SISTEMA MULTI-AGENTE PARA COMÉRCIO ELECTRÓNICO

por

Henrique Daniel de Avelar Lopes Cardoso

Tese de Mestrado em Inteligência Artificial e Computação

Orientada por

Professor Doutor Eugénio Oliveira

Faculdade de Economia

Universidade do Porto

1999

Nota Biográfica

Henrique Daniel de Avelar Lopes Cardoso nasceu em Cedofeita – Porto, no dia 12 de Agosto de 1974. Licenciou-se em Informática de Gestão, pela Universidade Portucalense, em Abril de 1997.

Agradecimentos

Foram várias as pessoas que me mostraram este mundo fascinante que é a Inteligência Artificial. Agradeço em particular ao meu professor e orientador, o Professor Doutor Eugénio Oliveira, pela forma como me incentivou na elaboração desta tese.

Gostaria também de agradecer a todos os membros do NIAD&R, pela forma como me acolheram, integraram e ajudaram. A todos os meus colegas de mestrado, pelo seu companheirismo. Ao Max Schaefer, pela sua colaboração e contribuição em algumas das ideias que estão na base deste trabalho.

Agradeço ainda à Fundação para a Ciência e a Tecnologia, por ter apoiado financeiramente este trabalho, pelo Subprograma Ciência e Tecnologia do 2º Quadro Comunitário de Apoio.

Finalmente, agradeço aos meus pais por todo o apoio que me deram quando optei pelo ingresso no mestrado e durante estes dois anos de duração.

Resumo

O Comércio Electrónico tem sofrido um crescimento explosivo, acompanhando a evolução da Internet. Contudo, há ainda poucas estruturas de suporte a esta nova forma de comércio, que hoje se resume, em grande parte, ao anúncio e encomenda electrónicos.

Este trabalho foi motivado pela criação de um sistema de suporte ao Comércio Electrónico mediado por agentes. Sendo assim, tem como base a utilização de agentes de "software" em negociação automática. Além disso, pretendemos estudar e desenvolver a utilização de um tipo particular de aprendizagem na negociação entre os agentes.

Assim, desenvolvemos um sistema denominado SMACE (Sistema Multi-Agente para Comércio Electrónico), permitindo que os utilizadores deleguem intenções de compra ou venda em agentes de "sofware" que negoceiam autonomamente, segundo um modelo de negociação multi-lateral (várias negociações simultâneas) e multi-dimensão (a negociação pode decorrer sobre vários atributos). O sistema foi utilizado como uma plataforma de testes de diferentes estratégias de negociação, permitindo ainda que agentes criados externamente interajam no mercado do SMACE. De facto, este sistema consiste numa plataforma que facilita a construção de agentes de "software" para Comércio Electrónico, que negoceiam segundo o modelo e protocolo de negociação adoptados. Os agentes podem ser criados através de uma API construída para o efeito.

Esta plataforma permitiu-nos avaliar o desempenho de agentes dotados de capacidades de aprendizagem, baseada na interacção com o ambiente. O algoritmo de aprendizagem implementado foi sendo sucessivamente adaptado, de modo a propiciar um melhor desempenho dos agentes no processo de negociação.

Os resultados obtidos mostram que a aprendizagem em negociação, e em especial no domínio do Comércio Electrónico, é possível. No entanto, dado o dinamismo dos mercados em geral, tal tarefa não é fácil.

Abstract

Electronic Commerce has undergone an explosive growing, following closely the evolution of the Internet. Yet, there are few supporting structures to this new form of commerce, which today mostly resumes to the electronic announcement and ordering.

This work was motivated by the creation of a system to support agent-mediated Electronic Commerce. So, it is based on using software agents in automated negotiation. Besides that, our intention was to study and develop the implementation of a certain kind of learning in the agent's negotiation process.

Therefore, we developed a system which we called SMACE (*Multi-Agent System for Electronic Commerce*), allowing the users to delegate buying or selling intentions to software agents that negotiate autonomously, following a multi-party (multiple simultaneous negotiations) and multi-issue (negotiation can be made over a set of attributes) negotiation model. The system was used as a testing platform for different negotiation strategies, also allowing externally created agents to interact in the SMACE market. In fact, this system consists of a platform that facilitates the construction of software agents for Electronic Commerce purposes, which negotiate under the adopted negotiation model and protocol. The agents can be created through an API built to that end.

This platform allowed us to evaluate the performance of agents enabled with learning capabilities, which are based on the interaction with the environment. The learning algorithm implemented was progressively adapted, in order to allow for better agents performance in the negotiation process.

The results obtained show that learning in negotiation, and especially in the Electronic Commerce domain, is possible. However, due to the dynamics of the markets in general, such task is not easy to accomplish.

Índice

INTRODUÇÃO 1

1. COMÉRCIO ELECTRÓNICO 3

1.1 O Comércio Electrónico no Contexto Actual 3

1.2 A Expansão de Mercados Electrónicos 6

2. AGENTES E COMÉRCIO ELECTRÓNICO 9

2.1 Sistemas Multi-Agente 9

2.1.1 Agente 10

2.1.2 Comunicação 12

2.1.3 Plataformas 14

2.2 O Modelo *Consumer Buying Behaviour (CBB)* 16

2.2.1 Selecção do Produto 18

2.2.2 Selecção do Vendedor 19

2.2.3 Negociação 20

3. NEGOCIAÇÃO 23

3.1 Negociação Competitiva vs. Cooperativa 23

3.2 Leilões Electrónicos 25

3.3 Outros Estudos 28

4. MERCADOS ELECTRÓNICOS BASEADOS EM AGENTES 30

4.1 AuctionBot 30

4.2 Fishmarket 31

4.3 Kasbah 32

4.4 Tete-a-Tete 35

4.5 Comparação 36

5. APRENDIZAGEM 38

5.1 Aprendizagem em Negociação 40

6. ABORDAGEM SMACE 42

6.1 O Mercado 42

6.2 Modelo de Negociação 43

6.3 Protocolo de Negociação 45

6.4 Plataforma e Comunicação 48

6.4.1 SMACE e JATLite 49

6.4.2 SMACE e KQML 50

6.5 Aprendizagem 51

6.6. Implementação 52

6.6.1 Encontro de Agentes 54

6.6.2 Dimensões de Negociação 55

7. TÁCTICAS 56

7.1 Tácticas Dependentes do Tempo 56

7.2 Tácticas Dependentes de Recursos 58

7.2.1 Tácticas de Tempo Máximo de Negociação Dinâmico 58

7.2.2 Tácticas de Estimativa de Recursos 58

7.3 Tácticas Dependentes do Comportamento 59

7.3.1 Imitação Proporcional 59

7.3.2 Imitação Absoluta 59

7.3.3 Imitação Proporcional Média 60

8. ESTRATÉGIAS 61

8.1 Estratégias Orientadas pelo Utilizador 62

8.2 Estratégias Adaptativas 62

9. IMPLEMENTAÇÃO DO ALGORITMO DE APRENDIZAGEM 65

9.1 Aprendizagem por Reforço – Aprendizagem-Q 65

9.1.1 Cálculo da Função de Valor 66

9.1.2 Escolha de Acções 67

9.2 Implementação do Algoritmo de Aprendizagem-Q 68

10. EXPERIÊNCIAS E RESULTADOS 72

10.1 Análise Inicial das Tácticas 73

10.1.1 Experiências com Agentes MTA 74

10.2 Experiências Iniciais com Agentes ABA 77

10.2.1 Configuração dos Cenários 78

10.2.2 Resultados 80

10.2.3 Análise dos Resultados 84

10.3 Melhorias no Algoritmo de Aprendizagem 86

10.3.1 Valores Iniciais Optimistas 86

10.3.2 Função de Reforço 87

10.3.3 Técnica dos "After-States" 89

10.4 Experiências com Agentes ABA Melhorados 89

10.4.1 Configuração dos Cenários 90

10.4.2 Resultados 92

11. CONCLUSÕES 99

APÊNDICES 101

A – JATLite 101

B – KQML 104

C – Mensagens KQML entre Agentes e o Mercado 107

D – Mensagens KQML entre Agentes de Mercado 108

REFERÊNCIAS BIBLIOGRÁFICAS 109

ANEXOS 114

A – Ecrãs do sistema SMACE 114

Índice de Figuras

Figura 2.1 Arquitectura básica de um agente 11

Figura 2.2 Comunicação entre agentes 12

Figura 2.3 Protocolo da Rede Contratual 13

Figura 6.1 Esquema do mercado 42

Figura 6.2 Problema de acordos simultâneos 46

Figura 6.3 Problema de “deadlock” 47

Figura 6.4 Actividade do fluxo de negociação entre dois agentes 48

Figura 6.5 Abordagem inicial SMACE e JATLite 49

Figura 6.6 Abordagem SMACE e JATLite 50

Figura 6.7 SMACE API 52

Figura 6.8 Encontro de agentes 55

Figura 9.1 Representação de um Estado 69

Figura 10.1 Funções polinomiais utilizadas nos agentes ABA 78

Figura 10.2 Curvas de negociação dos agentes MTA nos cenários 3 e 4 80

Figura 10.3 Utilidades e utilidade média do agente 1ABAs1 81

Figura 10.4 Utilidades e utilidade média do agente 2ABAs1 81

Figura 10.5 Percentagem de acordos obtidos pelo agente 3ABAs1 82

Figura 10.6 Utilidades e utilidade média do agente 3ABAs1 82

Figura 10.7 Percentagem de acordos obtidos pelo agente 4ABAs1 83

Figura 10.8 Utilidades e utilidade média do agente 4ABAs1 83

Figura 10.9 Transmissão de reforços 85

Figura 10.10 Função ingénua de cálculo do reforço 88

Figura 10.11 Função de cálculo do reforço baseada na utilidade média 88

Figura 10.12 Curvas de negociação dos agentes MTA no cenário 5 92

Figura 10.13 Utilidades e utilidade média do agente 1ABAIIs1 93

Figura 10.14 Utilidades e utilidade média do agente 2ABAIIs1 93

Figura 10.15 Percentagem de acordos obtidos pelo agente 3ABAIIs1 94

Figura 10.16 Utilidades e utilidade média do agente 3ABAIIs1 94

Figura 10.17 Percentagem de acordos obtidos pelo agente 4ABAIIs1 95

Figura 10.18 Utilidades e utilidade média do agente 4ABAIIs1 95

Figura 10.19 Evolução dos desvios padrão dos agentes ABA e ABAII nos cenários 3 e 4 97

Figura 10.20 Utilidades e utilidade média do agente 5ABAIIs1 98

Figura 10.21 Utilidades e utilidade média do agente 6ABAIIs1 98

Figura A.1 Níveis do JATLite 101

Figura B.1 Níveis do KQML 104

Índice de Tabelas

Tabela 2.1 Sistemas baseados em agentes e o modelo CBB 18

Tabela 4.1 Comparação de mercados electrónicos baseados em agentes..... 36

Tabela 10.1 Pesos relativos das tácticas para os MTAs vendedores 74

Tabela 10.2 Configuração do MTA comprador em cada cenário..... 75

Tabela 10.3 Utilidades médias obtidas pelos agentes MTA 76

Tabela 10.4 Configuração dos cenários com agentes ABA..... 79

Tabela 10.5 Resumo dos resultados das experiências com agentes ABA 84

Tabela 10.6 Configuração dos cenários com agentes ABAII 90

Tabela 10.7 Configuração dos cenários dinâmicos com agentes ABAII 91

Tabela 10.8 Resumo dos resultados das experiências com agentes ABAII 96

Tabela B.1 Parâmetros KQML reservados 106

Tabela C.1 Mensagens KQML trocadas entre os agentes e o mercado..... 107

Tabela D.1 Mensagens KQML trocadas entre os agentes 108

Introdução

Este trabalho de dissertação é subordinado ao tema *Sistema Multi-Agente para Comércio Electrónico*. Tal como o nome indica, a motivação que levou à escolha deste tema foi a disponibilização na Web de um sistema que viabilizasse o Comércio Electrónico mediado por agentes. O trabalho desenvolvido gira em torno de duas áreas fundamentais: a negociação e a aprendizagem automáticas.

Foi portanto definido como objectivo a construção de um sistema informático baseado em agentes, representando um mercado onde agentes compradores e vendedores negoceiam de uma forma multi-lateral (várias negociações simultâneas) e multi-dimensão (a negociação pode decorre sobre vários atributos). O sistema denomina-se **SMACE** (**S**istema **M**ulti-**A**gente para **Comércio Electrónico**) e inclui a possibilidade de criação de agentes predefinidos ou utilização de agentes externos.

A construção deste sistema permitiu obter uma plataforma de testes para algo mais ambicioso: demonstrar que é possível dotar os agentes do mercado com capacidades de aprendizagem que lhes permitam, pela sua experiência, obter melhores resultados no processo de negociação. Esse processo de aprendizagem foi incluído num tipo de agente específico, tendo sido realizados testes para a avaliação do seu desempenho. O sistema inclui portanto dois tipos de agentes predefinidos – com aprendizagem e sem aprendizagem –, podendo cada um destes ser utilizado com um leque variado de opções no que diz respeito à sua estratégia de negociação. Além destes, o sistema admite ainda a interacção de agentes criados externamente com o mercado, e disponibiliza ferramentas que facilitam a sua construção.

As experiências conduzidas e relatadas no presente relatório não pretenderam simular situações reais de Comércio Electrónico. Elas foram executadas em cenários bem definidos, certamente limitados, mas nos quais pudemos avaliar o desempenho dos agentes dotados de aprendizagem desenvolvidos em contraste com agentes usando estratégias fixas.

A implementação inicial do algoritmo de aprendizagem nos agentes foi levada a cabo, no NIAD&R (Núcleo de Inteligência Artificial Distribuída & Robótica da FEUP e LIACC), sob a orientação do Prof. Eugénio Oliveira, por Max Schaefer, estando

documentada em Schaefer (1999). A construção do sistema SMACE, bem como os resultados primários da sua utilização, encontra-se igualmente documentada em Lopes Cardoso *et al.* (1999).

O trabalho descrito neste relatório engloba várias áreas de conhecimento, tais como o Comércio Electrónico, sistemas multi-agente, negociação, aprendizagem e plataformas distribuídas. Estas áreas estão caracterizadas ao longo do relatório, com um grau crescente de detalhe, à medida que se vai apresentando o trabalho desenvolvido.

Na primeira parte do relatório é apresentado, de uma forma abrangente, o "estado da arte" da área envolvente do presente trabalho. O capítulo 1 introduz a temática do Comércio Electrónico e a sua utilização na Internet. O capítulo 2 aborda o conceito de agente, quer na perspectiva do indivíduo quer na sua integração em sistemas multi--agente. Descreve ainda a forma pela qual esta tecnologia pode ser e tem vindo a ser aplicada ao Comércio Electrónico. A negociação em si, parte fundamental do nosso trabalho, é descrita no capítulo 3. O capítulo 4 descreve alguns dos mercados electrónicos existentes baseados em agentes. Finalmente, no capítulo 5, é abordada a aprendizagem automática e, em especial, a aprendizagem em negociação.

Numa segunda parte deste relatório são descritas, mais pormenorizadamente, as metodologias adoptadas no desenvolvimento da tese. Assim, no capítulo 6 é descrito o sistema SMACE, incluindo alguns detalhes da sua implementação. O capítulo 7 debruça-se sobre as tácticas que os agentes utilizam para negociar. O capítulo 8 explica os dois tipos de estratégias incluídos nos dois tipos de agentes implementados. Finalmente, o capítulo 9 fornece pormenores sobre o tipo de aprendizagem automática utilizado e sua implementação.

Na terceira parte do relatório são apresentados os testes e os resultados de utilização do SMACE. O capítulo 10 descreve as experiências conduzidas e explica algumas melhorias introduzidas posteriormente nos agentes com aprendizagem, bem como os seus efeitos. O capítulo 11 termina o relatório, apresentando as conclusões mais relevantes do nosso trabalho.

1. Comércio Electrónico

O comércio é a forma pela qual clientes e fornecedores se encontram num determinado local e tempo, de modo a anunciar intenções de compra e venda, que eventualmente se correspondem provocando o início de transacções comerciais. Devido às inovações nas tecnologias de informação e comunicação durante os últimos anos, as restrições temporais e espaciais foram-se desvanecendo e, portanto, essas transacções comerciais tornaram-se mais fáceis [Rocha e Oliveira (1999)]. O Comércio Electrónico usa as tecnologias de informação e comunicação, possibilitando o suporte electrónico de documentos e, opcionalmente, a automatização dos processos de negociação e transacção.

1.1 O Comércio Electrónico no Contexto Actual

A popularidade da Internet e da "World Wide Web" (WWW) aumentou fortemente a importância e a necessidade do Comércio Electrónico. Contudo, este ainda não desenvolveu todo o seu potencial na redefinição do mercado no qual produtos e serviços são transaccionados. Algumas das razões que estão na origem deste atraso são os hábitos de compra dos consumidores, a confiança e segurança das transacções e o facto de estarmos na presença de modelos de mercado novos e portanto incertos [Moukas *et al.* (1998)]. Se bem que alguns destes problemas estejam a ser abordados por governos, pela indústria e por instituições de investigação, há ainda poucos procedimentos padrão adoptados, o que dificulta a sua resolução.

O processo tradicional de compra requer do consumidor um grande esforço, incluindo a procura de vendedores para o produto pretendido (por exemplo, pela consulta de catálogos), a comparação de preços e outras facetas do produto entre os vendedores disponíveis e, depois de a decisão estar tomada, a troca de dinheiro pelo produto, através de um canal acordado e idealmente seguro [Moukas *et al.* (1998), Terpsidis *et al.* (1997)].

Há cada vez mais empresas com sítios na "Web", possibilitando aos consumidores comprar via Internet. No entanto, a maioria destes sítios não alterou radicalmente a forma de comprar, consistindo muitas vezes em versões "on-line" de montras ou

catálogos de produtos, com capacidades desenvolvidas de pesquisa. Alguns deles possibilitam a pesquisa de produtos através da procura de palavras, outros organizam os produtos por categorias, outros possuem ainda mecanismos mais avançados de pesquisa. Também referidos como “sítios de anúncios classificados”, eles ajudam os utilizadores a encontrar anúncios de produtos do seu interesse, o que constitui apenas um passo do processo de compra.

Actualmente, a utilização da Internet como meio de compra e venda de produtos resume-se na sua essência ao anúncio electrónico – criação de catálogos “on-line” – e à pesquisa facilitada da informação constante nesses catálogos. Tem-se contudo verificado um aumento significativo do volume de transacções concretizadas via Internet, através de um número crescente de sítios que permitem a compra directa por pagamento electrónico.

O Comércio Electrónico envolve bastante mais do que a simples pesquisa de produtos expostos em montras ou catálogos virtuais e a comparação de preços. Engloba a automatização da negociação entre compradores e vendedores, com vista a encontrar um acordo quanto ao preço e outros termos da transacção. Trata também da encomenda de produtos e do pagamento electrónico.

A globalização do mercado está a pôr em causa a postura tradicional dos retalhistas face aos clientes. Estes últimos têm actualmente ao seu alcance um grande número de canais, fisicamente remotos. A descoberta de canais como a Internet afecta todos os retalhistas, independentemente da sua imagem de marca ou da sua dimensão. Com vista a explorar ao máximo o potencial do mercado na Internet, os vendedores “on-line” devem reorganizar radicalmente a sua política de “marketing”. O envolvimento mais activo do consumidor no processo de compra deve ser utilizado para criar novos produtos, bem como para desenvolver novas estratégias de “marketing”. Muitos retalhistas criaram já o seu sítio na “Web”, disponibilizando aos utilizadores informação sobre a localização de lojas, promoções de venda, oportunidades de emprego e montras de produtos. No entanto, o problema principal é que, muitas vezes, torna-se difícil encontrar a informação específica que é necessária num determinado momento, para um determinado utilizador.

Os retalhistas na Internet enfrentam o mesmo tipo de problemas que o Comércio Electrónico, dado que a Internet ainda não desenvolveu todo o seu potencial. Os principais obstáculos são a segurança das transacções, a abundância de informação não estruturada e muitas vezes irrelevante e a falta de informação hierarquizada. Há alguns esforços no sentido de estabelecer processos padrão na forma como decorrem as transacções, visando ao mesmo tempo aumentar a segurança das mesmas. A falta de informação estruturada na Internet afasta possíveis consumidores. Embora haja tecnologias implementadas que permitem filtrar a informação relevante e disponibilizar recomendações sobre produtos, estas não foram integradas numa infra-estrutura de Comércio Electrónico, o que poderia abstrair o utilizador de alguns problemas relacionados com o excesso de informação. Esta, sendo hierarquizada, poderia permitir a construção de fontes de informação tipo "páginas amarelas", guiando o utilizador pelas lojas virtuais apropriadas [Terpsidis *et al.* (1997)].

O retalho de hoje pode ser descrito como uma competição entre vendedores pela preferência do cliente. Portanto, a relação que um vendedor deseja manter com os clientes é cooperativa e a longo prazo, procurando maximizar a sua satisfação. Isto aumenta a probabilidade de novas compras e a aquisição de novos clientes através de uma boa reputação. As novas tecnologias estão a alterar radicalmente o comércio por retalho. Actualmente existem já novos sistemas, baseados em agentes, que reduzem os custos de transacção entre vendedores e consumidores, e que ajudam os vendedores a aumentar as suas vendas. O acesso a sítios comerciais faz-se de uma forma cada vez mais rápida e automática através de agentes pesquisadores, o que torna a competição entre vendedores mais crítica. Contudo, algumas características do comércio a retalho não mudarão. Os consumidores irão continuar a preferir os vendedores que melhor satisfaçam as suas necessidades, e estes continuarão a desejar manter uma relação duradoura com os clientes [Guttman e Maes (1998b)].

Tem-se especulado sobre o papel dos intermediários nos mercados "on-line". Há quem acredite que terão tendência a desaparecer, à medida que as tecnologias substituem o seu trabalho (reduzindo custos de transacção) e que os produtores vendem directamente aos consumidores [Guttman e Maes (1998b)]. No entanto, no contexto actual, todos os tradicionais interlocutores comerciais têm possibilidade de estar

presentes, com vantagem, no novo ambiente de negócio proporcionado pelo Comércio Electrónico.

1.2 A Expansão de Mercados Electrónicos

O crescimento da "Web" permitiu um aumento significativo das transacções electrónicas entre as organizações. Contudo, esta abertura do mercado, só por si, não conduziu a uma transformação radical na forma como as organizações se relacionam.

Nas formas de relacionamento electrónico entre as organizações continuam a distinguir-se dois mecanismos [Romm e Sudweeks (1998)]:

- *hierarquia electrónica*: as decisões de gestão controlam a actividade, especificando os atributos de produtos e serviços. O comprador não escolhe de entre vários vendedores potenciais, mas obtém produtos e serviços de um fornecedor predeterminado. O relacionamento entre o fornecedor e cada um dos compradores é considerado hierárquico;
- *mercado electrónico*: o fluxo de produtos e serviços é coordenado por um sistema descentralizado. A disponibilidade de produtos e serviços com diferentes preços, quantidades e qualidades variadas e outros atributos, não é determinada por direcções de gestão de uma firma fornecedora ou compradora dominante. Em mercados de informação perfeita, os compradores compram ao fornecedor que ofereça a melhor combinação de atributos dos seus produtos.

As redes electrónicas interorganizacionais podem aumentar a coordenação entre as firmas, de duas formas contrastantes [Romm e Sudweeks (1998)]. Por um lado, através da redução dos custos da procura de produtos e serviços, aponta-se para um crescimento do mecanismo tipo mercado electrónico. Os compradores podem, de uma forma rápida e barata, avaliar as ofertas de vários fornecedores. Por outro lado, a utilização de redes electrónicas permite estreitar o relacionamento entre as diferentes entidades, integrando os processos de interacção entre elas. Isto favorece as relações de longo prazo entre fornecedores e compradores, que estabelecem assim relações de tipo hierárquico.

Há uma série de factores que podem influenciar a tendência para um dos mecanismos electrónicos referidos. Um deles diz respeito à descrição dos atributos dos produtos. Os

produtos de menor complexidade são mais fáceis de descrever e procurar. Por esta razão, os mercados electrónicos desenvolver-se-ão primordialmente entre firmas que comercializam produtos simples. Um exemplo destes é o livro, que pode ser facilmente identificado através do ISBN. Um artigo difícil de identificar é por exemplo a fotografia, para a qual não existem descritores convencionais.

Outro factor importante é o grau de abertura da rede. Uma rede que facilita a comunicação entre as diferentes entidades, através da utilização de protocolos de domínio público, e cujos custos de ligação são baixos, é mais aberta e promove assim a interacção entre mais participantes.

A adesão de mais firmas a um ambiente de uma rede electrónica aberta depende também de quem controla a rede. Uma rede gerida por uma entidade independente do conjunto de vendedores e compradores conduz mais facilmente a relações de mercado entre estes.

Em redes de dados privadas, existentes principalmente antes do crescimento explosivo da Internet, as hierarquias electrónicas são mais comuns do que os mercados electrónicos [Romm e Sudweeks (1998)]. Ao contrário das redes de dados internas, que são usadas principalmente para reduzir os custos de operação, as redes externas permitem que as firmas criem novos serviços diferenciadores e produtos rentáveis. Contudo, o uso geral das redes abertas baseia-se no desenvolvimento de relações hierárquicas de longo prazo entre firmas. Os baixos custos das redes abertas permitem que estas relações hierárquicas se estendam mesmo a parceiros de negócio pequenos. Só quando as redes abertas se estendem ao consumidor final é que elas são usadas para suportar mercados electrónicos.

Para que o Comércio Electrónico se desenvolva, é necessária a presença de entidades que providenciem serviços de selecção como a procura de produtos, comparação e avaliação [Romm e Sudweeks (1998)]. Assim, para além dos poderes acrescidos que o Comércio Electrónico atribui a clientes e vendedores (globalização da oferta e da procura, automatização da negociação, etc.), também certo tipo de intermediários ganha importância e visibilidade, na disponibilização da informação relevante sobre o mercado em cada situação particular.

Está, portanto, criado todo o contexto de necessidades que apelam para o desenvolvimento das capacidades informáticas que suportem sistemas com as características interactivas pretendidas.

2. Agentes e Comércio Electrónico

Agentes de "software" são programas de computador com algum grau de inteligência e autonomia, que tenderão a ser cada vez mais utilizados na execução de tarefas repetitivas e demoradas. Para que os agentes sejam realmente uma ajuda "inteligente", eles devem ainda aprender os interesses dos utilizadores, através de técnicas de aprendizagem automática.

A utilização de agentes em Comércio Electrónico permite a criação de mercados virtuais, onde os utilizadores delegam tarefas nos seus agentes, que se encarregam de as executar. Em vez de compradores e vendedores, podemos ter agentes de "software" compradores e agentes "software" vendedores que comunicam e negoceiam automaticamente entre si [Terpsidis *et al.* (1997)].

O comportamento do consumidor no processo de compra foi já estudado e analisado, tendo sido representado no modelo *Consumer Buying Behavior (CBB)* [Guttman *et al.* (1998c)], que detalhamos na secção 2.2. A tecnologia baseada em agentes pode ser aplicada a uma ou várias etapas desse modelo.

2.1 Sistemas Multi-Agente

A *Inteligência Artificial Distribuída (IAD)* é um campo da inteligência artificial que se debruça sobre o estudo de entidades computacionais que interagem entre si, quer na perspectiva das suas capacidades internas quer na perspectiva da sua "sociabilização", incluindo a cooperação, levando à realização conjunta de objectivos predefinidos. Na IAD podem-se distinguir duas áreas principais, que estão intimamente relacionadas [O'Hare e Jennings (1996)]:

- *Resolução Cooperativa e Distribuída de Problemas (RCDP)*: considera como a tarefa de resolver um problema particular pode ser distribuída por um conjunto de módulos, que cooperam na divisão e partilha de conhecimento acerca do problema e da sua solução.
- *Sistemas Multi-Agente (SMA)*: preocupa-se com o comportamento de uma colecção de agentes autónomos para a resolução de um determinado problema.

O trabalho inicial em IAD assentou na área de RCDP e assumiu um elevado grau de cooperação entre os agentes, de modo a alcançar um objectivo comum [Yokoo *et al.* (1992)]. O trabalho mais recente em SMA sobre mercados, por outro lado, assume agentes egoístas e competitivos durante negociações sobre recursos limitados. Tal não impede, no entanto, de considerar a interacção necessária entre tais agentes como uma forma de cooperação em que o objectivo principal do SMA é permitir as transacções comerciais.

2.1.1 Agente

Existem inúmeras definições de agente, umas mais elaboradas do que outras, introduzindo propriedades mais exigentes e que ao mesmo tempo são mais subjectivas: é o caso da autonomia e da inteligência, por exemplo.

Uma definição abrangente de agente, que se aplica tanto a agentes humanos como a robots e agentes de "software", é a seguinte: um agente é algo que obtém conhecimento do seu ambiente através de sensores e actua nesse ambiente através de efectuadores [Russell e Norvig (1995)]. Uma outra definição, que pretende sintetizar o que possa haver de comum em todas as outras definições, diz que agentes de "software" são entidades computacionais persistentes e activas que percebem, raciocinam, agem e comunicam num ambiente [Huhns e Singh (1997)]. Sistemas multi-agente são conjuntos de agentes associados pela interacção num ambiente comum.

Um agente caracteriza-se por um conjunto de propriedades de domínio contínuo [Huhns e Singh (1997)]. Algumas dessas propriedades são intrínsecas do agente: tempo de vida (efémero a permanente), nível de cognição (reactivo a deliberativo), implementação (declarativo a procedimental), mobilidade (estacionário a móvel), adaptabilidade (fixo, ensinável ou autodidacta) e modelação (do ambiente ou de outros agentes). Outras são extrínsecas ao agente, definidas no contexto da sua relação com os outros: localização (local ou remoto), autonomia social (independente a controlado), sociabilidade (autista, atento, responsável ou membro), colaboração (cooperativo, competitivo ou antagonista) e interacção (com o ambiente e outros agentes).

O grau de sofisticação de um agente está directamente ligado à forma como ele selecciona as acções a tomar. O agente pode simplesmente reagir a eventos que ocorrem

no mundo envolvente, utilizando um conjunto de regras "situação-acção" – é o caso extremo de um *agente reactivo*. A tomada de decisão pode implicar um conhecimento prévio do mundo, que será memorizado pelo agente. Além disso, o agente pode ser gerido por objectivos, fazendo depender a escolha da acção a tomar do destino pretendido. O agente pode ainda basear as suas acções em funções de utilidade (medidas de satisfação para o agente). Neste extremo temos o *agente deliberativo* por excelência. O grau de sofisticação nem sempre é directamente proporcional ao desempenho do agente, pois em certos domínios (nomeadamente da robótica) um agente reactivo pode obter melhores resultados do que um agente dito deliberativo.

Uma noção de agente mais elaborada inclui na sua definição atitudes de informação acerca do mundo (conhecimento e crença) e pró-atitudes (intenções e obrigações) que guiam as acções do agente. Este tem a chamada arquitectura "BDI", onde a estrutura tripla <crença ("belief"), desejo ("desire"), intenção ("intention")> toma um papel activo no processo cognitivo do agente [O'Hare e Jennings (1996)].

A arquitectura típica básica de qualquer agente divide-se, de forma mais ou menos evidente, em duas partes:

- *sistema inteligente*: é a parte responsável por resolver os problemas do domínio da sua especialidade;
- *camada de cooperação*: é a parte que permite ao agente comunicar e interagir com outros agentes do sistema multi-agente em que está inserido.

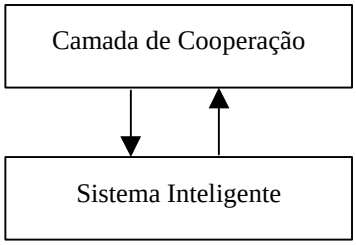


Figura 2.1 Arquitectura básica de um agente

O sistema inteligente é o componente que dá ao agente um carisma autónomo na resolução de problemas, capaz de tomar decisões inteligentes no que diz respeito à

execução das suas tarefas. A camada de cooperação permite que os agentes troquem mensagens entre si, por forma a interagirem.

2.1.2 Comunicação

Em sistemas multi-agente são usadas duas estratégias principais para suportar a comunicação entre agentes [O'Hare e Jennings (1996)]:

- troca directa de mensagens entre os agentes;
- troca indirecta de mensagens através de um repositório de dados partilhado – é o chamado *"blackboard"*.

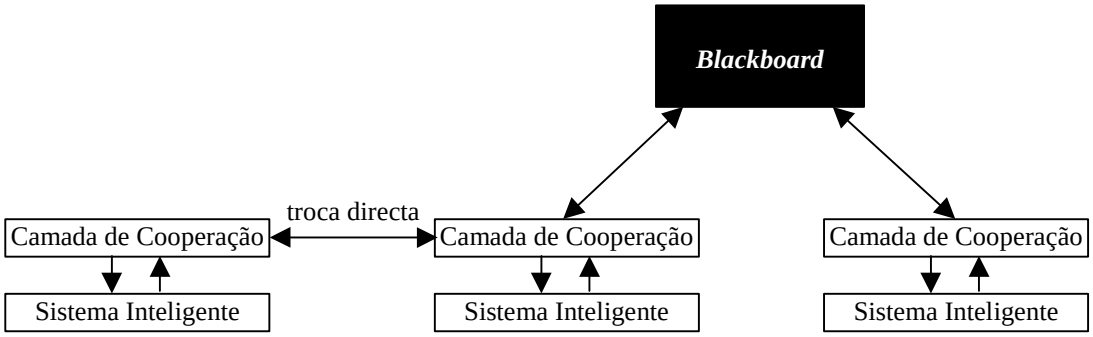


Figura 2.2 Comunicação entre agentes

Conceptualmente, o "blackboard" é um repositório onde os agentes escrevem mensagens e buscam informação. Esta informação encontra-se muitas vezes hierarquizada por níveis no "blackboard". Ao nível implementacional, contudo, este local pode funcionar como um agente auxiliar, com o qual os agentes comunicam de uma forma directa.

Os dois sistemas de comunicação podem ser combinados em sistemas complexos. Por exemplo, cada agente pode ser composto de vários subsistemas, que trocam informação utilizando um "blackboard" local; os agentes comunicam uns com os outros através da troca directa de mensagens.

A negociação entre agentes é a troca de mensagens, baseada num dado protocolo, com vista ao estabelecimento de um acordo. Um dos protocolos mais estudados para negociação foi inspirado nos processos contratuais das organizações humanas: é a *Rede Contratual* (*Contract Net*) [Smith (1988)]. Nesta, os agentes coordenam as suas

actividades através de contratos para atingir determinados objectivos. Um agente organizador anuncia tarefas para o grupo de agentes contraentes potenciais. Estes respondem ao agente organizador com licitações para a execução das tarefas. O agente organizador avalia as propostas e escolhe a melhor, celebrando depois o contrato.

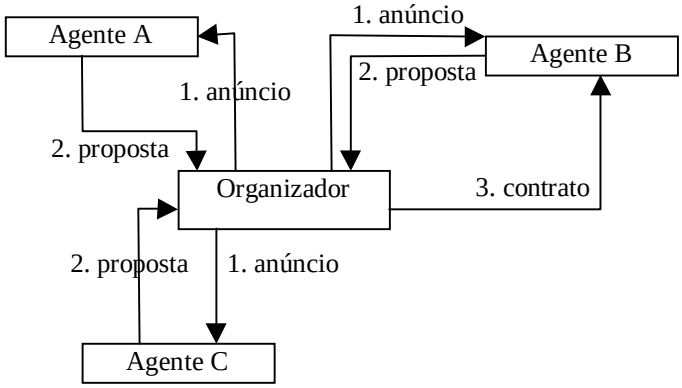


Figura 2.3 Protocolo da Rede Contratual

Seja para cooperar na prossecução de um determinado objectivo, seja para negociar de uma forma competitiva sobre um determinado recurso, os agentes têm necessidade de comunicar entre si. Para que essa comunicação seja possível, é imperioso que eles compreendam e utilizem uma linguagem comum.

Um dos grandes desafios para a compreensão entre agentes é que eles podem ter diferentes sistemas de crenças [Huhns e Singh (1997)]. Por exemplo, os agentes podem usar diferentes termos para o mesmo conceito. A utilização de uma representação comum é essencial para que a comunicação e coordenação tenham sucesso. Em agentes computacionais, isto representa uma *ontologia comum* – uma representação do conhecimento de um dado domínio que é partilhada por todos os agentes. Esta partilha de ontologias permite uniformizar o conteúdo semântico do vocabulário utilizado.

A solução para o problema da interoperação entre agentes construídos por diferentes entidades passa pela definição de uma linguagem comum, normalmente referida como *Agent Communication Language (ACL).* Um dos trabalhos mais importantes nesta área é o desenvolvimento de uma linguagem para a troca de informação e conhecimento: a *Knowledge Query Manipulation Language (KQML),* desenvolvida pelo consórcio *ARPA Knowledge Sharing Effort.*

Conceptualmente, a linguagem KQML está dividida em três níveis [Finin *et al.* (1994, 1997)]:

- *conteúdo*: é o conteúdo da mensagem propriamente dita, na linguagem de representação própria dos agentes – o KQML pode incluir qualquer linguagem de representação, sendo o conteúdo ignorado no que diz respeito à sua interpretação (esta tarefa cabe ao agente receptor);
- *mensagem*: encobre o conteúdo da mensagem que um agente quer transmitir a outro – este nível forma o cerne da linguagem KQML e determina os tipos de interacção que os agentes podem ter;
- *comunicação*: acrescenta alguns parâmetros à mensagem relacionados com a comunicação de baixo nível, como a identidade do remetente e do destinatário e um identificador único associado à comunicação.

Uma mensagem KQML consiste numa "*performativa*" e nos seus argumentos, incluindo o conteúdo da mensagem e um grupo de argumentos opcionais, que o descrevem de uma forma independente da sintaxe da linguagem de conteúdo [Finin *et al.* (1994)]. A "performativa" indica qual a acção a tomar pelo agente receptor quando recebe a mensagem.

Mais recentemente, a FIPA[1] (*Foundation for Intelligent Physical Agents*) promoveu uma especificação de uma linguagem ACL, na tentativa de a padronizar.

2.1.3 Plataformas

Existem algumas plataformas desenvolvidas para facilitar a implementação de sistemas multi-agente. Estas plataformas providenciam uma infra-estrutura dentro da qual os agentes operam e interagem. Um dos aspectos principais destas infra-estruturas é o seu suporte para a comunicação entre agentes.

Algumas dessas plataformas são as seguintes:

- Agent Building Environment (ABE)[2]

[1] URL: http://www.fipa.org/

[2] URL: http://badaam.csc.ncsu.edu/research/Abe/index.htm

- JATLite[3]
- Open Agent Architecture (OAA)[4]
- Voyager[5]

A plataforma ABE, desenvolvida pela IBM, pretende assistir o problema de integrar agentes inteligentes com aplicações existentes e novas. A ênfase está em dar resposta à tendência de aparecimento de agentes inteligentes em diversas áreas mas que não comunicam entre si, propondo uma arquitectura geral para a construção de aplicações baseadas em agentes. No ABE, o modelo do agente utiliza uma base de conhecimento assente em regras e um motor que lhe permite raciocinar e aprender. O agente possui interfaces com o utilizador e com outras aplicações, através de *adaptadores*. Cada adaptador é responsável por trabalhar com um domínio aplicacional específico, tipo de dispositivo ou protocolo (por exemplo, interfaces com a "Web", correio electrónico, etc.).

O JATLite, desenvolvido na Universidade de Stanford, é um conjunto de programas escritos na linguagem Java, que possibilitam a criação rápida de agentes de "software" que comunicam de uma forma robusta pela Internet. O JATLite facilita especialmente a construção de agentes que enviam e recebem mensagens usando a linguagem de comunicação KQML, que tende à padronização. Contudo, o JATLite não impõe nos agentes nenhuma teoria ou técnica de construção de agentes autónomos. A construção de "agentes inteligentes" é da responsabilidade do programador. A infra-estrutura do JATLite assenta no chamado *Agent Message Router (AMR)*. O AMR é uma aplicação Java que corre numa máquina ligada à Internet. Tem como funcionalidades principais o redireccionamento de mensagens entre os agentes e a manutenção da informação relativa aos endereços desses mesmos agentes. Os agentes podem ser portáteis ou móveis, embora o JATLite não assista a inclusão de mobilidade nos agentes. Eles registam-se no AMR usando um nome e "password", ligam-se e desligam-se de qualquer lugar da Internet e enviam e recebem mensagens.

[3] URL: http://java.stanford.edu/

[4] URL: http://www.ai.sri.com/~oaa/

[5] URL: http://www.objectspace.com/products/vgrOverview.htm

A OAA, desenvolvida pela SRI, permite a construção de agentes utilizando uma linguagem de comunicação inter-agente declarativa, baseada em lógica (*InterAgent Communication Language – ICL*). O enfoque está em construir comunidades distribuídas de agentes, onde estes são definidos como quaisquer processos de "software" que vão de encontro às convenções da sociedade OAA. Um agente satisfaz este requisito registando os serviços que pode fornecer de uma forma aceitável, sendo capaz de falar ICL e partilhando funcionalidades comuns a todos os agentes OAA, como a habilidade de disparar eventos, manipular dados de uma determinada maneira, etc.. Um agente especial, chamado *facilitador*, distribui tarefas pelos agentes.

A plataforma Voyager foi desenvolvida pela ObjectSpace. O seu propósito é o suporte a objectos móveis e agentes autónomos. A filosofia do Voyager assenta na noção de que um agente é simplesmente um tipo especial de objecto, que se pode mover independentemente, que pode continuar a executar enquanto se move e que de resto se comporta como qualquer outro objecto. Esta plataforma está implementada em Java, tirando partido de todas as vantagens de utilização desta linguagem. O Voyager providencia um *Object Request Broker (ORB)* para agentes construídos em Java, suportando alguns dos padrões mais comuns, como CORBA, RMI e DCOM. Um ORB é uma tecnologia intermediária que gere a comunicação e a troca de dados entre objectos. Outras plataformas semelhantes a esta, no que diz respeito à utilização de objectos e mobilidade, são a Odyssey da General Magic, a Aglets da IBM e a Concordia da Mitsubishi.

No presente trabalho foi utilizada a plataforma JATLite. As vantagens da sua utilização para o desenvolvimento pretendido estão descritas na secção 6.4.

2.2 O Modelo *Consumer Buying Behaviour (CBB)*

O modelo CBB propõe uma divisão do processo de transacção em seis etapas fundamentais [Guttman *et al.* (1998c)]:

- *Identificação da Necessidade (Need Identification)*: nesta etapa o consumidor apercebe-se de uma nova necessidade. Pode ser estimulado através da informação sobre produtos existentes.

- *Selecção do Produto (Product Brokering)*: consiste na recolha de informação com vista a determinar *o que* comprar. O consumidor avalia as alternativas de produtos baseado em critérios pessoais.
- *Selecção do Vendedor (Merchant Brokering)*: esta etapa combina o resultado da etapa anterior com informação específica sobre vendedores, para determinar *a quem* comprar o produto. O consumidor avalia as alternativas de acordo com critérios como o preço, garantia, disponibilidade, assistência, reputação, etc..
- *Negociação (Negotiation)*: consiste em *como* determinar os termos da transacção. A negociação varia em duração e complexidade de acordo com o mercado. Em mercados de retalho tradicionais, o preço e outros aspectos da transacção são muitas vezes fixos, não havendo por isso lugar a qualquer negociação.
- *Encomenda e Entrega (Purchase and Delivery)*: por vezes esta etapa finaliza a negociação, chegando ambas as partes a um acordo. Outras vezes, a compra e entrega do produto ocorre algum tempo depois (de acordo com as condições de pagamento e com a disponibilidade do produto).
- *Serviço e sua Avaliação (Service and Evaluation)*: esta etapa envolve o serviço prestado ao cliente e a sua avaliação sobre o produto que adquiriu.

Estas etapas representam uma aproximação e simplificação do comportamento do consumidor, não sendo um modelo rígido e sequencial.

Na perspectiva do modelo CBB, podem-se identificar os papéis dos agentes como mediadores no Comércio Electrónico. As características dos agentes (personalizados, semi-autónomos e inteligentes) tornam-nos apropriados para mediar estes comportamentos do consumidor, envolvendo a obtenção e filtragem de informação, avaliações personalizadas e coordenações/interacções com outros agentes. Estes papéis correspondem, principalmente, às etapas de *Selecção do Produto*, *Selecção do Vendedor* e *Negociação* do modelo CBB.

A tabela seguinte [Guttman *et al.* (1998c)] mostra as etapas de acção de alguns dos sistemas baseados em agentes existentes: *PersonaLogic*, *Firefly*, *BargainFinder*, *Jango*, *Kasbah*, *AuctionBot* e *Tete-a-Tete*.

	Persona Logic	*Firefly*	*Bargain Finder*	*Jango*	*Kasbah*	*Auction Bot*	*Tete-a-Tete*
Selecção do Produto	✓	✓		✓			✓
Selecção do Vendedor			✓	✓	✓		✓
Negociação					✓	✓	✓

Tabela 2.1 Sistemas baseados em agentes e o modelo CBB

Os agentes são capazes de examinar rapidamente um grande número de produtos antes de tomar uma decisão de compra. Além de automatizar o processo de busca de informação, eles podem também negociar com os vários vendedores de um produto.

2.2.1 Selecção do Produto

Na etapa de *Selecção do Produto*, o consumidor pretende identificar o que comprar.

PersonaLogic[6] é uma ferramenta que permite aos consumidores limitar a gama de produtos que melhor correspondem às suas necessidades, a partir de um grande espaço de características de produtos. O consumidor utiliza estas características restringindo os valores que são desejáveis no produto que procura. Um motor de satisfação de restrições ordena depois uma lista de todos os produtos que obedecem às condições indicadas, utilizando para isso um conjunto de "restrições leves", indicadas pelo utilizador como preferências não obrigatórias.

Firefly[7] é um outro sistema que também auxilia o utilizador a encontrar os produtos que procura. Este sistema faz recomendações baseado em opiniões fornecidas por outros utilizadores com interesses semelhantes. Este mecanismo tem o nome de "*automated collaborative filtering*".

[6] URL: http://www.personalogic.com/

[7] URL: http://www.firefly.com/

2.2.2 Selecção do Vendedor

Na etapa de *Selecção do Vendedor*, comparam-se as diferentes alternativas, em termos de vendedores. Obviamente que um vendedor preferia que os consumidores comprassem apenas no seu sítio na "Web". Contudo, os consumidores comparam os produtos de vários vendedores, o que tem vindo a ser facilitado por agentes de "software".

BargainFinder[8] é um agente que automatiza a busca dos preços de um determinado produto a vários vendedores, apresentando-os ao utilizador. Este tipo de automatização faz com que o único critério utilizado para a decisão do consumidor seja o preço. Alguns vendedores, que não pretendem competir baseando-se apenas no preço, bloquearam os pedidos do BargainFinder.

Jango[9] é uma optimização do BargainFinder que ultrapassa o problema do bloqueamento por parte dos vendedores, fazendo com que os pedidos sejam feitos do navegador da "Web" do utilizador.

Segundo Guttman e Maes (1998a), o aparecimento destes sistemas coloca problemas a ambas as partes do processo de compra. Se, por um lado, os vendedores não pretendem competir apenas baseados no preço, preferindo diferenciar-se através de outros aspectos inerentes à compra do produto, por outro lado os consumidores podem ser induzidos em erro quando são informados por estes sistemas. Por exemplo, um dos vendedores contactado, embora tendo um preço base mais elevado, poderia estar a fazer uma promoção, fazendo descontos na compra em quantidade. Um agente que ajude verdadeiramente o consumidor a realizar a melhor compra possível, deve considerar outros factores para além do preço, como a garantia, a assistência, modalidades de pagamento, etc., de acordo com as suas prioridades.

A diferenciação é necessária para um vendedor ter poder sobre o mercado, isto é, para que ele possa aumentar os preços dos seus produtos acima dos seus custos marginais. De outra forma, num mercado altamente competitivo, retalhistas e outros intermediários são obrigados a competir sobre os seus custos marginais, obtendo por isso pouco lucro

[8] URL: http://bf.cstar.ac.com/bf

[9] URL: http://www.jango.com/

[Guttman e Maes (1998a)]. A primeira geração de agentes de "software" que se destinam a comparar diversos vendedores baseia-se apenas no preço como termo de comparação. Isto torna bastante difícil a tarefa de diferenciação por parte dos vendedores.

Kasbah[10] é um sistema "on-line" de "anúncios classificados", no qual pessoas interessadas em comprar (vender) produtos criam agentes compradores (vendedores), que entram em contacto com agentes vendedores (compradores). A procura de agentes com interesses complementares é feita de forma autónoma, depois dos agentes estarem devidamente parametrizados pelo utilizador. A etapa de *Selecção do Vendedor* desenrola-se a par da etapa de *Negociação*, que é feita baseada no preço do produto. Agentes compradores e vendedores negoceiam entre si, tentando cada um deles obter o melhor acordo possível [Guttman *et al.* (1998c)].

Tete-a-Tete[11] é um novo sistema onde se pretende incluir funções multi-critério para avaliação de propostas no processo de compra, abrindo de novo a possibilidade da diferenciação. O sistema ordena as ofertas individuais dos vendedores de acordo com o seu valor global para o cliente.

2.2.3 Negociação

Na etapa de *Negociação*, o preço e outros termos da transacção são determinados. Negociar dinamicamente o preço de um produto em vez de o fixar livra o vendedor de ter que determinar o valor do produto *a priori*. Esta tarefa fica entregue ao próprio mercado. O resultado é que recursos (produtos) limitados são atribuídos de uma forma justa, isto é, a quem os valoriza mais [Guttman e Maes (1998a, 1998b), Guttman *et al.* (1998c)]. Os leilões constituem casos típicos de negociação, sendo abordados na secção 3.2.

Há alguns obstáculos à utilização da negociação. Certos tipos de leilões requerem que todas as pessoas estejam no mesmo local. Além disso, a negociação pode ser demasiado complicada ou frustrante para o consumidor médio. Finalmente, alguns protocolos de

[10] URL: http://kasbah.media.mit.edu/

[11] URL: http://ecommerce.media.mit.edu/Tete-a-Tete/

negociação prolongam-na durante um longo período de tempo, o que não se adequa a consumidores sem disponibilidade para isso [Guttman *et al.* (1998c)].

Alguns destes impedimentos desaparecem no mundo digital. Existem diversos leilões instalados em sítios na "Web", e que portanto não requerem que os participantes estejam todos no mesmo local. No entanto, estes sítios requerem que os consumidores conduzam as suas próprias estratégias de negociação, por vezes durante longos períodos de tempo. É neste ponto que os agentes de "software" podem ter um papel importante.

Existem já alguns sistemas de agentes que assistem os consumidores no processo de negociação.

AuctionBot[12] é um servidor de leilões. Os utilizadores criam novos leilões para vender produtos, escolhendo de entre um leque de tipos de leilões disponíveis e especificando os seus parâmetros. Vendedores e compradores podem licitar de acordo com os protocolos do leilão criado. Além disso, através de uma API disponibilizada pelo AuctionBot, os utilizadores podem criar agentes de "software" que, autonomamente, competem no mercado criado. Cabe aos utilizadores codificar as suas próprias estratégias de licitação. O sistema Fishmarket[13] é de algum modo semelhante ao AuctionBot, mas admitindo apenas um tipo de leilão: o *leilão holandês*. Tal como no AuctionBot, no Fishmarket os utilizadores constróem as suas estratégias de licitação nos seus agentes. Este sistema não está a ser utilizado no mundo real, sendo usado como plataforma de testes para agentes com diversas estratégias, através da organização de torneios.

Kasbah é um sistema multi-agente, em que os agentes automatizam as etapas de *Selecção do Vendedor* e *Negociação* do modelo CBB, tanto para compradores como para vendedores. Após o encontro entre compradores e vendedores, a única acção válida no protocolo de negociação é, para os compradores, licitar e para os vendedores responder "sim" ou "não" [Guttman *et al.* (1998c)]. De acordo com a sua estratégia de negociação, os compradores aumentam o seu valor de licitação e os vendedores baixam o valor pedido. Há três estratégias disponibilizadas para utilização nos agentes, que

[12] URL: http://auction.eecs.umich.edu/

[13] URL: http://www.iiia.csic.es/Projects/fishmarket/

correspondem a funções lineares, quadráticas e exponenciais para a alteração dos preços licitados/pedidos ao longo do tempo.

Tete-a-Tete é um sistema que permite aos agentes negociar segundo vários critérios, como a garantia, tempos de entrega, condições de pagamento e outros aspectos diferenciadores. A negociação entre os agentes é feita tendo em conta uma função de utilidade multi-atributo, que é usada para medir o grau de satisfação da oferta de cada vendedor. Na avaliação das propostas dos vendedores são usadas as restrições obtidas nas primeiras etapas do modelo CBB. Essencialmente, o Tete-a-Tete integra as etapas de *Selecção do Produto*, *Selecção do Vendedor* e *Negociação* [Guttman *et al.* (1998c)].

3. Negociação

O tipo de relacionamento entre os agentes, no que diz respeito aos seus objectivos, pode ir de completamente cooperativo a antagónico [O'Hare e Jennings (1996)]. Agentes cooperativos são capazes de resolver problemas interdependentes. O objectivo global do sistema multi-agente está, neste caso, distribuído pelos agentes que dele fazem parte. Por outro lado, agentes antagónicos não cooperam e podem até bloquear os objectivos uns dos outros. No meio deste espectro há lugar à negociação, um processo pelo qual os agentes comunicam de uma certa forma para chegar a uma decisão comum.

Embora a negociação seja importante na modelação de sistemas multi-agente, não há uma definição clara e comum do que é a negociação [O'Hare e Jennings (1996)]. A ideia comum, em todas as contribuições no seio da IAD, é que os agentes usam a negociação para a resolução de conflitos e coordenação entre eles.

O recente acréscimo de interesse em agentes de "software" autónomos e interactivos, e a sua aplicação potencial em áreas como o Comércio Electrónico, aumentou a importância da negociação automática [Zeng e Sycara (1996, 1998)].

3.1 Negociação Competitiva vs. Cooperativa

A literatura sobre negociação em transacções comerciais define dois tipos de negociação [Guttman e Maes (1998a)]:

- *Competitiva ou Distribuída*: consiste num processo de decisão em que as duas (ou mais) partes envolvidas competem por um único objectivo mutuamente exclusivo. Portanto, à medida que uma das partes melhora a sua utilidade a outra piora (por esta razão, a negociação é referida como sendo do tipo "win-lose"). A teoria de jogos descreve esta situação como um "jogo de soma zero".
- *Cooperativa ou Integrada*: é um processo de decisão em que as partes envolvidas negoceiam sobre vários objectivos interdependentes mas não mutuamente exclusivos. À medida que as várias dimensões da transacção são negociadas e se deslocam em várias direcções, é possível que todas as partes melhorem a sua

utilidade (a negociação é do chamado tipo “win-win”). A teoria de jogos classifica este tipo de negociação como um “jogo de soma diferente de zero”.

Exemplos de negociação competitiva podem ser encontrados em vários tipos de leilões. No sistema AuctionBot é possível haver leilões de vários tipos a decorrer simultaneamente. Um desses tipos é o chamado *continuous double auction (CDA)*, onde são admitidos múltiplos compradores e vendedores ao mesmo tempo, negociando multi--lateralmente. O sistema Kasbah também se aproxima deste tipo de leilões, mas a negociação é feita de uma forma bilateral, sendo melhor definido como um sistema de anúncios classificados com negociação [Guttman e Maes (1998b)].

Mesmo em negociações competitivas, as entidades envolvidas necessitam de cooperar suficientemente, de forma a levar a cabo a própria negociação e a acordar nas semânticas dos protocolos de negociação utilizados.

Os objectivos dos vendedores são o lucro a longo prazo, vendendo o maior número de produtos possível, ao maior número de clientes possível, o mais caro possível e com o mínimo de custos de transacção. Os objectivos principais dos consumidores são satisfazer as suas necessidades através da compra de produtos apropriados a vendedores apropriados, o mais barato possível e com o mínimo de custos de transacção. A negociação cooperativa pode ajudar a maximizar ambos os conjuntos de objectivos [Guttman e Maes (1998a, 1998b)].

Existem ferramentas para analisar quantitativa e qualitativamente a tomada de decisões envolvendo múltiplos objectivos interdependentes. A teoria da utilidade multi-atributo (*Multi-Attribute Utility Theory - MAUT*) analisa preferências com múltiplos atributos. Esta teoria tem sido aplicada em vários tipos de problemas de decisão. Um sistema que utiliza MAUT é o *Persuader* [Sycara (1992)], usado para resolver conflitos através da negociação em problemas de grupo. Outro exemplo é o *Logical Decisions for Windows (LDW)*[14], uma ferramenta que ajuda as pessoas a pensar e analisar os seus problemas. A teoria MAUT pode ajudar os consumidores a tomar decisões de compra. Pode também ajudar os vendedores na criação de políticas de valorização dos seus produtos, considerando o valor acrescentado proveniente da sua diferenciação [Guttman e Maes (1998a, 1998b)].

A teoria MAUT analisa quantitativamente os problemas de decisão, através de utilidades. Os problemas de satisfação de restrições (*Constraint Satisfaction Problems – CSP*) requerem uma análise de índole qualitativa, através de combinações de restrições. Um CSP é formulado através de variáveis, domínios e restrições. A tarefa de um motor de CSP é atribuir valores às variáveis, ao mesmo tempo que garante a satisfação das restrições. Pode haver também a possibilidade de definir "restrições leves", que não precisam de ser verificadas, mas que são utilizadas na ordenação das soluções. Há muitos problemas que podem ser formulados como CSPs, tais como escalonamento, planeamento, etc.. Em mercados por retalho, técnicas de CSP podem ser usadas para definir "restrições pesadas" e "restrições leves". Um exemplo desse uso encontra-se no sistema PersonaLogic, que auxilia o utilizador na etapa de *Selecção do Produto* do modelo CBB. Um aspecto importante dos CSPs é que eles conseguem explicar porque tomam determinadas decisões (como nos sistemas periciais), o que conduz à confiança do utilizador no processo de decisão.

Sistemas como o PersonaLogic podem ser alargados, de forma a cobrir também as etapas de *Selecção do Vendedor* e *Negociação* do modelo CBB. Este alargamento pode ser feito utilizando técnicas de satisfação de restrições distribuídas (*Distributed Constraint Satisfaction Problems - DCSP*). DCSPs são semelhantes a CSPs, com a diferença de que as variáveis e restrições estão distribuídas pelos agentes [Yokoo *et al.* (1992), Yokoo e Hirayama (1998)] (no caso do comércio a retalho, consumidores e vendedores).

O sistema Tete-a-Tete combina técnicas de MAUT e DCSP para mediar as negociações entre agentes compradores e vendedores [Guttman e Maes (1998b)], baseando a negociação não apenas no preço de um produto, mas num conjunto de dimensões que são consideradas no processo de compra.

3.2 Leilões Electrónicos

A negociação na Internet resume-se muitas vezes a uma das partes (tipicamente o vendedor) apresentar uma proposta de pegar ou largar (isto é, o preço de venda) – é o que normalmente acontece no comércio por retalho, em que os preços são fixos. Os

[14] URL: http://www.logicaldecisions.com/

leilões são um espaço privilegiado para o estudo de protocolos de negociação entre agentes. Consistem numa abordagem mais geral à determinação do preço, admitindo diversos protocolos de negociação, incluindo o preço fixo como um caso especial [Wurman *et al.* (1998b)].

Um leilão é uma instituição de mercado, com um conjunto de regras, onde se determinam atribuições de recursos e preços, a partir de licitações dos participantes do leilão. Os leilões consistem em procedimentos formalizados para transacções comerciais, nos quais a interacção dos participantes é governada por regras específicas [Romm e Sudweeks (1998)]. Na maior parte dos casos, um leiloeiro funciona como intermediário.

Existem diversos tipos de leilões, cujas diferenças se baseiam nas regras utilizadas para a participação dos intervenientes:

- *leilão inglês*: neste tipo de leilão os compradores vão licitando ofertas crescentes, terminando quando existe apenas uma proposta;
- *leilão holandês*: o vendedor anuncia o valor desejado de venda e vai decrescendo esse valor gradualmente, até que um comprador aceita o valor actual;
- *leilão FPSB (First Price Sealed Bid)*: é um leilão fechado, em que os compradores submetem uma proposta que dão a conhecer apenas ao vendedor; não há mais iterações, ganhando o comprador que ofereceu a proposta mais elevada, que deve pagar o valor que anunciou;
- *leilão Vickrey*: semelhante ao anterior, mas em que o vencedor deve pagar o valor da segunda maior proposta.

Todos os mecanismos atrás descritos são chamados “single-sided”, pois os licitadores são uniformemente do tipo comprador (vendedor), havendo portanto um só vendedor (comprador) que aceita as licitações de múltiplos compradores (vendedores). Há outros tipos de leilões, chamados “double-sided”, que admitem múltiplos compradores e vendedores ao mesmo tempo [Wurman *et al.* (1998a)]:

- *continuous double auction (CDA)*: compradores e vendedores transaccionam produtos assim que as suas licitações (de compra e de venda) se tornam compatíveis;

- *clearing house* ou *call market*: estes mercados são versões periódicas do CDA, aceitando licitações durante um período de tempo após o qual reiniciam.

Os leilões "on-line" estão cada vez mais populares, devido ao entretenimento que originam no processo de negociação dos preços dos produtos e à possibilidade de fazer bons negócios na compra dos produtos pretendidos. O número de leilões presentes na Internet é cada vez maior, sendo muitos deles prolongamentos de sítios de venda a retalho, em que os retalhistas desempenham o papel duplo de vendedores e leiloeiros.

Segundo Guttman e Maes (1998a, 1998b), apesar da crescente utilização deste tipo de Comércio Electrónico, os leilões "on-line" existentes têm características que fazem deles pouco compensadores, tanto para compradores como para vendedores. Embora alguns dos protocolos utilizados em leilões "on-line" sejam relativamente simples de compreender e utilizar, a determinação da melhor estratégia de licitação não é trivial. De facto, no leilão inglês, por exemplo, a licitação vencedora é quase sempre mais elevada do que o valor de mercado do produto. Este problema é conhecido como "a maldição do vencedor", sendo exacerbado em leilões onde os compradores valorizam os produtos por questões pessoais, tendendo irracionalmente a elevar as suas licitações acima do verdadeiro valor do produto. Embora a curto prazo isto possa representar um benefício para os vendedores, pode transformar-se rapidamente num problema, devido à possível insatisfação do cliente por ter pago mais do que o valor do produto. Há duas regras universais dos leilões que levam a isto: as licitações não são retiráveis e os produtos não podem ser devolvidos. Outro problema que leva à insatisfação do consumidor é o longo período que vai desde o início da fase de negociação até ao fim da fase de compra e entrega do produto, no modelo CBB. Alguns leilões na Internet podem demorar dias até que um licitador seja dado como vencedor. Além disso, dado que há apenas um vencedor, os outros consumidores terão que esperar até que o produto seja leiloado outra vez, reiniciando o processo de negociação. Outro problema que surge nos leilões é a utilização de "shills", que são licitadores ilegalmente criados por vendedores para viciar o jogo, estimulando o mercado através do aumento da sua licitação. Este tipo de mecanismo pode ser difícil de detectar, principalmente no mundo virtual.

Surgem também alguns problemas para os vendedores. Embora nos leilões estes não tenham que fixar o preço dos produtos que pretendem leiloar, quando se trata de bens de

produção eles têm que determinar a quantidade dos bens a leiloar e a frequência dos seus leilões. Isto requer um conhecimento da procura do produto, afectando directamente a gestão de inventários e os planos de produção. Os compradores também podem viciar o jogo, formando coligações, que são grupos de compradores que acordam não licitar mais do que um determinado valor. Desta forma, podem ser comprados produtos mais baratos do que se os compradores estivessem a competir entre si. Tal como os "shills", as coligações podem ser difíceis de detectar, especialmente em mercados "on-line", onde os compradores são virtuais.

Na maior parte dos leilões "on-line" existentes, que consistem em mecanismos "single-sided", em vez de vendedores competirem entre si pela preferência dos clientes, os consumidores é que competem por ofertas específicas dos vendedores. Por outro lado, dado que a negociação nesses leilões se baseia no preço do produto, os vendedores perdem a oportunidade de se diferenciarem, na fase de *Selecção do Vendedor* do modelo CBB.

3.3 Outros Estudos

Em Shmeil e Oliveira (1995) as organizações são modeladas através de agentes. Este trabalho foca um tipo bem definido de interacções: as que ocorrem entre agentes representando organizações produtoras e fornecedoras. O intento dos produtores é encontrar o fornecedor mais adequado para um produto ou serviço. Uma acção cooperativa é estabelecida quando um fornecedor celebra um contrato com um produtor. Este contrato é o resultado de um processo de selecção, que consiste na publicação de um anúncio por parte do produtor, seguido da avaliação das propostas recebidas dos fornecedores. O contrato é celebrado com o primeiro fornecedor que satisfaça as restrições do produtor. Este modelo de negociação é baseado na *Rede Contratual* (*Contract Net*) [Smith (1988)].

Em Faratin *et al.* (1998) a negociação é definida como um processo pelo qual uma decisão conjunta é tomada por duas ou mais partes. As partes começam por exprimir exigências contraditórias e chegam a acordo através de um processo de concessões e de procura de novas alternativas [Pruitt (1981)]. O trabalho desenvolvido em Faratin *et al.* (1998) debruça-se sobre a negociação "orientada ao serviço": um agente (o cliente)

requer a execução de um serviço a outro agente (o servidor). A negociação envolve a determinação dos termos e condições do contrato. O modelo de negociação adoptado é multi-lateral (como um CDA) e multi-dimensão. A negociação decorre através da troca de propostas entre os agentes, que vão concedendo nos valores pedidos/oferecidos utilizando diversos critérios, como o tempo, a disponibilidade de recursos ou o comportamento dos oponentes de negociação.

4. Mercados Electrónicos baseados em Agentes

Existem já alguns sistemas multi-agente funcionais, que permitem automatizar grande parte do processo de compra e venda de produtos. Para facilitar o estudo de mecanismos de negociação entre os agentes que compõem os sistemas multi-agente, alguns deles baseiam-se em leilões, espaço privilegiado para a implementação da negociação.

Os sistemas mais significativos, que podem ser utilizados para investigação ou para transacções no mundo real, são o *AuctionBot*, o *Fishmarket*, o *Kasbah* e o *Tete-a-Tete*, os quais descreveremos neste capítulo.

4.1 AuctionBot

O sistema AuctionBot, desenvolvido na Universidade de Michigan, é um servidor de leilões flexível, escalável e robusto, que suporta tanto agentes de "software" como agentes humanos [Wurman *et al.* (1998b)]. O sistema gere múltiplos leilões que poderão decorrer simultaneamente, admitindo vários tipos de leilões. Esta flexibilidade é alcançada decompondo o espaço dos leilões num conjunto de parâmetros ortogonais. Através destes parâmetros, o AuctionBot consegue implementar muitos dos tipos de leilões clássicos e mais alguns que nunca tinham sido estudados anteriormente.

Os leilões activos no AuctionBot são organizados num catálogo hierárquico, facilitando a pesquisa para os potenciais compradores que neles desejam licitar. Os utilizadores que pretendam criar novos leilões podem posicioná-los em qualquer secção do catálogo existente, ou expandir o catálogo, criando uma subcategoria apropriada. Têm também a opção de não listar o leilão no catálogo público, mantendo a sua visibilidade limitada a um grupo privado. Para participar em qualquer leilão, os utilizadores têm que se registar. Em qualquer momento, eles podem inspeccionar as suas contas através de uma página "Web", que apresenta uma visão organizada das suas licitações, leilões que eles iniciaram e transacções anteriores.

O AuctionBot disponibiliza uma API que permite aos utilizadores a criação de agentes, com estratégias de licitação por eles concebidas, que agem autonomamente. A interface com estes agentes de "software" é assegurada através de comunicação TCP/IP.

Os humanos interagem com o sistema via uma interface "Web". A interacção entre os agentes externos ("software" e humanos) e o sistema AuctionBot consiste na utilização de uma base de dados, onde os agentes colocam as suas licitações. Esta base de dados é monitorizada pelo leiloeiro, que processa os eventos dos diversos leilões e verifica as licitações pendentes.

O suporte de uma grande variedade de tipos de leilões, combinado com a API disponibilizada para a criação de agentes, faz do AuctionBot uma plataforma útil tanto para o Comércio Electrónico como para a investigação [Wurman *et al.* (1998b)].

4.2 Fishmarket

O Fishmarket é uma implementação de uma casa de leilões electrónica, inspirada na velha instituição do mercado da lota do peixe ("fish market") [Rodríguez-Aguilar *et al.* (1997)], tendo sido desenvolvido no *Artificial Intelligence Research Institute* de Barcelona. Neste sistema, agentes de "software" e humanos podem transaccionar. O Fishmarket não está a ser utilizado no mundo real, sendo usado como plataforma de testes de agentes com diversas estratégias, através da organização de torneios [Rodríguez-Aguilar *et al.* (1998a, 1998b)]. O protocolo utilizado no Fishmarket deriva directamente do modo como a venda de caixas de peixe é feita no mercado da lota do peixe. É o protocolo de "licitações descendentes", mais conhecido como *leilão holandês*.

No cenário do mercado da lota do peixe – um leilão em que o leiloeiro apresenta caixas de peixe, anunciando preços por ordem decrescente – existem vários agentes que desempenham funções bem definidas (agentes internos). Estes funcionam como intermediários, que o mercado utiliza para interagir com compradores e vendedores (agentes externos). Os agentes foram transferidos para a versão electrónica deste mercado, o Fishmarket. O *patrão do mercado* tem o papel de servidor de identidades, de supervisor do leilão e autoridade máxima. O *leiloeiro* encarrega-se do processo de licitação. Agentes vendedores registam os seus produtos num agente *admissor de vendedores*, recolhendo o seu dinheiro num *gestor de vendedores*, após o leiloeiro ter vendido os produtos no leilão. Os compradores registam-se num agente *admissor de*

compradores, e licitam produtos que pagam através de uma linha de crédito que é mantida pelo *gestor de compradores*.

Os agentes externos (compradores e vendedores) podem ter diversos graus de complexidade, cabendo aos utilizadores construí-los, podendo ser eles próprios agentes. Todos eles participam no Fishmarket exclusivamente através de interfaces de comunicação padronizados. Para cada agente externo existe um *interagent* [Martín *et al.* (1998)], que serve de intermediário de comunicação entre o agente e o resto do mercado. Agentes externos podem ser construídos em qualquer linguagem de programação que permita lançar o *interagent* como um processo filho, comunicando com ele pelo "standard input/output" e com o mercado via TCP/IP.

Os diversos componentes do mercado Fishmarket foram implementados em Java. Cada agente do mercado funciona como um processo com múltiplos "threads" de execução, que lhe permite responder a vários pedidos concorrentemente [Rodríguez--Aguilar *et al.* (1997)].

O Fishmarket permite avaliar o desempenho dos agentes compradores, através do registo de todas as suas intervenções no leilão. São organizados torneios em que participam diversos agentes compradores, podendo-se analisar comparativamente os seus resultados.

4.3 Kasbah

O Kasbah é um mercado virtual, criado no *MIT Media Laboratory*, no qual os utilizadores criam agentes predefinidos e nos quais delegam a compra e venda de produtos [Chavez e Maes (1996)]. O objectivo principal do sistema é permitir a criação de agentes que negoceiam autonomamente, procurando obter o melhor negócio possível para o utilizador. O Kasbah é um sistema de Comércio Electrónico baseado em mecanismos do tipo *continuous double auction (CDA)*.

No Kasbah, os agentes vendedores são como anúncios classificados. Quando um utilizador cria um agente vendedor, dá-lhe uma descrição do produto que pretende vender. Estes agentes são pró-activos, ou seja, tentam vender o produto entrando no mercado, contactando com compradores interessados e negociando com eles, de forma a encontrar o melhor negócio possível. Os agentes vendedores são autónomos, pois uma

vez largados no mercado eles negoceiam e tomam decisões, sem requerer a intervenção do utilizador. Ao criar um agente vendedor, o utilizador define alguns parâmetros para guiar o agente enquanto tenta vender o produto:

- data limite para a venda do produto;
- preço desejado;
- preço mais baixo aceitável.

Estes parâmetros definem o objectivo do agente: vender o produto em questão pelo valor mais alto possível – idealmente pelo preço desejado, mas podendo descer até ao preço mais baixo aceitável, tendo em conta a data limite especificada. A heurística utilizada pelos agentes vendedores com vista a atingir o seu objectivo é muito simples. O agente começa por pedir o preço desejado. Enquanto não houver compradores, o agente vai descendo gradualmente o preço. Ao chegar à data limite para a venda do produto (se ainda não houve compradores), o preço pedido pelo agente deve ser o preço mais baixo aceitável. O que vai determinar o desempenho do agente na prossecução do seu objectivo é a forma como ele vai baixando o preço pedido. O Kasbah disponibiliza três funções de decréscimo que o agente pode utilizar para baixar o preço pedido ao longo do tempo, e que no fundo são as estratégias de negociação que ele pode adoptar:

- *linear*: corresponde à implementação de um agente "ansioso" no que diz respeito à venda do produto, baixando o preço ao longo do tempo linearmente;
- *quadrática*: pretende-se representar um agente de "cabeça fria", que não se precipita, usando uma função intermédia para o decréscimo do preço pedido;
- *cúbica*: a analogia neste caso é a de um agente "guloso", que não abdica facilmente de parte do preço para vender mais rapidamente o produto, apenas o baixando já próximo da data limite de venda.

O utilizador pode alterar os parâmetros estratégicos dos seus agentes em qualquer momento. Após encontrar um comprador para o preço pedido, o agente vendedor contacta o utilizador, para que este confirme a aprovação do negócio.

Os agentes compradores funcionam de forma análoga e simétrica aos agentes vendedores. Na sua criação, o utilizador define a data limite para a compra do produto,

o preço desejado e o preço mais alto aceitável. O utilizador tem igualmente à sua disposição três funções para a subida do preço oferecido ao longo do tempo: linear (correspondendo a um agente "ansioso" por comprar o produto), quadrática (representando um agente de "cabeça fria") e cúbica (um agente "económico").

O mercado do Kasbah ajuda os agentes a encontrarem os seus pares (agentes interessados em comprar/vender aquilo que querem vender/comprar). Os produtos comprados e vendidos no Kasbah são descritos através de vectores de atributos (pares atributo/valor). Esta forma de representar os produtos facilita a sua comparação, bem como o encontro de agentes compradores e vendedores com interesses complementares. Depois de se encontrarem, os agentes comunicam directamente uns com os outros.

Assim que agentes compradores e vendedores chegam a acordo, e após a autorização dos respectivos utilizadores, a transacção física pode ocorrer. Este passo tem que ser dado pelos humanos. A última versão do Kasbah incorpora um mecanismo distribuído de confiança e reputação, chamado *Better Business Bureau* [Guttman *et al.* (1998c), Moukas *et al.* (1998)]. Quando uma transacção é finalizada, cada entidade envolvida pode avaliar a outra relativamente à forma como conduziu a sua parte do acordo. De forma a utilizar esta informação, os utilizadores têm ainda a possibilidade de indicar mais um parâmetro na criação dos seus agentes: o nível mínimo de reputação do dono dos agentes com os quais podem negociar.

No Kasbah, o mercado e os agentes são objectos (instâncias de classes). A sua execução em paralelo é simulada através de um algoritmo que faz circular fatias de tempo de execução por todos os objectos existentes. Os agentes e o mercado comunicam entre si invocando métodos por eles suportados. Esses métodos correspondem a um protocolo interno próprio, dado que todos os agentes estão predefinidos no sistema (utilizando as funções estratégicas acima descritas). Pretende-se evoluir para protocolos que permitam a comunicação entre agentes distribuídos por diversas máquinas, incluindo a possibilidade dos utilizadores criarem os seus próprios agentes, com as suas próprias estratégias, que interagem com o mercado Kasbah utilizando esses protocolos.

Uma experiência envolvendo cerca de 200 participantes na utilização do mercado do Kasbah, em Outubro de 1996, está relatada em Chavez *et al.* (1997) e Guttman *et al.*

(1997). Desenvolvimentos mais recentes no *MIT Media Laboratory* levaram à criação do projecto *PDA@shop* [Zacharia *et al.* (1998)], que pretende interligar o mercado físico com o mercado virtual do Kasbah. O projecto consiste em proporcionar ao utilizador a possibilidade de comparar os preços obtidos no local de compra com preços alcançados no sistema Kasbah, além de obter informação sobre a reputação do vendedor. O utilizador pode, estando num local de venda, através de um *Personal Digital Assistant (PDA)*, comunicar com o Kasbah e criar um *agente comparador*, que é um agente comprador com um tempo de vida mais limitado. O utilizador monitoriza os resultados do agente através do PDA, podendo-os comparar com o mercado físico. Este sistema estava em fase de testes no final de 1998.

4.4 Tete-a-Tete

O projecto Tete-a-Tete, desenvolvido no *MIT Media Laboratory*, engloba três áreas de investigação: sistemas multi-agente, desenho de interfaces homem-computador e Comércio Electrónico por retalho. O objectivo do projecto é disponibilizar tecnologias avançadas e ferramentas de visualização, para auxiliar o encaixe das necessidades dos consumidores com as ofertas dos vendedores, em mercados de retalho "on-line".

O Tete-a-Tete pretende dar aos utilizadores a possibilidade de negociar sobre várias dimensões de uma transacção. Dá a possibilidade aos vendedores de se diferenciarem em atributos do produto e do serviço além do preço, como tempos de entrega, garantia, etc. [Moukas *et al.* (1998)].

A negociação não utiliza funções simples de acréscimo e decréscimo de preços como no Kasbah. Em vez disso, os agentes compradores do Tete-a-Tete negoceiam argumentativamente com os agentes vendedores, utilizando as restrições de avaliação capturadas durante as etapas de *Selecção do Produto* e *Selecção do Vendedor* do modelo CBB, como dimensões de uma função de utilidade multi-atributo [Guttman *et al.* (1998c), Moukas *et al.* (1998)]. Esta função de utilidade é usada por um agente comprador para ordenar as ofertas dos vendedores, baseando-se em avaliar até que ponto elas satisfazem as suas preferências.

O Tete-a-Tete utiliza uma combinação de técnicas de MAUT e DCSP, para mediar as negociações entre agentes compradores e vendedores [Guttman e Maes (1998b)]. A

negociação toma portanto a forma cooperativa, através de múltiplos termos da transacção.

4.5 Comparação

Os mercados electrónicos descritos atrás diferem uns dos outros no que diz respeito, fundamentalmente, aos aspectos indicados na tabela 4.1.

		AuctionBot	***Fishmarket***	***Kasbah***	***Tete-a-Tete***
Tipo de sistema		servidor de leilões diversos	casa electrónica de leilões de tipo holandês	mercado virtual para compra e venda de produtos	mercado virtual para compra e venda de produtos
Objecto de negociação		preço	preço	preço	função de utilidade
Agentes de Software	*Criação*	pelo utilizador (API disponibili-zada)	pelo utilizador	agentes predefinidos e parametri-záveis	
	Estratégia	definida pelo utilizador	definida pelo utilizador	funções linear, quadrática e cúbica	
	Implementação	processos	processos	objectos	
	Comunicação	memória partilhada (base de dados)	TCP/IP, interface próprio (*interagent*)	invocação de métodos	
	Negociação	leilões conduzidos por agente leiloeiro	leilões conduzidos por agente leiloeiro	directa entre agentes comprado-res e vendedores	
Disponibilidade		acessível na "Web"	não	acessível na "Web"	não

Tabela 4.1 Comparação de mercados electrónicos baseados em agentes

Podemos portanto concluir, sobre os sistemas estudados, que:

- apenas o Tete-a-Tete utiliza uma função de utilidade para medir o grau de satisfação de uma proposta multi-atributo, sendo que os restantes sistemas baseiam a negociação no preço;
- apenas o AuctionBot possibilita a utilização de diversos protocolos de negociação, isto é, de diversos tipos de leilões;
- o Kasbah admite agentes compradores e vendedores, que negoceiam de uma forma directa, e providencia estratégias de negociação predefinidas.

5. Aprendizagem

Se o programador tem um conhecimento completo sobre o ambiente que o agente vai enfrentar, toda a "inteligência" requerida pode ser directamente implantada neste. Quando isto não acontece, isto é, quando o programador tem um conhecimento incompleto do problema, a aprendizagem é a única forma que o agente tem para adquirir os conhecimentos de que necessita [Russell e Norvig (1995)]. Neste caso, a construção do agente passa pela implementação de um algoritmo de aprendizagem.

A aprendizagem é um campo da Inteligência Artificial que pretende aumentar a capacidade dos programas, neste caso agentes, de serem capazes de lidar com situações nunca antes experimentadas. Esta área inclui uma vasta gama de metodologias e algoritmos de aprendizagem, que podem ser classificados segundo diversos parâmetros, como a estratégia de aprendizagem subjacente, a representação do conhecimento e o domínio de aplicação [Michalski *et al.* (1984)].

Quanto à estratégia de aprendizagem utilizada, uma forte distinção pode ser feita relativamente à necessidade de um supervisor. A *aprendizagem supervisionada* é a aprendizagem a partir de exemplos fornecidos por um supervisor externo. Estes exemplos são normalmente referidos como o *conjunto de treino*, contendo pares de entrada/saída. A aprendizagem consiste em *induzir* uma determinada função, procedimento ou regra de decisão, a partir do conjunto de treino constituído por instâncias, que permita classificar de forma aceitável um novo item de entrada. Um exemplo de aprendizagem por indução são as *árvores de decisão*, onde cada exemplo de treino fornecido é descrito por um conjunto de propriedades e um valor de saída que representa uma classe. A árvore de decisão, construída a partir dos exemplos de treino, é utilizada para classificar novas situações de entrada, descritas pelas mesmas propriedades. O programa C4.5 [Quinlan (1993)] permite gerar árvores ou regras de decisão desta forma.

Outro exemplo de aprendizagem para a representação de funções complexas são as *redes neuronais* [Rumelhart e McClelland (1986)], que pretendem modelizar o funcionamento do cérebro humano. São compostas por um conjunto de unidades

(neurónios) interligadas entre si, sendo esta configuração definida pelo programador da rede neuronal.

As *redes de crenças* [Pearl (1986)] permitem elaborar representações probabilísticas de conhecimento incerto, utilizando para o efeito *aprendizagem Bayesiana*. A ideia é criar hipóteses a partir dos dados, que permitam depois realizar predições. Estas são feitas com base nas probabilidades de cada uma das hipóteses formuladas.

Em certos domínios, como em problemas onde a interacção é um aspecto central, ou em ambientes dinâmicos, a aprendizagem supervisionada não é aplicável, pela impossibilidade de obter exemplos de treino convenientes. A *aprendizagem não supervisionada* não exige a presença de exemplos de treino, derivando desse facto quer a sua importância quer a dificuldade da sua aplicabilidade.

A *aprendizagem por reforço* (“*reinforcement learning”*) [Sutton e Barto (1998)] baseia-se nas interacções com o ambiente. Um agente aprende a executar as acções apropriadas em cada situação, através de um processo de recompensa das acções que produzem melhores resultados – o agente aprende a partir da sua própria experiência.

Os *algoritmos genéticos* [Koza (1992)] fazem uma analogia à evolução das espécies, baseando-se no facto de que os indivíduos melhor adaptados ao ambiente tenderão a sobreviver. Esta abordagem começa por avaliar o “grau de adaptação” de cada indivíduo de um grupo inicial. Os indivíduos com mais sucesso são seleccionados e reproduzidos para a geração seguinte. Eles caracterizam-se por um conjunto de parâmetros chamados genes, que são depois combinados no processo de reprodução. A este tipo de algoritmos também se dá o nome de algoritmos evolutivos.

Existem vários outros métodos de aprendizagem. As referências apresentadas não pretendem ser exaustivas, mas sim dar uma visão geral das metodologias mais importantes, para podermos situar o tipo de aprendizagem por nós seleccionada para os agentes que desenvolvemos.

5.1 Aprendizagem em Negociação

A construção de agentes autónomos que melhoram a sua competência para negociar, baseando-se na aprendizagem a partir das suas interacções com outros agentes, é ainda uma área emergente [Zeng e Sycara (1996, 1998)].

Em Matos *et al.* (1998) é utilizado um modelo de negociação em que os agentes fazem variar os valores das suas propostas com base em determinados critérios, modelados por funções parametrizáveis chamadas tácticas, que podem ser combinadas de uma forma ponderada. Uma estratégia é definida como uma função que altera os pesos das tácticas a utilizar em cada instante. A aprendizagem implementada neste estudo consiste na utilização de algoritmos genéticos, que permitem gerar uma sequência de populações, cada vez melhor adaptadas à situação, como resultado de um método de pesquisa modelado por um mecanismo de selecção, combinação (recombinação de material genético de novas formas) e mutação (introdução de novo material genético por modificações aleatórias) [Matos *et al.* (1998)]. No caso apresentado, os indivíduos da população são agentes negociantes e os materiais genéticos são os parâmetros das tácticas e os seus pesos relativos. O objectivo é determinar quais as melhores estratégias e verificar como e quando estas estratégias evoluem, dependendo do contexto e comportamento de negociação dos oponentes.

Em Zeng e Sycara (1996, 1998) é descrito o sistema Bazaar, um modelo de negociação sequencial que é capaz de aprender. Nesse trabalho a negociação é modelada explicitamente como uma tarefa sequencial de tomada de decisão, usando actualizações Bayesianas como mecanismo de aprendizagem subjacente. No modelo Bazaar, os agentes possuem informação acerca das condições evolutivas do ambiente e dos modelos de outros agentes. A aprendizagem Bayesiana é utilizada para actualizar o conhecimento e crença que cada agente tem, sobre o ambiente e outros agentes. Um exemplo de informação sobre um oponente, num processo de negociação para a transacção de um determinado produto, é o valor do seu preço de reserva. O comprador não sabe qual é o preço de reserva do vendedor, dado que esta informação é privada. Contudo, ele é capaz de alterar a sua crença acerca desse preço de reserva, através das suas interacções com o vendedor e do seu conhecimento sobre o domínio na área daquele negócio. A actualização Bayesiana é probabilística, e ocorre quando o

comprador recebe novos sinais do ambiente ou do vendedor, nomeadamente através da troca de propostas e contrapropostas. O conhecimento do domínio é fundamental, para que esta informação possa ser utilizada de forma eficiente.

6. Abordagem SMACE

No âmbito deste trabalho foi desenvolvido o sistema SMACE. Neste sistema os utilizadores podem criar agentes compradores e vendedores que negoceiam de uma forma autónoma, de modo a obter acordos sobre os serviços que procuram ou oferecem.

6.1 O Mercado

O esquema de mercado subjacente ao sistema SMACE é o representado na figura 6.1. O *Mercado* é uma entidade que representa o local onde os diversos agentes se encontram e através do qual negoceiam.

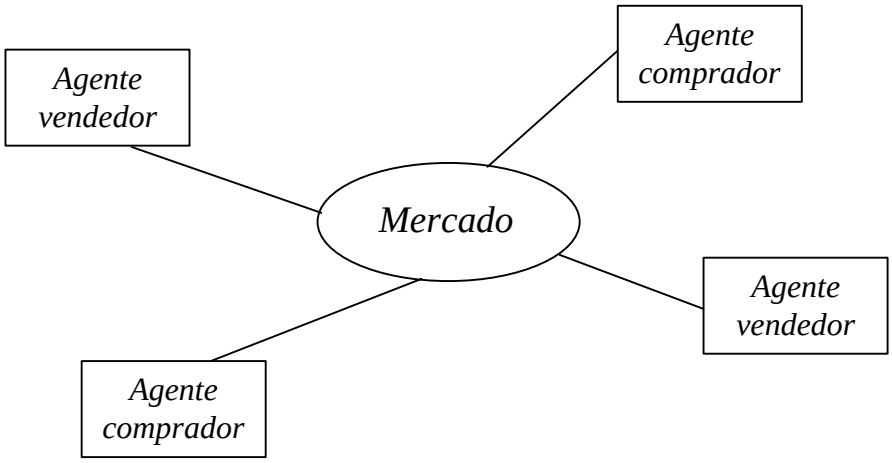


Figura 6.1 Esquema do mercado

O sistema suporta duas etapas do modelo CBB, referido na secção 2.2:

- *Selecção do Vendedor*: os agentes compradores e vendedores que pretendem transaccionar o mesmo produto contactam-se no Mercado.
- *Negociação*: os agentes negoceiam entre si sobre a prestação do serviço relacionado com o produto a transaccionar.

O Mercado tem a dupla função de facilitar o encontro entre agentes e de possibilitar a negociação entre eles. Este é um sistema aberto, de modo que os agentes compradores e vendedores estão potencialmente distantes uns dos outros. O Mercado funciona como

um local centralizado onde eles se podem encontrar de modo a interagirem, negociando sobre os serviços que oferecem ou procuram.

6.2 Modelo de Negociação

O modelo de negociação adoptado tem as seguintes características:

- *multi-lateral*: os agentes compradores/vendedores têm a habilidade de negociar simultaneamente com vários agentes vendedores/compradores. Em termos de leilões, esta abordagem está directamente relacionada com um *sealed bid continuous double auction*, onde tanto os compradores como os vendedores licitam (propõem) simultaneamente e o leilão não pára à medida que cada negociação é concluída (que cada acordo é consumado). Na prática, este modelo de negociação multi-lateral traduz-se em várias negociações bilaterais em curso, que se podem influenciar mutuamente. O modelo de negociação bilateral utilizado baseia-se no definido em Raiffa (1982). A negociação decorre através da troca de propostas entre os agentes.
- *multi-dimensão*: a negociação pode decorrer sobre um grupo de variáveis, em contraste com a maior parte dos leilões, que se debruçam apenas sobre o preço do produto. Estas variáveis podem incluir o preço, qualidade, garantia, modalidades de pagamento, etc.. Uma proposta consiste num valor para cada uma destas dimensões que caracterizam o objecto da negociação.

Este modelo de negociação é também chamado de modelo de negociação orientado ao serviço [Faratin *et al.* (1998)], dado que envolve dois papeis que estão, em princípio, em conflito: vendedores de serviços e compradores de serviços. Um serviço é algo que pode ser fornecido por um agente e requerido por outro agente, e acerca do qual eles podem negociar.

Como regra geral, podemos dizer que os oponentes no processo de negociação têm interesses simétricos em todas as dimensões sobre as quais decorre a negociação. Contudo, pode acontecer que os agentes tenham interesses similares numa dimensão específica, portanto a evolução das propostas de ambos pode evoluir na mesma direcção.

Neste modelo de negociação, os agentes do mercado com objectivos comuns competem entre si de modo a obter o melhor acordo possível com um dos agentes com os quais estão a negociar. Por outro lado, agentes com objectivos complementares (compradores e vendedores de um mesmo produto) negoceiam entre si, cooperando no sentido de encontrar um acordo que constitua uma solução mutuamente aceitável.

O modelo multi-lateral e multi-dimensão implementado no sistema SMACE foi adaptado de Faratin *et al.* (1998). A negociação multi-lateral traduz-se num conjunto de negociações bilaterais a decorrer simultaneamente. A sequência de propostas e contrapropostas numa negociação bilateral denomina-se *fluxo de negociação*. Estas propostas e contrapropostas são geradas por combinações lineares de funções chamadas *tácticas*. As tácticas usam um certo critério (tempo, recursos, etc.) na geração de uma proposta para uma dada dimensão. Podem ser atribuídos pesos diferentes a cada uma das tácticas usadas na combinação referida, representando a importância de cada critério na tomada de decisão. Os agentes podem querer mudar a sua escala de importância de critérios ao longo do tempo. Para isso, eles usam uma *estratégia*, definida como a forma pela qual um agente altera os pesos relativos das diferentes tácticas que guiam a sua tomada de decisão ao longo do tempo.

Numa negociação sobre n dimensões, para cada dimensão $j \in \{1, \ldots, n\}$, cada agente i define:

- uma gama de valores aceitáveis $[min^i_j, max^i_j]$
- uma função de pontuação $V^i_j: [min^i_j, max^i_j] \rightarrow [0, 1]$
- um peso w^i_j

A função de pontuação retorna a pontuação que o agente i atribui a um valor da dimensão j dentro da gama de valores aceitáveis. Quanto maior for a pontuação, maior é a utilidade para o agente. O peso associado a cada dimensão representa a importância relativa dessa dimensãona função global. Considerando pesos normalizados ($\sum_j w^i_j = 1$), define-se a função de pontuação do agente que, para uma determinada proposta $x = (x_1, \ldots, x_n)$, combina as pontuações das diferentes dimensões no espaço multi-dimensional definido pelas suas gamas de valores: $V^i(x) = \sum_j w^i_j V^i_j(x_j)$. A proposta é

avaliada com uma pontuação de zero se algum valor de uma dimensão estiver fora da gama de valores aceitáveis.

6.3 Protocolo de Negociação

Para o protocolo de negociação, assumiu-se que a entrega de mensagens é fidedigna e não se consideraram os atrasos das mensagens (porque são presumivelmente curtos).

Cada agente de mercado criado tem um determinado objectivo, que especifica a sua intenção de compra ou venda de um produto específico. Esse objectivo deve ser alcançado até um certo limite temporal t_{max}.

Um agente continua o seu processo de negociação até que uma de duas situações ocorra:

- o agente chega a acordo com outro agente;
- o agente chega ao fim do tempo disponível para alcançar o objectivo.

Uma vez lançado no mercado, o agente entra em contacto com outros agentes interessados na compra/venda do produto que ele pretende vender/comprar. Uma negociação bilateral começa logo que as duas partes – o vendedor e o comprador – se encontrem e concordem sobre a complementaridade dos seus objectivos (isto é, eles concordam que o que o comprador quer comprar é o que o vendedor pretende vender). Seguidamente, é criado um fluxo de negociação $x_{c \leftrightarrow v}$ entre os dois agentes. Este fluxo de negociação permanece activo enquanto a negociação entre os dois agentes decorrer. Essa actividade consiste na troca alternada de mensagens (cada mensagem é a resposta de um agente à mensagem do seu oponente).

Num fluxo de negociação $x_{a \leftrightarrow b}$, os agentes *a* e *b* trocam uma sequência de propostas e contrapropostas. O procedimento do agente *a*, ao receber uma proposta $x^{t}_{b \rightarrow a}$, é descrito pelo seguinte protocolo:

1) geração de uma contraproposta $x^{t'}_{a \rightarrow b}$;
2) comparação da utilidade da proposta recebida $x^{t}_{b \rightarrow a}$ e da contraproposta gerada $x^{t'}_{a \rightarrow b}$, através das funções de pontuação;
3) envio da resposta:

a) desistir da negociação se o tempo chegou ao fim ($t > t^a_{max}$) ou se o agente chegou a acordo com um outro agente; ou

b) aceitar a proposta recebida se esta tiver maior utilidade do que a contraproposta gerada: $V^a(x^t_{b \to a}) \geq V^a(x^{t'}_{a \to b})$; ou

c) enviar a contraproposta gerada se esta tiver maior utilidade do que a proposta recebida: $V^a(x^t_{b \to a}) < V^a(x^{t'}_{a \to b})$.

A resposta enviada pelo agente é descrita em Faratin *et al.* (1998) como a *função de interpretação*, mas a formalização aí apresentada parece não contemplar o facto de a negociação ser multi-lateral, e portanto cada agente poder estar, num dado momento, a negociar com vários oponentes simultaneamente.

Dado que virtualmente poderão existir vários agentes no mercado a negociar simultaneamente, múltiplos fluxos de negociação podem estar activos num dado momento, para um dado agente. Quando um agente chega a acordo, desiste da negociação com todos os outros agentes com os quais estava a negociar. Um problema (ilustrado na figura 6.2) poderia surgir se um agente recebesse, consecutivamente, duas ou mais mensagens de aceitação das suas propostas, de outros tantos agentes (assumindo que o objectivo do agente admite a transacção de apenas uma unidade do produto, e portanto apenas um acordo pode ser consumado).

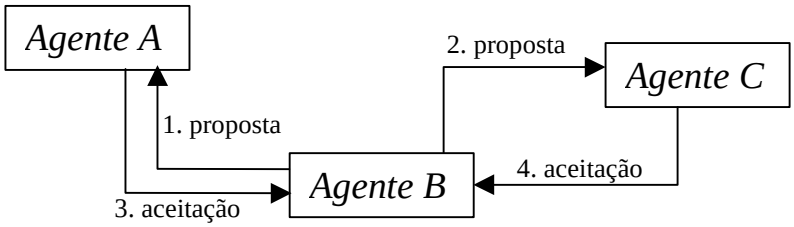


Figura 6.2 Problema de acordos simultâneos

Para resolver este problema, quando um agente *b* recebe uma mensagem de aceitação da sua proposta de um agente *a*, responderá com uma confirmação do acordo, ou sua rejeição. Portanto, se o agente *b* ainda não chegou a acordo com nenhum outro agente, ele irá confirmar o acordo com o agente *a*, o emissor da mensagem de aceitação da proposta. Para garantir que apenas um acordo será efectuado, o agente *a* espera até receber uma resposta do agente *b*.

Um problema de "deadlock" poderia surgir se um grupo de agentes estivesse à espera de confirmações de acordos numa configuração em ciclo fechado (figura 6.3).

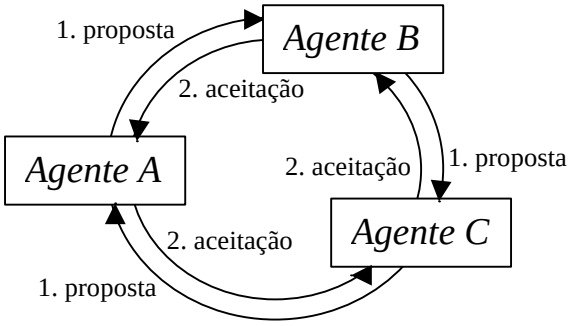


Figura 6.3 Problema de "deadlock"

Este problema foi abordado por nós de uma forma muito simples: o agente *a*, que está à espera de uma confirmação de acordo, desiste de quaisquer outras negociações em curso com outros agentes e rejeita quaisquer outras possibilidades de acordo. Isto significa que, eventualmente, ele irá perder acordos potenciais com outros agentes, no caso de receber uma rejeição do agente *b*. Dado que os atrasos de mensagens foram ignorados, o problema da perda de acordos não ocorrerá frequentemente, e foi portanto ignorado.

A actividade completa de um fluxo de negociação está ilustrada na figura 6.4.

Formalmente, um fluxo de negociação entre os agentes *a* e *b*, no tempo *t*, notado como $x^{t}_{a\leftrightarrow b}$, é uma sequência finita da forma $\left\{x^{t_1}_{a\to b}, x^{t_2}_{b\to a}, x^{t_3}_{a\to b}, \ldots\right\}$ onde:

(i) $t_{i+1} > t_i$, a sequência é ordenada no tempo,

(ii) para cada dimensão *j*, $x^{t_i}_{a\to b}[j] \in \left[\min^{a}_{j}, \max^{a}_{j}\right]$, $x^{t_{i+1}}_{b\to a}[j] \in \left[\min^{b}_{j}, \max^{b}_{j}\right]$, com $i=1,3,5,\ldots$,

(iii) o último elemento da sequência pode ser um de {*aceitar*, *desistir*}, ou um de {*rejeitar*, *acordo*} quando o penúltimo elemento é *aceitar*.

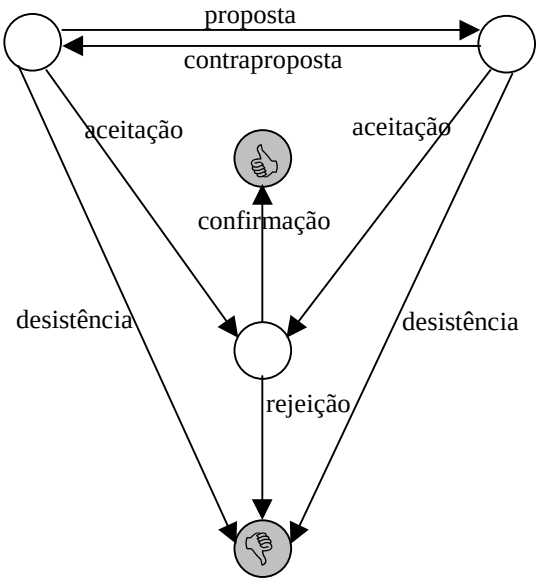


Figura 6.4 Actividade do fluxo de negociação entre dois agentes

6.4 Plataforma e Comunicação

A escolha de uma plataforma de desenvolvimento que facilitasse a construção do sistema multi-agente subjacente ao SMACE simplificaria, em maior ou menor grau, a fase de implementação computacional do sistema. A plataforma que melhor pareceu adaptar-se ao sistema pretendido foi o JATLite. Esta escolha facilitou a implementação da comunicação entre os agentes, dado que toda a infra-estrutura de comunicação necessária está já implementada no JATLite. Uma descrição mais pormenorizada da arquitectura e funcionamento da plataforma JATLite encontra-se no apêndice A.

De facto, a característica mais evidente que apontou para esta escolha foi a necessidade da existência de um local centralizado onde agentes potencialmente distantes se pudessem encontrar, de modo a iniciar o processo de negociação. Este local corresponde ao Mercado ilustrado na figura 6.1. Numa versão mais avançada, o JATLite permitiu implementar o próprio Mercado como um AMR com algumas modificações, de modo a incluir outras funcionalidades. Dado que a comunicação entre os agentes é feita usando a linguagem KQML, esta solução foi de encontro ao referido em Finin *et al.* (1997), onde se diz que *«um ambiente de agentes falando KQML pode ser enriquecido com agentes especiais, chamados facilitadores, que providenciam funções como: associação de endereços físicos com nomes simbólicos; registo de bases*

de dados e/ou serviços oferecidos e procurados por agentes; e serviços de comunicação (encaminhamento, encontro, etc.).»

Por outro lado, a mobilidade providenciada por plataformas como o Voyager não foi um requisito dos agentes no sistema SMACE.

6.4.1 SMACE e JATLite

A primeira abordagem na construção do sistema, com base na plataforma JATLite, corresponde ao ilustrado na figura 6.5.

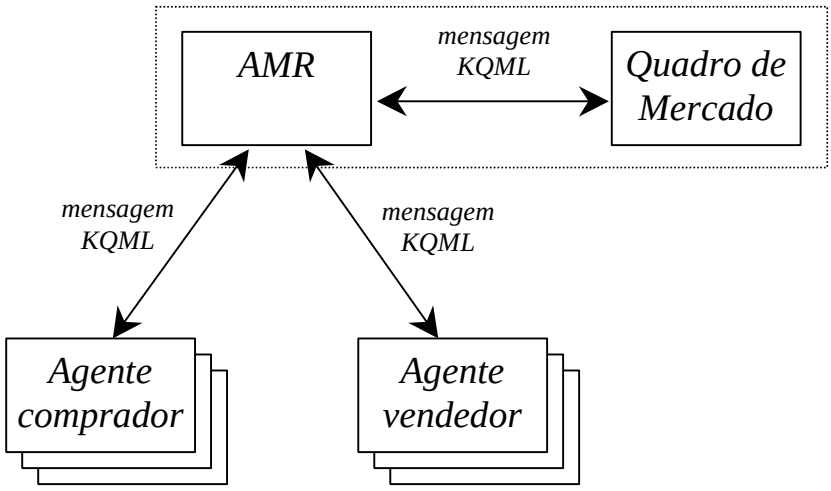


Figura 6.5 Abordagem inicial SMACE e JATLite

O *Quadro de Mercado* funciona como um "blackboard" onde os agentes compradores e vendedores anunciam os seus objectivos (compra ou venda de um determinado produto). Contudo, este "blackboard" não se limita a ser uma zona de memória partilhada por todos os agentes. É um outro agente, com funções específicas, com o qual os agentes comunicam no sentido de encontrarem parceiros de negociação.

Um melhoramento introduzido nesta abordagem inicial foi a inclusão das funcionalidades do *Quadro de Mercado* no próprio AMR. Esta solução permitiu eliminar os custos relacionados com a passagem de mensagens entre estas duas entidades, tornando assim o sistema mais eficiente. O esquema resultante é apresentado na figura 6.6.

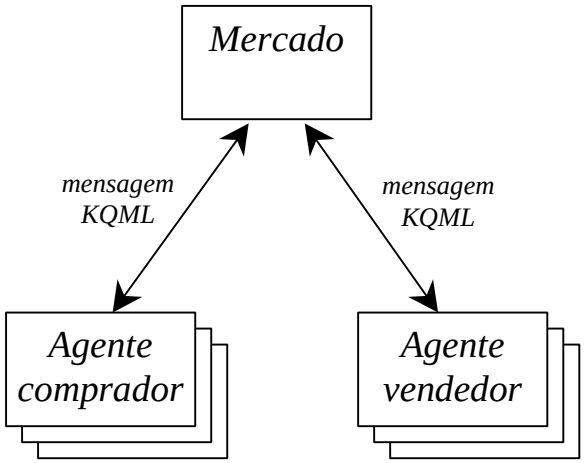


Figura 6.6 Abordagem SMACE e JATLite

Aqui, surge de novo o Mercado como entidade principal, que integra as funcionalidades do AMR e do *Quadro de Mercado*, permitindo:

- o anúncio dos objectivos de agentes compradores e vendedores;
- a procura de parceiros para negociação, a partir dos objectivos anunciados;
- a negociação entre agentes compradores e agentes vendedores, através da troca de mensagens KQML.

6.4.2 SMACE e KQML

A linguagem adoptada para comunicação entre os agentes foi o KQML, que tende a se tornar numa linguagem padrão para a troca de informação e conhecimento. A utilização da plataforma JATLite veio facilitar o uso desta linguagem, uma vez que inclui as funcionalidades necessárias para tratar de mensagens do tipo KQML de uma forma simples. O apêndice B descreve mais pormenorizadamente a linguagem KQML.

Não foram contudo utilizadas as "performativas" predefinidas na especificação do KQML [Labrou e Finin (1997)]. Uma comunidade de agentes pode optar por utilizar "performativas" adicionais, logo que acorde na sua interpretação e no protocolo associado a cada uma delas [Finin *et al.* (1997)]. Por isso, foi criado um conjunto de "performativas" específicas para utilização no SMACE. O apêndice C descreve as "performativas" adoptadas na comunicação entre os agentes e o mercado. Aquelas que

são utilizadas pelos agentes para comunicação e negociação entre si estão descritas no apêndice D.

6.5 Aprendizagem

O sistema SMACE constitui uma plataforma de teste de agentes com diversas estratégias de negociação. De modo a explorar esta funcionalidade do sistema, foi implementado e testado um tipo de agente específico que possui capacidade de aprendizagem. Essa aprendizagem permite-lhe implementar uma estratégia adaptativa, de modo a melhorar o seu desempenho com a sua experiência.

O agente adaptativo deveria ser capaz de melhorar o seu desempenho através das suas interacções com os restantes agentes. Portanto, o algoritmo de aprendizagem escolhido deveria possibilitar uma aprendizagem "on-line" e não supervisionada. Nestas circunstâncias, *aprendizagem por reforço* pareceu ser uma boa aposta.

Em contraste, a aprendizagem supervisionada é feita a partir de exemplos fornecidos num conjunto de treino. Esta não é adequada a este domínio, dado que em problemas de interacção – como a negociação num ambiente dinâmico – é vulgarmente impraticável obter exemplos representativos dos comportamentos desejados. Em território desconhecido – onde se espera que a aprendizagem seja mais benéfica – um agente tem de ser capaz de aprender a partir da sua própria experiência [Sutton e Barto (1998)].

O *algoritmo de aprendizagem-Q (Q-learning)* é um tipo de aprendizagem por reforço dos mais conhecidos, tendo sido adoptado para incluir nos agentes. O algoritmo de aprendizagem-Q selecciona uma acção num determinado estado dependendo da pontuação (qualidade) desse par estado-acção. Converge garantidamente para as melhores combinações de pares estado-acção, depois de cada acção ter sido experimentada as vezes suficientes. Uma questão fundamental na implementação deste algoritmo foi a definição, no domínio do sistema SMACE, do que é um estado e do que é uma acção. Outro aspecto importante, neste tipo de aprendizagem, é o compromisso entre tirar partido de acções consideradas de boa qualidade e a exploração de acções ainda não experimentadas ou anteriormente consideradas menos boas. Estas questões serão abordadas em concreto no capítulo 9.

6.6. Implementação

O sistema SMACE foi implementado em Java (*JDK1.1.4 API*)[15] e usa a plataforma JATLite para a criação de agentes que trocam mensagens KQML.

O SMACE consiste numa API, construída como uma extensão do JATLite, incluindo três níveis distintos, conforme se pode ver na figura 6.7.

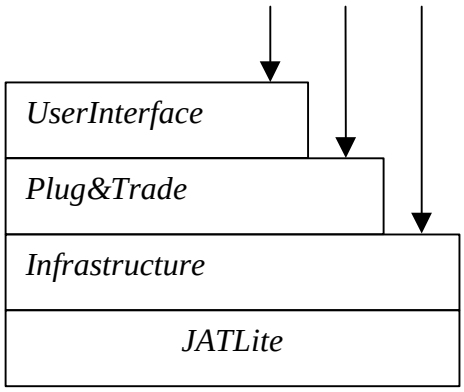


Figura 6.7 SMACE API

- *Infrastructure*[16] – este nível assegura toda a comunicação e negociação entre os agentes, consistindo em duas partes fundamentais:
  - *MarketAgent*: é um molde ("template") que pode ser utilizado para a criação de agentes de mercado, tendo já implementados a interacção com o *MarketPlace*, o modelo de negociação e o seu protocolo associado.
  - *MarketPlace*: é a aplicação que representa o mercado, como um espaço onde os agentes se encontram e transaccionam.
- *Plug&Trade* – este nível inclui agentes de mercado predefinidos, que podem ser vistos como exemplos de como um agente pode ser construído a partir do molde *MarketAgent*:

[15] URL: http://www.javasoft.com/

[16] *Infrastructure*, *Plug&Trade* e *UserInterface* traduzem-se por Infraestrutura, "Ligar e Negociar" e Interface com o Utilizador. No entanto, dada a difusão internacional que se pretende do protótipo mantivemos os nomes em inglês. Idêntica atitude tomamos relativamente aos termos *MarketAgent* (Agente de Mercado), *MarketPlace* (Mercado), *MultipleTacticAgent* (Agente de Múltiplas Tácticas) e *AdaptiveBehaviourAgent* (Agente de Comportamento Adaptativo).

  - *MultipleTacticAgent (MTA)*: um agente de mercado que pode usar uma combinação ponderada de três tácticas, uma de cada família de tácticas descritas no capítulo 7, para gerar as suas propostas de negociação.
  - *AdaptiveBehaviourAgent (ABA)*: um agente de mercado que é capaz de pesar as várias tácticas que usa de uma forma adaptativa, utilizando técnicas de aprendizagem por reforço.
- *UserInterface* – este nível consiste numa aplicação que providencia uma interface HTML para a criação e monitorização da actividade de agentes de mercado do nível *Plug&Trade*, assegurando também a sua persistência. O anexo A mostra alguns ecrãs desta interface com o utilizador.

Os agentes comunicam uns com os outros através do *MarketPlace*, um AMR do JATLite melhorado que:

- suporta o encaminhamento de mensagens entre os agentes;
- funciona como uma central de informações onde os agentes anunciam os seus objectivos e procuram oponentes.

O sistema SMACE permite que agentes criados em qualquer parte do mundo entrem no mercado e negoceiem (desde que utilizem o mesmo protocolo de negociação). Assim, para interagir com o sistema, o utilizador pode:

- criar o seu próprio agente, com a sua própria estratégia, em qualquer linguagem de programação ou plataforma que deseje;
- utilizar o nível *Infrastructure*, mais especificamente o *MarketAgent*, para criar o seu agente na linguagem Java. Esta solução permite que o utilizador não tenha que se preocupar com os detalhes de comunicação e negociação, debruçando-se sobre a construção da sua estratégia de negociação, isto é, sobre a capacidade deliberativa do seu agente;
- lançar um agente predefinido (do tipo MTA ou ABA), ajustando os seus parâmetros. Para isso, pode utilizar a interface do SMACE. Através desta interface a actividade do agente pode ser monitorizada e os seus parâmetros podem também ser ajustados.

6.6.1 Encontro de Agentes

O primeiro passo de um agente, para a prossecução do seu objectivo de compra ou venda de um determinado produto, consiste na procura de outras partes interessadas. O *MarketPlace* providencia a forma de concretizar este encontro, entre agentes que pretendem negociar a transacção do mesmo produto.

Quando é lançado no mercado, o agente começa por se registar e anunciar o seu objectivo no *MarketPlace*. Isto permite que outros agentes lançados posteriormente tenham acesso ao seu contacto. Esta informação é de acesso público, no sentido em que qualquer agente registado pode questionar o *MarketPlace* sobre os objectivos de outros agentes, ou obter os nomes dos agentes que têm um determinado objectivo. Este é o segundo passo: o agente questiona o *MarketPlace* sobre a existência no mercado de agentes com objectivos complementares ao dele, isto é, objectivos incluindo intenções simétricas sobre o mesmo produto.

Para facilitar a comparação dos produtos, estes são descritos através de pares descritor/valor, que no seu conjunto definem o objecto da negociação. Os valores dos mesmos descritores em objectivos diferentes são comparados para verificar se o produto é o mesmo. O sistema SMACE pode ser facilmente configurado para especificar quais os descritores que os agentes de mercado irão usar na descrição dos seus produtos.

Caso haja agentes no mercado com objectivos complementares ao do agente que fez o pedido ao *MarketPlace*, este responderá indicando os seus nomes. O agente procederá depois contactando os potenciais oponentes de negociação. Quando se contactam, os agentes começam por verificar os objectivos entre si. Este passo é conveniente por duas razões. Primeiro, porque pode acontecer que a informação disponibilizada pelo *MarketPlace* não esteja actualizada. Segundo, as funções de comparação de objectivos dos agentes podem ser mais refinadas do que aquelas aplicadas pelo *MarketPlace*. Por exemplo, o *MarketPlace* poderia concluir que a compra e a venda de um automóvel são objectivos complementares, mas um determinado agente poderia ser mais rigoroso na comparação dos dois objectivos, restringindo a marca do automóvel. Este refinamento pode ser feito através de uma utilização mais minuciosa dos descritores publicitados no *MarketPlace*, ou pela introdução, no processo de comparação, de outras informações que por qualquer razão o agente não quis tornar públicas. Finalmente, se acordarem que

têm de facto intenções simétricas sobre o mesmo produto, iniciarão o processo de negociação.

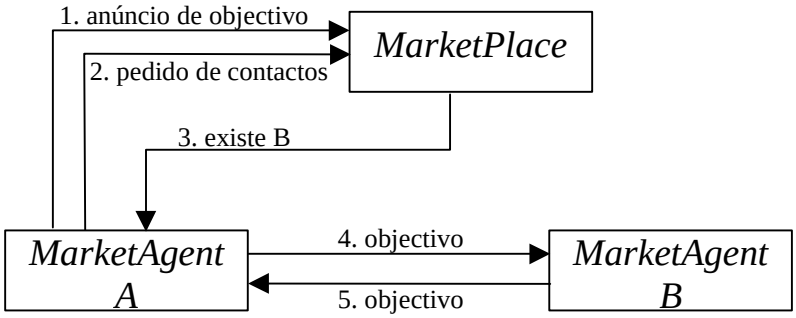


Figura 6.8 Encontro de agentes

As mensagens KQML trocadas entre os agentes e o mercado estão descritas no apêndice C. A mensagens KQML para a verificação de objectivos entre os agentes podem ser encontradas no apêndice D.

6.6.2 Dimensões de Negociação

Para que os agentes possam negociar de uma forma apropriada, além de usarem os mesmos protocolos de comunicação e negociação, eles devem acordar sobre quais dimensões decorrerá a negociação. Esta concordância é fundamental para o entendimento, por parte dos agentes, das propostas e contrapropostas que irão trocar entre si.

O sistema SMACE pode ser facilmente configurado para considerar qualquer número de dimensões. Todos os agentes de mercado criados no sistema usarão depois o mesmo conjunto de dimensões na negociação. Estas dimensões são todas consideradas uniformes, no sentido em que, para os agentes, elas não têm uma semântica associada. Portanto, cada agente de mercado irá definir um peso, uma gama de valores aceitáveis e uma função de pontuação para cada dimensão utilizada. Intenções simétricas (de compra e de venda) normalmente implicam funções de pontuação de alguma forma contrastantes. Por exemplo, a função de pontuação da dimensão preço é inversamente proporcional ao preço para os compradores e directamente proporcional ao preço para os vendedores.

7. Tácticas

Neste capítulo descrevemos as várias tácticas utilizadas pelos agentes negociadores no sistema SMACE.

O objectivo da negociação é a maximização da utilidade obtida numa transacção. Para isso, é necessário estudar a forma de preparar propostas e contrapropostas adequadas.

Os agentes de mercado predefinidos do sistema SMACE usam uma táctica específica, ou uma combinação várias tácticas, para gerar as suas propostas. As tácticas implementadas nesses agentes foram adoptadas de Faratin *et al.* (1998):

- *tácticas dependentes do tempo*: os agentes variam as suas propostas de acordo com o tempo que dispõem para negociar. Estas tácticas usam uma função dependente do tempo que pode ser parametrizada;
- *tácticas dependentes de recursos*: os agentes fazem variar as suas propostas com base na disponibilidade de um determinado recurso. Estas tácticas são semelhantes às dependentes do tempo, só que o domínio da função usada é a quantidade de um recurso que não o tempo. Isto pode ser conseguido tornando o tempo máximo de negociação dinâmico, ou fazendo com que a função dependa de uma estimativa da quantidade de um recurso;
- *tácticas dependentes do comportamento*: os agentes tentam imitar o comportamento dos seus oponentes num determinado grau. Diferentes tipos de imitações podem ser feitas, baseando-se na conduta da negociação de um oponente na sequência das suas propostas: *imitação proporcional*, *imitação absoluta* e *imitação proporcional média*.

7.1 Tácticas Dependentes do Tempo

As tácticas dependentes do tempo fazem variar as propostas de acordo com a proximidade do fim do tempo disponível para negociar. Um agente *a* tem que chegar a acordo até t^a_{max}. Uma proposta *x* para a dimensão *j* do agente *a* para o agente *b* no tempo *t*, com $0 \leq t \leq t^a_{max}$, pode ser calculada da seguinte forma:

$$x^{t}_{a\to b}[j] = \begin{cases} \min^{a}_{j} + \alpha^{a}_{j}(t)(\max^{a}_{j} - \min^{a}_{j}), \text{ se } V^{a}_{j} \text{ diminui} \\ \min^{a}_{j} + (1 - \alpha^{a}_{j}(t))(\max^{a}_{j} - \min^{a}_{j}), \text{ se } V^{a}_{j} \text{ aumenta} \end{cases}$$

onde V^a_j é a função de pontuação cujo gradiente reflecte a intenção do agente (para a dimensão preço, os compradores terão uma função de pontuação decrescente, ao passo que para os vendedores esta será crescente).

Qualquer função $\alpha^a_j(t)$ definindo o comportamento dependente do tempo deve satisfazer as seguintes restrições:

- $0 \leq \alpha^a_j(t) \leq 1$: as propostas estão dentro da gama de valores;
- $\alpha^a_j(0) = \kappa^a_j$: a variável κ^a_j ajusta o valor inicial, dentro da gama de valores, para o tempo inicial;
- $\alpha^a_j(t^a_{max}) = 1$: no final do tempo disponível para negociação, será proposto o valor de reserva – que obtém o menor resultado da função de pontuação V^a_j.

De modo a satisfazer estas restrições, duas classes de funções são apresentadas:

- Polinomiais

$$\alpha^a_j(t) = \kappa^a_j + (1 - \kappa^a_j)\left(\frac{\min(t, t_{\max})}{t_{\max}}\right)^{\frac{1}{\beta}}$$

- Exponenciais

$$\alpha^a_j(t) = e^{\left(1 - \frac{\min(t, t_{\max})}{t_{\max}}\right)^{\beta} \ln \kappa^a_j}$$

O parâmetro $\beta \in \Re^+$ é usado para ajustar o grau de convexidade da curva, permitindo a criação de um número infinito de tácticas possíveis. Um valor de $\beta = 1$ representa uma curva *linear*. Para valores de $\beta < 1$ o comportamento de ambas as classes pode ser descrito como *ambicioso*, isto é, as concessões são feitas perto do final do tempo disponível para negociação, de outra forma as propostas mudam apenas ligeiramente. Para $\beta > 1$ o comportamento é chamado de *concedente*. Um agente com este comportamento apressa-se no estabelecimento de um acordo e alcança o seu valor de reserva rapidamente.

7.2 Tácticas Dependentes de Recursos

As tácticas dependentes de recursos fazem variar as propostas com base na quantidade de um recurso disponível. Um comportamento ambicioso, utilizado na presença de uma grande quantidade de recursos, deve mudar para um comportamento concedente quando os recursos se tornam escassos.

7.2.1 Tácticas de Tempo Máximo de Negociação Dinâmico

Esta sub-família de tácticas faz variar o tempo máximo de negociação de acordo com a disponibilidade de um determinado recurso. O recurso modelado aqui é o número de agentes que estão a negociar e o tamanho médio dos fluxos de negociação activos. Se um agente vendedor *a* detecta muitos compradores interessados no seu produto, então não há necessidade de obter rapidamente um acordo. O grupo de agentes a negociar com o agente *a* no tempo *t* é

$$N^a(t) = \left\{ i \,\middle|\, x^t_{i \leftrightarrow a} \text{ está activo} \right\}$$

O tempo máximo de negociação dinâmico, usando o recurso acima descrito, é

$$t^a_{\max} = t + \mu^a \frac{\left|N^a(t)\right|^2}{\sum_i \left|x^t_{i \leftrightarrow a}\right|},$$

onde μ^a é o tempo que o agente *a* considera necessário numa negociação com um só oponente e $|x^t_{i \leftrightarrow a}|$ é o tamanho do fluxo de negociação entre o agente *a* e o agente *i*.

7.2.2 Tácticas de Estimativa de Recursos

As tácticas de estimativa de recursos medem a quantidade de um recurso num tempo *t*. A função α pode ser usada para modelar isto:

$$\alpha^a_j(t) = \kappa^a_j + (1 - \kappa^a_j) e^{-resource(t)}$$

A função *resource* é usada aqui para avaliar a quantidade de recursos disponíveis no tempo *t*. Os exemplos seguintes modelam

- os oponentes interessados: $resource(t) = \left|N^a(t)\right|$

- os oponentes interessados e o tamanho dos fluxos de negociação, como anteriormente: $resource(t) = \mu^a \frac{\left|N^a(t)\right|^2}{\sum_i \left|x^t_{i\leftrightarrow a}\right|}$

- o tempo: $resource(t) = \max(0, t_{\max} - t)$

7.3 Tácticas Dependentes do Comportamento

As tácticas dependentes do comportamento tentam imitar o comportamento dos oponentes do agente num certo grau. As tácticas desta família fazem contrapropostas influenciadas pelas acções anteriores do oponente. Apresentam-se três maneiras diferentes de usar comportamento "imitativo", assumindo o fluxo de negociação $\{..., x^{t_{n-2\delta}}_{b\to a}, x^{t_{n-2\delta+1}}_{a\to b}, x^{t_{n-2\delta+2}}_{b\to a}, ..., x^{t_{n-2}}_{b\to a}, x^{t_{n-1}}_{a\to b}, x^{t_n}_{b\to a}\}$ e $\delta \geq 1$.

7.3.1 Imitação Proporcional

Estas tácticas imitam proporcionalmente o comportamento de um oponente $\delta \geq 1$ passos atrás. O tamanho do fluxo de negociação tem que ser $n > 2\delta$. A contraproposta gerada é:

$$x^{t_{n+1}}_{a\to b}[\mathrm{j}] = \min(\max(\frac{x^{t_{n-2\delta}}_{b\to a}[\mathrm{j}]}{x^{t_{n-2\delta+2}}_{b\to a}[\mathrm{j}]} x^{t_{n-1}}_{a\to b}[\mathrm{j}], \min^{\mathrm{a}}_{\mathrm{j}}), \max^{a}_{j})$$

A contraproposta gerada é calculada com a última proposta ($x^{t_{n-1}}_{a\to b}[j]$) e com a evolução proporcional de duas propostas consecutivas do oponente ($x^{t_{n-2\delta}}_{b\to a}[j]$ e $x^{t_{n-2\delta+2}}_{b\to a}[j]$).

7.3.2 Imitação Absoluta

Estas tácticas imitam em termos absolutos o comportamento do oponente. Elas requerem a existência de um fluxo de negociação de tamanho $n > 2\delta$. A contraproposta resultante é:

$$x^{t_{n+1}}_{a\to b}[j] = \min(\max(x^{t_{n-1}}_{a\to b}[j] + (x^{t_{n-2\delta}}_{b\to a}[j] - x^{t_{n-2\delta+2}}_{b\to a}[j]) + (-1)^{s} R(M), \min^{a}_{j}), \max^{a}_{j})$$

onde $s = \begin{cases} 0, V^{a}_{j} \text{ diminui} \\ 1, V^{a}_{j} \text{ aumenta} \end{cases}$

Na fórmula anterior, a função *R(M)* providencia uma forma de fugir aos mínimos locais. Retorna um inteiro aleatório no intervalo *[0, M]*, onde *M* é o limiar do comportamento imitativo.

7.3.3 Imitação Proporcional Média

Estas tácticas imitam proporcionalmente, calculando a alteração média de um certo número de propostas em relação à última proposta. O parâmetro γ refere-se ao número de propostas passadas que são consideradas. A contraproposta obtida é:

$$x^{t_{n+1}}_{a\to b}[j] = \min(\max(\frac{x^{t_{n-2\gamma}}_{b\to a}[j]}{x^{t_{n}}_{b\to a}[j]} x^{t_{n-1}}_{a\to b}[j], \min^{a}_{j}), \max^{a}_{j})$$

para $n > 2\gamma$. O comportamento da imitação proporcional média quando se escolhe $\gamma = 1$ é semelhante à imitação proporcional com $\delta = 1$.

8. Estratégias

Um dos objectivos principais deste trabalho foi comparar o desempenho de agentes usando estratégias baseadas em comportamentos adaptativos com outros baseados em comportamentos menos dinâmicos, ou mesmo estáticos.

Assim que um agente esteja criado e correctamente parametrizado, pode ser activado de forma a contactar o mercado, começando assim um novo episódio na sua vida. Dentro de um episódio, o objectivo do agente não pode ser alterado. O episódio termina quando o agente é desactivado.

A geração de uma proposta consiste na geração de um valor para cada uma das dimensões utilizadas. A geração do valor de cada dimensão obedece à utilização de várias tácticas combinadas de uma forma ponderada. Uma estratégia pode ser interpretada como a forma pela qual se seleccionam as combinações ponderadas das tácticas descritas atrás. A estratégia de um agente determina que combinação de tácticas deve ser usada em cada instante particular de um episódio.

Pode-se então definir uma matriz de pesos de tácticas, a ser utilizada num determinado momento para a geração de uma proposta. Para uma negociação em *p* dimensões, combinando para cada uma delas *m* tácticas, temos:

$$\Gamma^{t}_{a\to b} = \begin{pmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \omega_{p1} & \omega_{p2} & \cdots & \omega_{pm} \end{pmatrix}$$

onde ω_{ji} é o peso da táctica *i* para a dimensão *j*.

Alterando os pesos, que definem a importância de cada táctica, ao longo do tempo, dependendo da situação, obtém-se um mecanismo que permite que o agente se adapte. A estratégia de negociação é então definida como uma função *f* que altera a matriz Γ ao longo do tempo, a partir do "estado mental" MS^{t}_{a} do agente – esse "estado mental" pode reflectir o seu conhecimento do ambiente:

$$\Gamma^{t+1}_{a\to b} = f\left(\Gamma^{t}_{a\to b}, MS^{t}_{a}\right)$$

À matriz Γ está associada a matriz das tácticas propriamente ditas. Cada táctica τ, aplicada à dimensão *j*, gera um valor no intervalo de valores aceitáveis para essa dimensão, utilizando um critério subjacente com base no estado mental do agente: $\tau_{ji}(MS_a) \rightarrow [min^a_j, max^a_j]$. O valor proposto, no tempo *t+1*, para a dimensão é então dado pela seguinte fórmula:

$$x^{t_{n+1}}_{a\rightarrow b}[j] = \sum_{i=1}^{m} \left(\omega_{ji} \tau_{ji} \left(MS_a^{t_{n+1}} \right) \right)$$

Para possibilitar o desenvolvimento e inclusão de capacidades de aprendizagem nos agentes, ou para que os utilizadores possam reutilizar um agente de sucesso, os *MarketAgents* não morrem no fim de um episódio. Eles podem ser reactivados com objectivos possivelmente diferentes e, se desejado, com outros valores para as dimensões.

8.1 Estratégias Orientadas pelo Utilizador

A estratégia mais simples consiste em usar a mesma matriz de combinação de tácticas em qualquer altura, isto é, $\Gamma^{t+1}_{a\rightarrow b} = \Gamma^{t}_{a\rightarrow b}$ para qualquer *t*.

O *UserInterface* do SMACE permite ao utilizador criar um MTA, que combina três tácticas diferentes – uma de cada família descrita atrás. As sub-tácticas são seleccionadas ajustando os parâmetros da família de tácticas correspondente. A par destes parâmetros, a combinação ponderada das tácticas mantém-se inalterável, a não ser que o utilizador forneça explicitamente valores diferentes. O utilizador pode interagir com o *UserInterface* para observar as acções do agente e alterar os seus parâmetros em qualquer altura, implementando desta forma a sua própria estratégia.

8.2 Estratégias Adaptativas

Há várias formas de obter um comportamento adaptativo, no ambiente proporcionado pelo sistema SMACE. Esse ambiente é dinâmico, isto é, vários agentes desconhecidos podem-se encontrar e negociar durante um período de tempo, alcançando um determinado resultado pelo seguimento de uma certa estratégia.

As tácticas providenciam uma forma de adaptação, num determinado grau, a diferentes situações, considerando determinados critérios. Mas que tácticas devem ser usadas em que situações? As experiências realizadas em Faratin *et al.* (1998) mostram que o tempo disponível para negociação e o valor da proposta inicial também influenciam os resultados obtidos.

A estratégia de um MTA é simples: ele usa a combinação ponderada de tácticas fornecida pelo utilizador. O agente agirá de acordo com esta parametrização inicial, independentemente da situação que enfrenta em cada momento particular.

Usando o molde *MarketAgent* da API do SMACE, foi incluído um algoritmo de aprendizagem num novo agente – *AdaptiveBehaviourAgent (ABA)* – para providenciar uma forma de adaptação do agente ao ambiente, pela modificação da combinação ponderada das tácticas utilizadas. Isto corresponde à função *f*, responsável pela alteração da matriz Γ.

A definição de uma estratégia como a forma de alterar os pesos da matriz Γ nem sempre é clara em Faratin *et al.* (1998) e em Matos *et al.* (1998). Por vezes, a estratégia passa por ser meramente uma determinada combinação ponderada de tácticas.

A aprendizagem implementada nos agentes ABA, baseada em técnicas de aprendizagem por reforço, incide sobre os pesos a usar em cada uma das tácticas utilizadas em cada dimensão no processo de negociação (isto é, sobre a matriz Γ).

Em Matos *et al.* (1998) é aplicado um algoritmo genético para determinar, de uma forma evolucionária, a melhor combinação de tácticas aplicada a determinada situação. A aprendizagem nesse caso é aplicada de forma mais abrangente: nos pesos e nos próprios parâmetros das tácticas. São apresentados dois tipos de experiências. Primeiro estuda-se a evolução dos parâmetros das tácticas, utilizando apenas uma táctica em cada dimensão. Verifica-se também o desempenho relativo de cada táctica aplicada relativamente às outras. Num segundo grupo de experiências estuda-se principalmente a evolução dos pesos usados em combinações de tácticas. Pretende-se determinar que combinações têm sucesso. Nestas experiências, os agentes usam a mesma configuração de pesos e parâmetros das tácticas durante o processo de negociação, de modo a avaliar o seu desempenho em comparação com outros. A aplicação do algoritmo genético prossegue até se obter uma população estável – o ponto de equilíbrio, no qual o

desempenho da grande maioria dos agentes é semelhante. O mesmo artigo aponta como trabalho futuro a produção de alterações da matriz Γ durante a vida do agente, para permitir que ele se adapte em negociações repetidas com os seus oponentes.

A implementação de técnicas de aprendizagem por reforço nos agentes ABA teve como objectivo dotar o agente de duas capacidades:

- aumentar o número de acordos em negociações consecutivas quando existem competidores;
- aumentar a utilidade obtida em vários acordos sucessivos resultantes de vários episódios.

Pretendeu-se verificar se o agente se adapta a um determinado ambiente de mercado, relacionado com a transacção de um determinado tipo de produto. Assim, embora a aprendizagem incida sobre os pesos das tácticas, a sua evolução é apenas um meio, e não o fim. A finalidade é a adaptação dos agentes pela melhoria dos seus resultados.

9. Implementação do Algoritmo de Aprendizagem

A aprendizagem por reforço permite que o agente aprenda a partir da sua própria experiência. No contexto do SMACE, pretende-se que, através da implementação deste tipo de aprendizagem, o agente ganhe sensibilidade sobre como o mercado funciona na transacção de determinado tipo de produtos. Essa sensibilidade refere-se não só ao ritmo habitual de aparecimento de novos agentes (dinâmica do mercado), mas também ao modo como os agentes compradores e vendedores relaxam normalmente os seus valores de compra e venda. O agente vai aperfeiçoando o seu desempenho à medida que vive mais e está presente em mais episódios de negociação.

9.1 Aprendizagem por Reforço – Aprendizagem-Q

A ideia que está na base da aprendizagem por reforço é aquela que nos ocorre imediatamente quando pensamos na natureza da aprendizagem: nós aprendemos interagindo com o ambiente [Sutton e Barto (1998)]. A definição da aprendizagem por reforço não passa pela caracterização de um algoritmo de aprendizagem, mas sim de um problema de aprendizagem. Vários algoritmos podem depois ser implementados de modo a resolver o problema.

No modelo de aprendizagem por reforço, o agente interage com o ambiente [Harmon (1996)]. Esta interacção consiste na percepção do ambiente e na selecção de uma acção para executar nesse ambiente. A acção altera o ambiente de alguma forma e esta alteração é comunicada ao agente através de um sinal de reforço.

Um estado caracteriza a situação do *ambiente* e é especificado por um conjunto de variáveis que o descrevem. Os sistemas de aprendizagem por reforço aprendem um mapeamento de situações a acções através de interacções com um ambiente dinâmico. A execução de uma acção num determinado estado dá origem a um reforço recebido pelo agente, na forma de um valor numérico. O agente aprende a executar acções que maximizam a soma dos reforços recebidos, desde o estado inicial até ao estado final. A escolha de uma *função de reforço* (ou recompensa) que transpareça, de uma forma apropriada, os objectivos do agente é fundamental.

O problema central da aprendizagem por reforço é portanto a escolha de acções. A *política* define o comportamento do agente, determinando que acção deve ser executada em cada estado. Quase todos os algoritmos que implementam este tipo de aprendizagem se baseiam em estimar *funções de valor* [Sutton e Barto (1998)] – funções que determinam quão bom é um estado (função estado-valor) ou quão bom é executar uma determinada acção num estado (função acção-valor).

9.1.1 Cálculo da Função de Valor

Em Sutton e Barto (1998) são apresentadas várias abordagens para resolver o problema da aprendizagem por reforço. Os métodos baseados na *Programação Dinâmica* requerem um modelo completo e perfeito do ambiente. Actualizam as estimativas do valor de um determinado estado com base em estimativas dos estados sucessores desse estado (este mecanismo é chamado de "bootstrapping"). Os métodos de *Monte Carlo* não assumem um modelo do ambiente, requerendo apenas experiência – sequências de amostras de estados, acções e respectivos reforços, resultado da interacção com o ambiente. Os métodos de aprendizagem por *Diferença Temporal* constituem uma combinação das ideias subjacentes às duas abordagens anteriores. Estes métodos conseguem aprender a partir da experiência, sem conterem um modelo da dinâmica do ambiente, mas actualizam as estimativas com base parcial noutras estimativas, sem terem de esperar pelo resultado final. Os métodos de *Temporal Difference* são os métodos de aprendizagem por reforço mais utilizados, sendo de entre estes o mais conhecido a *aprendizagem-Q (Q-learning).*

Um ambiente *determinístico* é caracterizado pelo facto de que a execução de uma acção num determinado estado leva sempre a um mesmo estado sucessor. Num ambiente desta natureza, bastaria implementar uma função estado-valor para obter um processo de decisão sobre a escolha da acção a executar: escolhe-se uma acção que conduza a um bom estado. Num ambiente *não-determinístico* o resultado da execução de uma acção num estado não é certo. Daí que seja necessário estimar o valor da execução dessa acção nesse estado. Para isso, a acção tem que ser executada várias vezes, de modo a que o valor estimado se aproxime do valor real.

O algoritmo de aprendizagem-Q constrói uma relação de pares estado-acção com valores (chamados valores de qualidade – *Q*). O valor de uma acção *a* considerada num estado *s* é representado por *Q(s,a)*, sendo actualizado pela seguinte fórmula:

$$Q(s,a) = Q(s,a) + \alpha [r + \gamma max_{a'} Q(s',a') - Q(s,a)]$$

onde:

$0 \leq \alpha \leq 1$	é a taxa de aprendizagem, representando o impacto da actualização no valor actual
r	é o reforço obtido pela execução de *a* em *s*
$0 \leq \gamma \leq 1$	é o factor de desconto, representando a importância dos valores Q nos estados futuros para o valor a ser actualizado
$max_{a'} Q(s',a')$	é o máximo valor para as acções do estado seguinte

A actualização dos valores de qualidade para as acções tomadas depende portanto do reforço obtido e dos valores de qualidade futuros.

O estudo da aprendizagem por reforço inclui numerosas variações [Sutton e Barto (1998)], que em determinados ambientes podem melhorar substancialmente o resultado da aprendizagem. A implementação progressiva da aprendizagem-Q nos agentes ABA passou por algumas dessas variações.

9.1.2 Escolha de Acções

A regra mais simples para a selecção de uma acção consiste em seleccionar aquela que tem um maior valor de qualidade (Q). Este método toma sempre partido do conhecimento actual para maximizar o reforço imediato. Contudo, não considera o facto de que podem existir acções, ainda não experimentadas, que conduzirão a um resultado melhor. Além disso, em ambientes dinâmicos e portanto não-determinísticos, acções que obtiveram resultados menos bons no passado podem melhorar a sua prestação no presente.

De facto, para obter um reforço continuado e de elevado valor, o agente deve preferir acções que tentou no passado e que descobriu serem eficientes na produção de reforços

[Sutton e Barto (1998)]. Mas para as descobrir ele tem que tentar acções que nunca seleccionou antes. Este dilema leva à necessidade de um compromisso entre:

- *"exploitation"*: tirar partido das acções que são consideradas boas;
- *"exploration"*: explorar acções desconhecidas ou menos boas.

Para satisfazer este compromisso, duas políticas para a escolha de acções são consideradas:

- ε*-greedy*: este método selecciona uniformemente, com uma probabilidade ε, uma acção que não é a melhor (logo escolhe a melhor com probabilidade 1-ε). De entre todas as acções não óptimas, a probabilidade de escolher acções boas e más é igual.
- *Softmax*: este método utiliza um grau de exploração τ (temperatura) para escolher de entre todas as acções possíveis, considerando a sua classificação (desta forma, a acção melhor tem a maior probabilidade de ser escolhida). Quanto mais elevada for a temperatura, mais equi-provável é a escolha das diversas acções (menos importância têm as suas classificações). No limite $\tau \rightarrow 0$, é sempre escolhida a acção melhor.

9.2 Implementação do Algoritmo de Aprendizagem-Q

A implementação inicial do algoritmo de aprendizagem-Q, que resultou na criação dos agentes ABA, foi desenhada e levada a cabo por Max Schaefer, estando documentada em Schaefer (1999). Faz-se aqui um resumo dessa implementação e de algumas melhorias posteriormente introduzidas por nós.

Um agente adaptativo, dotado de aprendizagem usando a aprendizagem-Q, executa basicamente os seguintes passos, para cada dimensão de negociação:

1. determina o estado actual
2. escolhe uma combinação ponderada de tácticas
3. usa a combinação ponderada de tácticas para gerar a próxima proposta
4. actualiza o valor da acção

O sucesso deste processo depende, em grande parte, de como os estados são caracterizados. A abordagem seguida distingue parâmetros gerais que permanecem inalterados durante um episódio e outros que identificam estados dentro de um episódio. Outro factor importante é a política de reforços implementada, que vai determinar a forma como o agente aprende a utilizar determinadas acções, de modo a atingir o seu objectivo cada vez mais eficientemente.

O objectivo da aprendizagem-Q é a escolha de acções que permitam obter reforços cada vez melhores. Essas acções, no contexto dos agentes ABA, consistem na aplicação de combinações ponderadas de tácticas na geração das propostas, para cada dimensão de negociação. Essas tácticas são predefinidas no início do processo de aprendizagem. Não são tácticas em abstracto, mas sim um conjunto limitado de tácticas concretas, definidas pela instanciação de todos os seus parâmetros. O agente aprende a combinar este conjunto concreto de tácticas, para uma determinada dimensão de negociação. Estas considerações, aliadas ao facto de que o agente não altera a sua intenção durante um episódio, constituem a configuração geral que todos os estados de um episódio terão em comum. Esta configuração é denominada de *Condições de Enquadramento*.

Há vários factores que podem caracterizar os estados no decorrer de um episódio. Aqueles que foram considerados referem-se ao número de outros agentes com os quais um determinado agente se encontra a negociar e à percentagem de tempo que tem disponível para negociar. Além disso, o estado contém uma variável que indica se o agente se encontra de facto a negociar, ou se está à espera que agentes oponentes entrem no mercado, permitindo também distinguir os estados inicial e final.

Resumindo, um estado é representado da seguinte forma:

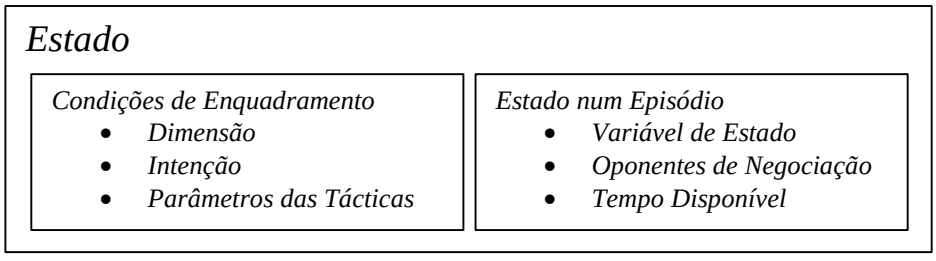


Figura 9.1 Representação de um Estado

Quanto à implementação dos reforços obtidos como resultado da execução das acções, o modelo inicialmente adoptado não considerava cenários dinâmicos. Este consistia em penalizar a perda de acordos com os oponentes, e em premiar a obtenção de um acordo com a utilidade associada ao valor acordado. Em cenários dinâmicos, novos agentes podem entrar no mercado em qualquer altura, daí que a perda de um acordo com um oponente não signifique que o agente não possa alcançar o seu objectivo. A versão final consistiu em penalizar apenas a situação de fracasso de alcance do objectivo, no final do episódio.

A aprendizagem-Q possibilita a utilização de reforços em estados intermédios, para actualização do valor das acções. Contudo, nos agentes ABA não foi introduzida esta facilidade, devido à dificuldade de definir o valor de um reforço a meio de uma negociação. Dado que a maior parte das tácticas implementadas não faz depender os valores gerados das propostas recebidas, não faz sentido reforçar uma acção com base na contraproposta do oponente. Penalizar uma acção levaria o agente a tender a consumar acordos com poucas propostas, mas uma vez que os custos de comunicação não são considerados, esta abordagem não seria correcta. Além disso, pretende-se que o agente aumente a sua utilidade nos acordos obtidos, o que será conseguido através de negociações mais longas (o agente aprenderá a esperar que o oponente conceda). Portanto, o reforço é calculado apenas no estado final de obtenção de acordo, sendo directamente proporcional à utilidade obtida. Diferentes reforços permitem honrar as acções que aumentam a utilidade do agente. No caso de acordo no valor de reserva, a utilidade é zero, assim como o reforço. O reforço máximo é atribuído para um acordo no valor de maior utilidade. O fim de um episódio sem o alcance do objectivo, ou seja, sem obtenção de acordo, dá origem a uma penalização (que consiste num reforço negativo). Isto permite distinguir esta situação de acordos nos valores de reserva. O mecanismo de penalização considera também que acções não exploradas podem ser melhores do que aquelas que foram penalizadas.

A aprendizagem-Q implementado nos agentes ABA sofreu algumas alterações, de acordo com a análise dos resultados das experiências levadas a cabo, conforme descrito no próximo capítulo. Essas alterações foram introduzidas com o intuito de melhorar a prestação dos agentes.

Para criar um agente ABA, torna-se portanto necessária a especificação de um conjunto de parâmetros de aprendizagem que este vai utilizar (ver secção 9.1). São eles:

- Selector Probabilístico: selector de acções – ε-greedy ou Softmax;
- Probabilidade: valor de ε, no caso de ε-greedy;
- Temperatura: valor de τ, no caso de Softmax;
- Taxa de Aprendizagem: taxa de aprendizagem α;
- Factor de Desconto: factor de desconto γ;
- Granularidade Temporal: granularidade da percentagem de tempo disponível para negociar, utilizada na caracterização do estado;
- Granularidade Negocial: granularidade do número de fluxos de negociação activos (número de agentes com os quais o agente está correntemente a negociar), utilizado na caracterização do estado.

10. Experiências e Resultados

De modo a avaliar o desempenho dos agentes dotados de aprendizagem, foram executadas diversas experiências, aqui relatadas. A análise progressiva dos resultados de tais experiências permitiu introduzir alterações nos agentes ABA, que conduziram a uma melhoria significativa nos resultados finais expostos no presente relatório.

Dado que todas as configurações de negociação dos agentes são privadas, estes começam a negociar sem saber se é possível obter um acordo [Faratin *et al.* (1998)]. Esta informação privada inclui as tácticas de negociação e também os parâmetros das dimensões de negociação – pesos, gamas de valores aceitáveis e funções de pontuação.

Nos testes realizados, os agentes foram criados de modo a propiciar a negociação entre eles e a possibilitar a obtenção de acordos. Isto é, eles foram configurados com o mesmo produto oferecido/requerido, com gamas de valores e tempos máximos de negociação aceitáveis. Por uma questão de simplicidade, a negociação decorre sobre uma única dimensão, o preço. Esta opção não afecta os resultados, uma vez que a aprendizagem implementada nos agentes ABA é feita em cada dimensão, de uma forma independente. As funções de pontuação (de cálculo da utilidade) utilizadas são linearmente crescentes (para os vendedores) e decrescentes (para os compradores).

A avaliação do desempenho dos agentes, nomeadamente dos agentes ABA, obriga à execução repetida de episódios de negociação. Só desta forma é que o processo de aprendizagem se pode desenrolar. Para a execução automática de vários episódios em cada um dos cenários projectados, foi construído um programa de testes que, a partir de informação fornecida relativamente aos agentes que compõem cada cenário, verifica quando é que um episódio terminou e reinicializa o cenário. Reinicializar um cenário significa reactivar cada um dos agentes que o compõem e fixar o seu tempo máximo de negociação. Para a execução automática dos cenários dinâmicos apresentados nas experiências mais avançadas, o programa de testes foi melhorado de modo a permitir a activação de cada um dos agentes de forma independente, num intervalo de tempo especificado. Um cenário dinâmico é simulado desta forma, através do aparecimento no mercado de agentes em tempos aleatoriamente diferentes.

As experiências relatadas no presente capítulo dividem-se em três grupos. Para estudar o comportamento das diversas tácticas apresentadas no capítulo 7, no primeiro grupo de experiências foram utilizados apenas agentes MTA. O Segundo grupo de experiências visou testar o algoritmo de aprendizagem implementado nos agentes ABA. Após a introdução de algumas modificações nesse algoritmo, com vista à melhoria dos resultados obtidos, foi conduzido um terceiro grupo de experiências, de modo a avaliar a eficácia de tais alterações.

Para facilitar a leitura, foi adoptada a seguinte nomenclatura para os agentes: <*nCenário*><*tipoAgente*><*intenção*><*n*>, onde <*nCenário*> é o número do cenário, <*tipoAgente*> é o tipo do agente (MTA ou ABA), <*intenção*> representa a intenção de compra (b – "buyer") ou venda (s – "seller") e <n> é o número de ordem de comprador ou vendedor. Por exemplo, o agente *1MTAb2* é o segundo comprador do cenário 1, sendo do tipo MTA. Quando não houver ambiguidade omitiremos o número do cenário.

10.1 Análise Inicial das Tácticas

Das experiências relatadas por Faratin *et al.* (1998), com agentes que utilizam combinações fixas de tácticas, é importante registar as seguintes observações:

- a quantidade de tempo disponível para negociação influencia o desempenho de cada uma das tácticas em relação às outras;
- a probabilidade de acordo e a sua utilidade são também condicionadas pelo valor inicial de oferta.

De facto, com maiores tempos de negociação, as tácticas que dependem do recurso que representa o tamanho dos fluxos de negociação concedem mais do que as dependentes do tempo, pois mais mensagens são trocadas no decorrer da negociação (há mais tempo disponível). Ao contrário, as tácticas dependentes do tempo concedem mais depressa quando o tempo disponível é mais curto.

As tácticas dependentes do comportamento são também influenciadas pelo tempo disponível. Dado que necessitam de um fluxo de negociação de dimensão mínima para poderem imitar, para tempos de negociação maiores elas têm mais oportunidades para

aceder ao comportamento do oponente e responder de acordo com o grau de imitação desejado.

10.1.1 Experiências com Agentes MTA

O primeiro conjunto de experiências visou verificar se existe algum tipo de combinação de tácticas que seja substancialmente melhor do que os outros. Ao mesmo tempo que se pretendeu combinar as três famílias de tácticas, foram criados agentes com comportamentos predominantes para cada uma das famílias.

Nestas experiências foram utilizados agentes MTA, que são agentes que combinam três tácticas, uma de cada família. Foram desenhados seis cenários, tendo cada um deles sido lançado em duas situações: para um tempo disponível de negociação de 5 minutos e 1 minuto. Em cada cenário foram criados três agentes vendedores (MTAs1, MTAs2 e MTAs3), competindo entre si, com a seguinte combinação ponderada de tácticas:

	Dep. Tempo	*Dep. Recursos*	*Dep. Comportamento*
MTAs1	70%	15%	15%
MTAs2	15%	70%	15%
MTAs3	15%	15%	70%

Tabela 10.1 Pesos relativos das tácticas para os MTAs vendedores

A táctica dependente do tempo utilizada foi polinomial, com $\beta = 1$ e $\kappa = 0$ (ver secção 7.1). Este valor para β permite obter uma curva *linear* de comportamento. Um comportamento concedente será melhor quando o tempo para negociação é curto e um comportamento ambicioso será mais adequado a tempos de negociação longos. Um comportamento *linear* será portanto mais robusto para ser usado nas duas situações. O valor de κ escolhido permite que os agentes alcancem a utilidade máxima, explorando totalmente a sua gama de valores. A táctica dependente de recursos utilizada foi a de tempo máximo de negociação dinâmico, usando uma função polinomial semelhante à anterior (ver secção 7.2.1). Foi escolhido um valor de $\mu = tmax$. A táctica dependente do comportamento escolhida foi de imitação proporcional, com $\delta = 1$ (ver secção 7.3.1) para permitir a sua aplicação o mais cedo possível.

O factor diferenciador de cada um dos cenários foi a configuração do agente comprador (MTAb1), com o qual os restantes agentes vão negociar:

	MTAb1
Cenário 1	Apenas dependente do tempo, polinomial com $\beta = 1$ (linear).
Cenário 2	Apenas dependente do tempo, polinomial com $\beta = 10$ (concedente).
Cenário 3	Apenas dependente do tempo, polinomial com $\beta = 0.1$ (ambicioso).
Cenário 4	Semelhante a MTAs1.
Cenário 5	Semelhante a MTAs2.
Cenário 6	Semelhante a MTAs3.

Tabela 10.2 Configuração do MTA comprador em cada cenário

Todos os agentes foram configurados com a mesma gama de valores para a única dimensão de negociação utilizada.

No primeiro grupo de experiências, onde foi utilizado um tempo máximo de negociação de 5 minutos, verificou-se que o agente MTAs2 obteve a grande maioria dos acordos, em todos os cenários. Isto deve-se ao facto de este agente ser mais concedente do que os outros, visto que a utilidade por ele obtida foi baixa. No cenário 2, o agente MTAs3 conseguiu alguns acordos, dado ser fundamentalmente um imitador de um MTAb1 concedente. O mesmo aconteceu no cenário 5. Contudo, mesmo neste cenário o agente MTAs2 obteve a grande maioria de acordos, pois o agente MTAs3 imitou um agente MTAb1 com uma táctica dominante dependente do recurso mas que não é tão concedente (porque tem três oponentes).

No segundo grupo de experiências, com um tempo máximo de negociação de 1 minuto, em todos os cenários se verificou um decréscimo do número de acordos obtidos pelo agente MTAs2. Este facto confirma que o tempo de negociação afecta os resultados. O agente MTAs3, sobretudo dependente do comportamento, ganhou a maioria dos acordos perdidos pelo agente MTAs2. Apenas no cenário 2, com um agente MTAb1 concedente, o vendedor MTAs1, fundamentalmente dependente do tempo, obteve um número significativo de acordos. Este agente concede mais no início do que

o MTAs3. As utilidades obtidas nos acordos dependem em grande parte da forma como o comprador MTAb1 concede. Assim, no cenário 2 (MTAb1 é concedente), o comprador obteve utilidades baixas e os vendedores obtiveram utilidades altas.

5min. / *1min.*	*Cenário 1*	*Cenário 2*	*Cenário 3*	*Cenário 4*	*Cenário 5*	*Cenário 6*	*Média*
MTAb1	0.85 0.77	0.42 0.3	0.93 0.93	0.83 0.75	0.73 0.71	0.76 0.72	0.75 0.7
MTAs1	0 0.5	0 0.72	0 0	0 0.43	0 0.46	0 0	0 0.35
MTAs2	0.15 0.22	0.57 0.68	0.07 0.07	0.16 0.24	0.26 0.25	0.23 0.26	0.24 0.29
MTAs3	0 0.36	0.61 0.71	0 0.22	0 0.35	0.35 0.35	0 0.35	0.16 0.39

Tabela 10.3 Utilidades médias obtidas pelos agentes MTA

Como se pode observar pela tabela 10.3, com um tempo de negociação maior o comprador MTAb1 foi capaz de obter uma utilidade média maior. Isto deve-se ao facto de ele conter uma táctica dependente do tempo, e por isso com tempos de negociação maiores demora mais tempo a conceder, esperando pelas concessões dos oponentes.

Embora o vendedor MTAs2, com uma táctica dominante dependente de recursos, tenha obtido a maior parte dos acordos em todos os cenários para as duas situações, ele obteve a pior utilidade média em todos os cenários, porque o recurso utilizado obriga-o a aplicar um comportamento concedente (excepto naqueles onde os seus competidores não obtiveram qualquer acordo, tendo por isso uma utilidade média de zero). O vendedor MTAs1, fundamentalmente dependente do tempo, foi aquele que conseguiu obter melhores utilidades nos acordos que fez. Tal se explica porque as tácticas dependentes do tempo não dependem de factores externos, como a disponibilidade de um determinado recurso ou o comportamento dos oponentes de negociação.

10.2 Experiências Iniciais com Agentes ABA

As experiências aqui relatadas tiveram como ponto de partida os resultados das experiências anteriores.

Quando um agente enfrenta a mesma situação vezes consecutivas (em cenários fixos, como os anteriores), ele deve ser capaz de melhorar a utilidade obtida nos acordos efectuados. Isto pode ser feito aprendendo a pesar adequadamente as tácticas utilizadas. Assim, considerando um conjunto fixo de tácticas, a aprendizagem implementada nos agentes ABA incide sobre os pesos a utilizar na combinação dessas mesmas tácticas. As tácticas escolhidas devem providenciar todos os comportamentos de negociação desejados: desde o extremo ambicioso até ao extremo concedente.

No primeiro grupo de experiências com agentes ABA, pretendeu-se fundamentalmente testar o algoritmo de aprendizagem implementado. Para isso, procurou-se construir cenários fixos que permitissem:

- perceber o funcionamento na prática do algoritmo de aprendizagem;
- prever e conhecer facilmente o resultado da aplicação das tácticas utilizadas;
- interpretar comparativamente os resultados do desempenho dos agentes.

As tácticas dependentes de recursos não foram utilizadas neste conjunto de experiências. Estas tácticas são influenciadas pelo tempo de negociação. Outro factor importante é o recurso considerado: nos cenários fixos construídos, não faz muito sentido utilizar tácticas dependentes do número de oponentes, pois este é constante. Os atrasos nas mensagens afectam também as tácticas que dependem do comprimento do fluxo de negociação. Na presença de atrasos, o número de mensagens trocadas num período de tempo é menor, tornando o agente menos concedente.

As tácticas dependentes do comportamento foram também dispensadas, pois embora tenham obtido bons resultados nas experiências iniciais elas imitam basicamente tácticas dependentes do tempo no oponente. Além disso, as tácticas dependentes do comportamento não nos permitem providenciar a gama desejada de comportamentos para os agentes – de ambicioso a concedente –, pois dependem do comportamento do oponente.

As tácticas utilizadas neste conjunto de experiências pertencem à família das dependentes do tempo. Estas obtiveram os melhores resultados nas experiências iniciais, e permitem-nos fornecer facilmente os comportamentos desejados.

10.2.1 Configuração dos Cenários

Foram desenhados 4 cenários, conforme descrito na tabela 10.4. Todos os agentes foram configurados com a mesma gama de valores para a única dimensão de negociação utilizada, tendo também o mesmo tempo máximo de negociação de 3 minutos.

Os agentes MTA foram criados com apenas uma táctica dependente do tempo, polinomial e com $\kappa = 0$. Os valores do parâmetro β variam conforme o cenário, sendo apresentados na tabela. Os agentes ABA foram criados com cinco tácticas dependentes do tempo, polinomiais, com $\kappa = 0$ e valores de $\beta = \{0.05; 0.5; 1; 2; 20\}$. Desta forma eles têm disponíveis comportamentos entre os extremos ambicioso e concedente, como mostra a figura 10.1. Um teste simples da função polinomial utilizada nas tácticas dependentes do tempo permitiu verificar que os valores de 0.05 e 20 para β constituem extremos razoáveis para utilização no sistema SMACE.

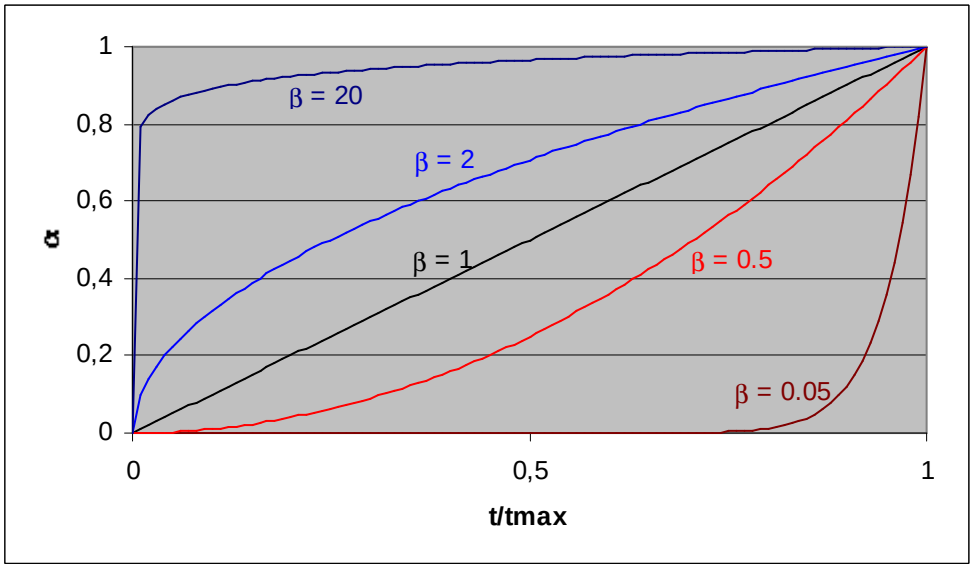


Figura 10.1 Funções polinomiais utilizadas nos agentes ABA

Os parâmetros de aprendizagem (ver secção 9.2) usados nestes agentes foram: *Softmax*, $\tau = 0.4$, $\alpha = 0.1$, $\gamma = 0.9$, *Time slice = 10*, *Gradient = 1.0*.

Cenário	*Agentes*	*Previsões para o agente ABA*
1	1ABAs1 1MTAb1: linear ($\beta = 1$)	Aumento da utilidade.
2	2ABAs1 2MTAb1: ambicioso ($\beta = 0.5$)	Aumento da utilidade (mais lento que no cenário 1).
3	3ABAs1 3MTAb1: linear ($\beta = 1$) 3MTAs2: linear ($\beta = 1$)	Aumento do número de acordos, seguido de aumento da utilidade até ao melhor possível.
4	4ABAs1 4MTAb1: linear ($\beta = 1$) 4MTAs2: concedente ($\beta = 5$)	Aumento do número de acordos, seguido de aumento da utilidade até ao melhor possível (mais lento que no cenário 3).

Tabela 10.4 Configuração dos cenários com agentes ABA

A melhoria da utilidade no cenário 1, para o agente 1ABAs1, não tem limite teórico, dada a inexistência de competidores e a igualdade dos tempos inicial e máximo de negociação. O agente 1ABAs1 poderá aumentá-la até um valor próximo de 1. No cenário 2, embora se verifiquem as mesmas condições, a obtenção de utilidades tão altas é mais difícil. Dado que o agente 2MTAb1 é ambicioso, o agente 2ABAs1 apenas conseguirá obter utilidades altas mais próximo do tempo limite, onde o seu oponente concede.

No cenário 3, o agente 3ABAs1 aprenderá a bater o seu competidor 3MTAs2 na obtenção de acordos, e deverá ser capaz de aumentar a utilidade até próximo do valor de utilidade alcançada por este. Dado que tanto o agente 3MTAs2 como o oponente 3MTAb1 são lineares e utilizam funções de utilidade lineares, o valor de utilidade alcançado num acordo entre os dois rondará os 0.5. O cenário 4 torna a situação mais difícil, dado que o agente 4MTAs2 se contenta com uma utilidade inferior, que ronda os 0.24. O agente 4ABAs1 não poderá portanto elevar a sua utilidade acima deste valor. A

figura 10.2 mostra as curvas de negociação dos agentes MTA destes dois cenários, indicando no eixo das ordenadas a utilidade das propostas destes agentes para um agente ABA vendedor.

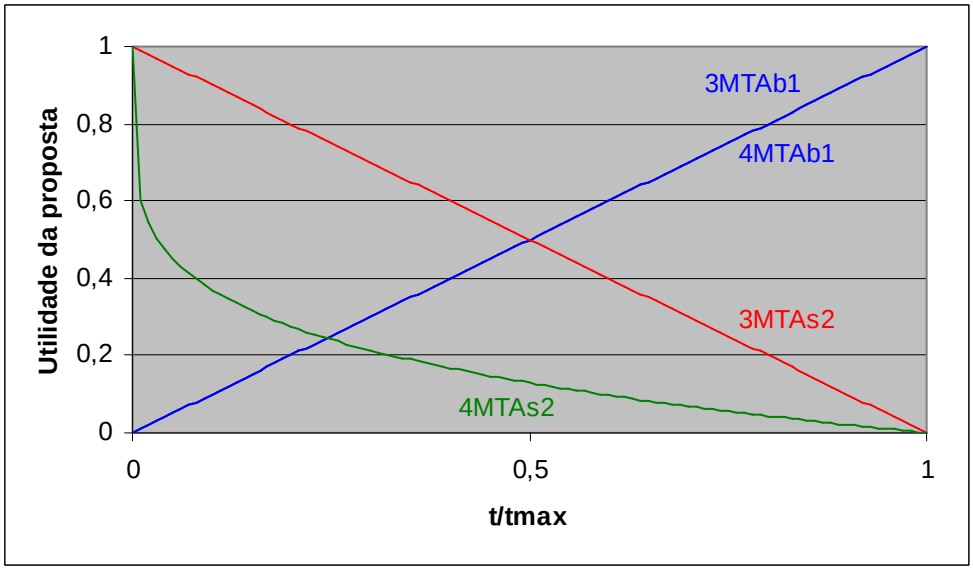


Figura 10.2 Curvas de negociação dos agentes MTA nos cenários 3 e 4

10.2.2 Resultados

Os resultados aqui apresentados referem-se à execução de 2000 episódios em cada um dos cenários descritos.

No cenário 1, o agente 1ABAs1 não conseguiu melhor do que uma utilidade média de 0.43, embora tenha efectuado alguns acordos com uma utilidade de até 0.71. A utilidade média encontrava-se com um andamento muito ligeiramente crescente, conforme mostra a figura 10.3.

No cenário 2, o agente 2ABAs1 obteve uma utilidade média de 0.35, ao fim dos 2000 episódios. Contudo, ele conseguiu alguns acordos com utilidades até 0.68. A figura 10.4 mostra que a utilidade média se encontrava com uma taxa de crescimento muito pequena.

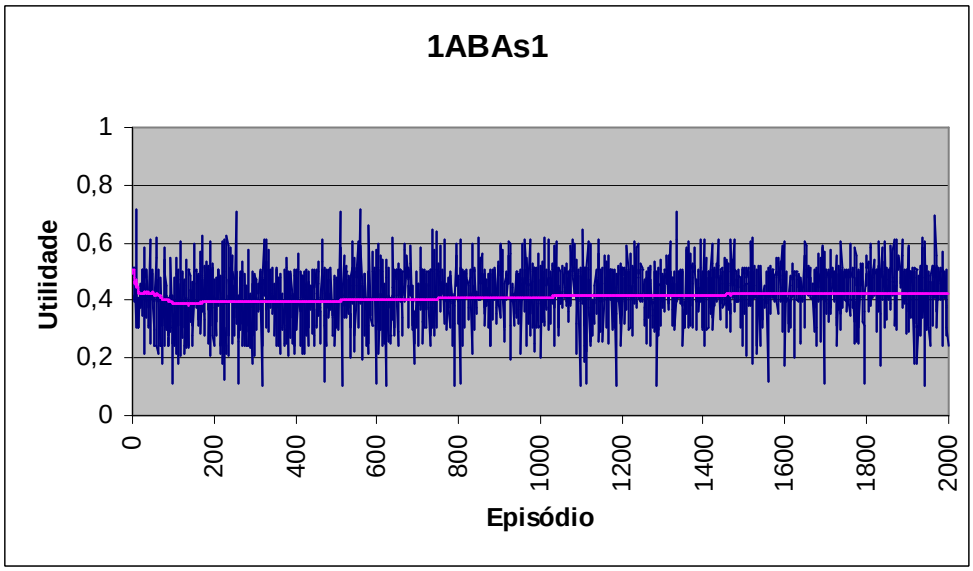


Figura 10.3 Utilidades e utilidade média do agente 1ABAs1

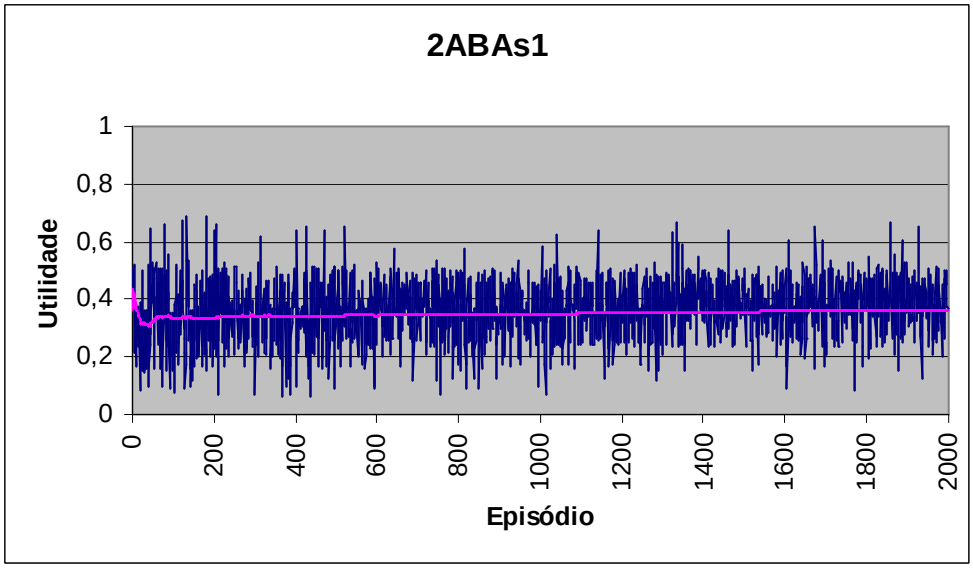


Figura 10.4 Utilidades e utilidade média do agente 2ABAs1

No cenário 3, o agente 3ABAs1 conseguiu bater o seu competidor em relação ao número de acordos obtidos, com 85% do total de acordos possíveis. Como mostra a figura 10.5, até sensivelmente ao episódio 600 houve alguma flutuação da percentagem de acordos obtidos, com o agente 3MTAs2 a obter um número de acordos significativo. Após este período inicial, o agente 3ABAs1 aumentou continuamente a percentagem de acordos obtidos.

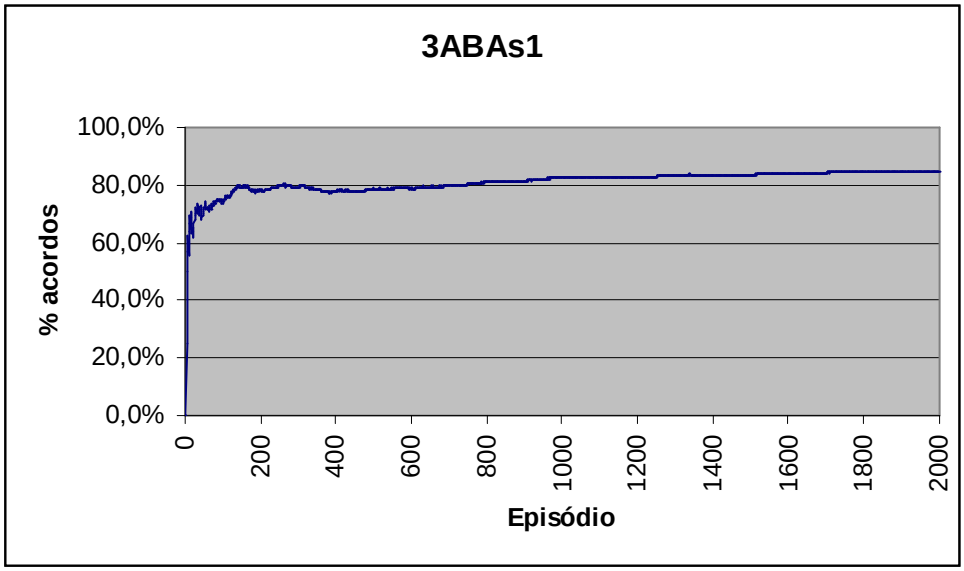


Figura 10.5 Percentagem de acordos obtidos pelo agente 3ABAs1

Quanto à utilidade obtida nos acordos efectuados, esta teve uma média final de 0.38, mostrando um crescimento lento. A figura 10.6 permite visualizar algumas utilidades com valores que rondam os 0.5.

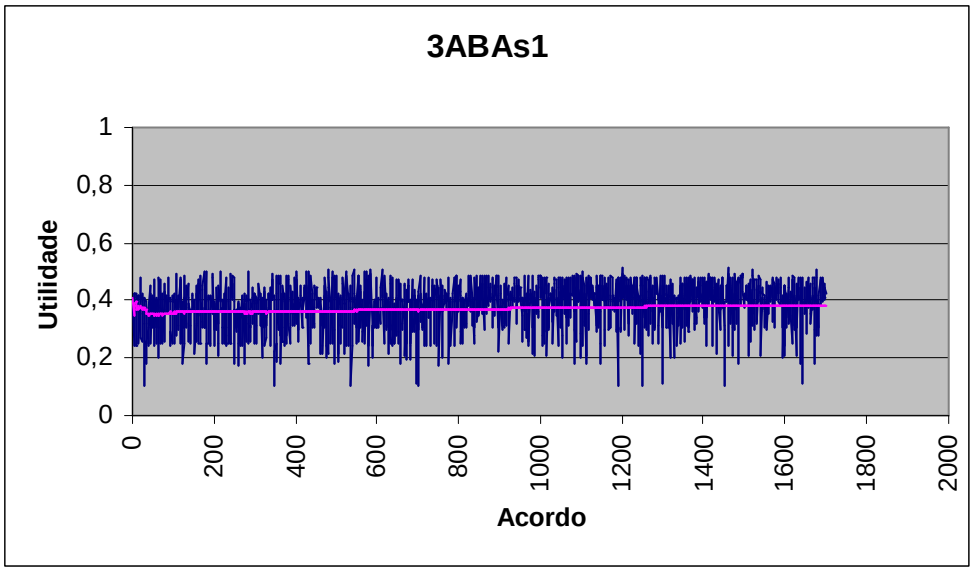


Figura 10.6 Utilidades e utilidade média do agente 3ABAs1

No cenário 4, o agente 4ABAs1, em condições mais difíceis, foi também capaz de bater o seu competidor 4MTAs2, como mostra a figura 10.7. Após um período inicial,

de cerca de 200 episódios, em que o agente se adaptou à situação, ele obteve no final 76% do total de acordos possíveis, com tendência para continuar a aumentar.

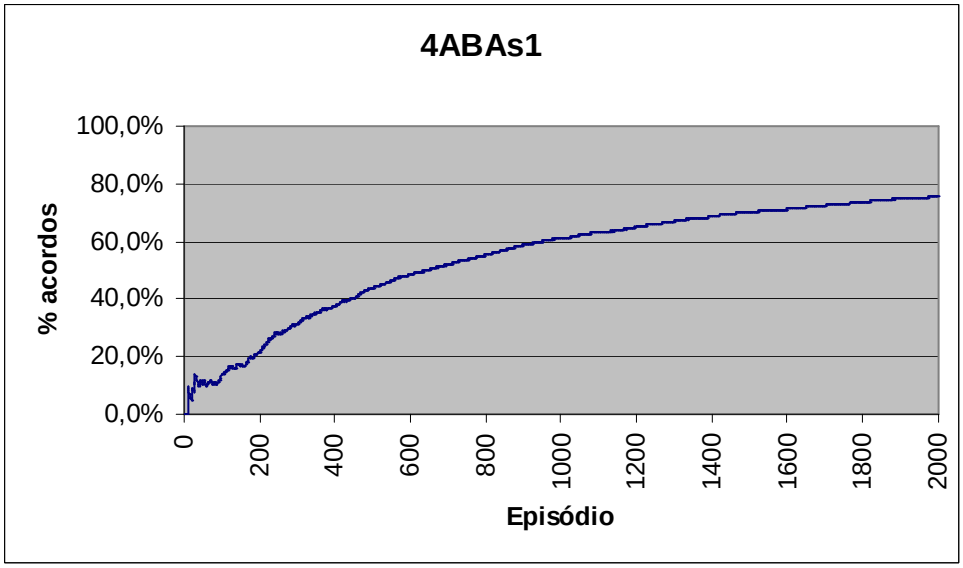


Figura 10.7 Percentagem de acordos obtidos pelo agente 4ABAs1

O agente 4ABAs1 conseguiu uma utilidade média de 0.21, praticamente constante desde o episódio 200. Na figura 10.8, é possível verificar uma aproximação notória das utilidades obtidas em cada um dos acordos em relação à utilidade média.

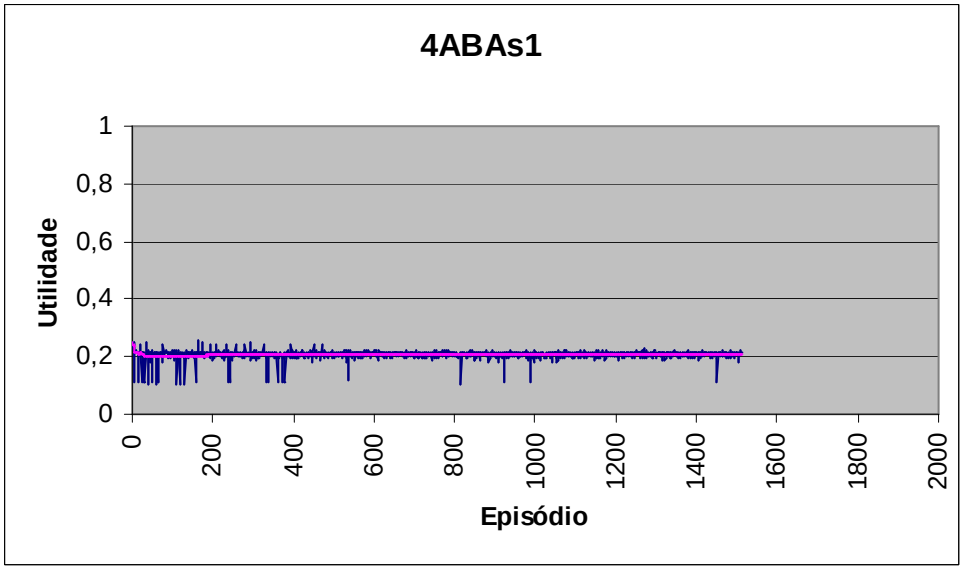


Figura 10.8 Utilidades e utilidade média do agente 4ABAs1

De uma forma geral, os resultados não foram tão bons como se esperava. A tabela 10.5 resume o desempenho obtido pelos agentes ABA, comparando-o com o máximo que eles poderiam obter.

	% acordos	*Utilidade média (máximo)*	*Desvio padrão*
1ABAs1	–	0.43 (1)	0.10
2ABAs1	–	0.35 (1)	0.10
3ABAs1	85%	0.38 (0.5)	0.08
4ABAs1	76%	0.21 (0.24)	0.02

Tabela 10.5 Resumo dos resultados das experiências com agentes ABA

O comportamento dos agentes ABA nos cenários 1 e 2 deixa bastante a desejar, pois a utilidade média obtida não se aproxima do máximo valor possível de obter. O desvio padrão, nestes dois cenários, mostra que os agentes ABA tinham ainda um leque considerável de acções em uso frequente (nestes cenários fixos, uma acção levará sempre ao mesmo valor de utilidade resultante).

Lendo os resultados dos cenários 3 e 4 depreende-se, como seria de esperar, que com espaços de manobra sucessivamente menores os resultados aproximam-se do topo das possibilidades. De facto, o número de acções que resultam na obtenção de acordos diminui no cenário 3 e torna-se muito pequeno no cenário 4. Os agentes 3ABAs1 e 4ABAs1 facilmente se aperceberam disto, daí o seu grande aumento na percentagem de acordos obtidos. Contudo, e tendo em conta a estabilidade destes cenários, no que diz respeito à utilidade os resultados não foram satisfatórios. Os 2000 episódios executados deveriam permitir desempenhos melhores.

10.2.3 Análise dos Resultados

Uma análise mais cuidada permitiu detectar a razão destes resultados menos bons. Valores de utilidade baixos significam que os acordos foram obtidos em espaços de tempo relativamente curtos, dado que os oponentes dos agentes ABA foram criados com uma táctica linear, excepto no cenário 2. Os agentes ABA não foram capazes de aprender a esperar o maior espaço de tempo possível pela cedência dos seus oponentes.

Esta interpretação é também válida para os cenários 2 e 4, embora se verifiquem condições mais difíceis para a obtenção de melhores resultados.

Uma réplica do cenário 1, mas em que o agente ABA foi configurado com o parâmetro de aprendizagem $\tau = 0.9$, não permitiu obter melhores resultados. Uma temperatura τ mais elevada aumenta a exploração das acções.

Tal como referido na secção 9.2, a implementação do algoritmo de aprendizagem-Q nos agentes ABA não considerou reforços em estados intermédios. Isto significa que a actualização do valor Q, associado a acções em estados intermédios, é feita apenas com base no máximo valor para as acções do estado seguinte. Como todos os valores Q de todas as acções são inicialmente nulos, aqueles que são actualizados primeiro correspondem a acções que conduzem directamente ao estado final, onde o reforço é aplicado. O reforço é depois, noutros episódios, transmitido de uma forma descontada a acções do estado anterior e assim sucessivamente.

O algoritmo de aprendizagem, tal como foi implementado, privilegia acções que reflectem comportamentos concedentes, pois desta forma a negociação passa por menos estados, levando a uma actualização mais rápida dos valores Q das acções iniciais. As acções de maiores valores Q têm maior probabilidade de serem escolhidas novamente. A figura 10.9 pretende ilustrar o problema. No exemplo ilustrado, a acção B só verá o seu valor Q actualizado após ser utilizada pela terceira vez, se o agente passar pelos mesmos estados intermédios.

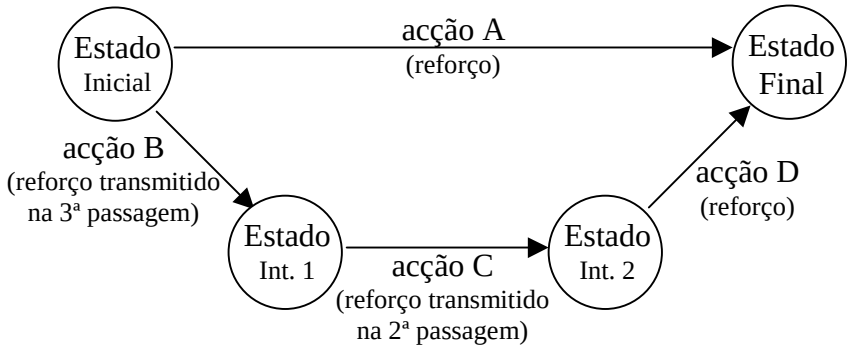


Figura 10.9 Transmissão de reforços

Para validar esta teoria explicativa, os cenários 1 e 3 foram reproduzidos, mas com uma alteração: os agentes ABA foram configurados com o parâmetro de aprendizagem

Time slice = 100. Este valor significa que, quanto ao tempo disponível para negociar, apenas um estado é considerado. Dado que estes cenários são fixos (os agentes em cena são sempre os mesmos), temos apenas um estado em cada episódio. Todas as acções serão portanto reforçadas no momento em que são utilizadas, uma vez que todas conduzem ao estado final. Os resultados obtidos confirmaram a teoria, mostrando uma melhoria substancial das utilidades médias obtidas.

Outra razão para a maior facilidade de obtenção de acordos em relação à aptidão para obter boas utilidades prende-se com a forma de cálculo do reforço. O valor de reforço negativo, atribuído como penalização quando o agente perde o acordo, foi –1. Às acções vitoriosas foi atribuído um reforço igual à utilidade obtida no acordo (variando portanto entre 0 e 1). A máxima variação entre duas acções bem sucedidas é portanto de 1, correspondente à distância mínima para a penalização. Esta encontra-se bem demarcada dos valores de reforço em caso de acordo, daí que seja mais fácil para o agente bater o seu competidor do que obter as melhores utilidades.

10.3 Melhorias no Algoritmo de Aprendizagem

Com vista a acelerar o processo de aprendizagem e a obter melhores resultados, algumas hipóteses de alteração do algoritmo de aprendizagem-Q implementado foram estudadas.

10.3.1 Valores Iniciais Optimistas

A evolução dos valores Q das acções e, consequentemente, a escolha das mesmas, é influenciada pelos valores iniciais estimados. Estes valores iniciais podem ser utilizados como uma maneira de fornecer algum conhecimento prévio ao sistema, sobre o nível de reforços que podem ser esperados [Sutton e Barto (1998)].

A utilização de valores iniciais optimistas (valores acima dos reforços esperados) permite também encorajar a exploração inicial. À medida que as acções vão sendo escolhidas, os seus valores são reduzidos para valores mais realistas, devido à obtenção de reforços mais baixos do que os valores iniciais. Mesmo quando a melhor acção é sempre escolhida, o sistema explora bastante numa fase inicial, até que todas as acções tenham sido utilizadas.

O problema da preferência de acções que conduzem directamente ao estado final, descrito na secção 10.2.3, pode ser resolvido com esta abordagem. De facto, utilizando valores iniciais optimistas, as acções cujos valores Q são alterados apenas na N^{ésima} passagem, serão escolhidas N vezes antes das acções mais concedentes serem executadas novamente.

Outra forma de melhorar os resultados obtidos pode passar por estimar os valores Q das acções de acordo com a utilidade esperada com a sua execução. Assim, acções mais ambiciosas terão valores iniciais mais altos do que acções mais concedentes, sendo portanto experimentadas primeiro. Esta solução é contudo mais trabalhosa, implicando ainda um conhecimento sobre a curva de concessão de cada uma das acções (recorde-se que, para os agentes ABA, acções são combinações ponderadas de tácticas que obedecem a vários critérios).

10.3.2 Função de Reforço

A obtenção de melhores utilidades, nos acordos efectuados pelos agentes ABA, pode ser facilitada pela distorção da função (até agora linear) de cálculo dos reforços empregues. Deve ser dada maior importância ao alcance das melhores utilidades possíveis.

Uma abordagem mais ingénua poder-nos-ia levar a uma função que, para a gama de valores de [0, 1], desse mais peso a utilidades próximas do máximo e menos peso a utilidades baixas, como por exemplo a ilustrada na figura 10.10.

Em cenários onde o agente ABA tem de competir com outros agentes esta função não é aceitável. Por exemplo, no caso do cenário 3 dos testes anteriores, a utilidade máxima possível para o agente ABA era de 0.5. Ora, para valores de utilidade no intervalo [0, 0.5] a função aproxima os valores, em vez de os afastar.

A função a empregar deve ser sensível ao tipo de cenário que o agente enfrenta. Um dado que nos permite ter alguma informação sobre a dificuldade do cenário em causa é a própria utilidade obtida nos acordos alcançados pelo agente. A maior utilidade conseguida até ao momento seria uma boa referência, mas apenas para cenários fixos. Considerando a hipótese de cenários dinâmicos, uma boa medida será a média das

utilidades obtidas. A figura 10.11 ilustra uma função de cálculo do reforço baseada nesta média.

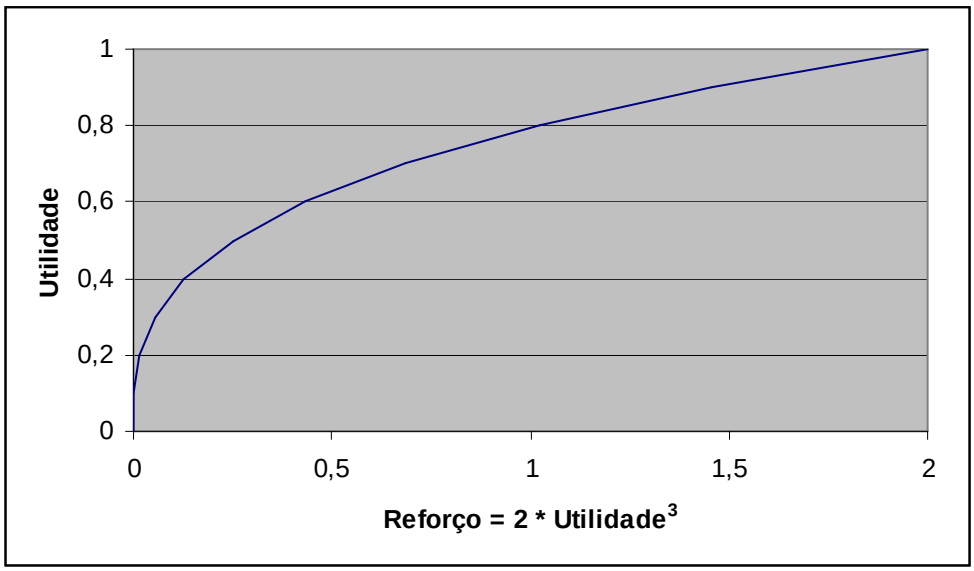


Figura 10.10 Função ingénua de cálculo do reforço

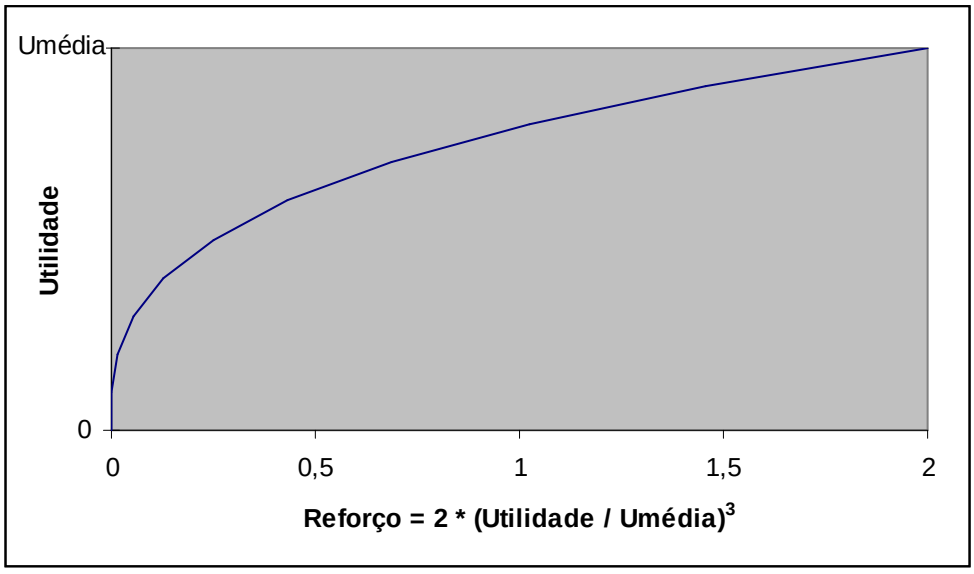


Figura 10.11 Função de cálculo do reforço baseada na utilidade média

Utilizando uma função como esta, o agente pode obter, para acordos com a mesma utilidade, reforços diferentes, dependendo da utilidade média obtida até então. Esta variação representa a percepção do agente sobre o tipo de cenário que está a enfrentar. Outro facto a ter em conta é que valores de utilidade fora do intervalo [0, Umédia]

obtêm valores de reforço superiores a 2. Valores de utilidade que contribuem para o aumento da média são considerados muito bons, daí que recebam reforços muito altos.

10.3.3 Técnica dos "After-States"

A aprendizagem pode ainda ser aperfeiçoada tendo em conta as características específicas do problema em causa. Um conceito que poderia ser aplicado neste caso baseia-se ao referido em Sutton e Barto (1998) como *"after-states"*.

Este conceito consiste em tirar partido do possível conhecimento sobre o determinismo das diversas acções. Após a chegada a um determinado estado, o algoritmo de aprendizagem terá em conta não só o estado anterior e a acção que levou ao estado actual, mas sim todos os caminhos que poderiam ter conduzido até aqui. Pretende-se desta forma acelerar o processo de aprendizagem, através da actualização simultânea dos valores Q de um conjunto de acções.

No caso concreto dos agentes ABA, qualquer acção mais ambiciosa do que uma acção executada levará ao mesmo estado seguinte. Há duas condições para que isto se verifique: o estado seguinte não corresponde à obtenção de um acordo e os oponentes do agente ABA não utilizam tácticas dependentes do comportamento (as suas propostas não dependem dos valores propostos pelo agente ABA). No caso de obtenção de acordo, as acções mais concedentes do que a acção executada obteriam o reforço associado ao valor proposto, resultado da aplicação dessas acções.

Além de permitir acelerar o processo de aprendizagem (que, na prática, consiste em corrigir os valores Q das acções), a implementação deste mecanismo pode reduzir o espaço de procura, que é especialmente grande quando se utilizam valores iniciais optimistas, dada a elevada exploração inicial. Por outro lado, requer o conhecimento necessário à ordenação das acções segundo o seu grau de concessão, o que, para um leque variado de acções, se pode tornar bastante difícil.

10.4 Experiências com Agentes ABA Melhorados

As alterações introduzidas nos agentes ABA, com vista à melhoria dos resultados obtidos, correspondem às duas primeiras alterações referidas na secção anterior. Todas as acções são agora inicializadas optimisticamente com o valor Q = 5. A função de

reforço utilizada é a ilustrada na figura 10.11. Estes agentes ABA melhorados serão referidos como agentes ABAII.

Para avaliar o efeito de tais melhoramentos, foi conduzido mais um grupo de experiências, com base nos cenários descritos a seguir.

10.4.1 Configuração dos Cenários

Neste grupo de experiências, foram construídos cenários que permitissem comparar os resultados com os obtidos anteriormente. Assim, foram recriados os cenários 1 a 4, mas com alterações no que diz respeito à configuração dos agentes ABAII. Estes foram criados com três tácticas dependentes do tempo, polinomiais, com $\kappa = 0$ e valores de $\beta = \{0.05; 1; 20\}$. Esta opção permite reduzir o espaço de procura, assegurando a mesma gama de comportamentos. Os agentes MTA, tal como anteriormente, foram criados com uma táctica dependente do tempo, polinomial e com $\kappa = 0$. A tabela 10.6 resume a configuração destes cenários.

Cenário	*Agentes*	*Previsões para o agente ABAII*
1	1ABAIIs1 1MTAb1: linear ($\beta = 1$)	Aumento da utilidade.
2	2ABAIIs1 2MTAb1: ambicioso ($\beta = 0.5$)	Aumento da utilidade (mais lento que no cenário 1).
3	3ABAIIs1 3MTAb1: linear ($\beta = 1$) 3MTAs2: linear ($\beta = 1$)	Aumento do número de acordos, seguido de aumento da utilidade até ao melhor possível.
4	4ABAIIs1 4MTAb1: linear ($\beta = 1$) 4MTAs2: concedente ($\beta = 5$)	Aumento do número de acordos, seguido de aumento da utilidade até ao melhor possível (mais lento que no cenário 3).

Tabela 10.6 Configuração dos cenários com agentes ABAII

Tal como nas experiências iniciais, todos os agentes foram configurados com a mesma gama de valores para a única dimensão de negociação utilizada, tendo também o mesmo tempo máximo de negociação de 3 minutos.

Os parâmetros de aprendizagem utilizados nos agentes ABAII foram os seguintes: *Softmax*, $\tau = 0.2$, $\alpha = 0.5$, $\gamma = 1$, *Time slice = 10*, *Gradient = 1.0*. Dado que a exploração inicial foi aumentada pela utilização de valores iniciais optimistas, após o agente encontrar as melhores acções a exploração deve ser menor. Daí a escolha de um valor de temperatura τ baixo. A taxa de aprendizagem α foi aumentada de modo a permitir alterações mais significativas dos valores Q das acções. O factor de desconto γ foi fixado de modo a não descontar os valores Q futuros, pois estes são importantes dada a inexistência de reforços em estados intermédios. Os parâmetros foram configurados com estes valores com vista a acelerar o processo de aprendizagem.

Além destes, foram desenhados dois cenários simples, para possibilitar a análise do desempenho dos agentes em situações minimamente dinâmicas, com a entrada de novos agentes em momentos diferentes. A tabela 10.7 resume a configuração dos dois cenários dinâmicos desenhados. A terceira coluna refere-se ao período de tempo relativo dentro do qual o agente foi lançado.

Cenário	*Agentes*	*Lançamento*	*Previsões para o agente ABAII*
5	5ABAIIs1 5MTAb1: *linear* ($\beta = 1$) 5MTAs2: *linear* ($\beta = 1$) 5MTAb2: *linear* ($\beta = 1$)	0% 0% 0% 40%	Aumento do número de acordos com 5MTAb2, acompanhado de aumento da utilidade.
6	6ABAIIs1 6MTAb1: *linear* ($\beta = 1$) 6MTAs2: *linear* ($\beta = 1$) 6MTAb2: *linear* ($\beta = 1$)	0% 0% 0% 50-80%	Aumento do número de acordos com 6MTAb2, acompanhado de aumento da utilidade.

Tabela 10.7 Configuração dos cenários dinâmicos com agentes ABAII

Nestes dois cenários, aos agentes ABAII foi adicionada a táctica dependente de recursos de tempo máximo de negociação dinâmico, com uma função polinomial linear ($\beta = 1$) e com $\kappa = 0$. Foi escolhido um valor de $\mu = tmax$.

No cenário 5, o agente 5ABAIIs1 deverá aprender a estabelecer os acordos com o agente 5MTAb2, pois só desta forma ele poderá obter utilidades acima de 0.5. A figura 10.12 fornece uma visão da evolução das curvas de negociação dos agentes MTA deste cenário, indicando no eixo das ordenadas a utilidade das propostas destes agentes para um agente ABAII vendedor.

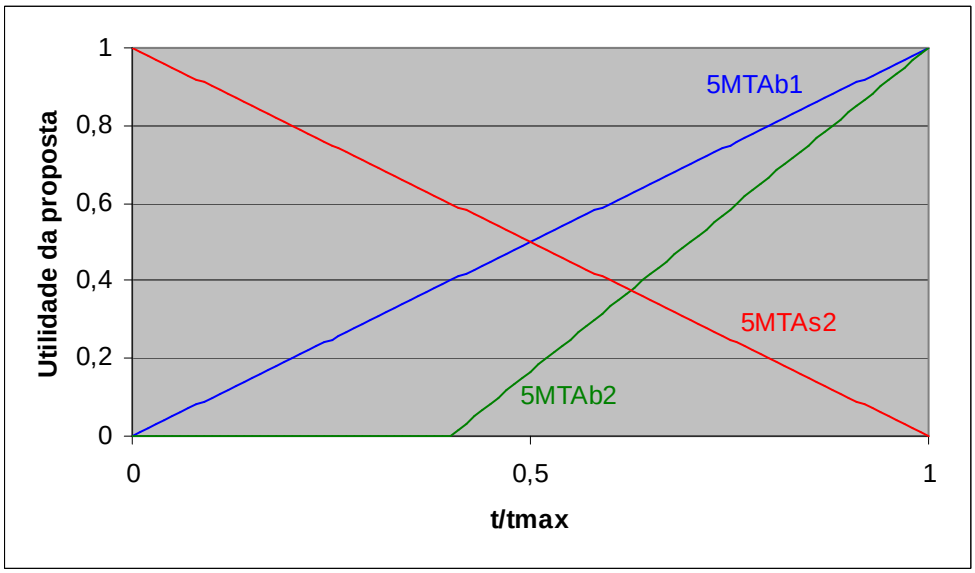


Figura 10.12 Curvas de negociação dos agentes MTA no cenário 5

No cenário 6, pela mesma razão, o agente 6ABAIIs1 deverá aprender a realizar acordos com o agente 6MTAb2. Contudo, a aprendizagem neste caso é mais difícil, dada a variação do tempo de lançamento deste agente nos diversos episódios. O agente 6ABAIIs1 deverá aprender a esperar sempre pela cedência do seu oponente, qualquer que seja o seu tempo de lançamento.

10.4.2 Resultados

Tal como nas experiências anteriores, foram executados 2000 episódios em cada um dos cenários descritos.

No cenário 1, o agente 1ABAIIs1 alcançou uma utilidade média de 0.69, com algumas utilidades a chegarem ao valor de 0.9. No final dos 2000 episódios, a utilidade média mostrava-se crescente, como se pode verificar pela figura 10.13.

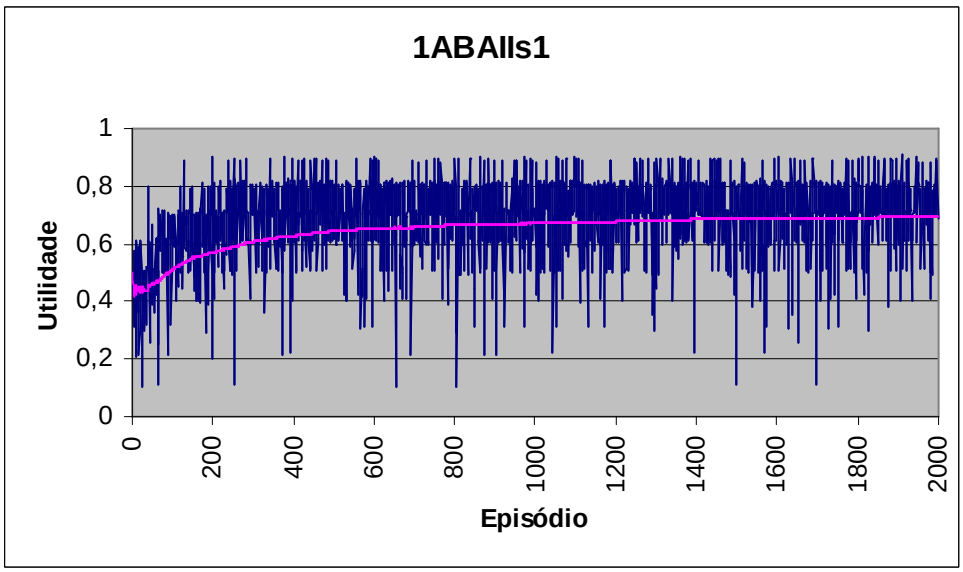


Figura 10.13 Utilidades e utilidade média do agente 1ABAIIs1

No cenário 2, o agente 2ABAIIs1 obteve uma utilidade média de 0.67. A figura 10.14 permite visualizar algumas utilidades de cerca de 0.85. A utilidade média encontrava-se com andamento crescente.

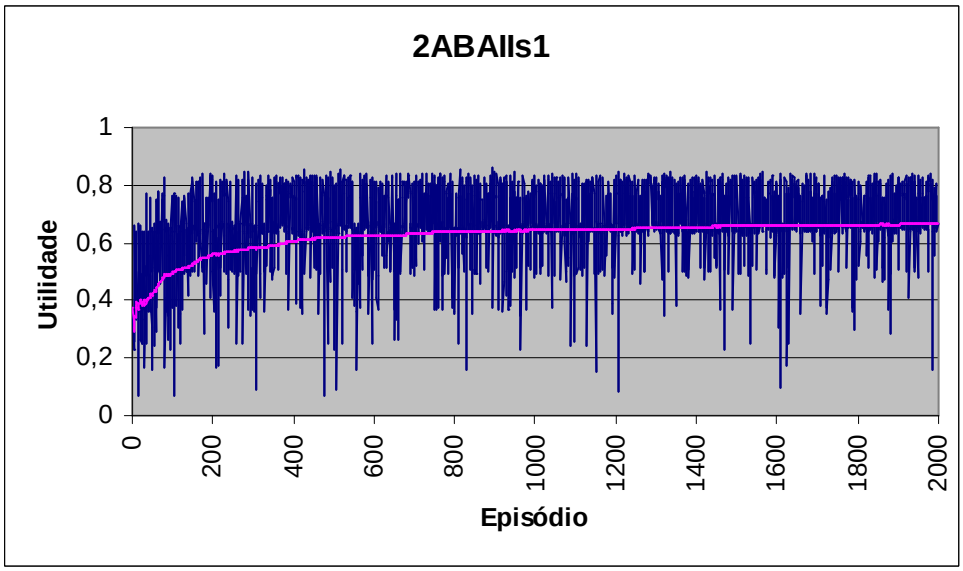


Figura 10.14 Utilidades e utilidade média do agente 2ABAIIs1

No cenário 3, o agente 3ABAIIs1 foi claramente superior ao seu competidor. Após um período inicial de cerca de 200 episódios, a sua percentagem de acordos obtidos não parou de subir, como mostra a figura 10.15. O agente 3ABAIIs1 conseguiu 93% dos acordos.

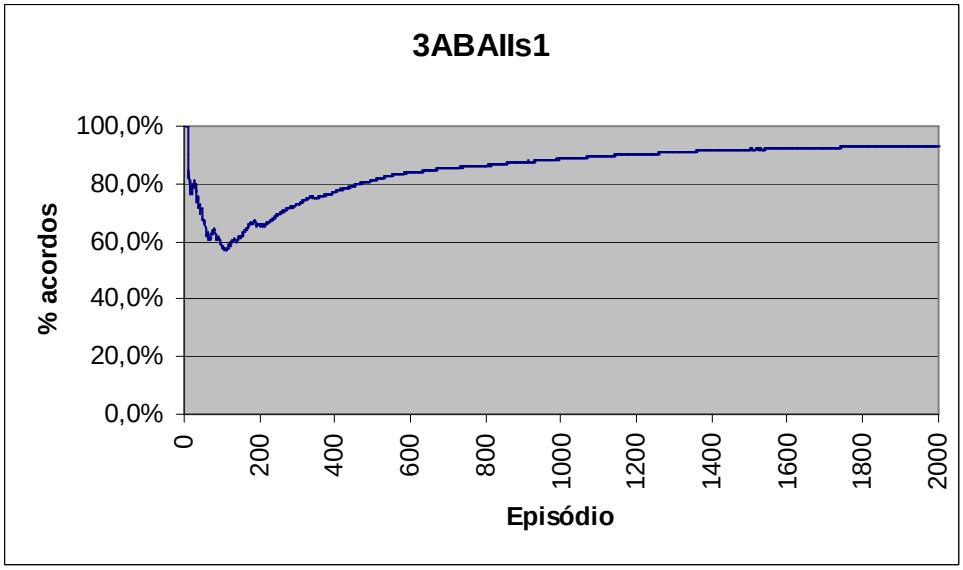


Figura 10.15 Percentagem de acordos obtidos pelo agente 3ABAIIs1

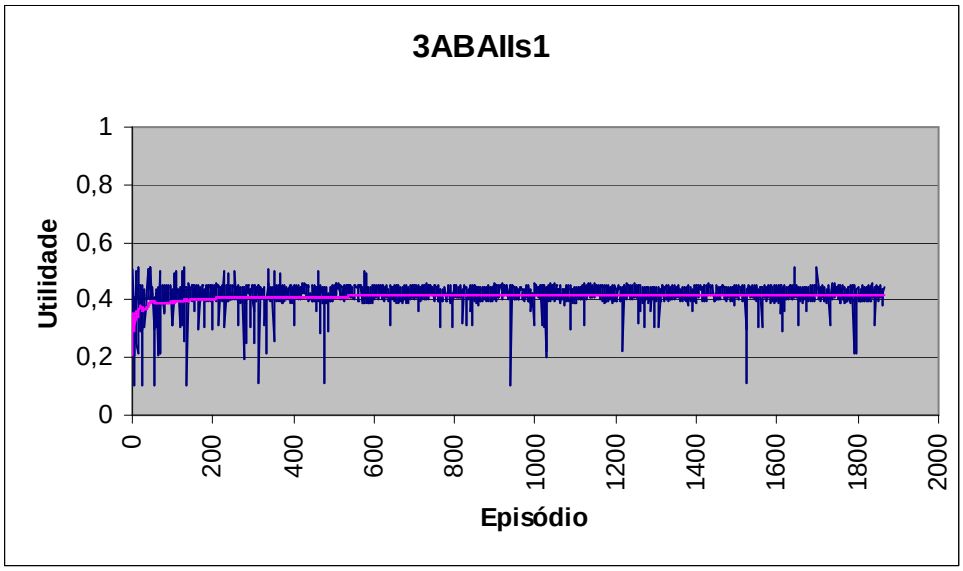


Figura 10.16 Utilidades e utilidade média do agente 3ABAIIs1

Nos acordos efectuados, o agente 3ABAIIs1 obteve uma utilidade média de 0.42. A figura 10.16 mostra que esta se manteve, após um crescimento inicial, praticamente constante, com as utilidades obtidas a terem uma variação pequena.

No cenário 4, o agente 4ABAIIs1 não teve dificuldades em bater o seu competidor. Como mostra a figura 10.17, após cerca de 100 episódios iniciais, o agente não parou de aumentar a sua percentagem de acordos obtidos, que chegou aos 93%.

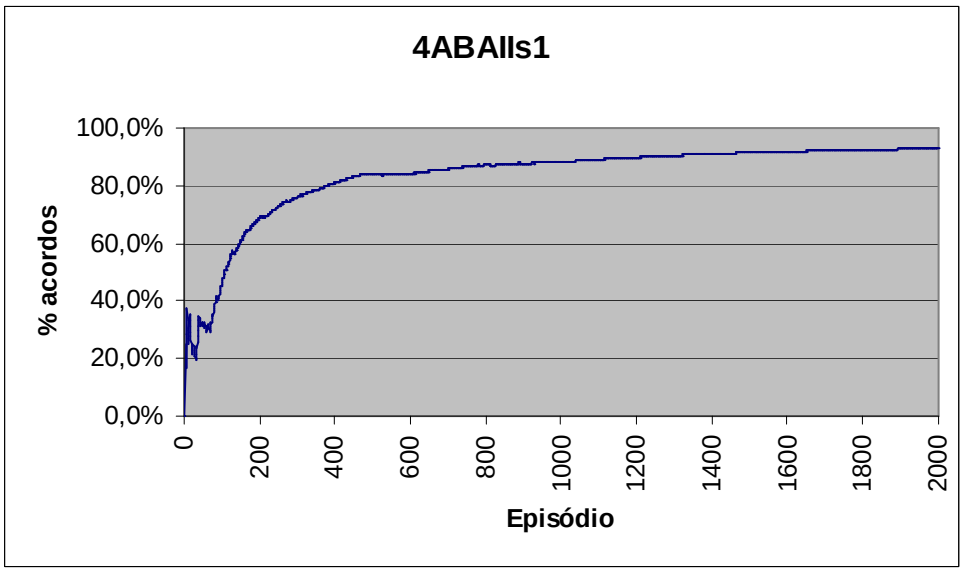


Figura 10.17 Percentagem de acordos obtidos pelo agente 4ABAIIs1

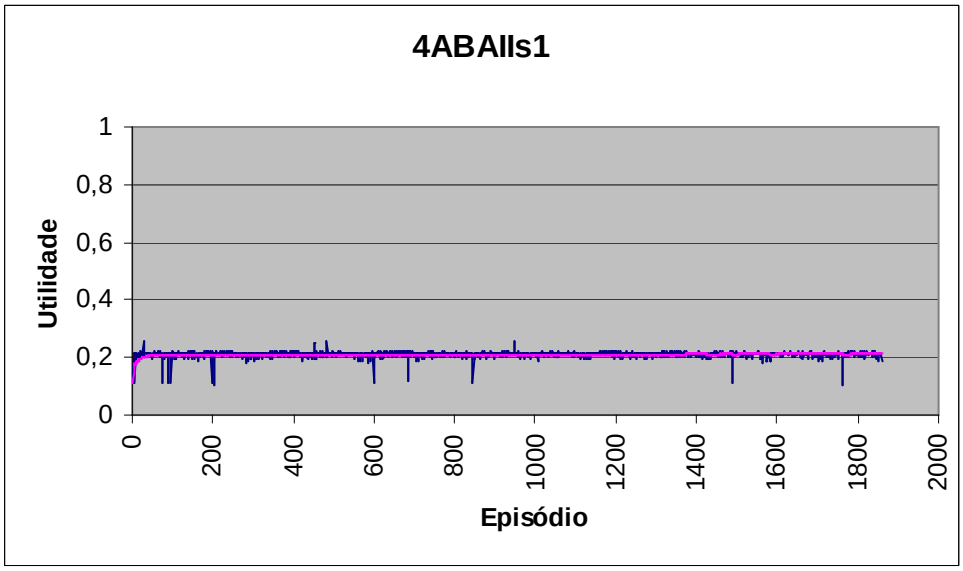


Figura 10.18 Utilidades e utilidade média do agente 4ABAIIs1

A utilidade média obtida pelo agente 4ABAIIs1 nos acordos efectuados fixou-se nos 0.21. A figura 10.18 mostra que a variação das utilidades obtidas foi praticamente nula, daí que a utilidade média se mantivesse constante desde o início.

Comparando com os resultados obtidos nas experiências com agentes ABA, estes resultados apresentam melhorias significativas. De facto, os agentes ABAII obtiveram um melhor desempenho, conforme mostra a tabela 10.8.

Cenário	*% acordos*		*Utilidade média (máximo)*		*Desvio padrão*	
	ABA	*ABAII*	*ABA*	*ABAII*	*ABA*	*ABAII*
1	–	–	0.43 (1)	0.69 (1)	0.10	0.13
2	–	–	0.35 (1)	0.67 (1)	0.10	0.14
3	85%	93%	0.38 (0.5)	0.42 (0.5)	0.08	0.04
4	76%	93%	0.21 (0.24)	0.21 (0.24)	0.02	0.01

Tabela 10.8 Resumo dos resultados das experiências com agentes ABAII

No que diz respeito aos cenários 1 e 2, a melhoria das utilidades médias obtidas é notória. Os gráficos que mostram a evolução das utilidades obtidas pelos agentes permitem observar o resultado da exploração inicial acrescida, levada a cabo pelos agentes ABAII: a faixa dentro da qual variam as utilidades obtidas tem uma elevação evidente na parte inicial das experiências. A descoberta destes valores mais altos de utilidade resultou, por outro lado, num aumento do desvio padrão. A exploração inicial torna os agentes ABAII menos sensíveis ao seu próprio sucesso dos primeiros episódios. As primeiras utilidades não são adoptadas como resultados a repetir, daí que as utilidades médias destes agentes se encontrassem em crescimento.

Os cenários 3 e 4 mostram que os agentes ABAII são mais rápidos em aprender a bater os seus competidores. A melhor utilidade média obtida no cenário 3 deve-se ao decréscimo rápido do desvio padrão. O agente conseguiu encontrar, em relativamente poucos episódios, as acções que conduzem às melhores utilidades possíveis. No cenário 4 não foi possível aumentar a utilidade média, dada a dificuldade da situação apresentada. Contudo, o desvio padrão diminuiu para metade do obtido anteriormente.

A figura 10.19 permite comparar visualmente a evolução dos desvios padrão dos agentes ABA e ABAII, nos cenários 3 e 4.

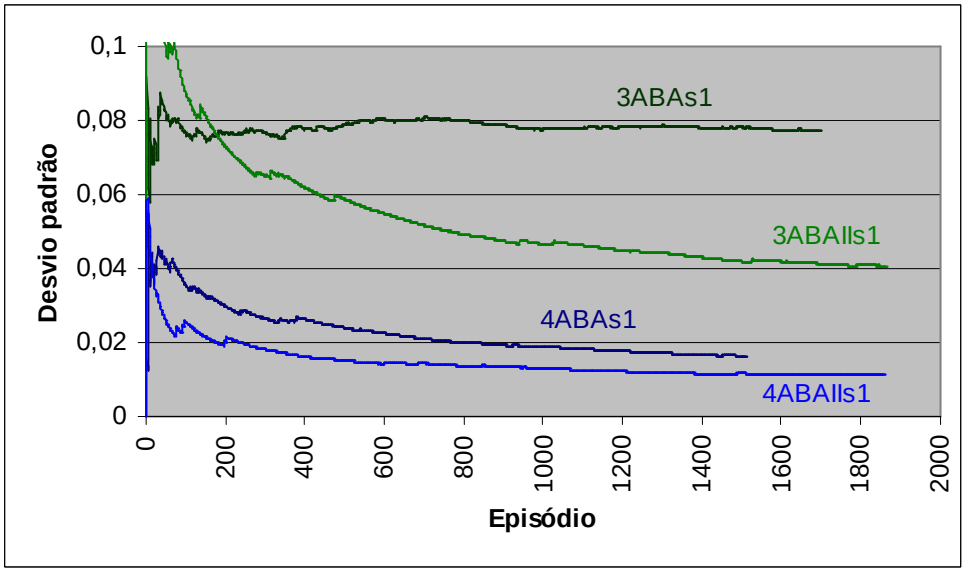


Figura 10.19 Evolução dos desvios padrão dos agentes ABA e ABAII nos cenários 3 e 4

As utilidades máximas teóricas apresentadas não foram alcançadas frequentemente, talvez devido à incerteza dos resultados obtidos quando se escolhem as acções que a elas podem levar. Problemas relacionados com o desempenho do próprio sistema SMACE, até agora ignorados, podem ter levado a este défice no rendimento dos agentes. As situações de acordo no extremo do tempo de negociação, onde os agentes ABAII poderiam obter os melhores resultados teóricos, são portanto incertas.

Os resultados dos cenários 5 e 6 permitem concluir que os agentes 5ABAIIs1 e 6ABAIIs1 foram capazes de aprender a obter acordos com os agentes 5MTAb2 e 6MTAb2, respectivamente. Só com estes agentes eles poderiam obter utilidades acima de 0.5, observáveis nas figuras 10.20 e 10.21. Os agentes ABAII conseguiram uma utilidade média superior a 0.5 em ambos os cenários. Contudo, dado o pequeno dinamismo introduzido nos cenários, verificaram-se desvios padrão bastante elevados, na ordem dos 0.16 pontos.

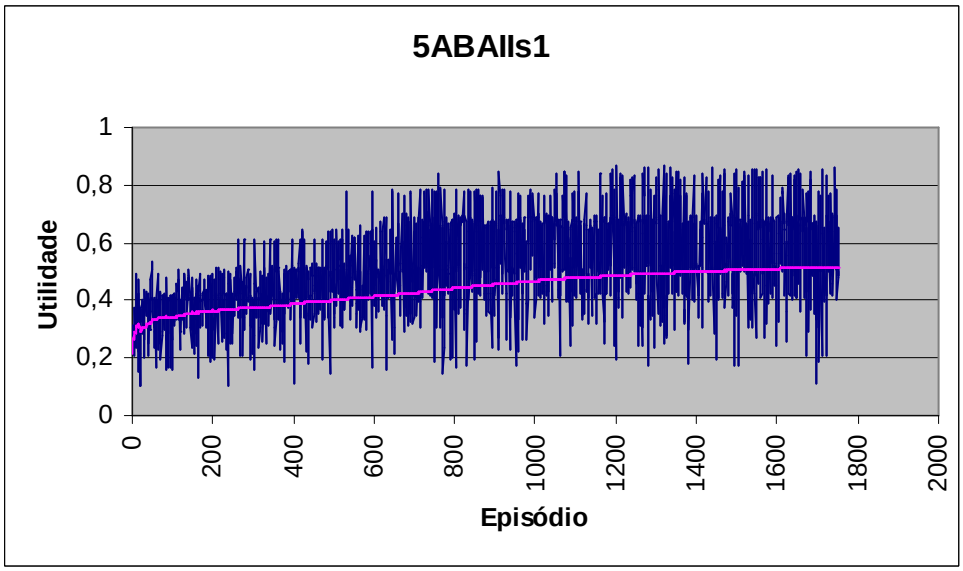


Figura 10.20 Utilidades e utilidade média do agente 5ABAIIs1

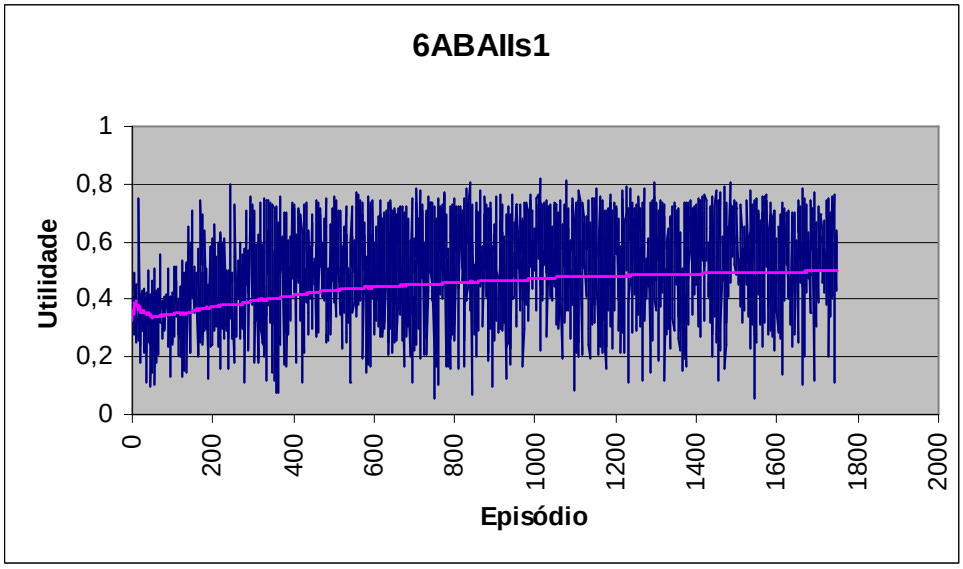


Figura 10.21 Utilidades e utilidade média do agente 6ABAIIs1

11. Conclusões

A evolução da Internet aponta para um forte crescimento no número de transacções feitas “on-line”. Em particular, tem-se observado um aumento significativo do número de leilões electrónicos, cada vez mais populares. As razões que apontam para este fenómeno são a sua novidade e o seu valor lúdico na negociação do preço dos produtos, bem como a possibilidade de se obterem bons negócios na aquisição dos produtos pretendidos [Guttman e Maes (1998a)].

Existem já diversos sítios na “Web” com agentes de “software”, que automatizam a busca e filtragem de informação relativa a produtos. Contudo, a utilização de agentes compradores e vendedores em Comércio Electrónico está ainda a dar os primeiros passos. Segundo Chavez *et al.* (1997), o nível de inteligência ou sofisticação que agentes compradores ou vendedores podem possuir não está restringido por limitações da tecnologia da Inteligência Artificial, mas sim por questões relacionadas com a confiança do utilizador. Para que os agentes sejam bem aceites neste domínio, é crucial que o seu comportamento possa ser facilmente compreendido e controlado pelo utilizador.

No caso concreto do sistema SMACE, a tecnologia baseada em agentes de “software” foi adequada. Esta abordagem permitiu delegar nos agentes intenções de compra ou venda, representando os interesses dos intervenientes no Comércio Electrónico. Esses agentes conduzem depois o processo de negociação de uma forma autónoma.

A plataforma JATLite, utilizada na construção do sistema SMACE, também pareceu apropriada a este domínio. Apesar do esforço inicial considerável para integrar o mercado electrónico com esta plataforma, essa integração foi satisfatória. A utilização do JATLite trouxe vantagens, no que diz respeito à comunicação entre os agentes de “software”, oferecendo ferramentas que nos permitiram ter uma base de partida para a construção de um sistema multi-agente aplicado ao Comércio Electrónico.

A aprendizagem automática em negociação é um processo não trivial. O estudo desenvolvido nesta área está ainda numa fase inicial. Neste trabalho, procurou-se dotar agentes de “software” participantes num mercado electrónico de capacidades de aprendizagem, que lhes permitissem melhorar o seu desempenho. Os resultados obtidos

mostram que é possível, utilizando técnicas de aprendizagem por reforço, alcançar tais melhorias. Em vez de simular situações reais de Comércio Electrónico, procurou-se, através de uma plataforma de testes construída para o efeito, exemplificar situações onde se pudessem observar os resultados de aplicação de uma forma de aprendizagem, baseada na interacção com o ambiente. A principal dificuldade de aplicação deste tipo de aprendizagem neste domínio reside no facto de que os mercados electrónicos, para além de incluírem factores privados dos agentes participantes, são bastante dinâmicos, daí que a adaptação dos agentes ao ambiente se torne mais problemática. De modo a possibilitar uma aplicação com sucesso de tal aprendizagem, esse ambiente terá que denotar algumas características que possam servir de referência no processo de adaptação dos agentes. Por exemplo, o comércio de um determinado produto poderá ser caracterizado por um determinado ritmo de oferta e procura, e os seus interlocutores poderão ter um padrão de comportamento minimamente uniforme.

Na continuação deste trabalho, além de melhorar a integração do algoritmo de aprendizagem-Q nos agentes adaptativos, pretende-se investigar a aplicabilidade de outras formas de aprendizagem, nomeadamente outros tipos de aprendizagem por reforço e algoritmos genéticos. Quanto ao modelo de negociação multi-dimensão, pretende-se estudar formas de correlação entre os valores das dimensões, quer na geração das propostas, quer no próprio processo de aprendizagem.

A arquitectura do sistema SMACE permite, de uma forma relativamente fácil, partir para estas novas direcções. A sua estruturação em camadas facilita também a sua expansão para a inclusão de outros tipos de agentes, que utilizam outras estratégias de negociação.

A disponibilização do SMACE na “Web” para a sua utilização “on-line” está já em curso, requerendo apenas algumas melhorias na interface com o utilizador.

APÊNDICES

A – JATLite

O JATLite é um conjunto de programas escritos na linguagem Java, que permite uma criação rápida de agentes de "software" e sistemas de agentes que comunicam de uma forma robusta pela Internet, com capacidades específicas de comunicação e interacção, mas sem impor nenhuma teoria ou técnica de agentes autónomos. A construção de "agentes inteligentes" é deixada a cargo do programador.

A infra-estrutura do JATLite assenta no *Agent Message Router (AMR)*. Os agentes podem ser portáteis ou mover-se de máquina para máquina. Eles registam-se no AMR usando um nome e palavra-chave, ligam-se e desligam-se da Internet e enviam e recebem mensagens. A comunicação é feita usando o protocolo padrão TCP/IP e "sockets". O AMR é uma aplicação Java que corre numa máquina ligada à Internet.

A arquitectura do JATLite está organizada como uma hierarquia de níveis, como mostra a figura A.1. Esses níveis constituem moldes que facilitam a criação de agentes, consistindo em colecções de classes Java.

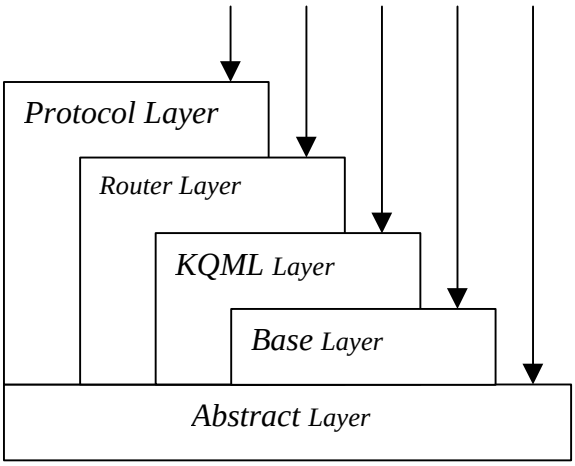


Figura A.1 Níveis do JATLite

Cada um dos níveis acrescenta funcionalidades ao nível anterior:

- *Abstract Layer*: classes abstractas necessárias para a implementação do JATLite;

- *Base Layer*: comunicação básica TCP/IP. Este nível não impõe restrições no que diz respeito à linguagem e protocolo das mensagens;
- *KQML Layer*: impõe a restrição de utilização da linguagem KQML, incluindo facilidades de tratamento de mensagens KQML;
- *Router Layer*: permite a comunicação entre agentes através do *Agent Message Router (AMR)*;
- *Protocol Layer*: suporta alguns serviços padrão da Internet, como o *SMTP* e o *FTP*.

O programador pode escolher o nível apropriado a partir do qual deseja construir o seu sistema. Contudo, para tirar partido das vantagens de utilização do AMR, deve iniciar no *Router Layer* ou no *Protocol Layer*.

O AMR é uma aplicação especializada que recebe mensagens dos agentes registados e encaminha-as para os destinatários correctos. Há quatro aspectos principais relacionados com o desenho do AMR:

- *Polling vs. Queuing* – O *polling* refere-se ao envio de mensagens directamente do agente emissor para o agente receptor. O problema neste tipo de comunicação é que a ligação entre os agentes é crítica, pois sem ela não é possível enviar mensagens. O AMR permite a utilização do *queuing*, no qual as mensagens são armazenadas numa fila de espera para futuro encaminhamento, quando possível (mecanismo de *routing*).
- *Agent Name Service (ANS)* – O AMR funciona como um servidor de ANS, que mantém dinamicamente os endereços dos agentes, que podem mudar. Esta funcionalidade não é necessária para o envio de mensagens, uma vez que o encaminhamento (*routing*) das mensagens é feito pelo próprio AMR. Contudo, os agentes podem pedir a este o endereço de um determinado agente.
- *Comunicação de Agentes Applet* – Os *applets* têm uma restrição de segurança, que lhes limita a comunicação para a máquina na qual reside o servidor HTTP que os lançou. O AMR permite que agentes implementados como *applets* comuniquem com agentes noutras máquinas, funcionando como um *proxy*.

- *Problema de Escalabilidade* – O envio de muitas mensagens para o AMR pode afectar o seu desempenho. Para contornar este problema podem ser usados vários AMRs, conforme necessário.

B – KQML

A linguagem KQML (*Knowledge Query Manipulation Language*), desenvolvida pelo consórcio *ARPA Knowledge Sharing Effort*, é uma linguagem para a troca de informação e conhecimento.

Conceptualmente, a linguagem KQML está dividida em três níveis [Finin *et al.* (1994, 1997)]:

- *conteúdo*: é o conteúdo da mensagem propriamente dita, na linguagem de representação própria dos agentes – o KQML pode incluir qualquer linguagem de representação, sendo o conteúdo ignorado no que diz respeito à sua interpretação (esta tarefa cabe ao agente receptor);
- *mensagem*: envolve o conteúdo da mensagem que um agente quer transmitir a outro – este nível forma o cerne da linguagem KQML e determina os tipos de interacções que os agentes podem ter;
- *comunicação*: acrescenta alguns parâmetros à mensagem relacionados com a comunicação de baixo nível, como a identidade do remetente e do destinatário e um identificador único associado à comunicação.

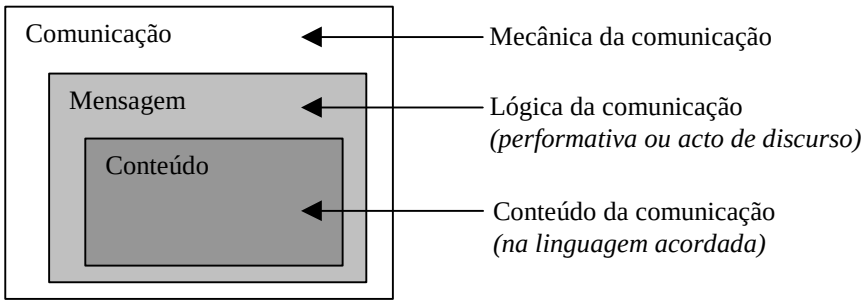


Figura B.1 Níveis do KQML

Uma mensagem KQML consiste numa *performativa* e nos seus argumentos, incluindo o conteúdo da mensagem e um grupo de argumentos opcionais, que o descrevem de uma forma independente da sintaxe da linguagem de conteúdo [Finin *et al.* (1994)]. A performativa indica qual a acção a tomar pelo agente receptor quando recebe a mensagem.

A especificação KQML define um conjunto de performativas reservadas. Este conjunto não é um mínimo requerido que deva ser implementado por todos os agentes que falem KQML, nem é tão pouco um conjunto fechado [Finin *et al.* (1997)]. Um agente KQML pode escolher implementar apenas algumas performativas. O conjunto é extensível, e uma comunidade de agentes pode optar por usar performativas adicionais, logo que acordem na sua interpretação e no protocolo associado a cada uma delas. Contudo, uma implementação que escolha implementar uma das performativas reservadas deve implementá-la na forma padrão.

A performativa inclui-se no nível de mensagem. O conjunto de performativas predefinidas pode-se dividir em três grupos, segundo Labrou e Finin (1997):

- *performativas de discurso*: estas são as performativas a serem usadas no contexto de um discurso de troca de informação e conhecimento entre dois agentes; por exemplo: tell, ask-if, ask-one, ask-all, etc.;
- *performativas de intervenção e mecânica de conversação*: o papel destas performativas é o de intervir no curso normal de uma conversação; por exemplo: error, sorry, discard, etc.;
- *performativas de rede e facilidades*: são principalmente performativas que permitem que os agentes encontrem outros agentes que possam processar as suas questões; por exemplo: register, unregister, forward, etc..

Há também um conjunto de parâmetros predefinidos na especificação KQML. A especificação de parâmetros reservados é útil por duas razões [Labrou e Finin (1997)]:

- permite impor um certo grau de uniformidade na semântica de parâmetros comuns, reduzindo assim a confusão do programador;
- permite suportar algum nível de compreensão, pelos programas, de performativas com nomes desconhecidos mas com parâmetros conhecidos.

A tabela seguinte resume alguns dos parâmetros reservados e seu significado [Labrou e Finin (1997)], incluindo também a sua distribuição pelos três níveis do KQML.

Parâmetro	*Significado*	*Nível*
:sender	remetente da performativa	comunicação
:receiver	destinatário da performativa	comunicação
:from	origem da performativa em :content	comunicação
:to	destino final da performativa em :content	comunicação
:in-reply-to	etiqueta de resposta a uma mensagem anterior	comunicação
:reply-with	etiqueta esperada numa resposta a esta mensagem	comunicação
:language	nome da linguagem de representação do :content	mensagem
:ontology	nome da ontologia assumida no :content	mensagem
:content	informação sobre a qual a performativa expressa uma atitude	conteúdo

Tabela B.1 Parâmetros KQML reservados

C – Mensagens KQML entre Agentes e o Mercado

	Performativa	*:content*	*Significado*	*Resp.*
1	announce	(<obj>)	agente anuncia o seu objectivo no mercado	
2	unannounce		agente retira anúncio do seu objectivo no mercado	
3	what-market-agents	(<obj>)	agente pede nomes de agentes com determinado objectivo	5
4	what-objective	<ag>	agente pede objectivo de determinado agente	6
5	market-agents	((<ag> obj>) (...))	mercado responde com nomes de agentes e seus objectivos	
6	market-agent	(<ag> obj>)	mercado responde com nome de agente e seu objectivo	

Tabela C.1 Mensagens KQML trocadas entre os agentes e o mercado

D – Mensagens KQML entre Agentes de Mercado

	Performativa	*:content*	*Significado*	*Resp.*
1	match-objective	(<obj>)	A questiona B se acha que os objectivos são complementares	2
2	objective-matched	(<obj>)	B confirma e envia objectivo, para verificação de A	3; 5
3	proposal	(<prop>)	proposta/contraproposta	3; 4; 5
4	accept		aceitação de proposta	6; 7
5	withdraw		desistência da negociação	
6	deal		confirmação de acordo	
7	nodeal		rejeição de acordo	

Tabela D.1 Mensagens KQML trocadas entre os agentes

Referências Bibliográficas

Chavez, A, e P. Maes (1996), "Kasbah: An Agent MarketPlace for Buying and Selling Goods", in *Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pp. 75-90.

Chavez, A., D. Dreilinger, R. Guttman e P. Maes (1997), "A Real-Life Experiment in Creating an Agent Marketplace", in *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*.

Faratin, P., C. Sierra e N. R. Jennings (1998), "Negotiation Decision Functions for Autonomous Agents", *International Journal of Robotics and Autonomous Systems*, 24 (3-4), pp. 159-182.

Finin, T., R. Fritzson, D. McKay e R. McEntire (1994), "KQML – A Language and Protocol for Knowledge and Information Exchange", in *Thirteenth International Distributed Artificial Intelligence Workshop*, Seattle WA.

Finin, T., Y. Labrou e J. Mayfield (1997), "KQML as an agent communication language", in *Software Agents*, J. M. Bradshaw (editor), MIT Press.

Guttman, R. H., P. Maes, A. Chavez e D. Dreilinger (1997), "Results from a Multi-Agent Electronic Marketplace Experiment", in *Poster Proceedings of the Eighth European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'97)*, M. Boman et al. (editores), pp. 86-98.

Guttman, R. H. e P. Maes (1998a), "Agent-mediated Integrative Negotiation for Retail Electronic Commerce", in *Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98)*.

Guttman, R. H. e P. Maes (1998b), "Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce", in *Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98)*

Guttman, R. H., A. G. Moukas e P. Maes (1998c), "Agent-mediated Electronic Commerce: A Survey", *Knowledge Engineering Review*.

Harmon, M. E. (1996), *Reinforcement Learning: A Tutorial*, http://www-anw.cs.umass.edu/~mharmon/rltutorial/frames.html

Huhns, M. N. e M. P. Singh (1997), *Readings in Agents*, Morgan Kaufmann.

Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Massachusetts: MIT Press.

Labrou, Y. e T. Finin (1997), "A Proposal for a new KQML Specification", Computer Science and Electrical Engineering Department (CSEE), University of Maryland Baltimore County (UMBC).

Lopes Cardoso, H., M. Schaefer e E. Oliveira (1999), "A Multi-Agent System for Electronic Commerce including Adaptive Strategic Behaviours", in *Proceedings of the 9th Portuguese Conference on Artificial Intelligence (EPIA'99)*, P. Barahona e J. J. Alferes (editores), Progress in Artificial Intelligence, Lecture Notes in Artificial Intelligence, 1695, pp. 252-266, Springer-Verlag.

Martín, F. J., E. Plaza, J. A. Rodríguez-Aguilar e J. Sabater (1998), "JIM, A Java Interagent for Multi-Agent Systems", in *Proceedings of the AAAI-98 Workshop on Software Tools for Developing Agents*.

Matos, N., C. Sierra e N. R. Jennings (1998), "Determining Successful Negotiation Strategies: An Evolutionary Approach", in *Proceedings, Third International Conference on Multi-Agent Systems (ICMAS-98)*, pp. 182-189, IEEE Computer Society.

Michalski, R. S., J. G. Carbonell e T. M. Mitchell (1984), *Machine Learning*, Springer-Verlag.

Moukas, A., R. Guttman e P. Maes (1998), "Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective", in *Proceedings of the First International Conference on Electronic Commerce (ICEC'98)*.

O'Hare, G. M. P. e N. R. Jennings (1996), *Foundations of Distributed Artificial Intelligence*, New York: John Wiley & Sons.

Pearl, J. (1986), "Fusion, propagation, and structuring in belief networks", in *Artificial Intelligence*, 29, pp. 241-288.

Pruitt, D. G. (1981), *Negotiation Behavior*, New York: Academic Press.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, San Mateo: Morgan Kaufmann.

Raiffa, H. (1982), *The Art and Science of Negotiation*, Cambridge: Harvard University Press.

Rocha, A. P. e E. Oliveira (1999), "Electronic Commerce: a technological perspective", in *O Futuro da Internet*, J. A. Alves et al. (editores), pp. 127-138, Centro Atlântico.

Rodríguez-Aguilar, J. A., P. Noriega, C. Sierra e J. Padget (1997), "FM96.5 A Java-based Electronic Auction House", in *Proceedings of the Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*.

Rodríguez-Aguilar, J. A., F. J. Martín, P. Noriega, P. Garcia e C. Sierra (1998a), "Competitive Scenarios for Heterogeneous Trading Agents", in *Proceedings of the Second International Conference on Autonomous Agents (Agents'98)*, K. P. Sycara e M. Wooldridge (editores), pp. 293-300, ACM Press.

Rodríguez-Aguilar, J. A., F. J. Martín, P. Noriega, P. Garcia e C. Sierra (1998b), "Towards a Test-bed for Trading Agents in Electronic Auction Markets", AI Communications.

Romm, C. T. e F. Sudweeks (1998), *Doing Business Electronically*, London: Springer-Verlag.

Rumelhart, D. E. e J. L. McClelland (1986), *Parallel Distributed Processing*, Massachusetts: MIT Press.

Russell, S. e P. Norvig (1995), *Artificial Intelligence: A Modern Approach*, New Jersey: Prentice Hall.

Schaefer, M. (1999), *Adaptive Behaviour Agents in Electronic Commerce*, Faculdade de Engenharia da Universidade do Porto.

Shmeil, M. A. H. e E. Oliveira (1995), "Detecting the Opportunities of Learning from the Interactions in a Society of Organizations", in *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence (SBIA'95)*, J. Wainer e A. Carvalho (editores), Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence, 991, pp. 242-252, Springer-Verlag.

Smith, R. G. (1988), "The Contract Net Protocol: High-level communication and control in a distributed problem solver", in *Readings in Distributed Artificial Intelligence*, A. H. Bond e L. Gasser (editores), San Mateo: Morgan Kaufmann.

Sutton, R. S. e A. G. Barto (1998), *Reinforcement Learning: An Introduction*, Cambridge: MIT Press.

Sycara, K. (1992), "The PERSUADER", in *The Encyclopedia of Artificial Intelligence*, D. Shapiro (editor), New York: John Wiley & Sons.

Terpsidis, I. S., A. Moukas, B. Pergioudakis, G. Doukidis e P. Maes (1997), "The potential of Electronic Commerce in re-engineering consumer-retailer relationships through Intelligent Agents", in *Proceedings of the European Conference on MM & E-Commerce*.

Wurman, P. R., W. E. Walsh e M. P. Wellman (1998a), "Flexible Double Auctions for Electronic Commerce: Theory and Implementation", *Decision Support Systems*, 24, pp. 17-27.

Wurman, P. R., M. P. Wellman e W. E. Walsh (1998b), "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents", in *Proceedings of the Second International Conference on Autonomous Agents (Agents'98)*, K. P. Sycara e M. Wooldridge (editores), pp. 301-308, ACM Press.

Yokoo, M., E. H. Durfee, T. Ishida e K. Kuwabara (1992), "Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving" in *Proceedings of the Twelfth International Conference on Distributed Computing Systems*, pp. 614-621.

Yokoo, M. e K. Hirayama (1998), "Distributed Constraint Satisfaction Algorithm for Complex Local Problems", in *Proceedings, Third International Conference on Multi-Agent Systems (ICMAS-98)*, pp. 372-379, IEEE Computer Society.

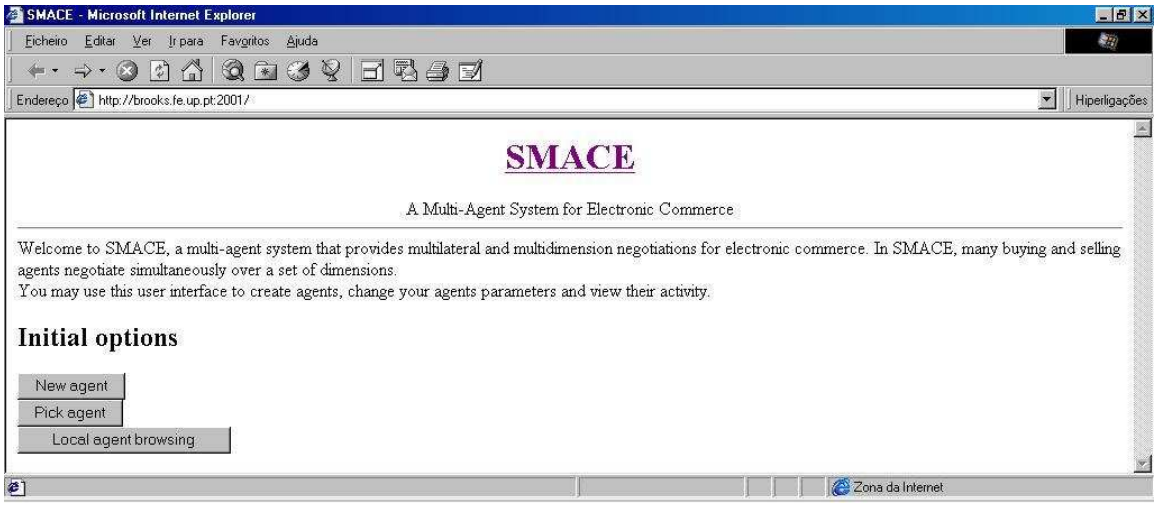
Zacharia, G., A. Moukas, R. Guttman e P. Maes (1998), "An agent system for comparative shopping at the point of sale", in *Proceedings of the European Conference on MM & E-Commerce*.

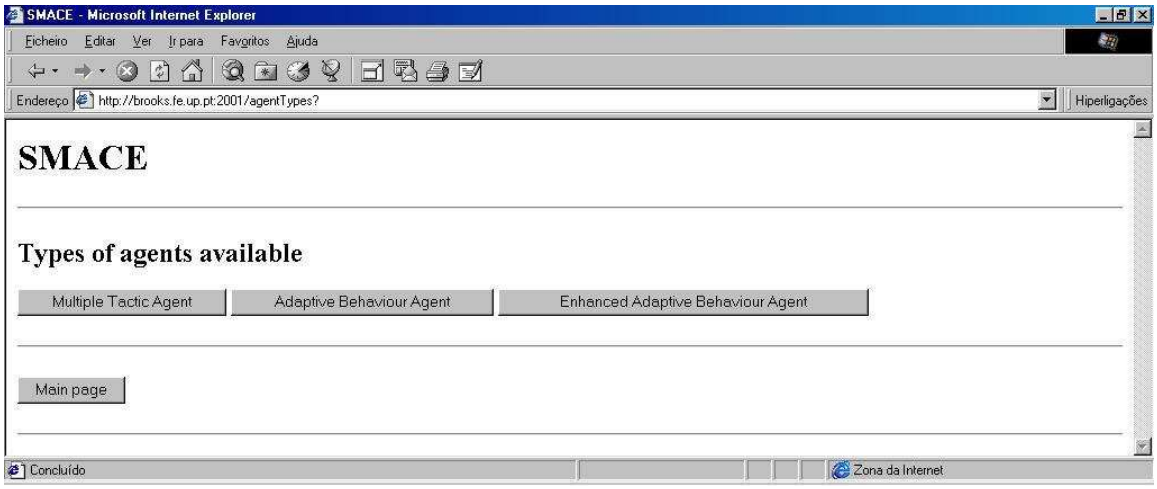
Zeng, D. e K. Sycara (1996), "How Can an Agent Learn to Negotiate?", in *Intelligent Agents III*, J. P. Muller et al. (editores), pp. 233-244, Springer-Verlag.

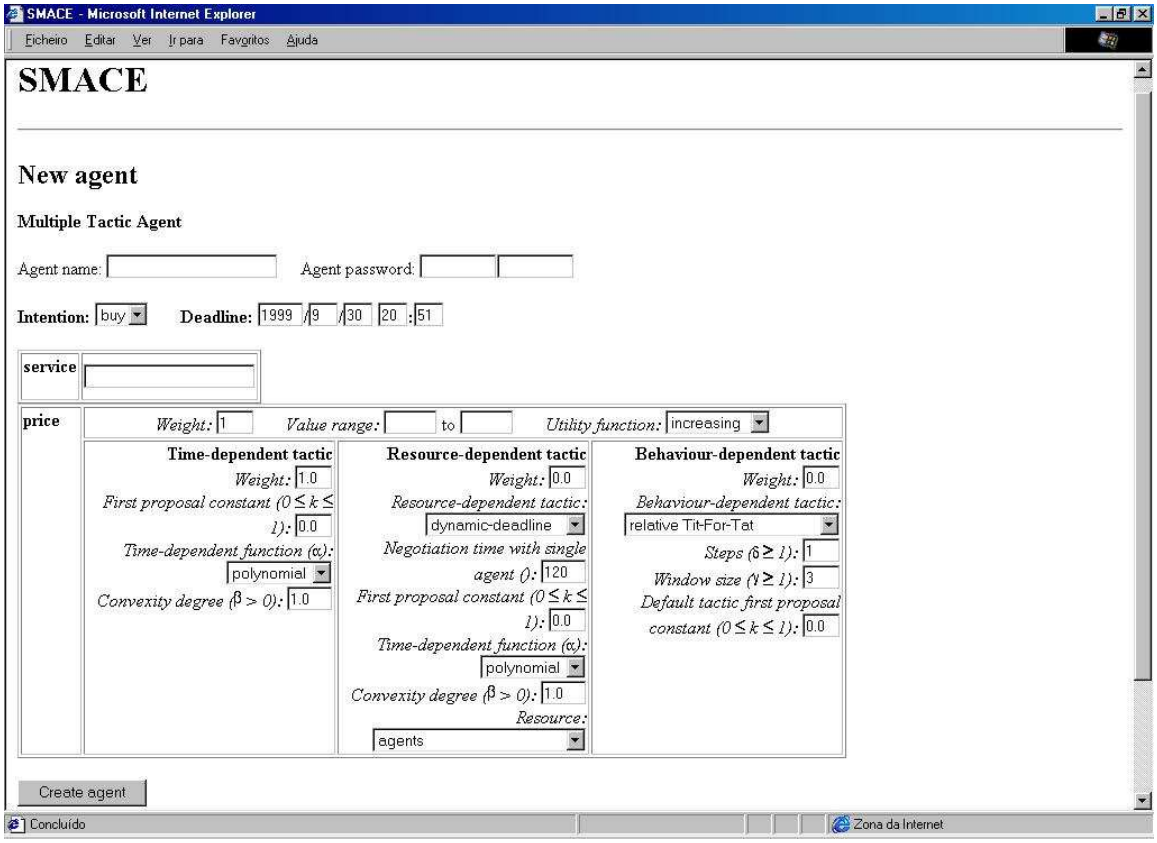
Zeng, D. e K. Sycara (1998), "Benefits of Learning in Negotiation", *International Journal of Human Computer Systems*, Vol. 48, pp. 125-141.

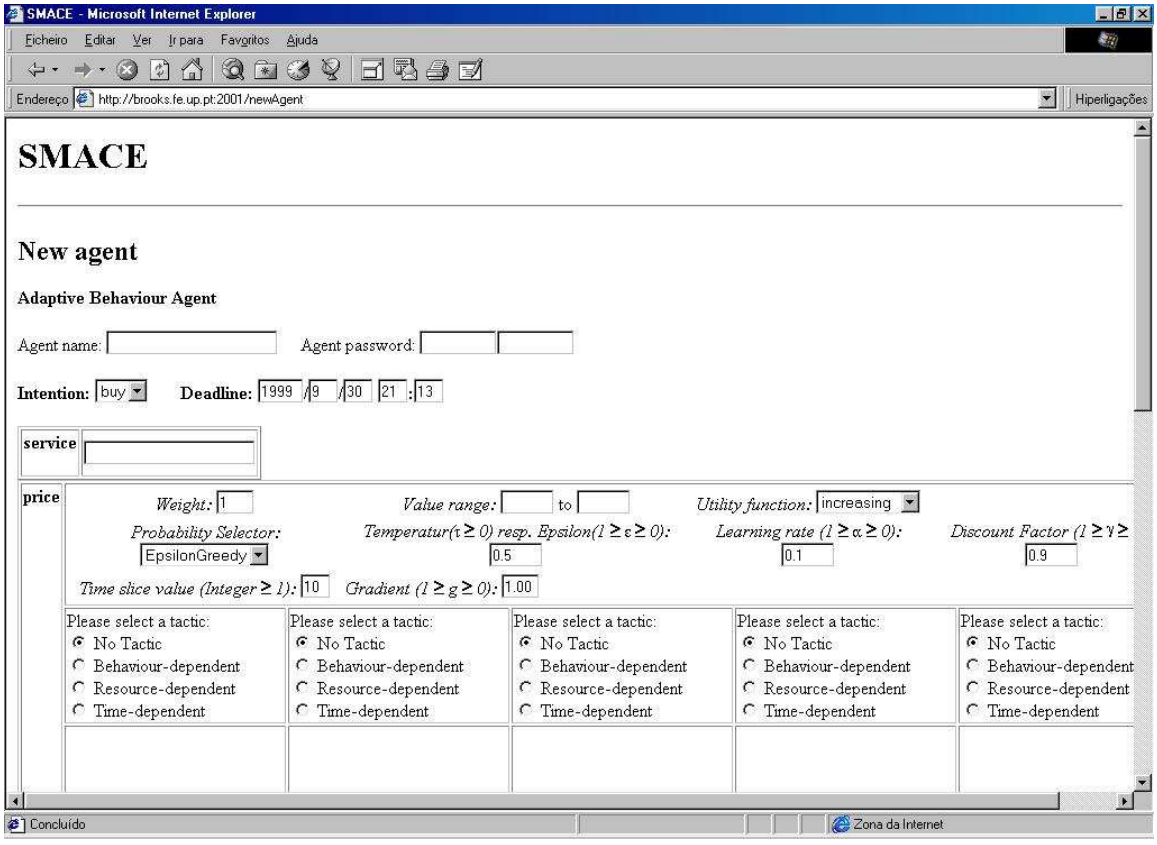
ANEXOS

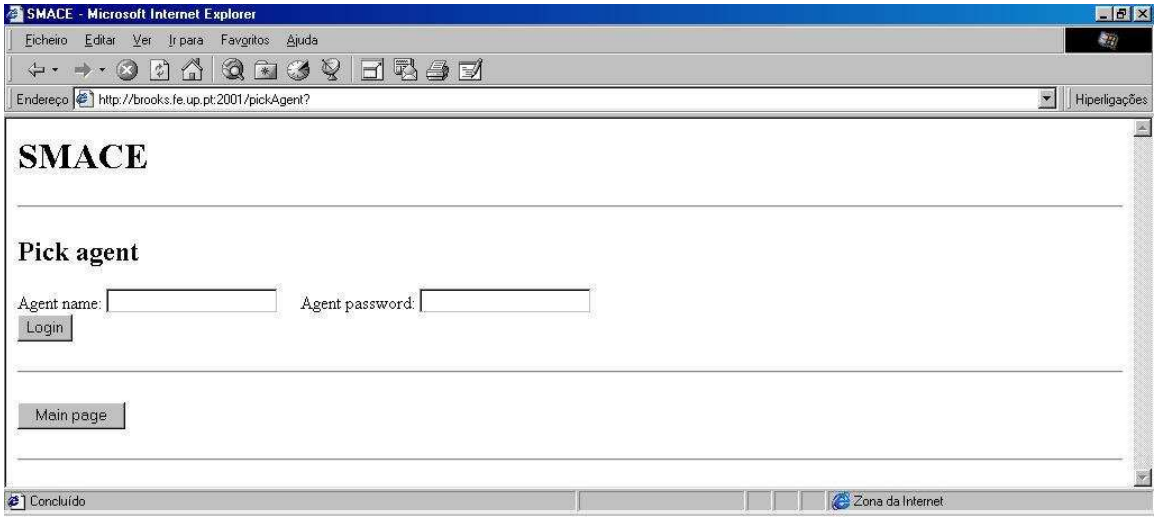
A – Ecrãs do sistema SMACE

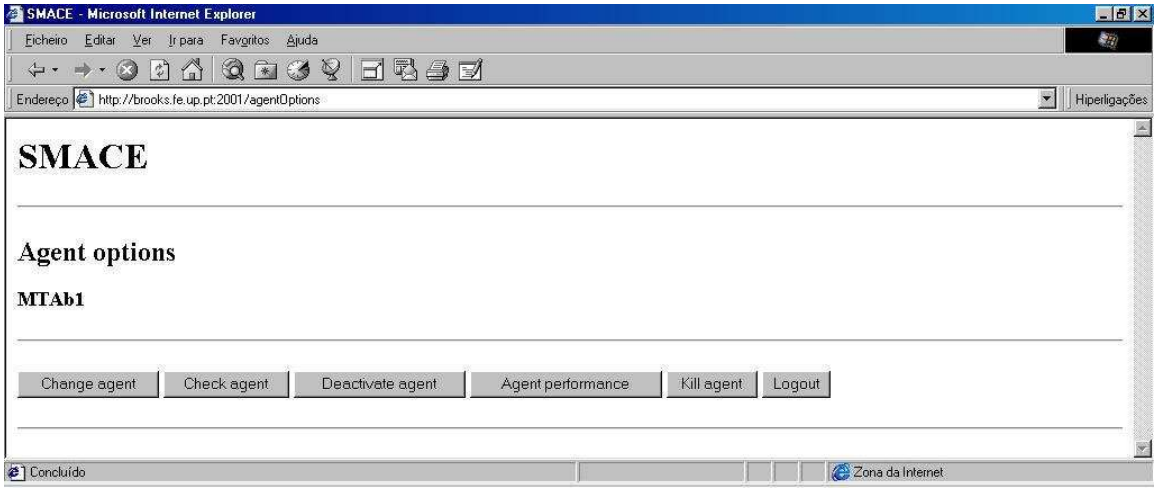


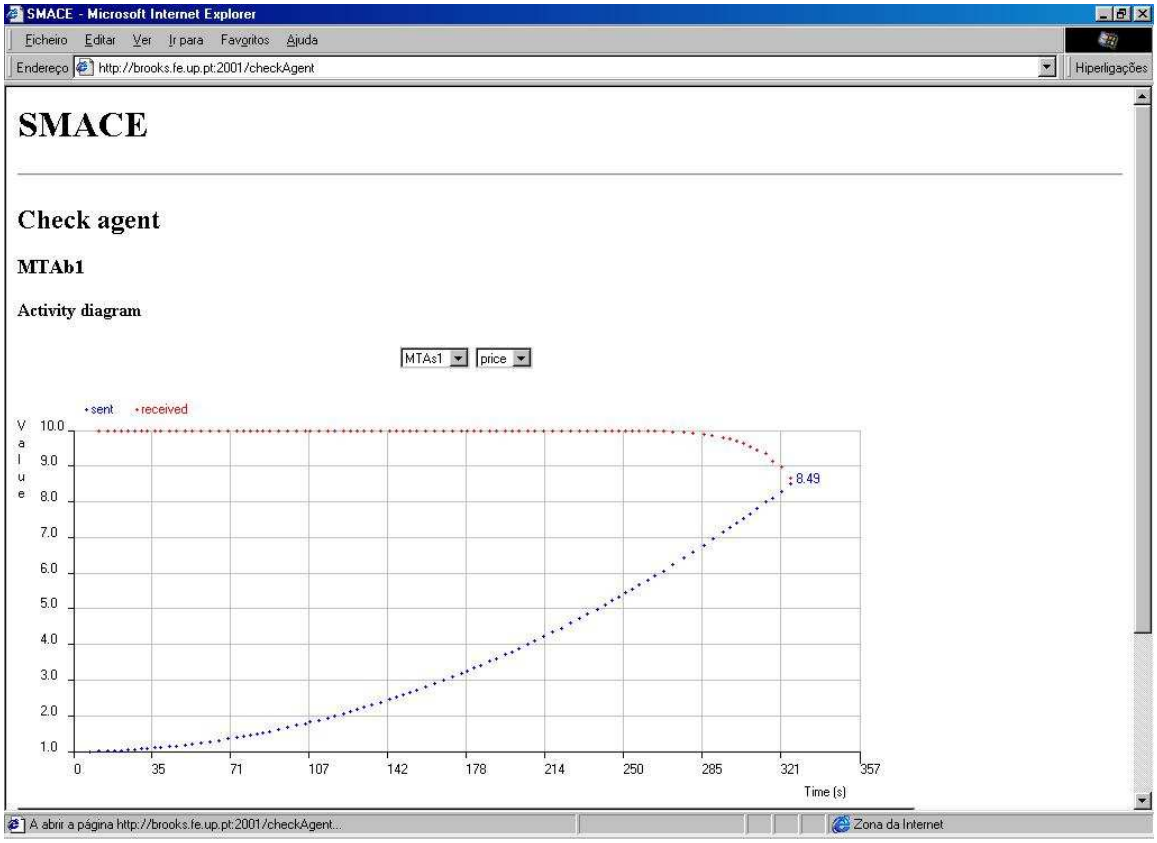


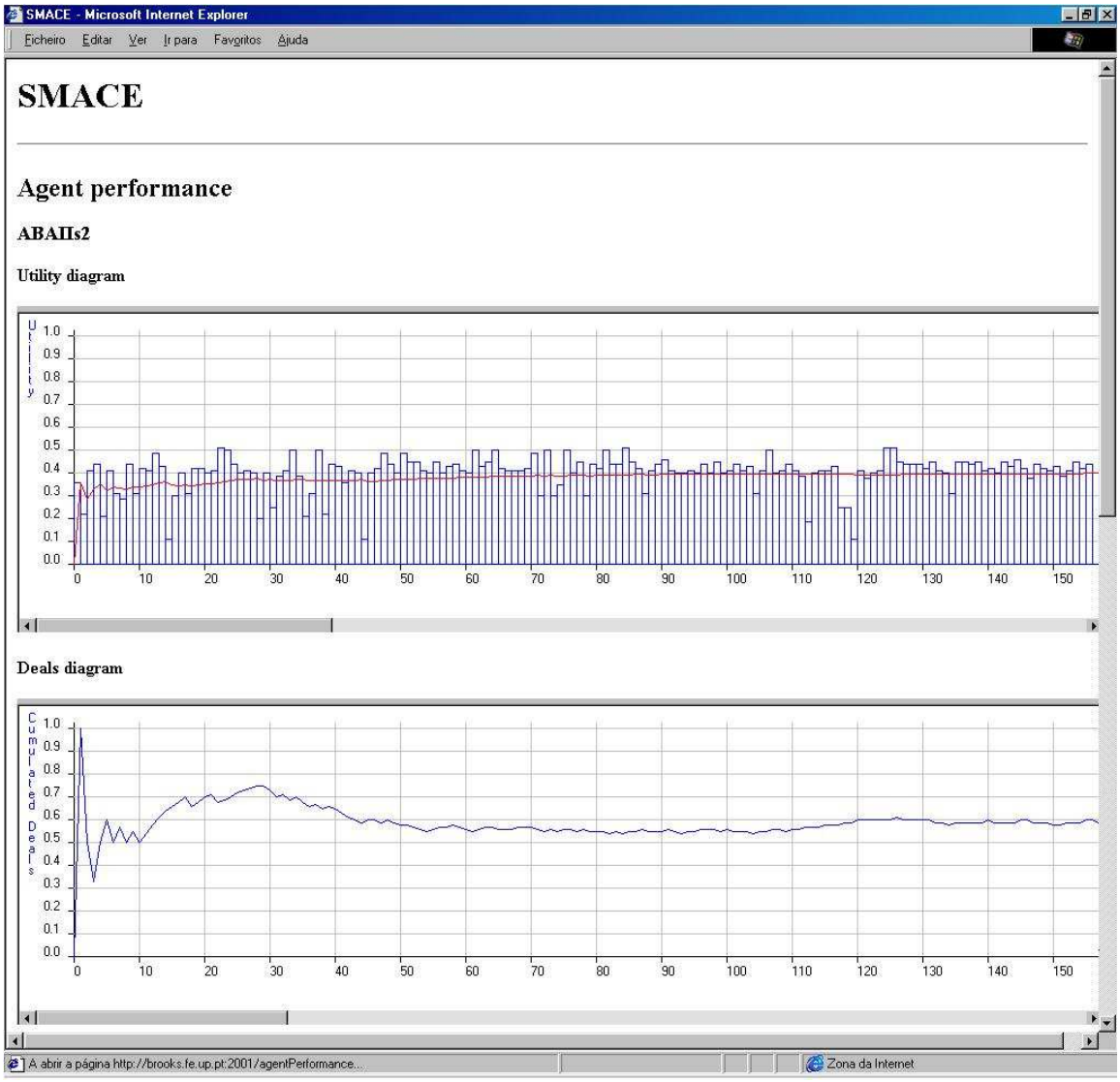

SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
SMACE
New agent
Multiple Tactic Agent
Agent name:
Agent password:
Intention: buy
Deadline: 1999 /9 /30 20 :51
service
price
Weight: 1
Value range: to
Utility function: increasing
Time-dependent tactic
Weight: 1.0
First proposal constant (0 ≤ k ≤ 1): 0.0
Time-dependent function (α): polynomial
Convexity degree (β > 0): 1.0
Resource-dependent tactic
Weight: 0.0
Resource-dependent tactic: dynamic-deadline
Negotiation time with single agent (): 120
First proposal constant (0 ≤ k ≤ 1): 0.0
Time-dependent function (α): polynomial
Convexity degree (β > 0): 1.0
Resource: agents
Behaviour-dependent tactic
Weight: 0.0
Behaviour-dependent tactic: relative Tit-For-Tat
Steps (δ ≥ 1): 1
Window size (γ ≥ 1): 3
Default tactic first proposal constant (0 ≤ k ≤ 1): 0.0
Create agent
Concluído
Zona da Internet


SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
Endereço http://brooks.fe.up.pt:2001/newAgent
Hiperligações
SMACE
New agent
Adaptive Behaviour Agent
Agent name:
Agent password:
Intention: buy
Deadline: 1999 /9 /30 21 :13
service
price
Weight: 1
Value range: to
Utility function: increasing
Probability Selector: EpsilonGreedy
Temperatur(τ ≥ 0) resp. Epsilon(1 ≥ ε ≥ 0): 0.5
Learning rate (1 ≥ α ≥ 0): 0.1
Discount Factor (1 ≥ γ ≥
0.9
Time slice value (Integer ≥ 1): 10
Gradient (1 ≥ g ≥ 0): 1.00
Please select a tactic:
No Tactic
Behaviour-dependent
Resource-dependent
Time-dependent
Concluído
Zona da Internet


SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
Endereço http://brooks.fe.up.pt:2001/pickAgent?
Hiperligações
SMACE
Pick agent
Agent name:
Agent password:
Login
Main page
Concluído
Zona da Internet


SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
Endereço http://brooks.fe.up.pt:2001/agentOptions
Hiperligações
SMACE
Agent options
MTAb1
Change agent
Check agent
Deactivate agent
Agent performance
Kill agent
Logout
Concluído
Zona da Internet


SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
Endereço http://brooks.fe.up.pt:2001/checkAgent
Hiperligações
SMACE
Check agent
MTAb1
Activity diagram
MTAs1
price
sent
received
Value
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
8.49
0
35
71
107
142
178
214
250
285
321
357
Time (s)
A abrir a página http://brooks.fe.up.pt:2001/checkAgent...
Zona da Internet


SMACE - Microsoft Internet Explorer
Ficheiro Editar Ver Ir para Favoritos Ajuda
SMACE
Agent performance
ABAIIs2
Utility diagram
Utility
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
Deals diagram
Cumulated Deals
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
A abrir a página http://brooks.fe.up.pt:2001/agentPerformance...
Zona da Internet