



José Alberto Rodrigues Pereira Sardinha

**MAS-School e ASYNC: Um Método e um Framework para
Construção de Agentes Inteligentes**

Tese de Doutorado

Tese apresentada como requisito parcial para
obtenção do título de Doutor pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientadores: Ruy Luiz Milidiú
Carlos José Pereira de Lucena

Rio de Janeiro, 18 de março de 2005

José Alberto Rodrigues Pereira Sardinha

**MAS-School e ASYNC: Um Método e um Framework para
Construção de Agentes Inteligentes**

Tese apresentada como requisito parcial para obtenção do título de Doutor pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Ruy Luiz Milidiú

Orientador

Departamento de Informática - PUC-Rio

Prof. Carlos José Pereira de Lucena

Co-Orientador

Departamento de Informática - PUC-Rio

Prof. Jaime Simão Sichman

Depto. Eng. Computação e Sistemas Digitais, Escola Politécnica - USP

Prof. Ricardo Choren Noya

Depto. Engenharia de Computação - IME

Prof. Bruno Feijó

Departamento de Informática - PUC-Rio

Prof. Hugo Fuks

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 18 de março de 2005

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

José Alberto Rodrigues Pereira Sardinha

Ficha Catalográfica

Sardinha, José Alberto Rodrigues Pereira

MAS-School e ASYNC: um método e um framework para construção de agentes inteligentes / José Alberto Rodrigues Pereira Sardinha; orientadores: Ruy Luiz Milidiú, Carlos José Pereira de Lucena. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2005.

152 f. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas

1. Informática – Teses. 2. Sistemas multi-agente. 3. Inteligência artificial. 4. Machine learning. 5. Engenharia de software. I. Milidiú, Ruy Luiz. II. Lucena, Carlos José Pereira de. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

À Luciana e ao meu Pai

Agradecimentos

Ao amigo e professor Ruy Milidiú, por toda a confiança, por todo o apoio, por todas as oportunidades, e pelos ensinamentos.

Ao amigo e professor Carlos Lucena, por toda a confiança, por todo o apoio e pela oportunidade de me incluir em um grupo de pesquisa tão extraordinário.

Aos alunos Patrick Paranhos e Pedro Cunha, futuros engenheiros e grandes pesquisadores. Essa tese tem muitas contribuições de vocês. Agradeço a vocês pelo empenho, dedicação, e trabalho.

Ao aluno Marco Molinaro, recém chegado ao nosso grupo de pesquisa, mas que já se mostra também um grande pesquisador. Obrigado pelas contribuições.

Aos meus grandes amigos do LES e LEARN: Alessandro, Viviane Silva, Choren, Guga, Daflon, Matheus, Fred Liporace, Fred, Laber, Arthur, Raul, Viviane Braconi, Lorenza, Taciana, Luiz Fernando, Patrick, Pedro, Marco, Helena, João, Chicão, Otávio, Daniel, e todos os outros que porventura eu esteja esquecendo. Obrigado pelas contribuições e apoio.

À Vera Menezes e Lena Ururahy, pelo apoio e carinho.

À minha família e principalmente ao meu pai, professor José Carlos Sardinha, e à sua mulher, professora Regina Lucia Sardinha, por estarem sempre ao meu lado nos momentos mais difíceis, sempre me incentivando.

À Luciana Salles, minha esposa, por todo amor, carinho, apoio e presença durante esses anos. Você esteve sempre do meu lado me incentivando.

Ao CNPq, pelo apoio financeiro.

Resumo

Sardinha, José Alberto Rodrigues Pereira. **MAS-School e ASYNC: Um Método e um Framework para Construção de Agentes Inteligentes**. Rio de Janeiro, 2005. 152p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Agentes de Software é uma tecnologia que permite criar simuladores e sistemas inteligentes que tomam decisões automaticamente. A primeira contribuição dessa tese é o MAS-School, um método para modelar e implementar agentes de software inteligentes desde as primeiras fases de desenvolvimento. Esse método também apresenta várias orientações de como incluir aprendizado na fase de *design* e implementação. O método apresenta no final uma estratégia incremental de desenvolvimento para permitir a avaliação do desempenho das técnicas de *machine learning*. A segunda contribuição dessa tese é o *framework* ASYNC. O ASYNC é composto por um conjunto de ferramentas de engenharia de software para auxiliar a construção de sistemas baseados em agentes assíncronos, cooperativos e inteligentes. Esta tese apresenta quatro estudos de casos complexos desenvolvidos com agentes inteligentes para exemplificar o uso do método e *framework*. A primeira aplicação apresenta um sistema baseado em agentes para criar promoções em um mercado varejista utilizando o conceito de agregação de produtos. A segunda aplicação apresenta um mercado virtual para compra e venda de bens. A terceira aplicação é um sistema multi-agente distribuído para um complexo cenário de *procurement* em leilões simultâneos e interdependentes. Essa aplicação participou do Trading Agent Competition em 2004 e obteve a terceira colocação. A quarta aplicação é um sistema multi-agente para um *Supply Chain Management*.

Palavras-chave

Sistemas Multi-Agente, Inteligência Artificial, *Machine Learning*, Engenharia de Software.

Abstract

Sardinha, José Alberto Rodrigues Pereira. **MAS-School e ASYNC: A Method and a Framework for Building Intelligent Agents**. Rio de Janeiro, 2005. 152p. Doctoral Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The agent technology is used to develop systems that perform several complex tasks. This thesis presents the MAS-School method for modeling and implementing intelligent agent-based systems. The method presents a systematic approach to support a disciplined introduction of machine learning techniques in multi-agent systems from an early stage of design. The proposed approach encompasses guidelines to both the design and implementation phases of an agent-based system. It is based on an incremental development strategy that largely relies on simulation and testing techniques. This thesis also presents the ASYNC framework that is composed of software engineering tools for building agent based system for asynchronous, cooperative and intelligent agents. This thesis presents four complex applications that used the proposed method and framework in the design and implementation phase. The first case study presents an application that discovers the most appealing offerings for consumers in a retail market. The second case study presents a virtual marketplace for buying and selling goods with automatic negotiation. The third case study is a multi-agent system for a complex procurement scenario with interdependent and simultaneous auctions. This system achieved the third place in the 2004 TAC Classic competition. The fourth case study is a multi-agent system for a PC manufacturer scenario based on sourcing of components, manufacturing of PC's and sales to customers.

Keywords

Multi-Agent Systems, Artificial Intelligence, Machine Learning, Software Engineering.

Sumário

1 Introdução	15
1.1. Sistemas Multi-Agente	16
1.2. A Engenharia de Software para Agentes Inteligentes	17
1.3. Aprendizado para Agentes Inteligentes	18
1.4. A Solução Proposta	20
1.5. Avaliação Empírica	25
1.6. Organização da Tese	26
2 A Construção de Sistemas com Agentes Inteligentes	28
2.1. ANote – Uma linguagem de Modelagem Simples e Poderosa	28
2.2. Frameworks Orientados a Objetos	32
2.3. Padrões de Projeto (Design Patterns)	33
2.4. Algoritmos para Agentes Inteligentes	35
2.4.1. Inteligência Artificial	35
2.4.2. Machine Learning	36
2.4.3. Predição em Séries Temporais	40
2.4.4. Otimização Combinatória	41
2.4.5. Análise de Agrupamento	42
3 ASYNC – Um Framework para Construção de Agentes Inteligentes	45
3.1. Estudo de Caso - A Aplicação Bundles.com	45
3.1.1. Simulação e Monitoramento	51
3.1.2. Resultados da Evolução do Sistema	52
3.2. ASYNC – Um Framework Orientado a Objetos para Construção de Agentes Distribuídos e Assíncronos	53
3.2.1. Como Instanciar uma Aplicação Baseada em Agentes com o Framework Orientado a Objetos do ASYNC	56
3.3. Instanciando o ASYNC para a Aplicação Bundles.com	58
3.4. ASYNC-D – A Ferramenta de Distribuição dos Agentes em CPUs	

Distintas de um Rede	60
3.5. Aplicação Babilônia	62
4 MAS-School - Um Método para Incluir Aprendizado em Sistemas Multi-Agente	64
4.1. Incluindo Aprendizado em Sistemas Multi-Agente	65
4.1.1. Objetivo Sistêmico & Seleção da Medida de Performance	66
4.1.2. Seleção do Agente & Definição do Objetivo do Aprendizado no Agente	68
4.1.3. Design do Aprendizado no Agente	68
4.1.4. Implementação Incremental & Avaliação de Performance	70
4.2. Agent Learning Pattern – Um Design para Incluir Aprendizado em Agentes	73
4.2.1. Intenção	73
4.2.2. Contexto	73
4.2.3. Problema	74
4.2.4. Forças	74
4.2.5. Solução	74
4.2.6. Estrutura	76
4.2.7. Exemplo	77
4.2.8. Dinâmica	77
4.2.9. Implementação	78
4.2.10. Conseqüências	81
4.2.11. Usos Conhecidos	81
4.2.12. Padrões Relacionados	82
5 A Aplicação LearnAgents	84
5.1. O TAC Classic	85
5.2. Edições Anteriores do TAC	85
5.3. A arquitetura do LearnAgents	88
5.4. O TAC <i>Classic</i> 2004 e Resultados	96
5.5. Os Algoritmos utilizados nos Agentes do LearnAgents	98
5.5.1. Preditor	98
5.5.2. Alocador	99

5.5.3. Ordenador	102
5.5.4. Negociador de Hotel	103
5.6. Utilizando o MAS-School no Processo de Desenvolvimento do LearnAgents	108
5.7. Utilizando o ASYNC na Implementação do LearnAgents	114
 6 A Aplicação LearnAgentsSCM	 118
6.1. O TAC Supply Chain Management	119
6.2. Evolução do TAC SCM	120
6.3. Arquitetura LearnAgentsSCM	122
6.4. A Engenharia de Algoritmos dos Agentes do LearnAgentsSCM	132
6.4.1. Gerente de Compras	132
6.4.2. Programador da Produção e Programador da Entrega	133
6.4.3. Gerente de Marketing	135
6.5. Utilizando o MAS-School no Processo de Desenvolvimento do LearnAgentsSCM	136
 7 Conclusões	 140
 8 Referências	 146

Lista de figuras

Figura 1: A decomposição dos objetivos relacionados com o objetivo do Problema Principal.	20
Figura 2: A arquitetura do Sistema Baseado em Agentes.	21
Figura 3: Cenário do Agente I.	22
Figura 4: Diagrama de Interação do Agente I e Agente II.	22
Figura 5: O diagrama de classes do Agente I.	23
Figura 6: A arquitetura de implementação do Sistema Multi-Agente.	24
Figura 7: A visão de objetivo do ANote.	29
Figura 8: A visão do agente do ANote.	29
Figura 9: A visão do cenário do ANote.	30
Figura 10: A visão de planejamento do ANote.	30
Figura 11: A visão de interação do ANote.	31
Figura 12: A visão de organização do ANote.	32
Figura 13: A árvore Minimax.	36
Figura 14: O Neurônio Artificial.	38
Figura 15: A decomposição dos objetivos.	46
Figura 16: apresenta a arquitetura do Bundles.com.	47
Figura 17: Diagrama de Cenários do Analista de Mercado.	47
Figura 18: Diagrama de Cenários do Comprador.	49
Figura 19: Os máximos locais e globais da função de fitness.	50
Figura 20: Diagrama de Cenários do Comprador.	50
Figura 21: Interface da Monitoração On-line.	52
Figura 22: Energia obtida pelos Agentes por instante de tempo.	53
Figura 23: O diagrama de Classes do Framework Orientado a Objetos ASYNC.	54
Figura 24: A instanciação do Agente Vendedor na aplicação Bundles.com.	58
Figura 25: Diagrama de cenário do Vendedor.	59
Figura 26: Os agentes do ASYNC-D.	61
Figura 27: A Console de Configuração do ASYNC-D.	62
Figura 28: As classes do Agente Comprador.	63
Figura 29: O método para incluir Aprendizado em um Sistema Multi-Agente.	66

Figura 30: A decomposição dos objetivos relacionados com o comércio de bens.	67
Figura 31: O diagrama de Classes (UML) do Preditor de Preços.	69
Figura 32: O processo incremental de desenvolvimento.	72
Figura 33: Um diagrama de Classes de um típico Agente.	75
Figura 34: O diagrama de Classes do <i>Agent Learning Pattern</i> .	76
Figura 35: O diagrama de Seqüência do <i>Agent Learning Pattern</i> .	78
Figura 36: O diagrama de Classes do agente Preditor.	78
Figura 37: Ilustração do TAC Classic.	85
Figura 38: A arquitetura do LearnAgents.	89
Figura 39: A decomposição dos objetivos relacionados com o comércio de bens.	90
Figura 40: Arquitetura do LearnAgents.	91
Figura 41: Diagrama de cenário dos Sensores.	92
Figura 42: Diagrama de cenário dos Preditores.	92
Figura 43: Diagrama de cenário dos Alocadores.	93
Figura 44: Diagrama de cenário do Ordenador.	94
Figura 45: Diagrama de cenário dos Negociadores.	94
Figura 46: Interface do Monitor.	95
Figura 47: Diagrama do Interação do LearnAgents.	95
Figura 48: Resultado do TAC de 2004.	97
Figura 49: Resultado do TAC de 2004.	99
Figura 50: A árvore Minimax.	104
Figura 51: A árvore Minimax (nível Maximizar) no Negociador de Hotel.	105
Figura 52: A árvore Minimax (nível Minimizar).	105
Figura 53: Gráfico dos lances enviados para leilão pelo Negociador de Hotel.	108
Figura 54: O diagrama de Classes (UML) do Preditor de Preços.	109
Figura 55: O Gráfico da Evolução do Benchmark do LearnAgents.	113
Figura 56: A instanciação do Agente Ordenador e Formato da Mensagem na aplicação. LearnAgents.	114
Figura 57: Diagrama de cenário do Ordenador.	115
Figura 58: O Jogo TAC SCM.	119
Figura 59: A Arquitetura do LearnAgentsSCM.	123

Figura 60: A decomposição dos objetivos relacionados com a produção, compra de matéria prima, e comercialização de bens.	124
Figura 61: O Diagrama de Agentes do LearnAgentsSCM.	125
Figura 62: Interface do Monitor.	126
Figura 63: Cenário da Seleção dos Pedidos por Cotações.	127
Figura 64: Cenário da Decisão dos Lances.	127
Figura 65: Cenário da Programação da Produção.	128
Figura 66: Cenário da Programação da Entrega e Ordens a Cancelar.	128
Figura 67: Cenário da Decisão de Compra dos Componentes.	129
Figura 68: Cenário da Negociação dos Componentes com os Fornecedores.	129
Figura 69: Diagrama de Interação do processo de vendas.	130
Figura 70: Diagrama de Interação do Processo de Compra de Matéria Prima.	130
Figura 71: Diagrama de Interação do Processo de Produção.	131
Figura 72: Diagrama de Interação do Processo de Entrega.	131
Figura 73: Nível de Estoque por tempo.	132
Figura 74: Decisão de Compra dos Componentes.	132
Figura 75: Gráfico da Evolução do Benchmark.	139

Lista de tabelas

Tabela 1: A mapeamento entre subproblemas e tipos de agentes.	21
Tabela 2: Mapeamento dos Conceitos de Agentes para as Classes OO.	22
Tabela 3: A mapeamento entre subproblemas e tipos de agentes.	46
Tabela 4: Exemplos de centróides encontrados com o algoritmo Isodata.	48
Tabela 5: Tabela com os padrões de consumo dos nove centróides.	48
Tabela 6: Mapeamento dos Conceitos de Agentes para as Classes da Aplicação instanciada.	57
Tabela 7: A mapeamento entre subproblemas e tipos de agentes.	68
Tabela 8: Resultado do TAC de 2000.	86
Tabela 9: Resultado do TAC de 2001.	87
Tabela 10: Resultado do TAC de 2002.	88
Tabela 11: Resultado do TAC de 2003.	88
Tabela 12: A mapeamento entre subproblemas e tipos de agentes.	91
Tabela 13: Resultado do TAC de 2004.	97
Tabela 14: Performance do Negociador de Hotel nas finais do TAC Classic de 2004.	107
Tabela 15: Benchmark das Versões do LearnAgents.	111
Tabela 16: Resultado do TAC de 2004.	113
Tabela 17: Resultado da Final do TAC SCM de 2003.	120
Tabela 18: Resultado da Final do TAC SCM de 2004.	121
Tabela 19: A mapeamento entre subproblemas e tipos de agentes.	125
Tabela 20: Resultados da Evolução do Benchmark.	138