

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CURSO DE MESTRADO**

RODRIGO PEROZZO NOLL

**RASTREABILIDADE ONTOLÓGICA
SOBRE O PROCESSO UNIFICADO**

Dissertação apresentada como requisito parcial para obtenção do grau de mestre, pelo Programa de Pós-Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Marcelo Blois Ribeiro.

PORTO ALEGRE

Janeiro, 2007



Dados Internacionais de Catalogação na Publicação (CIP)

N793r Noll, Rodrigo Perozzo

Rastreabilidade ontológica sobre o processo unificado /
Rodrigo Perozzo Noll. – Porto Alegre, 2007.

132 f.

Diss. (Mestrado em Ciência da Computação) – Fac.
de Informática, PUCRS.

Orientação: Prof. Dr. Marcelo Blois Ribeiro.

1. Informática. 2. Modelagens de Sistemas. 3.
Ontologia. 4. Sistemas de Informação. I. Ribeiro, Marcelo
Blois.

CDD 004.2

**Ficha Catalográfica elaborada pelo
Setor de Processamento Técnico da BC-PUCRS**

PUCRS

Campus Central
Av. Ipiranga, 6681 - prédio 16 - CEP 90619-900
Porto Alegre - RS - Brasil
Fone: +55 (51) 3320-3544 - Fax: +55 (51) 3320-3548
Email: bceadm@pucrs.br
www.pucrs.br/biblioteca



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Rastreabilidade Ontológica Sobre o Processo Unificado**", apresentada por Rodrigo Perozzo Noll, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 22/01/2007 pela Comissão Examinadora:


Prof. Dr. Marcelo Blois Ribeiro - PUCRS
Orientador (a)


Prof. Dr. Ricardo Melo Bastos - PUCRS


Prof. Dr. Ricardo Choren Noya - IME


Prof. Dr. Guilherme Horta Travassos - UFRJ

Homologada em 04/06/2007, conforme Ata No. 013/2007 pela Comissão Coordenadora.


Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P. 16 - sala 106 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@inf.pucrs.br

www.pucrs.br/facinf/pos

Para meus pais, Carlos e Sônia, pelo exemplo de vida e referênciã de caráter e honra e para minha esposa, Sandra, pelo apoio e incentivo em todos os momentos.

AGRADECIMENTOS

Aos meus pais, Carlos e Sônia, pela educação e exemplo de vida. Sua referência foi fundamental para estruturação de meu ser e meu caráter.

A minha esposa, Sandra, pela constante participação e carinho em cada detalhe de minha vida.

Ao meu orientador, Professor Marcelo Blois Ribeiro, pelo incentivo e compreensão fundamentais à realização deste trabalho e por suas valiosas críticas, que muito contribuíram para o meu crescimento pessoal.

Aos Professores Ricardo de Melo Bastos, Vera Lúcia Strube de Lima, Guilherme Horta Travassos e Toacy Cavalcante de Oliveira pelas importantes contribuições e sugestões necessárias para este trabalho.

Aos membros da banca pela aceitação do convite de participação na avaliação deste trabalho.

A todos do grupo de pesquisa ISEG e aos participantes do experimento por seu apoio, disponibilidade e interesse.

Aos meus colegas do CDPe e ao convênio Dell-PUCRS, por viabilizar a bolsa de estudos necessária para a realização deste trabalho.

A todos os professores e funcionários do Programa de Pós-Graduação em Ciência da Computação da PUCRS.

RESUMO

A rastreabilidade se refere à habilidade de relacionar informações em uma cadeia de processo. Este trabalho apresenta uma proposta de rastreabilidade pela integração de ontologias no Processo Unificado [JAC99]. Sugere-se a criação de uma ontologia a partir da modelagem de domínio, durante as etapas iniciais do Processo Unificado. Com o desenvolvimento desta ontologia, viabiliza-se a rastreabilidade baseada em conceitos durante o ciclo de vida do software. Esta abordagem permite a integração de diferentes modelos de um sistema de informação, incluindo negócios, requisitos, análise e projeto, em uma granularidade mais específica do que as convencionais abordagens de rastreabilidade baseadas em requisitos. Para apoiar os projetistas na criação da ontologia e no relacionamento dos artefatos, foi desenvolvida uma ferramenta para rastreabilidade semântica chamada ONTrace (*Ontological Tracing*), comprovando a viabilidade da proposta. Para a avaliação da proposta, foi desenvolvido um experimento para caracterizar a precisão e o esforço relacionados com a rastreabilidade apoiada por conceitos e por requisitos.

Palavras-Chave: Rastreabilidade, Modelagem do Conhecimento, Ontologia, Processo Unificado.

ABSTRACT

Traceability refers to the ability to link information in a process chain. This work presents a traceability proposal by the integration of ontologies into the Unified Process [JAC99]. The ontology was created through the domain modeling, proposed during the earlier phases of the Unified Process. Using this ontology, a concept-based traceability is provided throughout the software lifecycle. This approach allows the integration of different models of a software system including business, requirements, analysis and design models in a lower granularity degree than conventional requirements traceability approaches. To assist the designers in creating the ontology and linking concepts to artifacts it was developed a tool for semantic traceability called ONTrace (*Ontological Tracing*), enhancing the proposal's viability. To evaluate the process, an experiment was carried out to characterize the precision and effort related to the concept-based and requirement-based traceability.

Keywords: Traceability, Knowledge Modeling, Ontology, Unified Process.

LISTA DE FIGURAS

Figura 3.1 – Disciplinas, fases e iterações do Processo Unificado.....	36
Figura 3.2 – <i>Workflow</i> da Disciplina de Modelagem do Conhecimento.	39
Figura 3.3 – Extensão ao Processo Unificado.	40
Figura 3.4 – Cenário de estrutura UML.	42
Figura 3.5 – Processos da atividade de projeto.	43
Figura 3.6 – Processos da atividade de manutenção.....	45
Figura 4.1 – Rastreabilidade em Artefatos de Software.	55
Figura 4.2 – Estrutura Ontológica para Rastreabilidade.	58
Figura 4.3 – Primeiro cenário do exemplo de rastreabilidade ontológica.	59
Figura 4.4 – Segundo cenário do exemplo de rastreabilidade ontológica.	60
Figura 4.5 – Diagrama de Atividades para inferência.	64
Figura 4.6 – Interface da ferramenta para análise de viabilidade.....	64
Figura 4.7 – Interface da ferramenta ArgoUML+ONTrace.....	65
Figura 4.8 – Diagrama de Classes com as novas funcionalidades.	66
Figura 4.9 – Interface para geração da ontologia.	66
Figura 4.10 – Diagrama de colaboração para geração da ontologia.....	67
Figura 4.11 – Interface para Importação e Exportação da Ontologia.....	67
Figura 4.12 – Criação dos elos de rastreabilidade.	68
Figura 4.13 – Criação dos elos de rastreabilidade.	68
Figura 4.14 – Interface para recuperação de elos de rastreabilidade.....	69
Figura 4.15 – Diagrama de colaboração para recuperação de elos de rastreabilidade.	69
Figura 4.16 – Modelo de Domínio do sistema <i>isGym</i>	70
Figura 4.17 – Diagrama de casos de uso do sistema <i>isGym</i>	71
Figura 5.1 – Variáveis independentes e dependentes do estudo experimental.	88
Figura 5.2 – Gráfico de barras relativo ao esforço para definição dos elos de rastreabilidade.....	95
Figura 5.3 – Gráfico de barras relativo à precisão na recuperação dos elos de rastreabilidade.	96
Figura 5.4 – Gráfico de dispersão para a variável <i>esforço</i>	97
Figura 5.5 – Gráfico de dispersão para a variável <i>precisão</i>	100

LISTA DE TABELAS

Tabela 5.1 –Tabela de contingência.	90
Tabela 5.2 – Tabulação dos valores brutos obtidos após a execução do experimento.....	95
Tabela 5.3 – Teste de normalidade <i>Shapiro-Wilk</i> para a variável <i>esforço</i>	97
Tabela 5.4 – Teste de Levene para igualdade das variâncias sobre a variável <i>esforço</i>	98
Tabela 5.5 – Teste T para duas amostras independentes para a variável <i>esforço</i> , agrupada por <i>conceito e requisito</i>	99
Tabela 5.6 – Teste T para duas amostras independentes para a variável <i>esforço</i> , agrupada por <i>conceito e requisito</i>	99
Tabela 5.7 – Média e desvio padrão para a variável <i>precisão</i>	100
Tabela 5.8 – Teste de normalidade <i>Shapiro-Wilk</i> para a variável <i>precisão</i>	101
Tabela 5.9 – Teste não paramétrico de Mann-Whitney para a variável <i>precisão</i>	101
Tabela 5.10 – Estatística descritiva para a variável <i>precisão</i>	102
Tabela 5.11 – Resultados da avaliação qualitativa.	102
Tabela 5.12 – Média da satisfação das questões sobre as abordagens.....	103

LISTA DE QUADROS

Quadro 3.1 - Notação utilizada para definição do processo.....	38
Quadro 3.2 – Mapeamento entre estruturas UML e OWL.....	42
Quadro 4.1 – Dimensões de Rastreabilidade.....	56
Quadro 5.1 – Escalas das variáveis.....	94

LISTA DE ABREVIATURAS

CDPe	Centro de Desenvolvimento e Pesquisa.....	05
ODA	Ontology Driven Architectures.....	14
UML	Unified Modeling Language.....	15
MD	Modelo de Domínio.....	17
ODM	Ontology Definition Metamodel.....	17
OWL	Web Ontology Language.....	17
RDF	Resource Description Framework.....	21
OO	Orientado a Objetos.....	34
OMG	Object Management Group.....	38
SPEM	Software Process Engineering Metamodel.....	38
MDA	Model Driven Architectures.....	48
OWG	Ontology Working Group.....	42
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul.....	62
GQM	Goal Question Metric.....	76

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	QUESTÃO DE PESQUISA.....	15
1.2	OBJETIVOS.....	16
1.2.1	<i>Objetivo Geral.....</i>	<i>16</i>
1.2.2	<i>Objetivos Específicos.....</i>	<i>16</i>
1.3	PRINCIPAIS CONTRIBUIÇÕES.....	16
1.4	ESTRUTURA DO DOCUMENTO.....	17
2	ONTOLOGIAS.....	19
2.1	DEFINIÇÕES DE ONTOLOGIA.....	19
2.2	WEB ONTOLOGY LANGUAGE (OWL).....	20
2.3	CONSTRUÇÃO DE ONTOLOGIAS.....	24
2.3.1	<i>Metodologia de Uschold & King.....</i>	<i>24</i>
2.3.2	<i>Metodologia de Grüninger e Fox.....</i>	<i>25</i>
2.3.3	<i>Framework Methontology.....</i>	<i>26</i>
2.3.4	<i>Abordagem de Falbo e Rocha.....</i>	<i>27</i>
2.3.5	<i>Guia de Noy e McGuinness.....</i>	<i>28</i>
2.3.6	<i>Abordagem colaborativa de Holsapple e Joshi.....</i>	<i>29</i>
2.3.7	<i>Metodologia On-To-Knowledge.....</i>	<i>29</i>
2.3.8	<i>Knowledge Unified Process.....</i>	<i>30</i>
2.3.9	<i>Processo Helix-Spindle.....</i>	<i>31</i>
2.4	CONSIDERAÇÕES.....	32
3	INTEGRAÇÃO DE ONTOLOGIAS NO PROCESSO UNIFICADO.....	34
3.1	CONSTRUÇÃO DE ONTOLOGIAS E ENGENHARIA DE SOFTWARE.....	34
3.2	ONTOLOGIAS E O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.....	35
3.3	PROPOSTA DE INTEGRAÇÃO DE ONTOLOGIAS NO PROCESSO UNIFICADO.....	37
3.3.1	<i>Disciplina de Modelagem do Conhecimento.....</i>	<i>37</i>
3.3.1.1	<i>Projeto.....</i>	<i>40</i>
3.3.1.2	<i>Manutenção.....</i>	<i>44</i>
3.3.1.3	<i>Avaliação.....</i>	<i>45</i>
3.4	ANÁLISE DA METODOLOGIA SEGUNDO [FER99].....	46
3.5	TRABALHOS RELACIONADOS.....	48
3.5.1	<i>Estação TABA.....</i>	<i>48</i>
3.5.2	<i>ODE: Ontology-based software Development Environment.....</i>	<i>49</i>
3.5.3	<i>Proposta de Oberle.....</i>	<i>50</i>
3.5.4	<i>Ferramentas para desenvolvimento integrado a ontologias.....</i>	<i>51</i>
3.6	CONSIDERAÇÕES.....	51
4	RASTREABILIDADE ONTOLÓGICA.....	53
4.1	RASTREABILIDADE.....	53
4.1.1	<i>Dimensões de Rastreabilidade.....</i>	<i>54</i>
4.1.1.1	<i>Rastreabilidade Vertical e Horizontal.....</i>	<i>54</i>
4.1.1.2	<i>Pré-Rastreabilidade e Pós-Rastreabilidade.....</i>	<i>55</i>
4.1.1.3	<i>Relacionamentos Explícitos e Implícitos.....</i>	<i>55</i>
4.1.1.4	<i>Relacionamentos Estruturais e Cognitivos.....</i>	<i>55</i>
4.1.1.5	<i>Modelagem da Rastreabilidade.....</i>	<i>56</i>
4.2	RASTREABILIDADE ONTOLÓGICA.....	57
4.3	AValiação DAS DIMENSÕES DE RASTREABILIDADE ONTOLÓGICA.....	60
4.4	FERRAMENTA PARA RASTREABILIDADE ONTOLÓGICA.....	62
4.4.1	<i>Análise de Viabilidade Inicial.....</i>	<i>62</i>
4.4.2	<i>ArgoUML+ONTrace.....</i>	<i>65</i>
4.4.3	<i>Exemplo de aplicação da ferramenta.....</i>	<i>70</i>
4.5	CONSIDERAÇÕES.....	71
5	AVAlIAÇÃO DA PROPOSTA.....	73
5.1	REFERENCIAL TEÓRICO: ENGENHARIA DE SOFTWARE EXPERIMENTAL.....	74
5.1.1	<i>Definição.....</i>	<i>76</i>
5.1.2	<i>Planejamento.....</i>	<i>76</i>
5.1.2.1	<i>Seleção do Contexto.....</i>	<i>77</i>
5.1.2.2	<i>Formulação das Hipóteses.....</i>	<i>77</i>
5.1.2.3	<i>Seleção das variáveis.....</i>	<i>77</i>

5.1.2.4	Seleção dos Indivíduos.....	78
5.1.2.5	Projeto do experimento	78
5.1.2.5.1	Princípios genéricos de projeto	78
5.1.2.5.2	Padrão para tipo de projeto	79
5.1.2.6	Instrumentação.....	79
5.1.2.7	Análise da Validade	80
5.1.3	<i>Execução</i>	80
5.1.3.1	Preparação.....	81
5.1.3.2	Execução	81
5.1.3.3	Validação dos dados.....	81
5.1.4	<i>Análise e Interpretação</i>	81
5.1.5	<i>Empacotamento</i>	83
5.2	ESTUDO EXPERIMENTAL	84
5.2.1	<i>Definição</i>	84
5.2.1.1	Objetivo Global.....	84
5.2.1.2	Objetivo do estudo	84
5.2.1.3	Objetivo da medição	84
5.2.1.4	Questões.....	85
5.2.1.5	Métricas.....	85
5.2.2	<i>Planejamento</i>	86
5.2.2.1	Seleção do Contexto.....	86
5.2.2.2	Formulação das Hipóteses.....	86
5.2.2.3	Seleção das variáveis.....	88
5.2.2.3.1	Variáveis Independentes	88
5.2.2.3.2	Variáveis Dependentes.....	88
5.2.2.4	Seleção dos Indivíduos.....	89
5.2.2.5	Projeto do experimento	89
5.2.2.6	Padrão para tipo de projeto.....	89
5.2.2.7	Instrumentação.....	91
5.2.2.8	Análise da Validade	91
5.2.2.8.1	Validade interna.....	91
5.2.2.8.2	Validade externa	92
5.2.2.8.3	Validade de construção	92
5.2.2.8.4	Validade da conclusão	92
5.2.3	<i>Execução</i>	93
5.2.3.1	Preparação.....	93
5.2.3.2	Execução	94
5.2.4	<i>Análise e Interpretação</i>	94
5.2.4.1	Análise Tabular e Gráfica	95
5.2.4.2	Estatística Descritiva	96
5.2.4.3	Primeira Hipótese: Esforço	96
5.2.4.4	Segunda Hipótese: Precisão	100
5.3	AVALIAÇÃO QUALITATIVA	102
5.4	CONSIDERAÇÕES	103
6	CONCLUSÕES	105
	REFERÊNCIAS	107
	APÊNDICE A – TREINAMENTO PARA OS PARTICIPANTES DO EXPERIMENTO	114
	APÊNDICE B – TUTORIAL PARA RASTREABILIDADE INDEXADA POR CONCEITOS	127
	APÊNDICE C – TUTORIAL PARA RASTREABILIDADE INDEXADA POR REQUISITOS	128
	APÊNDICE D – PLANILHA DE RASTREABILIDADE ENTRE CLASSES DO PROJETO INDEXADAS POR CONCEITOS E REQUISITOS.....	129
	APÊNDICE E – AVALIAÇÃO QUALITATIVA DAS ABORDAGENS DE RASTREABILIDADE.....	130
	ANEXO A – TABELA DE COMPONENTES DA UML.....	131
	ANEXO B – TABELA DE DISTRIBUIÇÃO DE T	132

1 INTRODUÇÃO

O processo de desenvolvimento de software compreende a criação de diferentes modelos. Modelos representam uma abstração da realidade e podem ser utilizados para visualizar e controlar uma arquitetura, avaliando os riscos e identificando oportunidades. Durante a Engenharia de Software, modelos são desenvolvidos em diferentes perspectivas e níveis de abstrações. Estas perspectivas nem sempre são integradas, porém tendem a ser complementares, variando de acordo com sua especificidade [RUM98].

Cada modelo está relacionado a conceitos do mundo real. Estes conceitos podem ser representados por uma ontologia, isto é, um modelo que especifica os termos que descrevem e representam o conhecimento sobre um domínio de informação. Este trabalho sugere o uso de ontologias para apoiar o processo de desenvolvimento de software, particularmente o Processo Unificado. O objetivo é integrar as diferentes visões que modelam o sistema, relativas a negócios, requisitos e análise e projeto. Utilizando ontologias, permite-se relacionar estas visões, mapeando-as com os conceitos que representam o domínio.

Muito do esforço relativo à construção de ontologias corresponde à delimitação do escopo, explicitação do domínio e modelagem conceitual. Durante as etapas iniciais do Processo Unificado, os esforços correspondem a aspectos relacionados, como a modelagem de negócio e de domínio. É com base nesta premissa que se sugere a integração de ontologias no Processo Unificado, aproveitando o esforço despendido durante a engenharia de software para a construção da ontologia. Com esta integração, consegue-se explicitar os conceitos do domínio em um formalismo lógico, não previsto pelo Processo Unificado.

Neste contexto, será apresentada uma abordagem sistemática para a engenharia de ontologias, compilando as boas práticas encontradas na literatura e as integrando no Processo Unificado.

De acordo com o *Ontology Driven Architecture* [ODA06], definido pela [W3C06], alguns dos benefícios obtidos pela integração de ontologias nas práticas de Engenharia de Software são:

- Apoiar à modelagem de domínio e sua integração em diferentes fases do ciclo de vida do software;
- Reduzir as inconsistências entre os diferentes artefatos gerados no processo de desenvolvimento, devido principalmente as múltiplas perspectivas dos *stakeholders*;

- Apoiar a rastreabilidade entre estes artefatos, visando à melhoria da qualidade dos sistemas de informação através da análise de impacto, manutenções e evoluções, reuso de componentes, inspeção e compreensão do sistema.

Com o desenvolvimento de uma ontologia que represente o modelo de domínio de um sistema, possibilita-se a *rastreabilidade ontológica* (ou semântica), isto é, a possibilidade de relacionar os conceitos da ontologia com os artefatos gerados no processo de desenvolvimento.

Para comprovar a viabilidade da proposta, foi projetada e desenvolvida uma funcionalidade integrada a uma ferramenta de modelagem. Esta funcionalidade permite a geração de uma ontologia a partir da modelagem de domínio, definida através de um diagrama de classes UML [UML06]. Além disso, esta funcionalidade permite a recuperação dos elos de rastreabilidade definidos, utilizando um motor de inferência.

Para avaliar os benefícios da rastreabilidade ontológica e sua capacidade de indexar os artefatos, foi desenvolvido um experimento. O objetivo deste estudo é caracterizar relações de precisão e esforço entre a rastreabilidade indexada por conceitos da ontologia e a tradicional rastreabilidade indexada por requisitos.

1.1 QUESTÃO DE PESQUISA

Existem diversas propostas para construção de ontologias apresentadas na literatura, integradas ou não ao processo de desenvolvimento de software. A criação de uma ontologia específica, que represente o modelo conceitual de um sistema de informação, pode proporcionar ganhos efetivos durante o desenvolvimento de software, integrando as diferentes visões que modelam um sistema. Neste sentido, emerge a questão de pesquisa deste estudo: “É viável e interessante a rastreabilidade ontológica dos artefatos envolvidos com o processo de desenvolvimento de software através do mapeamento do conhecimento em uma ontologia?”.

1.2 OBJETIVOS

Uma vez definida a questão de pesquisa, definiu-se o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Prover a rastreabilidade ontológica entre os artefatos produzidos durante o ciclo de vida do software através da integração de ontologias no Processo Unificado.

1.2.2 Objetivos Específicos

Os objetivos definidos neste trabalho incluem:

- Avaliar as propostas de engenharia de ontologias encontradas na literatura, visando identificar padrões e melhores práticas;
- Integrar os padrões e as boas práticas identificados anteriormente no Processo Unificado. Esta integração visa estabelecer a sistemática para construção de uma ontologia coerente com o domínio da aplicação;
- Avaliar algumas propostas de classificação dos elos de rastreabilidade;
- Propor uma estrutura ontológica que permita à geração e recuperação dos elos de rastreabilidade;
- Avaliar a viabilidade da proposta apresentada;
- Comparar, em um estudo experimental, a proposta de rastreabilidade ontológica com relação a tradicional proposta de rastreabilidade indexada por requisitos.

1.3 PRINCIPAIS CONTRIBUIÇÕES

Para possibilitar a rastreabilidade ontológica sobre o Processo Unificado, inicialmente foi realizada uma análise de padrões e boas práticas de nove propostas de engenharia de ontologias. O objetivo foi proporcionar o embasamento necessário para construção de ontologias que mapeiem corretamente o domínio da aplicação. A Seção 2.4 apresenta a **análise das propostas** estudadas, compilando-as em uma seqüência de passos.

A partir destes passos, a Seção 3.3 apresenta uma proposta de **integração de ontologias no Processo Unificado**, sugerindo a criação de uma nova disciplina, chamada de Modelagem de Ontologia. A disciplina sugerida é definida em três principais atividades. Como resultado de uma destas atividades, obtém-se um artefato que corresponde à ontologia de domínio. Esta ontologia é gerada a partir do mapeamento de um artefato chamado Modelo de Domínio (MD), proposto pelo Processo Unificado, com base nas guias definidas pela [ODM06] para conversão de modelo UML em modelo ontológico.

Possuindo uma ontologia integrada ao processo de desenvolvimento, viabiliza-se a rastreabilidade semântica. A Seção 4.2 apresenta a estrutura que possibilita a **rastreabilidade ontológica** entre os artefatos envolvidos com o processo de desenvolvimento.

Para verificar a viabilidade da proposta, o Capítulo 5 apresenta a **implementação da ferramenta** que mapeia o Modelo de Domínio em uma ontologia e permite a criação dos elos de rastreabilidade, mapeando os conceitos da ontologia com os elementos de um modelo UML. Esta ferramenta também proporciona a aquisição dos elos de rastreabilidade previamente estabelecidos, utilizando um motor de inferência.

Para avaliação da rastreabilidade, a Seção 5.2 apresenta um experimento que objetiva **comparar a precisão e o esforço da rastreabilidade** indexada por conceitos da ontologia com a tradicional proposta de rastreabilidade indexada por requisitos.

1.4 ESTRUTURA DO DOCUMENTO

Este trabalho está estruturado em cinco partes:

1. Fundamentação teórica sobre ontologias e suas engenharias;
2. Proposta de integração de ontologias ao Processo Unificado;
3. Proposta de rastreabilidade ontológica;
4. Análise de viabilidade de ambas as propostas;
5. Estudo experimental sobre a rastreabilidade ontológica.

O Capítulo 2 apresenta a fundamentação teórica sobre ontologias, necessárias ao entendimento do trabalho. Nele, são apresentadas algumas definições de ontologias e a estrutura da linguagem de representação do conhecimento OWL (*Web Ontology Language*), sugerida pela [W3C06]. Além disso, são apresentadas algumas propostas para engenharia de ontologias.

O Capítulo 3 apresenta a proposta de integração de ontologias no Processo Unificado. Inicialmente, a Engenharia de Software é relacionada com a engenharia de ontologia, com o objetivo de identificar oportunidades. Posteriormente, sugere-se uma nova disciplina no Processo Unificado visando esta integração. Em seguida, é realizado um estudo comparando a disciplina sugerida com a proposta de [FER99], para avaliação de metodologias de engenharia de ontologias. Por fim, são apresentados alguns trabalhos relacionados.

O Capítulo 4 apresenta a proposta de rastreabilidade. Inicialmente, é apresentado o referencial teórico, incluindo algumas dimensões para caracterizar a rastreabilidade. Em seguida, é modelada a estrutura que possibilita a rastreabilidade ontológica. Por fim, caracteriza-se esta rastreabilidade com as dimensões apresentadas no referencial teórico.

O Capítulo 5 apresenta a ferramenta que viabiliza as propostas apresentadas anteriormente. Neste capítulo, é apresentada a modelagem e a arquitetura da ferramenta, incluindo diagramas e explicações contextualizadas.

O Capítulo 6 apresenta a avaliação do processo proposto utilizando técnicas de Engenharia de Software Experimental [WOH00] e [TRA02]. É apresentado o referencial teórico do processo de experimentação e, por fim, o estudo experimental.

O Capítulo 7 conclui o trabalho com uma síntese do que foi alcançado e algumas sugestões para trabalhos futuros.

2 ONTOLOGIAS

Este capítulo fornece uma visão geral sobre ontologias, apresentando os seus principais conceitos. Inicia-se com uma revisão sobre suas definições, visando discutir e clarificar como o conhecimento pode ser organizado e estruturado por uma ontologia. Para organizar o conhecimento, é necessária uma linguagem declarativa e, para este fim, foi desenvolvido um estudo sobre a linguagem de manipulação de ontologias OWL [W3C06]. Por fim, desenvolveu-se a análise de algumas propostas de engenharia de ontologias, visando identificar as melhores práticas para apoiar uma proposta de integração de ontologias no Processo Unificado.

2.1 DEFINIÇÕES DE ONTOLOGIA

O termo “ontologia” surgiu na filosofia através de Aristóteles e significa “uma explicação sistemática da existência”, isto é, a definição de um domínio do conhecimento em um nível genérico, utilizada para especificar o que existe ou o que se pode dizer sobre o mundo.

Em Ciência da Computação, ontologias representam a aquisição do conhecimento a partir de dados semi-estruturados utilizando um conjunto de métodos, técnicas ou processos automáticos ou semi-automáticos. Dentro de Ciência da Computação, o termo “ontologia” teve sua origem na comunidade de Inteligência Artificial.

Segundo Thomas Gruber, em [GRU93a], aquilo que existe é aquilo que pode ser representado. Quando o conhecimento sobre um domínio é especificado em uma linguagem declarativa, o conjunto de objetos que pode ser representado é denominado universo de discurso. Pode-se descrever uma ontologia de um programa pela definição de um conjunto representacional de termos. As definições associam os nomes de entidades do universo de discurso (como classes, relacionamentos, funções ou outros objetos) com um texto que descreve o que os nomes significam e com os axiomas formais, que tanto restringem a interpretação destes termos quanto a boa formação no seu uso.

Ontologias são utilizadas para definir os termos usados para descrever e representar um domínio de informação. Entende-se por domínio de informação uma específica área de conhecimento. Para tanto, ontologias incluem definições de conceitos e seus relacionamentos, tornando assim o conhecimento compartilhado e, de certa forma, reusável.

Gruber, em [GRU93b], define ontologias como uma “especificação explícita de uma conceituação”. Uma conceituação é uma abstração simplificada da realidade que se deseja representar, isto é, um conjunto de objetos, restrições, relacionamentos e entidades que se assumem necessárias em alguma área de aplicação.

A conceituação de Gruber foi modificada por Borst, em [BOR97], definindo ontologias como uma “especificação formal de uma conceituação compartilhada”. Esta definição enfatiza o fato que deve haver um acordo na conceituação do que é especificado.

Para se descrever uma ontologia, Maedche propõe, em [MAE02], uma estrutura (O) composta da quintupla: $O := \{C, R, Hc, rel, Ao\}$, onde:

- C e R são dois conjuntos disjuntos formados por conceitos e relacionamentos, respectivamente;
- Hc representa a taxonomia da ontologia, isto é, a hierarquia dos conceitos e relacionamentos;
- rel representa os conceitos não taxonômicos;
- Ao representa o conjunto de axiomas da ontologia.

As ontologias vêm sendo utilizadas para descrever artefatos com diferentes níveis de estruturas [W3C06]. Estes níveis variam desde simples taxonomias, *schemas* para metadados até teorias lógicas. Usualmente, elas são expressas em linguagens lógicas para que possam ser consistentes o suficiente para a extração do conhecimento.

A linguagem OWL (Web Ontology Language) é parcialmente mapeada através de lógica descritiva, que é um subconjunto da lógica de predicados, tornando possível um eficiente apoio lógico. Por apoio lógico, define-se o processamento do conteúdo da informação ao invés de sua simples apresentação.

2.2 WEB ONTOLOGY LANGUAGE (OWL)

Para se definir e manipular ontologias, sugere-se a utilização de linguagens que suportem estruturas para representação do conhecimento. Esta representação é realizada através da descrição formal de um conjunto de termos sobre um domínio específico. A definição de uma linguagem é necessária para a representação e descrição formal da estrutura que especifica uma conceituação.

Um termo que possui determinado significado pode variar sua semântica conforme o contexto empregado. Para solucionar este problema de ambigüidade e propor uma estrutura formal para a representação do conhecimento, surgiram algumas propostas de linguagens, como:

- KIF: *Knowledge Interchange Format* (1998) [NCI98];
- XOL: *XML-based Ontology Exchange Language* (1999) [KAR99];
- OIL: *Ontology Inference Layer* (1999) [OIL06];
- RDF: *Resource Definition Framework* (2000) [W3C06];
- SHOE: *Simple HTML Ontology Extensions* (2000) [SHO06];
- DAML: *DARPA Agent Markup Language* (2000) [DAM06];
- OWL: *Web Ontology Language* (2003).

Será desenvolvido nesta seção um estudo sobre a linguagem de definição e manipulação de ontologias chamada OWL. A linguagem OWL foi recomendada pela [W3C06] em fevereiro de 2004 como linguagem para manipulação de ontologias e seu diferencial é a capacidade de processamento semântico através de inferência.

Antoniou estabelece, em [ANT04], três requisitos para a definição de linguagens lógicas:

- **Sintaxe bem definida:** a importância desta sintaxe é clara e justificada pelas linguagens de programação, pois é uma condição necessária para o processamento computacional;
- **Semântica formal:** descreve precisamente o significado sobre um conhecimento. O termo “precisamente” significa que não estão associadas intuições subjetivas ou abertas a diferentes interpretações entre diferentes pessoas (ou máquinas). Esta semântica formal é estabelecida, por exemplo, através de uma lógica matemática que descreve:
 - *Transitividade:* se x é uma instância de uma classe A , e essa classe A é subclasse de uma classe B , então é possível inferir que x também é instância da classe B ;

- *Equivalência*: se a classe A é equivalente à classe B, e a classe B é equivalente à classe C, então é possível inferir que a classe A é equivalente à classe C;
- *Consistência*: supondo que x seja uma instância da classe A, e A é subclasse de $(B \cap C)$ e subclasse de D. Se B e D forem disjuntos, pode-se verificar uma inconsistência no modelo, pois A deveria ser vazio;
- *Classificação*: ao se estabelecer que uma declaração forma uma condição para definição de instância de uma classe A, e um indivíduo x possui esta declaração, então se pode concluir que x é uma instância de A;
- **Apoio lógico**: a semântica formal é seu pré-requisito, pois as derivações, como descritas anteriormente, podem ser realizadas automaticamente sem a intervenção humana. Estas verificações realizam a consistência ontológica e do conhecimento, os relacionamentos entre as classes e a classificação automática de suas instâncias.

Uma semântica formal e o seu apoio lógico são geralmente providos através do mapeamento de uma linguagem ontológica em um formalismo. Estes requisitos foram a base para uma divisão da linguagem OWL em três sublinguagens: OWL Full, OWL DL e OWL Lite.

A linguagem OWL é construída sobre RDF e RDF Schema e baseada na sintaxe XML. O modelo básico de dados do RDF, e herdado por OWL, é definido através de:

- **Recurso** (*resource*): qualquer entidade referenciada através de um URI (*Universal Resource Identifier*);
- **Propriedade** (*property*): representam recursos, características que representam recursos ou relacionamento entre recursos;
- **Declaração** (*statement*): corresponde a uma propriedade ou valor dessa propriedade associada a um recurso específico. Uma declaração é dividida em três partes: sujeito (recurso), predicado (propriedade do recurso) e objeto (valor da propriedade).

Para se estruturar um documento OWL, define-se em alto nível:

- **Classes**: conjunto de instâncias com características comuns.

- *Superclasse*: como as classes podem ser organizadas hierarquicamente, as instâncias diretas das subclasses são também instâncias das superclasses.
- *Relacionamentos*: as classes podem ser sobrepostas arbitrariamente.
- *Disjunções*: todas as classes podem potencialmente se sobrepor, porém em muitos casos é necessário fazer com que estas classes não compartilhem instâncias.
- **Propriedades**:
 - *Tipos (datatype properties)*: identificam os valores primitivos das instâncias, como *integer*, *float*, *string*, *boolean*, etc.
 - *Objetos (object properties)*: representam o vínculo de duas instâncias, isto é, seus relacionamentos.
 - *Inversa (inverseOf)*: representam um relacionamento bidirecional. Adicionando valores a uma propriedade, conseqüentemente, se adiciona valores a uma segunda.
 - *Transitivas (TransitiveProperty)*: Se a instância x está relacionada com a instância y, e a instância y está relacionada à instância z, então x está relacionado com z. Usado principalmente em relações “parte-de”.
- **Indivíduos**: representam os objetos em um domínio, isto é, instâncias específicas. Verifica-se aqui que dois nomes podem representar o mesmo objeto no mundo real.

A estruturação da linguagem OWL é pertinente no que tange a forma com a qual o conhecimento pode ser representado. Porém, apenas o entendimento da linguagem não é suficiente para estruturar o conhecimento, pois o mapeamento dos conceitos do mundo real em uma estrutura formal requer uma abordagem sistemática, assim como o desenvolvimento de software. Neste contexto, diversos autores, dentre eles [LOP97], [DEV02] e [KIS04], descrevem uma abordagem chamada *Engenharia Ontológica*. Na próxima seção, será apresentada a análise de algumas abordagens para construção de ontologias encontradas na literatura, visando extrair as melhores práticas.

2.3 CONSTRUÇÃO DE ONTOLOGIAS

A literatura fornece algumas abordagens sistemáticas para o desenvolvimento de ontologias. A construção de ontologias compreende um conjunto de técnicas e atividades necessárias para o desenvolvimento de um modelo lógico que represente determinado universo de discurso. É interessante a análise dessas propostas para extração dos principais aspectos necessários para apoiar o desenvolvimento de ontologias integrado ao Processo Unificado.

O processo de criação de ontologias pode seguir padrões não lineares. Cada equipe de desenvolvimento pode possuir seu conjunto de princípios, critérios de projeto e fases a seguir. A ausência de guias e métodos para estruturar a construção de ontologias pode gerar inconsistência entre as necessidades dos usuários e o modelo ontológico. A seguir, serão analisadas algumas das propostas encontradas na literatura.

2.3.1 Metodologia de Uschold & King

A metodologia proposta em [USC95] foi construída baseada no desenvolvimento do projeto da *Enterprise Ontology* e descreve as seguintes etapas para o desenvolvimento de ontologias:

1. **Identificação do propósito e escopo:** define o nível de formalidade em que a ontologia será descrita e representa uma especificação do intervalo de informação que caracteriza esta ontologia. Sugere-se utilizar *Cenários Motivacionais* (descrições de problemas) e *Questões de Competência Informal* (representação informal dos cenários utilizando a ontologia). Pode ser utilizada uma técnica como *brainstorming* para mapear os conceitos relevantes e descartar conceitos irrelevantes e sinônimos;
2. **Formalização:** corresponde à criação do código com suas definições formais e os termos axiomáticos, definidos na especificação da ontologia. Esta etapa compreende:
 - a. *Captura da ontologia:* identificação dos conceitos e relacionamentos relevantes para ontologia;
 - b. *Codificação:* mapeamento do conhecimento adquirido nas etapas anteriores em uma linguagem formal;

- c. *Integração de ontologias existentes*: reutilização de um domínio pertinente ao contexto em questão.
3. **Avaliação formal**: consiste na verificação da consistência, completude e a não redundância do modelo. Ela pode ser executada avaliando o propósito ou as questões de competência definidas anteriormente.
4. **Documentação**: representa a documentação do desenvolvimento da ontologia;

2.3.2 Metodologia de Grüninger e Fox

Grüninger e Fox propõem em [GRU95] uma metodologia baseada na experiência de desenvolvimento da ontologia do projeto TOVE (*Toronto Virtual Enterprise*). Esta metodologia descreve um domínio de processos de negócio e a modelagem de atividades com o uso de ontologias, permitindo construir um modelo lógico a partir de cenários informais expressos em linguagem natural.

As seguintes etapas são propostas nesta metodologia:

1. **Captura de cenários motivacionais**: o ponto de partida desta metodologia estabelece um conjunto de problemas encontrados em um domínio específico, expresso geralmente em linguagem natural.
2. **Formulação de questões de competência informal**: define os requisitos da ontologia baseados nos cenários motivacionais. As questões de competência representam perguntas que devem ser respondidas pela ontologia. Esta etapa é responsável pela validação dos cenários obtidos no estágio anterior;
3. **Especificação da terminologia da ontologia em uma linguagem formal**: corresponde a formalização das classes, atributos e relacionamentos da ontologia;
4. **Formulação das questões de competência formal**: os requisitos da ontologia são formalizados em uma linguagem declarativa;
5. **Especificações dos axiomas**: definição dos axiomas que especificam ou restringem os termos da ontologia, expressos em lógica de primeira ordem. Estes axiomas são orientados pelas questões de competência formal;
6. **Verificação da completude da ontologia**: corresponde a etapa de avaliação das questões de competência da ontologia, verificando se a mesma está completa em relação a sua especificação.

2.3.3 Framework Methontology

O *framework* Methontology, proposto por Fernández-López em [LOP97], sugere um ciclo de vida e um processo de desenvolvimento para ontologias baseado na evolução de protótipos. O ciclo de vida compreende a especificação, conceituação, formalização, integração e implementação da ontologia.

As fases do framework são:

1. **Planejamento:** tem por objetivo identificar e organizar as tarefas e recursos que deverão ser executados ao longo do ciclo de vida ontológico;
2. **Especificação:** esta fase identifica o propósito da ontologia, incluindo seus usuários, cenários de uso, o grau de formalidade necessário, etc. Além disso, esta fase se preocupa com a definição do escopo da ontologia, incluindo o conjunto representacional dos termos, suas características e granularidades. O resultado desta fase é um documento em linguagem natural que representa a ontologia;
3. **Aquisição do Conhecimento:** esta fase é executada concomitantemente com a fase (1). Trata-se de uma elicitación não-prescritiva do conhecimento através de discussões que englobam entrevistas com especialistas do domínio e análise de textos. É sugerido o uso de técnicas de aquisição de conhecimento de sistemas baseados em conhecimento;
4. **Conceituação:** compreende a identificação informal dos termos do domínio que correspondem a conceitos, instâncias, verbos relacionais ou propriedades. Como resultado desta fase, obtém-se um modelo conceitual que descreve o problema e sua solução;
5. **Formalização:** esta fase tem por objetivo mapear o conhecimento estruturado na fase (4) em um modelo formal semi-computável, isto é, que não é processável diretamente por um computador;
6. **Integração:** visando obter uniformidade entre ontologias, esta fase corresponde à verificação da possibilidade de incorporar outras ontologias. Esta fase é aplicável apenas quando se deseja reutilizar ontologias existentes;
7. **Implementação:** é a atividade que visa à formalização da ontologia em uma linguagem computável;

8. **Avaliação:** o *framework* Methontology enfatiza esta fase. Um conjunto de guias é fornecido para avaliar se a ontologia está incompleta, inconsistente ou redundante;
9. **Documentação:** representa uma relação de documentos obtidos como resultado das demais atividades. Esta fase é especialmente importante quando se deseja reutilizar ou compartilhar a ontologia com outras existentes.
10. **Manutenção:** esta fase é executada após o término do desenvolvimento da ontologia, caso seja necessária alguma manutenção.

É apresentado em [BLA98] o ambiente para construção de ontologias que utiliza o *framework* descrito, chamado ODE (*Ontology Design Environment*). Seu objetivo é apoiar os engenheiros durante o processo de desenvolvimento e o ciclo de vida da ontologia, automatizando e integrando cada fase descrita.

2.3.4 Abordagem de Falbo e Rocha

Falbo e Rocha apresentam em [FAL98] uma abordagem sistemática para o desenvolvimento de ontologias. Basicamente esta proposta consiste em:

1. **Identificar o propósito e a especificação dos requisitos da ontologia:** a primeira atividade a ser executada é a identificação do propósito e dos usos da ontologia, isto é, sua competência. Falbo sugere a utilização de cenários motivacionais para ontologia.
2. **Captura da ontologia:** capturar a conceituação do domínio com base na competência da ontologia. Neste ponto, devem ser identificados e organizados os conceitos, relações, propriedades e axiomas relevantes do domínio. São sugeridos alguns critérios para o projeto da ontologia: clareza, coerência e um mínimo de compromisso ontológico;
3. **Formalização:** tem como objetivo representar a conceituação em uma linguagem formal, definida sob lógica de primeira ordem, para representar suas estruturas de maneira precisa e não ambígua.
4. **Integração com outras ontologias:** caso seja necessário integrar a ontologia em desenvolvimento com outra já existente, sugere-se realizá-la durante os passos de captura e formalização.
5. **Avaliação:** o objetivo é verificar se a ontologia em desenvolvimento está de acordo com os requisitos especificados. Sugere-se utilizar as questões de

competências definidas anteriormente pelo especialista de domínio para a avaliação do modelo lógico;

6. **Documentação:** sugere-se desenvolver uma documentação que englobe o processo descrito, incluindo os propósitos, requisitos e cenários motivacionais, descrição informal e formal da conceituação e o critério de projeto adotado.

2.3.5 Guia de Noy e McGuinness

O guia de Noy e McGuinness [NOY01b] foi resultado da experiência adquirida durante a utilização das ferramentas Protégé [PRO06], Ontolingua [FAR00] e Chimaera [MCG00] como ambientes de edição de ontologia.

Segundo [NOY01b], não existe um método mais adequado para a construção de ontologias, pois sempre existem alternativas viáveis. A escolha da solução sempre vai depender da aplicação que se tem em mente, do escopo da ontologia e de possíveis integrações. Logo, o objetivo de sua proposta não é definir uma metodologia, mas sim um guia para o desenvolvimento de ontologias.

O processo proposto é iterativo e possui sete passos:

1. **Determinar o domínio e o escopo da ontologia:** neste passo, o objetivo é restringir o escopo da ontologia, definindo qual seu uso e as questões de competência que devem ser respondidas;
2. **Considerar o reuso de ontologias existentes:** o objetivo é limitar esforços pela integração de ontologias existentes ou manutenção de domínios que já possuem ontologias definidas;
3. **Listar os termos importantes da ontologia:** sugere-se desenvolver uma listagem dos conceitos relevantes do domínio, suas propriedades e seus relacionamentos, além de suas respectivas descrições;
4. **Definir as classes e suas hierarquias:** é definido um conjunto de guias no qual são abordadas questões importantes para o desenvolvimento de ontologias, tais como a organização taxonômica, adição de novos recursos na ontologia, etc.
5. **Definir as propriedades das classes:** os termos descritos no Passo 3 que não se tornaram classes, deverão ser mapeados em propriedades;
6. **Definir as facetas das propriedades:** definem-se aqui as facetas das propriedades, tais como cardinalidade, intervalo de valores, restrições, etc.

7. **Criar instâncias:** o objetivo é a criação de instâncias de classe e seus valores, definindo uma declaração do tipo sujeito – predicado – objeto.

2.3.6 Abordagem colaborativa de Holsapple e Joshi

A proposta de Holsapple e Joshi [HOL02] define cinco possíveis abordagens para o projeto de ontologias:

- *Inspiração:* consiste no projeto de uma ontologia baseado na imaginação e criatividade;
- *Indução:* consiste no projeto baseado na observação, investigação e análise de casos específicos;
- *Dedução:* sugere-se o projeto a partir da adoção de alguns princípios gerais para um caso específico;
- *Síntese:* o projeto de uma nova ontologia é resultado de uma compilação de uma série de outras ontologias;
- *Colaboração:* consiste no desenvolvimento a partir do esforço comum.

A abordagem colaborativa sugerida pelos autores é estruturada em quatro fases:

1. **Preparar:** nesta primeira fase, são definidos os critérios de projeto da ontologia e a definição de seu escopo;
2. **Ancorar:** é desenvolvida uma ontologia preliminar para “ancorar” o projeto, servindo como base para colaboração dos interessados no desenvolvimento da ontologia;
3. **Melhoria repetitiva e iterativa:** o objetivo é a convergência de diferentes visões sobre o modelo preliminar da ontologia para sua consolidação;
4. **Aplicação:** nesta fase ocorre a demonstração da utilidade da ontologia a partir de sua aplicação.

2.3.7 Metodologia On-To-Knowledge

A metodologia On-To-Knowledge, proposta em [SUR02], tem o objetivo de introduzir e manter aplicações de gestão do conhecimento baseadas em ontologias. Para isso, propõem-se as seguintes atividades:

1. Meta-processo de conhecimento, que compreende:
 - Estudo de viabilidade;

- Iniciação;
 - Refinamento;
 - Avaliação;
 - Aplicação;
 - Evolução.
2. Processo de conhecimento, que compreende:
- Criação do conhecimento e/ou importação de documentos e metadados;
 - Captura do conhecimento;
 - Recuperação;
 - Acesso;
 - Utilização do conhecimento.

2.3.8 Knowledge Unified Process

O Knowledge Unified Process (KUP) [ORL03] é uma proposta de Orleans e Lucena que objetiva unificar diversas propostas existentes na academia e na indústria para a construção de ontologias. O processo foi inspirado no Processo Unificado e é estruturado em um ciclo de vida iterativo e incremental, orientado a casos de uso e centrado na arquitetura, porém com o foco específico em ontologias.

O processo é dividido em 3 fases e 9 disciplinas. As fases são organizadas por iterações que correspondem a entregas bem definidas. As fases propostas são:

1. **Concepção:** ênfase no escopo da ontologia;
2. **Construção:** ênfase no projeto, implementação e aplicação da ontologia;
3. **Evolução:** integração de novos conceitos sobre a ontologia;

As disciplinas envolvidas com o processo constituem:

1. **Instanciação e Customização do KUP:** definir quais fases, disciplinas, papéis, atividades e artefatos farão parte do projeto em desenvolvimento;
2. **Análise de Viabilidade:** permitir a prévia identificação de possíveis oportunidades ou riscos que podem influenciar o projeto;
3. **Análise de Requisitos:** levantar iterativamente os requisitos que a ontologia deve atender;
4. **Projeto:** corresponde a geração da ontologia em linguagem formal ou em um modelo conceitual que permita sua implementação;

5. **Implementação:** transformação da representação conceitual ou formal em uma linguagem de implementação;
6. **Implantação:** corresponde à aplicação da ontologia desenvolvida e sua integração com as aplicações identificadas nas fases iniciais de projeto;
7. **Gerência do Projeto:** por ser um processo iterativo e incremental, esta disciplina é constante durante todo o ciclo de desenvolvimento da ontologia e corresponde a gerencia e configuração, documentação e treinamento.
8. **Processos de Suporte e Integralização:** ambos os processos foram definidos como trabalho futuro e correspondem aos processos de pós-desenvolvimento;

2.3.9 Processo Helix-Spindle

O modelo de processo Helix-Spindle, proposto em [KIS04], sugere a utilização de uma abordagem teórica e prática para o desenvolvimento de ontologias, ao invés de seguir exclusivamente uma destas abordagens. O princípio teórico e dedutivo especifica a ontologia em uma forma executável enquanto o princípio prático realiza sua validação.

O processo Helix-Spindle é baseado em um modelo de processo capturado a partir de três grandes fases, motivadas pela noção de um desenvolvimento incremental e de sucessivos refinamentos de versões de ontologias geradas. Conseqüentemente, consegue-se desenvolver ontologias, aumentando o nível de formalidade. As três fases do modelo de processo são:

1. **Concepção:** representação informal da ontologia como, por exemplo, de maneira textual;
2. **Elaboração:** representação semi-formal da ontologia, como por exemplo UML ou Ontolingua;
3. **Definição:** representação formal da ontologia, como por exemplo Backus Naur Form (BNF) [NAU60] ou lógica de predicados.

Todas as três fases consistem na construção da ontologia através de uma abordagem teórica. A ontologia resultante de cada fase é testada de forma empírica.

2.4 CONSIDERAÇÕES

Este capítulo procurou apresentar a definição de ontologias no contexto da Ciência da Computação, visando estruturar uma descrição e um entendimento comuns.

Para estruturar o conhecimento expresso em uma ontologia, é necessária a análise de linguagens lógicas. Para tanto, foi apresentado um estudo sobre OWL, visando analisar a forma na qual o conhecimento pode ser organizado em uma linguagem declarativa. A importância desta análise é válida para identificar as estruturas lógicas que mapeiam um modelo conceitual.

O estudo de uma linguagem lógica não é suficiente para o desenvolvimento de uma ontologia coerente com o domínio. Para tanto, foram avaliadas algumas propostas relacionadas à construção de ontologias, analisando as diferentes sistemáticas de desenvolvimento. O objetivo do estudo é identificar padrões e boas práticas que apoiem a criação de ontologias integradas ao processo de desenvolvimento de software.

Uma primeira observação com relação à construção de ontologias diz respeito aos métodos utilizados para o desenvolvimento das propostas que, em sua maioria, foram empíricos. Este tipo de abordagem implica que a ontologia gerada a partir de determinada metodologia possa apresentar diferentes graus de abstração e detalhamento se comparada à outra. Isso se deve à ênfase dada por diferentes autores em diferentes etapas do processo de desenvolvimento.

Frente às propostas apresentadas, percebe-se também uma convergência de pontos estruturais. É observado um certo padrão interativo para o desenvolvimento de ontologias, que incluem os seguintes passos:

1. Definição do escopo da ontologia;
2. Aquisição e explicitação dos conceitos do domínio (modelo conceitual);
3. Formalização dos conceitos (modelo formal);
4. Integração, se necessário, com ontologias existentes;
5. Definição de axiomas;
6. Validação.

Outra questão pertinente diz respeito à ênfase que determinadas propostas dão para alguns passos em detrimento de outros. Mesmo observando este fato, todas as propostas indicam que os seis passos são necessários.

Por fim, pode-se notar que muito do esforço empregado na construção de uma ontologia está relacionado com a compreensão do domínio e com sua conceituação. Durante as fases iniciais do Processo Unificado, os esforços se concentram em aspectos semelhantes, como a modelagem de negócios e a definição do domínio. Ainda que preocupado com a definição do domínio, o processo de desenvolvimento tradicional não utiliza linguagens lógicas para representar formalmente uma conceituação.

Frente aos resultados obtidos pela análise de diferentes propostas de construção de ontologias, sugere-se a elaboração de uma abordagem que consiga integrar, dentro do tradicional processo de desenvolvimento de software, um modelo lógico que relacione os conceitos do domínio do problema com a arquitetura da solução. Como resultado desta integração, objetiva-se uma sinergia entre os modelos.

3 INTEGRAÇÃO DE ONTOLOGIAS NO PROCESSO UNIFICADO

Este capítulo apresenta um paralelo entre Engenharia de Software e a construção de ontologias. Além disso, é apresentada uma proposta de integração de ontologias no Processo Unificado. O objetivo desta comparação é identificar oportunidades que permitam uma cooperação entre as comunidades de engenharia de software e de engenharia do conhecimento, possibilitando a inclusão de um modelo conceitual expresso através de uma ontologia no processo de desenvolvimento de software. Neste capítulo, será proposta a criação de uma nova disciplina no Processo Unificado para construção de ontologias. Com isso, consegue-se o apoio necessário para a proposta de rastreabilidade ontológica. Serão descritas as atividades relacionadas com esta nova disciplina, criando um elo com as propostas para construção de ontologias estudadas. Por fim, serão apresentados alguns trabalhos relacionados e algumas considerações.

3.1 CONSTRUÇÃO DE ONTOLOGIAS E ENGENHARIA DE SOFTWARE

Como ontologias surgiram na comunidade de Inteligência Artificial, pode existir uma crença de que seu desenvolvimento necessite de uma revolucionária metodologia ou tecnologia. Na verdade, ontologia é um conceito baseado em entidades e seus relacionamentos. Em muitos casos, pode-se criar uma ontologia a partir de uma metodologia de Engenharia de Software, recuperando suas entidades a partir de, por exemplo, diagramas de classes, modelos entidade-relacionamento ou uma análise estruturada de um sistema [DEV02].

Mesmo que a engenharia de ontologias e a Engenharia de Software concentrem esforços no mapeamento de conceitos e tradução entre modelos, existem algumas diferenças entre as duas abordagens. A complexidade relacionada a um modelo lógico é diferente da relacionada a um modelo orientado a objeto (OO). Por exemplo, a definição da lógica de predicados implica que os relacionamentos entre os conceitos sejam considerados elementos de primeira ordem, o que não ocorre em um paradigma OO (como o modelo de classes não estereotipado), na qual os relacionamentos são subordinados aos conceitos das classes. Logo, não é possível mapear um modelo ontológico completo, com todas as restrições lógicas, em um diagrama de classes OO. Isso deve porque estas representações OO apenas capturam a taxonomia dos conceitos e seus relacionamentos, e não as regras lógicas que uma ontologia define.

Outro ponto pertinente é relacionado com a tendência de desenvolvimento de ontologias contextualizadas, apoiadas pela proposta da *Web Semântica*. A Web Semântica,

idealizada por Tim Berners-Lee, descreve uma rede que interligue semanticamente os conteúdos publicados na Web [BER01]. A idéia desta rede é que os conteúdos não sejam compreensíveis apenas pelos humanos, mas também pelo processamento computacional utilizando, por exemplo, ontologias e agentes de software.

Segundo Tim Berners-Lee, a Web Semântica não é uma Web alternativa, mas uma extensão da atual, onde a informação é recuperada através de uma semântica bem definida, habilitando uma cooperação entre trabalho humano e computacional. Ela é baseada em uma estrutura descentralizada, onde agentes de software possam migrar em um ambiente de diversas estruturas, compreendendo e realizando tarefas sofisticadas para os usuários através da valoração semântica destes conteúdos.

Conforme observado em [BER01], a Web Semântica será formada por "ilhas de conhecimento", isto é, ontologias específicas para alguma aplicação que geram interoperabilidade com outras ontologias para recuperação semântica do conhecimento.

Diante desta tendência de descentralização, é interessante observar formas para construção de ontologias no processo de desenvolvimento de software, habilitando assim a modelagem semântica de sistemas de informação específicos.

3.2 ONTOLOGIAS E O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

O processo de desenvolvimento de software define o ciclo de vida de um produto de software. Entende-se por ciclo de vida, todas as atividades envolvidas com o produto, desde o levantamento de suas necessidades até o desenvolvimento, teste e manutenção.

O Processo Unificado é um exemplo de um processo de desenvolvimento bem sucedido. Ele organiza o desenvolvimento de software em quatro diferentes fases: iniciação, elaboração, construção e transição. Cada fase pode suportar diversas iterações e, por sua vez, cada iteração é categorizada em nove disciplinas: modelagem de negócio, requisitos, análise e projeto, implementação, teste, implantação, gerência de mudanças e configuração, gerência de projetos e ambiente. A Figura 3.1 apresenta a arquitetura geral do Processo Unificado.

Durante a disciplina de Modelagem de Negócios, o Processo Unificado sugere o desenvolvimento de uma representação de conceitos do mundo real, chamado de Modelo de Domínio. O Modelo de Domínio descreve os conceitos do domínio e seus relacionamentos, utilizando a sintaxe UML [UML06] do diagrama de classes.

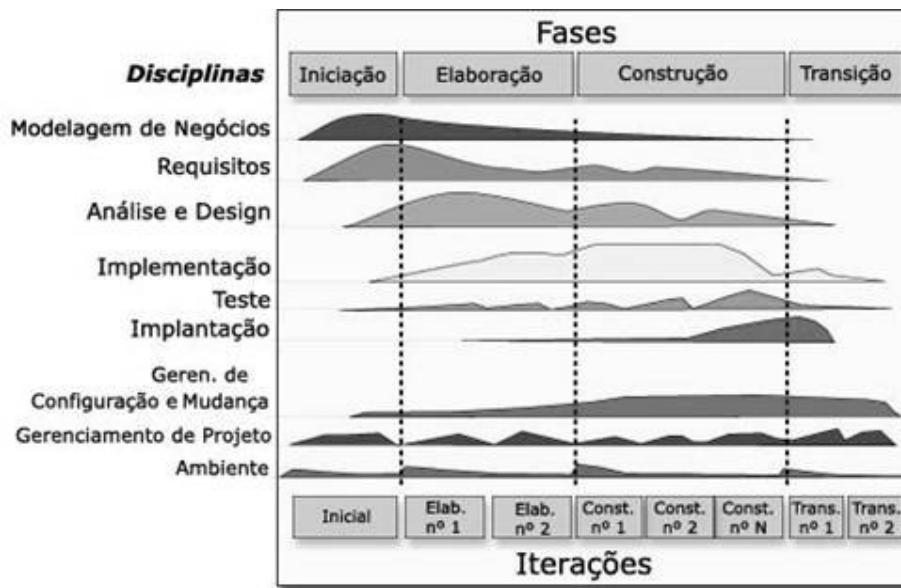


Figura 3.1 – Disciplinas, fases e iterações do Processo Unificado.

É importante lembrar que o objetivo deste modelo é entender o contexto do sistema, e não especificar a arquitetura de uma possível solução. Portanto, deve-se desenvolvê-lo em um alto nível de abstração e relativamente independente da solução considerada.

O Modelo de Domínio descreve apenas os conceitos mais significativos do sistema. Os demais conceitos podem ser listados num pequeno glossário do domínio. O objetivo do glossário e do modelo é o auxílio na elaboração de uma conceituação compartilhada entre os envolvidos com a definição e o desenvolvimento do sistema. Além da conceituação compartilhada, o modelo auxilia na identificação de possíveis ambigüidades que possam existir entre os vários subdomínios.

A análise do Modelo de Domínio permite observar sua similaridade com ontologias: ambos especificam os conceitos e seus relacionamentos no contexto do domínio de discurso. O Modelo de Domínio representa o domínio da aplicação usando uma visão de negócios e não uma visão arquitetural, com o objetivo de compreender o contexto do sistema e compartilhar essa conceituação entre os *stakeholders*. Ontologias também podem ser utilizadas para representar uma conceituação compartilhada entre os *stakeholders*, representando um domínio de aplicação em uma perspectiva de negócios.

Após a contextualização do domínio do problema, é preciso capturar as necessidades dos *stakeholders* para definição do sistema. É durante a disciplina de Requisitos que o time compreende as necessidades dos usuários, obtendo as características e restrições do sistema a partir do especialista de domínio. Para se desenvolver uma ontologia consistente,

é necessário obter o conhecimento explícito sobre o domínio. Utilizando o Modelo de Domínio como entrada, é possível criar uma ontologia e refiná-la usando a especificação de requisitos. Este refinamento deve identificar os conceitos, organizar a taxonomia e definir as regras lógicas que definem o domínio do problema.

Após a especificação dos requisitos, o software será modelado e sua arquitetura definida na disciplina de Análise e Projeto. O uso de ontologias nas fases anteriores do processo de desenvolvimento permite uma rastreabilidade semântica entre os conceitos do domínio e os elementos arquiteturais derivados desse domínio.

Para permitir a rastreabilidade semântica, este trabalho propõe a integração de ontologias no Processo Unificado. O Processo Unificado foi escolhido pela sua boa aceitação na comunidade acadêmica e industrial. Utilizando ontologias, é possível gerar elos de rastreabilidade entre o Modelo de Domínio, Especificação de Requisitos de Software, Modelos de Análise e Modelo de Projeto. Além disso, é possível obter relacionamentos implícitos e cognitivos entre os artefatos do sistema, utilizando inferências sobre a ontologia.

3.3 PROPOSTA DE INTEGRAÇÃO DE ONTOLOGIAS NO PROCESSO UNIFICADO

A presente proposta sugere a integração de ontologias no Processo Unificado de forma não intrusiva, visto que não são sugeridas modificações nas demais disciplinas existentes. A idéia básica é criar uma nova disciplina, chamada *Modelagem do Conhecimento*, para desenvolvimento de um modelo lógico que represente os conceitos do domínio do problema. Utilizando este modelo, será possível relacionar toda a especificação dos componentes de software com os conceitos da ontologia, possibilitando a rastreabilidade ontológica.

3.3.1 Disciplina de Modelagem do Conhecimento








A proposta de criação de uma nova disciplina no Processo Unificado visa à modelagem do conhecimento relativo ao domínio do problema. Para tanto, é necessário uma adaptação no Processo Unificado.

Definiu-se a ontologia como sendo o único artefato produzido durante a modelagem do conhecimento. Para sua construção, esta disciplina inclui algumas das boas

práticas identificadas na literatura, agora integradas ao desenvolvimento de um sistema de informação. Todas as propostas de construção de ontologias analisadas possuem como objetivo a engenharia de um modelo lógico para determinado domínio, independente da engenharia de um software específico.

Para a definição do processo que descreve a disciplina, será utilizada a notação do meta-modelo SPEM (*Software Process Engineering Metamodel*) [OMG06]. O Quadro 3.1 representa os elementos utilizados para este fim e seus significados.

Quadro 3.1 - Notação utilizada para definição do processo.

Elemento	Ícone proposto pelo SPEM	Elemento	Ícone proposto pelo SPEM
Artefatos		Papéis	
Atividades		Pacote de Processo	
Fases		Processo	
Iterações			

Serão utilizados alguns dos artefatos do Processo Unificado como entrada para esta disciplina. Suas atividades estão definidas conforme os passos identificados na Seção 2.4, que compreendem:

1. Definição do escopo da ontologia;
2. Aquisição e explicitação dos conceitos do domínio (modelo conceitual);
3. Formalização dos conceitos (modelo formal);
4. Integração, se necessário, com ontologias existentes;
5. Definição de axiomas;
6. Validação.

A disciplina proposta está estruturada em três principais atividades:

- **Projeto:** ocorre durante a fase inicial do processo, executando os passos 1, 2 e 3. Esta atividade recebe como entrada o artefato chamado Modelo de Domínio, proposto durante a disciplina de Modelagem de Negócios, para definir o conhecimento através do escopo da ontologia (1) e da explicitação de seus conceitos e relacionamentos (2). Como resultado, é possível formalizar uma ontologia preliminar (3), utilizando uma tradução automática deste modelo. Posteriormente, é possível refinar o

conhecimento expresso através da ontologia de acordo com os artefatos produzidos;

- **Manutenção:** ao terminar a atividade de projeto, inicia-se a manutenção do conhecimento, executando os passos 4 e 5. O objetivo é refinar a ontologia previamente gerada, adequando às estruturas lógicas que tanto definem quanto restringem o domínio. Também é possível a integração, se necessário, a ontologias existentes (4). Durante esta etapa, são definidos os axiomas do modelo lógico (5). Os conceitos da ontologia são refinados à medida que os documentos de Especificação de Requisitos, Modelo de Análise e Modelo de Projeto evoluam, pois é possível que surjam novos conceitos e relacionamentos ao longo do ciclo de vida do software;
- **Validação:** a validação do conhecimento é constante durante todo o processo de desenvolvimento, conforme o passo 6.

Nas próximas seções, serão detalhadas as atividades apresentadas. Para melhor compreender a disciplina, foi elaborado um diagrama que representa o *workflow* das atividades envolvidas com a proposta, representado pela Figura 3.2.

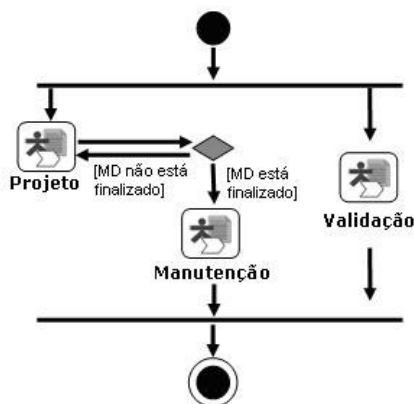


Figura 3.2 – *Workflow* da Disciplina de Modelagem do Conhecimento.

Após definidas as atividades da Modelagem do Conhecimento, faz-se necessária a sua integração no Processo Unificado. Foi desenvolvido um diagrama, apresentado pela Figura 3.3, que contempla a expectativa de dispersão da disciplina sobre as fases do Processo Unificado.

É possível perceber que as atividades da modelagem do conhecimento iniciam após o início da disciplina de Modelagem de Negócios e tem seu ponto de maior esforço

durante a transição entre a fase de Iniciação e a de Elaboração. É nesta fase que os projetistas já possuem um bom entendimento do domínio do problema e são capazes de representar o conhecimento envolvido com o software em desenvolvimento. Durante o desenvolvimento de software, é necessária a manutenção da ontologia, a qual garante a integridade dos novos conhecimentos obtidos durante o detalhamento de sua arquitetura.

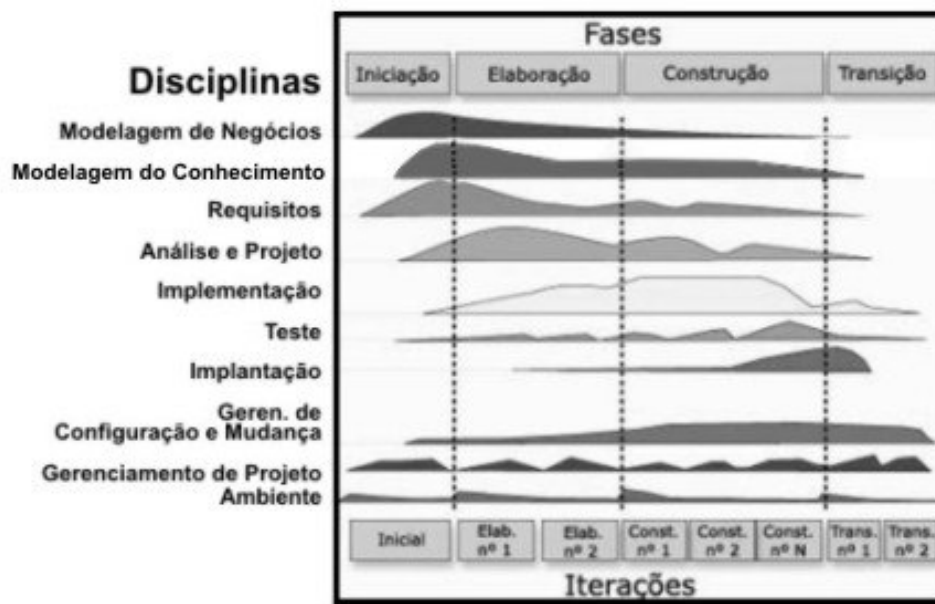


Figura 3.3 – Extensão ao Processo Unificado.

A seguir, apresentaremos em detalhes as atividades da disciplina de Modelagem do Conhecimento.

3.3.1.1 Projeto

Foi identificado que, durante a disciplina de Modelagem de Negócios, o Processo Unificado sugere o desenvolvimento de um artefato chamado Modelo de Domínio, que é uma representação conceitual do universo que se deseja modelar. Devido a sua similaridade com ontologias, o mesmo será utilizado como insumo para a atividade de projeto.

A partir do Modelo de Domínio, gerado durante a disciplina de Modelagem de Negócios, é definida uma versão preliminar da ontologia. Esta versão possui a mesma expressividade conceitual que o Modelo de Domínio, porém organizado em um formalismo lógico. Neste formalismo, ocorre a delimitação do escopo e a explicitação dos conceitos do domínio, além dos relacionamentos taxonômicos e não taxonômicos. Adicionalmente, esta versão da ontologia é refinada durante a disciplina de Análise de Requisitos, visando à manutenção de sua integridade caso surjam novos conhecimentos sobre o domínio. Da mesma

forma, sugere-se a atualização do Modelo de Domínio para manter sua integridade com o modelo de conhecimento expresso através da ontologia.

De acordo com [BOR97], existem três principais razões para que ontologias falhem na representação do conhecimento: o seu desenvolvimento não está vinculado a aplicações do mundo real, a sua estrutura contém o reuso de ontologias de domínios diferentes e o seu desenvolvimento compreende uma mera taxonomia de conceitos, que falha na captura das regras de inferência que representam o conhecimento tácito da aplicação.

É durante a engenharia de requisitos que se obtém o conhecimento sobre o domínio da aplicação e as necessidades dos *stakeholders*. Assim, os motivos de falha propostos por [BOR97] podem ser minimizados com um processo de engenharia de requisitos de qualidade, pois o software a ser construído está relacionado com uma aplicação do mundo real, onde se propõe a criação de uma ontologia específica e contextualizada. Nesta etapa de desenvolvimento de software, os conceitos relacionados ao domínio da aplicação são declarados explicitamente pelos *stakeholders*, sugerindo assim que as regras de inferência estejam de acordo com as características do sistema. Adicionalmente, as regras de negócios estabelecidas na especificação de requisitos e demais artefatos poderiam ser representadas por regras não taxonômicas da ontologia.

Existem algumas propostas, tais como [LEI93a] e [BRE03], para explicitação dos conceitos referentes aos requisitos do sistema com o objetivo de compartilhar e reusar o conhecimento sobre o domínio da aplicação. Os autores sugerem a criação de um léxico a partir da especificação de requisitos e sua tradução para uma ontologia.

Uma atividade importante durante a etapa de projeto é o refinamento do modelo conceitual. Para tanto, será seguida uma proposta semelhante à metodologia *On-To-Knowledge*. Após definida a versão preliminar da ontologia e a Especificação de Requisitos, é interessante percorrer este último e verificar se todos os termos relevantes estão explicitados na ontologia. Feito isso, este procedimento deve ser repetido avaliando a organização taxonômica (generalização e especialização) entre os conceitos. Por fim, é necessário avaliar os relacionamentos não taxonômicos, tais como os atributos e os relacionamentos entre os conceitos.

Apoiando a presente proposta, [KIS04] também sugere a formalização dos conceitos da ontologia utilizando a linguagem UML como uma etapa intermediária da engenharia ontológica. Segundo [LOP97], a formalização tem por objetivo transformar o conhecimento estruturado na atividade de conceituação em um modelo formal semi-computável.

Existe uma proposta do *Ontology Working Group* [OWG06] da OMG para definição de ontologias utilizando o chamado *Ontology Definition Meta-Model* (ODM). Portanto, também é possível especificar o Modelo de Domínio através desta proposta que estende a UML.

Após definido e validado o Modelo de Domínio, é realizada a extração automática de uma versão preliminar da ontologia. Esta versão é definida através do mapeamento de algumas estruturas, tais como classes, atributos, associações e generalizações em UML para estruturas com a mesma semântica em OWL. Será adotada esta linguagem devido a sua recomendação pela [W3C06], cabendo ao desenvolvedor definir qual das três de suas sublinguagens utilizar (Lite, DL ou Full). O Quadro 3.2 apresenta o guia de mapeamento das estruturas.

Quadro 3.2 – Mapeamento entre estruturas UML e OWL.

UML	OWL
Classe	<i>OWL Class</i>
Atributos	<i>Datatype Property</i> <ul style="list-style-type: none"> • Domínio: Classe Pai • Imagem: Tipo XSD definido pelo <i>XML Schema</i>.
Associações	<i>Object Property</i> <ul style="list-style-type: none"> • Domínio: Classe da associação inicial. • Imagem: Classe da associação final. • Restrições: <ol style="list-style-type: none"> 1. Se associação inicial e final é navegável, a propriedade é <u>simétrica</u>. 2. Se a multiplicidade das duas associações é igual, cria-se uma restrição de <u>cardinalidade</u>, senão cria restrições de <u>cardinalidade mínima e máxima</u>.
Generalização	Define relacionamento taxonômico entre as classes.

Foi utilizada a proposta ODM da [OWG06] como base para definir o guia de mapeamento. O documento ODM define as equivalências entre classes, atributos, associações, generalizações e restrições. Para exemplificar, a Figura 3.4 apresenta um cenário de uma estrutura UML.

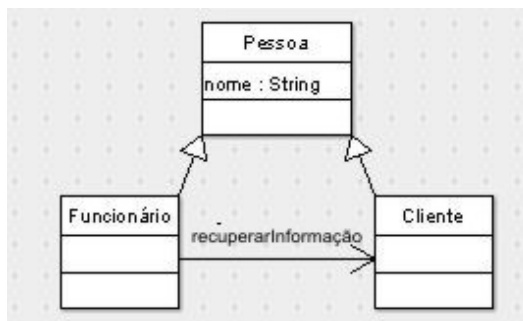


Figura 3.4 – Cenário de estrutura UML.

A estrutura ontológica derivada da Figura 3.4 corresponde a:

- Três OWL Classes: *Pessoa*, *Funcionário* e *Cliente*;
- Um *Datatype Property* chamado *nome*, que possui como domínio a OWL Class *Pessoa* e como imagem o seu tipo XSD definido pelo XML Schema: <http://www.w3.org/2001/XMLSchema#string>;
- Um *Object Property* chamado *recuperarInformação*, que possui com domínio a OWL Class *Funcionário* e como imagem a OWL Class *Cliente*.
- Uma estrutura taxonômica, organizando as OWL Classes *Cliente* e *Funcionário* como subclasse de OWL Class *Pessoa*.

Conforme já apresentado, não é possível o mapeamento completo de um modelo ontológico em um modelo de classes OO devido à falta de apoio destes modelos para representação de regras lógicas. No entanto, o oposto não é verdade. É possível mapear um modelo de classes em um modelo lógico e, posteriormente, adicionar as regras necessárias para a definição do domínio.

A Figura 3.5 apresenta o modelo de processo relacionado à atividade de projeto. A presente proposta inclui um novo papel no Processo Unificado, chamado *Engenheiro do Conhecimento*, que indica o responsável por executar as atividades relacionadas à disciplina de Modelagem do Conhecimento. O engenheiro do conhecimento recebe como entrada o Modelo de Domínio para extração da ontologia. Após, os novos conhecimentos obtidos a partir da Especificação de Requisitos devem ser formalizados na ontologia, refinando os conceitos e as definições de propriedades taxonômicas e não-taxonômicas, como *object properties* e *datatype properties*.

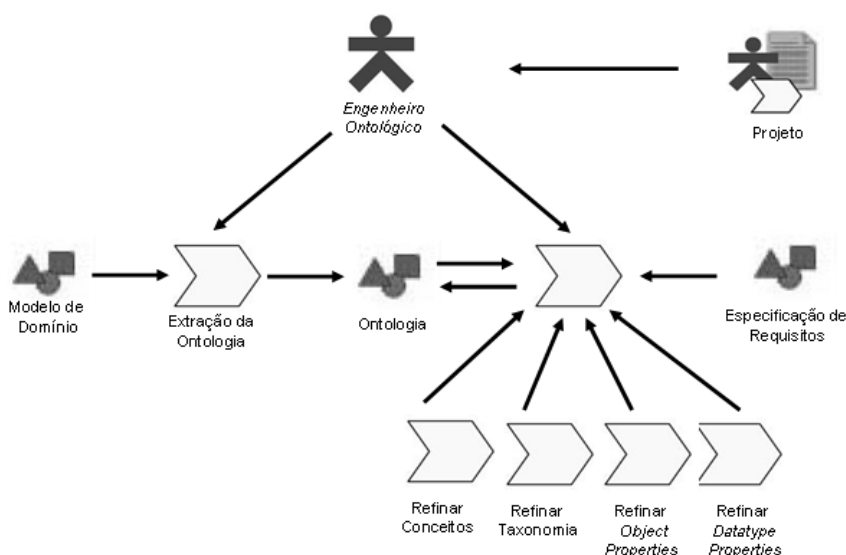


Figura 3.5 – Processos da atividade de projeto.

Terminada a atividade de projeto, tem-se como novo artefato a ontologia que será refinada de acordo com os novos conhecimentos obtidos nas próximas etapas do ciclo de vida do software.

3.3.1.2 Manutenção

A atividade de projeto não produz uma versão da ontologia que define o conhecimento completo sobre o domínio, pois este modelo não consegue mapear automaticamente a maioria das regras lógicas necessárias para restringir o conhecimento. Para resolver este problema, é possível refinar a ontologia utilizando ferramentas específicas para sua manipulação, como, por exemplo, o *Protégé* [PRO06]. O refinamento da ontologia pode ser realizado utilizando à mesma abordagem da fase anterior, baseada na metodologia *On-To-Knowledge*. O objetivo deste refinamento é formalizar o conhecimento adicional adquirido ao longo do processo de desenvolvimento, expressando-o através de regras lógicas sobre a ontologia e novos conceitos e relacionamentos. Como exemplo de regras lógicas, tem-se conjunções, disjunções, quantificação universal e existencial, negações, restrições de número, entre outros.

O principal objetivo desta fase é gerar os elos de rastreabilidade entre os artefatos desenvolvidos durante o projeto de software (Especificação de Requisitos, Modelo de Análise e de Projeto) com os conceitos que compõem a ontologia. Adicionalmente, o processo proposto irá refinar ao modelo ontológico para manutenção de sua integridade, adaptando-o com a evolução projeto.

Durante o processo de desenvolvimento, todos os artefatos gerados e seus componentes poderão ser mapeados através dos conceitos da ontologia, gerando, assim, os elos de rastreabilidade. Desta forma, consegue-se uma gestão integrada do conhecimento sobre os conceitos externos a arquitetura do sistema com as estruturas e processos do projeto de software. Posteriormente, será apresentada a forma como estes elos de rastreabilidade foram estruturados.

A Figura 3.6 representa os processos relacionados à atividade de Manutenção da disciplina de Modelagem do Conhecimento. Da mesma forma que na fase de Projeto, o ator engenheiro do conhecimento será o responsável pela coordenação das atividades desta fase. Ele receberá como entrada a Especificação de Requisitos, o Modelo de Análise, o Modelo de Projeto e a Ontologia. Esta última sofrerá freqüentes atualizações para manutenção de sua integridade ao longo do processo de desenvolvimento. As principais atividades desta fase são: refinar conceitos e taxonomia, identificar novos *object properties* e *datatype properties*,

adicionar as restrições lógicas e relacionar os elementos dos artefatos UML com os conceitos da ontologia. A atividade de manutenção será executada até o término do ciclo de vida do projeto de software.

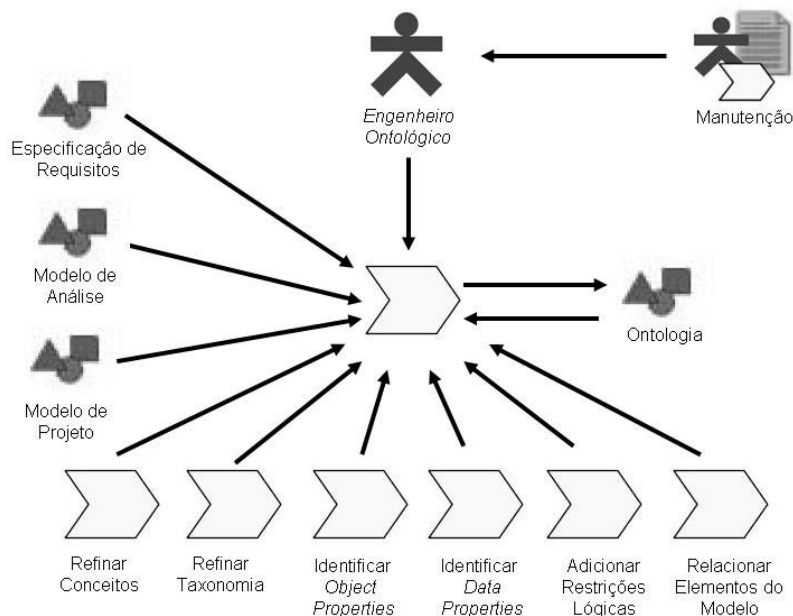


Figura 3.6 – Processos da atividade de manutenção.

3.3.1.3 Avaliação

Esta atividade corresponde à constante avaliação do conhecimento sobre o domínio do problema. O especialista do domínio é responsável por avaliar a integridade, completude e corretude deste conhecimento, definido pelo engenheiro do conhecimento e expresso através da ontologia, de forma análoga à idéia de gestão de conhecimento corporativo. Propõe-se utilizar uma abordagem semelhante às propostas *On-To-Knowledge* e *Helix-Spindle*, em que se verifica a consistência do modelo lógico a cada nova iteração do ciclo de desenvolvimento da ontologia. Esta validação deve ser executada utilizando uma abordagem indutiva e pragmática. Assim, garante-se uma adequada, consistente e coerente representação do vocabulário para a modelagem do domínio. Este processo propõe três tipos de validação para cada etapa do desenvolvimento da ontologia, sendo eles: Teste Unitário, de Integração e de Aceitação. Os dois primeiros devem ser executados pelo engenheiro do conhecimento e o último deve ser executado por agentes de softwares ou outras aplicações que utilizem a ontologia.

3.4 ANÁLISE DA METODOLOGIA SEGUNDO [FER99]

Fernández López desenvolveu em [FER99] um estudo para avaliar diversas metodologias de engenharia ontológica encontradas na literatura. Para tanto, o autor define uma série de critérios:

- C1: **Herança da engenharia do conhecimento:** considerando a influência da tradicional engenharia do conhecimento na metodologia em questão;
- C2: **Detalhamento da metodologia:** consideração de quais atividades e técnicas propostas pela metodologia são especificadas;
- C3: **Recomendação pela formalização do conhecimento:** consideração sobre formalismos lógicos, *frames*, etc.
- C4: **Estratégia para construção de ontologias:** discussão da estratégia adotada: dependente da aplicação, semi-dependente de aplicação ou independente de aplicação;
- C5: **Estratégia para identificação dos conceitos:** definição de estratégias, partindo de uma abordagem mais abstrata para uma mais concreta (*top-down*), da mais concreta para mais abstrata (*bottom-up*) ou da mais relevante para mais abstrata ou mais concreta (*middle-out*);
- C6: **Recomendação do ciclo de vida:** avaliação se a metodologia propõe um ciclo de vida implícito ou explícito para ontologia;
- C7: **Diferenças entre a metodologia e o padrão IEEE 1074-1995 [IEE96]:** comparação da metodologia com o padrão da IEEE para descrição do processo de desenvolvimento de software;
- C8: **Técnicas recomendadas:** especificação de quando técnicas particulares são propostas para serem executadas ao longo da metodologia;
- C9: **Definição das ontologias que foram desenvolvidas utilizando a metodologia em questão e quais sistemas foram construídos utilizando esta ontologia:** descrição das ontologias e dos sistemas que as utilizam.

O objetivo desta seção não é realizar uma validação da presente proposta, mas apenas compará-la aos critérios identificados por Fernández López. Com relação aos critérios estabelecidos, a proposta apresenta:

- C1: Embora não esteja relacionada diretamente com a engenharia do conhecimento, nossa proposta sugere a utilização de uma técnica para

modelagem do conhecimento que claramente identifica a aquisição, codificação e avaliação da ontologia;

- C2: Nossa metodologia especifica exatamente quais as atividades envolvidas com a engenharia ontológica, identificando responsáveis por sua execução. Adicionalmente, foram referenciadas diversas técnicas de outras metodologias que podem ser integradas a nossa proposta;
- C3: A ontologia é formalizada desde suas etapas iniciais. Durante a fase inicial, a versão preliminar da ontologia é mapeada em uma estrutura orientada a objetos (Modelo de Domínio), definindo apenas os conceitos e suas propriedades taxonômicas e não-taxonômicas. Durante a fase de manutenção, a ontologia é formalizada em uma linguagem (OWL) e todas as regras lógicas são implementadas;
- C4: A nossa proposta sugere um forte acoplamento da ontologia com o sistema de informação durante seu processo de desenvolvimento, isto é, dependente da aplicação;
- C5: É sugerida a utilização da abordagem *top-down*, isto é, a identificação de conceitos a partir de uma perspectiva mais abstrata e semi-formal para uma mais concreta e formal. A técnica para identificação dos conceitos sugerida é semelhante à metodologia *On-To-Knowledge*;
- C6: Nossa proposta sugere um ciclo de vida explícito para ontologia, definindo as atividades relacionadas desde sua concepção, projeto, manutenção e avaliação;
- C7: Por ser sugerida como uma disciplina integrada ao Processo Unificado, nossa proposta está relacionada ao padrão da IEEE.
- C8: Durante a especificação das atividades relacionadas com a engenharia ontológica, são sugeridas diversas técnicas e questões de competência como, por exemplo, a vinculação aos cenários motivacionais propostos por [USC95];
- C9: Nossa proposta possui apenas uma ontologia desenvolvida visando sua avaliação. Por ser uma proposta desenvolvida no escopo de uma dissertação de mestrado, sugere-se sua aplicação em domínios diferentes para trabalhos futuros.

3.5 TRABALHOS RELACIONADOS

Guarino, em [GUA98], afirma que todo o sistema de informação possui sua própria ontologia. A questão discutida pelo autor compreende o papel que uma explícita ontologia pode exercer durante o desenvolvimento de software, na qual ela guia a arquitetura utilizando como perspectiva central o domínio do problema. Frente a esta tendência, são encontradas na literatura diversas propostas que sugerem a integração de Engenharia de Software e ontologias.

A [W3C06] vem apoiando uma proposta chamada de *Ontology Driven Architectures* (ODA) [ODA06], que corresponde à aplicação dos benefícios das linguagens para representação do conhecimento nas práticas de Engenharia de Software. O objetivo desta integração é entusiasmar os engenheiros de software com a tecnologia da Web Semântica, encorajando uma colaboração mútua entre as duas comunidades.

Para exemplificar, a OMG vem desenvolvendo a iniciativa MDA (*Model Driven Architecture*) para utilizar os modelos não apenas com propósitos de projeto e manutenção, mas como a base para a geração de artefatos executáveis. O ODM provê a extensão do MDA, habilitando a representação de vocabulários do domínio de forma não ambígua e provendo consistência e validação aos modelos de projeto, utilizando ontologias.

Nesta seção serão apresentadas algumas propostas que sugerem a integração de ontologias com o processo de desenvolvimento de software. Inicialmente, serão apresentados alguns trabalhos desenvolvidos na comunidade científica e, posteriormente, algumas ferramentas que apóiam esta abordagem na indústria.

3.5.1 Estação TABA

A proposta desenvolvida em [OLI00] consiste na construção de ambientes de software orientados a domínio, isto é, apoiar o desenvolvimento de software com base no conhecimento do domínio.

Os autores definem como requisitos para a proposta:

1. É necessária a investigação do domínio durante o desenvolvimento utilizando algum nível de formalização;
2. Permitir ao desenvolvedor trabalhar diretamente com o domínio do problema;
3. Apoiar a identificação, acesso e uso de objetos do conhecimento do domínio;

4. Apoiar o trabalho colaborativo entre usuários (ou especialistas) do domínio e desenvolvedores dado que, com o conhecimento do domínio integrado no ambiente de desenvolvimento, o trabalho com os usuários pode ser mais fácil utilizando um vocabulário comum.

De forma geral, a proposta consiste em mapear o conhecimento do domínio e permitir seu acesso durante o processo de desenvolvimento de software. Os autores sugerem uma nova sub-atividade no processo de desenvolvimento, chamada "investigação do domínio", semelhante à modelagem de negócios do Processo Unificado. Para especificar o domínio, é sugerido o uso de ontologias. A partir da ontologia de domínio, é sugerida a derivação de algumas entidades arquiteturais, tais como modelagem de dados e objetos.

A proposta da estação TABA se relaciona com o nosso trabalho através da criação da sub-atividade necessária à vinculação do modelo conceitual com as demais etapas do desenvolvimento de software. Em nossa proposta, é sugerida a integração de ontologias no Processo Unificado pela criação de uma nova disciplina, visando apoiar o processo de desenvolvimento e não na geração da arquitetura da solução pela derivação do modelo lógico.

Nossa proposta enfatiza o desenvolvimento de uma ontologia utilizando a linguagem sugerida pela [W3C06], através do mapeamento do Modelo de Domínio em uma ontologia, utilizando como guia a proposta definida pelo [OWG06]. Adicionalmente, é apresentado um processo detalhado para a criação da ontologia, baseado em diversas propostas de engenharia ontológica bem aceitas na literatura, definindo assim todas as entidades e regras lógicas que compõem a ontologia.

3.5.2 ODE: Ontology-based software Development Environment

Falbo, em [FAL04], apresenta ODE (*Ontology-based software Development Environment*), um ambiente para desenvolvimento de software baseado em ontologias. ODE possui como base quatro distintas ontologias para processo, qualidade, artefatos e riscos de software com o objetivo de estruturar o ambiente e a infra-estrutura da ferramenta. A proposta consiste na derivação sistemática de infra-estruturas de objetos a partir de ontologias, isto é, a derivação dos conceitos da ontologia em um modelo de objetos.

São definidos três níveis para manter a amarração semântica entre os objetos de ODE:

- **Nível ontológico** (ontologia): é responsável pela descrição da ontologia propriamente dita, isto é, o modelo conceitual que especifica o domínio da aplicação;
- **Meta-nível** (conhecimento): representa a derivação das classes e instâncias do nível ontológico, descrevendo o conhecimento do ambiente e do domínio da aplicação;
- **Nível base** (controle): define as classes responsáveis pela implementação das aplicações no contexto do ambiente, sob a perspectiva da arquitetura do software. Estas classes são derivadas da ontologia base.

A proposta do autor consiste em uma nova abordagem para o desenvolvimento de software, baseada na evolução de uma ontologia que especifica um modelo conceitual até a arquitetura que implementa o software. A proposta é semelhante à Estação TABA, porém com maior ênfase na construção de ontologias e na sistemática na derivação do sistema de informação.

A proposta sugerida por Falbo é alternativa ao Processo Unificado, divergindo da apresentada neste trabalho. A nossa proposta consiste na integração de ontologias ao Processo Unificado, aproveitando o esforço da aquisição do conhecimento do domínio e provendo os benefícios que uma linguagem lógica oferece para apoiar o ciclo de vida do software.

3.5.3 Proposta de Oberle

Oberle, em [OBE04], sugere o desenvolvimento e gerenciamento de componentes de software em um servidor de aplicação baseado em ontologias, isto é, a utilização de ontologias para mapear a descrição de componentes de software em um servidor de aplicação. A idéia compreende em carregar os meta-dados semânticos e a ontologia em um motor de inferência e realizar a recuperação de componentes a partir de suas descrições.

Para a descrição de um componente utilizando ontologias, é necessário o conhecimento implícito do domínio do problema que o componente de software implementa. A abordagem proposta pelo autor não apresenta como este conhecimento é evoluído em uma arquitetura, apenas sugere como aproveitar a oportunidade da descrição do componente no contexto de um servidor de aplicação. Nossa proposta é complementar a de [OBE04], visto que se preocupa com o ciclo de vida completo para construção do componente e sua ontologia associada.

3.5.4 Ferramentas para desenvolvimento integrado a ontologias

Da mesma forma que é observada uma tendência na comunidade científica para integrar ontologias ao desenvolvimento de software, a comunidade industrial também vem apoiando esta idéia. A IBM vem participando ativamente no desenvolvimento da proposta ODM do [OWG06] e apresenta o *IBM Integrated Ontology Development Toolkit* [IOD06]. Esta ferramenta é integrada ao Eclipse [ECL06] e apóia a transformação de ontologia em outros modelos orientados a objeto, além da manipulação de modelos lógicos, como OWL e RDF. Apesar do suporte para o mapeamento da ontologia em outros modelos, esta ferramenta não provê o apoio a rastreabilidade e a integração dos conceitos do domínio com todos os elementos UML do projeto de software, apresentado em nossa proposta.

Existem propostas de ferramentas exclusivas para manipulação de ontologias, como *Protégé* [PRO06], *OILEd* [BEC01] e *OntoEdit* [SUR02], porém não serão apresentados detalhes neste trabalho. O foco aqui é apresentar ferramentas que integrem ontologias com a Engenharia de Software, e não exclusivamente uma das vertentes.

3.6 CONSIDERAÇÕES

Apesar de algumas diferenças entre o paradigma orientado a objetos e o ontológico, neste trabalho é sugerida uma sinergia entre suas engenharias, apoiando a proposta de integração de ontologias no Processo Unificado. O objetivo desta integração não é sugerir uma proposta inovadora para construção de ontologias, mas sim aproveitar os esforços despendidos durante o processo de desenvolvimento de software na construção de ontologias. Além disso, nossa proposta tem como base diversas metodologias de engenharia ontológica aceitas na comunidade científica, integrando suas sistemáticas e boas práticas.

Conforme já apresentado, muito do esforço relativo à construção de ontologias está relacionado com a identificação do propósito, explicitação do domínio, delimitação do escopo e modelagem conceitual. Os esforços iniciais do Processo Unificado compreendem aspectos relacionados a estes, onde os *stakeholders* realizam a compreensão do domínio e a modelagem de negócios. É com base nesta premissa que nossa proposta se apóia. Por essa integração, muito do esforço despendido com a engenharia ontológica pode ser salvo através de uma Engenharia de Software com qualidade.

Para exemplificar, toma-se como exemplo os cenários motivacionais proposto por [USC95] e apoiados por [GRU95], [LOP97] e [FAL98]. Estes cenários podem corresponder a casos de uso do processo de desenvolvimento de software que, sob uma perspectiva de negócios, estabelecem um conjunto de problemas que devem ser resolvidos através de uma modelagem conceitual. Adicionalmente, devido à similaridade entre o modelo ontológico e o modelo de domínio, é possível formalizar uma estrutura que representa as bases da ontologia.

Por outro lado, integrando ontologias ao desenvolvimento de software, se possibilita uma gestão integrada dos conceitos do problema com as diversas visões que compõem um sistema de informação, incluindo uma perspectiva de negócios, requisitos, análise e projeto. Por gestão integrada, compreende-se o relacionamento dos conceitos do sistema com a arquitetura que especifica sua solução, visando recuperar elos de rastreabilidade que representam o conhecimento tácito obtido durante o processo de desenvolvimento. Posteriormente, será apresentada a proposta de estruturação e recuperação destes elos.

4 RASTREABILIDADE ONTOLÓGICA

A rastreabilidade é a capacidade de inter-relacionar e recuperar entidades que se identificam durante um processo. No processo de desenvolvimento, o termo rastreabilidade diz respeito ao relacionamento entre os requisitos do software com as necessidades dos usuários e as entidades que compõem a solução computacional. Utilizando ontologias, é possível relacionar os conceitos pertinentes às necessidades dos usuários com os artefatos desenvolvidos durante o processo de desenvolvimento. Este capítulo apresenta algumas dimensões para classificação de elos de rastreabilidade e uma proposta para estruturar estes elos, utilizando um modelo lógico. O objetivo é a redução da granularidade destes elos, de requisitos para conceitos, visando recuperar informações mais apuradas. Para verificar a viabilidade e aplicabilidade da proposta descrita, é necessário o desenvolvimento de uma ferramenta que apóie tanto a engenharia do conhecimento quanto a Engenharia de Software. De posse de uma ontologia integrada ao processo de desenvolvimento, torna-se possível a proposta de rastreabilidade indexada por conceitos. Este capítulo adicionalmente apresenta a ferramenta desenvolvida para automatizar a rastreabilidade ontológica ao longo do ciclo de vida de um sistema de informação.

4.1 RASTREABILIDADE

Um ponto fundamental para rastreabilidade consiste em perceber um sistema de informação como um conjunto de documentos e artefatos, que descrevem o software em diferentes níveis de abstração, e não apenas como uma implementação. Neste contexto, [PFL90] define rastreabilidade como a habilidade de relacionar os itens dependentes dentro de um modelo e seus correspondentes em diferentes modelos.

De acordo com [LIN94], rastreabilidade é uma propriedade de um sistema e um fator de qualidade que corresponde ao entendimento do software e ao acompanhamento da documentação de seus modelos. Rastreabilidade, em uma perspectiva mais ampla, corresponde à habilidade de relacionar e entender completamente o processo de desenvolvimento e seus artefatos.

A rastreabilidade pode prover um importante apoio para o desenvolvimento de software e sua manutenção, apoiando tanto uma evolução *top-down* quanto *bottom-up*. Permite-se uma compreensão do sistema, apoiando mudanças para sua manutenção, incluindo análise de impacto e reuso de softwares existentes. Adicionalmente, esta técnica provê um apoio essencial para o entendimento da organização estrutural entre os requisitos e seus artefatos derivados [PAL00].

O grau de rastreabilidade pode ser obtido em diferentes níveis, desde uma convenção para nomes de documentos até elos de rastreabilidade explícitos, que relacionam

itens de modelos. Dentro desta última perspectiva, existem diferentes dimensões que podem ser avaliadas para classificar os relacionamentos entre os modelos.

4.1.1 Dimensões de Rastreabilidade

A rastreabilidade possui diversas classificações bem aceitas na literatura para avaliar a qualidade e o alcance dos relacionamentos. A idéia desta seção é introduzir aspectos específicos de rastreabilidade para conduzir nosso estudo, identificando oportunidades que qualifiquem a rastreabilidade semântica.

4.1.1.1 Rastreabilidade Vertical e Horizontal

A primeira dimensão de rastreabilidade diz respeito à interconexão dos itens relativos aos modelos. Nesta perspectiva, [PFL90] apresenta as seguintes métricas:

- **Rastreabilidade Vertical:** os itens relacionados pertencem ao mesmo artefato;
- **Rastreabilidade Horizontal:** os itens relacionados pertencem a artefatos diferentes.

Ambos os tipos de rastreabilidade são necessários para compreender um conjunto completo de relacionamentos pertinentes a uma análise de impacto, por exemplo. Adicionalmente, esta divisão permite uma visão para o processo de manutenção de um conjunto de artefatos e seus respectivos ciclos de vida.

Para o mapeamento, tanto vertical quanto horizontal, a autora sugere o uso de grafos, onde um conjunto de objetos corresponde aos nodos e os relacionamentos correspondem a arestas. É sugerida uma modularização entre requisitos, projeto, código e teste que estruturam uma organização vertical de seus componentes, e estes componentes se relacionam horizontalmente com os demais módulos. A Figura 4.1 apresenta a proposta de organização horizontal entre componentes de software.

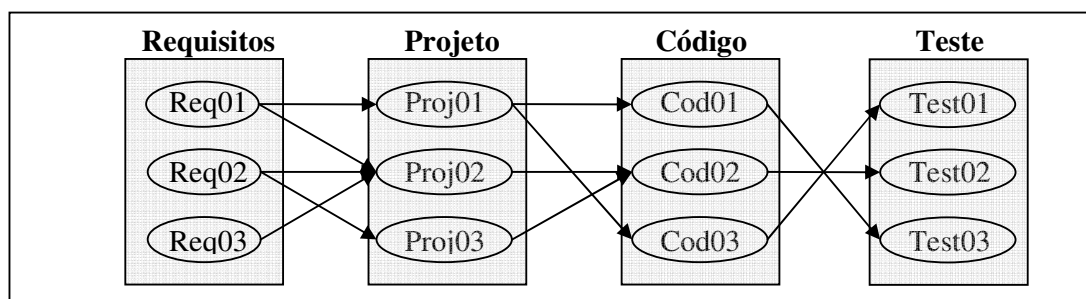


Figura 4.1 – Rastreabilidade em Artefatos de Software.

4.1.1.2 Pré-Rastreabilidade e Pós-Rastreabilidade

Um termo mais comum para rastreabilidade é apresentado como *Rastreabilidade de Requisitos*, que corresponde à habilidade de relacionar requisitos específicos com suas implementações [GOT94]. Nesta perspectiva, Gotel e Finkelstein, apresentam uma divisão de rastreabilidade em duas principais dimensões:

- **Pré-Rastreabilidade:** capacidade de rastrear, a partir dos requisitos, as necessidades dos usuários;
- **Pós-Rastreabilidade:** capacidade de recuperar como cada requisito é implementado.

Dentre estas duas classificações, Finkelstein argumenta sobre a dificuldade da pré-rastreabilidade [FIN91], devido às inconsistências características das fases iniciais de entendimento do domínio e elicitação de requisitos.

4.1.1.3 Relacionamentos Explícitos e Implícitos

Lindvall, em [LIN94], nos apresenta duas classificações para rastreabilidade levando em conta o tipo de relacionamento entre os elementos, que são:

- **Relacionamento Explícito:** representa a capacidade de recuperar relacionamentos entre dois modelos utilizando elos previamente definidos;
- **Relacionamento Implícito:** em oposição ao relacionamento explícito, representam associações organizadas por convenções como, por exemplo, padronização para nomes de entidades.

4.1.1.4 Relacionamentos Estruturais e Cognitivos

Uma abordagem explorada por [BIA00] diz respeito a relações obtidas através de sintaxe e semântica dos modelos, que consistem:

- **Relacionamentos Estruturais:** corresponde a rastreabilidade associada a uma dependência sintática entre alguns artefatos, originadas a partir do código fonte;
- **Relacionamentos Cognitivos:** dizem respeito aos elos de rastreabilidade obtidos através da semântica do conhecimento, associadas às decisões de projeto [RUG90].

Frente a estas quatro propostas apresentadas, o Quadro 4.1 apresenta uma categorização das dimensões. Posteriormente, estas categorias serão utilizadas para classificar a rastreabilidade semântica.

Quadro 4.1 – Dimensões de Rastreabilidade.

Dimensão	Categoria	
D1	Vertical	Horizontal
D2	Pré-Rastreabilidade	Pós-Rastreabilidade
D3	Explícitos	Implícitos
D4	Estruturais	Cognitivos

4.1.1.5 Modelagem da Rastreabilidade

De acordo com [LEI03b], a rastreabilidade impõe o chamado de *Princípio de Circularidade*, isto é, é possível rastrear apenas aquilo que foi registrado e é possível registrar apenas aquilo que se percebe um rastro.

Para a modelagem da rastreabilidade, são necessários três aspectos:

- **Definição:** corresponde à especificação dos rastros;
- **Produção:** corresponde à captura dos rastros, através de um registro explícito do relacionamento entre os itens;
- **Extração dos elos:** corresponde a recuperação dos relacionamentos registrados.

A utilização de matriz de rastreabilidade para indexar requisitos aos demais componentes de software é uma técnica bastante aceita. Uma matriz de rastreabilidade é uma tabela que correlaciona itens de projeto entre si para manutenção da completude de seus relacionamentos.

Diversas ferramentas provêem uma automação na elaboração desta matriz, conforme verificado em [INC06]. A estrutura básica de uma matriz de rastreabilidade consiste em uma organização dos requisitos em colunas e uma lista de artefatos em linhas. Para cada registro de associação entre os requisitos e os artefatos, é registrada a associação na célula correspondente da matriz.

4.2 RASTREABILIDADE ONTOLÓGICA

No capítulo anterior, foi apresentada uma proposta de processo para a construção de uma ontologia relativa a um sistema de informação. Neste processo, foram explicitamente definidos todos os conceitos e seus relacionamentos, incluindo propriedades taxonômicas e não-taxonômicas. Entretanto, o principal objetivo de se desenvolver uma ontologia é a possibilidade de relacionar todos os artefatos e seus elementos aos conceitos da ontologia durante o processo de desenvolvimento de software.

Utilizando ontologias, torna-se possível a recuperação de elos de rastreabilidade implícitos e cognitivos, utilizando um motor de inferência. Para permitir um mapeamento conceito-artefato, é proposta uma estrutura ontológica organizada sobre um conceito chamado *ONTraceBasis*. O conceito *ONTraceBasis* é um recurso que sustenta os mecanismos que provêm a rastreabilidade ontológica entre os artefatos. Sobre este conceito estão estruturados outros dois:

- **Artifacts:** é uma classe que representa uma instância de um elemento UML, tal como um caso de uso, classes em diagrama de classes, etc., geradas durante a modelagem do software. Este artefato é caracterizado por duas propriedades do tipo *datatype*:
 - *ontraceRecover*: esta propriedade relaciona a instância do elemento UML, definida pela classe *Artifacts*, aos conceitos da ontologia, representados por classes OWL;
 - *ontraceArtifactsTypedBy*: representa o tipo do artefato, definido pelo conceito *Model*, apresentado a seguir;
- **Model:** [RUM98] definiu todos os tipos previstos pela UML. Estes tipos estão definidos dentro desta estrutura ontológica. A classe *Model* organiza hierarquicamente os tipos conforme área, visão, diagramas e conceitos. O relacionamento dos artefatos com seus respectivos tipos é importante para filtros de consulta, retornando apenas elementos como, por exemplo, casos de uso ou classes. O Anexo A apresenta os componentes UML definidos pelo autor.

A estrutura de rastreabilidade é incluída na ontologia gerada a partir do modelo de domínio e, para cada elo de rastreabilidade, é criado ou atualizado uma instância do recurso *Artifacts*. Estas instâncias estão relacionadas a conceitos da ontologia que serão utilizados

como índice para recuperação de elos de rastreabilidade. A Figura 4.2 apresenta um exemplo de estrutura instanciada, utilizando a notação provida pela [OWG06] para representação de ontologias.

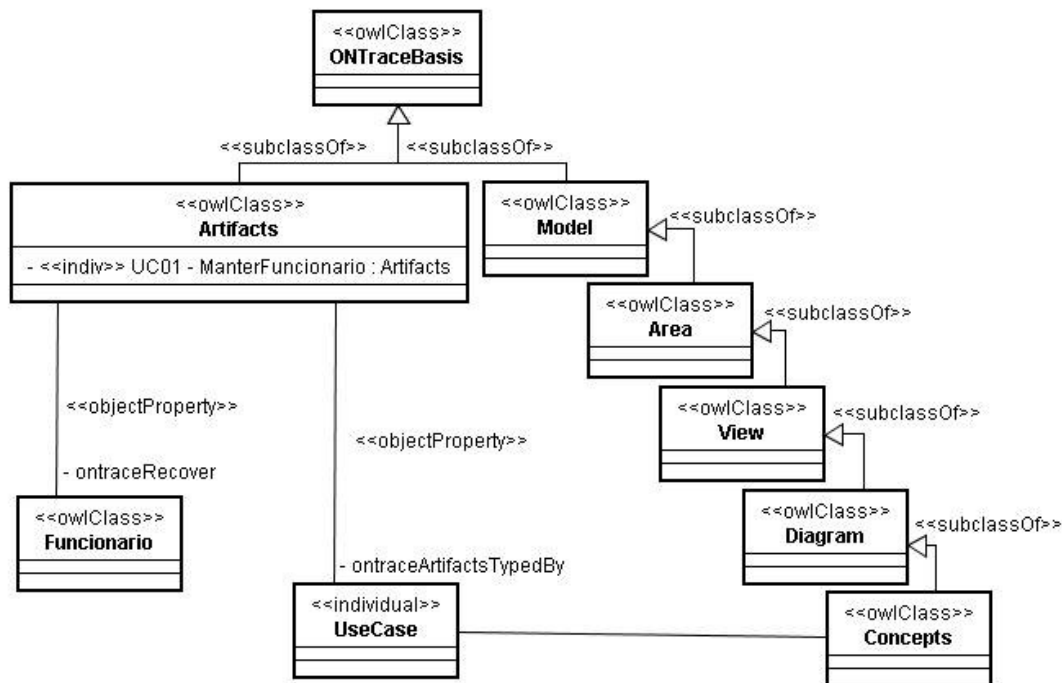


Figura 4.2 – Estrutura Ontológica para Rastreabilidade.

A Figura 4.2 exemplifica um elo de rastreabilidade entre o caso de uso “*UC01 – Manter Funcionário*” e o conceito da ontologia “*Funcionário*” através da propriedade “*ontraceRecover*”. Adicionalmente, este elo especifica que este indivíduo é do tipo “*UseCase*”, que é um dos tipos definidos pela taxonomia estabelecida sobre a classe “*Model*”.

Para cada estrutura do modelo que se deseja indexar pela ontologia, é gerado um novo indivíduo no modelo lógico que o relaciona a um ou mais conceitos. Para exemplificar a sistemática da proposta, será apresentado um exemplo extraído de um cenário de loja virtual.

Supondo que:

- O caso de uso “*Manter Cliente*” esteja relacionado ao conceito da ontologia “*Cliente*”;
- O caso de uso “*Registro de Compra*” esteja relacionado aos conceitos da ontologia “*Funcionário*”, “*Produto*” e “*Cliente*”.

A Figura 4.3 apresenta o cenário relacionando o diagrama de casos de uso com os conceitos da ontologia, expressos em OWL.

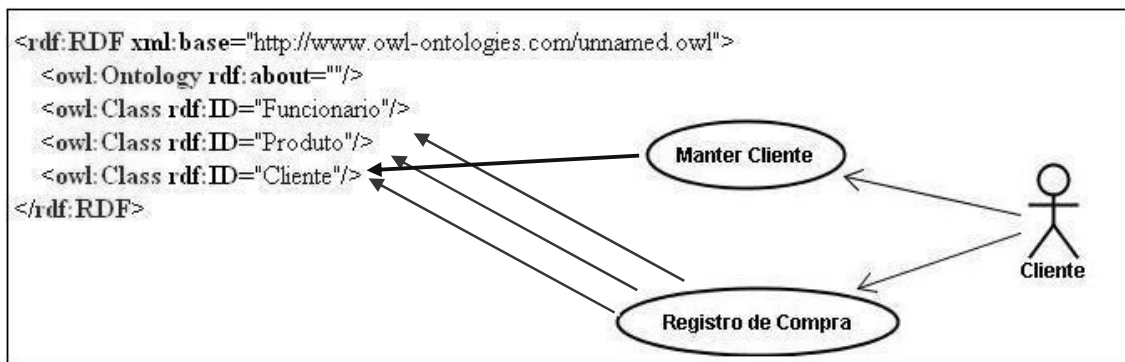


Figura 4.3 – Primeiro cenário do exemplo de rastreabilidade ontológica.

Então é possível dizer que os casos de uso “*Manter Cliente*” e “*Registro de Compra*” estão relacionados diretamente através do conceito “*Cliente*”.

No mesmo exemplo, supondo o conceito “*Funcionário*” se relacione com o conceito “*Cliente*” através de uma propriedade chamada “*recuperaInformacao*”. Sabendo que o caso de uso “*Manter Funcionário*” está relacionado com o conceito da ontologia “*Funcionário*”. Com esta configuração, ilustrada pela Figura 4.4, é possível inferir uma nova informação sobre os artefatos: o caso de uso “*Manter Cliente*” está relacionado ao caso de uso “*Manter Funcionário*” porque os conceitos da ontologia que relacionam ambos estão associados através de uma propriedade. Esta propriedade indica que um item conceitual é relacionado a outro e que os elementos associados podem exercer impacto um sobre o outro. Resumindo, diante da configuração:

- O caso de uso “*Manter Cliente*” está relacionado ao conceito da ontologia “*Cliente*”;
- O caso de uso “*Manter Funcionário*” está relacionado ao conceito da ontologia “*Funcionário*”;
- O conceito “*Funcionário*” está relacionado com o conceito “*Cliente*” através de uma propriedade chamada “*recuperaInformacao*”.

É possível dizer que o caso de uso “*Manter Cliente*” está relacionado ao caso de uso “*Manter Funcionário*”. Esta relação é indireta e pode ser recuperada através de regras de inferência, justificando o uso de ontologias para rastreabilidade.

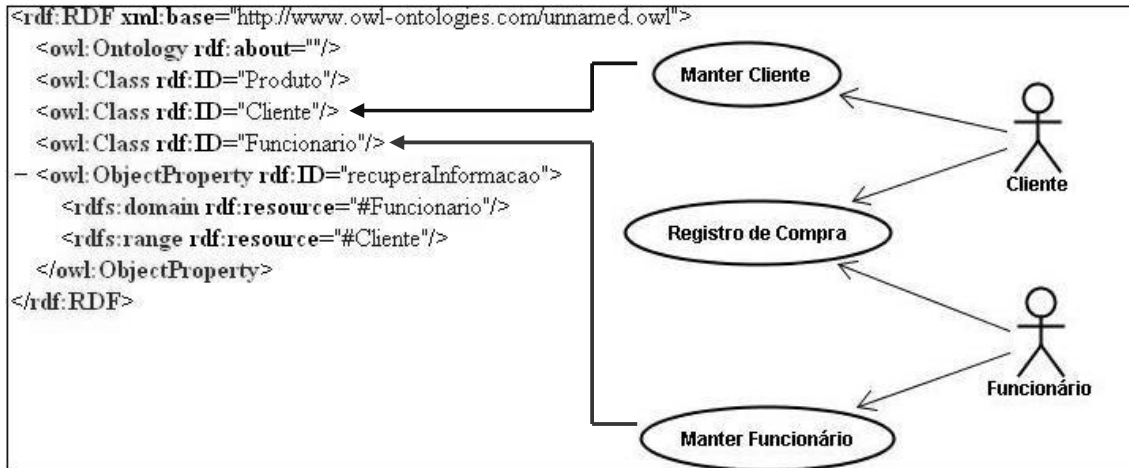


Figura 4.4 – Segundo cenário do exemplo de rastreabilidade ontológica.

Utilizando esta abordagem, é possível relacionar o sistema inteiro (modelos e seus elementos) aos conceitos da ontologia e recuperá-los através de relacionamentos diretos e indiretos.

Nossa proposta consiste na elaboração de um grafo de rastreabilidade que não relacione apenas os elementos arquiteturais com os requisitos, mas sim com os conceitos que definem o domínio do problema. A idéia da utilização de grafos para este fim também é sugerida por [PFL90]. A idéia de rastreabilidade compreende a possibilidade de criar indivíduos da ontologia que relacionem as visões de negócios, análise e projeto do sistema com os conceitos da ontologia.

Para suportar o processo de modelagem do conhecimento e a capacidade de aquisição dos elos de rastreabilidade do projeto de software, foi desenvolvido um protótipo para automatizar a proposta, que será apresentado no próximo capítulo.

4.3 AVALIAÇÃO DAS DIMENSÕES DE RASTREABILIDADE ONTOLÓGICA

Dentre as propostas apresentadas para classificações de rastreabilidade, foram apresentadas quatro dimensões que definem tipos de rastros entre os artefatos. Utilizando um grafo que relacione conceitos de uma ontologia com artefatos do sistema, as seguintes dimensões foram caracterizadas:

- Rastreabilidade Vertical: é possível indexar elementos do mesmo artefato a diferentes conceitos da ontologia;

- Rastreabilidade Horizontal: é possível indexar elementos de diferentes artefatos a diferentes conceitos da ontologia;
- Pré-Rastreabilidade: as necessidades dos usuários estão mapeadas em conceitos do domínio através da ontologia, portanto não é necessário fazer a regressão de requisitos para estes conceitos;
- Pós-Rastreabilidade: é possível recuperar como cada conceito foi implementado através de sua correta indexação;
- Relacionamentos Explícitos: é possível recuperar relacionamentos através dos indivíduos da ontologia que representam explicitamente os elos;
- Relacionamentos Implícitos: através de convenções, é possível a recuperação de relacionamentos implícitos. Adicionalmente, a ontologia apresenta regras de inferências que mapeiam o domínio, possibilitando a recuperação de relacionamentos implícitos, tais como transitividade, equivalência, consistência, classificação, etc., utilizando um motor de inferência;
- Relacionamentos Estruturais: a dependência estrutural entre os elementos do modelo é mapeada em uma taxonomia na ontologia, que indexa os itens de projeto. Para exemplificar, a alteração em uma superclasse UML pode impactar suas subclasses. A estrutura UML é mapeada em conceitos organizados hierarquicamente na ontologia, e o impacto pode ser obtido através da recuperação dos relacionamentos indiretos apresentados na Seção 4.2;
- Relacionamentos Cognitivos: ontologias especificam explicitamente a semântica de um domínio, correspondendo ao conhecimento do problema. Ao se indexar os elementos de projeto aos conceitos, e estes conceitos estando organizados sobre regras lógicas, é possível recuperar relacionamentos cognitivos. Esta abordagem não suporta o relacionamento de decisões de projeto aos conceitos, apenas estruturas que modelam o sistema de informação.

Com a proposta de rastreabilidade ontológica, é possível cobrir todas as dimensões apresentadas. Na próxima seção, será apresentada uma ferramenta que apóia nossa proposta e, no capítulo seguinte, um estudo que a avalia experimentalmente.

4.4 FERRAMENTA PARA RASTREABILIDADE ONTOLÓGICA

Para a integração de um modelo lógico a um ambiente de modelagem de sistemas, foi identificada a necessidade de ampliar as funcionalidades das tradicionais ferramentas CASE (*Computer-Aided Software Engineering*).

Inicialmente, foi desenvolvido um levantamento das ferramentas *open-source* de modelagem, identificando como as mais populares:

- ArgoUML [ARG06];
- Bouml [BOU06];
- ESS-Model [ESS06];
- Fujaba [FUJ06];
- Umbrello UML Modeler [UMB06].

Após a análise de aspectos como popularidade, eficiência, estabilidade e qualidade do código fonte, optou-se pelo ArgoUML como ferramenta padrão. Paralelamente a avaliação das ferramentas, uma análise de viabilidade inicial foi desenvolvida para proposta de rastreabilidade ontológica.

4.4.1 Análise de Viabilidade Inicial

Inicialmente, a análise de viabilidade foi desenvolvida com o objetivo de validar a recuperação de elos de rastreabilidade entre conceitos da ontologia e os itens do projeto de software. Para tanto, a análise foi estruturada em quatro etapas:

1. Definição de um sistema de informação;
2. Construção da ontologia;
3. Criação dos elos de rastreabilidade;
4. Recuperação destes elos.

Para exemplificar nossa proposta, foi escolhido um sistema modelado e desenvolvido por alunos da graduação durante uma disciplina da Faculdade de Informática da PUCRS no domínio de uma academia desportiva. O objetivo desta escolha é a utilização de um sistema elaborado por projetistas independentes, que não possuem conhecimento desta proposta.

Após a escolha do sistema, foi gerada uma ontologia utilizando o *Protégé*. Esta ontologia foi extraída a partir do Modelo de Domínio, utilizando o critério de mapeamento

expresso no Quadro 3.2. A estrutura de rastreabilidade proposta na Seção 4.2 também foi adicionada ao modelo lógico, possibilitando a rastreabilidade semântica.

Com a ontologia desenvolvida, foi possível a criação de indivíduos que representam elos de rastreabilidade. Para cada elemento UML do sistema modelado, foi desenvolvido um novo indivíduo na ontologia que indica o seu tipo e o relaciona aos conceitos pertinentes.

A atividade final da análise de viabilidade foi à utilização de um motor de inferência para manipulação da ontologia e realização de consulta sobre as declarações. Utilizando o *framework* Jena [JEN06], foi possível a recuperação dos elos entre os diversos componentes do sistema. Foi desenvolvida uma aplicação que realiza a seguinte sistemática:

1. Recuperar um conjunto de conceitos primários que se relacionam com determinado artefato do sistema, sabendo que um artefato pode possuir mais de um conceito associado;
2. Para cada conceito primário, recuperar o conjunto de artefatos que se relacionam com este conceito.
3. Para cada conceito primário, recuperar o conjunto de conceitos secundários, isto é, aqueles conceitos que estão relacionados com cada conceito primário;
4. Para cada conceito secundário, recuperar o conjunto de artefatos que se relacionam a este conceito.

As atividades relacionadas com a recuperação do relacionamento *artefato-conceito-artefato* estão representadas pelo diagrama de atividades expresso na Figura 4.5. A análise de viabilidade foi desenvolvida como uma aplicação Web, chamada de *ONTrace* (*Ontological Tracing*), e sua interface é apresentada na Figura 4.6.

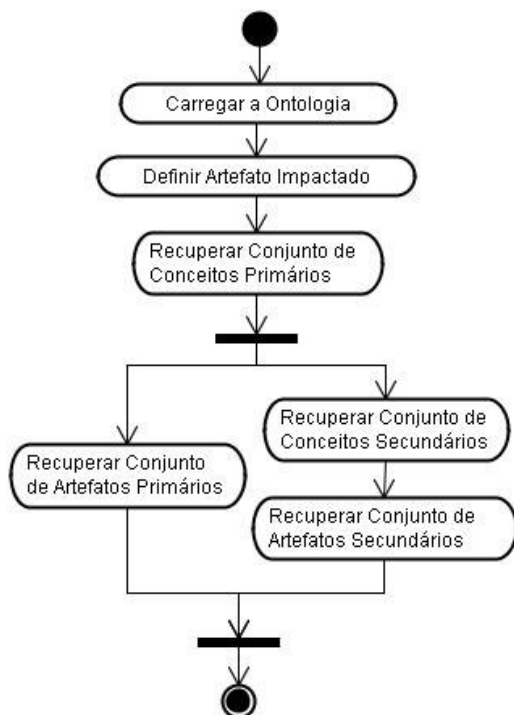


Figura 4.5 – Diagrama de Atividades para inferência.

Instâncias	Ontologia			Instâncias
Artefato Inicial	Conceito Primário	Propriedade	Conceito Secundário	Artefato Final
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC02-EfetuarCompra
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC02.1-CadastrarOrcamento
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC02.2-AlterarUltimoOrcamento
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC02.3-CancelarCompra
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC02.4-EfetuarCompra
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03-ManterFichaDeAcompanhamentoDeCliente
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.1-CriarFichaPessoal
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.2-AlterarFichaDeBiometria
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.3-AlterarFichaPessoal
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.4-AlterarUltimaFichaDeBiometria
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.5-CriarAtividadeDeTreinamento
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC03.6-AlterarUltimaAtividadeDeTreinamento
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC04-AcessarFichaDeAcompanhamentoDeCliente
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC04.1-AcessarFichaPessoal
UC01-EmitirPedidoDeCompra	Funcionario	mantemCadastro	Funcionario	UC04.2-AcessarFichaDeBiometria

Figura 4.6 – Interface da ferramenta para análise de viabilidade.

Depois de verificada a viabilidade da solução, a próxima etapa consistiu na evolução de uma ferramenta de modelagem para suportar nossa proposta.

4.4.2 ArgoUML+ONTrace

ArgoUML+ONTrace corresponde a uma ferramenta de modelagem com apoio à rastreabilidade ontológica. Conforme apresentado, após a análise de diversas ferramentas para modelagem, optou-se por desenvolver novas funcionalidades sobre o ArgoUML. A Figura 4.7 apresenta a interface da ferramenta ArgoUML+ONTrace.

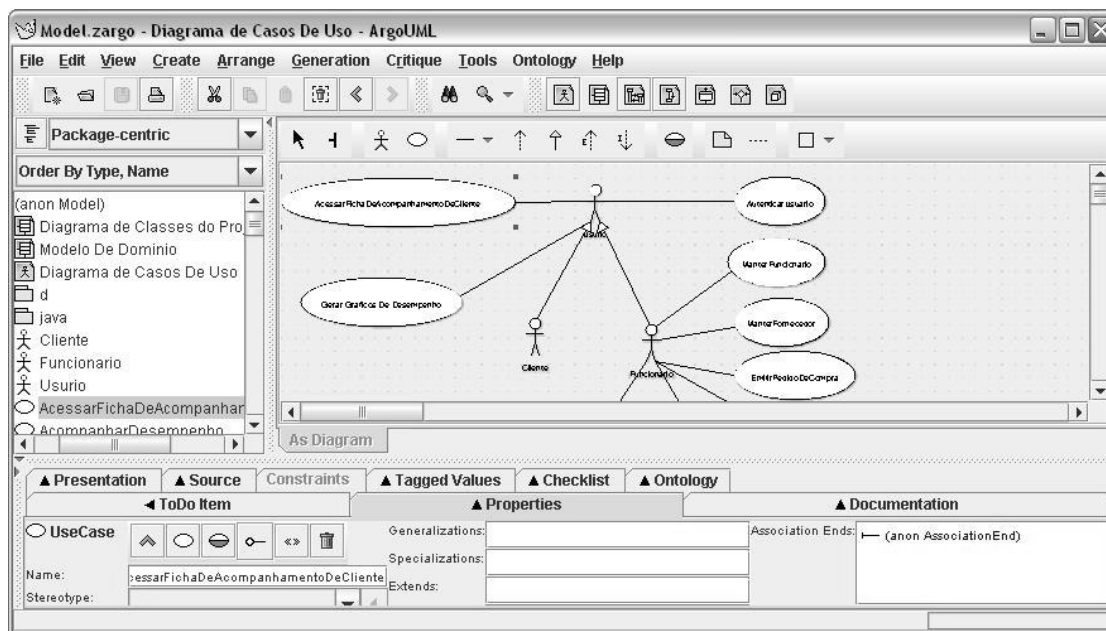


Figura 4.7 – Interface da ferramenta ArgoUML+ONTrace.

A arquitetura da funcionalidade desenvolvida é apresentada pelo diagrama de classes, definido pela Figura 4.8.

Existem três classes de interface:

- *DialogGenerateOntology*: responsável pela geração da ontologia a partir de um diagrama de classes;
- *TabOntology*: responsável por apresentar os recursos da ontologia, que podem ser associados aos elementos UML do projeto de software
- *DialogOntrace*: responsável pela recuperação dos elos de rastreabilidade.

A classe *OntologyPersister* é responsável pela tradução de um diagrama de classes em UML para um modelo lógico em OWL, realizando a persistência da ontologia. Esta classe cria uma nova instância da classe *OntModel*, pertencente à API (*Application Programming Interface*) do Jena, que manipula as informações da ontologia expressa em OWL.

A classe *Ontrace* é responsável pela inferência sobre a ontologia, recuperando os elos de rastreabilidade. Ela recupera informações de relacionamentos diretos (conjunto primário de artefatos) ou indiretos (conjunto secundário de artefatos), utilizando um modelo de inferência pertencente ao Jena e chamado de *InfModel*.

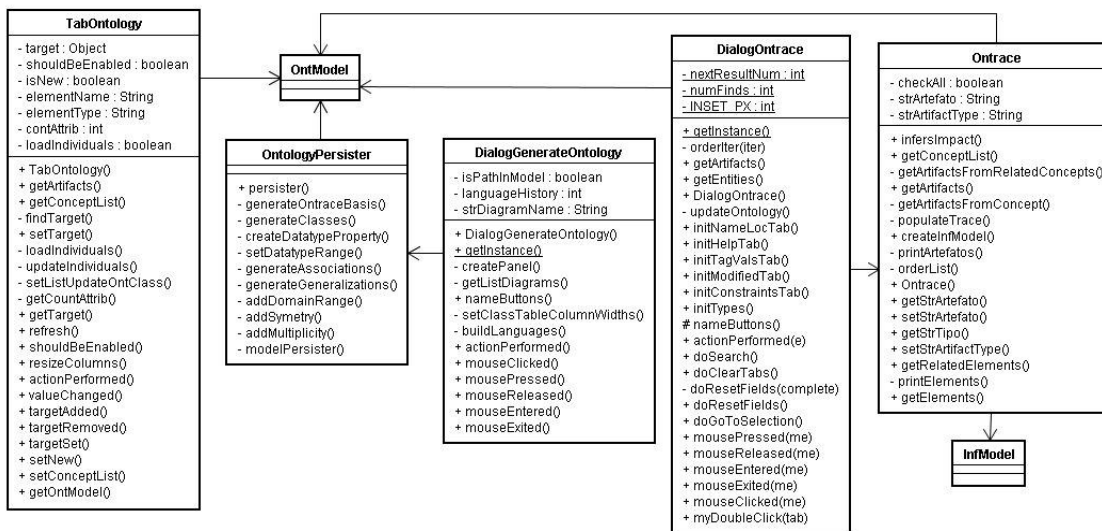


Figura 4.8 – Diagrama de Classes com as novas funcionalidades.

Para a geração automática da ontologia, a partir de um diagrama de classes, é realizado o mapeamento apresentado no Quadro 3.2. A Figura 4.9 representa a interface para geração da ontologia, no qual o usuário seleciona no menu superior (“*Ontology*”) a opção “*Generate OWL*” e escolhe, entre todos os diagramas de classes do projeto, qual representa o Modelo de Domínio. Esta etapa corresponde à transição da atividade de Projeto para Manutenção da disciplina de Modelagem do Conhecimento.

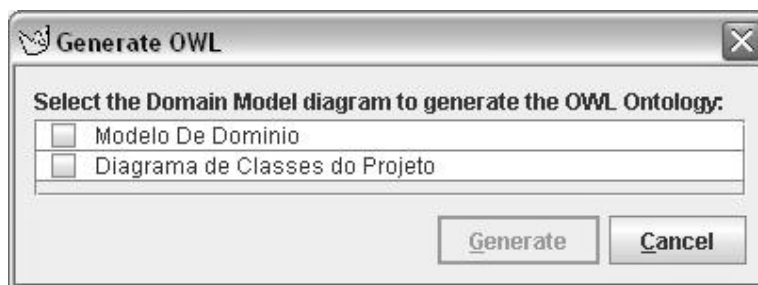


Figura 4.9 – Interface para geração da ontologia.

O processo computacional é iniciado pela classe de interface *DialogGenerateOntology* e realizado pela classe *OntologyPersister*, que cria um novo modelo ontológico (*OntModel*). O procedimento de geração do modelo consiste em percorrer o diagrama de classes, identificando suas classes, associações e generalizações para geração das primitivas equivalentes em OWL. Por fim, é realizada a persistência do modelo no projeto em

execução. A Figura 4.10 ilustra o diagrama de colaboração que representa o processo de geração da ontologia.

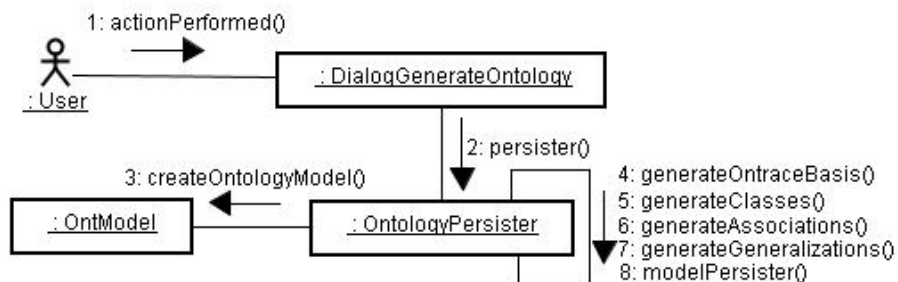


Figura 4.10 – Diagrama de colaboração para geração da ontologia.

A ontologia pode ser exportada e importada do ArgoUML+ONTrace a qualquer momento, em conformidade com a atividade de Manutenção. No menu superior da ferramenta, chamado “Ontology”, existe a opção “*Export ontology as OWL*” e “*Import OWL ONTrace ontology*”, conforme ilustração da Figura 4.11. O objetivo deste procedimento é refinar a ontologia para representar todas as restrições que especificam o domínio. A ontologia exportada pode ser editada utilizando ferramentas específicas para manipulação de ontologia, como o *Protégé*. Posteriormente, é possível realizar a importação da ontologia editada.

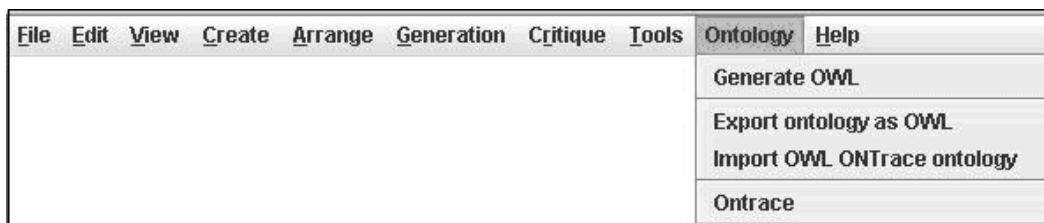


Figura 4.11 – Interface para Importação e Exportação da Ontologia.

Após a geração da ontologia, um painel na parte inferior da ferramenta, apresentado na Figura 4.12, é preenchido com todos os recursos da ontologia, que incluem:

- Classes OWL;
- Propriedades do tipo *DataType*;
- Propriedades do tipo *Object*.

Assim, é possível relacionar cada elemento arquitetural com os recursos da ontologia, apenas clicando em um *checkbox*.

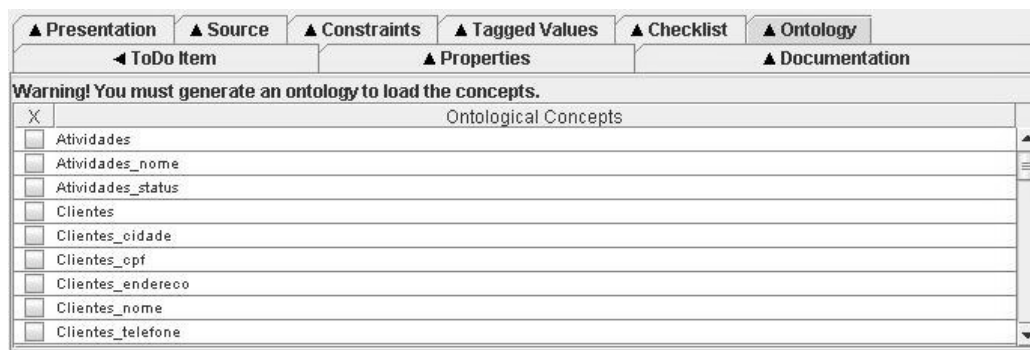


Figura 4.12 – Criação dos elos de rastreabilidade.

O processo para atualização de um indivíduo está expresso no diagrama de colaboração da Figura 4.13 e é realizado pela classe *TabOntology*. Esta classe cria um novo indivíduo (elemento UML) e o vincula a uma propriedade que o relaciona aos conceitos definidos na ontologia, utilizando para isso a classe *OntModel*. Caso já exista o indivíduo no modelo lógico, o procedimento corresponde à atualização de seus conceitos.

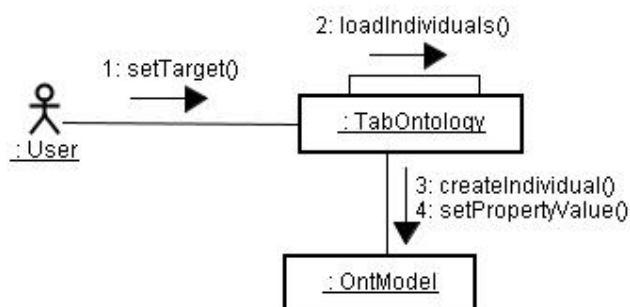


Figura 4.13 – Criação dos elos de rastreabilidade.

Para se recuperar os elos de rastreabilidade, o usuário pode realizar uma consulta sobre a ontologia, utilizando um motor de inferência. Esta funcionalidade recupera os elementos relacionados pelo mesmo conceito (relacionamento direto) e por conceitos associados (relacionamento indireto). A Figura 4.14 apresenta a interface pela qual é realizada recuperação dos elos de rastreabilidade.

O nível de encadeamento para recuperação dos conceitos relacionados pode ser variado, isto é, o número de relacionamentos conceito-conceito observados para recuperação de artefatos é configurável. Em nosso estudo, foi trabalhado apenas com o nível dois: conceito que se relaciona com outro conceito.

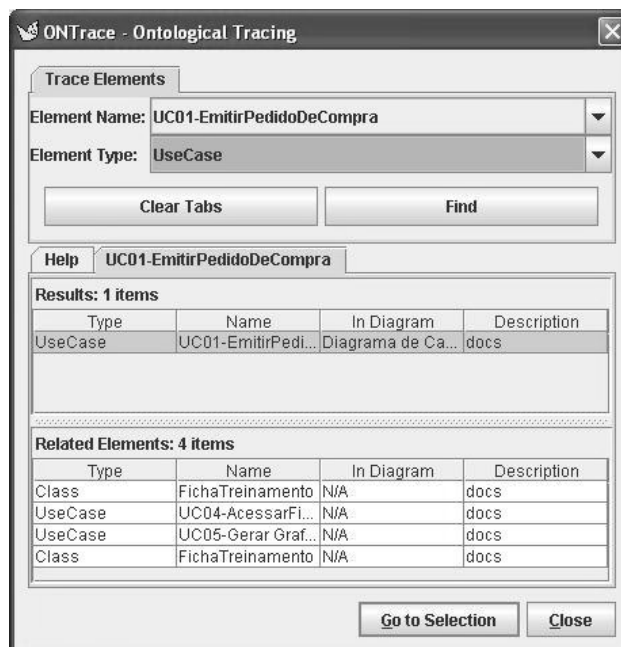


Figura 4.14 – Interface para recuperação de elos de rastreabilidade.

A Figura 4.15 apresenta o diagrama de colaboração que estrutura a recuperação dos elos, realizada pela classe de interface *DialogOntrace*. Esta, por sua vez, através da classe de negócios *Ontrace*, cria um novo modelo de inferência (*InfModel*) e realiza as consultas para artefatos indexados pelo mesmo conceito e por conceitos relacionados.

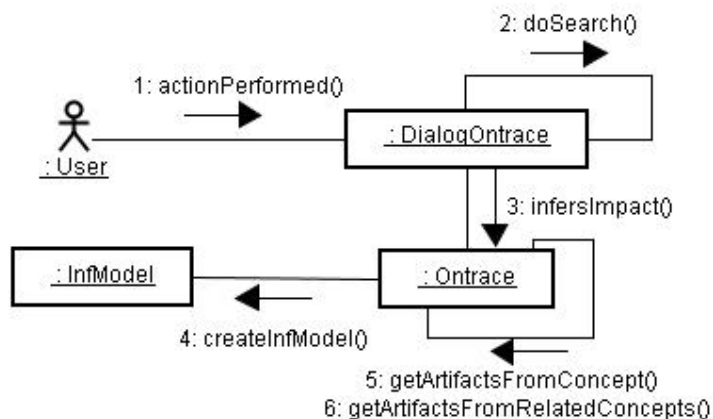


Figura 4.15 – Diagrama de colaboração para recuperação de elos de rastreabilidade.

Os cenários apresentados demonstram o funcionamento da ferramenta, permitindo uma avaliação de sua usabilidade no processo de desenvolvimento de software.

4.4.3 Exemplo de aplicação da ferramenta

Para exemplificar o uso da ferramenta associada à modelagem do conhecimento, iremos utilizar um sistema modelado e desenvolvido por alunos de graduação da faculdade de Ciência da Computação da PUCRS. O domínio do sistema consiste em uma academia desportiva, chamada *isGym*. Utilizaremos a mesma modelagem para a avaliação da proposta, presente no próximo capítulo. Durante a modelagem de negócio, o Processo Unificado sugere o desenvolvimento de um artefato chamado Modelo de Domínio, que representa a modelagem conceitual do universo de discurso. O modelo de domínio, apresentado na Figura 4.16, é utilizado como entrada para a modelagem do conhecimento.

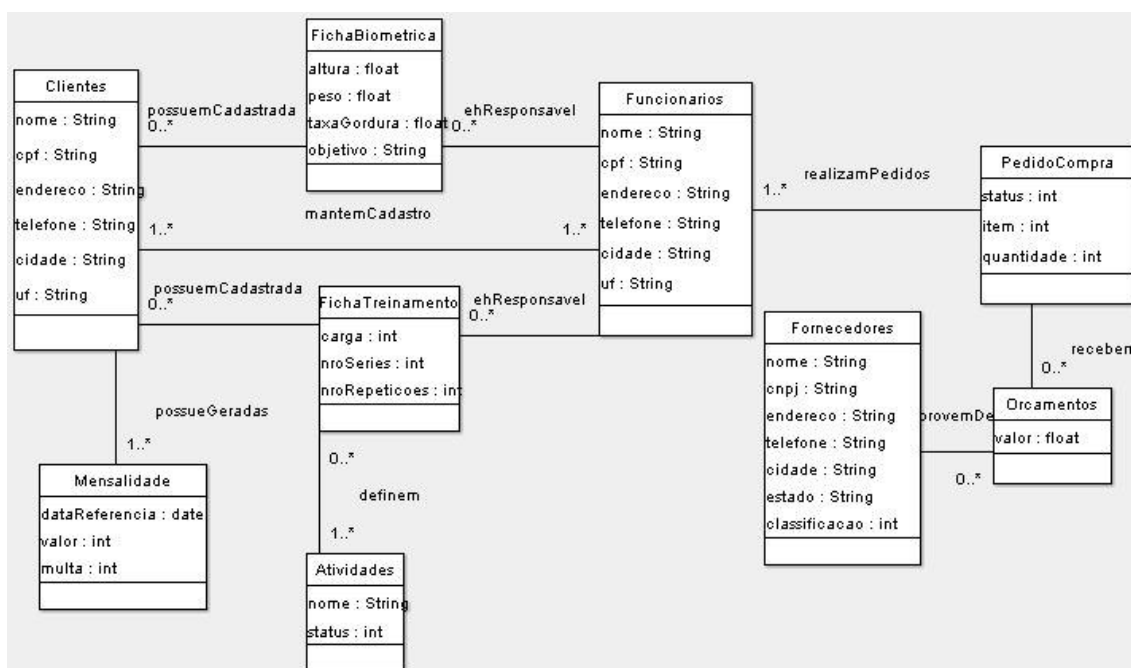


Figura 4.16 – Modelo de Domínio do sistema *isGym*.

Após a maturação deste modelo, é realizada a extração automática da ontologia utilizando a interface apresentada na Figura 4.9. Após esta extração, todos os conceitos relacionados com o domínio são apresentados em um painel na parte inferior da ferramenta, conforme a Figura 4.12.

Em conformidade com atividade de Projeto e Manutenção da disciplina proposta, a ontologia pode ser exportada (Figura 4.11) para uma ferramenta de edição de ontologias, como o [PRO06].

Após a geração da ontologia, é possível mapear o conhecimento desenvolvido durante a modelagem do sistema com os conceitos da ontologia. Para exemplificar, a Figura 4.17 representa o diagrama de casos de uso do sistema *isGym*.

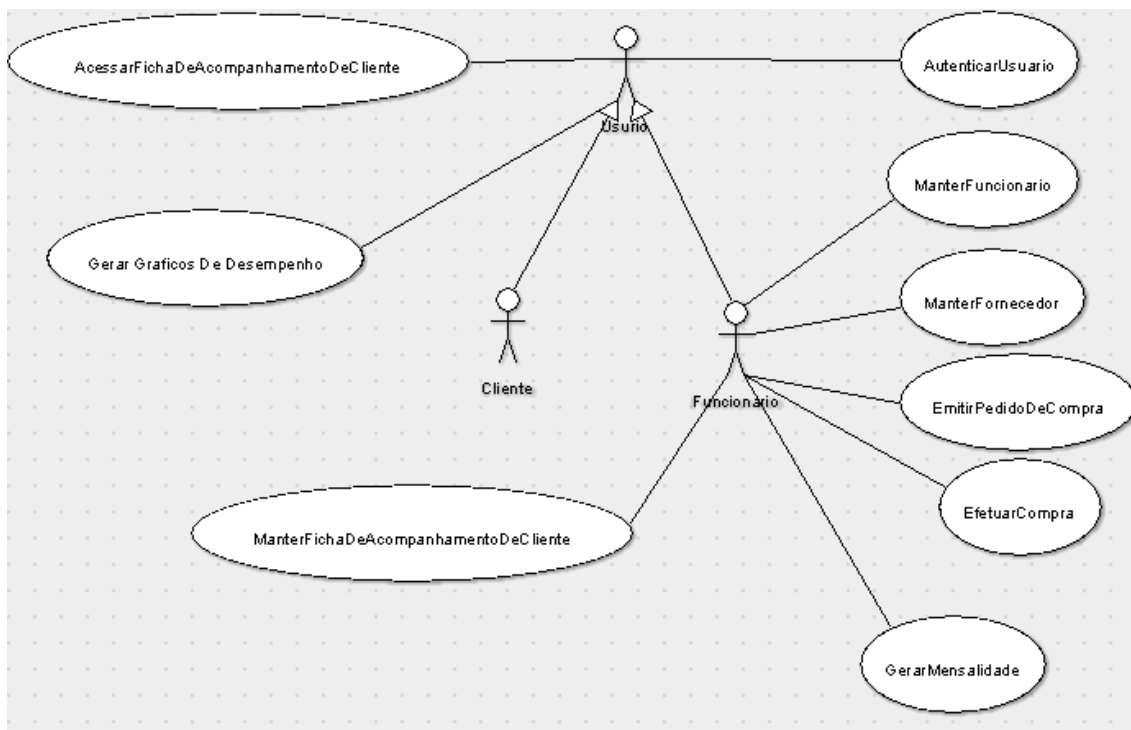


Figura 4.17 – Diagrama de casos de uso do sistema *isGym*.

Pela descrição do caso de uso *Autenticar Usuário*, fornecida junto com a modelagem do sistema, é possível identificar que este caso de uso se relaciona com os conceitos *Cliente* e *Funcionário* da ontologia. Esta atividade de identificação dos conceitos relacionados aos itens de projeto é repetida para os demais elementos que compõem o modelo de software. A atividade associada à definição dos relacionamentos ocorre pela seleção de determinado item e o preenchimento dos *checkboxs* correspondentes no painel apresentado na Figura 4.12.

Após a definição dos elos de rastreabilidade entre os elementos do projeto de software, é possível a recuperação destes relacionamentos através da interface apresentada pela Figura 4.14.

4.5 CONSIDERAÇÕES

Até o presente capítulo, foram sistematicamente apresentadas as etapas envolvidas com a proposta de integração de ontologias no processo de desenvolvimento. Nos capítulos iniciais, foi realizado um estudo sobre ontologias e aspectos relacionados, identificando suas estruturas, linguagens e propostas para desenvolvimento. Com base nestas propostas, foi

sugerida a integração de ontologias no Processo Unificado, visando aproveitar os esforços de modelagem de negócio e entendimento do domínio, comum a ambos.

Com o desenvolvimento de uma ontologia que representa a modelagem conceitual de um sistema, se torna possível a rastreabilidade dos artefatos envolvidos com o seu desenvolvimento. Esta estrutura é apresentada no presente capítulo, consistindo em uma rastreabilidade apoiada por conceitos.

Utilizando a proposta de indexação por conceitos, sugere-se um aumento da quantidade de relacionamentos entre os artefatos se comparada a requisitos, o que implica em uma apuração maior dos elos. Isto é verificado pois um requisito é composto por um ou mais conceitos do domínio, então a granularidade dos elos envolvidos com os conceitos tende a ser menor se comparada a requisitos. Em contrapartida, aumentando o número de índices, aumenta-se o trabalho envolvido com a atividade de definir os rastros. Serão apresentados estudos que avaliem estas dimensões nos capítulos posteriores. O objetivo deste estudo é realizar uma medição das variáveis que definem a rastreabilidade indexadas por conceitos e por requisitos, verificando os benefícios da presente contribuição. Adicionalmente, será apresentado um experimento realizado utilizando técnicas de Engenharia de Software Experimental [WOH00] e [TRA02].

5 AVALIAÇÃO DA PROPOSTA

A presente proposta apresenta duas perspectivas para sua avaliação. A primeira está relacionada com a necessidade de desenvolver um produto de software que comprove a viabilidade da integração de ontologias no Processo Unificado, apresentada no capítulo anterior. A segunda perspectiva está relacionada com o processo de utilização da proposta e seus benefícios. Para se avaliar um processo, é necessário que pessoas o utilizem. Neste contexto, estudos experimentais são determinantes para uma boa avaliação, provendo uma disciplinada, sistemática, quantificada e controlada forma de avaliar as atividades desenvolvidas por humanos.

Existe uma discussão na comunidade de Engenharia de Software sobre sua consideração como engenharia ou ciência. Esta discussão é devido ao seu caráter multidisciplinar que correlaciona questões técnicas, tais como linguagens de programação, sistemas operacionais, sintaxe e semântica, com questões sociais e psicológicas, característica da engenharia e da produção [WOH00].

Para avaliação de processo onde o fator humano é considerado, a literatura prove algumas abordagens baseadas em uma estratégia experimental. Em [PFL98], as seguintes abordagens são definidas para avaliação de processos, produtos e recursos:

- **Análise das características:** é a mais simples e corresponde a uma abordagem subjetiva, utilizada para atribuir um valor e classificar os atributos de vários métodos, visando decidir qual utilizar. Esta abordagem corresponde a um estudo em retrospectiva e é útil para estreitar o leque de opções a serem escolhidas, porém não avalia o comportamento em termos de causa e efeito;
- **Pesquisa de opinião (survey):** é um estudo em retrospectiva que visa documentar as relações e os resultados de certa situação. Durante a realização da pesquisa, registram-se as informações sobre uma situação, comparando-as com informações semelhantes. Não ocorre à manipulação de variáveis neste estudo;
- **Estudo de Caso:** ao contrário das anteriores, esta abordagem define previamente o que se deseja investigar, identificando os principais fatores que possam afetar o resultado de uma atividade. Após sua execução, é realizada a documentação de suas entradas, restrições, recursos e saídas. Esta abordagem é utilizada principalmente para observar projetos ou atividades, sem muito controle sobre o objeto de estudo;

- **Experimento:** representam o tipo de estudo mais controlado, geralmente realizado em laboratórios. Nesta abordagem, os valores das variáveis independentes (entradas do processo de experimentação) são manipulados para se observar as mudanças nos valores das variáveis dependentes (saídas do processo de experimentação). Ao término da execução do experimento, os resultados são analisados, interpretados, apresentados e, por fim, empacotados.

Em [KIT95], é observado que as diferenças entre os métodos de pesquisa são refletidas em suas escalas. Por sua natureza, como os experimentos requerem bastante controle, eles tendem a ser pequenos, envolvendo um reduzido número de pessoas ou eventos. Pode-se pensar em experimentos como “pesquisas em um ambiente restrito”. Os estudos de caso geralmente abordam um projeto típico em vez de tentar obter informações sobre todos os possíveis casos; eles podem ser considerados como “pesquisas em um ambiente típico”.

O objetivo de nosso trabalho consiste em investigar métodos alternativos para rastreabilidade, identificando relações como “melhor do que” ou “mais preciso que”. É possível, assim, isolar as variáveis que determinam esta relatividade do resto do processo e manipulá-las, avaliando os resultados a partir dos tipos de combinações possíveis. Para isso, torna-se necessário um controle sobre as variáveis independentes do experimento.

Frente às observações aqui apresentadas, optou-se pela utilização de um experimento para avaliação de nossa proposta. Devido ao experimento ser puramente quantitativo [WOH00], é necessário uma abordagem adicional para a avaliação qualitativa. Para este fim, será utilizada a pesquisa de opinião integrada ao experimento.

5.1 REFERENCIAL TEÓRICO: ENGENHARIA DE SOFTWARE EXPERIMENTAL

O processo de experimentação requer um preparo para conduzir e analisar corretamente o objeto de estudo. Este esforço representa seu maior benefício: a capacidade de controlar as variáveis relacionadas às entradas do exercício, permitindo a geração de conclusões significativas para o experimento. Na presente avaliação, faz-se necessário o controle dos indivíduos que estarão aplicando a proposta de rastreabilidade, utilizando tanto o método indexado por conceitos quanto por requisitos. Para conduzir o experimento, será

utilizado como guia as propostas de [WOH00] e [TRA02], de maneira complementar. A avaliação do experimento será apoiada por [BIS04] e [ARA06].

O processo de experimentação adotado compreende os seguintes passos:

- **Definição:** corresponde à definição clara das hipóteses e dos objetivos a partir do problema a ser resolvido. É proposto um *framework* para suas constituintes, que compreende: objeto de estudo, propósito, foco na qualidade, perspectiva e contexto;
- **Planejamento:** corresponde a fundamentação do experimento, onde seu contexto é explicitado em detalhes. Este passo compreende a formalização da hipótese nula (hipótese fundamental na qual o experimento objetiva rejeitar em favor das demais) e das alternativas (demais hipóteses que o experimento visa apoiar), determinação das variáveis independentes e dependentes, seleção dos participantes, preparação conceitual da experimentação e a consideração sobre a validade do experimento;
- **Execução:** este passo compreende a preparação, execução e validação dos dados. É durante esta etapa que ocorre a interação humana, sendo necessário preparar os participantes sob o ponto de vista moral e metodológico, evitando assim resultados errôneos ou desinteressados.
- **Análise e Interpretação:** com a entrada de dados no passo anterior, é possível a análise e interpretação do experimento. Para tanto, ocorre à compreensão dos dados e a verificação das hipóteses utilizando estatística descritiva, possibilitando assim conclusões sobre a validade do experimento;
- **Empacotamento:** este passo compreende a documentação dos resultados e a estruturação do experimento, visando possibilitar sua replicação.

É importante esclarecer que um experimento nunca irá proporcionar uma resposta final para uma questão, porém uma resposta específica para uma determinada configuração. O empacotamento é importante para que o experimento seja replicado em circunstâncias diversas, provendo um conhecimento adicional sobre os conceitos estudados.

Nas próximas seções, serão detalhados os passos para condução do experimento.

5.1.1 Definição

A fase de definição corresponde à fundamentação dos objetivos do experimento e a definição de seu escopo, identificando os objetos e os grupos de estudo envolvidos. Será utilizada à abordagem de [BAS94] chamada *Goal Question Metric* (GQM), que corresponde a uma medição para melhoria de processo de software dirigida por objetivos. Esta abordagem parte da definição dos objetivos (nível conceitual) para o estabelecimento de questões (nível operacional), que tentam caracterizar o processo em termos de métricas (nível quantitativo).

5.1.2 Planejamento

Para que o processo de experimentação seja bem sucedido, é necessário um planejamento de todas as atividades envolvidas. A fase de planejamento pode ser dividida em sete passos:

1. **Seleção do contexto:** com base em sua definição, este passo seleciona o ambiente no qual o experimento será executado;
2. **Formulação das hipóteses:** um experimento é normalmente formulado através de hipóteses, que representam a base para a análise estatística.
3. **Seleção das variáveis:** este passo compreende a definição das variáveis independentes e dependentes do experimento, isto é, as entradas e saídas do processo que será observado;
4. **Seleção dos indivíduos:** esta atividade se relaciona com a generalização dos resultados do experimento, com o objetivo de definir um conjunto representativo de indivíduos;
5. **Projeto:** determina a forma na qual o experimento será conduzido, incluindo seus objetos e seus participantes. A escolha do projeto correto é crucial para a validade das conclusões;
6. **Instrumentação:** representa a implementação prática, fornecendo os meios de como conduzir e monitorar o experimento;
7. **Análise da Validade:** realiza a avaliação dos resultados do experimento. Esta análise inclui a validade interna e externa, além da validade de sua construção e de sua conclusão.

5.1.2.1 Seleção do Contexto

O contexto é composto pelas condições na qual o experimento será executado. A seleção adotada contempla as seguintes dimensões:

- **Processo:**
 - *In-vitro* ou *In vivo*: o primeiro corresponde à experimentação controlada em um laboratório e o segundo em um projeto real;
- **Participantes:**
 - *Alunos* ou *Profissionais*: define os indivíduos que farão parte do experimento;
- **Realidade:**
 - *Problema desenvolvido em sala de aula* ou *Problema real*: define o tamanho do problema que será estudado;
- **Generalidade:**
 - *Específico* ou *Geral*: apresenta o escopo da validade do experimento, em âmbito específico e contextualizado ou em todo o domínio da Engenharia de Software;

5.1.2.2 Formulação das Hipóteses

A formulação das hipóteses corresponde à definição formal sobre o que se pretende com o experimento. Conforme já apresentado, a hipótese fundamental se chama Hipótese Nula (H_0) e representa a não ocorrência da relação de causa e de efeito do experimento, ou seja, a não derivação de seus objetivos. Portanto, o objetivo do experimento é rejeitá-la em prol de uma ou mais hipóteses alternativas (H_1, H_2, \dots, H_n).

5.1.2.3 Seleção das variáveis

A escolha das variáveis para o experimento não é uma tarefa trivial e geralmente requer certo conhecimento sobre o domínio do problema. É necessário que as variáveis independentes (entrada da experimentação) sejam controladas e que exerçam alguma influência sobre as variáveis dependentes (saída da experimentação).

Um experimento compreende observações sobre a alteração de uma ou mais variáveis independentes, também chamadas de fatores. Durante o estudo, é definido um fator e as demais variáveis independentes são fixadas. Isso é necessário para saber qual fator é responsável pelo fenômeno observado, também chamado de tratamento.

Os tratamentos são aplicados para a combinação de indivíduos e objetos. Para exemplificar em termos práticos, pode-se considerar um objeto como o conjunto de artefatos recuperados pela rastreabilidade indexada por requisitos ou por conceitos. Os indivíduos podem ser representados pelos participantes que definem os elos de rastreabilidade e a aquisição do conjunto de artefatos rastreados. Ambos, indivíduos e objetos, são variáveis independentes do experimento. Um experimento é estruturado através de testes, onde a combinação de tratamentos, indivíduos e objetos são avaliados.

5.1.2.4 Seleção dos Indivíduos

A seleção dos indivíduos é particularmente importante para a geração de resultados relevantes durante o experimento. Para tanto, é necessário escolher uma população representativa.

5.1.2.5 Projeto do experimento

Para se obter conclusões significativas em um experimento, é necessário recuperar cuidadosamente os dados e aplicar métodos de análise estatística. O projeto do experimento corresponde à forma na qual todas as atividades serão planejadas e conduzidas para obtenção destas conclusões.

O experimento consiste em uma série de testes sobre os tratamentos de variáveis independentes. A atenção nesta fase de projeto consiste em identificar o número de vezes que os testes serão executados para que se torne visível os efeitos dos tratamentos sobre as variáveis.

5.1.2.5.1 Princípios genéricos de projeto

A literatura provê alguns princípios genéricos para o projeto do experimento:

- **Aleatoriedade:** este princípio estabelece uma ordem aleatória para a alocação de indivíduos e objetos;
- **Obstrução:** este princípio é estabelecido pela premissa de que algumas vezes um fator de probabilidade possa exercer algum efeito indesejável sobre o resultado do experimento. Caso o fator do efeito seja visível, é possível utilizar uma técnica que o bloqueie, aumentando assim a precisão dos resultados;

- **Balanceamento:** este princípio estabelece que cada tratamento sobre as variáveis possua o mesmo número de indivíduos.

5.1.2.5.2 *Padrão para tipo de projeto*

Na maioria dos experimentos, são formuladas hipóteses e é definido um método para analisá-las estatisticamente.

Existem duas abordagens para sua execução, que são:

- **Projeto completamente aleatório:** consiste em avaliar cada instância do fator aleatoriamente entre os dois tratamentos, isto é, cada instância será aplicada em apenas um dos tratamentos, definido aleatoriamente. Sugere-se que exista um balanceamento na distribuição dos tratamentos entre os fatores;
- **Projeto de comparação pareado:** consiste em avaliar todas as instâncias do fator com os dois tratamentos. O objetivo é aumentar a precisão do experimento através do pareamento do objeto experimentado. Esta abordagem demanda um cuidado extra sobre o efeito da ordem do experimento. Sugere-se a utilização de escolha aleatória e balanceada para a execução do fator sobre os tratamentos.

Durante o projeto, define-se a tabela de contingência que representam a distribuição do fator sobre os tratamentos, definindo se o projeto será aleatório ou pareado. Conforme o projeto escolhido, existem testes específicos para a avaliação das hipóteses.

5.1.2.6 **Instrumentação**

O objetivo da instrumentação é proporcionar os meios para condução do experimento e sua análise. Sugerem-se as seguintes definições de instrumentos:

- **Objetos:** os objetos podem ser, por exemplo, documentos de especificação ou código fonte;
- **Guias:** são especialmente úteis para apoiar os participantes no experimento e incluem, por exemplo, descrição de processos, tutoriais e *checklists*;
- **Métricas:** são obtidas no experimento através da coleta de dados, normalmente através de entrevistas ou formulários preenchidos pelos participantes.

5.1.2.7 Análise da Validade

Um ponto crítico durante o experimento é a análise de sua validade. Sugere-se que esta preocupação ocorra desde o seu planejamento, prevendo algumas questões sobre a avaliação do experimento. A literatura sugere quatro tipos de validação dos resultados.

- *Validade interna*: esta validade define se o relacionamento observado entre o tratamento e o resultado é casual e não resultado de algum fator não previsto. A atenção principal é dada aos participantes durante a condução do experimento;
- *Validade externa*: esta validade sugere a generalização dos resultados obtidos durante o experimento em práticas industriais. É avaliada a representatividade dos participantes com relação ao público alvo;
- *Validade de construção*: a validade de construção avalia a relação de causa e efeito na qual o experimento é idealizado. A atenção desta validade se concentra em mapear a teoria, que motiva o experimento, nos indivíduos. Ao término, são observados os efeitos da experimentação;
- *Validade da conclusão*: esta validade corresponde à capacidade de chegar a uma conclusão correta a respeito dos tratamentos e dos resultados do experimento. Para tanto, é necessário escolher os testes estatísticos, os participantes e a confiabilidade das medidas e da implementação dos tratamentos.

5.1.3 Execução

A execução compreende a etapa operacional da experimentação, onde ocorre o envolvimento direto dos participantes. Ainda que o experimento seja perfeitamente planejado e os dados coletados sejam analisados com os métodos mais apropriados, o resultado será inválido se os indivíduos não se engajarem de maneira séria com o experimento.

A execução do experimento compreende três passos:

1. **Preparação**: corresponde a escolha dos participantes e dos materiais preparados para a experimentação;
2. **Execução**: é quando os participantes executam as tarefas do experimento, de acordo com os diferentes tratamentos previstos;

3. **Validação dos dados:** corresponde a validação dos dados coletados após a execução do experimento.

5.1.3.1 Preparação

Existem alguns preparativos que devem ser cautelosamente analisados antes da execução do experimento. Dois pontos são avaliados nesta etapa: prover a informação aos participantes e preparar os materiais necessários ao experimento, tal como formulários e ferramentas.

5.1.3.2 Execução

A execução de um experimento pode se dar de diferentes formas. Sua duração é variável e pode ocorrer em um período de tempo curto, no qual o responsável está presente em todos os detalhes da execução. Outra perspectiva é quando o tempo é longo, tornando inviável o responsável participar de cada detalhe da execução do experimento.

5.1.3.3 Validação dos dados

Após a coleta de dados, é necessário verificar se os dados são razoáveis e se foram coletados corretamente. Isso lida com aspectos específicos, como o entendimento dos participantes sobre os formulários e seu correto preenchimento, com a seriedade em que os participantes conduziram o experimento, com a remoção de dados a partir da análise, etc. É importante revisar se o experimento foi conduzido conforme o planejado e, para cada erro percebido, os dados devem ser considerados inválidos.

5.1.4 Análise e Interpretação

Após a coleta de dados experimentais gerados na fase anterior, é necessário extrair conclusões. Para se recuperar conclusões válidas, é necessário interpretar os dados experimentais.

A primeira observação sobre os dados brutos obtidos no experimento diz respeito à medida de suas escalas. As escalas determinam quais operações que podem ser executadas sobre os valores das variáveis. A literatura define os seguintes tipos de escalas:

- **Nominal:** representam diferentes valores que não possuem interpretação numérica nem ordenação;

- **Ordinal:** representam diferentes valores que podem ser ordenados, porém não possuem uma representação numérica;
- **Intervalar:** representam diferentes valores que podem ser ordenados e a distância entre os números possui a mesma interpretação;
- **Razão:** representam diferentes valores que podem ser ordenados e a distância e razão entre os mesmos pode ser interpretada.

Após a análise das medidas das escalas para as variáveis, é importante avaliar a apresentação e o processamento numérico dos dados obtidos através da estatística descritiva. Posteriormente, sugere-se a eliminação de distorções que comprometam a validade das conclusões. Por fim, é realizado o teste das hipóteses para extração das conclusões do experimento.

A estatística descritiva lida com a apresentação e o processamento numérico de um conjunto de dados. Para este fim, sugere-se representar graficamente estes dados para identificação de alguns aspectos interessantes, como os indicadores de medidas. O objetivo da estatística descritiva é analisar a distribuição geral de um conjunto de dados. Esta etapa deve ser executada antes da validação das hipóteses para compreender a natureza dos dados e identificar os dados anormais ou os valores extremos (*outliers*).

Com relação ao teste das hipóteses, existem duas formas:

- **Paramétricos:** utilizam formas fechadas, derivadas de propriedades de distribuições de frequências conhecidas e exigem que os dados possuam normalidade e homocedasticidade;
- **Não-Paramétricos:** devem ser usados quando os dados não atendem os requisitos de normalidade e de homocedasticidade. Esta abordagem utiliza *rankings* de valores observados como parâmetro ao invés dos valores propriamente ditos.

A escolha do teste das hipóteses adequado demanda a análise de dois requisitos: normalidade e da homocedasticidade. Neste contexto, apresentamos:

- **Normalidade:** os valores tendem a se concentrar próximos de uma média, e quanto maior a distância dessa média, menor a frequência das observações;
- **Homocedasticidade:** implica em uma variância constante entre o conjunto de dados que serão testados.

Depois de avaliada a normalidade e a homocedasticidade, sugerem-se dois tipos de teste de acordo com os supostos paramétricos, em um projeto com um fator e dois tratamentos:

- **Teste T:** este teste é utilizado para comparar duas médias a partir de uma hipótese nula. Esta hipótese pressupõe que não existem diferenças significativas entre dois grupos;
- **Mann-Whitney:** este teste serve para provar se dois grupos independentes procedem da mesma população.

5.1.5 Empacotamento

Após a análise e interpretação, sugere-se o empacotamento do experimento devido à sua necessidade de replicação. Ao executar o experimento em contextos distintos, possibilita-se a aquisição de novos conhecimentos a respeito dos conceitos estudados. Para isso, é importante documentar os diversos aspectos relacionados com o experimento, entre eles:

- **Comunidade:** apresenta os pesquisadores e participantes relacionados com o experimento, além dos interessados que possam aproveitar os resultados obtidos;
- **Organização:** apresenta o planejamento, projeto e demais informações referentes à preparação, diretrizes, execução e instrumentação do experimento;
- **Artefatos:** apresenta os artefatos definidos durante a instrumentação do experimento;
- **Resultados:** incluem a descrição detalhada dos resultados recebidos, com os dados brutos, refinados (pela eliminação dos *outliers*) e analisados. Os dados analisados são utilizados para testar as hipóteses e fazer as conclusões sobre o experimento.

O empacotamento do experimento proposto está documentado na Seção 5.2, onde estarão presentes todas as informações necessárias para a sua replicação.

5.2 ESTUDO EXPERIMENTAL

5.2.1 Definição

Foi utilizada a abordagem GQM para a definição do estudo, estabelecendo o objetivo global, os objetivos de estudo e de medição. Por fim, serão apresentadas as questões e suas métricas.

5.2.1.1 Objetivo Global

Comparar, no Processo Unificado, a precisão e o esforço da rastreabilidade de artefatos indexada por conceitos da ontologia com relação a tradicional proposta de indexação por requisitos.

5.2.1.2 Objetivo do estudo

Comparar, no Processo Unificado, a rastreabilidade de artefatos indexada por requisitos com a indexada por conceitos,

Com o propósito de caracterizar o tempo despendido no uso e os elementos recuperados por cada uma das abordagens,

Com foco no esforço e na precisão,

Sob o ponto de vista do arquiteto de software,

No contexto de manutenção de um sistema de informação desenvolvido por estudantes (*toy example*) no domínio de uma academia desportiva.

5.2.1.3 Objetivo da medição

Em uma manutenção de um sistema de informação, caracterizar:

- Qual o esforço (medido em minutos por cada participante) necessário para indexação dos artefatos utilizando a rastreabilidade apoiada por requisitos e por conceitos;
- Qual a precisão definida através da completude e corretude dos artefatos recuperados através da rastreabilidade indexada por requisitos e por conceitos, com relação aos artefatos que realmente se relacionam no sistema.

5.2.1.4 Questões

1. O esforço para definição dos elos de rastreabilidade indexados por conceitos é igual ao esforço para definição dos elos indexados por requisitos?
2. A precisão na recuperação de artefatos utilizando a rastreabilidade indexada por requisitos é igual à precisão utilizando a rastreabilidade indexada por conceitos?

5.2.1.5 Métricas

A métrica associada à **Questão 1** corresponde ao esforço medido pela relação do tempo gasto em minutos por cada participante durante a definição dos elos de rastreabilidade em cada abordagem.

A métrica relacionada à **Questão 2** corresponde à precisão dos elos de rastreabilidade de cada abordagem, com relação à completude e corretude dos artefatos recuperados. Por precisão, foi definido em nosso estudo como sendo a razão entre o conjunto correto de artefatos recuperados e o conjunto total de artefatos relacionados. O conjunto correto recuperado por determinada técnica compreende a intersecção entre o conjunto total de artefatos recuperados por esta técnica e o conjunto ideal, que representa o relacionamento real entre os artefatos (gabarito). O conjunto ideal será definido por um especialista através da inspeção do sistema. O conjunto total de artefatos relacionados é a união entre o conjunto de artefatos recuperados com o conjunto ideal. O cálculo da precisão se dá por:

$$P = \frac{qtdElementos(R \cap I)}{qtdElementos(R \cup I)}$$

Onde:

R: Conjunto de artefatos recuperados utilizando determinada técnica;

I: Conjunto ideal de artefatos relacionados (gabarito);

qtdElementos(): função que recupera a quantidade de elementos do conjunto.

5.2.2 Planejamento

5.2.2.1 Seleção do Contexto

Para a condução do experimento de rastreabilidade, foi escolhido o contexto de uma universidade. Embora diversos autores argumentem sobre a necessidade de conduzir experimentos em ambientes realistas [SJO02], semelhantes aos encontrados na indústria, essa abordagem demanda riscos e custos não previstos no escopo desta dissertação de mestrado. Dentro das dimensões apresentadas, caracterizam-se:

- **Processo:** será utilizada à abordagem *In-vitro*, na qual o conjunto de participantes executará o experimento em um ambiente controlado. Este experimento não se dará durante o desenvolvimento de software industrial, isto é, ele será *off-line*;
- **Participantes:** o experimento será conduzido por alunos de graduação e pós-graduação da Faculdade de Informática da PUCRS;
- **Realidade:** o problema estudado será de sala de aula (*toy example*) e corresponde a um sistema modelado e desenvolvido por alunos da graduação, durante uma disciplina da Faculdade de Informática da PUCRS, no domínio de uma academia desportiva. O objetivo desta escolha é a utilização de um sistema que se aproxime do real e que foi desenvolvido sem a intervenção do pesquisador;
- **Generalidade:** o experimento é específico e com validade apenas no escopo do presente estudo.

5.2.2.2 Formulação das Hipóteses

No presente experimento, duas hipóteses foram definidas informalmente:

1. Conforme apresentado na Seção 4.1.1.5, a rastreabilidade impõe o chamado *Princípio de Circularidade*. Para a aquisição dos elos de rastreabilidade, é necessário um esforço para definí-los. Sugere-se que o esforço gasto com a definição dos elos utilizando requisitos seja igual à definição dos elos utilizando conceitos da ontologia. Esta hipótese mapeia a Questão 1, definida na Seção 5.2.1.4;
2. Sugere-se que a rastreabilidade indexada por requisitos recupera o mesmo conjunto de artefatos que a indexada por conceitos de uma ontologia, com

relação ao conjunto ideal de artefatos relacionados. Esta hipótese mapeia a Questão 2, definida na Seção 5.2.1.4.

Com base na definição informal, viabiliza-se a formalização das hipóteses e a definição de suas medidas para avaliação.

1. **Hipótese Nula, H_0** : O esforço envolvido com a definição dos elos de rastreabilidade utilizando requisitos é igual ao esforço envolvido utilizando conceitos da ontologia.
 - a. **Medidas**: O esforço será avaliado pelo tempo gasto em minutos com a definição dos elos de rastreabilidade em cada abordagem, isto é, a diferença entre o tempo final e o tempo inicial de cada abordagem, onde:
 - i. Δt_{req} : representa a variação de tempo gasto em minutos para indexação utilizando requisitos;
 - ii. Δt_{conc} : representa a variação de tempo gasto em minutos para indexação utilizando conceitos;
 - b. **H_0** : $\Delta t_{req} = \Delta t_{conc}$
 - c. **Hipótese Alternativa, H_1** : O esforço envolvido com a definição dos elos de rastreabilidade utilizando conceitos é maior do que o esforço envolvido utilizando requisitos da ontologia.
 - i. **H_1** : $\Delta t_{conc} > \Delta t_{req}$
 - d. **Hipótese Alternativa, H_2** : O esforço envolvido com a definição dos elos de rastreabilidade utilizando requisitos é maior do que o esforço envolvido utilizando conceitos da ontologia.
 - i. **H_2** : $\Delta t_{req} > \Delta t_{conc}$
2. **Hipótese Nula, H_0** : A precisão na aquisição dos elos de rastreabilidade indexados por requisitos é igual a dos elos indexados por conceitos.
 - a. **Medidas**: Conforme apresentado na Seção 5.2.1.5, a precisão será avaliada pela relação entre o conjunto correto de artefatos recuperados e o conjunto total, onde:
 - i. P_{req} : Precisão associada à rastreabilidade indexada por requisitos;
 - ii. P_{conc} : Precisão associada à rastreabilidade indexada por conceitos;
 - b. **H_0** : $P_{req} = P_{conc}$

- c. **Hipótese Alternativa, H_1** : A precisão na aquisição dos elos de rastreabilidade indexados por requisitos é maior do que os elos indexados por conceitos.

$$H_1: P_{req} > P_{conc}$$

- d. **Hipótese Alternativa, H_2** : A precisão na aquisição dos elos de rastreabilidade indexados por conceitos é maior do que os elos indexados por requisitos.

$$H_2: P_{conc} > P_{req}$$

5.2.2.3 Seleção das variáveis

5.2.2.3.1 Variáveis Independentes

Assumiram-se como variáveis independentes:

- Experiência do time de manutenção;
- Técnica para definição dos elos de rastreabilidade;

5.2.2.3.2 Variáveis Dependentes

Assumiram-se como variáveis dependentes:

- **Esforço** para a definição dos elos de rastreabilidade;
- **Precisão** através do número de artefatos recuperados utilizando determinada técnica com relação ao conjunto total de artefatos.

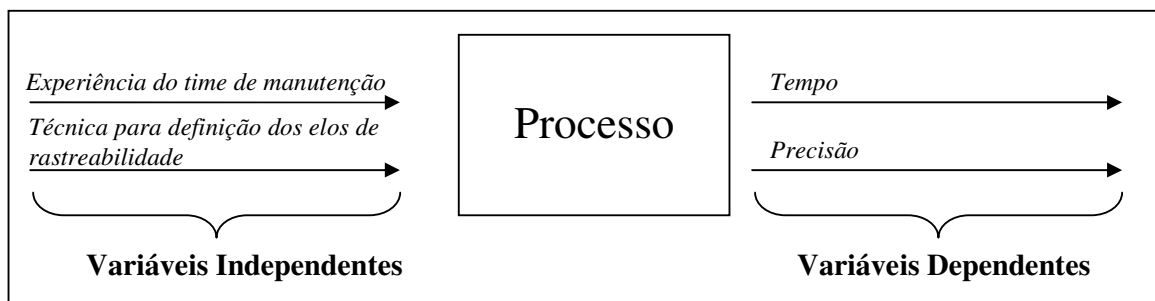


Figura 5.1 – Variáveis independentes e dependentes do estudo experimental.

5.2.2.4 Seleção dos Indivíduos

Conforme já apresentado, a população definida para o experimento é formada por alunos do curso de graduação e pós-graduação da Faculdade de Informática da PUCRS. Será um total de doze alunos, sendo:

- Seis alunos da graduação;
- Seis alunos da pós-graduação.

Não será utilizada uma amostragem probabilística para seleção dos indivíduos, mas uma amostragem não probabilística:

- **Amostragem por conveniência:** serão escolhidas as pessoas mais convenientes para o experimento;
- **Amostragem por quota:** esta abordagem implica na escolha de indivíduos de diferentes populações, no caso, graduação e pós-graduação. Pressupõe-se que a experiência com a modelagem de sistemas e com o processo de desenvolvimento entre estas duas populações seja diferente.

5.2.2.5 Projeto do experimento

Dentre os princípios genéricos para o projeto do experimento, caracterizamos:

- **Aleatoriedade:** a aleatoriedade será utilizada para definir quais participantes irão executar cada abordagem de rastreabilidade (requisito ou conceito);
- **Obstrução:** durante a experimentação, muitos dos participantes não possuem o mesmo nível de experiência acadêmica e profissional. Para minimizar o efeito da experiência sobre o experimento, os indivíduos foram selecionados utilizando o critério de quota e conveniência;
- **Balanceamento:** este princípio será utilizado em nosso experimento para que cada proposta de rastreabilidade seja executada pela mesma quantidade de participantes.

5.2.2.6 Padrão para tipo de projeto

Para cada hipótese, serão utilizadas as seguintes notações:

- | | |
|--------------|--|
| μ_{req} | Rastreabilidade indexada por requisitos; |
| μ_{conc} | Rastreabilidade indexada por conceitos. |

O tipo de projeto apresentado procura investigar se μ_{conc} possui o mesmo esforço e precisão que μ_{req} . Para este fim, será utilizada uma abordagem chamada **um fator com dois tratamentos**. O fator, neste experimento, consiste na técnica que será utilizada e os tratamentos consistem na rastreabilidade indexada por conceitos e por requisitos.

Com relação às abordagens apresentadas no referencial teórico, é possível escolher tanto o projeto pareado quanto o aleatório, porém a complexidade associada ao primeiro é maior do que o segundo. Caso se opte por um projeto pareado no contexto deste experimento, é interessante considerar um novo tratamento que represente a experiência dos participantes na execução dos demais tratamentos. Adicionalmente, o ganho com relação ao grau de liberdade entre as duas abordagens não apresenta significativa diferença. Por motivos de projeto e complexidade, não utilizaremos o projeto pareado descrito, mas sim o projeto completamente aleatório, onde cada participante executará apenas uma abordagem definida aleatoriamente. A Tabela 5.1 é chamada de **tabela de contingência** e representa a distribuição do fator sobre os dois tratamentos.

Tabela 5.1 –Tabela de contingência.

#Participante	μ_{conc}	μ_{req}
1	X	
2		X
3	X	
4	X	
5		X
6		X
7	X	
8		X
9	X	
10		X
11	X	
12		X

A Tabela 5.1 representa a forma na qual o experimento será executado. Percebe-se que a ordem foi definida de forma aleatória e balanceada entre os fatores.

Para a realização dos testes das hipóteses, em um contexto de um fator e dois tratamentos aleatórios, a literatura sugere o teste de significância chamado **Teste T** para duas amostras independentes, caso seja realizado um teste paramétrico, ou **Mann-Whitney**, caso o teste seja não-paramétrico. A definição do teste a ser aplicado ocorrerá após a análise da normalidade (através do teste de *Shapiro-Wilk*) e variância dos dados obtidos pela execução do experimento (Teste de Levene).

5.2.2.7 Instrumentação

No presente experimento, será utilizado:

- **Objetos:** a modelagem UML de um diagrama de classes de projeto de um sistema de informação no qual deverão ser definidos os elos de rastreabilidade, a descrição dos casos de uso e o Modelo de Domínio. Para o estabelecimento dos elos de rastreabilidade, será fornecido o apoio ferramental para cada abordagem: uma ferramenta CASE (ArgoUML+ONTrace) para indexação dos conceitos e uma matriz de rastreabilidade, utilizando o MS Excel, para indexação dos requisitos. Foi previsto para as duas abordagens uma normalização dos esforços visando não causar distorções nos resultados pela utilização de ferramentas distintas;
- **Guias:** será fornecido um treinamento para os participantes, apresentando as duas abordagens de rastreabilidade, o contexto do experimento e a motivação da equipe. Também será fornecido um documento sobre como proceder durante o experimento;
- **Métricas:** Os dados serão recuperados através de formulários preenchido pelos participantes, com base na ferramenta ArgoUML e na matriz definida no MS Excel.

5.2.2.8 Análise da Validade

5.2.2.8.1 *Validade interna*

Serão avaliados alguns critérios, tais como:

- *Histórico:* a data de aplicação do experimento será criteriosamente definida, evitando períodos nos quais os participantes possam sofrer influências externas;
- *Maturação:* durante o treinamento, serão utilizadas técnicas de motivação para incentivar positivamente os participantes;
- *Seleção dos grupos:* será utilizada uma abordagem para nivelar o conhecimento dos participantes através de um treinamento sobre as técnicas. A execução das atividades será individual;

- *Difusão*: durante o treinamento, será desenvolvida uma motivação que não incentive interação entre os participantes. Adicionalmente, haverá um policiamento durante a experimentação para evitar este tipo de interação;

5.2.2.8.2 *Validade externa*

Para esta avaliação, será adotada a interação da seleção, ou seja, os participantes que foram selecionados possuem um perfil apto aos tratamentos do experimento, apresentando, em sua maioria, conhecimento prévio sobre processo de desenvolvimento de software e modelagem de sistemas, além de experiência em indústria.

5.2.2.8.3 *Validade de construção*

Durante nosso experimento, serão avaliados:

- *Inadequada explicação pré-operacional*: consiste na explicação operacional do experimento, visando maturar a forma na qual serão definidos os elos de rastreabilidade e a extração dos dados;
- *Adivinhação de hipóteses*: devido ao fato dos participantes serem humanos, é possível sua interação com o experimento, sugerindo novas hipóteses e exercitando a criatividade. É importante manter o foco no estudo planejado;
- *Expectativas do condutor do experimento*: ao se conduzir um experimento, o responsável pode exercer influências sobre as variáveis envolvidas e sobre o material elaborado. Durante a presente proposta, todo o material utilizado será previamente avaliado por outro responsável.

5.2.2.8.4 *Validade da conclusão*

Serão avaliadas as seguintes perspectivas:

- *Manipulação dos dados*: como os dados resultantes do experimento serão manipulados pelo pesquisador, é possível que os mesmos sofram algumas variações, tal como o coeficiente de significância para validação dos resultados;
- *Confiabilidade das medidas*: esta perspectiva sugere que medidas subjetivas possam ser influenciadas pelo pesquisador. Em nossa proposta,

as medidas foram objetivamente definidas, não dependendo do critério humano;

- *Confiabilidade na implementação dos tratamentos*: consiste no risco em que diferentes participantes possam implementar de forma distinta os processos estabelecidos pelo experimento. Este risco não será evitado em nosso estudo, visto que não se pode interferir no caráter subjetivo de indexação dos artefatos. Possivelmente, diferentes participantes definirão elos de rastreabilidade distintos;
- *Configurações do ambiente do experimento*: consiste nas interferências externas do ambiente que podem influenciar os resultados durante a execução do experimento. O experimento será executado em um laboratório isolado, onde será proibida a interação externa como celulares, saídas, etc.;
- *Heterogeneidade aleatória dos participantes*: a escolha de diferentes participantes com diferentes experiências pode exercer um risco na variação dos resultados.

5.2.3 Execução

5.2.3.1 Preparação

Para preparar a execução do experimento, atentou-se para:

- *Consenso com o experimento*: de acordo com [WOH00], se os participantes não concordam com os objetivos da pesquisa ou não tem conhecimento sobre o experimento, corre-se o risco de que sua participação não ocorra em encontro aos objetivos. Durante a experimentação, a preparação dos participantes deverá fornecer o embasamento necessário sobre o experimento, clarificando quais os objetivos e metas almeçadas;
- *Resultados sensíveis*: é possível que o resultado obtido pelo experimento se influencie por questões pessoais, como a sensibilidade dos participantes por estarem sendo avaliados. Será adotada uma postura de anonimato dos participantes em toda a descrição da experimentação.

Com relação à instrumentação, todas as variáveis e os recursos foram criteriosamente estabelecidos antes da execução do experimento. Foi apresentado um treinamento específico para cada grupo (rastreadabilidade indexada por conceitos e por requisitos), contextualizando os objetivos, a técnica, a motivação e o procedimento técnico para condução do experimento. O treinamento de ambos os grupos foi compilado no Apêndice A. Adicionalmente, para apoiar os participantes em suas tarefas foi fornecido um tutorial sobre como executar as atividades para a rastreadabilidade indexada por conceitos (Apêndice B) e por requisitos (Apêndice C). Os próprios participantes foram responsáveis pela coleta dos dados, utilizando um formulário para preenchimento dos resultados, conforme Apêndice D. Outro critério a ser considerado é a questão do anonimato, onde os nomes dos participantes não serão registrados.

5.2.3.2 Execução

A presente proposta estrutura o experimento em um curto período de tempo, na qual o pesquisador estará envolvido em todos os detalhes da execução. Durante a execução, será definida a coleta dos dados e o ambiente no qual ocorrerá o experimento.

A coleta de dados será de responsabilidade dos participantes, através do preenchimento de formulários. Esta opção é justificada, pois o pesquisador não poderá se envolver na definição dos dados resultados do experimento.

Durante a execução do experimento, o responsável estará à disposição dos participantes para o esclarecimento das dúvidas que surgirem ao longo do processo.

5.2.4 Análise e Interpretação

A primeira análise apresentada diz respeito à classificação das escalas das variáveis definidas no experimento, apresentada no Quadro 5.1. Com esta classificação, é possível determinar as operações que podem ser aplicadas sobre as variáveis.

Quadro 5.1 – Escalas das variáveis

Variáveis	Nome	Escala
Dependentes	Tempo	Razão
	Precisão	Razão
Independentes	Técnica de rastreadabilidade	Nominal

5.2.4.1 Análise Tabular e Gráfica

O experimento obteve como resultado os dados apresentados na Tabela 5.2.

Tabela 5.2 – Tabulação dos valores brutos obtidos após a execução do experimento.

Abordagem	Participante	Esforço	Precisão
Conceito	C01	4,00	1,00
Conceito	C02	5,00	1,00
Conceito	C03	3,00	1,00
Conceito	C04	8,00	0,69
Conceito	C05	4,00	1,00
Conceito	C06	4,00	1,00
Requisito	R01	20,00	0,51
Requisito	R02	17,00	0,34
Requisito	R03	22,00	0,51
Requisito	R04	20,00	0,51
Requisito	R05	30,00	0,51
Requisito	R06	22,00	0,53

A Figura 5.2 representa o gráfico de barras com o esforço de cada participante para a definição dos elos de rastreabilidade, indexados por conceitos e por requisitos. Todas as análises apresentadas neste experimento foram feitas utilizando o pacote estatístico SPSS [SPS06].

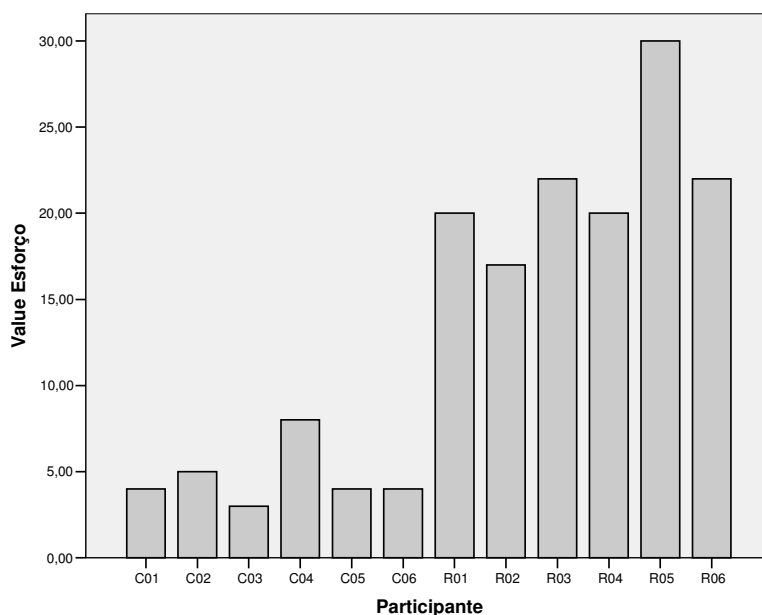


Figura 5.2 – Gráfico de barras relativo ao esforço para definição dos elos de rastreabilidade.

A Figura 5.3 representa o gráfico de barras com a precisão dos artefatos recuperados pelos participantes.

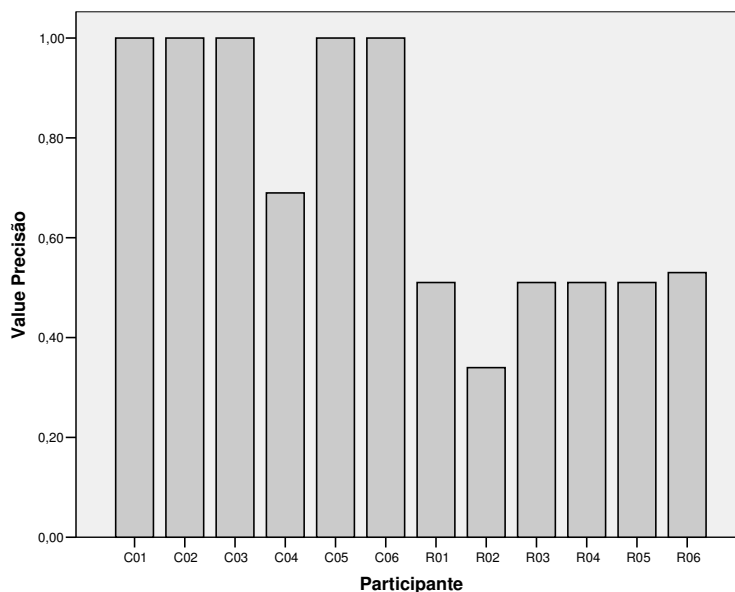


Figura 5.3 – Gráfico de barras relativo à precisão na recuperação dos elos de rastreabilidade.

5.2.4.2 Estatística Descritiva

Conforme apresentado, as variáveis dependentes estão caracterizadas na escala razão, o que permite o cálculo da normalidade e homocedasticidade, necessária para definir o tipo de teste das hipóteses (paramétrico ou não-paramétrico). Conforme definido no projeto do experimento, o padrão para tipo de teste previsto é o Teste T para duas amostras independentes, caso o teste empregado seja paramétrico, ou Mann-Whitney, caso seja não paramétrico.

Nossa avaliação será executada para as hipóteses: esforço e precisão. Para cada hipótese, os dados serão caracterizados, visualizando tendências centrais e dispersões. Posteriormente, sugere-se a eliminação de dados anormais ou incertos, que distorcem a integridade da conclusão, através da redução do intervalo de dados. Por último, será realizado o teste das hipóteses que compreende a avaliação estatística dos dados até certo nível de significância. O nível de significância adotado (*p-value*) para todos os testes é de 5%. O *p-value* compreende o menor nível de significância com que se pode rejeitar a hipótese nula.

5.2.4.3 Primeira Hipótese: Esforço

Uma análise inicial da distribuição é eficiente para avaliar o comportamento das amostras. Utilizaremos o gráfico de dispersão *boxplot*, apresentado na Figura 5.4, para identificação dos *outliers*.

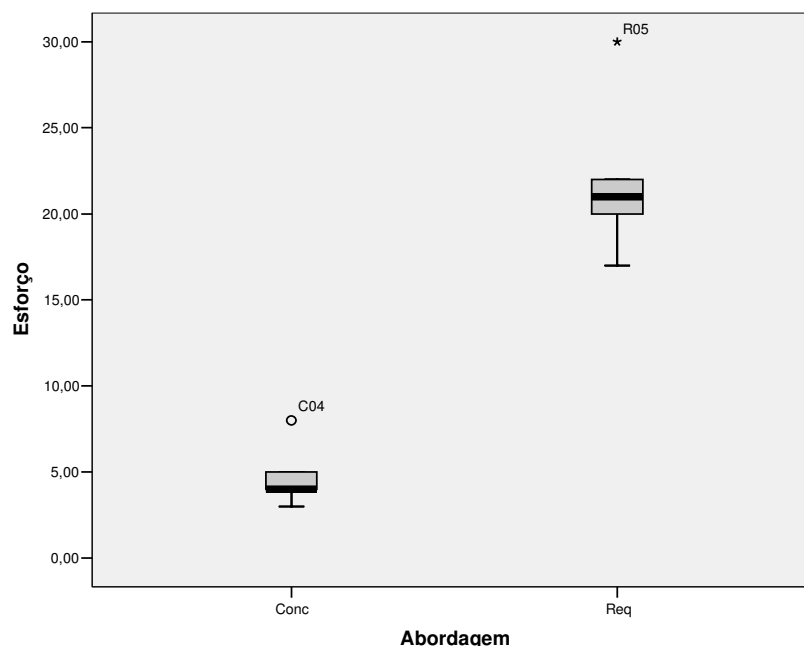


Figura 5.4 – Gráfico de dispersão para a variável *esforço*.

Conforme apresentado na Figura 5.4, a variável *esforço* possui *outliers* moderados, com exceção do participante R05. Um possível motivo para a ocorrência do *outlier* R05 é o fator de experiência sobre o processo de desenvolvimento e modelagem de sistemas. Por decisão de projeto, o valor deste participante será eliminado da amostra visando evitar distorções. Outra observação sobre o gráfico de dispersão é a pequena variabilidade entre as duas amostras.

A próxima etapa consiste em identificar se os dados seguem uma distribuição normal. Para se avaliar a normalidade, é definida uma hipótese nula e uma hipótese alternativa, conforme:

- **H₀**: a distribuição é normal;
- **H₁**: a distribuição não é normal.

Existem duas formas para se avaliar a distribuição normal dos dados, que compreendem o *Teste de Kolmogorov-Smirnov* e o *Teste de Shapiro-Wilk*. O primeiro é utilizado para identificar a normalidade em variáveis com pelo menos 30 valores e o segundo em variáveis com menos de 50 valores. A Tabela 5.3 apresenta os testes de normalidades para a amostra utilizando o *Teste de Shapiro-Wilk*.

Tabela 5.3 – Teste de normalidade *Shapiro-Wilk* para a variável *esforço*.

Variável	Abordagem	Estatística	Grau de Liberdade	Significância
Esforço	Conceito	0,797	6	0,055
	Requisito	0,862	5	0,236

Com base na Tabela 5.3, observa-se que a significância dos dados do teste de *Shapiro-Wilk* é superior, em ambas as amostras (conceito e requisito), ao nível de significância definido (0,05 ou 5%). Com esta informação, não há indícios para rejeitar a hipótese nula sobre a distribuição da normalidade, conseguindo assim o primeiro requisito para utilização de teste paramétrico para duas amostras independentes.

O segundo requisito requer a análise da homocedasticidade, tornando necessário analisar a variância das duas amostras. Com este objetivo, definem-se duas hipóteses:

- **H₀**: As variâncias são iguais;
- **H₁**: As variâncias não são iguais.

O teste da hipótese acima é realizado com a significância obtida diretamente através do *Teste de Levene*. O Teste de Levene é usado para testar se k amostras têm a mesma variância. A Tabela 5.4 apresenta os resultados obtidos para este teste.

Tabela 5.4 – Teste de Levene para igualdade das variâncias sobre a variável *esforço*.

Variável	Variâncias Iguais	Significância
Esforço	Assumindo	0,770
	Não assumindo	0,000

Com base na Tabela 5.4, verifica-se que o nível de significância para variâncias iguais (0,770) é superior ao nível de significância definido (0,05 ou 5%). Com esta informação, não se consegue rejeitar a hipótese nula para variâncias, conseguindo o segundo requisito para utilização do teste paramétrico.

Conforme definido no planejamento do projeto do experimento, o teste indicado para avaliação das hipóteses é o Teste T para duas amostras independentes. Consegue-se validar sua utilização por ser um teste paramétrico, no qual seus requisitos foram explicitamente atendidos.

Com base na declaração das hipóteses, tem-se:

$$H_0: \mu_{\text{req}} = \mu_{\text{conc}}$$

$$H_1: \mu_{\text{conc}} > \mu_{\text{req}}$$

$$H_2: \mu_{\text{req}} > \mu_{\text{conc}}$$

O critério para rejeição de H_0 em favor de H_1 é:

$$H_1: (\mu_{\text{conc}} > \mu_{\text{req}}): \text{rejeita-se } H_0 \text{ se } t_0 > t_{\alpha, n+m-2}, \text{ onde}$$

t_0 : é o valor t obtido através da aplicação do Teste T.

$t_{\alpha, n+m-2}$: é o valor obtido pela tabela de Distribuição de T, apresentada no Anexo B, onde $n+m-2$ representa o grau de

liberdade, sendo n o número de participantes de uma abordagem e m o número de participantes da outra abordagem. O grau de liberdade de nossa amostra é 9.

O Teste T para duas amostras independentes foi aplicado neste contexto e o resultado está apresentado na Tabela 5.5.

Tabela 5.5 – Teste T para duas amostras independentes para a variável *esforço*, agrupada por *conceito* e *requisito*.

Variável	Variâncias Iguais	T	Grau de Liberdade	Significância (bicaudal)
Esforço	Assumindo	-13,576	9	0,000
	Não assumindo	-13,363	7,984	0,000

Com base na Tabela 5.5, obtemos o valor de t_0 (= -13,579) e, com base no Anexo D, obtemos o valor de $t_{\alpha, n+m-2}$ (= 2,26). Como $t_0 < t_{\alpha, n+m-2}$, então não se consegue rejeitar a hipótese nula a um nível de significância de 5% em favor de $H_1: \mu_{conc} > \mu_{req}$.

O critério para rejeição de H_0 em favor de H_2 é:

$$H_2: (\mu_{req} > \mu_{conc}) \text{ rejeita-se } H_0 \text{ se } t_0 > t_{\alpha, n+m-2}$$

O Teste T foi aplicado neste contexto e o resultado está apresentado na Tabela 5.6.

Tabela 5.6 – Teste T para duas amostras independentes para a variável *esforço*, agrupada por *conceito* e *requisito*.

Variável	Variâncias Iguais	T	Grau de Liberdade	Significância (bicaudal)
Esforço	Assumindo	13,576	9	0,000
	Não assumindo	13,363	7,984	0,000

Com base na Tabela 5.6, obtemos o valor de t_0 (= 13,579) e o valor de $t_{\alpha, n+m-2}$ (= 2,26). Como $t_0 > t_{\alpha, n+m-2}$, consegue-se rejeitar a hipótese nula a um nível de significância de 5% em favor de $H_2: \mu_{req} > \mu_{conc}$.

Pelas análises apresentadas, pode-se concluir que existe diferença estatisticamente significativa em relação ao esforço na definição dos elos de rastreabilidade utilizando conceitos e utilizando requisitos. O esforço necessário para indexação dos requisitos é maior do que dos conceitos.

5.2.4.4 Segunda Hipótese: Precisão

Da mesma forma que na análise da primeira hipótese, utilizaremos o gráfico de dispersão *boxplot* para identificação dos *outliers*, apresentado na Figura 5.5.

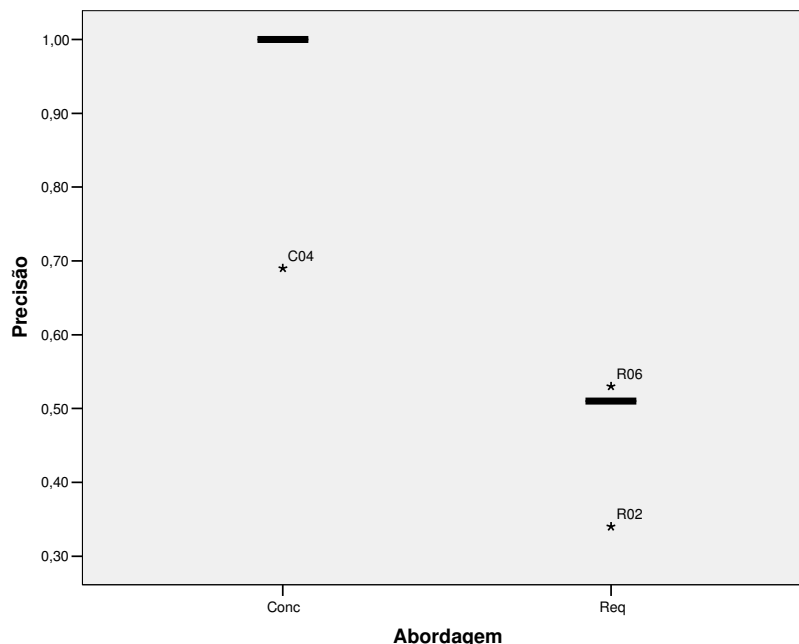


Figura 5.5 – Gráfico de dispersão para a variável *precisão*.

De acordo com a Figura 5.5, a variável *precisão* possui três *outliers*. Para escolha de quais participantes serão excluídos da amostra, optou-se pela identificação numérica. A Tabela 5.7 apresenta algumas medidas estatísticas para a variável *precisão*, agrupadas por abordagem.

Tabela 5.7 – Média e desvio padrão para a variável *precisão*.

Abordagem	Média	Desvio Padrão
Conceitos	0,948	0,126
Requisitos	0,485	0,072

Por critério de projeto, estabeleceu-se que os valores extremos que não atingirem a média com mais de dois desvios padrões, serão removidos da amostra. Neste contexto, apenas o participante C04 foi eliminado do conjunto de dados. Os demais *outliers* não foram considerados críticos para a validade das conclusões.

A próxima etapa consiste em identificar se os dados seguem uma distribuição normal. Definem-se assim as hipóteses:

- H_0 : a distribuição é normal;
- H_1 : a distribuição não é normal.

Tabela 5.8 – Teste de normalidade *Shapiro-Wilk* para a variável *precisão*.

Variável	Abordagem	Estatística	Grau de Liberdade	Significância
Precisão	Requisito	0,584	6	0,000

Com base na Tabela 5.8, observa-se que a significância dos dados do teste de Shapiro-Wilk é inferior, na abordagem por requisitos, ao nível de significância definido (0,05 ou 5%). Sendo assim, há indícios para rejeitar a hipótese nula e, conseqüentemente, não se pode aplicar um teste paramétrico para avaliação das hipóteses. Optou-se por aplicar o teste Mann-Whitney, para duas amostras independentes, por se tratar de uma alternativa não-paramétrica para o Teste T.

O teste de Mann-Whitney para duas amostras independentes é utilizado para comprovar se as diferenças entre as médias observadas nos dois grupos independentes são estatisticamente significativas. Com base na declaração das hipóteses, sugere-se:

H_0 : Não há diferença entre as médias ($\mu_{req} = \mu_{conc}$)

H_1 : Há diferença entre as médias ($\mu_{req} \neq \mu_{conc}$)

O resultado do teste Mann-Whitney foi aplicado sobre as amostras e está apresentado na Tabela 5.9.

Tabela 5.9 – Teste não paramétrico de Mann-Whitney para a variável *precisão*.

Variável	U de Mann-Whitney	W de Wilcoxon	Z	Sig. Assimpt. (bilateral)	Sig. Exata [2*(Sig.Unilateral)]
Precisão	0,000	21,000	-2,947	0,003	0,004 ^a

(a) Não corrigidos para os empates

Como o grau de significação associado (Sig. Assimpt.) é 0,003 e é menor que a significância assumida de 0,005, deve-se rejeitar H_0 . Frente aos resultados apresentados para a variável *precisão*, existe diferença de média entre a rastreabilidade apoiada por requisitos e por conceitos.

Pela análise estatística dos dados, consegue-se recuperar duas informações:

- A distribuição da precisão não é normal, o que implica na execução de testes não paramétricos;
- Utilizando o teste Mann-Whitney, conseguiu-se verificar que existem diferenças entre as médias das duas amostras μ_{req} e μ_{conc} .

Utilizando o teste de Mann-Whitney, conseguiu-se apenas rejeitar a hipótese nula, porém não foi possível avaliar as hipóteses alternativas, pois não é possível extrair relações de

“maior do que” com o teste aplicado. Porém, sugere-se comparar a análise descritiva das médias da amostra conforme a Tabela 5.10.

Tabela 5.10 – Estatística descritiva para a variável *precisão*.

Abordagem	Média
Requisitos	0,485
Conceitos	1,000

Comparando as médias apresentadas, e com base nas médias das duas abordagens, observa-se que a precisão na aquisição dos elos de rastreabilidade por conceitos é maior do que por requisitos.

5.3 AVALIAÇÃO QUALITATIVA

Conforme verificado no referencial teórico deste capítulo, um experimento é responsável apenas por uma avaliação quantitativa. Para a análise qualitativa das duas abordagens de rastreabilidade, foi desenvolvida uma pesquisa de opinião integrada ao experimento. Ao término da execução, cada participante respondeu um questionário conforme Apêndice E. Como critério de resposta, foi utilizado a escala Likert de 5 pontos para o grau de satisfação.

Os objetivos almejados com a pesquisa foram apresentar a opinião dos participantes quanto à usabilidade, utilidade e esforço necessários em cada abordagem.

Os dados brutos obtidos como resultados estão apresentados na Tabela 5.11.

Tabela 5.11 – Resultados da avaliação qualitativa.

Abordagem	Participante	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Conceito	C01	4	4	5	3	4	3	4
Conceito	C02	5	5	5	5	5	4	5
Conceito	C03	4	4	5	3	3	5	5
Conceito	C04	4	4	5	5	5	4	5
Conceito	C05	5	5	5	5	5	5	5
Conceito	C06	4	5	5	5	5	4	5
Requisito	R01	4	5	4	4	4	4	4
Requisito	R02	4	4	5	4	4	5	4
Requisito	R03	4	5	5	4	4	4	4
Requisito	R04	2	4	4	2	2	2	4
Requisito	R05	5	3	4	3	5	3	4
Requisito	R06	4	4	4	5	5	5	5

Em uma primeira análise, explicitamos a média aritmética das questões em cada uma das abordagens, apresentada na Tabela 5.12.

Tabela 5.12 – Média da satisfação das questões sobre as abordagens.

Abordagem	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Conceito	4,33	4,50	5,00	4,33	4,50	4,17	4,83
Requisito	3,83	4,17	4,33	3,67	4,00	3,83	4,17

É possível observar que a média para todas as questões em uma abordagem por conceitos é maior do que por requisitos. Por termos utilizado uma escala de Likert variando de 1 a 5, com grau de satisfação progressiva, verificou-se que a avaliação qualitativa da proposta de rastreabilidade ontológica foi superior, em termos gerais, a rastreabilidade por requisitos.

5.4 CONSIDERAÇÕES

Este capítulo apresentou uma avaliação quantitativa e qualitativa da proposta de rastreabilidade ontológica, utilizando um estudo experimental e uma pesquisa de opinião. O objetivo foi verificar a aplicabilidade e a relevância da proposta quando comparada a tradicional forma de indexação de artefatos por requisitos.

É importante clarificar que os resultados obtidos através da experimentação não são generalizáveis para todo o processo de desenvolvimento de software. O contexto escolhido para o experimento é relativo à rastreabilidade de classes de projeto indexadas por requisitos e por conceitos da ontologia, durante a manutenção do sistema no domínio de uma academia desportiva.

Durante a indexação por requisitos, o esforço para definição dos elos de rastreabilidade correspondeu percorrer a descrição dos casos de uso e identificar as classes de projeto associadas a cada caso de uso. Durante a definição dos elos a partir de conceitos, o esforço compreendeu a observação da modelagem conceitual do sistema, através do Modelo de Domínio e da ontologia, relacionando cada classe ao seu respectivo conceito.

Os dados relativos ao esforço obtiveram uma distribuição normal com uma diferença estatisticamente insignificante entre suas variâncias (homocedasticidade). Diante desta circunstância, foi executado o Teste T (paramétrico) e se verificou a validade da hipótese alternativa que estabelece que o esforço relativo à indexação das classes de projeto pelos conceitos do domínio é menor do que a indexação pela especificação dos casos de uso.

A precisão apresentou um fenômeno distinto do esforço, onde a amostra não apresentou uma distribuição normal, impossibilitando a utilização do Teste T. O teste Mann-Whitney foi à escolha definida para avaliação das hipóteses. Este teste apenas informa se dois grupos independentes procedem da mesma população. O resultado foi que existem diferenças entre as médias das duas amostras de requisitos e conceitos. Neste contexto, não foi possível avaliar estatisticamente as hipóteses alternativas e se optou por comparar a análise descritiva das médias. Após comparar as médias, verificou-se que a precisão na recuperação da indexação por conceitos é maior que por requisitos.

Durante a análise da precisão, os resultados indicaram um fenômeno bastante positivo para a indexação por conceitos em comparação com a indexação por requisitos. A razão deste fenômeno pode ser que, pela redução da granularidade dos elementos que indexam os itens de projeto, de requisitos para conceitos, obtiveram-se elos mais apurados. Este fenômeno é explicado porque um requisito pode englobar mais de uma classe, por exemplo. Ao se recuperar as classes indexadas por requisitos, corre-se o risco de recuperar outras classes que não se relacionam diretamente com a consultada, isto é, falsos positivos e falsos negativos.

É possível também que o resultado seja reflexo de um planejamento e treinamento não balanceados ou inadequados de ambas as propostas de rastreabilidade, gerando como consequência o efeito chamado *Experiência de Hawthorne*. Este efeito é observado em uma experimentação como consequência da percepção dos participantes sobre a abordagem e não pela abordagem em si. É possível que os participantes induzisse que a abordagem por conceitos seja mais eficiente que por requisitos e distorcessem a realidade.

Para aumentar o conhecimento sobre o esforço e precisão em diferentes contextos, definindo a validade da experimentação, sugere-se replicações do experimento em modelos distintos e mais realistas, indexando diferentes estruturas que não apenas as classes de projeto. Sugere-se avaliar, por exemplo, diagramas de atividades, seqüência, componente, implantação, etc. Além disso, é interessante sua replicação em elementos rastreáveis com uma granularidade mais específica, relacionando, por exemplo, atributos, métodos, associações, interações, objetos, etc. Generalizando o experimento, possibilita-se a extração de novas informações sobre a proposta em diferentes perspectivas do processo de desenvolvimento.

6 CONCLUSÕES

Este trabalho apresentou uma proposta para rastreabilidade ontológica sobre o Processo Unificado. Esta proposta sugere a integração de ontologias no ciclo de vida do software. Com esta integração, viabilizou-se a rastreabilidade semântica com a definição e a aquisição dos relacionamentos entre conceitos e artefatos.

Foi verificada a viabilidade das propostas através da modelagem e implementação de uma ferramenta. Posteriormente, foi caracterizada a rastreabilidade apoiada por conceitos e requisitos, utilizando técnicas de Engenharia de Software Experimental.

A proposta de rastreabilidade ontológica foi aceita em dois congressos internacionais:

- *The 14th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, realizada no Arizona (USA) em março de 2007, sob o título de “Ontological Traceability over the Unified Process”. Esta publicação foi aceita como *full paper*.
- *The 22nd Annual ACM Symposium on Applied Computing*, realizada em Seoul (Korea), em março de 2007, sob o título de “Enhancing Traceability using Ontologies”. Esta publicação foi aceita como *short paper*;

Este trabalho procurou contribuir para o estado da arte sobre a rastreabilidade de artefatos durante ciclo de vida do software, abrangendo os seguintes tópicos:

- Identificação de padrões e boas práticas para construção de ontologias;
- Definição de uma disciplina para modelagem do conhecimento integrada ao Processo Unificado;
- Revisão sobre as classificações dos elos de rastreabilidade;
- Definição de uma estrutura ontológica para geração e aquisição dos elos de rastreabilidade;
- Modelagem e implementação de uma ferramenta que integre ontologias no tradicional processo de desenvolvimento de software, permitindo a modelagem do conhecimento e a rastreabilidade semântica;
- Definição de um experimento para avaliação da proposta de rastreabilidade ontológica com relação a tradicional proposta de rastreabilidade indexada por requisitos.

Estas contribuições motivam futuros pesquisadores em explorar os benefícios da utilização de linguagens para representação do conhecimento, tal como OWL, na Engenharia de Software. Com o objetivo de incentivar trabalhos futuros e apresentar as limitações deste trabalho, sugere-se:

1. Aplicação da metodologia para geração de aplicações na Web Semântica;
2. Definição mais precisa sobre a atividade de manutenção da disciplina de Modelagem do Conhecimento, abordando a integração de ferramentas específicas para manipulação de ontologias. Sugere-se também a extensão em detalhes da atividade de avaliação;
3. Evolução do protótipo, visando adaptar uma ferramenta de modelagem que esteja em conformidade com UML 2.0. Devido a limitações do ArgoUML, o protótipo atual está de acordo com UML 1.3;
4. Replicações do experimento *in vivo* em ambiente corporativo, variando, por exemplo, a modelagem do sistema, os elementos rastreáveis e a etapa do processo de desenvolvimento;
5. Aplicação da proposta na indústria.

Finalmente, o avanço na pesquisa sobre linguagens para representação do conhecimento e seu relacionamento com a Engenharia de Software pode indicar novas melhorias na proposta de rastreabilidade ontológica não citadas neste trabalho.

REFERÊNCIAS

- [ANT04] ANTONIOU, G.; HARMELEN, F. “A Semantic Web Primer”. Cambridge: MIT Press, 2004, 447 p.
- [ARA06] ARAÚJO, M. A.; BARROS, M.; TRAVASSOS, G.; MURTA, L. “Métodos Estatísticos aplicados em Engenharia de Software Experimental”. In: XXI SBBD - XX SBES, 2006, 320 p.
- [ARG06] ARGOUML. “A UML design tool”. Capturado em: <http://argouml.tigris.org>, Julho 2006.
- [BAS94] BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. “The Goal Question Metric Approach; Encyclopedia of Software Engineering”. New York: Wiley-Interscience, 1994, 578 p.
- [BEC01] BECHHOFFER, S.; HORROCKS, I.; GOBLE, C.; STEVENS, R. “OilEd: a Reason-able Ontology Editor for the Semantic Web”. In: Proc. KI2001, 2001, pp. 396-408.
- [BER01] BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. “The Semantic Web”. Capturado em: <http://www.scientificamerican.com>, Abril 2006.
- [BIA00] BIANCHI, A.; FASOLINO, A. R.; VISAGGIO, G. “An exploratory case study of the maintenance effectiveness of traceability models”. In: Proc. International Workshop on Program Comprehension, 2000, pp. 149–158.
- [BIS04] BISQUERRA, R.; SARRIERA, J. C.; MARTÍNEZ, F. “Introdução a estatística. Enfoque informático com o pacote estatístico SPSS”. Porto Alegre: Artmed Editora, 2004, 255 p.
- [BLA98] BLAZQUEZ, M.; FERNANDEZ, M.; GARCIA-PINAR, J. M.; GOMEZ-PEREZ, A. “Building ontologies at the knowledge level using the ontology design environment”. In: 11th International Workshop on Knowledge Acquisition, Modeling and Management, 1998, pp. 296-304.
- [BOR97] BORST, W. N. “Construction of Engineering Ontologies”. Tese de doutorado, University of Twente, Enschede, 1997, 215 p.
- [BOU06] BOUML. “A free UML tool box”. Capturado em: <http://bouml.free.fr>, Julho 2006.

- [BRE03] BREITMAN, K.K.; LEITE, J. "Lexicon Based Ontology Construction". In: 2nd. International Workshop on Software Engineering for Large Scale Multi Agent Systems (SELMAS). Oregon: ACM Computer Press, 2004, 132 p.
- [DAM06] DAML.org. "The DARPA Agent Markup Language Homepage". Capturado em: <http://www.daml.org>, Agosto 2006.
- [DAV02] DAVIES, J.; HARMELEN, F.; FENSEL, D.; "Towards the Semantic Web: Ontology-driven Knowledge Management". New York: John Wiley & Sons Inc., 2002, 288 p.
- [DEV02] DEVEDZIĆ, V. "Understanding ontological engineering". Communications of the ACM, vol. 45-4, Abril 2002, pp. 136-144.
- [ECL06] EMFT. "The Eclipse Modeling Framework Technology". Capturado em: <http://www.eclipse.org/emft/projects/eodm>, Setembro 2006.
- [ESS06] ESS-MODEL. Capturado em: <http://essmodel.sourceforge.net>, Julho 2006.
- [FAL98] FALBO, R. A.; MENEZES, C. S.; ROCHA, A. R. C. "A Systematic approach for Building Ontologies". In: Proc. of the IBERAMIA, 1998, 144 p.
- [FAL04] FALBO, R. A. "Ontologias e Ambientes de Desenvolvimento de Software Semânticos". In: IV Jornada Iberoamericana de Engenharia de Software e Engenharia do Conhecimento (JIISIC), 2004, pp. 277-292.
- [FAR00] FARQUHAR, A.; FIKES, R.; RICE, J. "The Ontolingua Server: a Tool for Collaborative Ontology Construction". In: International Journal of Human-Computer Studies, 1997, pp. 707-727.
- [FER99] FERNÁNDEZ-LÓPEZ, M. "Overview of Methodologies for Building Ontologies". In: Proc. IJCAI, 1999, pp. 411-413.
- [FIN91] FINKELSTEIN, A. "Tracing Back from Requirements". In: IEEE Colloquium on Tools & Techniques for Maintaining Traceability During Design, 1991, pp. 73-82.
- [FUJ06] FUJABA. "Tool Suite". Capturado em: <http://www.fujaba.de>, Julho 2006.


- [GOT94] GOTEL, O.; FILKENSTEIN, A. "Modeling the Contribution Structure Underlying Requirements". In: First International Workshop on Requirements Engineering, 1994, pp. 71-81.
- [GRU93a] GRUBER, T.R. "A translation approach to portable ontology specifications" In: Knowledge Acquisition, 1993, pp. 199-220.
- [GRU93b] GRUBER, T.R. "Towards Principles for the Design of Ontologies Used for Knowledge Sharing". International Journal of Human and Computer Studies, vol. 43-5/6, 1993, pp. 907-928.
- [GRU95] GRUNINGER, MICHAEL; FOX, MARK. S. "Methodology for the Design and Evaluation of Ontologies". In: Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI), 1995, pp. 234-241.
- [GUA98] GUARINO, N. "Formal Ontology and Information System". In: Proc FOIS, 1998, pp. 3-15.
- [HOL02] HOLSAPPLE, C. W.; JOSHI, K. D. "A Collaborative Approach to Ontology Design". Communications of the ACM, vol. 45-2, 2002, pp. 212-221.
- [IEE96] IEEE Standard for Developing Software Life Cycle Processes. New York: IEEE Computer Society, 1996, 176 p.
- [INC06] INCOSE. "Requirements Management Tools Survey". Capturado em: <http://www.paper-review.com/tools/rms/read.php>, Setembro 2006.
- [IOD06] IBM. "Integrated Ontology Development Toolkit". Capturado em: <http://www.alphaworks.ibm.com/tech/semanticstk>, Outubro 2006.
- [JAC99] JACOBSON, I.; BOOCH G.; RUMBAUGH, J. "The Unified Software Development Process". Upper Saddle River: Addison-Wesley Professional, 1999, 592 p.
- [JEN06] JENA "Semantic Web Framework". Capturado em: <http://jena.sourceforge.net>, Julho 2006.
- [KAR99] KARP, P. D.; CHAUDHRI, V. K.; THOMERE, J. F. "XOL: An XML-Based Ontology Exchange Language". Tech. Rep. 559. AI Center, SRI International, 1999, 68 p.

- [KIS04] KISHORE, R.; ZHANG, H.; RAMESH, R. "A Helix-Spindle Model for Ontological Engineering". *Communications of the ACM*, 2004, vol. 47-2, pp. 42-54.
- [KIT95] KITCHENHAM, B.; PICKARD, L.; PFLEEGER, S. L. "Case Studies for Method and Tool Evaluation". *IEEE Software*, vol. 12-4, 1995, pp. 52-62.
- [LEI93a] LEITE, J; FRANCO, A. "A Strategy For Conceptual Model Acquisition". In: *IEEE International Symposium on Requirements Engineering*, 1993, pp. 243-246.
- [LEI03b] LEITE, J. C. S. P.; DOORN, J. H. "Perspectives on Software Requirements". Massachusetts: Kluwer Academic Press, 2003, 296 p.
- [LIN94] LINDVALL, M. "A study of traceability in object-oriented systems development". Tese de Licenciatura, Linköping Studies in Science and Technology, No 462, Department of Computer and Information Science, Linköping University, Institute of Technology, Sweden, 1994, 145 p.
- [LOP97] FERNÁNDEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. "Methontology: From Ontological Art towards Ontological Engineering". In: *Proc. AAAI97*, 1997, pp. 33-40.
- [MAE02] MAEDCHE, A. "Ontology Learning for The Semantic Web". Boston: Kluwer Academic Publishers, 2002, 244 p.
- [MCG00] MCGUINNESS, D. L.; FIKES, R.; RICE, J.; WILDER, S. "The Chimaera Ontology Environment". In: *Proc. 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, 2000, pp.1123-1124.
- [NAU60] NAUR, P. "Revised Report on the Algorithmic Language ALGOL 60". *Communications of the ACM*, vol. 3-5, 1960, pp. 299-314.
- [NCI98] NATIONAL COMMITTEE FOR INFORMATION TECHNOLOGY STANDARDS. "American National Standard for Knowledge Interchange Format, 1998". Capturado em: <http://logic.stanford.edu/kif/dpans.html>, Outubro 2006.
- [NOY01b] NOY, N. F.; MCGUINNESS, D. L. "Ontology Development 101: A Guide to Creating Your First Ontology". Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, 2001, 96 p.


- [OBE04] OBERLE, D.; EBERHART, A.; STAAB, S.; VOLZ, R. “Developing and Managing Software Components in an Ontology-based Application Server”. In: 5th International Middleware Conference (ACM/IFIP/USENIX), 2004, pp. 459-478.
- [OBE05] OBERLE, D.; LAMPARTER, S.; EBERHART, A.; STAAB, S. “Semantic Management of Web Services”. In: 3rd International Conference on Service Oriented Computing (ICSOC), 2005, pp. 320-332.
- [ODA06] W3C. “Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering”. Capturado em: <http://www.w3.org/2001/sw/BestPractices/SE/ODA>, Outubro 2006.
- [OIL06] OIL Home Page. Capturado em: <http://oil.semanticweb.org>, Outubro 2006.
- [OLI00] Oliveira, K.M.; Santos, G.; Zlot, F.; Guedes, G.; Cerqueira, A.; Gallota, C.; Machado, L. F.; Lima, K.; Rapchan, F.J.; Falbo, R.A.; Travassos, G.H.; Rocha, A.R.C. “Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio na Estação TABA”. In: IDEAS, 2000, pp. 120-131.
- [OMG06] OMG. “Object Management Group”. Capturado em: <http://www.omg.org>, Julho 2006.
- [ORL03] ORLEAN, D.; LUCENA, C. “Um Processo Unificado para Engenharia de Ontologias”. Dissertação de Mestrado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 2003, 115 p.
- [OWG06] OMG. “Ontology Working Group”. Capturado em: <http://ontology.omg.org>, Julho 2006.
- [PAL00] PALMER, J. D., “Traceability – Software Requirements Engineering”. New York: IEEE Computer Society Press, 2000, pp. 412-422.
- [PFL90] PFLEEGER, S.L.; BOHNER, S. “A framework for maintenance metrics”. In: Proc. of Conf. On Software Maintenance, 1990, pp. 118-127.
- [PFL98] PFLEEGER, S. L. “Software Engineering: Theory and Practice”. Amsterdam: Prentice Hall, 1998, 245 p.
- [PRO06] PROTÉGÉ. “The Protégé Ontology Editor and Knowledge Acquisition System”. Capturado em: <http://protege.stanford.edu>, Julho 2006.

- [RUG90] RUGABER, S.; ORNBURN, S.; LEBLANC, R.J. "Recognizing Design Decisions in Programs". New York: IEEE Software, 1990, pp. 46- 54.
- [RUM98] RUMBAUGH, F.; JACKOBSON, I.; BOOCH, G. "The Unified Modeling Language Reference Manual". New York: Addison-Wesley, 1998, 550 p.
- [SHO06] SHOE. "Simple HTML Ontology Extensions". Capturado em: <http://www.cs.umd.edu/projects/plus/SHOE>, Novembro 2006.
- [SJO02] SJØBERG D.; ANDA B.; ARISHOLM, E.; DYBÅ, T.; JØRGENSEN, M.; KARAHASANOVIC, A.; KOREN, E.; VOKÁC, M. "Conducting Realistic Experiments in Software Engineering". In: Proc. ISESE, 2002, pp. 17-26.
- [SPS06] SPSS. "SPSS Statistical Analysis". Capturado em: <http://www.spss.com.br>, Novembro 2006.
- [SUR02] SURE, Y.; STUDER, R. "On-To-Knowledge Methodology". Project Deliverable D18 (WP5), EU IST-1999-10132, Institute AIFB, University of Karlsruhe, 2002, 215 p.
- [TRAV02] TRAVASSOS, G.H.; GUROV, D.; AMARAL, G. "Introdução a Engenharia de Software Experimental". Relatório Técnico, PESC 590/02, COPE/UFRJ, 2002, 66 p.
- [UMB06] Umbrello. "UML Modeller". Capturado em: <http://uml.sourceforge.net>, Julho 2006.
- [UML06] UML. "Unified Modeling Language". Capturado em: <http://www.uml.org>, Abril 2006.
- [USC95] USCHOLD, M.; KING, M. "Towards a Methodology for Building Ontologies". In: Proc. IJCAI, 1995, pp. 320-333.
- [VIL05] VILLELA, K.; SANTOS, G.; SCHNAIDER, L.; ROCHA, A. R.; TRAVASSOS, G. H. "The Use of an Enterprise Ontology to Support Knowledge Management in Software Development Environments". Journal of the Brazilian Computer Society, vol. 11-2, Novembro 2005, pp. 45-60.
- [W3C05] W3C. "World Wide Web Consortium". Capturado em: <http://www.w3.org>, Julho 2006.

- [WOH00] WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.; REGNELL, B.; WESSLÉN, A. Experimentation in software engineering: an introduction. Boston: Kluwer Academic Publishers, 2000, 204 p.


APÊNDICE A – TREINAMENTO PARA OS PARTICIPANTES DO EXPERIMENTO

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação
Grupo de Engenharia de Sistemas Inteligentes




Engenharia de Software Experimental

Rastreabilidade Ontológica sobre o Processo Unificado



ISEG

Rodrigo Noll
Orientador: Marcelo Blois Ribeiro



Agenda

- Estratégias Experimentais
- Engenharia de Software Experimental
 - Definição
 - Planejamento
 - Execução

Rastreabilidade Ontológica sobre o Processo Unificado 2 **PUCRS**

Estratégias Experimentais



- Análise das Características;
- Pesquisa de Opinião (Survey);
- Estudo de Caso;
- Estudo Experimental.

Engenharia de Software Experimental



Definição

- **Objetivo Global**
 - Comparar, no Processo Unificado, a precisão e o esforço da rastreabilidade de artefatos indexada por conceitos da ontologia com relação a tradicional proposta de indexação por requisitos.

Engenharia de Software Experimental



Definição

□ Objetivo do Estudo

- **Comparar** a rastreabilidade de artefatos indexada por requisitos com a indexada por conceitos no Processo Unificado,
- **Com o propósito de** caracterizar o tempo despendido no uso e os elementos recuperados por cada uma das abordagens,
- **Com foco no** esforço e precisão,
- **Sob o ponto de vista** do arquiteto de software,
- **No contexto de** manutenção de um sistema de informação desenvolvido por estudantes (toy sample) no domínio de uma academia desportiva.

Engenharia de Software Experimental




Definição

□ Objetivo de Medição

- Qual o esforço (medido em minutos para cada participante) necessário para indexação dos artefatos utilizando a rastreabilidade apoiada por requisitos e por conceitos;
- Qual a precisão definida através da completude e corretude dos artefatos recuperados através da rastreabilidade indexada por requisitos e conceitos, com relação aos artefatos que realmente se relacionam no sistema.

Engenharia de Software Experimental




Planejamento

- Seleção dos indivíduos de forma totalmente aleatória.

Rastreabilidade Ontológica sobre o Processo Unificado 7 **PUCRS**

Engenharia de Software Experimental




Planejamento

- Instrumentação
 - Objetos:
 - Modelagem UML de um sistema
 - Descrição do Casos de Uso
 - Matriz de rastreabilidade (MS Excel)
 - Ferramenta CASE (ArgoUML+ONTrace)
 - Guias:
 - Treinamento
 - Tutorial sobre como proceder durante o experimento;
 - Métricas:
 - Através de formulários preenchido pelos participantes

Rastreabilidade Ontológica sobre o Processo Unificado 8 **PUCRS**


Engenharia de Software Experimental



Treinamento para os participantes que utilizarão a abordagem de indexação por requisitos.

Rastreabilidade Ontológica sobre o Processo Unificado 9 PUCRS

Engenharia de Software Experimental



Execução – Etapa 1 – Definição dos elos de rastreabilidade

3.1. Emitir pedido de compra

Identificação	UC1
Caso de Uso	Emitir pedido de compra.
Ato	Funcionário.
Pre-Condição	Funcionário estar autenticado no sistema.
Pos-Condição	Pedido de compra cadastrado no sistema.
Cenário	Emitir pedido de compra.

3.1.1. Emitir pedido de compra

Seqüência típica de eventos:

Ação do ator	Resposta do sistema
1. Este caso de uso começa quando o funcionário seleciona a opção Emitir pedido de compra no sistema.	2. Sistema apresenta formulário para inserção do pedido de compra.
3. Funcionário informa a quantidade e o nome do item a ser pedido.	4. Sistema verifica se quantidade e nome do item são reais e se quantidade e número não negativo.
	5. Sistema cadastra pedido de compra com a identificação do funcionário que realizou esta operação, status "em análise" e confirma cadastro do pedido de compra.

Reza Inédia:		Reza Fennel:											
	depende-on	BR	BR2	U01	U04	BR5	U06	BR7	U08	BR9	U10	BR11	U12
Atividade	CP01												
Classe	CP02												
Funcionário	CP03												
Funcionário	CP04												
Funcionário	CP05												
Funcionário	CP06												
Funcionário	CP07												
Funcionário	CP08												
Funcionário	CP09												
Funcionário	CP10												

Rastreabilidade Ontológica sobre o Processo Unificado 10 PUCRS

Engenharia de Software Experimental

Execução – Etapa 1 – Definição dos elos de rastreabilidade

3.1. Emitir pedido de compra

Identificação	UC1
Caso de Uso	Emitir pedido de compra.
Atores	Funcionário
Pre-Condição	Funcionário estar autenticado no sistema.
Pós-Condição	Pedido de compra cadastrado no sistema.
Cenário	Emitir pedido de compra.

3.1.1. Emitir pedido de compra

Seqüência típica de eventos

Ação do ator	Resposta do sistema
1 Este caso de uso começa quando o Funcionário seleciona a opção Emitir pedido de compra no sistema.	2 Sistema apresenta formulário para inserção do pedido de compra.
3 Funcionário informa a quantidade e o nome do ser pedido.	4 Sistema verifica se quantidade e nome do item são nulos e se quantidade é numérico não negativo.
	5 Sistema cadastra pedido de compra com a identificação do funcionário que realizou esta operação, status "em análise" e continua "cadastro do pedido de compra."

Base Inicial:		Base Final:								
depends-on	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09	
Atividade	CP01									
Cliente	CP02									
Ficha Biométrica	CP03									
Ficha Treinamento	CP04									
Funcionários	CP05									
Mensalidade	CP06									
Orcamentos	CP08									
PedidoCompra	CP09									

Rastreabilidade Ontológica sobre o Processo Unificado 11

Engenharia de Software Experimental

Execução – Etapa 2 – Aquisição dos elos

	depends-on	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09
Atividade	CP01			X	X	X				
Cliente	CP02			X	X	X	X			X
Ficha Biométrica	CP03			X	X	X				
Ficha Treinamento	CP04			X	X	X				
Fornecedores	CP05		X						X	
Funcionários	CP06	X	X	X	X	X	X	X	X	X
Mensalidade	CP07									X
Orcamentos	CP08		X							
PedidoCompra	CP09	X	X							

Rastreabilidade Ontológica sobre o Processo Unificado 12

Engenharia de Software Experimental

Execução – Etapa 2 – Aquisição dos elos

	depends-on	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09
Fornecedores	CP05		X						X	
Funcionarios	CP06	X	X	X	X	X	X	X	X	X
Orcamentos	CP08		X							
PedidoCompra	CP09	X	X							

		Identificação dos Elos								
		Artefato			Rastreabilidade					
Diagrama de Classes		CP06	CP08	CP09						
CP05-Fornecedores										

Rastreabilidade Ontológica sobre o Processo Unificado 13

Engenharia de Software Experimental

Execução – Etapa 2 – Aquisição dos elos

	depends-on	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09
Atividade	CP01			X	X	X				
Cliente	CP02			X	X	X	X			X
FichaBiometrica	CP03			X	X	X				
FichaTreinamento	CP04			X	X	X				
Fornecedores	CP05		X						X	
Funcionarios	CP06	X	X	X	X	X	X	X	X	X
Mensalidade	CP07									X
Orcamentos	CP08		X							
PedidoCompra	CP09	X	X							

Rastreabilidade Ontológica sobre o Processo Unificado 14

Engenharia de Software Experimental

Execução – Etapa 2 – Aquisição dos elos

	depends-on	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09
Fornecedores	CP05		X						X	
Funcionarios	CP06	X	X	X	X	X	X	X	X	X

Identificação dos Elos	
Artefato	Rastreabilidade
Diagrama de Classes	CP05-Fornecedores CP06 CP08 CP09

Rastreabilidade Ontológica sobre o Processo Unificado 15 **FUCRS**

Engenharia de Software Experimental

Treinamento para os participantes que utilizarão a abordagem de indexação por conceitos.

Rastreabilidade Ontológica sobre o Processo Unificado 16 **FUCRS**

Engenharia de Software Experimental

Execução – Etapa 1 Definição dos Elos de Rastreabilidade



Engenharia de Software Experimental

Execução – Etapa 2 Aquisição dos Elos de Rastreabilidade

Legenda no Excel

Classe de Projeto	ID	Descrição
Classe de Projeto CP01	Abstrado	
Classe de Projeto CP02	Classica	
Classe de Projeto CP03	FichaBiometrica	
Classe de Projeto CP04	FichaTreinamento	
Classe de Projeto CP05	Fornecedores	
Classe de Projeto CP06	Funcionarios	
Classe de Projeto CP07	Mensalidade	
Classe de Projeto CP08	Orcamentos	
Classe de Projeto CP09	PedidoCompra	

Identificação dos Elos		Rastreabilidade	
Diagrama de	Artefato	Diagrama de	Artefato
CP03 - FichBiometrica	CP02		

Rastreabilidade Ontológica sobre o Processo Unificado 20

FUCRS

Engenharia de Software Experimental

Execução – Etapa 2 Aquisição dos Elos de Rastreabilidade

Legenda no Excel

Tipo	ID	Descrição
Classe de Projeto	CP01	Atividade
Classe de Projeto	CP02	Cliente
Classe de Projeto	CP03	FichaBiometrica
Classe de Projeto	CP04	FichaTreinamento
Classe de Projeto	CP05	Fornecedores
Classe de Projeto	CP06	Funcionarios
Classe de Projeto	CP07	Mensalidade
Classe de Projeto	CP08	Orcamentos
Classe de Projeto	CP09	PedidoCompra

Identificação dos Elos									
Artifato	Rastreabilidade								
CP03 - FichaBiometrica	CP02	CP04							

Rastreabilidade Ontológica sobre o Processo Unificado 21

Engenharia de Software Experimental

Execução – Etapa 2 Aquisição dos Elos de Rastreabilidade

Legenda no Excel

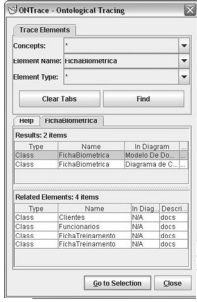
Tipo	ID	Descrição
Classe de Projeto	CP01	Atividade
Classe de Projeto	CP02	Cliente
Classe de Projeto	CP03	FichaBiometrica
Classe de Projeto	CP04	FichaTreinamento
Classe de Projeto	CP05	Fornecedores
Classe de Projeto	CP06	Funcionarios
Classe de Projeto	CP07	Mensalidade
Classe de Projeto	CP08	Orcamentos
Classe de Projeto	CP09	PedidoCompra

Identificação dos Elos									
Artifato	Rastreabilidade								
CP03 - FichaBiometrica	CP02	CP04	CP06						

Rastreabilidade Ontológica sobre o Processo Unificado 22

Engenharia de Software Experimental

Execução – Etapa 2 Aquisição dos Elos de Rastreabilidade



Legenda no Excel

Tipo	ID	Descrição
Classe de Projeto	CP01	Atividade
Classe de Projeto	CP02	Cliente
Classe de Projeto	CP03	FichaBiometrica
Classe de Projeto	CP04	FichaTreinamento
Classe de Projeto	CP05	Fornecedores
Classe de Projeto	CP06	Funcionarios
Classe de Projeto	CP07	Mensalidade
Classe de Projeto	CP08	Orcamentos
Classe de Projeto	CP09	PedidoCompra

Identificação dos Elos	
Artifato	Rastreabilidade
CP03 - FichaBiometrica	

Rastreabilidade Ontológica sobre o Processo Unificado 23

FUCRS

Engenharia de Software Experimental

Execução

- Instruções Gerais:
 - Deve-se fazer o registro do início e término de cada etapa;
 - Não é possível a interação entre os grupos;
 - Manter os celulares desligados;
 - Evitar a interação com ambiente externo ao experimento;
 - Quaisquer dúvidas sobre os procedimentos, perguntar exclusivamente para a equipe responsável;

Rastreabilidade Ontológica sobre o Processo Unificado 24

FUCRS

Engenharia de Software Experimental



Execução

- Observações:
 - O resultado esperado não demanda rapidez, mas atenção e qualidade;
 - Durante a documentação do experimento, os participantes serão referenciados de forma anônima. Não haverá nenhuma referência a nomes.



ISEG

Obrigado
[Dúvidas / Sugestões?]

rnoll@inf.pucrs.br

APÊNDICE B – TUTORIAL PARA RASTREABILIDADE INDEXADA POR CONCEITOS



ENGENHARIA DE SOFTWARE EXPERIMENTAL

RASTREABILIDADE INDEXADA POR CONCEITOS

Rastreabilidade ontológica sobre o Processo Unificado

Mestrando: Rodrigo Perozzo Noll

O presente documento apresenta as instruções necessárias para condução do experimento, utilizando o ArgoUML e suas novas funcionalidades para definição dos elos de rastreabilidade apoiados por conceitos.

Atividade #01: Definição dos elos de rastreabilidade indexados por conceitos

1. **Registrar o horário de início da atividade.**
2. Abra o projeto entregue do ArgoUML chamado "isGym.zargo".
3. Neste projeto, a ontologia ainda não está criada. O projeto contém apenas os diagramas: Modelo de Domínio, Diagrama de Casos de Uso e Diagrama de Classes do Projeto.
4. Para **gerar a ontologia**, clique em **Ontology** no menu superior e selecione a opção **Generate OWL**. Selecione então o modelo a partir do qual será gerada a ontologia (no caso, Modelo de Domínio). Depois de gerada a ontologia, os conceitos relevantes da ontologia estarão na aba inferior, chamada Ontology.
5. Após gerar a ontologia, é necessário **relacionar todas as classes** que compõem o *diagrama de classe de projeto* com os conceitos da ontologia.
6. **Registrar o horário de término da atividade.**

Atividade #02: Rastreabilidade de determinados elementos do modelo

1. Após relacionar as classes aos conceitos, é possível recuperar os elos de rastreabilidade. Para verificar, vá ao menu superior **Ontology->Ontrace**
2. Os resultados compreendem os elementos que se relacionam diretamente ou indiretamente com os conceitos da ontologia. Na aba de *Help* do diálogo *Ontrace* existe a explicação de como é gerada essa rastreabilidade, caso haja interesse.

Vocês deverão rastrear os artefatos definidos na planilha "*Rastreabilidade*" do arquivo "Rastreabilidade_conc.xls". Para cada artefato solicitado, vocês deverão completar a lista de rastreabilidade, com base na legenda apresentada na planilha "*Legendas*".

APÊNDICE C – TUTORIAL PARA RASTREABILIDADE INDEXADA POR REQUISITOS



ENGENHARIA DE SOFTWARE EXPERIMENTAL

RASTREABILIDADE INDEXADA POR REQUISITOS

Rastreabilidade ontológica sobre o Processo Unificado
Mestrando: Rodrigo Perozzo Noll

Atividade #01: Definição dos elos de rastreabilidade indexados por requisitos

7. **Registrar o horário de início da atividade.**
8. Abra a descrição dos casos de uso chamada "isGym.pdf".
9. Para cada descrição de caso de uso, identificar as classes de projeto que estão relacionadas.
10. A relação das classes de projeto definidas encontram-se na planilha "Matriz Requisitos" do arquivo "Rastreabilidade_req.xls". Para cada caso de uso definido nesta matriz, identificar quais as classes relacionadas.
11. Repetir as atividades acima para cada caso de uso.
12. **Registrar o horário de término da atividade.**

Atividade #02: Rastreabilidade de determinados elementos do modelo

3. Após relacionar os requisitos as classes, é necessário recuperar os elos de rastreabilidade.
4. Para cada classe definida na planilha "Rastreabilidade", identificar quais os casos de uso que estão relacionadas à mesma.
5. Para cada caso de uso identificado, recuperar todas as classes que eles se relacionam utilizando a opção de filtro na tabela.
6. Preencher a planilha de "Rastreabilidade" com as classes associadas ao caso de uso.
7. Repetir o procedimento para todas as classes definidas nesta planilha.

APÊNDICE E – AVALIAÇÃO QUALITATIVA DAS ABORDAGENS DE RASTREABILIDADE



ENGENHARIA DE SOFTWARE EXPERIMENTAL

Rastreabilidade ontológica sobre o Processo Unificado
Mestrando: Rodrigo Perozzo Noll

Avaliação Qualitativa da Proposta de Rastreabilidade

(1) Muito Ruim	(2) Ruim	(3) Regular	(4) Boa	(5) Muito Boa
Q1: Como você considera a usabilidade desta técnica de rastreabilidade?				
Q2: Como você considera a utilidade desta técnica de rastreabilidade?				

(1) Muito trabalhoso	(2) Pouco trabalhoso	(3) Regular	(4) Cômodo	(5) Muito Cômodo
Q3: Como você considera o esforço para aprendizagem da técnica?				
Q4: Como você considera o esforço para a indexação dos elementos do modelo utilizando esta técnica de rastreabilidade?				
Q5: Como você considera o esforço para recuperação de elementos relacionados utilizando esta técnica de rastreabilidade?				

(1) Discordo Plenamente	(2) Discordo	(3) Não discordo nem concordo	(4) Concordo	(5) Concordo Plenamente
Q6: Você usaria na prática novamente esta técnica de rastreabilidade?				
Q7: A técnica de rastreabilidade utilizada atende o que se propôs ?				

ANEXO A – TABELA DE COMPONENTES DA UML

Área	Visão	Diagrama	Conceito
Estrutural	Visão Estática	Diagrama de Classe	Classe, Associação, Generalização, Dependência, Realização, Interface.
	Visão de Casos de Uso	Diagrama de Caso de Uso	Caso de Uso, Ator, Associação, Extensão, Inclusão, Generalização.
	Visão de Implementação	Diagrama de Componente	Componente, Interface, Dependência, Realização.
	Visão de Implantação	Diagrama de Implantação	Nodo, Componente, Dependência, Localização.
Dinâmico	Visão de Máquina de Estados	Diagrama de Estados	Estado, Evento, Transição, Ação.
	Visão de Atividades	Diagrama de Atividades	Estado, Atividade, Transição Completa, Fork, Join.
	Visão de Interação	Diagrama de Sequência	Interação, Objeto, Mensagem, Ativação.
		Diagrama de Colaboração	Colaboração, Interação, Perfil de Colaboração, Mensagem.
Modelo de Gerência	Visão de Modelo de Gerência	Diagramas de Classe	Pacote, Subsistema, Modelo.
Estensibilidade	Todos	Todos	Restrições, Estereótipos, Tagged Values.

ANEXO B – TABELA DE DISTRIBUIÇÃO DE T

Graus de Liberdade	Probabilidade, p a 0.05
1	12.71
2	4.30
3	3.18
4	2.78
5	2.57
6	2.45
7	2.37
8	2.31
9	2.26
10	2.23
11	2.20
12	2.18
13	2.16
14	2.14
15	2.13
16	2.12
17	2.11
18	2.10
19	2.09
20	2.09
21	2.08
22	2.07
23	2.07
24	2.06
25	2.06
26	2.06
27	2.05
28	2.05
29	2.05
30	2.04
40	2.02
60	2.00
120	1.98
∞	1.96