



**Cidiane Aracaty Lobato**

**Um Framework Orientado a Aspectos  
para Mobilidade de Agentes de Software**

**Dissertação de Mestrado**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Informática.

Orientadores: Carlos José Pereira de Lucena  
Alessandro Fabricio Garcia

Rio de Janeiro  
Dezembro de 2005



**Cidiane Aracaty Lobato**

**Um Framework Orientado a Aspectos para  
Mobilidade de Agentes de Software**

Dissertação apresentada ao Programa de Pós-graduação em Informática da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Informática. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Carlos José Pereira de Lucena**

Orientador

Departamento de Informática – PUC-Rio

**Prof. Alessandro Fabricio Garcia**

Co-orientador

Computing Department – Lancaster University - UK

**Prof. Simone Diniz Junqueira Barbosa**

Departamento de Informática – PUC-Rio

**Dr. Raul Cesar Baptista Martins**

Departamento de Informática – PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro  
Dezembro de 2005

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

### **Cidiane Aracaty Lobato**

Graduou-se no Curso de Bacharelado em Ciência da Computação da Universidade Federal do Pará (UFPA) em 2003. É pesquisadora nas áreas de Engenharia de Software de Sistemas Multi-Agentes e Desenvolvimento de Software Orientado a Aspectos no Laboratório de Engenharia de Software (LES) da PUC-Rio.

#### Ficha Catalográfica

Lobato, Cidiane Aracaty

Um framework orientado a aspectos para mobilidade de agentes de software / Cidiane Aracaty Lobato ; orientadores: Carlos José Pereira de Lucena, Alessandro Fabrício Garcia. – Rio de Janeiro : PUC, Departamento de Informática, 2005.

157 f. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Sistemas Multi-Agentes. 3. Agentes Móveis. 4. Plataformas de Mobilidade. 5. Desenvolvimento de Software Orientado a Aspectos. 6. Arquitetura de Software. 7. Frameworks. I. Lucena, Carlos José Pereira de. II. Garcia, Alessandro Fabrício. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

*Ao Senhor, cujas misericórdias cantarei eternamente.*

## Agradecimentos

A Deus, por sua Providência e Misericórdia, as quais regem minha vida. Agradeço ao Senhor por ter-me formado incansavelmente nestes anos de Mestrado, ensinado-me a dar sentido a todas as coisas, em uma sempre mais crescente obra de maturidade e liberdade interior.

À Comunidade Católica Shalom, por resgatar em mim o sentido do absoluto, do eterno e do definitivo, através da vivência de valores como a oração, a fraternidade e o serviço ao próximo.

À minha família, que mesmo à distância, cuidou de estar sempre unida a mim nos momentos mais difíceis. Particularmente, agradeço aos meus pais, que são para mim um verdadeiro tesouro, uma jóia preciosa, uma oportunidade imperdível de crescimento na caridade.

Ao meu orientador Prof<sup>o</sup> Lucena e ao meu co-orientador Alessandro, agradeço todo o apoio recebido. Sinto-me grata pelo dom de ter convivido com pessoas tão gentis, experientes e profissionais. Agradeço a confiança, a paciência, o incentivo.

Às minhas amigas Cleide, Karla, Mariana e Morgana, que suportaram minhas ausências e impulsionaram-me à superação, à responsabilidade e ao compromisso. À Cleide, por sua confiança generosa, pelo referencial de vida. À Karla, por ter assumido o sacrifício envolvido na elaboração da dissertação, permanecendo ao meu lado nos momentos mais importantes. À Mariana, por sua alegria, espontaneidade e doçura. À Morgana, pela delicadeza e silêncio consoladores.

Aos colegas do Laboratório de Engenharia de Software, particularmente aos amigos Akeo, Carolina, Cláudio, Eduardo, Gustavo, Leonardo, Lyrene, Míriam, Roberta, Rodrigo, Uirá e Viviane, pelo ambiente de trabalho favorável.

À Fundação Padre Leonel Franca, à PUC-Rio e ao CNPq pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

## Resumo

Lobato, Cidiane. **Um Framework Orientado a Aspectos para Mobilidade de Agentes de Software**. Rio de Janeiro, 2005. 157p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Engenheiros de sistemas multi-agentes (SMAs) móveis devem lidar com o projeto e a implementação de mobilidade de código, além do projeto e implementação de funcionalidades básicas e outros requisitos. À medida que a complexidade dos SMAs aumenta, questões referentes à mobilidade não podem ser modularizadas somente a partir da utilização de abstrações e mecanismos orientados a objetos (OO). Além disso, vários programadores frequentemente evidenciam a presença de problemas de entrelaçamento e espalhamento em suas linhas de código referentes à mobilidade. Apesar de tais problemas, os desenvolvedores de SMAs têm se apoiado vastamente na utilização de interfaces de programação de aplicações (APIs) OO das plataformas de mobilidade e na linguagem de programação Java. O resultado é a produção de SMAs que são difíceis de entender, manter e reutilizar. Este trabalho apresenta três contribuições principais. Primeiramente, é realizada uma análise das soluções existentes para separação dos interesses de mobilidade. Além disso, são propostos a arquitetura de software ArchM e o *framework* AspectM, ambos baseados em aspectos, visando: (i) uma clara separação dos interesses de mobilidade em relação às funcionalidades básicas e outros requisitos, (ii) uma introdução transparente do código de mobilidade em agentes estacionários, e (iii) uma integração flexível dos SMAs com plataformas de mobilidade existentes. O desenvolvimento de software orientado a aspectos é um novo paradigma com abstrações e mecanismos de composição que possibilitam o aumento do grau de reusabilidade e manutenibilidade dos sistemas. Os benefícios de ArchM e AspectM são ilustrados através de dois estudos de caso e com o uso de duas plataformas de mobilidade.

## Palavras-chave

Sistemas Multi-Agentes; Agentes Móveis; Plataformas de Mobilidade; Desenvolvimento de Software Orientado a Aspectos; Arquitetura de Software; *Frameworks*.

## Abstract

Lobato, Cidiane. **An Aspect-Oriented Framework for Software Agents Mobility**. Rio de Janeiro, 2005. 157p. Master Thesis - Computer Science Department, Pontifical Catholic University of Rio de Janeiro.

Software engineers of multi-agent systems (MASs) are concerned with the design and implementation of the mobility issues in addition to the agents' basic functionalities and other agent-related concerns. As the agents' complexity increases, mobility issues cannot be modularized based only on object-oriented (OO) abstractions and mechanisms. Besides, some programmers frequently evidence the presence of mobility tangling and scattering problems on their systems. Despite these problems, MAS developers have mostly relied on application programming interfaces (APIs) OO from mobility platforms and on the use of the Java programming language. The result is the production of MASs that are difficult to understand, maintain, and reuse. This work presents three main contributions. Firstly, an analysis of existent solutions for modularization of mobility issues. Besides, the proposal of the ArchM software architecture and the AspectM framework, both based on aspects, which support: (i) improved modularization of the mobility issues, that is, a clean separation between the mobility-specific concerns and the other agent concerns, (ii) a seamless introduction of code mobility into stationary agents, and (iii) a flexible integration with multiple mobility platforms. The aspect-oriented software development (AOSD) is a new paradigm with abstractions and composition mechanisms that support an improvement of reuse and maintainability of systems. The benefits of the AspectM framework are illustrated through two case studies and through use of two mobility platforms.

## Keywords

Multi-Agent Systems; Mobile Agents; Mobility Platforms; Aspect-Oriented Software Development; Software Architecture; Frameworks.

# Sumário

1	Introdução	14
1.1.	Problema	14
1.2.	Limitações dos Trabalhos Relacionados	17
1.3.	Solução Proposta	19
1.4.	Objetivos	21
1.5.	Organização do Texto	22
2	Sistemas Multi-Agentes Móveis	23
2.1.	Definições Básicas de SMAs	23
2.2.	Definições de SMAs Móveis	26
2.3.	Agentes Móveis e Paradigmas da Computação em Rede	29
2.4.	Vantagens e Desvantagens de Agentes Móveis	33
2.5.	Áreas de Aplicação de SMAs Móveis	34
2.6.	Plataformas de Mobilidade	35
3	Separação dos Interesses de Mobilidade	42
3.1.	Descrição do Estudo de Caso	43
3.2.	APIs de Plataformas	44
3.3.	Padrões de Projeto	48
3.4.	Abordagem <i>RoleEP</i>	52
3.5.	Requisitos de SMAs Móveis	57
4	Desenvolvimento de Software Orientado a Aspectos	59
4.1.	Exemplo de Interesse Transversal: Tratamento de Erros	60
4.2.	Programação Orientada a Aspectos	62
4.3.	Projeto Orientado a Aspectos	67
4.4.	Arquitetura Orientada a Aspectos	69
5	A Arquitetura de Software ArchM	71
5.1.	Uma Arquitetura de Software para Mobilidade	72

5.2. Componentes de ArchM	73
5.3. Interfaces de ArchM	75
5.4. Análise de ArchM	77
6 O <i>Framework AspectM</i>	78
6.1. Funcionalidades Tratadas	78
6.2. Estrutura Geral do Projeto	81
6.3. Projeto Detalhado do Componente <i>MobilityProtocol</i>	90
6.4. Projeto Detalhado do Componente <i>MobilityManagement</i>	102
7 Estudos de Caso	112
7.1. <i>Expert Committee</i>	112
7.2. <i>MobiGrid Framework</i>	128
7.3. Análise da Arquitetura ArchM e do <i>Framework AspectM</i>	139
8 Discussões Finais e Conclusões	145
8.1. Comparação entre <i>RoleEP/EpsilonJ</i> e ArchM/ <i>AspectM</i>	145
8.2. Contribuições	149
8.3. Trabalhos Futuros	151
9 Referências	152

## Lista de Figuras

<b>Figura 1.</b> Entrelaçamento de Interesses no Código de SMAs	16
<b>Figura 2.</b> Estrutura Geral de SMAs Móveis	27
<b>Figura 3.</b> Ciclo de Vida de Agentes Móveis	27
<b>Figura 4.</b> Paradigma Cliente-Servidor	30
<b>Figura 5.</b> Paradigma de Execução Remota	30
<b>Figura 6.</b> Paradigma de Código sob Demanda	31
<b>Figura 7.</b> Paradigma de Agentes Móveis	31
<b>Figura 8.</b> Tipos de Agentes, Papéis e Planos no Expert Committee	43
<b>Figura 9.</b> Solução para EC baseada somente em APIs	46
<b>Figura 10.</b> Colaboração de Classes no Padrão Master-Slave	50
<b>Figura 11.</b> Colaboração de Classes no Padrão Itinerary	50
<b>Figura 12.</b> Solução para EC baseada em APIs e Padrões de Projeto	51
<b>Figura 13.</b> Conceitos do Modelo de RoleEP	54
<b>Figura 14.</b> Solução para EC baseada em RoleEP e EpsilonJ	56
<b>Figura 15.</b> Tratamento de Erros: Exemplo de Interesse Transversal	61
<b>Figura 16.</b> Implementação do Aspecto FaultHandler	66
<b>Figura 17.</b> Projeto do Aspecto FaultHandler	67
<b>Figura 18.</b> Interação entre Classe Server e Aspecto FaultHandler	68
<b>Figura 19.</b> Modelagem do Aspecto Arquitetural FaultHandler	70
<b>Figura 20.</b> A Arquitetura de Software ArchM	73
<b>Figura 21.</b> Subpassos do Ciclo de Vida de Agentes Móveis	81
<b>Figura 22.</b> Estrutura OO para Separação de Mobilidade em SMAs	82
<b>Figura 23.</b> Estrutura OA para Separação de Mobilidade em SMAs	85
<b>Figura 24.</b> Aplicação de Container Introduction em MobileElement	86
<b>Figura 25.</b> Detecção do Conjunto de Junção de Inicialização	88
<b>Figura 26.</b> Projeto Detalhado do Componente MobilityProtocol	92
<b>Figura 27.</b> Protocolo de Instanciação no Componente MobilityProtocol	93
<b>Figura 28.</b> Protocolo de Movimentação no Componente MobilityProtocol	95
<b>Figura 29.</b> Protocolo de Inicialização no Componente MobilityProtocol	96
<b>Figura 30.</b> Protocolo de Destruição no Componente MobilityProtocol	97

<b>Figura 31.</b> Instanciação do AspectM para o Agente UserAgent	99
<b>Figura 32.</b> Protocolo de Movimentação do Agente UserAgent	101
<b>Figura 33.</b> Pontos de Flexibilização de Plataformas no AspectM	107
<b>Figura 34.</b> ReferenceCreator e ReferenceManager na Instanciação	108
<b>Figura 35.</b> ReferenceManager no Protocolo de Inicialização do Agente	110
<b>Figura 36.</b> ReferenceManager no Protocolo de Destruição do Agente	111
<b>Figura 37.</b> Diagrama de Classes do Sistema Expert Committee	113
<b>Figura 38.</b> Implementação do Papel Chair no Expert Committee	115
<b>Figura 39.</b> Sistema Expert Committee utilizando AspectM e JADE	117
<b>Figura 40.</b> Implementação da Classe de Itinerário ChairItinerary	120
<b>Figura 41.</b> Protocolo de Instanciação no Aspecto ChairMobility	121
<b>Figura 42.</b> Protocolo de Movimentação no Aspecto ChairMobility	123
<b>Figura 43.</b> Protocolo de Inicialização no Aspecto ChairMobility	124
<b>Figura 44.</b> Protocolo de Destruição no Aspecto ChairMobility	124
<b>Figura 45.</b> Cenário do Protocolo de Movimentação do Papel Chair	125
<b>Figura 46.</b> Sistema Expert Committee utilizando AspectM e Aglets	127
<b>Figura 47.</b> Arquitetura Geral do Framework MobiGrid	131
<b>Figura 48.</b> Diagrama de Classes do Framework MobiGrid	132
<b>Figura 49.</b> Reengenharia do Framework MobiGrid utilizando AspectM	134
<b>Figura 50.</b> Reengenharia de MobiGrid: Aspecto ServerMobility	137
<b>Figura 51.</b> Reengenharia de MobiGrid: Aspecto ManagerMobility	137
<b>Figura 52.</b> Reengenharia de MobiGrid: Aspecto TaskMobility	138

## Lista de Tabelas

<b>Tabela 1.</b> Comparação entre Paradigmas da Computação em Rede	32
<b>Tabela 2.</b> Ciclo de Vida dos <i>Aglets</i> e Métodos Relacionados	37
<b>Tabela 3.</b> Nível de Satisfação de Requisitos de Mobilidade por AspectM143	

## Lista de Siglas e Abreviaturas

API	<i>Application Programming Interface</i>
ArchM	<i>Architecting Mobility</i> ou Arquetetando Mobilidade
AspectM	<i>Aspectizing Mobility</i> ou Aspectizando Mobilidade
DSOA	Desenvolvimento de Software Orientado a Aspectos
EC	<i>Expert Committee</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
JADE	<i>Java Agent DEvelopment Framework</i>
JDK	<i>Java Development Kit</i>
JVM	<i>Java Virtual Machine</i> ou Máquina Virtual Java
MASIF	<i>Mobile Agent System Interoperability Facility</i>
OA	Orientação a Aspectos
OO	Orientação a Objetos
POA	Programação Orientada a Aspectos
SMA	Sistema Multi-Agentes
UML	<i>Unified Modeling Language</i>