

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Aluizio Haendchen Filho

**Um *Framework* do tipo *Middleware* para
Sistemas Multi-Agentes na Internet**

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: Carlos José Pereira de Lucena
Viviane Torres da Silva

Rio de Janeiro, 13 de setembro de 2005



Aluizio Haendchen Filho

**Um *Framework* do tipo *Middleware* para
Sistemas Multi-Agentes na Internet**

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.

Carlos José Pereira de Lucena

Orientador

Departamento de Informática – PUC-Rio

Viviane Torres da Silva

co-orientadora

Departamento de Informática – PUC-Rio

Markus Endler

Departamento de Informática – PUC-Rio

Renato Fontoura de Gusmão Cerqueira

Departamento de Informática – PUC-Rio

Claudia Maria Lima Werner

Departamento de Informática – COPPE - UFRJ

Rosa Maria Vicari

Departamento de Informática - UFRGS

prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 13 de setembro de 2005

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

Aluizio Haendchen Filho

Graduou-se em Administração de Empresas pela UNISINOS. Obteve o título de mestre em Ciência da Computação pelo CPGCC-UFRGS, no ano de 2000.

Ficha Catalográfica

Haendchen Filho, Aluizio

Um *framework* do tipo *middleware* para sistemas multi-agentes na internet / Aluizio Haendchen Filho ; orientador: Carlos José Pereira de Lucena ; co-orientadora: Viviane Torres da Silva. – Rio de Janeiro : PUC, Departamento de Informática, 2005.

175 f. ; 30 cm

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

Incluí referências bibliográficas.

1. Informática – Teses. 2. *Framework de Middleware*. 3. Sistemas multi-agentes. 4. Arquitetura orientada a serviços. . I. Lucena, Carlos José Pereira de. II. Silva, Viviane Torres da. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais, Alúzio (em memória) e Clélia.

Agradecimentos

Gostaria de agradecer aos meus orientadores Professor Lucena e Viviane, pelo estímulo e parceria para a realização deste trabalho.

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Gostaria também de agradecer, com todo o carinho, a Miriam Sayão, em quem sempre encontrei o necessário incentivo e apoio nos momentos mais difíceis. Não poderia também deixar de agradecer a Silvia Sayão, por sua serenidade e discernimento. Aos amigos Hércules Prado, por sua participação segura em todos os momentos, e Clovis Peres, que por seu exercício de contextualização facilitou uma exata compreensão da dimensão dessa pesquisa científica.

Aos meus irmãos Roberto e Maria Isabel, que sempre me incentivaram.

Aos meus alunos da UniverCidade Rafael Castaneda e Nuno Caminada, que tiveram participação fundamental no trabalho. Aos alunos Victor e Marcelo Cruz, pelo apoio que prestaram ao longo do projeto. Gostaria ainda de agradecer ao Jorge de Abreu Soares e Claudia Ferlin, da UniverCidade, pelo incentivo e apoio operacional.

Resumo

Haendchen Filho, Aluízio. **Um Framework do tipo Middleware para Sistemas Multi-Agentes na Internet**. Rio de Janeiro, 2005. 175 p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A crescente expansão e evolução da Internet têm impulsionado a demanda por aplicações baseadas em agentes e a busca por padrões e ferramentas para dar suporte ao desenvolvimento destas aplicações. MIDAS (Middleware for Intelligent and Distributed Agent-based Systems) define uma arquitetura flexível e adaptativa, fracamente acoplada, orientada a serviços que provê uma plataforma de execução e um *framework* para facilitar o desenvolvimento de sistemas multi-agentes na Internet. A arquitetura do *framework* adere e estende as especificações da arquitetura de referência WSA (Web Services Architecture), sendo formada por duas estruturas básicas: uma concreta e outra abstrata. A estrutura concreta é composta por um conjunto de agentes intermediários dinâmicos e pró-ativos, que provêm serviços de infraestrutura. Estes serviços possibilitam que os desenvolvedores possam abstrair funcionalidades complexas, tais como comunicação, concorrência, gerenciamento e interoperabilidade. A segunda estrutura é composta por classes abstratas que representam os agentes e componentes de aplicações. As classes abstratas definem os *hot-spots* a partir dos quais as particularidades específicas das aplicações podem ser implementadas e regulam o fluxo de interação entre a implementação concreta dos agentes e a arquitetura. A abordagem estende os atuais padrões definidos pela arquitetura de referência WSA introduzindo o conceito de agente abstrato, e mecanismos complementares para dar suporte ao *workflow* e modelo de comunicação dos agentes.

Palavras-chave

Framework de Middleware, Sistemas Multi-Agentes, Arquitetura Orientada a Serviços.

Abstract

Haendchen Filho, Aluizio. **A Middleware Framework for Multi-Agent Systems in the Internet**. Rio de Janeiro, 2005. 175 p. DSc Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The increasing widespread and evolution of the Internet have stimulated the demand for agent-based applications, and the search for standards and tools to support the development of these applications. MIDAS (Middleware for Intelligent and Distributed Agent-based Systems) defines a flexible, adaptable, and loosely coupled service-oriented architecture, that provides an agent platform and a framework to facilitate the development of multi-agent systems in the Internet. The framework architecture adheres and extends the specifications of the WSA (Web Services Architecture) reference model, being composed by two basic structures: a concrete and an abstract one. The concrete structure is composed by a set of dynamic and pro-actives agents, which provide infrastructure services. These services enable the developer to abstract complex functionalities, such as communication, concurrency, management and interoperability. The second structure is composed by abstract classes, which represent the agents and components of applications. The abstract classes define the hot-spots from which the specific particularities of the applications can be implemented, and regulate the interaction flow between the concrete implementation of the agents and the architecture. The proposed architecture extends the current concepts defined by the WSA reference model, introducing the concept of abstract agent and complementary mechanisms to provide support for the agent's workflow and communication model.

Keywords

Middleware Framework, Multi-Agent Systems, Service-Oriented Architecture

Sumário

1	Introdução	14
1.1.	Contexto	15
1.1.1	Sistemas Multi-Agentes na Internet.....	15
1.1.2	Arquiteturas Orientadas a Serviços e Serviços Web.....	17
1.2.	Motivação	19
1.2.1.	Dificuldades Inerentes ao Desenvolvimento.....	19
1.2.2.	Estruturas Fortemente Acopladas	20
1.2.3.	Integração Agentes - Serviços Web e Interoperabilidade.....	21
1.3.	Proposta	22
1.4.	Objetivos	24
1.4.1.	Framework de Middleware com Arquitetura Flexível.....	25
1.4.2.	Facilitar o Projeto Detalhado e Implementação.....	27
1.4.3.	Integração Agentes - Serviços Web e Interoperabilidade.....	28
1.5.	Contribuições da Tese	29
1.6.	Organização da Tese	32
2	Fundamentos	33
2.1	Tecnologia de Agentes.....	33
2.1.1	Agentes de Software	33
2.1.2	Perspectivas de Agentes.....	35
2.1.3	Classificação de Agentes	37
2.1.4	Modelos e Estruturas Organizacionais	38
2.1.5	Comunicação	41
2.2	Middleware	43
2.2.1	Definição de Middleware	43
2.2.2	Frameworks do tipo Middleware.....	45
2.3	Arquitetura de Referência WSA	47
2.3.1	Modelo Orientado a Mensagens	49
2.3.2	Modelo Orientado a Serviços	50

2.3.3 Modelo Orientado a Recursos	51
2.3.4 Modelo de Gerenciamento	52
2.4 Serviços Web e Interfaces Orientadas a Serviços	53
2.5 Discussão.....	56
3 A plataforma MIDAS.....	57
3.1 Conceitos Básicos	59
3.2 O modelo de Papéis	62
3.2.1 Os Papéis dos Agentes Intermediários	63
3.2.1.1 O Agente Broker.....	63
3.2.1.2 O Agente Proxy	66
3.2.1.3 O Agente Catalog.....	68
3.2.1.4 O Agente Manager	70
3.2.1.5 O Agente Blackboard	73
3.2.1.6 O Agente DBPool	74
3.2.2 Os Papéis dos Agentes de Aplicações.....	75
3.2.2.1 A Classe Abstrata Agent	75
3.2.2.2 A Classe Abstrata Component	77
3.3 O Modelo de Comunicação	78
3.4 O Modelo Estrutural	81
3.4.1 A Arquitetura do Agent Container.....	81
3.4.1.1 O Agente C-Broker	83
3.4.1.2 O Agente Proxy	84
3.4.1.3 O Agente C-Catalog	85
3.4.1.4 O Agente C-Manager	87
3.4.1.5 O Agent Model	94
3.4.2 A Arquitetura do Servidor de Front-End	95
3.4.3 A Arquitetura do Component Container.....	98
3.5 Discussão.....	100
4 Estudos de Caso	102
4.1 O Sistema Expert Committee	106
4.1.1 Metodologia.....	106
4.1.2 Especificação de Requisitos.....	110

4.1.3 Análise.....	112
4.1.4 Projeto Detalhado e Implementação	115
4.1.4.1 Os Agentes do EC.....	115
4.1.4.2 Os Componentes do EC.....	121
4.1.5 As Tarefas de Gerenciamento.....	123
4.2 O Protótipo Academic.Info.....	125
4.3 O Protótipo Academic.Net	128
4.4 O Protótipo Therapp.....	129
4.5 O Protótipo MobileInvest	132
4.6 Discussão.....	133
5 Trabalhos Relacionados.....	137
5.1 JADE – Java Agent DEvelopment.....	137
5.2 CASA – Collaborative Agent System Architecture	140
5.3 O <i>Middleware</i> AgentScape.....	142
5.4 SOMA – Secure and Open Mobile Agent.....	144
5.5 Comparação entre as Abordagens.....	146
5.6 Discussão.....	147
6 Conclusão	157
6.1 Contribuições da Tese	157
6.2 Restrições	162
6.3 Trabalhos Futuros	164
6.3.1 Mobilidade	164
6.3.2 Segurança	165
7 Referências bibliográficas	167

Lista de figuras

Figura 1 - Arquitetura genérica da plataforma	22
Figura 2 - Visão de um Sistema Multi-Agente	39
Figura 3 - Camadas representando o domínio	46
Figura 4 - Modelos de Referência de WSA	48
Figura 5 - Principais conceitos do MOM e os seus relacionamentos	49
Figura 6 - Principais conceitos do SOM e seus relacionamentos.....	51
Figura 7 - Principais conceitos do ROM e seus relacionamentos	52
Figura 8 - Principais conceitos do modelo de gerenciamento de políticas	53
Figura 9 - Integração entre serviços Web e serviços de negócios	55
Figura 10 - Estratégia para a definição dos componentes da arquitetura	58
Figura 11 - A plataforma MIDAS e os principais conceitos.....	60
Figura 12 - Responsabilidades e colaborações do agente C-Broker	64
Figura 13 - Responsabilidades e colaborações do agente S-Broker.....	65
Figura 14 - Responsabilidades e colaborações do agente Proxy.....	67
Figura 15 - Responsabilidades e colaborações do agente C-Catalog.....	68
Figura 16 - Responsabilidades e colaborações do agente S-Catalog.....	70
Figura 17 - Responsabilidades e colaborações do agente C-Manager.....	71
Figura 18 - Responsabilidades e colaborações do agente S-Manager	72
Figura 19 - Responsabilidades e colaborações do agente Blackboard.....	73
Figura 20 - Responsabilidades e colaborações do agente DBPool.....	74
Figura 21 - Responsabilidades e colaborações da classe abstrata Agent	76
Figura 22 - Responsabilidades e colaborações da classe abstrata Component	77
Figura 23 - Caminho da mensagem em uma requisição de serviço Web	79
Figura 24 - Caminho da mensagem de uma requisição intra-plataforma.....	80
Figura 25 - Comunicação intra-container.....	81
Figura 26 - Arquitetura do framework do Agent Container (AC).....	82
Figura 27 - A estrutura do agente C-Broker.....	83
Figura 28 - A estrutura do agente Proxy.....	84
Figura 29 - Estrutura do agente C-Catalog.....	86
Figura 30 - Estrutura do agente C-Manager	88
Figura 31 - Diagrama de seqüência para uma chamada local síncrona	90
Figura 32 - Diagrama de seqüência para uma chamada local assíncrona.....	91

Figura 33 - Assistente GUI para a perspectiva View	92
Figura 34 - Assistente GUI para a perspectiva Build	93
Figura 35 - Diagrama de classes do Agent Model.....	94
Figura 36 - Arquitetura do framework do servidor de front-end	96
Figura 37 - Diagrama de classes do agente S-Broker.....	97
Figura 38 - Arquitetura do framework do Component Container	99
Figura 39 - Relacionamentos entre os modelos	107
Figura 40 - Diagrama de atividades do sistema EC	114
Figura 41 - Método lifecycle do agente Chair	116
Figura 42 - Implementação de interface provide do agente Coordinator.....	117
Figura 43 - O agente Coordinator escreve uma mensagem no blackboard	118
Figura 44 - Implementação da interface boardChanged do agente Reviewer .	118
Figura 45 - Método operateSelection do agente Coordinator	119
Figura 46 - Invocação remota se serviço efetuada pelo agente Coordinator ...	120
Figura 47 - Método verifyAcceptedArticles do agente Chair.....	120
Figura 48 - Agente requisitando serviço de acesso a dados p/componente	121
Figura 49 - Componente ArticleData efetuando uma query no Hibernate	122
Figura 50 - Componente ArticleData executando getUnrevisedArticles.....	123
Figura 51 - Assistente GUI para a modalidade de gerenciamento global	124
Figura 52 - Componente representando um serviço Web em um Container AI125	
Figura 53 - Agente invocando serviço Web representado por componente.....	126
Figura 54 - Método provide do componente ResearcherData.....	127
Figura 55 - O protótipo Academic.Net	128
Figura 56 - Interface gráfica para acionada pelo agente Tutor.....	131
Figura 57 - Modelo de referência FIPA.....	138
Figura 58 - Arquitetura de software de uma plataforma de agentes JADE	139
Figura 59 - Os conceitos básicos na arquitetura CASA.....	141
Figura 60 - A arquitetura de framework AgentScape.....	143
Figura 61 - A arquitetura do framework SOMA.....	145
Figura 62 - O agente Chair cria uma instanciada interface de descrição	148
Figura 63 - Chair adiciona parâmetros à interface e despacha a requisição....	149
Figura 64 - Chair enviando mensagem ao Coordinator.....	149
Figura 65 - Intra-classe ReceberPedidoSecundario do agente Coordinator	150
Figura 66 - O Coordinator adiciona um novo comportamento ao seu leque	150
Figura 67 - O Coordinator no MIDAS implementa o método boardChanged ...	151
Figura 68 - O agente Coordinator se registra como ouvinte do Blackboard.....	151

Figura 69 - Classe da implementação JADE simulando o papel de um proxy .	152
Figura 70 - Servlet do agente controlador de acesso a GUI's em JADE	153
Figura 71 - Descrição de um agente MIDAS na especificação structure.xml...	153
Figura 72 - Descrição de um serviço MIDAS na especificação services.xml ...	154
Figura 73 - Requisição de um serviço no MIDAS	154
Figura 74 - O módulo de regras e segurança	166

Lista de tabelas

Tabela 1 - Tipos de agentes de <i>middleware</i> e as suas propriedades.....	61
Tabela 2 - Protótipos implementados e as suas características	103
Tabela 3 - <i>Template</i> para o papel Event.....	113
Tabela 4 - Os protótipos implementados e as propriedades exercitadas	134
Tabela 5 - Comparação entre características dos <i>frameworks</i>	146