Ingrid Oliveira de Nunes

A Domain Engineering Process for Developing Multi-agent Systems Product Lines

MASTER DISSERTATION

DEPARTMENT OF INFORMATICS

Postgraduate Program in Informatics

Rio de Janeiro

March 2009

Pontifícia Universidade Católica
do Rio de Janeiro

Ingrid Oliveira de Nunes

A Domain Engineering Process for Developing Multi-agent Systems Product Lines

Master Dissertation

Dissertation presented to the Postgraduate Program in Informatics of the Department of Informatics, PUC–Rio, as partial fulfillment of the requirements for the Master Degree in Informatics.

Supervisor : Prof. Carlos José Pereira de Lucena
Co-supervisor: Prof. Uirá Kulesza

Rio de Janeiro
March, 2009

Pontifícia Universidade Católica
do Rio de Janeiro

Ingrid Oliveira de Nunes

A Domain Engineering Process for Developing Multi-agent Systems Product Lines

Dissertation presented to the Postgraduate Program in Informatics of Department of Informatics, PUC–Rio, as partial fulfillment of the requirements for the Master Degree in Informatics.

Prof. Carlos José Pereira de Lucena
Supervisor
Departmento de Informática — PUC–Rio

Prof. Uirá Kulesza
Co-supervisor
Departamento de Informática e Matemática Aplicada – UFRN

Prof. Ricardo Choren Noya
Seção de Engenharia de Computação – IME

Prof. Simone Diniz Junqueira Barbosa
Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal
Coordinator of the Centro Técnico Científico — PUC–Rio

Rio de Janeiro— March 31, 2009

Ingrid Oliveira de Nunes

She has completed her undergraduate course at the Federal University of Rio Grande do Sul (Porto Alegre, Brazil), receiving the degree of Bachelor in Computer Science in 2006. She worked as a software developer at the e-Core Desenvolvimento de Software company from 2005 to 2007. She is currently a Ph.D. student and researcher at the Pontifical Catholic University of Rio de Janeiro, and integrates the Software Engineering Laboratory (LES). Her focus is Software Engineering, and her main research interests are multi-agent systems and software product lines.

Ficha Catalográfica

Nunes, Ingrid Oliveira de

A Domain Engineering Process for Developing Multi-agent Systems Product Lines / Ingrid Oliveira de Nunes; orientador: Carlos José Pereira de Lucena; co–orientador: Uirá Kulesza. — Rio de Janeiro : PUC–Rio, Departamento de Informática, 2009.

138 f: ; 29,7 cm

1. Dissertação (mestrado) - Pontifical Catholic University of Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. Multi-agent Systems Product Lines. 3. Domain Engineering. 4. Software Process. 5. Software Product Lines. 6. Multi-agent Systems. 7. Software Engineering. I. Lucena, Carlos José Pereira de Lucena. II. Kulesza, Uirá. III. Pontifical Catholic University of Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

To my parents, Daltro and Suzana.
To my brothers, Gustavo and Matthias.
I love you all.

Acknowledgments

I would like to thank all of the people that have helped me in the preparation of this dissertation.

First and foremost, I would like to extend my sincere thanks to my supervisors Professor Carlos Lucena and Professor Uirá Kulesza. I would like to express my gratitude to Professor Lucena for his invaluable support and guidance. His guiding advices lighted up my way to go. Thank you for believing in me and giving me several opportunities. I would also like to thank Professor Uirá, who has provided tremendous help over this period with editing, research assistance, feedback and encouragement. Both of them are insightful professors and are people I am thankful to have been able to work with.

I would also like to thank the other members of my examining committee, Professor Ricardo Choren and Professor Simone Barbosa, for taking the time of reading my dissertation and giving me very constructive feedbacks.

My sincere appreciation is then extended to Professor Michael Luck, who visited our University last year. I am thankful for his valuable suggestions and useful discussions. Special thanks to Leandro Daflon for his SPEM lessons and Professor Viviane Torres da Silva, who gave me valuable feedback even though she recently came from Spain.

I would like to thank all my lab members for their help, support and friendship. A special thank you for Camila Nunes, who has been working with me since I started my master and has been a great friend, and Elder Cirilo, who has discussed great ideas with me and has also been a wonderful friend. Vera Menezes, thank you for helping in so many things during my master, her friendship and our endless conversations.

My thanks and gratitude goes to my family and friends, who, even far away, always believed me and encouraged me. Every time that I go to Porto Alegre, I see that truly feelings do not change.

I am especially indebted to my parents, Daltro Nunes and Suzana Nunes, for their love and support throughout my studies and work. Without their encouragement, assistance, patience, generosity and support, this dissertation would not have been able to be completed.

Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the Pontifical Catholic University of Rio de Janeiro (PUC-Rio). Graduate scholarship as well as the research assistantship have provided personal financial support for day to day living while pursuing the master degree. These supports are duly acknowledged.

Abstract

Nunes, Ingrid Oliveira de; Lucena, Carlos José Pereira de Lucena; Kulesza, Uirá. **A Domain Engineering Process for Developing Multi-agent Systems Product Lines**. Rio de Janeiro, 2009. 138p. Master Dissertation — Department of Informática, Pontifical Catholic University of Rio de Janeiro.

Multi-agent System Product Lines (MAS-PLs) have emerged to integrate two promising trends of software engineering: Software Product Lines (SPLs) and Agent-oriented Software Engineering. The former came to promote reduced time-to-market, lower development costs and higher quality on the development of applications that share common and variable features. These applications are derived in a systematic way, based on the exploitation of application commonalities and large-scale reuse. The later is a new software engineering paradigm that addresses the development of Multi-agent Systems (MASs), which are composed by autonomous and pro-active entities, named agents. This paradigm aims at helping on the development of complex and distributed systems, characterized by a large number of parts that have many interactions. The main goal of MAS-PLs is to incorporate the benefits of both software engineering disciplines and to help the industrial exploitation of agent technology. Only recent research work aimed at integrating SPL and MAS. In this context, this work presents a domain engineering process for developing MAS-PLs. We define the process activities with their tasks and work products, in order to provide a process that addresses modeling agent variability and agent features traceability. Our approach is the result of an investigation of how current SPL, MAS and MAS-PL approaches can model MAS-PLs. Based on this experience, we propose our process, which includes some techniques and notations of some of these approaches: PLUS method, PASSI methodology and MAS-ML modeling language. In particular, the scenario we are currently exploring is the incorporation of autonomous and pro-active behavior into existing web systems. The main idea is to introduce software agents into existing web applications in order to (semi)automate tasks, such as providing autonomous recommendation of products and information to users.

Keywords

Multi-agent Systems Product Lines. Domain Engineering. Software Process. Software Product Lines. Multi-agent Systems. Software Engineering.

Resumo

Nunes, Ingrid Oliveira de; Lucena, Carlos José Pereira de Lucena; Kulesza, Uirá. **Um Processo para Engenharia de Domínio para o Desenvolvimento de Linhas de Produto de Sistemas Multi-agentes**. Rio de Janeiro, 2009. 138p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Desde a introdução de agentes no início dos anos 70, sistemas de software evoluíram de sua natureza isolada e puramente reativa para serem abertos, autônomos, altamente interativos e sensíveis ao contexto. Sistemas Multi-agentes (SMAs), os quais se baseiam na abstração de agentes, surgiram como uma importante abordagem para sistemas complexos e distribuídos com estas características. Agentes são entidades pró-ativas, reativas, autônomas e que exibem características sociais. A propriedade de autonomia refere-se à habilidade de um agente de agir sem a intervenção de humanos e outros sistemas, dado que um agente autônomo tem controle tanto sobre seu estado interno como seu comportamento.

O advento de abordagens computacionais autonômicas, pervasivas e inspiradas na biologia, nas quais muitas propriedades de agentes tornam-se essenciais, conduziu a uma adoção crescente de abordagens multi-agentes para a construção de sistemas distribuídos. Conseqüentemente, avanços na área da Engenharia de Software Orientada a Agentes (ESOA) têm sido propostos através de novas técnicas, tais como metodologias, linguagens de modelagem, processos, e estratégias de implementação direcionadas a SMAs. Entretanto, mesmo que muitas metodologias da ESOA tenham sido introduzidas na literatura, metodologias de SMAs existentes carecem de suporte apropriado de um processo sistemático que guie o desenvolvimento de aplicações multi-agentes e que tirem proveito do potencial reuso oferecido pelas Linhas de Produto de Software (LPSs) e customização em massa em larga escala.

Metodologias de SMA combinadas com LPSs e customização em massa criam novos problemas os quais potencializam a necessidade de novos modelos e processos de desenvolvimento. Assim, neste trabalho, nós oferecemos as seguintes contribuições: (i) nós propomos um novo processo de engenharia de domínio para o desenvolvimento de Linhas de Produto de Sistemas Multi-agentes (LP-SMAs); (ii) nós estendemos abordagens de modelagem existentes para dar suporte à análise, projeto e realização de linhas de produto baseadas em SMAs através da introdução de modelos novos e modificados, bem como provemos técnicas para modelar e rastrear variabilidades de agentes; e (iii) nós aplicamos o processo sistemático em estudos de caso específicos para ilustrar e avaliar a abordagem. Nossa abordagem modela *features* de agentes de

forma independente, conseqüentemente facilitando a incorporação de agentes em sistemas existentes, os quais foram projetados com outros paradigmas (e.g. orientação-a-objetos). A relevância deste trabalho é que ele melhora o entendimento de como LP-SMAs são desenvolvidas e demonstra um processo sistemático que envolve análise, projeto e técnicas de implementação para dar suporte à sua construção.

LPSs estão se tornando uma técnica essencial de reuso que tem sido aplicada na construção de sistemas de software produzidos em massa. Uma LPS é uma família de produtos de software relacionados, construídos a partir de um conjunto comum de componentes de software, onde cada componente tipicamente implementa uma *feature* distinta. LPSs exploram e beneficiam-se de *features* comuns e variáveis de um conjunto de sistemas e conduzem ao desenvolvimento de arquiteturas flexíveis que suportam a derivação de sistemas customizados. Benefícios das LPSs incluem um tempo médio mais baixo para criar e fazer o *deploy* de um produto; um número médio de defeitos mais baixo por produto (i.e. maior qualidade nos produtos); menor esforço para fazer o *deploy* e manter um produto; menor *time-to-market* para novos produtos; maiores margens de lucro; e redução de risco no desenvolvimento de novos produtos. Enquanto LPSs têm demonstrado benefícios à engenharia de software, elas não lidam com abstrações relacionadas com agentes, tais como crenças, objetivos e planos, para permitir uma derivação sistemática de SMAs. Customização em massa permite que sistemas sejam mais flexíveis e coerentes com as necessidades dos clientes, o que significa que as *features* e opções dos produtos podem ser selecionadas e produtos finais podem ser efetivamente gerados de forma que a produção de software seja facilitada, os custos sejam mais baixos, e o tempo para o produto ser entregue ao usuário seja reduzido. Entretanto, a fim de suportar o desenvolvimento de LP-SMAs, *features* devem ser associadas com abstrações relacionadas com agentes de forma explícita.

Resumindo, nem as abordagens existentes de LPS oferecem modelos para projetar conceitos de agentes e mapear esses conceitos em *features*, nem as metodologias de SMA consideram *features* comuns e variáveis nos modelos de agentes, como também modularização e rastreamento de *features*. A maioria das abordagens atuais de LPS são voltadas para as tecnologias orientadas-a-objetos e orientadas-a-componentes, e apenas pesquisa recente visou à combinação de LPSs e SMAs. O resultado dessa integração é denominada Linha de Produtos de Sistema Multi-agentes (LP-SMA), e tem por objetivo obter vantagens de ambas tecnologias para assistir à exploração industrial da tecnologia de agentes. Entretanto, não há trabalho anterior que ofereça um processo de engenharia de domínio para o desenvolvimento de LP-SMAs.

Nós caracterizamos uma LP-SMA por uma LPS que contenha *features* de agentes, as quais são *features* que apresentam propriedades típicas de agentes, tais como comportamento autônomo ou pró-ativo, e que conseqüentemente pode ser apropriadamente modelado utilizando-se as abstrações de agentes. Para modelar essas *features* de agentes, é essencial rastrear o seu comportamento e a localização de sua variabilidade. Nosso processo de engenharia de domínio define fases, atividades e *work products* que envolvem tanto a modelagem como rastreabilidade de *features* de agentes. Na definição do nosso processo, nós introduzimos modelos novos e modificados, bem como tiramos proveito de algumas atividades e notações que consistem partes de métodos de abordagens existentes de LPSs e SMAs. As abordagens que foram utilizadas em nosso processo foram: (i) PLUS, que é um método baseado na UML, o qual fornece notações para a documentação de variabilidade; (ii) PASSI, que é uma metodologia orientada-a-agentes, a qual propõe alguns diagramas que foram usados para especificar *features* de agentes em nível de análise; e (iii) MAS-ML, a qual foi a linguagem de modelagem utilizada na fase de projeto do domínio para modelar agentes, papéis e organizações. Além disso, nós explicitamente separamos a modelagem de *features* de agentes e que não são de agentes, a fim de facilitar a integração da nossa abordagem com abordagens que não envolvam *features* de agentes (e.g. orientação-a-objetos), e para dar suporte à evolução de aplicações e LPSs existentes que foram desenvolvidas usando outras metodologias através da incorporação de *features* de agentes. Um cenário específico que nós estamos explorando é a evolução de aplicações *web* desenvolvidas com arquiteturas tradicionais, às quais nós adicionamos *features* de agentes, tais como recomendação autônoma de produtos e informações aos usuários.

Palavras–chave

Linhas de Produto de Sistemas Multi-agentes. Engenharia de Domínio. Processo de Software. Linhas de Produto de Software. Sistemas Multi-agentes. Engenharia de Software.

Table of Contents

List of Figures

List of Tables

Abbreviation and Acronym List

AOP Aspect-oriented Programming

AOSE Agent-oriented Software Engineering

API Application Program Interface

BDI belief-desire-intention

DAO Data Access Object

EC ExpertCommittee

FIPA Foundation for Intelligent Physical Agents

FODA Feature-Oriented Domain Analysis

FORM Feature-Oriented Reuse Method

GUI Graphical User Interface

KobrA Komponentenbasierte Anwendungsentwicklung

MaCMAS Methodology Fragment for Analysing Complex MultiAgent Systems

MAS-PL Multi-agent System Product Line

MAS Multi-agent System

MDD Model-driven Development

MPP marketing and product plan

MVC model-view-controller

OMG Object Management Group

OLIS OnLine Intelligent Services

OVM Orthogonal Variability Model

PASSI Process for Agent Societies Specification and Implementation

PLUS Product Line UML-based Software Engineering

SPEM Software Process Engineering Metamodel Specification

SPL Software Product Line

SPLE Software Product Line Engineering

TAO Taming Agents and Objects

UML Unified Modeling Language

WAF Web Application Framework

XML Extensible Markup Language