

Maíra Athanázio de Cerqueira Gatti

**Engineering Self-Organizing Emergent
Multi-Agent Systems: A Design Method
and Architecture**

TESE DE DOUTORADO

Thesis presented to the Postgraduate Program in Informatics
of the Departamento de Informática PUC-Rio as partial
fulfillment of the requirements for the degree of Doutor em
Informática

Advisor: Prof. Carlos José Pereira de Lucena

Rio de Janeiro
December 2009



Maíra Athanázio de Cerqueira Gatti

**Engineering Self-Organizing Emergent
Multi-Agent Systems: A Design Method
and Architecture**

Thesis presented to the Postgraduate Program in Informatics
of the Departamento de Informática PUC-Rio as partial
fulfillment of the requirements for the degree of Doutor em
Informática.

Prof. Carlos José Pereira de Lucena

Advisor

Departamento de Informática — PUC-Rio

Prof. Vera Maria Benjamim Werneck

Departamento de Informática e Ciência da Computação –
UERJ

Prof. Viviane Torres da Silva

Departamento de Ciência da Computação – UFF

Prof. Antônio Furtado

Departamento de Informática – PUC-Rio

Prof. Bruno Feijó

Departamento de Informática – PUC-Rio

Prof. José Eugênio Leal

Coordinator of the Centro Técnico Científico da PUC-Rio

Rio de Janeiro , 21/12/2009

All rights reserved.

Maíra Athanázio de Cerqueira Gatti

Maíra Gatti graduate in Informatics and Computer Science from Rio de Janeiro State University (2004) and master's at Informatics in Software Engineering from Pontifícia Universidade Católica do Rio de Janeiro (2006) in cooperation with Laboratoire d'informatique de Paris 6, Université Pierre et Marie Curie (Paris 6) France. During her PhD, she had collaborative period in King's College London, U. of London, grantee of CNPq, Scholarship award. And in The School of Computer Science, University of Waterloo, Canada, grantee of CNPq-Funding project, Scholarship award. She has experience in Computer Science, focusing on Software Engineering, acting on the following subjects: multi-agent systems, self-organizing systems, adaptive systems, distributed systems, service-oriented systems, fault-tolerance, autonomic computing and computational biology.

Bibliographic data

Gatti, Maíra Athanázio de Cerqueira

Engineering Self-Organizing Emergent Multi-Agent Systems: A Design Method and Architecture / Maíra Athanázio de Cerqueira Gatti; advisor: Carlos José Pereira de Lucena. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2009.

v., 153 f. il. ; 29,7 cm

1. Tese (Doutorado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

Inclui bibliografia.

1. Departamento de Informática – Tese. 2. Auto-Organização. 3. Projeto. 4. Simulação. 5. Arquitetura. 6. Sistemas Multi-Agentes. 7. Engenharia de Software. I. Lucena, Carlos José Pereira de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To dad and mom.

Acknowledgments

I would like to thanks everyone who helped me during the development of this work, this includes my advisor Prof. Dr. Carlos Lucena, the Software Engineering Lab's research group (LES) from the PUC-Rio university, the LANDIC research group from the UFRJ university and the CNPq - National Counsel of Technological and Scientific Development agency for giving me the financial support during my PhD.

Many thanks goes out to Prof. Dr. Michael Luck for giving me the opportunity to work with the Agents and Intelligent Systems (AIS) sub-group of the Software Engineering research group in the Department of Computer Science at King's College London. The friendly and supportive atmosphere inherent to the AIS group contributed essentially to the final outcome of my studies. I am also grateful to the King's College London for giving me the opportunity to work as a Visiting Research Associate in the Department of Computer Science during my PhD.

Finally, a very special thanks goes out to my love Rodrigo, my parents, my British family, specially Linda, and all my dearest friends.

Resumo

Gatti, Maíra Athanázio de Cerqueira; Lucena, Carlos José Pereira de. **Engenharia de Sistemas Multi-Agentes Auto-Organizáveis: Um Método de Projeto e Arquitetura**. Rio de Janeiro, 2009. 153p. Tese de Doutorado — Departamento de Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Arquitetos de software utilizam cada vez mais mecanismos de auto-organização para projetar sistemas distribuídos em um ambiente dinâmico, com ruído e imprevisível. Neste cenário, além de não existir uma entidade centralizadora que possua o conhecimento completo do estado do ambiente como um todo, mecanismos de auto-organização são principalmente inspirados pela natureza e permitem o comportamento com controle descentralizado. Existem duas linhas de pesquisa que direcionam esta tese: a primeira, como é o caso de qualquer paradigma de engenharia de software, o sucesso e abrangência de uso de sistemas auto-organizáveis requerem notações que explorem o uso de abstrações relacionadas a auto-organização e promovam a rastreabilidade a partir de modelos de projeto à código, e requerem métodos de engenharia para prover know-how e guiar um engenheiro durante o projeto da aplicação. A segunda linha de pesquisa foca no incentivo do reuso de software em sistemas auto-organizáveis. Os objetivos desta tese são: prover um método de engenharia baseado em simulação para apoiar o projeto, desenvolvimento, simulação, validação e refinamento de sistemas multi-agentes auto-organizáveis; e prover uma arquitetura baseada em simulação. Um projeto arquitetural ajuda no desenvolvimento de uma estrutura de programa modular e na representação dos relacionamentos de controle entre módulos e encoraja o engenheiro de software a se concentrar no projeto arquitetural antes de se preocupar com otimizações e código. Esta tese apresenta: um ciclo de vida de desenvolvimento iterativo baseado no Processo Unificado, SSOA – uma arquitetura de auto-organização baseada em simulação, e o framework que implementa a arquitetura. O método de projeto e a arquitetura foram avaliados através de três domínios de aplicação diferentes: veículos guiados automatizados, contratos que governam sistemas multi-agentes emergentes, e modelagem computacional do comportamento de células-tronco.

Palavras-chave

Auto-Organização; Projeto; Simulação; Arquitetura; Sistemas Multi-Agentes; Engenharia de Software;

Abstract

Gatti, Maíra Athanázio de Cerqueira; Lucena, Carlos José Pereira de. **Engineering Self-Organizing Emergent Multi-Agent Systems: A Design Method and Architecture**. Rio de Janeiro, 2009. 153p. DSc Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Software architects are increasingly relying on self-organizing mechanisms to design distributed systems within a dynamic, noisy and unpredictable environment. At any point in time, no centralized entity has complete knowledge of the state of the environment as a whole and self-organizing mechanisms are mainly naturally-inspired which enables the decentralized control. There are two tracks that drive the research in this thesis: first, as it is the case with any new software engineering paradigm, the successful and widespread deployment of self-organizing systems require notations that explore the use of self-organizing related abstractions and promote the traceability from the design models to code, and engineering methods that provides know-how and guides an engineer during an application design. The second research track is to promote software reuse of self-organizing systems. This thesis's goals are: to provide a simulation-based engineering method to support the design, development, simulation, validation and refinement of self-organizing multi-agent systems; and to provide a simulation-based architecture. An architectural design helps on the development of a modular program structure and on the representation of the control relationships between modules and encourages the software engineer to concentrate on architectural design before worrying about optimizations or code. We present: an iterative development life-cycle based on a customization of the Unified Process, the SSOA – Simulation-based Self-Organizing Architecture –, and the framework that implements the architecture. We have evaluated the design method and architecture using three different application domains: the automated guided vehicles, the contracts that govern emergent multi-agent systems, and the stem cell behavior computational modeling.

Keywords

Self-Organization; Design; Simulation; Architecture; Multi-Agent Systems; Software Engineering;

Contents

1	Introduction	14
1.1	Problem Statement	15
1.2	Goals	17
1.3	Contributions	18
1.4	Structure of the Thesis	19
2	Background	21
2.1	Self-Organizing Emergent Systems	21
2.2	ABS: Agent-based Simulation	25
2.3	Case Studies	28
2.4	Chapter Remarks	37
3	Notational Support	38
3.1	Information Flow Design	38
3.2	UML Customization with Coordinated Statecharts	39
3.3	Exemplar Application: Toward a Self-Organizing Pattern Language	45
3.4	Chapter Remarks	57
4	SSOA: A Simulation-based Self-Organizing Architecture	58
4.1	SSOA Iterative Life-Cycle	58
4.2	The Architecture Requirements	63
4.3	The SSOA Description and Dynamics	68
4.4	A Framework that Implements SSOA	75
4.5	Implementation Blueprints	76
4.6	Chapter Remarks	77
5	Case Studies	79
5.1	The Automated Guided Vehicles	79
5.2	Contracts that Govern Emergent Multi-Agent Systems	87
5.3	Computational Modeling of Stem Cell Behavior	100
5.4	Manager Overhead	111
5.5	Chapter Remarks	111
6	Related Work	112
6.1	Modeling and Designing Self-Organizing Systems	112
6.2	Architectures and Middlewares for Self-Organizing Systems	121
6.3	Validation and Verification of Self-Organizing Systems	123
6.4	Chapter Remarks	125
7	Conclusions and Future Work	127
7.1	Contributions	127
7.2	Limitations	129
7.3	Future Directions	130
A	List of Publications (PhD related)	151

A.1	Article in Journal/Book	151
A.2	Contributions at International Conferences, Published in the Proceedings	151
A.3	Contributions at National (Brazilian) Conferences, Published in the Proceedings	152
A.4	Technical Reports (selected)	153

List of Figures

2.1	AGV case: example factory lay-out, from [De Wolf 2007]	29
2.2	The Aerospace Scenario	31
2.3	The Stem Cell Division and Differentiation Process	35
3.1	Cordinated Statecharts Concept	41
3.2	The customized UML 2.0 meta-model	42
3.3	The coordinated state machine notation for an agent's behavior	42
3.4	Coordinated state machine notation - input versus output co-ordinated signal perceptions	44
3.5	Self-Organizing Design Patterns and Their Relationships	47
3.6	Micro-Architecture Structure	48
3.7	Micro-Architecture Dynamics	49
3.8	AGV Structure	51
3.9	AGV Dynamics: Dispatching Property	52
4.1	The Design Iterative Life-Cycle w.r.t the Unified Process	59
4.2	Coordination Support for Feedback Loops	65
4.3	The Multi-Environment Perspective	66
4.4	The Multi-Environment Perspective: a)Graph: each agent or sub-environment can be located in a node and perceives its neighbors; b) 2D double point grid: each agent or sub-environment can be located in a discrete 2D double point position in the grid; c)3D continuous grid: each agent or sub-environment can be located in a 3D floating point grid.	67
4.5	The SSOA Components	68
4.6	The MESOF Meta-Model	69
4.7	The coordination between two agents	72
4.8	The Manager Meta-Model	74
5.1	The AGV Class Diagram (partial view)	81
5.2	The Dispatching Property w.r.t feedback loop	82
5.3	AGV Goals and SSOA Manager Component instantiation	84
5.4	Routing throughput analysis	86
5.5	Dispatching throughput analysis	87
5.6	The Aerospace Aftercare System Class Diagram (partial view)	88
5.7	The Aircraft and Airline Operator Coordinated Statecharts	93
5.8	The Engine Manufacturer and Part Supplier Coordinated Statecharts	94
5.9	Integration process for maximizing the number of times the aircraft was in the flying state (NF) and minimizing the costs regarding norms violations (NC)	97
5.10	Plot of metric as specified in max A	99
5.11	The Stem Cell Class Diagram (partial view)	101
5.12	The Spatial Self-Organization	102
5.13	The proteins diffusion and cell perception	103

5.14	Conceptual cell life-cycle behavior	106
5.15	Gap 1 phase checkpoint coordinated statecharts	107
5.16	The desired number of stem cells evolution	110
5.17	Manager overhead in the simulation	111
6.1	The Customised Unified Process from [De Wolf 2007]	114
6.2	Information flows, from [De Wolf 2007]	115
6.3	Architectural pattern featuring environmental agents as artefact administrators, from [Gardelli 2008]	116
6.4	The ADELFE methodology [Picard 2004]	117
7.1	The dynamic meta-model editor	132

List of Tables

5.1	Norm structures for contractual clauses in simulation. Nx O refers to an obligation for norm class x, Nx P for a permission for norm class x	89
5.2	External Actions	91
5.3	Airline Operator Actions	91
5.4	Engine Manufacturer Actions	91
5.5	Part Supplier Actions	92
5.6	Contracts Variations	98
5.7	External Actions	104
5.8	Protein and Intracellular Actions	105
5.9	Cell life-cycle Actions.	105

*Reuse: Is someone using your novel design to
solve similar problems?*

Maíra Gatti.