



Viviane Torres da Silva

**Uma Linguagem de Modelagem para Sistemas
Multi-agentes Baseada em um Framework Conceitual
para Agentes e Objetos**

Tese de Doutorado

Tese apresentada ao Programa de Pós-graduação
em Informática da PUC-Rio como requisito parcial
para obtenção do título de Doutor em Informática.

Orientador: Carlos José Pereira de Lucena

Rio de Janeiro, 31 de março 2004.



Viviane Torres da Silva

**Uma Linguagem de Modelagem para Sistemas
Multi-agentes Baseada em um Framework Conceitual
para Agentes e Objetos**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Carlos José Pereira de Lucena

Orientador

Departamento de Informática – PUC-Rio

Prof. Arndt von Staa

Departamento de Informática – PUC-Rio

Profa. Simone Diniz Junqueira Barbosa

Departamento de Informática – PUC-Rio

Prof. Bruno Feijó

Departamento de Informática – PUC-Rio

Prof. Jaime Simão Sichman

Departamento de Engenharia de Computação e Sistemas Digitais – USP

Prof. Brian Henderson-Sellers

Faculty of Information Technology – University of Technology Sydney

Prof. José Eugenio Leal

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 31 de março de 2004.

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

Viviane Torres da Silva

Graduou-se em Engenharia da Computação na PUC-Rio em 1998. Obteve o grau de Mestre em Informática na PUC-Rio em 2000. É pesquisadora da área de Engenharia de Software para Sistemas Multi-Agentes do Laboratório de Engenharia de Software (LES) da PUC-Rio.

Ficha Catalográfica

Silva, Viviane Torres da

Uma linguagem de modelagem para sistemas multi-agentes baseada em um framework conceitual para agentes e objetos / Viviane Torres da Silva ; orientador: Carlos José Pereira de Lucena. – Rio de Janeiro : PUC, Departamento de Informática, 2004.

252 f. : il. ; 30 cm

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

Inclui referências bibliográficas.

1. Informática – Teses. 2. Sistemas multi-agentes. 3. Framework conceitual. 4. Linguagem de modelagem. 5. Meta-modelo. 6. Refinamento. 7. Arquitetura de software. I. Lucena, Carlos José Pereira de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

Agradecimentos

Ao amigo e professor Carlos José Pereira de Lucena por toda a amizade, por todo o carinho, por toda a confiança, por todo o apoio e por todas as oportunidades.

A todos os membros da banca por aceitarem o convite para participar da banca e por ajudaram a enriquecer o trabalho com suas críticas.

À minha família e principalmente aos meus pais, Rosalina e Joaquim, e ao meu irmão Gustavo por estarem sempre ao meu lado nos momentos mais difíceis, sempre me incentivando.

Ao meu grande amor Christiano Braga, por todo amor, carinho, apoio e presença durante esses anos.

Aos meus amigos da PUC pelas palavras de incentivo e pela grande ajuda concedida durante todo o doutorado.

Aos meus amigos fora da PUC por toda a distração que me propiciaram esquecer os momentos mais difíceis.

À amiga Vera Menezes por toda a ajuda e carinho.

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

A Deus, por simplesmente tudo.

Resumo

Silva, Viviane Torres. **Uma Linguagem de Modelagem para Sistemas Multi-agentes Baseada em um Framework Conceitual para Agentes e Objetos**. Rio de Janeiro, 2004. 252p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Como um novo e poderoso paradigma para modelagem e implementação de sistemas de software, os sistemas multiagentes necessitam de metodologias, linguagens de modelagem, plataformas de desenvolvimento e linguagens de programação que explorem seus benefícios e características particulares. Contudo, diferentes metodologias, linguagens e plataformas para sistemas multiagentes propõem abstrações variadas e com definições muito diferentes. Nesse contexto, é necessário criar *frameworks* conceituais que definam as abstrações, seus relacionamentos e seus comportamentos.

Como em qualquer novo paradigma para engenharia de software, o sucesso e a difusão de sistemas multiagentes requerem, entre outras tecnologias de software baseadas em agentes, linguagens de modelagem que explorem o uso de abstrações relacionadas a agentes e promovam o refinamento dos modelos de design para código.

Esta tese contempla a definição de um *framework* conceitual para sistemas multiagentes chamado TAO e uma linguagem de modelagem para sistemas multiagentes chamada MAS-ML. Os objetivos desta tese são descrever os aspectos estáticos e dinâmicos das abstrações freqüentemente utilizadas em sistemas multiagentes definindo um *framework* conceitual, propor uma linguagem de modelagem que descreva diagramas estáticos e dinâmicos para modelar esses aspectos e descrever o refinamento dos modelos estáticos para código.

Palavras-chave

Sistemas Multiagentes; Framework Conceitual; Linguagem de Modelagem; Metamodelo; Refinamento; Arquitetura de Software.

Abstract

Silva, Viviane Torres. **From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language.** Rio de Janeiro, 2004. 252. Doctoral Thesis – Computer Science Department, Pontifical Catholic University of Rio de Janeiro.

As a powerful and new paradigm for designing and implementing software systems, multi-agent systems require methodologies, modeling languages, development platforms and programming languages that explore their benefits and their peculiar characteristics. However, different methodologies, languages and platforms for multi-agent systems propose very distinct and varied sets of abstraction. In this context, there is a need for creating a conceptual framework that defines the frequently used multi-agent system abstractions, their relationships and their behavior.

As it is the case with any new software engineering paradigm, the successful and widespread deployment of multi-agent systems require modeling languages, among other agent-based software technologies, that explore the use of agent-related abstractions and promote the traceability from the design models to code.

This thesis contemplates the definition of a multi-agent system conceptual framework called TAO and of a multi-agent system modeling language called MAS-ML. Our goals are to describe the structural and dynamic aspects of the abstractions commonly used in multi-agent systems by defining a conceptual framework, to propose a modeling language that describes structural and dynamic diagrams to model such aspects and to present the traceability from the structural models into code.

Keywords

Multi-Agent System; Conceptual Framework; Modeling Language; Meta-Model; Refinement; Software Architecture.bb

Sumário

1	Introdução	1
1.1.	Declaração do Problema e Limitações dos Trabalhos Relacionados	2
1.1.1.	Um <i>Framework</i> Conceitual para SMAs	2
1.1.2.	Uma Linguagem de Modelagem para SMAs	3
1.2.	Solução Proposta	6
1.2.1.	O <i>Framework</i> Conceitual TAO	7
1.2.2.	A Linguagem de Modelagem MAS-ML	7
1.3.	O Relacionamento entre UML, TAO e MAS-ML	8
1.4.	As Principais Contribuições	10
1.5.	Descrição da Tese	11
2	Trabalhos Relacionados	12
2.1.	Avaliação de <i>Frameworks</i> Conceituais	12
2.1.1.	KAoS	13
2.1.2.	<i>Framework</i> Conceitual d'Inverno e Luck	14
2.1.3.	<i>Framework</i> Conceitual Yu e Schmid	15
2.2.	Visão Geral do TAO	15
2.3.	Avaliação das Linguagens de Modelagem que Estendem UML	16
2.3.1.	Linguagens de Modelagem que Estendem UML	18
2.3.1.1.	AUML	18
2.3.1.2.	A Linguagem de Modelagem AOR e o Metamodelo de AOR	21
2.3.1.3.	Transformação Gráfica e UML	23
2.3.2.	Metodologias que Incorporam as Linguagens de Modelagem que Estendem a UML	24
2.3.2.1.	MESSAGE	25
2.3.2.2.	Tropos	27
2.3.2.3.	Prometheus	28
2.3.2.4.	MaSE	30
2.4.	Visão Geral da MAS-ML	31

3 O <i>Framework</i> Conceitual TAO	33
3.1. Aspectos Estruturais do TAO	33
3.1.1. Objeto	34
3.1.2. Agente	35
3.1.3. Ambiente	37
3.1.4. Organização	38
3.1.5. Papel	39
3.1.6. Relacionamentos	41
3.2. Aspectos Dinâmicos do TAO	46
3.2.1. Processos Dinâmicos Primitivos	48
3.2.1.1. A Criação das Entidades	48
3.2.1.2. Padrão de Criação	50
3.2.1.3. A Destruição das Entidades	51
3.2.1.4. Padrão de Destruição	52
3.2.2. Processos Dinâmicos de Alto Nível	53
3.2.2.1. Relacionamentos <i>Ownership</i> e <i>Play</i>	55
3.2.2.2. Relacionamento <i>Inhabit</i>	61
3.2.2.3. Hierarquia de Processos Dinâmicos de Alto Nível	64
3.3. Avaliação do TAO	65
3.4. Discussão	66
4 A linguagem de Modelagem MAS-ML	67
4.1. Os Mecanismos de Extensão de UML	67
4.2. Metamodelo de MAS-ML	68
4.2.1. Agente	69
4.2.1.1. Propriedades do Agente	70
4.2.1.2. Associações de Agentes	73
4.2.2. Papel de Objeto	73
4.2.2.1. Propriedades de Papel de Objeto	74
4.2.2.2. Associações de Papel de Objeto	74
4.2.3. Papel de Agente	75
4.2.3.1. Propriedades dos Papéis de Agente	75
4.2.3.2. Associações de Papéis de Agente	77

4.2.4. Organização	78
4.2.4.1. Propriedades da Organização	78
4.2.4.2. Associações da Organização	79
4.2.5. Ambiente	80
4.2.5.1. Propriedades do Ambiente	80
4.2.5.2. Associações do Ambiente	81
4.2.6. O Relacionamento <i>Association</i>	83
4.2.7. O Relacionamento <i>Aggregation</i>	83
4.2.8. O Relacionamento <i>Generalization</i>	84
4.2.9. O Relacionamento <i>Dependency</i>	84
4.2.10. O Relacionamento <i>Inhabit</i>	84
4.2.11. O Relacionamento <i>Ownership</i>	85
4.2.12. O Relacionamento <i>Play</i>	85
4.2.13. O Relacionamento <i>Control</i>	86
4.3. Os Diagramas Estruturais da MAS-ML	86
4.3.1. Elementos de Diagramas Estruturais	87
4.3.1.1. ObjectRoleClass	87
4.3.1.2. AgentClass	88
4.3.1.3. AgentRoleClass	89
4.3.1.4. OrganizationClass	90
4.3.1.5. EnvironmentClass	91
4.3.1.6. O Relacionamento <i>Inhabit</i>	91
4.3.1.7. O Relacionamento <i>Ownership</i>	92
4.3.1.8. O Relacionamento <i>Play</i>	92
4.3.1.9. O Relacionamento <i>Control</i>	93
4.3.2. Diagrama de Classes	94
4.3.3. Diagrama de Organização	95
4.3.4. Diagrama de Papel	95
4.4. Diagramas Dinâmicos da MAS-ML	96
4.4.1. Elementos de Diagramas Dinâmicos	97
4.4.1.1. <i>Pathname</i> do Objeto	97
4.4.1.2. <i>Pathname</i> do Agente	99
4.4.1.3. <i>Pathname</i> da Organização	99

4.4.1.4. <i>Pathname</i> do Ambiente	100
4.4.2. Extensão do Conceito de Mensagem	101
4.4.3. Os estereótipos Associados a Mensagens	102
4.4.3.1. Os Estereótipos <<create>> e <<destroy>>	102
4.4.3.2. O Estereótipo <<role_commitment>>	104
4.4.3.3. O Estereótipo <<role_cancel>>	106
4.4.3.4. Os Estereótipos <<role_deactivate>> e <<role_activate>>	107
4.4.3.5. Os Estereótipos <<role_change>>	108
4.4.3.6. Usando os Estereótipos para Modelar os Aspectos Dinâmicos Independentes do Domínio	109
4.4.4. Modelando Planos e Ações	111
4.4.5. Modelando Protocolos	112
4.4.6. Modelando Concorrência e Distribuição	113
4.5. Discussão	114
5 Gerando Código a Partir de Diagramas de MAS-ML	116
5.1. A Linguagem de Programação TXL	117
5.2. Transformador MAS-ML2Java	118
5.2.1. Fase de Transformação de Entidades	118
5.2.2. Fase de Transformação de Relacionamentos	119
5.3. A Gramática de MAS-ML	120
5.4. Regras de Transformação da Linguagem MAS-ML	125
5.4.1. Regras de Transformação para Entidades Independentes do Domínio	125
5.4.1.1. A Regra de Agentes Básica	126
5.4.1.2. A Regra de Organizações Básica	128
5.4.1.3. A Regra de Papéis de Agentes Básica	130
5.4.1.4. A Regra de Papéis de Objetos Básica	131
5.4.1.5. A Regra de Ambientes Básica	132
5.4.2. Regras para Entidades Dependentes do Domínio	134
5.4.2.1. A Regra de Agentes	135
5.4.2.2. A Regra de Organizações	137
5.4.2.3. A Regra de Papéis de Agentes	138

5.4.2.4. A Regra de Papéis de Objetos	139
5.4.2.5. A Regra de Ambientes Proativos	140
5.4.2.6. A Regra de Ambientes Reativos	140
5.4.3. Regras para Relacionamentos Dependentes do Domínio	141
5.4.3.1. A Regra do Relacionamento <i>Specialization</i>	141
5.4.3.2. As Regras dos Relacionamentos <i>Association</i> e <i>Aggregation</i>	142
5.4.3.3. A Regra do Relacionamento <i>Control</i>	143
5.4.3.4. A Regra do Relacionamento <i>Dependency</i>	144
5.4.3.5. A Regra do Relacionamento <i>Inhabit</i>	145
5.4.3.6. A Regra do Relacionamento <i>Play</i>	146
5.4.3.7. A Regra do Relacionamento <i>Ownership</i>	148
5.5. Discussão	149
6 Usando MAS-ML para Modelar um Sistema Multiagentes	152
6.1. A Abordagem de Modelagem	153
6.1.1. Modelando Aspectos Estruturais	153
6.1.1.1. Identificação do Ambiente e da Organização Principal	153
6.1.1.2. Identificação de Papéis	154
6.1.1.3. Identificação das Entidades	154
6.1.2. Modelando Aspectos Dinâmicos	155
6.2. O Exemplo do Mercado Virtual	156
6.2.1. Diagramas Estruturais	157
6.2.1.1. Identificação da Organização Principal e Ambiente	158
6.2.1.2. Identificação de Papéis: papéis exercidos por suborganizações	159
6.2.1.3. Identificação de Suborganizações	161
6.2.1.4. Identificação de Papéis: papéis exercidos por agentes e objetos na organização principal	164
6.2.1.5. Identificação de Papéis: papéis exercidos por agentes e objetos em livrarias de livros importados e sebos	166
6.2.1.6. Identificação de Classes e Agentes	169
6.2.2. Diagramas Dinâmicos	172
6.2.3. Transformando Modelos Estruturais de MAS-ML em Código Java	177
6.2.4. A Transformação do Ambiente	177

6.2.5. As Transformações das Organizações	179
6.2.6. As Transformações dos Agentes	182
6.2.7. As Transformações dos Papéis dos Agentes	185
6.2.8. As Transformações dos Papéis de Objetos	187
6.2.9. As Transformações de Classes	189
6.3. Avaliação da Linguagem MAS-ML	192
6.4. Outros Exemplos Modelados Usando MAS-ML	193
6.5. Discussão	195
7 Conclusão e Trabalhos Futuros	198
7.1. Análise de Nossas Principais Contribuições	200
7.1.1. O <i>Framework</i> Conceitual TAO	200
7.1.2. As Extensões ao <i>Framework</i> Conceitual TAO	201
7.1.3. A Linguagem de Modelagem MAS-ML	201
7.1.4. A Abordagem de Modelagem	202
7.1.5. A Arquitetura Abstrata OO para SMAs	202
7.1.6. A Gramática de MAS-ML	203
7.1.7. O Transformador MAS-ML2Java	203
7.2. Trabalhos Futuros	204
7.2.1. Desenvolvimento de uma Ferramenta de Modelagem MAS-ML	204
7.2.1.1. Estender a Gramática de MAS-ML para Incorporar Aspectos Dinâmicos de SMAs	206
7.2.1.2. Estender a Arquitetura Abstrata	206
7.2.1.3. Criar MAS-ML DTD com Base na Gramática de MAS-ML Estendida	207
7.2.1.4. Transformar XMIs com Base em MAS-ML DTD em XMIs com Base em UML DTD	207
7.2.2. Formalizar a Semântica de MAS-ML	208
7.2.3. Analisar Outros Diagramas de UML	208
8 Referências	210
Apêndice I A Gramática MAS-ML	216

Abreviações

AgML – Agent Modeling Language

AOR – Agent-Object-Relationship

AORML – Agent-Object-Relationship Modeling Language

AUML – Agent Unified Modeling Language

ER – Entity-Relationship

FIPA – Foundation for Intelligent Physical Agents

KAoS – Knowledgeable Agent-oriented System

MaSE – Multi-agent System Engineering

SMA(s) – Sistemas Multi-Agentes(s)

MAS-ML – Multi-Agent System Modeling Language

MOF – Meta Object Facility Specification

OMG – Object Management Group

OO – Object-Oriented

TAO – Taming Agents and Objects

TXL – Turing eXtender Language

UML – Unified Modeling Language

Lista de figuras

Figura 1 – Arquitetura de metadados com quatro camadas.	10
Figura 2 - Os relacionamentos e as entidades do TAO.	46
Figura 3 – Padrão de criação.	51
Figura 4 – Padrão de destruição.	53
Figura 5 – Estados de um papel do agente.	55
Figura 6 – Entidade comprometendo-se com um papel.	59
Figura 7 – Entidade ativando um papel.	59
Figura 8 – Entidade cancelando um papel.	61
Figura 9 – Entidade desativando um papel.	61
Figura 10 – Entidade se movendo de um ambiente para outro.	64
Figura 11 – Hierarquia dos processos dinâmicos de alto nível.	65
Figura 12 – O metamodelo de UML estendido para incorporar agentes e suas propriedades.	72
Figura 13 – As associações entre AgentClass e outras metaclasses.	73
Figura 14 – O metamodelo de UML estendido para incorporar papéis de objeto.	74
Figura 15 – As associações entre ObjectRoleClass e outras metaclasses.	75
Figura 16 – O metamodelo de UML estendido para incorporar papéis do agente e suas propriedades.	77
Figura 17 – As associações entre AgentRoleClass e outras metaclasses.	77
Figura 18 – O metamodelo de UML estendido para incorporar organização e suas propriedades.	79
Figura 19 – As associações entre OrganizationClass e outras metaclasses.	80
Figura 20 – O metamodelo de UML estendido para incorporar ambientes e suas propriedades.	81
Figura 21 – As associações entre EnvironmentClass e outras metaclasses.	82
Figura 22 – O metamodelo de UML estendido para incorporar as entidades da MAS-ML e suas propriedades.	82
Figura 23 - As associações entre as metaclasses que representam as entidades do TAO.	83
Figura 24 – O metamodelo de UML estendido para incorporar os relacionamento	

MAS-ML. 86

Figura 25 – Um método sendo chamado e uma mensagem de agente sendo enviada.	102
Figura 26 – Criação e destruição de uma organização.	104
Figura 27 – Comprometendo-se com um papel.	106
Figura 28 – Cancelando um papel.	107
Figura 29 – Ativando e desativando um papel.	108
Figura 30 – Alterando papéis.	109
Figura 31 – Modelando a execução de um plano e suas ações.	112
Figura 32 – O protocolo de negociação.	113
Figura 33 – Transformação das entidades.	119
Figura 34 – Transformação de relacionamentos.	120
Figura 35 –A gramática de MAS-ML usada na fase de transformação de entidades(parcial)	122
Figura 36 – A gramática do relacionamento play.	122
Figura 37 – A gramática da classe do agente.	123
Figura 38 – As gramáticas de objetivo e crença.	123
Figura 39 – A gramática de plano.	124
Figura 40 – A gramática de ação.	124
Figura 41 – Descrição de um relacionamento play e de um agente usando a gramática.	125
Figura 42 – A classe abstrata Agent, as classes que representam suas propriedades e outras classes relacionadas a ela.	128
Figura 43 – A classe abstrata Organization, as classes que representam suas propriedades e outras classes relacionadas a ela.	130
Figura 44 – A classe abstrata Agent Role, as classes relacionadas a suas propriedades e outras classes relacionadas a ela.	131
Figura 45 – A classe abstrata Object Role e a classe Organization relacionada a ela.	132
Figura 46 – A classe abstrata Environment e as classes relacionadas a ela.	133
Figura 47 – A arquitetura abstrata orientada a objetos para sistemas multiagentes.	134
Figura 48 – A classe User Agent.	137

Figura 49 – A classe de organização General Store.	138
Figura 50 – A classe do papel do agente Buyer.	139
Figura 51 – A classe do papel de objeto Desire.	140
Figura 52 – A classe de ambiente Virtual Marketplace.	141
Figura 53 – Os relacionamentos specialization entre Buyer of Imported Books e Buyer.	141
Figura 54 – O relacionamento association entre Buyer e Seller.	142
Figura 55 – O relacionamento aggregation entre Buyer e Buyer of Imported Books.	143
Figura 56 – O relacionamento control ilustrado em Market of Special Goods que controla Seller of Imported Books.	143
Figura 57 – O relacionamento control.	144
Figura 58 – O relacionamento dependency entre Client e Supplier.	145
Figura 59 – O relacionamento inhabit entre User Agent e Virtual marketplace.	146
Figura 60 – O relacionamento inhabit.	146
Figura 61 – O relacionamento play entre Store Agent e o papel do agente Seller.	147
Figura 62 – O relacionamento play relacionando um agente a um papel do agente.	147
Figura 63 – O relacionamento play relacionando uma classe a um papel de objeto.	148
Figura 64 – O relacionamento ownership.	149
Figura 65 – O diagrama de organização modelando a organização principal.	158
Figura 66 – A classe de ambiente Virtual Marketplace (parcial).	159
Figura 67 – A classe de organização General Store (parcial).	159
Figura 68 – A classe do papel Market of special goods (parcial).	160
Figura 69 – A classe do papel Market of used goods (parcial).	161
Figura 70 – A classe de organização Imported bookstore (parcial).	162
Figura 71 – A classe de organização de Sebos (parcial).	163
Figura 72 – O diagrama de organização da classe de organização Imported Bookstore.	164
Figura 73 – O diagrama de organização da classe de organização Second-hand Bookstore.	164

Figura 74 – A classe do papel Buyer (parcial).	165
Figura 75 – A classe do papel Seller (parcial).	165
Figura 76 – A classe do papel Desire (parcial).	165
Figura 77 – A classe do papel Offer (parcial).	166
Figura 78 – A classe do papel Buyer of imported books (parcial).	167
Figura 79 – A classe do papel Buyer of second-hand books (parcial).	167
Figura 80 – A classe do papel Seller of imported books (parcial).	167
Figura 81 – A classe do papel Seller of second-hand books (parcial).	167
Figura 82 – As classes de papel desire e offer of imported books e as classes de papel desire e offer of second-hand books.	168
Figura 83 – O diagrama de papel (parte I).	169
Figura 84 – O diagrama de papel (parte II).	169
Figura 85 – As classes Book, Imported book e Second-hand book.	170
Figura 86 – A classe User agent (parcial).	171
Figura 87 – A classe Store agent (parcial).	172
Figura 88 – O diagrama de classes.	172
Figura 89 – Um agente do usuário negociando com um agente de loja.	174
Figura 90 – Um protocolo definido pelos papéis de comprador e mercado de produtos usados.	175
Figura 91 – Um agente de usuário entrando em uma organização, comprometendo-se com um novo papel e cancelando um papel.	176
Figura 92 – Um agente de usuário se movendo de um ambiente para outro.	177
Figura 93 – A descrição textual da classe de ambiente Virtual Marketplace (parcial).	178
Figura 94 – A descrição textual dos relacionamentos da classe Virtual Marketplace.	178
Figura 95 – A classe orientada a objetos Virtual Marketplace e seus relacionamentos.	179
Figura 96 – A descrição textual da classe de organização General Store (parcial).	179
Figura 97 – O código Java para a classe General Store (parcial).	180
Figura 98 – Duas ações e um plano relacionado à classe General Store.	181
Figura 99 – A descrição textual dos relacionamentos da classe de organização	

General Store.	181
Figura 100 – A classe orientada a objetos General Store e seus relacionamentos.	182
Figura 101 – A descrição textual da classe User Agent (parcial).	182
Figura 102 – O código Java da classe User Agent (parcial).	183
Figura 103 – Duas ações e dois planos relacionados à classe User Agent.	184
Figura 104 – A descrição textual dos relacionamentos da classe User Agent.	184
Figura 105 – A classe User Agent orientada a objetos e seus relacionamentos.	185
Figura 106 – A descrição textual da classe Buyer (parcial).	185
Figura 107 – O código Java da classe Buyer (parcial).	186
Figura 108 – Um protocolo relacionado à classe Buyer.	186
Figura 109 – Os relacionamentos da classe Buyer.	187
Figura 110 – A classe orientada a objetos Buyer e seus relacionamentos.	187
Figura 111 – A descrição textual da classe do papel de objeto Desire (parcial).	188
Figura 112 – Os relacionamentos da classe Desire.	188
Figura 113 – A classe orientada a objetos Desire e seus relacionamentos.	188
Figura 114 – A descrição textual da classe Book (parcial).	189
Figura 115 – Os relacionamentos da classe Book.	189
Figura 116 – A classe orientada a objetos Book e seus relacionamentos.	189
Figura 117 – Um diagrama de classes de UML ilustrando todas as entidades definidas no exemplo.	191
Figura 118 – A ferramenta de MAS-ML.	206
Figura 119 – A gramática MAS-ML usada na fase de transformação das entidades.	226
Figura 120 – O sub-conjunto da gramática MAS-ML usada na fase de transformação dos relacionamentos.	231
Figura 121 – Descrição textual do exemplo do Virtual Marketplace.	252