são escolhidos para cada cliente aleatória e uniformemente numa faixa entre $50 e $200.

Um outro conceito existente relacionado aos pacotes de viagem é a sua viabilidade. Um pacote de viagem é viável se ele contém quartos para todas as noites entre as datas de chegada e partida do cliente. O pacote deve possuir também um mínimo de dois dias de viagem, ou seja, um cliente não pode chegar e partir no mesmo dia. Com os *tickets* de entretenimentos também deve-se montar um pacote. Um pacote de entretenimento deve conter no máximo um ticket de cada tipo. Além disso, para que o pacote seja viável, não pode haver dois tickets para o mesmo dia e os *tickets* devem ser para dias nos quais o cliente está em Tampa.

Pontuação Final No final de cada instância do jogo, um agente de viagens possui diversas passagens aéreas, reservas de hotel e diversos tickets de entretenimento. Quanto aos *tickets* de entretenimento, caso o agente termine a instância do jogo com um número negativo de *tickets* (porque ele vendeu tickets que não possuía), ele recebe uma multa de $200 por cada ticket devido. O TAC possui um módulo chamado *scorer*, responsável por pegar os bens adquiridos por cada agente participante do jogo e montar com eles viagens viáveis para seus clientes.

C.2.2 O *Agentware*

O *Agentware* é um *framework* construído em Java fornecido pelos desenvolvedores do TAC, que provê uma base para a construção de agentes para o seu ambiente. Utilizando o *Agentware*, os desenvolvedores podem se concentrar no desenvolvimento das estratégias de atuação no jogo, pois o *framework* já provê algumas facilidades, tais como:

- Conexão e login no servidor;
- Recuperação de informações sobre o jogo;
- Comunicação assíncrona com o servidor do TAC;
- Armazenamento das informações sobre os bens já conseguidos;
- Armazenamento das preferências dos clientes.

Descrição das Principais Classes e Métodos

***AgentImpl*:** Esta é a classe abstrata que deve ser estendida caso se deseje construir um agente para o TAC utilizando o *framework Agentware*. Os métodos abstratos dessa classe são automaticamente chamados quando acontece um evento no jogo. A classe possui ainda um atributo do tipo da classe *TACAgent*, através do qual (pelos métodos da classe) tem acesso às informações sobre o jogo.

***Bid*:** Esta é a classe que representa a estrutura de um lance no jogo. Quando se cria um novo lance, *new Bid (int: auction)*, deve-se passar o código do leilão no qual o lance será feito. Para se adicionar um sub-lance a *Bid*, usa-se o método *addBidPoint (int: alloc, float: price)*. A alocação pode ser um inteiro positivo, anunciando que se quer comprar a quantidade indicada, ou um inteiro negativo, indicando que se quer vender. *Price* indica o preço pelo qual o agente quer comprar ou vender as unidades anunciadas.

***Quote*:** Esta é a classe que representa a estrutura de uma cotação no jogo. Em alguns métodos da classe *AgentImpl*, como o *quoteUpdated(Quote:quote)*, o agente recebe a cotação que acaba de ser atualizada.

C.2.3 O *Dummy Agent*

O *DummyAgent* é um agente construído pela equipe de desenvolvedores do TAC para ilustrar o uso do *AgentWare*. Ele trata de forma bem simples os problemas do TAC, e justamente por isso tem uma pontuação média baixa no jogo, cerca de 1400 pontos.

Alocação de Clientes

No inicio do jogo, após receber as preferências de seus clientes, o *Dummy* faz a alocação dos mesmos de acordo com suas preferências. Isto é, ele decide que irá comprar exatamente os bens preferidos por seus clientes.

Lances Iniciais

Após a fase de alocação, são feitos os lances iniciais nos leilões. Para cada tipo de leilão o *Dummy Agent* faz o lance inicial de uma maneira diferente. Para os leilões de passagens

aéreas, o lance inicial oferecido é sempre de $1000, um valor acima do preço máximo ao qual um leilão desse tipo pode chegar. Com isso, o *Dummy* garante que todas as passagens aéreas serão compradas no inicio no jogo, pelo primeiro preço oferecido pela companhia aérea. Para os leilões de hotéis, é oferecido também um lance fixo de $200. Já para os leilões duplos de entretenimento, são oferecidos lances iniciais de compra ou venda. Caso o agente possua tickets de determinado entretenimento sobrando, ele oferece um valor de $200 para a venda. Caso ele precise comprar tickets, ele oferece um lance de $50 para essa compra.

Novos Lances Durante o Jogo

Durante o jogo, cada vez que uma cotação é atualizada, o Dummy decide qual será a sua atitude em relação a essa atualização (menos nos leilões de passagens aéreas, nos quais as mudanças não são mais acompanhadas durante o jogo, pois todas as passagens necessárias já foram compradas no início).

Para qualquer leilão de hotel, a cada atualização, o *Dummy* verifica se ele ainda precisa comprar quartos naquele leilão especifico. Caso precise, ele verifica ainda qual é a quantidade de quartos que ele hipoteticamente já ganharia caso o leilão fechasse naquele momento e compara essa quantidade com a quantidade de quartos que ele ainda precisa comprar. Se a quantidade hipoteticamente de quartos ganhos for menor que a quantidade necessária de quartos, então é preciso aumentar o lance de compra. O novo lance será sempre o preço de compra mais $50.

No caso dos leilões de entretenimento, a cada atualização o *Dummy* compara a quantidade que ele possui do *ticket* leiloado com a quantidade necessária. Caso ele possua mais tickets do que precisa, ele diminui o preço do seu lance de venda dos tickets excedentes. A diminuição do preço se dá em função do tempo decorrente de jogo. Quanto mais o tempo passa, mais o preço diminui. Este valor é obtido pelo cálculo:

$$200 - (agent.getGameTime() * 120)/720000 \tag{C.1}$$

Caso ele possua menos *tickets* do que precisa, ele aumenta o preço do lance de compra dos tickets necessários também em função do tempo decorrente de jogo. Quanto mais o tempo passa mais o preço aumenta. Este valor é obtido pelo cálculo:

$$50 + (agent.getGameTime() * 100)/720000 \tag{C.2}$$

Apêndice D

Conjuntos de Regras para o Fornecedor 1

Conjunto 1

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 5) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 5) ENTÃO Beta = 0,1
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 1

Conjunto 2

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 10) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 10) ENTÃO Beta = 0,2
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 1

Conjunto 3

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 7) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 7) ENTÃO Beta = 0,1
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 1

Conjunto 4

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 5) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 5) ENTÃO Beta = 1

- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 10

Conjunto 5

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 8) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 8) ENTÃO Beta = 0,5
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 10

Conjunto 6

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 5) ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 5) ENTÃO Beta = 2
- SE (TempoMedio1 = 0) E (TempoMedio1 = 0) ENTÃO Beta = 1

Conjunto 7

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 7) ENTÃO Beta = 1
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 7) ENTÃO Beta = 0,1

Conjunto 8

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 7) ENTÃO Beta = 5
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 7) ENTÃO Beta = 1

Conjunto 9

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 4) ENTÃO Beta = 0,5
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 4) ENTÃO Beta = 8

Conjunto 10

- SE EstouGanhando ENTÃO NaoEnviarPropostas
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) > 7) ENTÃO Beta = 1
- SE ((TempoRestante * 2) / (TempoMedio1 + TempoMedio2) <= 7) ENTÃO Beta = 10