

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**BRUNO NANDOLPHO MACHADO**

**DOCUMENTAÇÃO SEMÂNTICA NA  
ENGENHARIA DE REQUISITOS**

**VITÓRIA  
2012**

BRUNO NANDOLPHO MACHADO

# **DOCUMENTAÇÃO SEMÂNTICA NA ENGENHARIA DE REQUISITOS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientador: Prof. Dr. Ricardo de Almeida Falbo

**VITÓRIA  
2012**

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

M149d Machado, Bruno Nandolpho, 1985-  
Documentação semântica na engenharia de requisitos /  
Bruno Nandolpho Machado. – 2012.  
148 f. : il.

Orientador: Ricardo de Almeida Falbo.  
Dissertação (Mestrado em Informática) – Universidade  
Federal do Espírito Santo, Centro Tecnológico.

1. Documentação. 2. Semântica. 3. Engenharia de  
requisitos. 4. Ontologias (Recuperação da informação). I. Falbo,  
Ricardo de Almeida. II. Universidade Federal do Espírito Santo.  
Centro Tecnológico. III. Título.

CDU: 004

---

# Documentação Semântica na Engenharia de Requisitos

**Bruno Nandolpho Machado**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 13/04/2012 por:



Prof. Dr. Ricardo de Almeida Falbo, Dsc - DI/UFES



Prof<sup>a</sup>. Dr<sup>a</sup>. Monaléssa Perini Barcellos - DI/UFES



Prof<sup>a</sup>. Dr<sup>a</sup>. Flávia Maria Santoro, DSc - UFRJ/UNIRIO

“... e o saber se  
multiplicará”.

Dan. 12:4

## **AGRADECIMENTOS**

Agradeço a Deus pela vida que me deu e que sustenta até hoje, e pelo conhecimento que conquistei até agora, mas peço a Ele para me dar sabedoria para conquistar muito mais.

Agradeço a meus pais, Dilson e Tereza, que me apoiaram e entenderam minha ausência em compromissos familiares, os dias que vivi apenas em meu quarto e as constantes tarefas que deixei acumular pela falta de tempo (como vasilhas na pia).

A minha noiva Jaice, que me deu apoio e também me teve pelas metades nesse período turbulento chamado mestrado. Enfim, os feriados e finais de semana serão utilizados para passeios.

Aos amigos do NEMO e aos que fizeram mestrado comigo, que me ajudaram a manter o ritmo e não achar que eu era o único que perdia horas de sono.

Ao Ricardo Falbo, pelo exemplo e postura como orientador e professor, pela paciência e puxões de orelha na hora certa. Sinto apenas por não ter usufruído mais de seus conhecimentos e de seu convívio por fazer a maior parte do mestrado também trabalhando.

Aos amigos de trabalho da PRODEST, Caio e Vinícius, que possibilitaram as diversas saídas para que pudesse cumprir meus créditos.

Aos amigos de fora do círculo da computação, que ouviram as conversas mais esquisitas sobre o assunto, mas ainda perguntavam: “Do que é seu mestrado mesmo?”.

A todos que tive contato nesses anos de faculdade e mestrado e que contribuíram para que chegasse esse momento, fico muito feliz por participarem de alguma forma da minha vida.

## RESUMO

Documentos vêm desempenhando um papel chave na difusão do conhecimento humano, sendo utilizados na maioria das organizações como forma de armazenamento do conhecimento. Entretanto, a maioria do conhecimento armazenado em documentos eletrônicos está disponível apenas para ser interpretado por humanos e não por sistemas computacionais.

A comunidade de Web Semântica defende que, para que o conteúdo disponível na web esteja acessível para sistemas (ou agentes inteligentes), é necessário adicionar metadados baseados em ontologias de domínio ao conteúdo das páginas. Essa ideia aplica-se também no contexto de documentos gerados a partir de ferramentas *desktop*, como o Microsoft Word e o Open Office Writer, transformando esses documentos em documentos semânticos. Neste contexto, há algumas infraestruturas baseadas em ontologias de apoio à gerência de documentos semânticos, desenvolvidas para apoiar e facilitar o processo de adição de metadados nos documentos, além de permitir a extração, armazenamento e busca de conteúdo a partir de anotações semânticas. Dentre elas, há a Plataforma de Gerenciamento de Documentos Semânticos (PGDS) desenvolvida por Arantes (2010) no Núcleo de Estudos em Modelagem Conceitual e Ontologia (NEMO).

Esta dissertação tem por objetivo especializar a PGDS para apoiar a documentação semântica de requisitos, de modo a prover funcionalidades de apoio específicas para o domínio da Engenharia de Requisitos. Para tal, foi feita uma avaliação preliminar da aplicação da PGDS a esse domínio através de seu uso prático, o que permitiu identificar diversas oportunidades de melhoria. A partir das oportunidades levantadas, novas funcionalidades gerais foram adicionadas à PGDS, evoluindo-a, bem como foi idealizada uma especialização da PGDS para o domínio da Engenharia de Requisitos, com o objetivo de explorar os recursos providos por ela para criar funcionalidades específicas para apoiar os processos envolvidos e os problemas desse domínio, não se limitando apenas em manter a PGDS como uma plataforma de aplicação geral. Para tal, foi também desenvolvida uma extensão da Ontologia Requisitos usada em (ARANTES, 2010).

Palavras chave: Documentação Semântica, Anotações Semânticas, Engenharia de Requisitos, Ontologias.

## ABSTRACT

Documents have played a key role in the dissemination of human knowledge, being used in most organizations as a way of storing knowledge. However, most of the knowledge stored in electronic documents is available only to be interpreted by humans and not by computer systems.

The Semantic Web community argues that to make available web content accessible to systems (or intelligent agents), it is necessary to add ontology-based metadata to page content. This idea can also be applied in the context of documents generated using desktop tools such as Microsoft Word and Open Office Writer, transforming these documents into semantic documents. In this context, there are some ontology-based infrastructures for supporting semantic document management. These infrastructures are developed to support and facilitate the process of adding metadata to documents, and allow content extraction, storage and retrieval from semantic annotations. Among them, there is the Semantic Document Management Platform (SDMP) developed by Arantes (2010) in the Ontology & Conceptual Modeling Research Group

This dissertation aims to specialize the SDMP for supporting requirements semantic documentation, in order to provide support features aimed at the field of Requirements Engineering (RE). To this end, we performed a preliminary assessment of the SDMP application to this area through its practical use, which allowed identifying several improvement opportunities. From these improvement opportunities, new general features were added to the SDMP, and we designed a specialization of it to the domain of Requirements Engineering. This specialization aims to explore the resources provided by the SDMP for creating specific functionalities to support RE processes and problems related to them. In this way, the SDMP is no longer limited to general application. For developing the RE specialization we also developed an extension of the Requirements Ontology used in (ARANTES, 2010).

Key-words: Semantic Documentation, Semantic Annotations, Requirements Engineering, Ontologies.

## LISTA DE FIGURAS

Figura 2.1 – Classificação de Ontologias proposta por Guarino (1998) .....	17
Figura 2.1 - (a) Modelo parcial da ontologia de domínio definida para o <i>template</i> semântico M1, (b) <i>Template</i> Semântico baseado na ontologia de domínio e (c) Documento semântico gerado a partir do <i>template</i> M1.....	27
Figura 2.2 – (a) Aplicação de um estilo de formatação e (b) Definição de um campo de usuário através da interface ODF. ....	29
Figura 2.3 – <i>Template</i> contendo uma tabela de alocações de Analistas em Projetos .....	32
Figura 2.3 – Processo de geração de versão e execução do MEVID (ARANTES, 2010).....	34
Figura 2.4 – Atividades do Processo de Engenharia de Requisitos .....	36
Figura 2.4 – Mudanças em Requisitos (MACHADO et al., 2011).....	44
Figura 3.1 – Fragmento do <i>Template</i> de Especificação de Requisitos do NEMO .....	49
Figura 3.2 – Criação de uma nova seção .....	50
Figura 3.3 – Anotação da subseção 3.1 para capturar o subsistema.....	51
Figura 3.4 – Fragmento do <i>template</i> de Documento de Especificação de Requisitos .....	53
Figura 3.5 – Fragmento de <i>template</i> de Documento de Especificação de Requisitos reformulado.....	54
Figura 3.6 – Parte do <i>Template</i> do Documento de Especificação de Requisitos.....	56
Figura 3.7 – <i>Template</i> com a nova anotação de tipo de tabela.....	58
Figura 3.8 – Ilustração do mecanismo de busca do nó tabela na árvore ODF .....	59
Figura 3.9 – Formulário de Consulta da PGDS para visualizar as diferenças existentes entre versões de um mesmo documento em um repositório semântico. ....	60
Figura 3.10 – Novo formulário para visualizar a diferença semântica evolutiva existente entre versões de um mesmo documento em um repositório semântico. ....	61
Figura 3.11 - Resultados da consulta para diferença semântica evolutiva existente entre versões de um mesmo documento.....	62
Figura 3.13 – Formulário de busca para rastrear a evolução dos estados do indivíduo em um documento particular, dado um repositório semântico. ....	64
Figura 3.14 – Passo a passo da busca para rastrear a evolução dos estados de um indivíduo em um documento particular, dado um repositório semântico. ....	65
Figura 3.15 – Estado global do indivíduo no Repositório Semântico. ....	66
Figura 3.16 – Funcionalidade para cadastrar interesse em indivíduo .....	68
Figura 3.17 – Consulta de Garantia da Qualidade.....	70
Figura 3.18 – Cadastro de item de Verificação do Checklist de Qualidade .....	71
Figura 3.19 – Diagrama de Sequência da Interação entre Cliente e o Webservice REST de PGDS .	72
Figura 3.21 – Edição da anotação semântica para a anotação TITULO_PROJETO.....	74
Figura 3.22 - Criação de uma instrução de instanciação para a anotação TITULO_PROJETO .....	74
Figura 3.23 – Edição da anotação semântica para tabela TIPOCASOUSOCADASTRAL .....	76
Figura 3.24 – <i>Template</i> de Especificação de Requisitos na PGDS com os estilos e campos de usuário definidos .....	77

Figura 4.1 – Fragmento da Ontologia de Requisitos Estendida (adaptado de (ARANTES, 2010))	81
Figura 4.2 – Ontologia de Requisitos Reformulada: Módulo principal	83
Figura 4.3 – Ontologia de Requisitos: Módulo de Casos de Uso	87
Figura 4.4 – Ontologia de Requisitos: integração com uma possível Ontologia de Testes	88
Figura 4.5 – Funcionalidade de Priorização de Requisitos	94
Figura 4.6 – Fragmento da Tabela de Requisitos Funcionais do Documento de Requisitos do Projeto control_carga (Controle de Cargas)	95
Figura 4.7 – Alerta de Sugestão de Priorização	96
Figura 4.8– Sugestão de Priorização	97
Figura 4.9 – Relações de Rastreabilidade	99
Figura 4.10 – Rastreabilidade Horizontal Requisito x Requisitos Dependentes	100
Figura 4.11 – Rastreabilidade Vertical com destaque para a relação entre Requisito e Caso de Teste	101
Figura 4.12 – Matriz de Rastreabilidade Vertical Requisito x Caso de Teste	102
Figura 4.13 – Análise de Impacto de Mudança no Requisito RF03	104
Figura 4.14 – Checklist de Requisitos	107
Figura 4.15 – Relatório de Mudanças em Documento de Requisitos	109
Figura 4.16 – Consulta de indivíduos especializada para o domínio de requisitos	110

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	MOTIVAÇÃO	9
1.2	OBJETIVOS DO TRABALHO	10
1.3	HISTÓRICO DO TRABALHO	11
1.4	ORGANIZAÇÃO DO TRABALHO	12
<b>2</b>	<b>DOCUMENTAÇÃO SEMÂNTICA E ENGENHARIA DE REQUISITOS</b>	<b>14</b>
2.1	DOCUMENTAÇÃO SEMÂNTICA E ONTOLOGIAS	14
2.1.1	Documentação Semântica	14
2.1.2	Ontologias	16
2.1.3	Ontologia de Fundamentação UFO e OntoUML	19
2.1.4	Ontologias E Documentação Semântica	21
2.2	INFRAESTRUTURAS DE APOIO À DOCUMENTAÇÃO SEMÂNTICA	22
2.3	PGDS - PLATAFORMA DE GERÊNCIA DE DOCUMENTOS SEMÂNTICOS	25
2.3.1	Módulo de Anotação em Modelos de Documento (MAMD)	26
2.4	ENGENHARIA DE REQUISITOS	35
2.4.1	Levantamento	37
2.4.2	Análise e Negociação	37
2.4.3	Documentação	38
2.4.4	Verificação e Validação	39
2.4.5	Gerência de Requisitos	39
2.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO	40
<b>3</b>	<b>EVOLUÇÃO DA PLATAFORMA DE GERÊNCIA DE DOCUMENTOS SEMÂNTICOS</b>	<b>43</b>
3.1	ANÁLISE PRELIMINAR DA PGDS	43
3.2	EVOLUÇÃO DA LINGUAGEM E TIPOS DE ANOTAÇÕES	47
3.2.1	Anotação em Seções e Subseções	48
3.2.2	Anotação para Ignorar Linhas de uma Tabela	52
3.2.3	Nova Anotação para Tipo de Tabela	55
3.3	EVOLUÇÃO DAS FUNCIONALIDADES EXISTENTES NA PGDS	59
3.4	NOVAS FUNCIONALIDADES	68
3.4.1	Garantia da Qualidade	69
3.4.2	WebService para Consulta Externa	71
3.4.3	Apoio à Criação de Anotações Semânticas	72
3.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO	77

<b>4</b>	<b>ESPECIALIZAÇÃO DA PLATAFORMA DE GERÊNCIA DE DOCUMENTOS SEMÂNTICOS PARA APOIAR A ENGENHARIA DE REQUISITOS</b> .....	<b>79</b>
4.1	MOTIVAÇÃO.....	79
4.2	ONTOLOGIA DE REQUISITOS.....	79
4.2.1	Ontologia de Requisitos: Versões Anteriores.....	80
4.2.2	Ontologia de Requisitos: Nova Versão.....	81
4.2.3	Projeto e Implementação da Nova Versão da Ontologia.....	89
4.3	APOIO À ENGENHARIA DE REQUISITOS USANDO A PGDS.....	93
4.3.1	Priorização de Requisitos.....	93
4.3.2	Rastreabilidade horizontal e vertical.....	98
4.3.3	Análise de impacto de alteração em requisito.....	102
4.3.4	Garantia da Qualidade de Requisitos.....	105
4.3.5	Notificação de Alterações no Documento de Requisitos.....	108
4.3.6	Consultar Indivíduos do domínio.....	109
4.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	110
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>112</b>
5.1	CONCLUSÕES.....	112
5.2	LIMITAÇÕES E PERSPECTIVAS FUTURAS.....	113
	<b>Referências Bibliográficas</b> .....	<b>116</b>
	<b>Apêndice A – Modelos de Documento e Anotações</b> .....	<b>122</b>

# 1 INTRODUÇÃO

Este capítulo tem como objetivo principal apresentar a motivação para o desenvolvimento deste trabalho (Seção 1.1), apresentar o objetivo geral e os objetivos específicos deste trabalho (Seção 1.2), descrever um histórico do desenvolvimento da pesquisa (Seção 1.3) e por fim apresentar como esta dissertação está organizada e estruturada (Seção 1.4).

## 1.1 MOTIVAÇÃO

Documentos são importantes recursos descrevendo o que uma organização sabe. Estima-se que entre 80 a 85% da informação armazenada pelas companhias esteja na forma de documentos, seja na forma de resultados de pesquisas, métodos, soluções de projeto, políticas de segurança ou procedimentos para operações. Dessa forma, os repositórios de documentos que surgem em organizações são geralmente complexos e neste cenário, técnicas eficientes de busca e recuperação crescem em importância, à medida que o volume de documentos cresce. Tendo isso em vista, um gerenciamento eficiente de documentos é uma parte importante da estratégia de gestão de conhecimento de uma organização (UREN et al., 2006).

Entretanto, a maioria do conhecimento em documentos eletrônicos está disponível apenas para ser interpretado por humanos e não por sistemas computacionais, tornando mais complexa a tarefa de indexação e recuperação eficiente desses recursos. Na Web Semântica, para que o conteúdo disponível na web esteja acessível para sistemas, advoga-se que é necessário adicionar metadados baseados em ontologias ao conteúdo exposto em páginas web. Seguindo essa visão, é possível definir anotações em conteúdos web, originadas a partir de ontologias de domínio, com o intuito de enriquecer esse conteúdo, e permitir que haja um processamento semântico do mesmo (BERNERSLEE et al., 2001). Surge, então, o termo anotação semântica, que é o ato de se adicionar metadados, transformando um documento sintático em um documento semântico. Esse mesmo conceito pode ser utilizado não somente na web, mas também com documentos gerados a partir de ferramentas *desktop*.

Ontologias desempenham um papel fundamental na anotação semântica, visto que proveem um arcabouço para a conceituação e modelagem de conhecimento compartilhado.

Alguns trabalhos têm sido desenvolvidos na literatura com objetivo de se criar infraestruturas de apoio à documentação semântica, tais como (TALLIS, 2003), (LANFRANCHI et al., 2005), (ERIKSSON, 2007) e (NESIC, 2010). Essas infraestruturas apresentam funcionalidades úteis, como *plugins* para automatizar as anotações semânticas e serviços de busca e indexação de conteúdo semântico utilizando ontologias de domínio. Dentre esses trabalhos, há a Plataforma de Gerência de Documentos Semânticos do NEMO (PGDS) desenvolvida por Arantes (2010) que tem objetivos e funcionalidades em comum com as existentes na literatura.

Um ponto notado é que, apesar dessas infraestruturas utilizarem ontologias de domínios específicos, tais como domínios comercial, militar, governo e organizações, as mesmas possuem apenas funcionalidades de caráter geral para trabalhar com ontologias. Muitas vezes, para prover um apoio mais efetivo ao domínio, é necessário explorar de modo específico a conceituação definida na ontologia. Em outras palavras, é preciso conhecer os elementos da ontologia (seus conceitos, relações e propriedades) e explorá-los em funcionalidades específicas de domínio, de modo a apoiar tarefas do domínio.

Dessa maneira, vislumbrou-se a possibilidade de estender a PGDS para um domínio específico. Dado que em (ARANTES, 2010) uma utilização inicial da PGDS foi feita no domínio de requisitos, foi proposto que esse domínio fosse a Engenharia de Requisitos.

## 1.2 OBJETIVOS DO TRABALHO

Considerando as motivações apresentadas, o objetivo geral deste trabalho é especializar a Plataforma de Gerência de Documentos Semânticos (PGDS) para apoiar a documentação semântica de requisitos, de modo a prover funcionalidades de apoio específicas para o domínio da Engenharia de Requisitos.

Esse objetivo geral pode ser detalhado nos seguintes objetivos específicos:

- **Evoluir a PGDS em suas funcionalidades gerais:** uma vez que a PGDS foi fruto do trabalho de pesquisa de Arantes (2010), é um objetivo específico deste trabalho aperfeiçoá-la, identificando oportunidades de melhoria e procurando eliminar ou minimizar limitações identificadas.
- **Evoluir a Ontologia de Requisitos:** de modo a ampliar a cobertura da Ontologia de Requisitos proposta inicialmente em (ARANTES, 2010), a mesma deve ser evoluída

de modo a considerar conceitos, relações e propriedades necessários para explorar algumas das potencialidades da PGDS quando aplicada ao domínio da Engenharia de Requisitos.

- **Desenvolver funcionalidades específicas de apoio à Engenharia de Requisitos:** tomando por base a conceituação estabelecida pela nova versão da Ontologia de Requisitos e a nova versão da PGDS, desenvolver funcionalidades específicas que apoiem tarefas realizadas no contexto da Engenharia de Requisitos, obtendo vantagens da utilização da PGDS como base para essas funcionalidades.

### 1.3 HISTÓRICO DO TRABALHO

Para alcançar os objetivos listados na seção anterior, foram realizadas as atividades descritas a seguir.

Em primeiro lugar, foi realizado um levantamento bibliográfico relacionado com documentação semântica, verificando como esse tema vem sendo tratado pela comunidade científica. Nesse levantamento, diversos trabalhos têm sido desenvolvidos para criação de infraestruturas para apoiar a Documentação Semântica de maneira a prover um ferramental para automatizar os diversos processos envolvidos. É possível verificar nesses trabalhos a tendência de se utilizar documentos combinados com ontologias, com o intuito de prover semântica e permitir que usuários acessem o conhecimento de múltiplas formas. Dentre esses trabalhos temos a Plataforma de Gerência de Documentos Semânticos do NEMO (PGDS) desenvolvida por Arantes (2010). A PGDS foi construída para apoiar a construção de modelos de documentos (*templates*) com anotações semânticas, utilizando em adição um sistema de controle de versão com o objetivo de manter o histórico de alterações não só dos documentos semânticos gerados a partir dos modelos semânticos, mas principalmente do conteúdo extraído de cada versão desses documentos. O trabalho desenvolvido por Arantes (2010) foi alvo de um estudo profundo e a PGDS foi utilizada como base deste trabalho.

Em um segundo momento, a PGDS foi alvo de uma avaliação preliminar prática no contexto da disciplina de Engenharia de Requisitos (MACHADO et al., 2011). A plataforma foi utilizada para gerenciar documentos de requisitos gerados por grupos de alunos e então coletar informações, identificando uma série de oportunidades de melhoria com o intuito de aprimorar aspectos que ainda apresentavam limitações, além de motivar a criação de novas

funcionalidades. Nesse momento também foi considerada uma análise das infraestruturas para apoiar a Documentação Semântica existentes na literatura e suas funcionalidades, comparando-as com as existentes na versão original da PGDS. Assim, uma frente de trabalho foi posta na direção de realizar melhorias levantadas na plataforma.

Por outro lado, dados os resultados obtidos com a análise preliminar de aplicação da PGDS no domínio de requisitos e as consequentes oportunidades de melhoria levantadas relativas ao domínio da Engenharia de Requisitos, vislumbramos a utilidade de especializar a PGDS para esse domínio. A ideia seria explorar as funcionalidades gerais providas pela PGDS para desenvolver novas facilidades, agora específicas para apoiar a realização de atividades da Engenharia de Requisitos. Surgiu, então, nesse momento uma segunda frente de trabalho, relacionada com a especialização da PGDS.

Para realizar a extensão proposta, foi necessário estender também a Ontologia de Requisitos utilizada na análise preliminar da PGDS, ontologia essa proposta em (ARANTES, 2010), para contemplar outros conceitos relacionados à Engenharia de Requisitos. Para tal, foi realizado um levantamento da literatura sobre o assunto para que os novos conceitos, atributos e relações tivessem uma base conceitual de consenso em relação ao discutido na literatura relativa a essa área. A extensão da ontologia permitiu que novos conceitos fossem tratados pela plataforma e anotados nos documentos relacionados a requisitos, possibilitando que novas funcionalidades pudessem ser desenvolvidas para explorar os novos conceitos passíveis de serem anotados nos documentos nesse domínio.

Por fim, de posse da nova versão da Ontologia de Requisitos e a nova versão da PGDS, foram desenvolvidas funcionalidades específicas para apoiar atividades do processo de Engenharia de Requisitos, dentre elas Análise e Negociação de Requisitos, Verificação e Validação e Gerência de Requisitos.

## **1.4 ORGANIZAÇÃO DO TRABALHO**

A estrutura desta dissertação está organizada em cinco capítulos, sendo o primeiro este capítulo introdutório, e os demais conteúdos desdobrados da seguinte forma:

- *Capítulo 2 - Documentação Semântica e Engenharia de Requisitos:* apresenta uma revisão da literatura sobre Documentação Semântica, Ontologias e Engenharia de Requisitos, que foram a base para a condução deste trabalho.
- *Capítulo 3 - Evolução da Plataforma de Gerência de Documentos Semânticos:* apresenta a evolução da Plataforma de Gerência de Documentos Semânticos, apresentando as melhorias realizadas nessa plataforma no contexto deste trabalho.
- *Capítulo 4 – Especialização da Plataforma de Gerência de Documentos Semânticos para apoiar a Engenharia de Requisitos:* apresenta a especialização da plataforma para o domínio da Engenharia de Requisitos, apresentando a evolução da Ontologia de Requisitos e as funcionalidades desenvolvidas.
- *Capítulo 5 – Considerações Finais:* apresenta as conclusões do trabalho, as contribuições, limitações e propostas de trabalhos futuros.

## **2 DOCUMENTAÇÃO SEMÂNTICA E ENGENHARIA DE REQUISITOS**

Este capítulo tem como objetivo principal apresentar a fundamentação teórica para o trabalho realizado. A Seção 2.1 aborda os temas Documentação Semântica e Ontologias, apresentando também a utilização conjunta de ambos. A Seção 2.2 apresenta algumas infraestruturas de documentação semântica existentes na literatura. A Seção 2.3 apresenta a plataforma de gerência de documentos semânticos (PGDS) desenvolvida por Arantes (2010), a qual foi aperfeiçoada e estendida neste trabalho. A Seção 2.4 apresenta uma breve revisão sobre Engenharia de Requisitos, tema relevante, tendo em vista que a especialização da PGDS realizada neste trabalho foi feita para esse domínio. A Seção 2.5 apresenta trabalhos relacionados à utilização de documentação semântica na Engenharia de Software. Finalmente, a Seção 2.6 apresenta as considerações finais do capítulo.

### **2.1 DOCUMENTAÇÃO SEMÂNTICA E ONTOLOGIAS**

#### **2.1.1 Documentação Semântica**

Documentos são importantes recursos descrevendo o que uma organização sabe. Estima-se que entre 80 a 85% da informação armazenada pelas companhias esteja na forma de documentos. De fato, para alguns profissionais documentos são efetivamente o produto que eles vendem. Exemplos desses produtos incluem contratos e relatórios de consultorias e pesquisas de consumidores. Dessa maneira, documentos vêm desempenhando um papel chave na difusão do conhecimento humano e continuam a ter importância na era da comunicação digital e da Internet, na qual documentos textuais armazenam grande parte do conhecimento (UREN et al., 2006; ERIKSSON, 2006).

O centro da memória organizacional geralmente consiste de grandes conjuntos de documentos internos que descrevem eventos passados, resultados de pesquisas, métodos, soluções de projeto, políticas de segurança, procedimentos para operações etc. Como esses repositórios de documentos são geralmente complexos, técnicas eficientes de busca e recuperação crescem em importância, à medida que esse volume cresce. Tendo isso em vista, um gerenciamento eficiente de documentos é uma parte importante da estratégia de gerência de conhecimento de uma organização (ERIKSSON, 2007). De certa maneira, a introdução e aceitação de documentos eletrônicos nas recentes décadas têm permitido o

desenvolvimento desses repositórios de documentos baseados em computadores, uma vez que as versões impressas estão sendo substituídas (ERIKSSON; BANG, 2006). Entretanto, a maioria do conhecimento em documentos eletrônicos está disponível apenas para ser interpretado por humanos e não por sistemas computacionais.

No contexto da Internet e da Web Semântica, para que o conteúdo disponível na web esteja acessível para sistemas (ou agentes inteligentes) buscarem, processarem e interpretarem, é necessário adicionar metadados ao conteúdo exposto em páginas web (BERNERSLEE et al., 2001). Esses metadados vão além de simples informações de autoria das páginas web ou apenas parte de seu conteúdo (como palavras chave, por exemplo). Metadado, geralmente interpretado como “dados sobre dados”, pode ser usado como um mecanismo para expressar semântica de um dado com o objetivo de facilitar a busca, recuperação, entendimento e utilização do dado (SICILIA, 2006). Dessa maneira, a visão da web semântica é ter na Web dados definidos e ligados de uma maneira que possam ser utilizados por máquinas não apenas para apresentação desses dados, como é feito tradicionalmente, mas para automação, integração e reúso de dados entre várias aplicações (BERNERSLEE et al., 2006). Nessa evolução de Web Sintática para Web Semântica, surge o termo anotação semântica.

Para Kiryakov (2003), anotação semântica é a utilização e geração de metadados específicos com o objetivo de permitir novos métodos de acesso à informação, estendendo os já existentes. Por ter sido cunhado em meio à visão da web semântica, normalmente o termo anotação semântica é associado ao ato de anotar recursos encontrados na Web com metadados, mas seu conceito vai além e não se prende unicamente a recursos encontrados somente na Web, podendo ser aplicado a documentos gerados a partir de ferramentas *desktop*, como o Microsoft Word ou Open Office Writer, ou até mesmo imagens (ARANTES, 2010).

Ao se adicionar anotações semânticas em um documento, alcançamos uma nova geração de documentos que podem ser referenciados como Documentos Inteligentes ou Documentos Semânticos, que são documentos que “sabem” sobre seu próprio conteúdo com o objetivo de que processos automatizados “saibam o que fazer com ele” (UREN et al., 2006). Neste texto adotamos o termo Documento Semântico para referenciar documentos *desktop* que foram enriquecidos com anotações semânticas.

Voltando ainda a ideia da web semântica, em prol da interoperabilidade entre sistemas produtores e consumidores de conteúdo, significados bem definidos devem ser garantidos entre quem anota o conteúdo disponível em uma página web e quem a consome. Dadas as

diversas fontes de dados heterogêneas existentes na Web, o grande desafio para a Web Semântica é prover uma linguagem capaz de descrever dados em diferentes sistemas de representação de conhecimento existentes na Web (BERNERSLEE et al., 2001). Para Sicilia (2006), além da linguagem para descrever esses metadados necessitar de representações de conhecimento compartilhadas, deve ser provido um vocabulário básico para que declarações de metadados possam ser construídas.

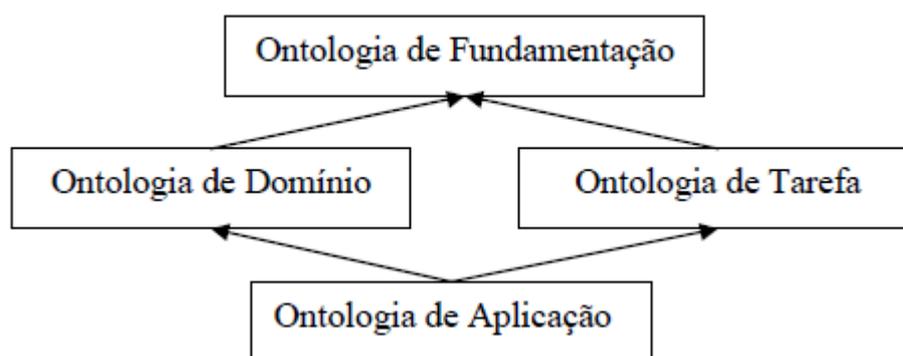
Pelo fato de que Ontologias proveem um arcabouço para a conceituação e modelagem de conhecimento compartilhado para diversas áreas (GRUBER, 1993), diversos autores apostam em ontologias como uma saída para a viabilização da Web Semântica (KIRYAKOV, 2004; BERNERSLEE et al., 2001; VARGAS-VERA et al, 2001; UREN et al., 2006; FENSEL, 2003). Por outro lado, recentes pesquisas no campo da utilização de semântica no contexto de aplicações *desktop* têm mostrado que um número de características de tecnologias da Web Semântica são também aplicáveis para o problema da gerência de informações em documentos *desktop*, seja na habilidade de representar dados em um formato genérico e independente de aplicação, como na maneira de descrever os recursos com um vocabulário formalizado e de maneira flexível (SCHANDL; HASLHOFER, 2009). Portanto, ontologias podem desempenhar também um papel fundamental na anotação semântica em documentos em formato *desktop*.

### 2.1.2 Ontologias

Uma definição muito citada para o termo ontologia é a sugerida por Gruber (1993), que diz que uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada. Nessa definição, conceituação se refere a um modelo abstrato de um fenômeno do mundo que identifica seus conceitos relevantes. Explícita significa que os conceitos usados e as restrições sobre seu uso são definidos explicitamente. Formal é referente ao fato de ser passível de entendimento por máquinas. Compartilhada reflete o fato de uma ontologia dever capturar o conhecimento consensual aceito por uma comunidade.

Guarino (1998) sugere uma classificação de ontologias em quatro tipos, a saber: (i) Ontologias de Fundamentação ou de topo, que descrevem conceitos muito gerais, como espaço, tempo, problema, objeto, evento, ação etc., os quais são independentes de um problema ou domínio particular; (ii) ontologias de domínio, que descrevem o vocabulário relacionado a um domínio específico como, por exemplo, medicina, direito, automóveis etc.;

(iii) ontologias de tarefa, que descrevem o vocabulário relacionado a uma tarefa genérica, como, por exemplo, diagnose, venda etc.; e por fim, (iv) ontologias de aplicação, que descrevem conceitos dependentes tanto de um domínio quanto de uma tarefa particulares, os quais são, frequentemente, especializações de ontologias relacionadas. A Figura 2.1 mostra o relacionamento existente entre os tipos de ontologias apresentados. Ontologias de fundamentação são a base para a construção de ontologias de domínio e de tarefa, onde termos de fundamentação são especializados. Ademais, quando essas são combinadas, geralmente entidades de domínio desempenham papéis enquanto executam certa tarefa.



**Figura 2.1 – Classificação de Ontologias proposta por Guarino (1998)**

Um ponto importante que deve ser enfatizado é a diferente utilização do termo ontologia em Sistemas de Informação, Inteligência Artificial e a comunidade da Web Semântica. Em Sistemas de Informação, o termo ontologia tem sido utilizado em conformidade com as suas definições em filosofia, que é um sistema de categorias independente da linguagem. Em contraste, na maior parte das demais áreas da computação (as duas últimas áreas incluídas), o termo ontologia é, em geral, utilizado como um artefato de engenharia concreto concebido para um fim específico e representado em uma linguagem específica (GUIZZARDI, 2007).

À luz dessas noções contrastantes de ontologias, Guizzardi (2007) levanta uma série de questões: O que exatamente é uma ontologia de domínio? Como isso se relaciona com outras representações concretas, tais como modelos conceituais e metamodelos? Como é relacionada à ontologia no sentido filosófico? Além disso, durante os anos muitas linguagens têm sido utilizadas para representar ontologias de domínio. Exemplos incluem o cálculo de predicados, KIF, Ontolingua, UML, EER, LINGO, ORM, da CML, DAML + OIL, F-Logic, OWL. Quais são as características que uma linguagem adequada para representar modelos

conceituais, em geral, e ontologias de domínio, em particular, devem ter? Em particular, são as linguagens da Web Semântica linguagens de representação adequadas para ontologias?

Para tentar responder a essas perguntas, Guizzardi (2007) advoga a necessidade de duas classes de linguagens de representação complementares na Engenharia de Ontologias: uma contendo linguagens filosoficamente bem fundamentadas, com foco em expressividade e clareza conceitual, e a outra com linguagens focadas em preocupações computacionais (por exemplo, decidibilidade, raciocínio automatizado eficiente etc.). Essa ideia é análoga à utilizada na disciplina de Engenharia de Software, onde há uma distinção clara entre modelagem conceitual, projeto e implementação. Na modelagem conceitual, uma especificação independente de solução é produzida, cujo objetivo é fazer uma descrição clara e precisa dos elementos de domínio para fins de comunicação, aprendizagem e resolução de problemas. Na fase de projeto, essa especificação conceitual é transformada em uma especificação de projeto, levando em consideração uma série de questões como estilos arquitetônicos e critérios não-funcionais de qualidade a serem maximizados. A mesma especificação conceitual pode potencialmente ser usada para produzir um número de projetos (designs) diferentes. Finalmente, na fase de implementação, um projeto é codificado em uma linguagem alvo a ser então implantada em um ambiente computacional. Mais uma vez, a partir do mesmo desenho, diferentes implementações podem ser produzidas.

Na fase de modelagem conceitual na Engenharia Ontologia, linguagens altamente expressivas devem ser usadas para criar ontologias fortemente axiomatizadas que se aproximam tão bem quanto possível da ontologia ideal do domínio. O objetivo dessas linguagens é a adequação de representação, uma vez que as especificações resultantes são destinadas a ser utilizadas pelos seres humanos em tarefas como comunicação, análise de domínio e resolução de problemas. As ontologias de domínio resultantes (ontologias de referência) devem ser utilizadas de forma *off-line* para ajudar os seres humanos em tarefas como negociação de significado e estabelecimento de consenso. Por outro lado, uma vez que os utilizadores estão de acordo com uma conceituação comum, versões de uma ontologia de referência podem ser criadas. Essas versões são nomeadas na literatura de ontologias leves (*lightweight ontologies*). Ao contrário das ontologias de referência, ontologias leves não estão focadas na adequação da representação, mas são projetadas com o foco na garantia de propriedades computacionais desejáveis. Um exemplo de linguagem adequada à criação de ontologias leves é a OWL (GUIZZARDI, 2007).

As necessidades levantadas pela definição de Guarino (1998) com a utilização de ontologias de fundamentação para descrever conceitos muito gerais e que são a base para a construção de ontologias de domínio reforçam a utilidade de um arcabouço rico em expressividade e clareza para a modelagem conceitual de ontologias (ontologias de referência). Por outro lado, a definição de Gruber (1993) vai de encontro à representação de ontologias leves no nível de implementação: representações explícitas e compartilhadas, tendo como foco o seu entendimento por máquinas. Dessa maneira, com relação à definição de ontologia utilizada neste trabalho, adotamos o termo **ontologia** para designar o que Guizzardi chama de ontologias de domínio de referência, enquanto adotamos o termo **implementação da ontologia de domínio** para designar ontologias leves. Assim, ambas as definições, em seu devido lugar, apresentam visões complementares do conceito de ontologia no processo de Engenharia de Ontologias.

Com base no exposto, neste trabalho adotamos o seguinte processo em duas etapas na criação das ontologias:

- Modelagem Conceitual: produz uma especificação de solução independente de implementação, que tem por objetivo fazer uma descrição clara e precisa dos elementos de domínio para fins de comunicação, aprendizagem e resolução de problemas.
- Projeto e Implementação: Transforma o modelo conceitual da ontologia levando em consideração o ambiente computacional, codificando a mesma em uma linguagem processável por máquina.

Dessa maneira, como linguagem para modelagem conceitual das ontologias de domínio foi utilizada a OntoUML, linguagem essa que utiliza a Ontologia de Fundamentação Unificada (*Unified Foundational Ontology – UFO*), que é apresentada em seguida. Por fim, a implementação da ontologia de domínio foi codificada utilizando a linguagem OWL (MCGUINESS; HARMELEN, 2010).

### 2.1.3 Ontologia de Fundamentação UFO e OntoUML

Ontologias de fundamentação servem como base para o estabelecimento de consenso e negociação entre humanos. Elas têm sido utilizadas com sucesso para melhorar a qualidade de linguagens de modelagem e modelos conceituais (GUIZZARDI, 2005b). Ontologias de fundamentação proveem um vocabulário de conceitos que são independentes

de um problema ou domínio particular. Neste trabalho, a Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* – UFO) (GUIZZARDI, 2005b) foi utilizada como base para a criação de ontologias de domínio em nível conceitual (ontologia de referência).

UFO tem sido desenvolvida baseada em um número de teorias das áreas de Ontologias Formais, Lógica Filosófica, Filosofia da Linguagem, Linguística e Psicologia Cognitiva. Ela é dividida em três partes complementares: a UFO-A é uma ontologia de indivíduos duradouros (*endurants*) e é o cerne da UFO; a UFO-B é uma ontologia de eventos (*perdurants*); por fim, a UFO-C é uma ontologia de entidades sociais, construída sobre as partes A e B da UFO. Em sua tese, Guizzardi (2005b) propõe um perfil UML ontologicamente correto, chamada posteriormente de OntoUML. Esse perfil é uma extensão da UML 2.0, ontologicamente bem fundamentada e possui um metamodelo isomórfico à UFO-A. OntoUML permite produzir ontologias de qualidade e usá-las como um modelo conceitual de alta expressividade.

Para a criação de ontologias de domínio propostas nesse trabalho, foram suficientes os conceitos definidos na UFO-A, sendo relevantes os seguintes conceitos: *Kind*, *Subkind*, *Role*, *Category* e *Mode*. Para um entendimento melhor, esses conceitos são descritos a seguir. Vale ressaltar que o conteúdo apresentado abaixo foi elaborado com base em (GUIZZARDI, 2005b) e (ZAMBORLINI, 2011). Foram apresentados os termos em inglês, bem como nos diagramas posteriores, de modo a facilitar uma ligação com as principais referências à UFO, quase todas elas escritas em inglês.

**Rigidez e Anti-rigidez:** Alguns conceitos têm a característica de serem necessariamente aplicáveis a seus indivíduos enquanto eles existirem. Essa característica é definida em UFO-A como rigidez. Por exemplo, o conceito Pessoa é dito **rígido**, pois todos os indivíduos por ele classificados não podem deixar de ser pessoas enquanto existirem. Caso um indivíduo pessoa João, por exemplo, deixe de ser pessoa, necessariamente ele deve deixar de existir. Em contrapartida, outros conceitos têm a característica de serem aplicáveis a seus indivíduos de maneira contingente, isto é, indicam uma possibilidade. O conceito Estudante é **anti-rígido**, pois todos os indivíduos por ele classificados, e.g. João, podem deixar de ser estudantes e ainda continuar existindo.

Em OntoUML, uma classe com o estereótipo «**kind**» representa um conceito com a propriedade de rigidez e com a característica de prover um critério ou princípio de identidade para seus indivíduos que permita distingui-los e contá-los. Uma heurística útil para chegar a tal conclusão é verificar se é possível contar os indivíduos. Podemos utilizar como exemplo os conceitos de Maça, Carro ou Pessoa, dentre outros. Vemos que tais conceitos proveem

um critério de identidade aos indivíduos por ele classificados e que faz sentido perguntar “quantas maçãs há nesta caixa?”, ou “quantos carros existem nesse estacionamento” ou “quantas pessoas visitaram o museu?”. Em contrapartida, o conceito Vermelho não provê tal critério, pois não faz sentido perguntar “quantos vermelhos há nesta caixa?”. Considerando-se a impressão digital como o princípio de identidade provido a toda instância do conceito Pessoa, este é do tipo *Kind*, enquanto os conceitos Homem e Mulher são do tipo *SubKind* (estereótipo «**subkind**»), pois, além de serem rígidos, especializam o conceito Pessoa e, portanto, herdaram dele o princípio de identidade.

Por outro lado, uma classe com o estereótipo «**role**» representa um conceito com a propriedade de anti-rigidez, mais precisamente um conceito que tem participação em um evento ou numa determinada relação. Isto é, algo que pode ser assumido em um “mundo”, mas necessariamente não em todos os mundos (anti-rígido e relacionalmente dependente). Por exemplo, Mãe é um papel para o *SubKind* Mulher mediante a existência da relação de maternidade com uma instância do papel (*role*) Filho do tipo (*kind*) Pessoa.

Continuando, o estereótipo «**category**» é usado para representar entidades que pertencem a *kinds* diferentes, mas que compartilham uma propriedade comum essencial (ou seja, uma propriedade que eles não podem deixar de ter). Por exemplo, uma categoria Entidade Racional pode ser especializada pelas espécies Pessoa e Agente Artificial (*kinds* diferentes).

Por fim, o estereótipo «**mode**» é usado para representar entidades que todas as instâncias são existencialmente dependentes de uma outra entidade (e exatamente esta entidade). Exemplos típicos de modos são cor, sintoma etc., pois as mesmas só existem se existir, por exemplo, um carro para possuir a cor ou uma pessoa para possuir o sintoma.

#### 2.1.4 Ontologias E Documentação Semântica

Documentos semânticos têm como objetivo combinar documentos e ontologias, permitindo com que usuários acessem o conhecimento de múltiplas formas. Adicionando anotações e incluindo ontologias em documentos eletrônicos, grandes repositórios de documentos podem se beneficiar, facilitando as buscas e a recuperação de informações, além de prover um formato de representação e comunicação que permita tanto leitura humana e quanto processamento por máquinas, de modo a utilizarem uma estrutura comum para representação do conhecimento (ERIKSSON, 2007).

A partir de uma base de dados com conteúdo semântico indexado, consultas semânticas baseadas em ontologias são viáveis de serem feitas, sendo possível especificar o tipo de entidade, restrições, atributos e relações com outras entidades de interesse, permitindo uma pesquisa muito mais avançada (KIRYAKOV, 2004).

Por outro lado, com ontologias de domínio definidas, anotações semânticas podem ser criadas em torno de um documento com a ajuda de ferramentas de apoio. A partir de então, é possível extrair e relacionar o conteúdo semântico de um documento de acordo com a ontologia de domínio utilizada. De posse desse conteúdo e do conteúdo de outros documentos que permeiam a organização, há a possibilidade de se prover uma integração de informações, de forma a unir os conteúdos extraídos e possibilitando uma visão mais abrangente sobre o conhecimento de uma organização (ARANTES, 2010).

Entretanto, o real uso e sucesso de anotações em documentos semânticos dependem fortemente do custo e esforço para que sejam feitas as anotações nesses documentos, uma vez que os autores de documentos não podem rotineiramente sofrer com essa sobrecarga. Assim sendo, é necessário prover ferramentas para reduzir o esforço requerido para criação do conteúdo do documento e suas anotações semânticas associadas (UREN et AL., 2006; TALLIS, 2003).

Nesses cenários, diversas infraestruturas foram propostas visando facilitar o processo de anotação semântica, além de prover recursos para a recuperação, indexação e armazenamento dos conteúdos semânticos. A próxima subseção apresenta brevemente algumas infraestruturas propostas para apoiar a diversas facetas da documentação semântica.

## **2.2 INFRAESTRUTURAS DE APOIO À DOCUMENTAÇÃO SEMÂNTICA**

Diversos trabalhos têm sido desenvolvidos para criação de infraestruturas para apoiar a Documentação Semântica de maneira a prover um ferramental para automatizar os diversos processos envolvidos. A seguir, são apresentadas infraestruturas existentes na literatura desenvolvidas com esse propósito. Essas infraestruturas estão relacionadas com a utilização de metadados e ontologias para enriquecer documentos, com automatização e apoio à criação de anotações e funcionalidades específicas para esse propósito, além de mecanismos para gerenciar documentos semânticos nas plataformas desktop e web e também a possibilidade de se realizar extração e busca semântica no conteúdo gerado.

PDFTab (ERIKSSON, 2007) é uma extensão da ferramenta Protègè (ambiente de desenvolvimento de ontologias) que apóia a criação de documentos semânticos. PDFTab permite aos usuários importar documentos em formato PDF no próprio Protègè e adicionar *links* para ontologias utilizando classes e propriedades da ontologia como anotação. Em outras palavras, PDFTab tira vantagem do rico ambiente Protègè de criação de ontologias para criar documentos semânticos e do fato de que o formato PDF tem suporte à marcação e comentários em palavras e regiões de página, registrando esses comentários dentro do próprio arquivo PDF. Os metadados adicionados referenciam algum conceito ou indivíduo de uma ontologia de domínio implementada em OWL. Para tal, um conjunto de ontologias permite o estabelecimento de relações entre anotações semânticas (Ontologia de Anotações), elementos encontrados em um documento (Ontologia de Documentos) e elementos de uma ontologia de domínio. Diversas funcionalidades podem ser implementadas em torno do repositório de documentos semânticos criado com PDFTab, tais como: (i) raciocínio para que sejam gerados novos fatos a partir dos dados existentes na base de dados; (ii) verificação de consistência com o objetivo de encontrar padrões e relacionamentos não desejáveis no repositório de dados, tais como relacionamentos contraditórios e inconsistência de dados; (iii) navegação sobre os dados contidos no repositório semântico provenientes dos documentos, oferecendo, dessa forma, uma visão gráfica dos documentos e dos seus respectivos dados; e (iv) buscas semânticas, onde um motor de busca possa procurar pelos sinônimos do texto procurado com base na ontologia de domínio.

SemanticWord (TALLIS, 2003) estende o Microsoft Word para permitir anotações semânticas. Primeiramente, foram adicionadas na barra de ferramentas do MS Word funcionalidades para anotar trechos de texto diretamente no momento de criação do conteúdo do documento, além de mostrar anotações junto ao texto e apoiar a manipulação delas utilizando diretamente o mouse. Os metadados adicionados nos trechos de textos são salvos no formato de uma tripla (base de relação na web semântica: sujeito – predicado – objeto). Em segundo lugar, SemanticWord integra o MS Word com o AeroDAML, um sistema de extração automática de informação que analisa e anota o texto do documento que está sendo digitado, parecendo para o autor um serviço análogo à checagem de ortografia do MS Word. Por fim, SemanticWord permite a rápida composição de textos anotados através da instanciação de *templates*, permitindo criar espaços reservados onde uma referência para uma instância de uma ontologia será carregada quando o texto for preenchido.

AktiveDoc (LANFRANCHI et al., 2005) é um sistema de apoio à gerência de conhecimento no processo de edição e leitura de documentos. Sua principal funcionalidade é apoiar usuários (autores e leitores) a compartilharem e reusarem conhecimento com anotações em momento oportuno. Ela permite a anotação de documentos em três níveis: anotação baseada em ontologia, declarações em texto livre e enriquecimento sob demanda. O módulo de ontologia também é responsável por interpretar ontologias em RDF carregadas pelo usuário e visualizá-las de maneira apropriada. Enquanto muitos sistemas modificam o formato original do documento para adicionar anotações, AktiveDoc salva as anotações em uma base de dados separada. Os documentos são salvos em sistemas de gerência de conhecimento, onde cada documento é logicamente associado a suas anotações. Vale ressaltar que AktiveDOc é uma aplicação cliente-servidor integrada na Web e provê tanto anotações manuais quanto semi-automáticas (através do módulo sob demanda). Uma das maneiras de se utilizar o conteúdo semântico armazenado é através do módulo de informação, onde sugestões ativas de conteúdo relevante para os autores são apresentados mediante a troca de conhecimento, via *web services* semânticos, de outras bases de conhecimento, via ontologia, ou da própria base da ferramenta.

Nesic (2010) propõe em sua tese de doutorado uma arquitetura para gerência de documentos semânticos chamada de SDArch, na qual há funcionalidades divididas em três processos principais: criação de documentos semânticos, busca semântica e navegação em documentos semânticos. A criação de documentos semânticos é baseada na transformação automática de documentos desktop convencionais em documentos semânticos utilizando ontologias de domínio, onde descrições RDF são adicionadas em unidades do documento através de anotações semânticas feitas por um *plugin* desenvolvido para o MS Office 2007 chamado SemanticDoc. Neste contexto, são unidades de documento conteúdos como seções, parágrafos, slides, tabelas, gráficos, vídeo etc., sendo que esses podem ser anotados em documentos MS Word e MS PowerPoint. Por outro lado, a busca e a navegação semântica em documentos são realizadas utilizando as unidades semânticas anotadas no documento e as ligações semânticas entre essas unidades e ontologias de domínio. Inicialmente a busca pode ser feita pelo usuário em um texto livre, sendo transformada em uma consulta a ser executada no índice de conceitos de SDArch. Depois dessa busca, serviços utilizam as ligações entre os índices para recuperar nos documentos as unidades anotadas com aqueles conceitos, além de utilizar as preferências do usuário para reordenar o resultado da busca, sendo esse processo chamado de personalização da busca semântica em documentos. Por último, a navegação semântica é realizada através de consultas ao repositório RDF das ligações semânticas que conectam as unidades de textos dos documentos. Existe também um módulo para navegação semântica onde é possível

através da URI do documento e um conceito, formular uma consulta SPARQL e executá-la em cima do repositório RDF existente.

Em (POPOV et al., 2006) foi desenvolvido um sistema colaborativo que tem como objetivo gerenciar documentos eletrônicos *desktop* utilizando tecnologias Web 2.0, descrevendo como sistemas *desktop* podem utilizar recursos web para enriquecer documentos com metadados. Em sua abordagem, de posse de um documento (focado principalmente em documentos em formato PDF, é possível realizar anotações com metadados utilizando a ferramenta de anotação proposta pelos autores. Esses metadados utilizam a plataforma XMP<sup>1</sup>, que possibilita inserir diretamente em um documento metadados escritos em RDF. De posse dos documentos anotados, *wrappers* utilizam o motor Jena para fazer a extração do conteúdo a partir de uma ontologia de domínio e utilizam o Apache Lucene<sup>2</sup> para indexar os metadados. Esses documentos, por fim, são armazenados em um repositório. Existe, ainda, um componente de busca que permite que sejam realizadas buscas baseadas em ontologias, utilizando um editor de ontologia e uma busca baseada em *tags* e palavras-chave.

A próxima subseção apresenta de maneira mais detalhada a infraestrutura de gerenciamento de documentos semânticos definida em (ARANTES, 2010), que foi a infraestrutura utilizada para o desenvolvimento deste trabalho. Essa infraestrutura faz uso de ontologias de domínio no apoio à anotação semântica, além de prover uma forma de anotação baseada em modelos de documentos (*templates*). Ademais, é possível também gerenciar as diversas versões de documentos semânticos *desktop*, sendo o conteúdo semântico de cada versão extraído e ficando disponível para realizar buscas semânticas.

## 2.3 PGDS - PLATAFORMA DE GERÊNCIA DE DOCUMENTOS SEMÂNTICOS

A Plataforma de Gerência Documentos Semânticos - PGDS (ARANTES, 2010) foi construída para apoiar a construção de modelos de documentos (*templates*) com anotações semânticas, utilizando em adição o sistema de controle de versão Subversion (COLLINS-SUSSMAN et al., 2008) com o objetivo de manter o histórico de alterações não só dos documentos semânticos gerados a partir dos modelos semânticos, mas principalmente do conteúdo semântico extraído de cada versão desse documento encontrado na plataforma.

---

<sup>1</sup> Adobe XMP, <http://www.adobe.com/products/xmp/>, visitado em 05/03/2012.

<sup>2</sup> Apache Lucene. <http://lucene.apache.org/java/docs>, visitado em 10/03/2012

Na arquitetura proposta, o conteúdo de um documento ou de sua versão considerado é o conjunto de indivíduos que têm um mapeamento direto para conceitos e propriedades de uma ontologia de domínio e passível de extração automática. Elementos do documento que não possuem esse mapeamento são desconsiderados. Esses indivíduos e os valores de suas propriedades estão descritos no documento na forma de estruturas do mesmo, a saber: trechos de texto e células de tabela.

Vale ressaltar que as ontologias utilizadas na PGDS são ontologias leves codificadas na linguagem OWL (*Ontology Web Language*) (MCGUINESS; HARMELEN, 2010), atual padrão de implementação de ontologias na *web* (HORROCKS et al. 2003).

A plataforma possui um repositório central de documentos semânticos responsável por armazenar os documentos anotados e manter o histórico de evolução dos mesmos. Esse repositório faz uso do sistema de gerência de configuração Subversion junto com o banco de dados Postgres. A ideia é que, para cada grupo de documentos coesos, seja criada uma instância de um repositório subversion e o mesmo seja associado a um registro no banco de dados que indica o caminho para esse repositório. Por exemplo, em uma empresa de desenvolvimento de software, cada projeto seria um agrupamento coeso de documentos e teria uma instância de repositório separada. Além do repositório, ela possui ainda três módulos principais, a saber: Módulo de Anotação em Modelos de Documento (MAMD), Módulo de Extração, Versionamento e Integração de Dados (MEVID), Módulo de Busca e Rastreabilidade (MBR). Esses módulos são descritos a seguir, sendo que o MAMD é descrito mais detalhadamente devido à sua complexidade e relevância para este trabalho.

### **2.3.1 Módulo de Anotação em Modelos de Documento (MAMD)**

Este módulo provê um ferramental para efetuar anotações semânticas em modelos de documentos de texto (*templates*) no formato ODF, baseadas em ontologias de domínio. Dessa forma, instâncias de documento criadas a partir desse modelo são enriquecidas de maneira indireta sem a intervenção humana, como ilustra a Figura 2.1.

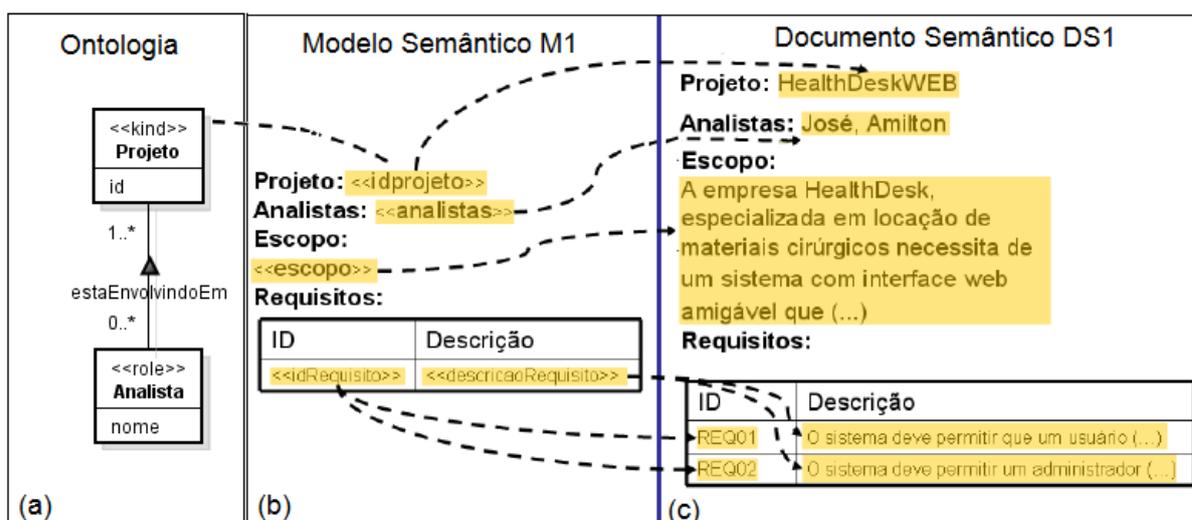


Figura 2.1 - (a) Modelo parcial da ontologia de domínio definida para o *template* semântico M1, (b) *Template* Semântico baseado na ontologia de domínio e (c) Documento semântico gerado a partir do *template* M1

O modelo (*template*) semântico previamente anotado apresentado na Figura 2.1(b) mostra, por exemplo, um trecho do *template* com a anotação `<<idprojeto>>`. Essa anotação poderia corresponder à criação de uma instância do conceito *Projeto* existente na ontologia de domínio (Figura 2.1(a)) utilizando o trecho como identificador do conceito. Ao se utilizar o modelo semântico M1, cria-se um documento semântico DS1 (Figura 2.1(c)). Dessa maneira, é feita uma ligação entre o conteúdo adicionado no espaço reservado `<<idprojeto>>` (no caso, **HealthDeskWeb**) no DS1 e esse valor é utilizado para criar uma instância do conceito *Projeto*.

Para conseguir tal efeito, a plataforma dispõe de uma linguagem especializada para realizar anotações semânticas em um documento ODF. Atualmente, duas estruturas de conteúdo ODF podem ser anotadas: fragmentos de texto e tabelas. Ambas permitem que o responsável pela criação da anotação possa adicionar um conjunto de ações a serem tomadas quando o texto for analisado pelo Módulo de Extração, Versionamento e Integração de Dados (MEVID). Essas instruções representam tanto a criação de indivíduos como de relacionamentos definidos nas ontologias de domínio.

Nessa linguagem, a sintaxe da instrução responsável por criar indivíduos durante a extração do conteúdo semântico de um documento é a seguinte:

```
instance(arg, concept, accessVariable)
```

Após a execução dessa instrução, um novo indivíduo é criado, sendo este uma instância do conceito passado como segundo argumento da instrução (*concept*). Além disso,

todos os indivíduos possuem um identificador como propriedade. No caso da anotação, esse identificador é gerado a partir do valor do primeiro argumento (*arg*). O terceiro argumento (*accessVariable*) é uma referência para o indivíduo criado no mapa de referências gerado pelo MEVID. Dessa forma esse indivíduo pode ser referenciado em relacionamentos.

Para a criação de relacionamentos entre indivíduos e entre indivíduos e valores, a sintaxe da instrução é a seguinte:

```
property(arg1, property, arg2)
```

Essa instrução representa triplas sujeito - predicado - objeto, onde o sujeito é o indivíduo referenciado no primeiro argumento (*arg1*), o predicado é o segundo argumento (*property*) e o objeto é uma referência a outro indivíduo ou a um valor passado como terceiro argumento (*arg2*).

Para criar anotações semânticas, dois elementos de ODF são utilizados. O primeiro deles são os estilos de formatação de texto. Um estilo de formatação de texto define uma série de características visuais que um trecho de texto com esse estilo vai apresentar. Todo estilo de formatação em ODF possui um nome único. O segundo elemento de ODF utilizado são os campos de usuário. Em ODF, um campo de usuário permite a associação entre um nome e um valor textual, ambos especificados pelo próprio usuário. Um campo de usuário não é necessariamente visível durante a edição do documento, podendo ser adicionado ao conteúdo do mesmo quando necessário. A Figura 2.2(a) abaixo mostra em um documento ODF o momento onde o estilo “ênfase forte” é aplicado ao trecho “Documento de Requisitos”. Já a Figura 2.2(b) mostra o momento de criação de um campo de usuário que armazenará um valor no formato texto, bem como a definição do seu valor (123456).

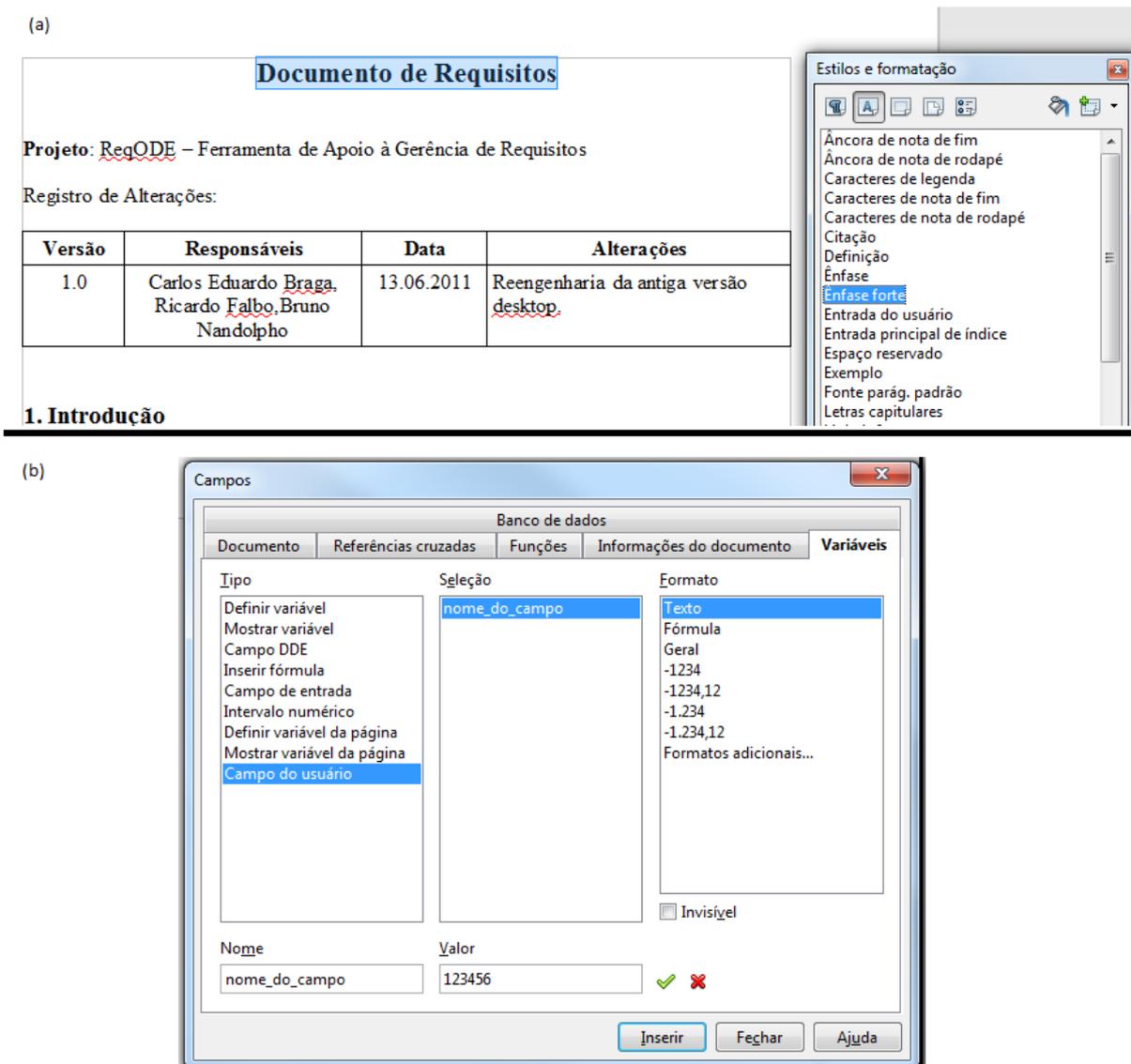


Figura 2.2 – (a) Aplicação de um estilo de formatação e (b) Definição de um campo de usuário através da interface ODF.

Como dito anteriormente, é possível realizar a anotação semântica em dois elementos ODF, que são descritos detalhadamente a seguir:

### Trechos de Texto

1. Deve-se criar um estilo de formatação seguindo o padrão **SemanticAnnotation-ref-<nomeCampoUsuario>**. O valor de <nomeCampoUsuario> é usado para definir o campo de usuário que armazenará o conjunto de instruções dessa anotação semântica.

2. Criar um campo de usuário com o mesmo nome do valor definido em `<nomeCampoUsuario>` no passo 1 que contenha uma das seguintes seqüências: (i) `[[textspan]]` seguido do conjunto de instruções separadas por vírgula ou (ii) `[[break with'<separador>']]`, também seguido do conjunto de instruções.

As palavras-chaves `[[textspan]]` e `[[break with'<separador>']]` representam ações pré-definidas para o MEVID realizar em cima do restante das instruções contidas no campo de usuário. Elas têm o seguinte comportamento: Quando uma anotação se inicia com `[[textspan]]`, significa que o MEVID irá utilizar o trecho anotado como um conteúdo único, que será armazenado em uma variável global acessível pela palavra-chave `{content}`. Voltando ao nosso exemplo, se fossemos criar o conjunto de instruções referente à criação do conceito *Projeto* mostrado na Figura 2.1, ela teria da seguinte forma:

```
[[textspan]];instance({content},http://localhost/ontologies/SE/onto.owl#Projeto,$projeto);
```

Ou seja, MEVID identificará que se trata de uma anotação do tipo `textspan` e irá substituir a palavra-chave `{content}` pelo conteúdo anotado (*HealthDeskWeb*). Ao final da execução, ainda teríamos uma referência para a instância de projeto *HealthDeskWeb* no mapa de variáveis do MEVID com a chave *projeto*. Por fim, o resultado dessa execução para o documento DS1 gera um conteúdo semântico em termos de RDF corresponde ao trecho abaixo:

```
<rdf:Description rdf:about="http://localhost/ontologies/SE/onto.owl#HealthDeskWEB">
  <rdf:type rdf:resource="http://localhost/ontologies/SE/onto.owl#Projeto"/>
</rdf:Description>
```

De outra maneira, quando uma anotação se inicia com `[[break with'<separador>']]` significa que o MEVID irá utilizar o trecho anotado como um conjunto de vários itens. Para tal, MEVID utiliza o símbolo definido em `<separador>` para quebrar o trecho de texto anotado nos vários itens, sendo que a variável de iteração que contém o conteúdo do item corrente é a palavra-chave `{slice}`. Sendo assim, MEVID executará o conjunto de instruções para cada item definido. Tomemos como exemplo uma proposta de anotação para extrair os analistas definidos para o projeto, conforme definido no *template* da Figura 2.1(b). O conjunto de instruções referente a essa anotação é:

```
[[break with ', ']]
instance({slice},http://localhost/ontologies/SE/onto.owl#Analista,$var_analista);
property($var_analista,http://localhost/ontologies/SE/onto.owl#alocadoEm,$projeto);
```

Utilizando novamente como exemplo o documento semântico da Figura 2.1(c), MEVID identificará que se trata de uma anotação do tipo lista de itens e irá quebrar o trecho anotado pelo separador ‘,’, gerando 2 *slices*: { “José” e “Amilton”}. Para cada *slice* será executada a seguinte sequência de instruções:

```
instance({slice},http://localhost/ontologies/SE/onto.owl#Analista,$var_analista);
```

- Criação de uma instância do conceito *Analista*, tendo como identificador o conteúdo definido na variável *slice* e armazenamento da referência para essa instância na variável *\$var\_analista* no mapa de variáveis.

```
property($var_analista,http://localhost/ontologies/SE/onto.owl#alocadoEm,$projeto);
```

- Criação do relacionamento *estaEnvolvidoEm* entre os conceitos cuja referência está na variável *\$var\_analista* com a referência que está na variável *\$projeto*.

Esse procedimento é realizado para cada *slice*, gerando um conteúdo semântico em termos de RDF corresponde ao trecho abaixo:

```
<rdf:Description rdf:about="http://localhost/ontologies/SE/onto.owl#Jose">
  <estaEnvolvidoEm rdf:resource="http://localhost/ontologies/SE/onto.owl#HealthDeskWeb"/>
  <rdf:type rdf:resource="http://localhost/ontologies/SE/onto.owl#Analista"/>
</rdf:Description>

<rdf:Description rdf:about="http://localhost/ontologies/SE/onto.owl#Amilton">
  <estaEnvolvidoEm rdf:resource="http://localhost/ontologies/SE/onto.owl#HealthDeskWeb"/>
  <rdf:type rdf:resource="http://localhost/ontologies/SE/onto.owl#Analista"/>
</rdf:Description>
```

### ***Anotação em Tabela***

A anotação semântica em tabelas não requer a criação de um estilo de formatação como as anotações de trechos de texto. A anotação em tabelas requer apenas que o nome da tabela tenha o mesmo nome do campo de usuário que irá conter o conjunto de instruções a ser processado para aquela tabela.

Para cada linha da tabela, o conjunto de instruções referentes a uma coluna específica (encontrado no conteúdo do campo de usuário) é executada. Dessa forma, toda célula de uma tabela anotada tem seu conjunto de instruções executado. O código do conteúdo do

campo de usuário usado no nome da tabela deve ter o seguinte formato: **[[at <numeroColuna>]]**, seguido de um conjunto de instruções, onde *<numeroColuna>* corresponde ao número da coluna da tabela, sendo 0 a primeira coluna. Para cada coluna, pode-se definir um conjunto de instruções específico. Se o campo de usuário não contemplar uma das colunas da tabela, esta não é considerada como sendo anotada semanticamente. Assim, uma tabela com duas colunas poderia ser anotada da seguinte forma:

1. [[at 0]]
2. (conjunto de instruções)
3. [[at 1]]
4. (conjunto de instruções)

A linha 1 do código acima define que a coluna sendo tratada é a coluna 0 (a primeira da tabela). Na linha 2 é definido o conjunto de instruções a serem executadas quando o MEVID estiver tratando a primeira célula de qualquer linha dessa tabela.

Para ilustrar, seja o template definido na Figura 2.3, que define uma tabela com conteúdo de alocações de analistas para projetos, onde a primeira coluna representa o nome do Analista e a segunda coluna representa o nome do Projeto ao qual o analista está alocado. Para anotar semanticamente essa tabela, foi definido (como poder ser visto pela seta 1 na Figura 2.3) o nome dessa tabela como **“tabelaAnalistas”**.

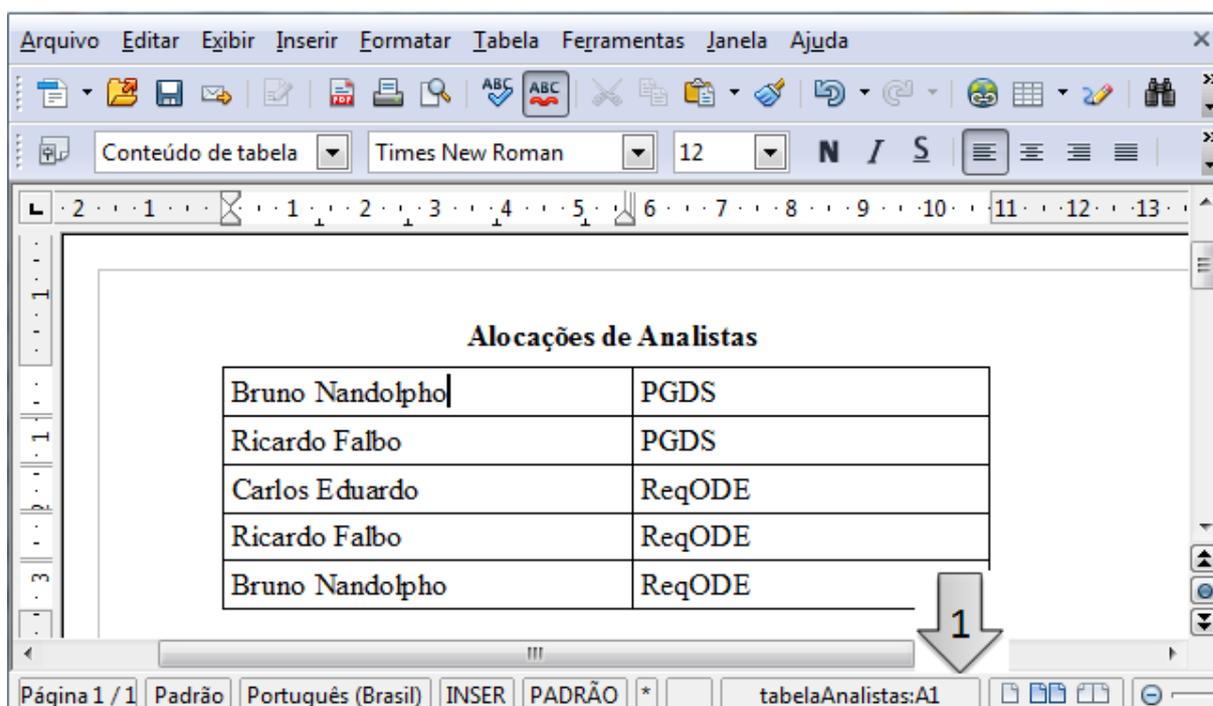


Figura 2.3 – Template contendo uma tabela de alocações de Analistas em Projetos

Além disso, foi criado também um campo de usuário de nome **tabelaAnalistas** contendo no seu valor o seguinte conjunto de instruções:

```

1. [[at 0]]
2.     instance({content},
       http://localhost/ontologies/SE/onto.owl#Analista,$var_analista););
3. [[at 1]]
4.     instance({content},
       http://localhost/ontologies/SE/onto.owl#Projeto,$var_projeto););
5.     property($var_analista,
       http://localhost/ontologies/SE/onto.owl#alocadoEm,$projeto);

```

É importante notar que o conteúdo semântico gerado por essa tabela é muito parecido com o conteúdo semântico gerado pelas anotações definidas para trechos de texto anteriormente, visto que descrevem alocações de analista não apenas em um projeto, mas em vários. Dessa maneira, é possível observar que a plataforma já oferece flexibilidade na definição anotações para estruturas de texto que se deseje registrar conteúdos semânticos, permitindo a criação de *templates* variados.

Por fim, essa linguagem é utilizada em conjunto diretamente com o módulo MEVID, que é descrito a seguir.

### **Módulo de Extração, Versionamento e Integração de Dados (MEVID)**

A partir do momento que uma nova versão é gerada no repositório Subversion associado à PGDS, o MEVID é acionado para extrair o conteúdo semântico de possíveis documentos semânticos que foram adicionado, alterados ou removidos nessa versão. Para realizar essa tarefa, o MEVID utiliza como base as informações existentes no *log* do Subversion, a linguagem de anotação e a ontologia de domínio definida. Para cada documento existente no *log*, é necessário verificar se o mesmo é um documento semântico. Para identificar que um documento possui conteúdo semântico para ser extraído, foi definido um campo de usuário em todos os documentos semânticos com a seguinte chave e valor: “SemanticDocument”, “true”. Caso seja identificada a presença desse campo e seu valor, é iniciado o processo de extração e versionamento de dados. Dessa maneira, para cada adição ou alteração de documento semântico no Repositório de Documento Semânticos (RDS), um novo conteúdo semântico é extraído e armazenado na plataforma, permitindo que sejam realizados comparativos entre os dados de cada versão do documento. Além disso, ao final do processamento, é gerado um modelo semântico global, que representa a união dos conteúdos semânticos de todos os documentos existentes no repositório da plataforma, permitindo, assim, que se tenha uma visão global das informações

armazenadas. A Figura 2.3 apresenta o processo de geração de versão e execução do MEVID.

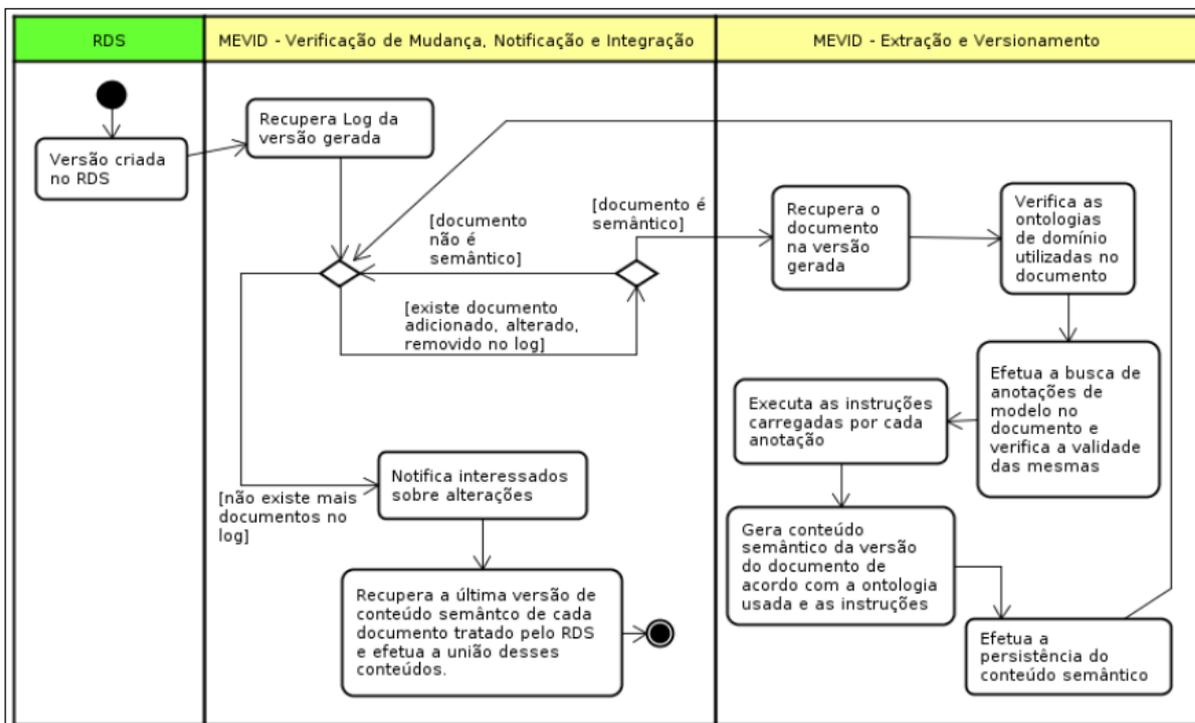


Figura 2.3 – Processo de geração de versão e execução do MEVID (ARANTES, 2010)

### Módulo de Busca e Rastreabilidade (MBR)

Esse módulo provê uma interface web de serviços de acesso à plataforma para que usuários possam realizar as seguintes consultas:

- Busca e rastreabilidade de dados em um repositório: possibilidade de se executar consultas SPARQL em um repositório, exibindo o resultado da consulta.
- Visualização da evolução de um indivíduo referenciado no repositório de dados: Dado a URI de um indivíduo, é exibido o estado do indivíduo em cada versão de um repositório selecionado.
- Visualização da evolução de um indivíduo referenciado em um documento específico de um repositório de dados: Dado a URI de um indivíduo, é exibido o estado do indivíduo em cada versão do documento de um repositório.

- Visualização do histórico de alterações de um determinado documento semântico: Dado um documento, exibe as alterações (adição, alteração e remoção de indivíduos) entre as versões existentes do documento em um repositório.
- Assinatura para notificação de alteração sobre um dado indivíduo no repositório: permite que, dada a lista de indivíduos alterados em uma nova versão do documento no repositório, usuários da plataforma que tiverem se cadastrado para receber informações sobre esse indivíduo sejam notificados através de um email sobre a alteração realizada.

Uma vez que a PGDS foi especializada neste trabalho para o domínio de Engenharia de Requisitos, a seguir uma breve revisão sobre o tema é apresentada.

## 2.4 ENGENHARIA DE REQUISITOS

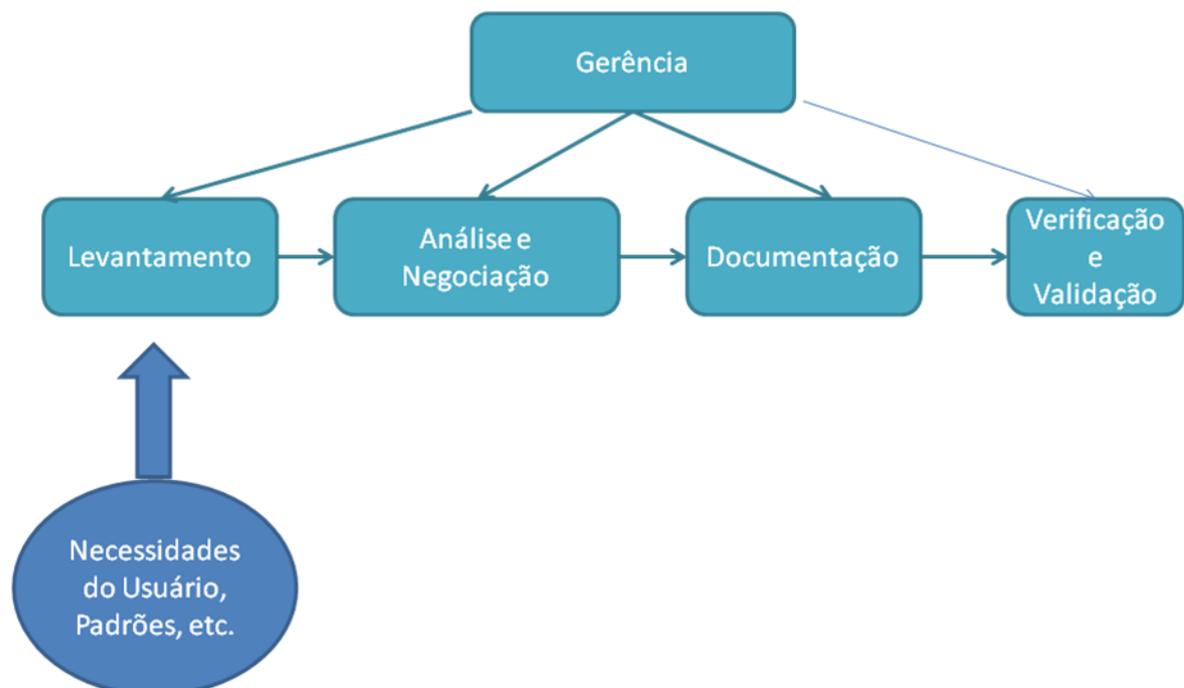
Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que ele deve oferecer e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de Engenharia de Requisitos (SOMMERVILLE, 2011). Assim sendo, a Engenharia de Requisitos (ER) é o ramo da Engenharia de Software que envolve as atividades relativas a desenvolver, documentar e dar manutenção ao conjunto de requisitos de um sistema. Ela pode ainda ser descrita como um processo, ou seja, um conjunto organizado de atividades, métodos, técnicas, práticas e transformações que ajudam a derivar, validar e manter os requisitos gerados. O processo de Engenharia de Requisitos envolve criatividade, interação de diferentes pessoas, conhecimento e experiência para transformar informações diversas (sobre padrões da organização, sobre leis, necessidades de usuários etc.) em modelos, especificações de sistemas e documentos que direcionem o desenvolvimento de software (KOTONYA; SOMMERVILLE, 1998).

Diversos benefícios podem ser obtidos com um processo de Engenharia de Requisitos de alta qualidade, dentre os quais Wiegers (2003) cita:

- Menos defeitos nos requisitos;
- Redução de retrabalho de desenvolvimento;
- Menos funcionalidades desnecessárias;

- Diminuição de custos;
- Desenvolvimento mais rápido;
- Menos problemas de comunicação;
- Redução de escopo de alterações;
- Caos de projeto reduzido;
- Estimativas de teste de sistema mais confiáveis;
- Alta satisfação dos clientes e membros da equipe.

Nesse trabalho, é tomado como base o processo de Engenharia de Requisitos definido por Kotonya e Sommerville (1998), que propõem um processo composto das seguintes atividades mostradas na Figura 2.4, que são: levantamento, análise e negociação, documentação, verificação e validação e gerência de requisitos.



**Figura 2.4 – Atividades do Processo de Engenharia de Requisitos**

A seguir, as atividades desse processo são descritas.

### 2.4.1 Levantamento

É a fase de descoberta dos requisitos através de consulta aos stakeholders, documentos de sistemas, conhecimento de domínio e estudos de mercado. Outros nomes utilizados para designar essa atividade são aquisição de requisitos ou descoberta de requisitos. Desenvolvedores de sistemas e engenheiros trabalham com clientes e usuários finais para entender acerca do problema a ser resolvido e descobrir os serviços do sistema, o desempenho que o mesmo deverá ter, restrições de hardware etc.. Isso não envolve apenas perguntar às pessoas o que elas querem, mas requer uma análise cuidadosa da organização, do domínio de aplicação e dos processos de negócio onde o sistema será utilizado (KOTONYA; SOMMERVILLE, 1998).

Para Aurum e Wohlin (2005), essa atividade envolve diversos estágios que são intensamente comunicativos, tais como: (i) entendimento do domínio da aplicação, que investiga e examina em detalhe o “mundo real” no qual ao sistema irá residir, bem como o relacionamento desse ambiente com o sistema (aspectos políticos, organizacionais e sociais); (ii) identificação das fontes de requisitos, desde as fontes mais óbvias como as necessidades dos *stakeholders*, até usuários e especialistas no domínio, documentação de outros sistemas, regras de negócio descritas em manuais, regulamentações etc; (iii) seleção das técnicas, abordagens e ferramentas a serem utilizadas, as quais, dependem do contexto específico que o projeto se encontra. Reuniões, questionários, investigação de documentos, prototipação, entrevistas (formais ou informais) e etnografia são exemplos de técnicas utilizadas na atividade de levantamento (SOMMERVILLE, 2011; PRESSMAN, 2006).

### 2.4.2 Análise e Negociação

A Análise de Requisitos e Negociação é uma atividade cara porque pessoas experientes despendem tempo lendo documentos cuidadosamente e pensando sobre a implicação das afirmações desses documentos, envolvendo o estudo cuidadoso dos produtos do levantamento de requisitos e raciocínio acerca das implicações dessas afirmações, com o objetivo de identificar os problemas e conflitos na definição dos requisitos (KOTONYA; SOMMERVILLE, 1998).

Diversos estágios dessa atividade são apresentados em (SOMMERVILLE, 2011; SWEBOK, 2004; KOTONYA; SOMMERVILLE, 1998), a saber:

- **Classificação de Requisitos:** Toma uma coleção de requisitos não estruturados, agrupando requisitos relacionados e organizando em grupos coerentes. Requisitos podem ser classificados em diversas dimensões, tais como funcionais e não funcionais, de acordo com sua volatilidade ou prioridade, e agrupados geralmente em uma arquitetura dividida em subsistemas.
- **Modelagem Conceitual:** Envolve o desenvolvimento de modelos do mundo real para solucionar o problema. É a chave para a análise de requisitos, tendo como propósito o entendimento do problema.
- **Negociação e Priorização:** Inevitavelmente, quando os vários *stakeholders* estão envolvidos, os requisitos entram em conflito. Normalmente, os *stakeholders* precisam se encontrar para resolver as diferenças e chegar a um acordo. Essa atividade está relacionada a encontrar e resolver os conflitos por meio da negociação de requisitos, visto que requisitos podem ser mutuamente incompatíveis. Por outro lado, muitos projetos de software têm mais requisitos candidatos do que podem ser implementados com as restrições de tempo e custo. A priorização auxilia a identificar os requisitos que agregam mais valor, distinguindo desse conjunto poucos que são realmente críticos dos muitos triviais existentes.

### 2.4.3 Documentação

Nessa atividade, os requisitos acordados são documentados com o devido nível de detalhe. Em geral, o documento de requisitos de software, muitas vezes chamado de Especificação de Requisitos, é uma declaração oficial do que os desenvolvedores do sistema devem implementar. Esse documento tem um conjunto diversificado de usuários, que vão desde a alta administração da organização que está pagando o sistema, até os engenheiros responsáveis pelo desenvolvimento. Essa diversidade é um indicativo de que o documento de requisitos precisa ter um compromisso com a comunicação, em geral utilizando diagramas gráficos, modelos e também a linguagem natural, de maneira a facilitar sua interpretação (SOMMERVILLE, 2011).

#### 2.4.4 Verificação e Validação

Nas atividades de Verificação e Validação de Requisitos examina-se a especificação para assegurar que: (i) todos os requisitos do sistema tenham sido declarados de modo não ambíguo, (ii) as inconsistências, omissões e erros tenham sido detectados e corrigidos, (iii) os requisitos estão em conformidade com as características de qualidade e (iv) realmente satisfazem a necessidades dos usuários (PRESSMAN, 2006) (KOTONYA; SOMMERVILLE, 1998) (WIEGERS, 2003).

#### 2.4.5 Gerência de Requisitos

Em paralelo a todas as atividades citadas anteriormente, a atividade de Gerência de Requisitos fica responsável por gerenciar as mudanças nos requisitos. Ela é definida como o conjunto de atividades que ajudam a equipe de projeto a identificar, controlar e rastrear requisitos e gerenciar mudanças de requisitos em qualquer época, à medida que o projeto prossegue.

Mudanças são inevitáveis, já que prioridades de negócio mudam, erros ou omissões em requisitos são descobertos e novos requisitos surgem. A gerência de requisitos pretende manter rastreável essas mudanças e garantir as mesmas serão realizadas no documento de requisitos de uma maneira controlada e atualizando de maneira consistente os relacionamentos entre requisitos e outros documentos do processo de desenvolvimento (KOTONYA; SOMMERVILLE, 1998).

Segundo Sommerville (2011), o planejamento é o primeiro estágio essencial no gerenciamento de requisitos e determina o nível de detalhamento requerido. Durante esse estágio, diversas decisões devem ser tomadas, tais como:

- Identificar os Requisitos unicamente para que eles possam ser comparados e referenciados por outros requisitos e usados em avaliações de rastreabilidade
- Definir um conjunto de atividades que avaliam o impacto e o custo de mudanças.
- Definir políticas de rastreabilidade que estabeleçam os relacionamentos entre cada requisitos e entre os requisitos e o projeto do sistema. A rastreabilidade de requisitos é possível, basicamente, se houver ligações entre requisitos, e entre requisitos e outros elementos do processo de software. Dessa forma, a identificação da

composição de requisitos, das dependências entre requisitos, de requisitos conflitantes, da origem dos requisitos e de seus interessados, além da identificação de em qual artefato (documento, módulo, diagrama, componente etc) produzido durante o desenvolvimento de software um requisito é tratado é de fundamental importância para que a rastreabilidade possa ser implantada.

Definir ferramentas de apoio é uma tarefa também importante, visto que o armazenamento, gerenciamento de mudanças e o gerenciamento da rastreabilidade são simplificados quando ferramentas ativas de apoio estão disponíveis (SOMMERVILLE, 2011).

Em segundo lugar, promover um processo formal de gerenciamento de mudanças auxilia no tratamento de forma consistente das alterações no documento de requisitos, além de realizá-las de maneira controlada. Nesse aspecto, Sommerville (2011) apresenta três estágios em um processo de gerenciamento de mudanças, a saber:

- Análise do problema e especificação de mudanças: Durante esse estágio, analisa-se o problema ou propostas de mudança a fim de se verificar sua validade.
- Análise de mudanças e custo: O efeito da mudança é avaliado por meio de informações de rastreabilidade, sendo que o custo de se fazer a mudança é estimado em termos de modificações no documento de requisitos e, se apropriado, no projeto e implementação do sistema.
- Implementação das mudanças: Executar modificações no documento de requisitos e, quando necessário, no projeto e na implementação.

## **2.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Documentos vêm desempenhando um papel chave na difusão do conhecimento humano, sendo ainda utilizados na maioria das organizações como forma de armazenamento do conhecimento. Entretanto, a maioria do conhecimento em documentos eletrônicos está disponível apenas para ser interpretado por humanos e não por sistemas computacionais. No contexto da Web Semântica, coloca-se que, para que o conteúdo disponível na web esteja acessível para sistemas (ou agentes inteligentes) buscarem, processarem e interpretarem, é necessário adicionar metadados ao conteúdo exposto em páginas web. Essa ideia é aplicável também no contexto de documentos gerados a partir de

ferramentas *desktop* como o Microsoft Word e Open Office Writer, transformando esses documentos em documentos semânticos.

Ontologias desempenham um papel fundamental na anotação semântica, tanto em páginas web quanto em documentos em formato *desktop*, visto que proveem um arcabouço para a conceituação e modelagem de conhecimento compartilhado, sendo que as mesmas já são utilizadas para promover esses metadados na web semântica.

Alguns trabalhos relacionados com infraestruturas de apoio à documentação semântica foram apresentados neste capítulo e tem relação direta com este trabalho. Essas infraestruturas apresentam funcionalidades úteis, como *plugins* para automatizar as anotações semânticas e interfaces com outros serviços, além de permitirem um conjunto grande de elementos de documentos a serem anotados, tais como seções, parágrafos, tabelas etc. Considerando esses aspectos e uma avaliação preliminar prática da Plataforma de Gerência de Documentos Semânticos (PGDS) no contexto de uma disciplina de Engenharia de Requisitos (MACHADO et al., 2011), foram identificadas oportunidades de melhorias para aprimorar a PGDS, conforme discutido no próximo capítulo.

Por outro lado, algumas das infraestruturas apresentadas foram utilizadas em domínios específicos, tais como domínios comercial, militar, governo e organizações (POPOV et al., 2006) estatísticas populacionais na Suécia e diagnósticos clínicos (ERIKSSON, 2007) e domínio acadêmico (CARR et al, 2004). Bruggeman et al. (2000) apresentam como a documentação desempenha também um papel fundamental durante o ciclo de vida dos projetos de engenharia, sugerindo a utilização da documentação semântica nesse contexto para prover buscas, compartilhamento e troca de informação de maneira facilitada e mais satisfatória.

Apesar de existirem ontologias de domínio específicas para a utilização nas infraestruturas citadas, as mesmas possuem apenas funcionalidades de caráter geral para trabalhar com ontologias que permitem realizar algum raciocínio, mas existe uma dificuldade em responder algumas questões específicas do domínio relacionadas com o documento, bem como dar apoio a tarefas específicas desse domínio. Muitas vezes, para prover um apoio mais efetivo ao domínio, é necessário explorar de modo específico a conceituação definida na ontologia. Em outras palavras, é preciso conhecer os elementos da ontologia (seus conceitos, relações e propriedades) e explorá-los em funcionalidades específicas de domínio, de modo a apoiar tarefas do domínio. Entretanto, isso não tem sido feito no contexto das infraestruturas citadas anteriormente.

Até onde conhecemos, não foram encontrados trabalhos que utilizassem uma infraestrutura de documentação semântica na Engenharia de Software, mais especificamente, ainda, no domínio de Engenharia de Requisitos (a menos da versão inicial da PGDS, que foi utilizada para o domínio da Gerência de Requisitos).

Assim, há uma motivação de estender a PGDS para o domínio da Engenharia de Requisitos. O Capítulo 4 apresenta a especialização desenvolvida neste trabalho. A seguir, o Capítulo 3 apresenta a avaliação preliminar realizada e as melhorias feitas na PGDS no contexto mais geral, a partir das oportunidades de melhoria identificadas nessa avaliação.