

Maurício Serrano

Desenvolvimento Intencional de Software Transparente Baseado em Argumentação

Tese de Doutorado

Tese apresentada ao Programa de Pós-Graduação em Informática do Departamento de Informática da PUC-Rio como parte dos requisitos parciais para obtenção do título de Doutor em Informática.

Orientador: Prof. Julio Cesar Sampaio do Prado Leite

Rio de Janeiro

Agosto 2011

Maurício Serrano

Desenvolvimento Intencional de Software Transparente Baseado em Argumentação

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. Julio Cesar Sampaio do Prado Leite
Orientador
Departamento de Informática – PUC-Rio

Prof. Carlos José Pereira de Lucena
Departamento de Informática – PUC-Rio

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

Prof. Sérgio Donizetti Zorzo
Departamento de Computação – UFSCar

Prof(a). Claudia Cappelli Aló
Departamento de Informática Aplicada – UNIRIO

Prof. José Eugenio Leal
Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 23 de Agosto de 2011.

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

Maurício Serrano

Graduou-se em Engenharia de Computação (2001) e obteve o título de mestre em Ciência da Computação (2003) na Universidade Federal de São Carlos (UFSCar) – financiado pela CAPES (Bolsista por Mérito). Trabalhou no Projeto Genoma do Camarão (2003-2004) – financiado pela FAPESP. Durante seu Doutorado – financiado pela FAPESP/CAPES (Bolsista por Mérito) – na Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), sob a orientação do Professor Julio Cesar Sampaio do Prado Leite, teve um período de cooperação na University of Toronto (UofT), sob a orientação do Professor John Mylopoulos. Tem experiência em Engenharia de Software e Engenharia de Requisitos com foco, principalmente, nos tópicos: Orientação à Meta, Sistemas Multi-Agentes Intencionais e Transparência de Software.

Ficha Catalográfica

Serrano, Maurício

Desenvolvimento intencional de software transparente baseado em argumentação / Maurício Serrano; orientador: Julio Cesar Sampaio do Prado Leite. – 2011.

xviii., 283 f : il. ; 30 cm.

.....

Tese (Doutorado em Informática) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2011.

Inclui bibliografia.

1. Informática – Teses. 2. Transparência de software. 3. Requisitos de transparência. 4. Argumentação. 5. Validação relativa. 6. Rastreabilidade. 7. Máquina de raciocínio. 8. Modelos intencionais. 9. Sistemas multi-agentes intencionais. I. Leite, Julio Cesar Sampaio do Prado. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para Minha Amada Família:
Minha Esposa *Milene*,
Minha Mãe *Maria Elizabeth*,
Meu Pai *Edson (in memoriam)*,
Minhas Irmãs *Adriana* e *Mariana*,
Minha Sogra *Elizabeth*,
Meu Sogro *Antonio* e
Minha Cunhada *Gabriela*.

Agradecimentos

Meus mais especiais agradecimentos ao meu Orientador, Professor Julio Cesar Sampaio do Prado Leite, pela dedicação e pelo apoio ao longo de toda a pesquisa.

Meu muito obrigado a minha querida família pelo carinho, pelos conselhos e pela paciência.

Agradeço aos Professores John Mylopoulos e Eric Yu pela maravilhosa experiência na University of Toronto, Canadá.

Agradeço ainda ao Professor Carlos José Pereira de Lucena e à Professora Simone Diniz Junqueira Barbosa pelas reuniões e contribuições ao longo de toda a pesquisa.

Agradecimentos carinhos aos colegas do Grupo de Transparência de Software e do Laboratório de Engenharia de Software (LES) da PUC-Rio.

Meu muito obrigado ao corpo docente e aos funcionários da PUC-Rio pela atenção e pela prontidão em ajudar.

Meus sinceros agradecimentos aos membros da banca, Professores e Professoras, pela participação na defesa da tese de doutorado.

Finalmente, agradeço à agência de fomento CNPq pelo apoio financeiro.

Resumo

Serrano, Maurício; Leite, Julio Cesar Sampaio do Prado. **Desenvolvimento Intencional de Software Transparente Baseado em Argumentação**. Rio de Janeiro, 2011. 283p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Transparência é um critério de qualidade crítico para sociedades democráticas modernas. Como o software permeia a sociedade, a transparência se tornou uma preocupação para softwares operando em domínios públicos, sejam eles *eGovernment*, *eCommerce* ou softwares sociais. Dessa forma, a transparência de software está se tornando um critério de qualidade que demanda mais atenção dos desenvolvedores de software. Requisitos de transparência em um sistema de software estão relacionados a requisitos não-funcionais, como disponibilidade, usabilidade, informatividade, entendimento e auditabilidade. Entretanto, requisitos de transparência são especialmente difíceis de serem validados devido à natureza subjetiva dos conceitos envolvidos. Essa tese propõe o desenvolvimento intencional de software transparente dirigido por requisitos de transparência. Os requisitos de transparência são elicitados com o apoio de um catálogo de padrões de requisitos, relativamente validados pelos interessados através do uso de argumentação e representados em modelos intencionais. Modelos intencionais são fundamentais para a transparência de software, uma vez que associam aos requisitos as metas e os critérios de qualidade esperados pelos interessados e que justificam as decisões tomadas. Um sistema exemplo foi implementado como um sistema multi-agentes intencional, ou seja, com agentes colaborativos que implementam o modelo *Belief-Desire-Intention* e que são capazes de raciocinar sobre metas e critérios de qualidade. Essa tese discute as questões importantes para o sucesso da nossa abordagem de desenvolvimento de software transparente, como: (i) rastreabilidade requisitos-código e código-requisitos; (ii) o uso de lógica nebulosa para desenvolver uma máquina de raciocínio para agentes intencionais; (iii) a aplicação de argumentação para a validação relativa de requisitos de transparência através da obtenção de um consenso entre os interessados; e (iv) pré-rastreabilidade colaborativa para modelos intencionais baseada nas interações sociais. Nossas idéias foram validadas através de estudos de caso em diferentes domínios, tal como computação ubíqua e aplicações Web.

Palavras-chave

Transparência de Software; Requisitos de Transparência; Argumentação; Validação Relativa; Rastreabilidade; Máquina de Raciocínio; Modelos Intencionais; Sistemas Multi-Agentes Intencionais.

Abstract

Serrano, Maurício; Leite, Julio Cesar Sampaio do Prado. **Intentional Development of Transparent Software Based on Argumentation**. Rio de Janeiro, 2011. 283p. DSc Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Transparency is a critical quality *criterion* to modern democratic societies. As software permeates society, transparency has become a concern to public domain software, as *eGovernment*, *eCommerce* or social software. Therefore, software transparency is becoming a quality *criterion* that demands more attention from software developers. In particular, transparency requirements of a software system are related to non-functional requirements, e.g. availability, usability, informativeness, understandability and auditability. However, transparency requirements are particularly difficult to validate due to the subjective nature of the involved concepts. This thesis proposes a transparency-requirements-driven intentional development of transparent software. Transparency requirements are elicited with the support of a requirements patterns catalog, relatively validated by the stakeholders through argumentation and represented on intentional models. Intentional models are fundamental to software transparency, as they associate goals and quality criteria expected by the stakeholders with the software requirements. The goals and quality criteria also justify the decisions made during software development. A system was implemented as an intentional multi-agents system, i.e., a system with collaborative agents that implement the Belief-Desire-Intention model and that are capable of reasoning about goals and quality criteria. This thesis discusses important questions to the success of our approach to the development of transparent software, such as: (i) forward and backward traceability; (ii) a fuzzy-logic based reasoning engine for intentional agents; (iii) the application of an argumentation *framework* to relatively validate transparency requirements through stakeholders' multi-party agreement; and (iv) collaborative pre-traceability for intentional models based on social interactions. Our ideas were validated through case studies from different domains, such as ubiquitous computing and Web applications.

Keywords

Software Transparency; Transparency Requirements; Argumentation;
Relative Validation; Traceability; Reasoning Engine; Intentional Models;
Intentional Multi-Agent Systems.

Sumário

1. Introdução	20
1.1. Motivação	20
1.2. Caracterização do Problema	23
1.3. Enfoque da Solução	24
1.4. Organização da Tese	25
2. Conceitos e Apoio Tecnológico	26
2.1. Modelos Intencionais: Os <i>Frameworks</i> i^* e NFR	26
2.2. Transparência de Software	29
2.3. Sistemas Multi-Agentes Intencionais	30
2.4. O <i>Framework Acceptability Evaluation</i>	34
2.5. Considerações Finais	36
3. Anexação de Requisitos ao Código	38
3.1. Associações entre Abstrações dos Modelos i^* e BDI e Código JADEX	40
3.2. Heurísticas Transformacionais de Desenho e de Implementação	43
3.3. Geração dos Rastros a partir das Heurísticas Transformacionais	53
3.4. Trabalhos Relacionados	54
3.5. Considerações Finais	55
4. Raciocínio sobre Metas Flexíveis em Tempo de Execução	56
4.1. O Simulador de Propagação	59
4.2. Máquina de Raciocínio Qualitativa	65
4.3. Trabalhos Relacionados	83
4.4. Considerações Finais	87
5. Argumentação sobre Requisitos de Transparência	88
5.1. Representando Modelos Intencionais no <i>Framework ACE</i>	90
5.2. Condição de Aceitabilidade	96

5.3. Captura dos Argumentos	99
5.4. Validação Relativa dos Requisitos de Transparência	101
5.5. Apoio Ferramental	111
5.6. Trabalhos Relacionados	115
5.7. Considerações Finais	116
6. Pré-Rastreabilidade	118
6.1. Visão Geral do Modelo ITrace	121
6.2. Representação Gráfica dos Modelos ITrace	123
6.3. Pré-Rastreabilidade Colaborativa	130
6.4. Trabalhos Relacionados	132
6.5. Considerações Finais	132
7. Abordagem Integrada para o Desenvolvimento de Software Transparente	134
7.1. Abordagem Proposta para Desenvolvimento Intencional Baseada em Argumentação	135
7.2. Estudo de Caso: Aplicação <i>Web Lattes-Scholar</i>	138
7.2.1. Atividade Argumentar	139
7.2.2. Atividade Desenhar	146
7.2.3. Atividade Implementar	150
7.2.4. Atividades Armazenar e Consultar	155
7.3. Considerações Finais	156
8. Avaliação da Abordagem Integrada	158
8.1. Avaliação Quantitativa da Aplicação da Abordagem	160
8.2. Avaliação Qualitativa da Aplicação da Abordagem	166
8.3. Avaliação da Abordagem Integrada por Construção	169
8.4. Análise das Versões Anteriores do <i>Lattes-Scholar</i>	176
8.5. Considerações Finais	180
9. Conclusão	181
9.1. Contextualização	181
9.2. Resumo	182

9.3. Contribuições dos Trabalhos Pontuais	184
9.3.1. Anexação de Requisitos ao Código	184
9.3.2. Raciocínio sobre Metas Flexíveis em Tempo de Execução	185
9.3.3. Argumentação sobre Requisitos de Transparência	186
9.3.4. Pré-Rastreabilidade	187
9.4. Estudos de Caso para os Trabalhos Pontuais	189
9.5. Contribuições Relacionadas à Abordagem Integrada	190
9.6. Estudo de Caso para a Abordagem Integrada	191
9.7. Avaliação da Abordagem Integrada	194
9.8. Esclarecimento das Questões Levantadas na Introdução	195
9.9. Limitações	197
9.10. Trabalhos Futuros	198
Apêndice A. Exemplos de Uso do Lattes-Scholar	212
A.1 Processo Iniciado pelo Consumidor com o Nome do Pesquisador	212
A.2 Processo Iniciado pelo Consumidor com a URL do Currículo Lattes do Pesquisador	214
Apêndice B. Código-Fonte da Máquina de Raciocínio Nebulosa para Metas Flexíveis	217
B.1 Código-fonte da Classe MetaPlan	217
B.2 Código-fonte da Classe Softgoal	227
B.3 Código-fonte da Classe LeafSoftgoal	228
B.4 Código-fonte da Classe DecomposedSoftgoal	233
B.5 Código-fonte da Enumeração DecompositionType	235
B.6 Código-fonte da Classe ImpactSoftgoal	235
B.7 Código-fonte da Classe Task	237
B.8 Código-fonte da Classe Contribution	239
B.9 Código-fonte da Enumeração ContributionType	240
Apêndice C. Código-Fonte das Principais Situações do Estudo de Caso Lattes-Scholar	242
C.1 Situação “Buscar o Pesquisador”	242
C.1.1 Código-fonte do ADF do agente Buscador buscador.agent.xml	242

C.1.2	Código-fonte da Classe ExecuteRPRequest.java	245
C.1.3	Código-fonte do ADF da Capacidade Google	
	Google.capability.xml	248
C.1.4	Código-fonte da Classe BuscarPesquisadoresNoGoogle.java	250
C.2	Situação “Decidir entre Utilizar o Agente Fornecedor de Citações Local ou um Remoto”	256
C.2.1	Código-fonte do ADF do Agente Fornecedor de Citações	
	FornecedorDeCitações.agent.xml	256
C.2.2	Código-fonte da Classe SetupAgent.java	260
C.2.3	Código-fonte da Classe	
	RequisitarONumeroDeCitacoesParaOGoogleScholar.java	263
C.2.4	Código-fonte da Classe DelegarAMetaAUmAgenteRemoto.java	264
C.3	Código-fonte para a Situação “Obter as Citações da Publicação no Google Scholar”	266
C.3.1	Código-fonte do ADF do Agente Fornecedor de Citações	
	FornecedorDeCitacoes.agent.xml	267
C.3.2	Código-fonte da Classe ExecuteRPRequest.java	269
C.3.3	Código-fonte do ADF da Capacidade Scholar	
	Scholar.capability.xml	272
C.3.4	Código-fonte da Classe ObterAsCitacoesNoGoogleScholar.java	275
C.3.5	Código-fonte da Classe	
	RecuperarAPaginaDaPublicacaoNoGoogle Scholar.java	277
C.3.6	Código-fonte da Página lattesscholar.xhtml	281

Lista de Figuras

Figura 2.1 - Um modelo i^*	27
Figura 2.2 - Um exemplo de um SIG	28
Figura 2.3 - O SIG de Transparência, traduzido de (Leite e Cappelli 2010)	29
Figura 2.4 - Visão Geral da Arquitetura intra-agente utilizada no JADEX, extraída de (Pokahr e Braubach 2007)	33
Figura 2.5 - O schema XML de um agente Jadex, extraído de (JADEX 2007a)	33
Figura 2.6 - Exemplo de um plano que escolhe um movimento entre vários possíveis, extraído de (JADEX 2007a)	34
Figura 2.7 - Um exemplo da notação gráfica da linguagem ACE	36
Figura 3.1 - Relações entre as abstrações do <i>framework</i> i^* e do modelo BDI	41
Figura 3.2 - Relações entre as abstrações do modelo BDI e código JADEX	43
Figura 3.3 - Aplicação das heurísticas de desenho no modelo arquitetural parcial do Lattes-Scholar	48
Figura 3.4 - Trechos de código que podem ser obtidos a partir dos cenários	52
Figura 4.1 - Os graus de satisfação das metas flexíveis vistos como uma variável nebulosa.	61
Figura 4.2 - Duas tarefas contribuindo para uma meta	62
Figura 4.3 - Habilitando o agente com a capacidade da máquina de raciocínio qualitativa.	67
Figura 4.4 - A variável nebulosa (domínio e conjuntos nebulosos) de uma meta flexível	67
Figura 4.5 - Código parcial para a classe “Task”	68
Figura 4.6 - Código parcial para a classe “Contribution”	68
Figura 4.7 - Código de geração de regras nebulosas para o tipo de contribuição “help”.	68
Figura 4.8 - Metas flexíveis, contribuições e tarefas como crenças dos agentes	69
Figura 4.9 - Inicialização de uma tarefa e sua contribuição no <i>setup</i> do agente	69
Figura 4.10 - Código parcial da classe abstrata “Softgoal”	69
Figura 4.11 - Estrutura meios-fim do Lattes-Scholar	72
Figura 4.12 - Estrutura meios-fim do Sistema de Iluminação Inteligente	83

Figura 5.1 - As quatro estruturas i^* recorrentes representadas como proposições na linguagem ACE	92
Figura 5.2 - Padrão de requisitos “Objetivos de Transparência de Software”, extraído do CTS (CTS 2011).	94
Figura 5.3 - As três estruturas SIGs recorrentes representadas como proposições na linguagem ACE	95
Figura 5.4 - Os <i>late-requirements</i> do Lattes-Scholar modelados no <i>framework</i> i^*	97
Figura 5.5 - A estrutura i^* meios-fim da Figura 5.4 traduzida para a linguagem ACE	98
Figura 5.6 - A evolução do Grafo D de D(0) até D(1)	100
Figura 5.7 - Grafo resultante D(2)=D(3)=D(N)	101
Figura 5.8 - Um SIG parcial de Transparência	102
Figura 5.9 - Um conflito (C1) entre o RNF Validade ($i(nfr9)$) e a tarefa $t1$ ($i(t1)$)	103
Figura 5.10 - O argumento $p2$ de um interessado a favor de uma estrutura meios-fim alternativa	104
Figura 5.11 - Argumentos ($p3$ e $p4$) de dois interessados	105
Figura 5.12 - A regra de preferência ilustrando a preferência pela proposição $p4$	106
Figura 5.13 - A estrutura i^* meios-fim evoluída	106
Figura 5.14 - Um conflito (C8) entre o RNF Clareza ($i(nfr7)$) e a estrutura i^* meios-fim (I_{ME3})	108
Figura 5.15 - Um argumento $p7$ de um interessado a favor da substituição da estrutura meios-fim I_{ME3} por uma estrutura meios-fim do tipo Ou-exclusivo	110
Figura 5.16 - A regra de preferência ilustrando a preferência pela proposição $p7$	110
Figura 5.17 - A estrutura i^* meios-fim evoluída	111
Figura 5.18 - O grafo ACE completo para o primeiro estudo de caso	113
Figura 5.19 - O grafo do léxico do SIG de Transparência	114
Figura 5.20 - O símbolo “iv-nfr8” enriquecido com elos para outros símbolos do LEL	114
Figura 6.1 - Estrutura em camadas do ITrace	124

Figura 6.2 - Etnografia aplicada nas gravações audiovisuais de uma reunião	125
Figura 6.3 - Granularidade grossa para evolução de artefatos	125
Figura 6.4 - Modelo ITrace da decomposição do RNF Acurácia	127
Figura 6.5 - ITrace de um processo de avaliação de equipes de engenheiros de requisitos	129
Figura 6.6 - Explorando a funcionalidade de hipermídia do Flickr® (Flickr 2011b)	131
Figura 7.1 - O processo de desenvolvimento proposto modelado em SADT	137
Figura 7.2 - <i>Storyboard</i> produzido com quadros dos vídeos da segunda reunião	140
Figura 7.3 - O grafo ACE que representa a discussão sobre a disponibilização do Lattes-Scholar como uma aplicação <i>Web</i>	142
Figura 7.4 - Modelo ITrace da primeira reunião	143
Figura 7.5 - Modelo de Atores Estratégicos do Lattes-Scholar	144
Figura 7.6 - Modelo de Dependências Estratégicas do Lattes-Scholar para a situação “Recuperar citações”	145
Figura 7.7 - Modelo arquitetural em i* do SMA do Lattes-Scholar para a situação “Recuperar Citações”	147
Figura 7.8 - Modelo de <i>Rationale</i> Estratégico parcial do papel “Fornecedor de Citações” na situação “Recuperar Citações”	148
Figura 7.9 - A tarefa “Requisitar o número de citações para o Google Scholar” detalhada como um cenário	149
Figura 7.10 - A especificação BDI do papel “Fornecedor de Citações” como um agente não executável	150
Figura 7.11 - ADF da capacidade “FornecedorDeCitacoes”, produzido pelas heurísticas transformacionais de implementação	151
Figura 7.12 - Crenças definidas na capacidade “FornecedorDeCitacoes”	152
Figura 7.13 - As metas da capacidade “FornecedorDeCitacoes”	153
Figura 7.14 - Os planos da capacidade “FornecedorDeCitacoes”	154
Figura 7.15 - O plano “RequisitarONumeroDeCitacoesParaOGoogle Scholar” anotado com cenários	154
Figura 7.16 - Implementando os episódios entre os comentários	155
Figura 7.17 - A estrutura hierárquica/cronológica utilizada no Wiki do projeto	156

Lista de Tabelas

Tabela 2.1 - Operacionalizações e questões para a meta flexível folha Validade	30
Tabela 3.1 - Heurísticas transformacionais de desenho: do <i>framework i*</i> para a especificação BDI	44
Tabela 3.2 - Heurísticas transformacionais de implementação: da especificação BDI para código JADEX	49
Tabela 4.1 - Regras de propagação do <i>framework i*</i> (ISTARWIKI 2011).	62
Tabela 4.2 - Situação inicial das metas flexíveis	73
Tabela 4.3 - Impactos atribuídos às tarefas na análise da primeira alternativa	73
Tabela 4.4 - Regras nebulosas executadas na análise da primeira alternativa	74
Tabela 4.5 - Impactos atribuídos às metas flexíveis caso a primeira alternativa seja selecionada	75
Tabela 4.6 - Impactos atribuídos às tarefas na análise da segunda alternativa	75
Tabela 4.7 - Regras nebulosas executadas na análise da segunda alternativa	76
Tabela 4.8 - Impactos atribuídos às metas flexíveis caso a primeira alternativa seja selecionada	76
Tabela 4.9 – Situação atual das metas flexíveis	77
Tabela 4.10 - Impactos atribuídos às tarefas na análise da primeira alternativa	78
Tabela 4.11 - Regras nebulosas executadas na análise da primeira alternativa	79
Tabela 4.12 – Conjunto de impactos resultante da seleção da primeira alternativa	80
Tabela 4.13 - Impactos atribuídos às tarefas na análise da segunda alternativa	80
Tabela 4.14 - Regras nebulosas executadas na análise da segunda alternativa	81
Tabela 4.15 – Conjunto de impactos resultante da seleção da segunda alternativa	82
Tabela 5.1 - A informação do símbolo “iv-p4”	112
Tabela 5.2 - A informação do símbolo “ir-t4”	112
Tabela 5.3 - A informação do símbolo “ir-me3”	112

Lista de Abreviaturas e Siglas

AC - **A**ceptability **C**ondition
ACE - **A**ceptability **E**valuation
AIF - **A**rgument **I**nterchange **F**ormat
API - **A**pplication **P**rogram **I**nterface
ADF - **A**gent **D**efinition **F**ile
BDI - **B**elief **D**esire **I**ntention
C&L - **C**enários e **L**éxico
CASE - **C**omputer-**A**ided **S**oftware **E**ngineering
CTS - **C**atálogo de **T**ransparência de **S**oftware
DEP - **D**ependency
ER - **E**ngenharia de **R**equisitos
FIPA - **F**oundation for **I**ntelligent **P**hysical **A**gents
GORE - **G**oal-**O**riented **R**equirements **E**ngineering
GQM - **G**oal-**Q**uestion-**M**etric
GQO - **G**oal-**Q**uestion-**O**perationalization
i* - iStar, iEstrela ou Intencionalidade Distribuída
IDE - **I**ntegrated **D**evelopment **E**nvironment
JADE - **J**ava **A**gent **D**evelopment **E**nvironment
JADEX - **J**ava **A**gent **D**evelopment **E**nvironment **eX**tension
LEL - **L**éxico **E**stendido da **L**inguagem
ME - **M**eans-**E**nd
NFR - **N**on-**F**unctional **R**equirements
OME - **O**rganization **M**odelling **E**nvironment
OP - **O**perationalization
RNF - **R**equisito **N**ão-**F**uncional
SA - **S**trategic **A**ctors
SADT - **S**tructured **A**nalysis and **D**esign **T**echnique
SD - **S**trategic **D**ependency
SIG - **S**oftgoals **I**nterdependency **G**raph

SMA - **S**istema **M**ulti-**A**gentes

SR - **S**trategic **R**ationale

TROPOS - from ***trepō*** in Greek, which means "turning or adopting a new manner"

UML - **U**nified **M**odeling **L**anguage

URL - **U**niform **R**esource **L**ocation

XML - **eX**tensible **M**arkup **L**anguage