



**Guilherme Nascimento Pate Santos**

**Introduzindo Variabilidade no Desenvolvimento de  
Sistemas Multi-Agentes**

**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  
Ricardo Choren Noya

Rio de Janeiro  
Março de 2007



**Guilherme Nascimento Pate Santos**

**Introduzindo Variabilidade no  
Desenvolvimento de Sistemas Multi-Agentes**

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. Dr. Carlos José Pereira de Lucena**

Orientador

Departamento de Informática – PUC-Rio

**Prof. Dr. Ricardo Choren Noya**

Co-orientador

Departamento de Engenharia de Sistemas – IME

**Prof. Dr. Firmo Freire**

Departamento de Informática – PUC-Rio

**Prof. Dr. Simone Diniz Junqueira**

**Barbosa**

Departamento de Informática – PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 22 de Maio de 2007

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

### **Guilherme Nascimento Pate Santos**

Graduou-se em Matemática na Universidade do Estado do Rio de Janeiro (UERJ) em 2003, é Pós-graduado em Projeto Análise e Gerência de Sistemas pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2004 e no mesmo ano foi professor substituto do departamento de Informática e Ciência da Computação da Universidade do Estado do Rio de Janeiro (UERJ).

#### Ficha Catalográfica

Santos, Guilherme Nascimento Pate

Introduzindo variabilidade no desenvolvimento de sistemas multi-agentes / Guilherme Nascimento Pate Santos; orientadores: Carlos José Pereira de Lucena, Ricardo Choren Noya. – Rio de Janeiro: PUC, Departamento de Informática, 2007.

110 f.; 30 cm

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

Inclui referências bibliográficas.

1. Informática – Teses. 2. linguagem de modelagem. 3. Sistemas Multi-Agentes. 4. Frameworks. 5. Hooks. 6. RDL. 7. Engenharia de Software. I. Lucena, Carlos José Pereira de. II. Noya, Ricardo Choren. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

*A Deus, a minha família e a todos que me ajudaram nessa jornada.*

## Agradecimentos

A Deus, por ter-me dado calma nas horas mais críticas.

A toda a minha família, principalmente minha mãe Dona Valeria (Mamita), meu pai Sr. Eliomar (Papito), minha irmã Juliana (Julicana), minha vó Dona Thereza (TT), meu avô Sr. Mario (Velinho), meu primo Leonardo (Leoncio) e minha dinda Ana Maria (a barraqueira) que estão sempre unidos nos momentos mais difíceis.

A minha namorada Thais Barbosa pelo apoio e carinho durante estes dois anos de Mestrado.

A amiga, ex.professora e orientadora Claudia Angeli pela orientação em minha Iniciação Científica, pelo incentivo para continuar meus estudos e pelos conselhos que foram de grande valia.

Aos amigos, ex.professores e orientadores Paulo Rogério Sabini e Aruquia Barbosa Matos pela orientação na monografia de graduação e pelo incentivo em continuar meus estudos.

Ao meu orientador Prof<sup>o</sup> Carlos Lucena e ao meu co-orientador Prof<sup>o</sup> Ricardo Choren, agradeço todos os ensinamentos e apoio recebido. Sinto-me grato por ter convivido com pessoas tão geniais, experientes e profissionais. Agradeço a confiança, a paciência e o incentivo.

Ao Alberto Sardinha pelas explicações sobre o *LearnAgents*, foram extremamente esclarecedoras.

Aos amigos que ingressaram no Mestrado junto comigo André Fialho, Daniel Xavier, Rafael Machado da Rocha, Ricardo Gralhoz, Renato Novais e Rodrigo Laiola que sempre arrumam uma forma de ajudar e animar uns aos outros.

Aos colegas do Laboratório de Engenharia de Software, particularmente aos

amigos Luiz Fernando (LF), Rafael Machado da Rocha, Ricardo Gralhoz, Renato Moraes, Rodrigo Paes e Uirá Kulesza pelo ambiente de trabalho formidável.

A todos os professores e funcionários do departamento de informática pelos ensinamentos, conselhos e ajudas.

A CAPES e principalmente aos meus pais Eliomar e Valéria pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

## Resumo

Santos, Guilherme Nascimento Pate. **Introduzindo Variabilidade no Desenvolvimento de Sistemas Multi-Agentes**. Rio de Janeiro, 2007. 110p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As linguagens de modelagem de agentes visam representar o sistema e seus agentes através de diagramas, os quais permitem explicitar seus objetivos, planos e ações. Mesmo as linguagens fornecendo todos esses mecanismos de representação, alguns sistemas ainda não podem ser expressos de maneira ideal, pois tais linguagens visam representar sistemas inteiros e não uma linha de produção com várias instâncias. O método proposto visa num primeiro momento determinar o mapeamento dos pontos de flexibilização em agentes de software. Planos e ações podem ser flexibilizados por apresentarem características de variabilidade. A variabilidade pode ser interpretada de duas formas: a variabilidade de planos que possibilita que um agente tenha vários planos distintos, acarretando aplicações distintas em função do seu plano; e a variabilidade de ações que possibilita que ações sejam executadas de maneiras distintas o que também acarreta aplicações distintas. Com isso podem ser observadas ações e planos abstratos que serão herdados por outras ações e planos concretos e que definirão uma instância para novas aplicações. Para isto o método proposto utiliza diagramas, *tags* e documentação para guiar a instanciação dos planos e ações. Esses agentes gerarão instâncias de aplicações devido seus planos e ações os quais determinam uma linha de produto de software e conseqüentemente a idéia de *framework*. Esta abordagem traz para o mundo dos agentes as vantagens observadas com o uso destas técnicas em orientação a objeto. Além disso, as linguagens atuais poderão utilizar concomitantemente tal método. Os benefícios da solução serão apresentados através de estudos de caso.

## Palavras-chave

Linguagens de Modelagem de Agentes; Sistemas Multi-Agentes; *Frameworks*; linha de Produto; *Hooks*; Variabilidades; Engenharia de Software.

## Abstract

Santos, Guilherme Nascimento Pate. **Introducing a Variability into Development of Multi-Agent Systems**. Rio de Janeiro, 2007. 110p. Master Thesis - Computer Science Department, Pontifical Catholic University of Rio de Janeiro.

The current agents' modeling languages aim at representing the system and their agents in a clear way by diagrams, which permit shows their goals, plans and actions. Even with all provided by the language, some systems can't be represented a correct form yet, because the current agents' modeling languages represent only a whole system and not a product line. In this way the method proposed tries to determine a mapping of the flexibility points into software agents. The flexibility points into agents are defined to their plans and actions. Such points can be flexible if show a variability characteristic. The variability is presented by two points of view: the variability of plans and variability of action, where the variability of plans can enable many distinct plan for an agent, in other words, accept distinct applications for each one of their plans; and the variability of actions that enable the actions are executed in distinct form that result in distinct applications. With that, we can see an abstract actions and abstract plans will be inherited by the concrete actions and concrete plans, and that in the future will define new applications. For this the method uses the diagrams, tags and documentation. The documentation is used like a guide in a plan instance and action instance. After, these agents can generate the instance of new applications by yours owner plans and actions that's determine the software product line and consequently it is possible to use the framework idea. With this approach it is possible to introduce into agent world all the advantage of frameworks and product lines, techniques that are traditionally used in object orientation. Moreover, the approach can be used concomitantly with current agents' modeling languages. The benefits of the approach will be shown in more details through a case study.

## Keywords

Agents' Modeling Language; Multi-Agent Systems; Frameworks; Product Line; Hooks; Variability; Software Engineer.



## Sumário

1 Introdução	17
1.1. Descrição do Problema	18
1.2. Motivação	20
1.3. Abordagem	21
1.4. Contribuições	21
1.5. Estrutura da Dissertação	22
2 Conceitos Básicos	24
2.1. Agentes e Sistemas Multi-Agentes	25
2.2. Variabilidade e Framework	27
3 Trabalhos Relacionados	32
3.1. Linguagens de Modelagem de SMA	32
3.2. Linguagens de Modelagem de SMA e Representação de Variabilidade	40
3.3. Pontos de Flexibilização de SMA	43
4 A Solução	46
4.1. Problema Exemplo	48
4.2. Flexibilização de Plano	56
4.3. Flexibilização de Ação	68
4.4. Instanciando uma Aplicação	72
4.5. Discussão	76
5 Estudo de Caso	79
5.1. Sistema LearnAgents	79
5.2. Modelando LearnAgents com o ANote	81
5.3. Pontos de Flexibilização do Sistema LearnAgents	86
5.4. Modelando LearnAgents com o Método GP-MVAP e o ANote	87

6 Conclusões e Trabalhos Futuros	91
6.1. Contribuições	92
6.2. Trabalhos Futuros	93
7 Referências	95
Anexo A - Ciclo médico paciente	100
Anexo B - Modelagem CQPM com as linguagens apresentadas	103
Anexo C - Um Exemplo de Flexibilização de Ação	107

## Lista de Figuras

Figura 1. Estrutura do agente de software.	25
Figura 2. Interação entre os agentes de um SMA.	27
Figura 3. Diagrama de protocolo da AUML.	33
Figura 4. Diagrama de classe da AUML.	33
Figura 5. Diagrama de <i>deployment</i> da AUML.	34
Figura 6. Elementos da notação i* utilizada pela metodologia Tropos.	35
Figura 7. Modelagem parcial com o Tropos.	35
Figura 8. Meta-modelo do ANote.	37
Figura 9. Elementos do diagrama de objetivo do ANote.	37
Figura 10. Elementos do diagrama de agente do ANote.	38
Figura 11. Elementos do diagrama de ambiente do ANote.	38
Figura 12. Elementos do diagrama de organização do ANote.	39
Figura 13. Elementos do diagrama planejamento do ANote.	40
Figura 14. Elementos do diagrama de comunicação do ANote.	40
Figura 15. Representação de um <i>framework</i> .	43
Figura 16. Decomposição dos objetivos relacionados com a prescrição.	52
Figura 17. Arquitetura do sistema CQPM.	52
Figura 18. Diagrama <i>flexibility view</i> .	58
Figura 19. Diagrama <i>action note view</i> .	59
Figura 20. Uso conjunto dos diagramas <i>flexibility view</i> e <i>action note view</i> .	60
Figura 21. Diagrama <i>planning view</i> do ANote estendido para facilitar a integração com o método GP-MVAP.	61
Figura 22. Agente dosador do sistema CQPM modelado com o <i>planning view</i> estendido.	65
Figura 23. Diagrama <i>flexibility view</i> modelando o plano do agente dosador do sistema CQPM.	67
Figura 24. Diagrama <i>flexibility view</i> sendo utilizado em conjunto com o <i>action note view</i> .	68
Figura 25. Diagrama <i>planning view</i> estendido para flexibilizar ações.	69
Figura 26. Arquitetura do <i>LearnAgents</i> .	80

Figura 27. Decomposição dos objetivos relacionados com o comércio de bens.	81
Figura 28. Arquitetura formal do <i>LearnAgents</i> .	82
Figura 29. <i>Flexibility view</i> para o agente negociador.	88
Figura 30. Ciclo de vida Médico-Paciente.	100
Figura 31. Escopo do sistema.	101
Figura 32. O processo de diagnóstico e seus subprocessos.	101
Figura 33. O processo de prescrição e seus subprocessos.	102
Figura 34. Modelo parcial de interação de protocolos da AUMML para o agente dosador da aplicação estudo de caso CQPM.	103
Figura 35. Modelo parcial de objetivo do agente dosador da aplicação estudo de caso CQPM.	104
Figura 36. Modelo <i>planning view</i> do agente dosador com a linguagem ANote.	105
Figura 37. Diagrama <i>planning view</i> estendido para flexibilizar ações.	108

## Lista de Tabelas

Tabela 1. Elementos do diagrama de cenário do ANote.	39
Tabela 2. Mapeamento entre subproblemas e tipos de agentes.	52
Tabela 3. Diagrama <i>scenario view</i> do agente medicamentoso.	53
Tabela 4. Diagrama <i>scenario view</i> do agente regulamentador.	54
Tabela 5. Diagrama <i>scenario view</i> do agente dosador.	55
Tabela 6. Diagrama <i>scenario view</i> do agente cíclico.	55
Tabela 7. Diagrama <i>scenario view</i> do ANote estendido para facilitar a integração com o método GP-MVAP.	61
Tabela 8. Tabela de definições dos elementos utilizados nos diagramas referentes à flexibilização de planos.	62
Tabela 9. Tabela de definições dos elementos utilizados nos novos diagramas referentes à flexibilização de planos.	63
Tabela 10. Agente dosador do sistema CQPM modelado com o <i>scenario view</i> estendido.	65
Tabela 11. Diagrama <i>scenario view</i> estendido para flexibilizar ações.	70
Tabela 12. Tabela de definições dos elementos utilizados nos diagramas a fim de flexibilizar uma ação.	71
Tabela 13. Tabela de definições dos elementos utilizados nos diagramas a fim de flexibilizar uma ação.	71
Tabela 14. Diagrama <i>instantiation view</i> baseado em RDL e utilizado para guiar o processo de instanciação de planos e ações.	72
Tabela 15. Tabela de definição dos principais elementos utilizados no <i>instantiation view</i> .	75
Tabela 16. Diagrama <i>instantiation view</i> guiando a instanciação de um plano.	76
Tabela 17. Mapeamento entre sub-problemas e tipos de agentes.	81
Tabela 18. Diagrama <i>scenario view</i> dos agentes sensores.	82
Tabela 19. Diagrama <i>scenario view</i> dos agentes preditores.	83
Tabela 20. Diagrama <i>scenario view</i> de interação do agente alocador mestre com os alocadores escravos.	83
Tabela 21. Diagrama <i>scenario view</i> do agente alocador escravo.	84

Tabela 22. Diagrama <i>scenario view</i> de interação do agente alocador mestre o agente ordenador.	84
Tabela 23. Diagrama <i>scenario view</i> do agente ordenador.	85
Tabela 24. Diagrama <i>scenario view</i> dos agentes negociadores.	85
Tabela 25. Diagrama <i>scenario view</i> estendido.	87
Tabela 26. Técnica baseada em RDL utilizada para guiar a instanciação de planos.	89
Tabela 27. Modelo <i>scenario view</i> do agente dosador com a linguagem ANote.	105
Tabela 28. Diagrama <i>scenario view</i> estendido para flexibilizar ações.	108
Tabela 29. Diagrama <i>instantiation view</i> guiando a instanciação de uma ação.	109

## Lista de Siglas e Abreviaturas

API	<i>Application Programming Interface</i>
AUML	<i>Agent UML</i>
UML	<i>Unified Modeling Language</i>
RDL	<i>Reuse Description Language</i>
xFIT	<i>XML-based Framework Instantiation Tool</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
AAMAS	<i>Autonomous Agents and MultiAgent Systems</i>
TAC	<i>Trading Agente Competition</i>
CPU	<i>Central Processing Unit</i> ou Unidade Central de Processamento
OO	Orientação a Objeto
SMA	Sistema Multi-Agente
CQPM	Controle de Qualidade em Prescrições Médicas
GP-MVAP	Guilherme Pate - Method for Variability Agent Plan ou GP - Method for Variability Agent Plan

*Não existe sanduíche de graça.*