

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DEPARTAMENTO DE INFORMÁTICA
MESTRADO EM INFORMÁTICA**

LUCAS DE OLIVEIRA ARANTES

**DOCUMENTAÇÃO SEMÂNTICA NO APOIO
À INTEGRAÇÃO DE DADOS E
RASTREABILIDADE**

VITÓRIA

2010

LUCAS DE OLIVEIRA ARANTES

**DOCUMENTAÇÃO SEMÂNTICA NO APOIO
À INTEGRAÇÃO DE DADOS E
RASTREABILIDADE**

Dissertação apresentada ao
Mestrado 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

2010

LUCAS DE OLIVEIRA ARANTES

**DOCUMENTAÇÃO SEMÂNTICA NO APOIO À
INTEGRAÇÃO DE DADOS E RASTREABILIDADE**

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 27 de agosto de 2010.

COMISSÃO EXAMINADORA

Prof. Ricardo de Almeida Falbo, D. Sc.
Universidade Federal do Espírito Santo (UFES)
Orientador

Prof^a. Renata Silva Souza Guizzardi (PhD)
SRA-IRST-ITC, Trento, Itália

Prof. Rodrigo Quites Reis, D. Sc.
Universidade Federal do Pará (UFPA)

*A minha família, que, graças a Deus, teima em
crescer cada vez mais.*

*Ao mar, que manteve boa parte da sanidade
que disponho hoje.*

AGRADECIMENTOS

Aos meus pais, irmãs, irmãos, sobrinhos e sobrinhas que apoiaram a minha decisão de levar o trabalho adiante e compreenderam o constante “sumiço”.

Aos amigos do LabES (Laboratório de Engenharia de Software) e NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias), que não só entenderam minhas loucuras (ou fingiram de forma excelente) como trouxeram um valioso suporte ao trabalho.

Aos amigos da *McNose Vintage Tube Amplifiers*, que por alguns instantes retiraram meus pensamentos dos estudos acadêmicos.

Aos amigos da Projeta Sistemas de Informação, que apoiaram o trabalho e permitiram minha pausa para alcançar o término do mesmo.

Ao professor e amigo Ricardo, pela orientação constante e involuntária.

A Deus, por manter todas essas pessoas ao meu lado.

RESUMO

Documentos são mecanismos importantes para registro e disseminação de conhecimento e comunicação entre seres humanos. Organizações de software contam com esse mecanismo para a realização de projetos. Essas organizações, quando maduras, definem uma série de modelos de documento de software para apoiar o processo de documentação. Adicionalmente, as mesmas contam com processos e mecanismos para registrar o histórico evolutivo dos documentos envolvidos em um projeto.

A quantidade de documentos gerados durante o ciclo de vida de um projeto de software pode ser consideravelmente grande. Apesar dos sistemas de controle de versão facilitarem o controle da evolução dos documentos, os mesmos não são capazes de apoiar a integração e a visualização objetiva de suas informações. Somado a isso, documentos são normalmente tratados como um mero conjunto de caracteres com estilos de formatação voltados para renderização em ferramentas de edição de texto (em inglês, *document authoring tool*). Neste contexto, a adição de metadados baseados em ontologias de domínio apoia o desenvolvimento de documentos semânticos que podem ser utilizados para contornar esse cenário.

O processo de anotação semântica de documentos, quando feito de forma manual, é enfadonho e suscetível a erros. O uso de modelos de documento com anotações semânticas é uma opção viável para a automatização parcial desse processo. Assim, a união de mecanismos de integração contidos em sistemas de controle de versão e modelos de documento de software semanticamente anotados é um meio potencial de apoiar a integração de informações e, por conseguinte, a execução de projetos de software.

Esta dissertação trata da utilização de modelos de documento de software anotados semanticamente e mecanismos de controle de versão para registrar a evolução do conteúdo semântico existente em documentos gerados e para prover serviços de integração e visibilidade dos dados relativos aos documentos. Para tal, uma Plataforma para Gerenciamento de Documentos Semânticos é definida.

Palavras-chave: Documentação de Software, Documentação Semântica, Ontologias, Gerência de Configuração de Software, Rastreabilidade.

ABSTRACT

Documents are important mechanisms to register and disseminate knowledge and allow communication between human beings. Software Organizations use this mechanism to conduct projects. When these organizations reach a certain level of maturity they tend to use software document models in order to support the documentation process. Additionally, organizations count on processes and mechanisms to track the evolution of documents involved in a project.

The amount of documents produced during a software project lifecycle can be considerably large. Despite the fact that version control systems support document centralization they are not capable of support information integration and visibility. Summing up, documents are treated as a set of characters tied with formatting styles because their primary goal is to be well rendered on a document authoring tool. In this context, adding domain ontology based metadata can help developing semantic documents, which, in turn, can help overcoming this scenario.

Manual Semantic Annotation is notably dull and error prone. Combining document models semantic annotation is a viable option in order to achieve the annotation process automatization. Thus, uniting integration mechanisms provided by version control systems and software document models semantically annotated is a potential way of supporting information integration and, therefore, executing software projects.

This work treats the use of software document models annotated semantically and version control mechanisms to track and register the evolution of a semantic document content (instantiated from a semantic model) and the provision of data visibility and integration services. In this sense, a Platform for Managing Semantic Documents is defined.

Keywords: Software Documentation, Semantic Documentation, Ontologies, Software Configuration Management, Traceability.

LISTA DE ABREVIATURAS

DS	Documento Semântico
RD.....	Repositório de Dados
RDS	Repositório de Documentos Semânticos
PGDS.....	Plataforma de Gerenciamento de Documentos Semânticos
MAMD.....	Módulo de Anotação em Modelos de Documento
MEVID.....	Módulo de Extração, Versionamento e Integração de Dados
MBR.....	Módulo de Busca e Rastreabilidade

LISTA DE FIGURAS

Figura 2.1- A estrutura da <i>Web Semântica</i>	26
Figura 2.2 – (a)Edição de Metadados gerais do OpenOffice; (b) Código Gerado pelo OpenOffice (resumido).....	31
Figura 2.3 – (a)Edição de Metadados Customizáveis do OpenOffice; (b) Código Gerado pelo OpenOffice(resumido).....	31
Figura 2.4 - Relacionamento entre os Tipos de Ontologia segundo Guarino (1998).....	34
Figura 2.5 - Estrutura de um Documento Semântico e sua ligação com Ontologias - adaptado de Eriksson e Bang (2006) e Eriksson (2007b).....	36
Figura 2.6 - Anotação Semântica de documento com apoio de Ferramenta - adaptado de Eriksson e Bang (2006) e Eriksson (2007b).....	37
Figura 2.7 - Visão geral da arquitetura proposta por Eriksson e Bang (2006).....	40
Figura 2.8 - Conjunto de ontologias e seu relacionamento com documentos semânticos (Eriksson e Bang 2006).	41
Figura 2.9 - Visão geral da arquitetura proposta em (Kim et al., 2006).....	43
Figura 3.1 - Visão Geral da Plataforma de Gerenciamento de Documentos Semânticos	56
Figura 3.2 - (a) Modelo Semântico e (b) Documento semântico originado a partir do modelo....	62
Figura 3.3 - Grafo gerado a partir da instanciação do modelo semântico da Figura 2.....	63
Figura 3.4 - Visão geral do Módulo de Anotação em Modelos de Documento (MAMD)	72
Figura 3.5 - Processo de geração de versão e execução do MEVID	74
Figura 3.6 - Adição de Documento Semântico na Plataforma (MEVID)	75
Figura 3.7 - Alteração de Documento Semântico na Plataforma (MEVID)	75
Figura 3.8 - Remoção de um Documento Semântico na Plataforma (MEVID).....	76
Figura 3.9 - Diferença no conteúdo semântico de duas versões de um mesmo DS (MEVID)	77
Figura 3.10 - Integração de Dados (MEVID).....	79
Figura 3.11 – Cenário de busca usando o MBR.....	80
Figura 3.12 – Cenário de verificação da evolução de um indivíduo	82
Figura 3.13 - Evolução de um documento semântico.	83
Figura 3.14 – Visão da Arquitetura Geral da PGDS (pontuada por elemento de arquitetura).....	85
Figura 4.1 - Seção da Ontologia de Organização, adaptado de (BARCELLOS; FALBO, 2009). 94	
Figura 4.2 - Ontologia de Requisito, adaptado de (FALBO; NARDI, 2008).....	95
Figura 4.4 - Modelo de Projeto das Ontologias.....	97
Figura 4.5 - Implementação do Modelo de Projeto na ferramenta Protégé (visão de classes).....	98
Figura 4.6 - Implementação do Modelo de Projeto na ferramenta Protégé (visão de propriedades)	98
Figura 4.7 – Consulta SPARQL da Matriz de Rastreabilidade de Requisitos do Projeto	99
Figura 4.8 - Consulta SPARQL para identificação de não conformidades	101
Figura 4.9 - Interface Gráfica do MBR - Consulta com SPARQL.....	102
Figura 4.10 - Comparativo entre o Modelo de Documento de Requisitos original (à esquerda) e sua nova formatação (à direita) para comportar anotações semânticas.....	104
Figura 4.11 - Modelo de Documento de Requisitos com Anotações Semânticas	105
Figura 4.12 - Campos de usuário definidos no <i>OpenOffice Writer</i> para o Modelo de Documento de Requisitos.....	106
Figura 4.13 - Comparativo entre o Modelo de Descrição de Caso de Uso original (à esquerda) e sua nova formatação (à direita) para comportar anotações semânticas.....	108

Figura 4.14 - Modelo de Descrição de Caso de Uso da Organização X (modificado)	109
Figura 4.15 - Campos de usuário definidos no OpenOffice Writer para o Modelo de Descrição de Caso de Uso	110
Figura 4.16 – Resultado da Consulta de Não Conformidades na versão 2 (a) e rastreabilidade para os documentos relativos a um dos itens do resultado (b).	117
Figura 4.17 - Resultado da Consulta de Não Conformidades na versão 4 (a) e rastreabilidade para os documentos relativos a um dos itens do resultado (b).	118
Figura 4.18 – Corpo da Mensagem de Notificação de Mudança de Instância	120
Figura 4.19 - Formulário de pesquisa de rastreabilidade evolutiva de um indivíduo.	121
Figura 4.20 – Resultado da Pesquisa de rastreabilidade evolutiva do caso de uso Efetuar_Pagamento: (a) e (b) propriedades e valores da instância nas versões 8 e 9 respectivamente; (c) e (d) instâncias e propriedades que se relacionam com o indivíduo nas versões 8 e 9 respectivamente.	122
Figura 4.21 – Formulário de Pesquisa de Rastreabilidade Evolutiva de Documento Semântico	123
Figura 4.22 - Resultados para a de Rastreabilidade Evolutiva de Documento Semântico.....	123
Figura A.1 – Relação do Padrão MVC e os Módulos da Plataforma.....	137
Figura A.2 - Diagrama de dependência entre os módulos da PGDS.....	138
Figura A.3 - Diagrama de Classes do pacote <i>mamd::model::domain</i>	139
Figura A.4 - Diagrama de Classes de Domínio do pacote <i>mamd::model::domain::odt</i>	140
Figura A.5 - Diagrama de Classes de Domínio do pacote <i>mevid::model::domain</i>	141
Figura A.6 - Modelo de Classes do pacote <i>mevid::model::persistence</i>	143
Figura A.7 - Diagrama de Pacotes do pacote <i>mevid::model::service</i>	144
Figura A.8 - Diagrama de Classes do pacote <i>mevid::model::service::extraction</i>	145
Figura A.9 - Diagrama de Classes do pacote <i>mevid::model::service::vin</i>	146
Figura A.10 - Diagrama de Classes do pacote <i>mevid::controller</i>	147
Figura A.11 - Diagrama de Classes do pacote <i>mbr::model::service</i>	148
Figura A.12 - Diagrama de Classes do pacote <i>mbr::controller e mbr::view</i>	149

LISTA DE TABELAS

Tabela 2.1- Exemplo de tabela anotada com metadados - retirado de Sefton et al., (2009)	45
Tabela 2.2 - Comparativo entre os trabalhos estudados e características consideradas importantes	50
Tabela 3.1 - Relação das Tecnologias utilizadas.....	84
Tabela 3.1 - Relação das Tecnologias utilizadas (Continuação)	85
Tabela 3.2 - Relação de Pontos Arquiteturais e Tecnologias utilizadas.....	86
Tabela 4.1 - Exemplo de Resultados da Matriz de Rastreabilidade	99
Tabela 4.2 - Conjunto de Anotações Semânticas do Modelo de Documento de Requisitos.....	106
Tabela 4.3 - Anotações Semânticas do Modelo de Descrição de Caso de Uso.....	111
Tabela 4.4 - Resultados da Matriz de Rastreabilidade na versão 1 do Projeto.....	114
Tabela 4.5 - Resultados da Matriz de Rastreabilidade na versão 4 do Projeto.....	114
Tabela 4.5 - Resultados da Matriz de Rastreabilidade na versão 4 do Projeto (Continuação)....	115
Tabela 4.6 - Resultados da Matriz de Rastreabilidade na versão 9 do Projeto.....	116

SUMÁRIO

Capítulo 1 - Introdução	14
1.1 Motivação	14
1.2 Objetivos	17
1.3 Metodologia e Histórico do Trabalho	19
1.4 Organização do Trabalho	21
 Capítulo 2 - Documentação Semântica.....	22
2.1 - Documentação de Software	22
2.2 – Web Semântica	25
2.3 - Anotação Semântica.....	29
2.4 - Ontologias	33
2.5 - Documentação Semântica	35
2.6 - Semântica em Sistemas de Gerência de Configuração de Software.....	47
2.7 - Considerações Finais do Capítulo.....	48
 Capítulo 3 - Uma Plataforma para Gerenciamento de Documentos Semânticos.....	52
3.1 - Introdução	52
3.2 - Visão Geral da Arquitetura	55
3.3 - Anotação Semântica em Modelos de Documento	61
3.4 - Linguagem de Anotações em Modelos de Documentos.....	64
3.5 - Funcionamento da Plataforma de Gerenciamento de Documentos Semânticos.....	70
3.5.1 - Módulo de Anotação em Modelos de Documento (MAMD)	71
3.5.2 - Módulo de Extração, Versionamento e Integração de Dados (MEVID)	72
3.5.3 - Módulo de Busca e Rastreabilidade (MBR)	80
3.6 – Projeto e Implementação da Plataforma de Gerenciamento de Documentos Semânticos	83
3.7 - Comparação com Trabalhos Relacionados	87
3.8 - Conclusões do Capítulo	89
 Capítulo 4 - Uso da Plataforma de Gerenciamento de Documentos Semânticos no Apoio à Gerência de Requisitos	90
4.1 - Introdução	90
4.2 - Cenário de Aplicação	91
4.3 - Ontologias Utilizadas.....	93
4.3.1 - Ontologia de Organizações de Software	94
4.3.2 - Ontologia de Requisitos de Software	95
4.3.3 - Projeto e Implementação das Ontologias	96
4.4 - Anotação Semântica em Modelos de Documento	102
4.4.1 - Anotação Semântica do Modelo de Documento de Requisitos	103
4.4.2 - Anotação Semântica do Modelo de Descrição de Casos de Uso	107
4.5 - Uso da Plataforma no apoio à Engenharia de Requisitos	112

4.5.1 - Matriz de Rastreabilidade de Requisitos do Projeto	113
4.5.2 - Identificação de Não Conformidades relacionadas a Requisitos	116
4.5.3 - Notificação de Mudança.....	119
4.5.4 - Rastreabilidade Evolutiva de Instância e de Documento Semântico	121
4.6 - Conclusões do Capítulo	124
Capítulo 5 - Conclusões e Perspectivas Futuras	125
5.1 – Conclusões	125
5.2 – Limitações e Perspectivas Futuras	127
Referências Bibliográficas	131
Apêndice A – Projeto da Plataforma de Gerenciamento de Documentos Semânticos	137
3.6.1 - Módulo de Anotação em Modelos de Documento (MAMD)	138
3.6.2 - Módulo de Extração, Versionamento e Integração de Dados (MEVID)	140
3.6.3 - Módulo de Busca e Rastreabilidade (MBR)	147
Apêndice B – Modelos de Documento do NEMO Anotados Semanticamente	150
Apêndice C – Evolução de uma Matriz de Rastreabilidade de Requisitos	153
Anexo A – Modelos de Documento do NEMO.....	162
Anexo B – Documentação Final	165

Capítulo 1

Introdução

O objetivo deste capítulo é apontar ao leitor a motivação que levou o trabalho a ser conduzido (seção 1.1), descrever os objetivos buscados com a realização do mesmo (seção 1.2), apresentar a forma com que o mesmo foi materializado (seção 1.3), considerando alterações efetuadas perante alguns obstáculos encontrados, e, finalmente, estruturar a organização da dissertação, provendo uma forma mais facilitada de conduzir a leitura (seção 1.4).

1.1 Motivação

Historicamente, documentos vêm sendo uma das ferramentas chave para registro e disseminação do conhecimento humano como um todo. Para várias áreas existentes (negócios, governo e meio científico, por exemplo), a documentação escrita ainda é o principal meio de descrição e comunicação do conhecimento humano (ERIKSSON, 2007). No contexto de projetos de engenharia, um ponto importante é a permanente busca e troca de informações entre os especialistas envolvidos para que as atividades sejam realizadas de forma adequada. Neste cenário, a documentação desempenha também um papel fundamental durante o ciclo de vida dos projetos (BRUGGEMANN et al., 2000), mantendo não só a comunicação, mas também o entendimento efetivo do projeto (FORWARD; LETHBRIDGE, 2002).

O surgimento da documentação eletrônica facilitou o trabalho de documentação. Neste contexto, as organizações desenvolvedoras de software fazem uso da mesma na realização de projetos, em um processo de documentação de software que provê uma série de benefícios, dentre eles: a redução de tempo e esforço despendidos na fase de desenvolvimento do software, maior facilidade de entendimento do software por usuários finais e apoio às atividades referentes à localização das informações acerca de projetos de software (ROCHA et al., 2001).

Embora a contribuição da documentação de software seja clara, a aplicação da mesma é comprovadamente difícil. Alguns dos obstáculos na execução dessa atividade são (LETHBRIDGE et al., 2003) (ROCHA et al., 2001): o processo é, normalmente, enxergado como burocrático, sendo fonte de potenciais atrasos em projetos; há dificuldade na geração de documentos precisos e consistentes; e há grande esforço na busca de informações referentes a projetos finalizados que fizeram uso de documentação. Esses problemas são, muitas vezes,

acarretados pela ausência de relacionamento entre os vários documentos, que normalmente se referem a uma pequena parcela do processo de software (ROCHA et al., 2001). Mais do que isso, as informações contidas em um projeto podem estar espalhadas em diferentes arquivos, dificultando a visão geral do mesmo. Por esse motivo, as informações embutidas em cada documento e as interrelações entre elas nem sempre são enxergadas pelos envolvidos que, adicionalmente, não têm acesso automatizado a essas informações em momento oportuno.

Um dos fatores pelos quais as informações não são facilmente alcançadas é a falta de semântica existente nos documentos gerados. Tradicionalmente, documentos de texto vêm sendo tratados como um conjunto de caracteres com uma certa formatação de texto para serem manipulados por ferramentas de documentação, mas com conteúdo compreensível apenas por pessoas. Para que os documentos disponham de semântica, é importante que exista uma forma de adicionar metadados aos mesmos. Essa ideia compartilha a visão da *web* semântica (BERNERS LEE et al., 2001). Seguindo essa visão na documentação, é possível definir anotações em documentos, originadas a partir de ontologias de domínio, com o intuito de enriquecer o conteúdo do documento, e permitir que haja um processamento semântico do mesmo, facilitando, assim, a integração de sistemas e agentes que possam usufruir do conteúdo semântico dos documentos.

O ato de anotar documentos semanticamente, quando feito de forma manual, é comprovadamente tedioso e suscetível a erros (ERIKSSON, 2007) (UREN et al., 2006). Assim, automatizar, pelo menos parcialmente, o processo de anotação semântica de documentos é fundamental. Uma das estratégias para automatização do processo de anotação é fazer uso de modelos de documento semanticamente anotados (TALLIS, 2003), de modo que, anotando modelos de documentos e os utilizando na geração de documentos, estes últimos já seriam produzidos com anotações semânticas. Essa estratégia é particularmente interessante para organizações de software.

As organizações de software que apresentam maturidade contam com uma série de modelos de documento (*templates*) especializados para certos tipos de documentos, tais como documento de requisitos, descrição de caso de uso, plano de riscos, etc. A produção de modelos semânticos (modelos de documentos anotados semanticamente) no contexto da Engenharia de Software pode facilitar, portanto, a integração das informações que residem em documentos gerados a partir desses modelos. Um cenário típico de aplicação é o apoio à Gerência de Conhecimento.

A Gerência de Conhecimento visa assegurar o fluxo e a captura do conhecimento no nível da instituição e promover o surgimento de novos conhecimentos (NATALI, 2003). Segundo Heijst et al. e Dieng et al. (*apud* NATALI, 2003), a sistematização da Gerência de Conhecimento pode se dar pela implantação de serviços que atuam em torno de uma memória organizacional que armazena os itens de conhecimento adquiridos em uma organização. Neste cenário o conteúdo semântico, extraído a partir de documentos semânticos gerados com o uso dos modelos semânticos, pode ser utilizado para extrair conhecimento relevante da memória organizacional, provendo condições para a implantação de serviços de disseminação do conhecimento. Um cenário no contexto de Engenharia de Software é a integração de sistemas relacionados com os papéis que os recursos humanos desempenham em projetos executados pela organização com a notificação de possíveis itens de conhecimento de interesse dessas pessoas. Por exemplo, um gerente de projetos poderia ser notificado automaticamente sobre modificações ocorridas nos projetos em que ele está trabalhando. Nesse contexto, alterações em requisitos descritos em documentos de especificação de requisitos e adição de novos riscos em um documento de plano de risco seriam informações passíveis de notificação.

Em organizações de software, a execução de projetos envolve a geração de uma quantidade considerável de documentos. Assim, centralizar a localização dos mesmos é importante, não só para organização dos projetos, mas também para facilitar o acesso aos documentos. Embora manter algum nível de centralização de documentos de software seja fundamental para a organização, manter um histórico da evolução de cada documento de software é uma tarefa crucial para o ciclo de vida de projetos, já que permite auditorias sobre a evolução do projeto. Neste contexto os sistemas de gerência de configuração de software desempenham um papel fundamental, com destaque para os sistemas de controle de versão que fazem esse serviço de forma transparente. Sistemas de controle de versão possuem um repositório de controle de versão, que mantém o histórico evolutivo dos documentos, além de ser o principal ponto de integração com outros sistemas, através de um maquinário conhecido como gancho (*hooks*), que permite a execução de *scripts* segundo alguma fase de acesso ao repositório. Entretanto, os sistemas de controle de versão convencionais tratam artefatos de software como arquivos em um sistema de arquivos convencional e, normalmente, não têm entendimento algum sobre os dados que cada elemento contido ali carrega (NGUYEN, 2006). Portanto, para sistemas de controle de versão convencionais, as alterações em documentos de software são vistas como

substituições de arquivos binários ou modificações textuais em arquivos de texto. Ao mesmo tempo, sob um foco semântico, as alterações em documentos de software espelham inúmeras situações relevantes, tais como a adição de um novo risco no projeto ou a mudança na probabilidade de ocorrência de um risco previamente analisado, aumento do número de requisitos funcionais, mudança na descrição de um caso de uso etc. Assim, a união de mecanismos de integração contidos em sistemas de controle de versão e modelos de documento de software semanticamente anotados é um meio potencial de apoiar a integração de informações e, por conseguinte, a execução de projetos de software.

1.2 Objetivos

Este trabalho tem como objetivo geral desenvolver uma plataforma de gerenciamento de documentos de software semânticos. De maneira sucinta, a plataforma visa apoiar a criação de modelos de documentos com anotações semânticas, com a finalidade de automatizar parcialmente o processo de anotação semântica de documentos. Adicionalmente a abordagem visa aliar a documentação semântica com sistemas de controle de versão, mantendo o histórico de alterações não só dos documentos semânticos gerados a partir dos modelos semânticos, mas principalmente do conteúdo semântico extraído de cada versão de documento encontrado na plataforma. Com esse aparato é possível trabalhar um outro objetivo: montar um serviço para notificação dos envolvidos em um projeto, a partir de uma alteração registrada na plataforma. Por fim, a abordagem visa integrar o conteúdo semântico extraído de vários documentos contidos no repositório de controle de versão para prover uma visibilidade geral da situação do repositório quanto ao seu conteúdo semântico. Assim, são objetivos específicos deste trabalho:

- **Definir uma forma de criar anotações semânticas em modelos de documentos:** para permitir que os dados contidos em um documento sejam passíveis de manipulação por máquinas, é fundamental que esse documento seja enriquecido com metadados que permitam a extração da informação. O processo de anotação semântica deve fazer uso de ontologias de domínio para padronizar o vocabulário utilizado nas anotações dos modelos de documentos e, por conseguinte dos documentos gerados a partir deles;
- **Definir uma estratégia para extração, versionamento e integração do conteúdo semântico de documentos semanticamente enriquecidos:** sistemas de controle de versão convencionais não são capazes de armazenar o histórico de modificações do

conteúdo semântico de um documento semântico. Para isso é necessário estender o conceito de versionamento oferecido pela maior parte das ferramentas de controle de versão atuais, provendo o versionamento do conteúdo semântico gerado por cada versão de cada documento semântico tratado pela ferramenta de controle de versão. O conteúdo semântico de um documento é o conjunto de fatos extraídos a partir das anotações feitas no mesmo. Para que seja possível manter o versionamento desse conteúdo, é fundamental que uma estratégia de extração do conteúdo semântico do documento seja elaborada segundo a ideia proposta no primeiro objetivo, ou seja, extrair o conteúdo semântico de documentos semânticos instanciados a partir de modelos semânticos de documento. Com o versionamento de conteúdo semântico materializado é possível efetuar a integração do conteúdo semântico dos diversos documentos armazenados no repositório de controle de versão.

- **Definir uma estratégia para notificação às partes interessadas de alterações ocorridas no conteúdo semântico de documentos:** alterações em documentos envolvem não só mudanças sintáticas, mas também semânticas. Sistemas de controle de versão convencionais não são especializados para capturar as diferenças semânticas entre duas versões de um mesmo documento. Ao mesmo tempo, permitem a adição de pontos de notificação, conhecidos como ganchos (*hooks*), ativados durante alguma fase do processo de controle dos repositórios de controle de versão. Em algumas situações é importante conhecer as modificações semânticas ocorridas em um *checkin*, por exemplo. Nesse contexto a notificação de alteração semântica tem grande valor, sinalizando a modificação de algum elemento do conteúdo semântico modificado.
- **Implementar e empregar a infraestrutura proposta no domínio de Engenharia de Software:** para que a abordagem seja avaliada, uma implementação relativa à infraestrutura proposta é fundamental. Essa implementação deve contemplar a arquitetura como um todo e características providas pela plataforma, a saber: extração, persistência e versionamento de dados acerca de um conjunto de documentos e um sistema que permita que os usuários sejam notificados das alterações ocorridas nesses documentos. Em especial, um cenário dentro do domínio de Engenharia de Requisitos de Software visa mostrar o apoio oferecido pela plataforma no contexto de Engenharia de Software.

1.3 Metodologia e Histórico do Trabalho

Este trabalho se iniciou com a revisão da literatura das seguintes áreas: Documentação de Software, *Web Semântica*, Ontologias e Documentação Semântica. Durante o estudo da literatura foi observada uma lacuna no que tange a sistemas de apoio à documentação semântica em Engenharia de Software. Alguns trabalhos relativos ao tópico Documentação Semântica, em especial, formaram uma base para a análise dessa lacuna, a saber os trabalhos de Embley (1998), Nguyen (2006), Eriksson (2007), Eriksson e Bang (2006), Kim et al. (2006), Sefton et al. (2009) e Nesic et al. (2008).

Alguns dos trabalhos estudados tinham como interesse comum o desenvolvimento de plataformas para gerenciamento de documentos semânticos (ERIKSSON 2007, ERIKSSON; BANG 2006, KIM et al., 2006, NESIC et al., 2008). Entretanto, havia uma lacuna no que tange à integração de sistemas de controle de versões em ambientes de desenvolvimento de software e documentos semânticos.

Com essas variáveis em mente foi elaborado um esboço inicial de uma plataforma para gerenciamento de documentos semânticos baseados em modelos de documento de software semanticamente anotados e sistemas de controle de versão de documentos. Para isso, foram estudadas algumas formas de se anotar documentos semanticamente com o objetivo de entender como alguns trabalhos tratavam a questão de anotações semânticas em *templates*. Trabalhos como (SEFTON et al., 2009) e (TALLIS, 2003) foram importantes nessa etapa. Além disso, foram escolhidos os instrumentos tecnológicos para materialização da plataforma, a saber: o formato de documento ODF (*Open Document Format*), formato largamente utilizado atualmente, além de ser um padrão internacional (ISO/IEC 26300) (ISO,2006); a ferramenta de edição de documentos de texto *OpenOffice Writer*, por ser uma das ferramentas de natureza livre mais utilizadas atualmente e por utilizar o formato ODF como formato padrão de arquivo; o sistema de controle de versões *Subversion*, pela robustez do mesmo e pelo ferramental existente implementado na linguagem Java que permite a manipulação desse sistema de forma facilitada; e, finalmente, a plataforma Java Enterprise, que conta com a linguagem Java e um aparato tecnológico (ferramentas de apoio ao desenvolvimento, por exemplo) já amadurecido.

No que se tange à anotação semântica de documentos foi encontrado um obstáculo. A ideia primária era prover uma forma de se anotar modelos de documento de maneira análoga à

que outros autores faziam para anotar documentos: fazendo o uso de uma tríade de ontologias, de documento, de anotação e de domínio (KIM et al., 2006) (ERIKSSON; BANG, 2006). Inicialmente foi estudada a ontologia de documentos definida em Nunes (2005), mas essa ontologia foca na estrutura de um documento, definindo elementos como título, seção e subseção. Para que seja possível adicionar anotações em um documento, é necessário que haja a definição dos seus elementos de conteúdo, tais como parágrafos, trechos de texto, tabelas, colunas e linhas de tabelas, imagens etc. Por esses motivos, essa abordagem foi desviada para a produção e uso de uma linguagem especializada para anotação em modelos de documento. Aproveitando o fato de que a especificação ODF define todos os elementos levantados para anotação usados nesta dissertação, a linguagem de anotação de modelos proposta foi definida de forma a fazer uso da própria especificação ODF, adicionado a relação de elementos da linguagem com ontologias de domínio. Dessa forma foi possível gerar conteúdo semântico, embora não fosse utilizada uma ontologia de documentos propriamente dita.

A implementação de um cenário para avaliação do uso da plataforma que materializa parte das funcionalidades propostas neste trabalho foi conduzida. Para tal, o primeiro passo foi a definição de um cenário dentro do domínio de Engenharia de Software, mais especificamente na subárea de Engenharia de Requisitos. Para tal, foram escolhidos um conjunto de modelos de documento de software e um conjunto de ontologias de domínio. No que se refere às ontologias de domínio, as seguintes ontologias foram usadas: a Ontologia de Requisitos de Software proposta em (FALBO; NARDI, 2008) e um fragmento de uma Ontologia de Casos de Uso, produzida e apresentada nesta dissertação. As ontologias foram utilizadas juntamente com a linguagem de anotação em modelos de documentos para a criação de modelos de documento semânticos. Estes, por sua vez, foram utilizados para a produção de documentos de software semânticos, os quais são efetivamente os insumos para plataforma desenvolvida.

Adicionalmente foi feito um estudo sobre linguagens de implementação de ontologias, com o foco nos atuais padrões definidos pelo W3C (*World Wide Web Consortium*). A linguagem escolhida para a implementação das ontologias de domínio usadas nesta dissertação foi a linguagem OWL (*Ontology Web Language*) (MCGUINESS; HARMELEN, 2010), por ser atualmente o padrão de implementação de ontologias na *web*, por ser expressiva o bastante para permitir uma implementação com nível de fidelidade à ontologia significativo e por existir um conjunto considerável de ferramentas implementadas na linguagem de programação escolhida

neste trabalho (Java) que manipulam elementos dessa linguagem. Dessa forma foi possível materializar as ontologias e usá-las nas anotações de modelos de documentos.

Em seguida foi dada continuidade na implementação da plataforma, materializando o primeiro cenário: extração de conteúdo semântico de documentos semanticamente anotados a partir de modelos semânticos. Esses esforços propiciaram os primeiros resultados práticos do trabalho e, a partir deles, foi publicado o artigo intitulado “*An Infrastructure for Managing Semantic Documents*” (ARANTES; FALBO, 2010) no evento *Joint 5th International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE) - International Workshop on Metamodels, Ontologies and Semantic Technologies (MOST)*.

Posteriormente, foi conduzida a implementação de outras características da plataforma, dentre elas o versionamento do conteúdo semântico, a integração dos conteúdos versionados, a notificação de alteração de conteúdo semântico e pesquisas semânticas sobre o repositório de controle de versões da plataforma, e seus resultados são também apresentados nesta dissertação.

1.4 Organização do Trabalho

Esta dissertação está organizada da seguinte forma: no Capítulo 2 – Documentação Semântica – é apresentada a revisão da literatura correlata com o trabalho aqui apresentado; no Capítulo 3 – Uma Plataforma para Gerenciamento de Documentos Semânticos – a concepção da plataforma para gerenciamento de documentos semânticos é apresentada, focando nos serviços e relações existentes entre as partes da arquitetura; o Capítulo 4 – Uso da Plataforma para Gerenciamento de Documentos Semânticos no Apoio à Gerência de Requisitos – descreve o cenário de uso da plataforma; finalmente, o Capítulo 5 – Conclusões e Perspectivas Futuras – apresenta as conclusões do trabalho, juntamente com propostas para continuação e aperfeiçoamentos futuros do mesmo.

Capítulo 2

Documentação Semântica

Este capítulo tem como objetivo principal apresentar a fundamentação teórica para o trabalho realizado. A seção 2.1 aborda a Documentação de Software. A seção 2.2 discute a *Web Semântica*, contextualizando alguns obstáculos existentes na mesma. A seção 2.3 apresenta o conceito de Anotação Semântica, peça chave tanto para a materialização da *Web Semântica* quanto para a Documentação Semântica. A seção 2.4 versa sobre Ontologias e o papel das mesmas. A seção 2.5 trata do foco principal do trabalho, a Documentação Semântica, discutindo alguns trabalhos relacionados. A seção 2.6 levanta uma breve discussão sobre semântica em sistemas de gerência de configuração de software. Finalmente, a seção 2.7 apresenta as considerações finais do capítulo.

2.1 - Documentação de Software

Para que haja colaboração apropriada dentro de um contexto de atividades de engenharia, um ponto importante é a permanente busca e troca de informações entre os especialistas envolvidos. A documentação visa manter a comunicação e o entendimento efetivo entre os envolvidos em um projeto (BRUGGEMANN et al., 2000), (FORWARD; LETHBRIDGE, 2002).

No contexto de um projeto de software, a existência da documentação é fundamental e pode influenciar o sucesso ou fracasso de um projeto. Para que a documentação de software seja conduzida de forma organizada, um processo de apoio ao ciclo de vida do software deve ser implementado. O Processo de Documentação de Software visa registrar a evolução do software, criando uma base que permita um melhor entendimento do mesmo por usuários e que facilite o desenvolvimento e manutenção futura por desenvolvedores. O Processo de Documentação conta com as seguintes fases (ROCHA et al., 2001):

1. Implementação do Processo: visa identificar os requisitos de documentação, identificar os tipos de documentos tratados e seus relacionamentos, estabelecer a organização dos documentos, elaborar um plano de produção e distribuição, elaborar um plano de definição da linguagem para os documentos e estudar diferentes tecnologias alternativas para a documentação.
2. Projeto e Desenvolvimento: visa projetar a estrutura interna dos documentos e seu

comportamento dinâmico, incorporar aspectos de hipermídia ao projeto, projetar a apresentação dos documentos e finalmente revisar o plano de produção e distribuição dos documentos.

3. Produção: visa preparar o ambiente de produção, produzir e revisar os documentos, conceber a identificação dos documentos e, finalmente, publicar os documentos.
4. Manutenção: visa efetuar a manutenção (alteração) em documentos, além de eliminar ou armazenar documentos.

Segundo Rocha et al. (2001), a implementação do processo de documentação acarreta inúmeros benefícios para a organização como um todo, dentre eles a redução de tempo e esforço despendidos na fase de desenvolvimento do software, um melhor entendimento do funcionamento do software pelos usuários finais e o apoio às atividades referentes à localização das informações acerca do projeto de software.

Durante o processo de documentação, algumas organizações de software definem e usam artefatos conhecidos como modelos de documento. Esses modelos de documento permitem a instanciação de documentos de software de forma padronizada, agilizando a busca das informações ali contidas. A norma ISO 12207 define alguns resultados da execução bem sucedida de um processo de gerência de documentos de software, dentre eles destacam-se os seguintes resultados (ISO, 2008, tradução nossa):

- “os padrões a serem aplicados para o desenvolvimento da documentação de software são identificados;
- [...] o conteúdo e o propósito de toda documentação são especificados, revisados e aprovados;
- a documentação é desenvolvida e disponibilizada em conformidade com os padrões identificados [...]”

Ou seja, o conteúdo de um documento produzido durante o ciclo de vida de um projeto de software deve ser especificado a priori e produzido e disponibilizado de acordo com esse modelo. Além disso, a atividade de Projeto e Desenvolvimento do processo de Gerência de Documentação de Software dessa mesma norma contempla, dentre outras, as seguintes tarefas:

- “Cada documento identificado deve ser projetado de acordo com os padrões aplicáveis de documentação para meio, formato, descrição de conteúdo, numeração de páginas, posicionamento de figuras e tabelas, marca de propriedade/segurança, empacotamento e outros itens de apresentação. [...]”
- Os documentos preparados devem ser revisado e editados para se adequarem ao formato, conteúdo técnico e estilo de apresentação de acordo com os padrões de documentação. [...]”

Nesse contexto, o uso de modelos de documento ajuda a alcançar esse resultado. No domínio de Engenharia de Requisitos, por exemplo, o padrão IEEE/ANSI 830-1998 (IEEE, 1998)

provê indicações de conteúdo para modelos de documento de especificação de requisitos. No domínio de Gerência de Configuração, a norma ISO 10007 (ISO, 2005), responsável por fornecer diretrizes sobre o uso de Gerência de Configuração em uma organização, conta também com um modelo de documentos para Planos de Gerência de Configuração.

Apesar de trazer benefícios claros, o processo de documentação de software é questionado por alguns desenvolvedores, pois é tratado, muitas vezes, como um processo burocrático que pode levar a atrasos significativos dentro do cronograma de um projeto. Além disso, manter documentos consistentes e precisos é uma tarefa árdua e custosa. Por esses motivos, alguns projetos acabam gerando a documentação durante a fase de implementação, após o seu término ou, no pior dos casos, até mesmo não gerando documentação de nenhuma espécie. Em especial esse último caso acarreta problemas significativos de manutenção e uso do software. O resultado de uma pesquisa realizada por Lethbridge et al. (2003) aponta que a maior parte da documentação de software de uma empresa não é atualizada de forma apropriada. Ou seja, as alterações realizadas em um dado momento do ciclo de vida de um projeto de software nem sempre são refletidas em todos os documentos afetados, mantendo as novas informações em pequenas porções dos artefatos do projeto que, nem sempre, são acessíveis a todos os envolvidos no projeto.

Um caso levantado pela pesquisa feita por Lethbridge et al. (2003) expõe que uma alteração tardia em um projeto de software normalmente afeta apenas documentos de especificação de teste. Uma especificação de teste é, normalmente, reflexo de um documento de requisitos. No caso analisado pela pesquisa as alterações efetuadas em documentos de teste não são apropriadamente refletidas nos documentos de requisitos. Um outro ponto analisado na mesma pesquisa aponta que a busca por informações contidas nos documentos de um projeto é muitas vezes necessária e, em especial, há um grande esforço na execução de tal tarefa. A integração dos dados entre os documentos é também um fator relevante, já que em um ambiente de desenvolvimento, projetos de software contam com vários documentos e esses podem ter interrelações essenciais. Em resumo, dados de extrema importância estão espalhados em uma série de documentos da organização, dificultando a compilação de conteúdo relevante para a tomada de decisões.

Existe, portanto, a necessidade de se documentar, de forma a permitir o relacionamento e a integração dos dados encontrados em vários documentos, além de garantir o acesso dessa

compilação de dados aos envolvidos/interessados. Nesse contexto, um possível ponto de partida para superar alguns desses obstáculos é o uso de recursos básicos da *web*, tais como a criação de relações de navegação entre páginas, permitindo, portanto, o uso de páginas *web* como documentos e criando interrelações entre os mesmos. Apesar da contribuição que esse tipo de recurso oferece, ele não é suficiente para suprir todas as necessidades levantadas para a melhoria efetiva no Processo de Documentação, pois o interrelacionamento de documentos simplesmente não contempla explicitamente a interrelação dos dados contidos nos mesmos. Assim, a combinação de dados existentes em diversos documentos interrelacionados não é tratada de maneira satisfatória.

Com o crescente uso da *web* para geração de documentação, uma das escolhas para superar tais obstáculos é o uso da *Web Semântica*.

2.2 – Web Semântica

Desde a sua concepção, as páginas encontradas na *World Wide Web* (*www*) são, em sua maioria, escritas em linguagens especializadas em formatação de texto e relacionamento entre páginas, como é o caso do formato HTML – *HyperText Markup Language* – largamente utilizado para criação das páginas encontradas na *web* atualmente. Páginas dessa natureza foram criadas para serem renderizadas corretamente em navegadores (*browsers*), permitindo, portanto, a leitura humana das mesmas, mas não sendo possível a interpretação conjunta de máquinas (BERNERS-LEE et al., 2001; FENSEL et al., 2003).

Atualmente, o crescimento contínuo de conteúdo na *web* começa a mostrar seus problemas. Uma simples busca por informações específicas pode se tornar uma tarefa complexa, pois a enorme quantidade de materiais relevantes levantados pelas máquinas de busca nem sempre contém o mesmo sentido buscado pelo indivíduo que efetua a busca (FENSEL et al., 2003; BREITMAN, 2005). Adicionalmente, acompanhar mudanças em páginas normalmente requer a leitura completa do conteúdo da página, mesmo quando a mudança da mesma é mínima. Outro problema recorrente da *web* atual é que as páginas que permitem soluções de leitura automatizada por máquina, com a finalidade de prover interoperabilidade entre diferentes sistemas, normalmente não seguem um padrão de escrita, fazendo com que cada solução de leitura de máquinas saiba como efetuar a leitura de cada página específica. Além disso, essas páginas não carregam a semântica de seu conteúdo, mantendo os dados adicionais para leitura de

máquina em arquivos separados, podendo ocasionar falta de sincronia entre a página e os arquivos que descrevem dados da mesma. Esses cenários são fontes do aumento dos esforços aplicados tanto na manutenção de páginas quanto na integração entre sistemas que dependem das mesmas (FENSEL et al. 2003).

Nesse contexto uma forma alternativa de se tratar o conteúdo localizado na *web* vem sendo aprimorado. A visão da *web* semântica, idealizada em (BERNERSLEE et al., 2001), objetiva enriquecer a *web* atual com informações que permitam tanto a leitura do conteúdo das páginas *web* por seres humanos quanto o acesso às informações por sistemas computacionais. Dessa forma, a *web* semântica eleva o processamento de dados da *web* para um outro nível, onde é possível a existência de sistemas que busquem, leiam e interpretem o conteúdo da *web* para efetuar tarefas por intermédio de agentes ou serviços *web* (BREITMAN, 2005). Adicionalmente, as informações extraídas podem ser utilizadas para derivação de novas informações (KANELLOPOULOS; KOTSIANTIS, 2007).

A visão da *web* semântica se dá pela construção de camadas dependentes, onde nas camadas de base é normalmente encontrado o conteúdo *web* sintático, enquanto as camadas mais altas encapsulam o conteúdo com informações e funcionalidades extras que permitem tratar a semântica. Por consequência, a *web* semântica não substitui a *web* sintática, sendo de fato uma extensão da mesma.

A Figura 2.1 apresenta uma visão geral da arquitetura de camadas da *web* semântica, na qual há as seguintes camadas:

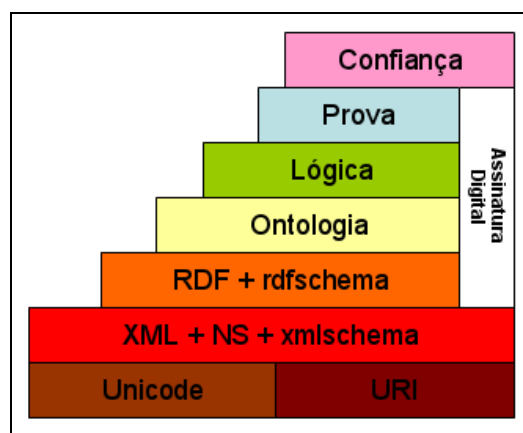


Figura 2.1- A estrutura da Web Semântica

- *Unicode* e *URI*: são basicamente elementos para codificação (*Unicode*) e de nomeação dos recursos da *web* (*URI – Universal Resource Identifier*). A *web* sintática é, em grande parte, dependente dessa base e seu conteúdo é primariamente escrito em páginas HTML com uso de scripts (p.ex., JavaScript) e formatação com estilos escritos em CSS (*Cascading Style Sheets*);
- XML + NS + *xmlschema*: comporta as tecnologias baseadas em XML (*eXtensible Markup Language*), que permitem usuários criarem vocabulários de forma estruturada. O formato XML permite o intercâmbio de documentos por intermédio de uma rede, facilitando a interoperabilidade entre sistemas de informação (BRIETMAN, 2005). Segundo Kanellopoulos e Kotsiantis (2007), o uso dessa tecnologia não é o bastante para prover semântica em documentos e, por esse motivo, aplicativos que fazem a leitura de documentos XML não podem interpretar seu conteúdo de maneira apropriada;
- RDF + *rdfschema*: concebido com base na linguagem XML, o *Resource Description Framework* (RDF) é focado na definição de uma sintaxe e de um modelo de dados padronizados (KANNELOPOULOS; KOTSIANTIS, 2007; BRIETMAN, 2005). RDF se apoia em uma estrutura comum para a representação dos metadados, a tripla, que comporta os seguintes elementos (KANNELOPOULOS; KOTSIANTIS, 2007; BRIETMAN, 2005): um sujeito, um predicado e um objeto. O RDF-S (*RDF Schema*) acrescentou inúmeros elementos ao modelo inicial do RDF, permitindo a definição de classes (sentido similar ao de classes em orientação a objetos), propriedades e hierarquia das mesmas. Dessa forma, desenvolvedores podem usar o RDF-S para elaborar vocabulários específicos que serão materializados na forma de dados RDF, além de especificar os tipos de objetos que poderão usufruir de certas propriedades definidas no modelo (PULIDO et al., 2006);
- Ontologia: RDF e RDF-S foram largamente criticados por não ser expressivos o bastante quando se trata de descrição formal de um domínio. Estruturas formais como negação, conjunção e disjunção não são contempladas em RDF-S (BRIETMAN, 2005), o que minimiza as possibilidades de geração e extração de informação por parte de mecanismos de raciocínio. Essa camada tenta justamente suprir essa necessidade. Dentre as linguagens encontradas nessa camada temos as linguagens: DAML (*DARPA Agent Modeling*

Language), elaborada com base na lógica descritiva e implementada a partir de RDF e RDF-S (FENSEL et al., 2003), e OWL (*Ontology Web Language*) (MCGUINESS; HARMELEN, 2010), atual padrão de implementação de ontologias na *web* (HORROCKS et al. 2003), também construída usando RDF-S e RDF. A maior diferença existente entre as características de RDF-S e OWL é que OWL permite a definição de restrições nas propriedades definidas em uma classe encontrada no modelo (HORROCKS et al., 2003). Portanto, é possível, por exemplo, manter uma restrição em uma classe chamada “PessoaBrasileira”, dizendo que qualquer indivíduo da classe “Pessoa” que possui o valor “Brasil” na propriedade “nacionalidade” é um indivíduo da classe “PessoaBrasileira”. Em sua concepção OWL foi elaborada de forma a manter a compatibilidade máxima com RDF-S e, por esse motivo, foi dividida em três categorias de linguagem:

- OWL *Full*: direcionada para usuários que desejam o máximo de expressividade e de liberdade sintática, mantendo a compatibilidade com RDF. Em contrapartida, não é computacionalmente eficiente e não garante que toda pesquisa em torno de um grafo terá uma resposta;
 - OWL DL: cunhada com base na Lógica Descritiva (DL – *Description Logics*), permite expressividade considerável com completude computacional (todas as conclusões são computáveis) e decidibilidade (qualquer computação tem tempo finito de execução). A expressividade é restrita no sentido de que permite apenas um subconjunto de construções permitidas em OWL *Full*;
 - OWL *Lite*: é o menor subconjunto da linguagem OWL, suportando basicamente a criação de hierarquias de classes e uso de restrições.
- Lógica: contempla as linguagens e tecnologias referentes à escrita de regras aplicadas no sentido de materializar axiomas ontológicos da camada abaixo, permitindo inferências lógicas que fazem parte da construção de ontologias (KANNELOPOULOS; KOTSIANTIS, 2007);
 - Prova: Para que a visão da *web* semântica seja alcançada e a interação entre agentes possa ser bem-sucedida, o resultado da interação esperado por um cliente deve ser acompanhado de um conjunto de justificativas (prova) compreensíveis pelo cliente. Nesse contexto a

camada de Prova provê ferramental para formalização das justificativas usadas para alcançar um resultado de interação entre agentes (SILVA et al., 2006);

- **Confiança:** discute a confiabilidade automatizada sobre os resultados gerados a partir da interação entre agentes. A *web* semântica demanda confiabilidade, já que a interação entre os agentes é feita sem ajuda humana e, em alguns casos, as informações levantadas para o atendimento de uma requisição podem ser contraditórias (ARTZ; GIL, 2007).

No contexto de documentação em geral, várias ferramentas de uso livre foram produzidas com o intuito de permitir tanto a produção colaborativa de documentos quanto o armazenamento central dos mesmos (Wikipedia¹, Scribd², Google Docs³). A visão da *web* semântica pode ser utilizada com a finalidade de apoiar esse gerenciamento (produção, armazenamento e manutenção), permitindo a criação de documentos com riqueza semântica. Para que esse cenário se concretize é importante que o enriquecimento semântico de um documento seja dado de forma facilitada ou transparente (UREN et al., 2006). O ato de enriquecer um documento semanticamente é conhecido como anotação semântica, tema discutido a seguir.

2.3 - Anotação Semântica

A discussão principal da *web* semântica gira em torno da representação formal do conteúdo existente na *web*. Com a materialização dessas representações, mecanismos que utilizam a *web* como fontes de dados podem compreender e processar o conteúdo *web*, permitindo a interação entre diferentes mecanismos para a realização de tarefas. Para alcançar esse cenário, a existência de uma estrutura que capture o sentido em torno do conteúdo é fundamental. Berners-Lee et al. (2001) enfatizam a necessidade de se “anotar” o conteúdo sintático da *web* com metadados. Um metadado é um dado que carrega uma informação estruturada necessária para o entendimento do significado de um outro dado contido em algum tipo de mídia (CAPLAN, 2003). A adição de metadados, portanto, enriquece o conteúdo das páginas semanticamente. Sánchez-Fernández e Fernández-García (2005) discutem o significado

¹ Wikipedia, a enciclopédia livre, http://pt.wikipedia.org/wiki/P%C3%A1gina_principal, visitado em 15/16/2010

² Scribd, <http://www.scribd.com>, visitado em 15/16/2010

³ Google Docs, <http://docs.google.com>, visitado em 15/16/2010

dos termos “metadado semântico” e “anotação semântica”. Segundo eles, o primeiro se refere à instância de metadado referente ao conteúdo de uma página *web*, enquanto o segundo se refere ao ato de anotar documentos com metadados. Segundo KIRYAKOV et al., (2003), anotação semântica é o ato de geração e uso de metadados estruturados específicos com a finalidade de permitir novos métodos de acesso à informação e estender métodos existentes. Kannelopoulos e Kostiantis (2007) tratam o termo em um contexto puramente *web*, descrevendo-o como “o processo de ‘etiquetar’ recursos encontrados na *web* com a finalidade de enriquecê-los com semântica”.

Por ter sido cunhado em meio à visão da *web* semântica, normalmente o termo 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 aplicados a documentos gerados a partir de ferramentas *desktop*, como o Microsoft Word ou Open Office Writer, ou até mesmo imagens. Assim, no contexto deste trabalho, é adotada uma definição de anotação semântica mais livre de contexto, a saber: Anotação semântica, enquanto processo, é o ato de adicionar metadados baseados em ontologias de domínio em um tipo de mídia de forma geral, sendo possível criá-los em torno da mídia como um todo ou em um trecho do conteúdo sintático do mesmo, na tentativa de adicionar sentido para a parte anotada. Anotação Semântica, enquanto objeto, é o próprio metadado baseado em uma ontologia de domínio que está relacionado a alguma instância de mídia anotável.

Para criar uma anotação semântica em sua forma mais básica, um usuário deve editar o conteúdo do documento, modificando sua estrutura, de forma que a anotação não seja parte do conteúdo visível para leitura humana, mas ao mesmo tempo seja passível de processamento por máquinas. Já que o ato de editar manualmente a estrutura de um documento é uma tarefa que pode levar a erros graves, podendo até comprometer a renderização gráfica do documento, algumas ferramentas proveem interfaces gráficas para o gerenciamento de metadados. O processador de texto da ferramenta *OpenOffice*, por exemplo, permite que você inclua metadados que registram título, assunto, palavras-chave e comentários adicionais ao texto. Adicionalmente a mesma ferramenta permite que se manipule um mapa de propriedades parcialmente customizáveis, no qual o usuário pode adicionar uma chave (um conjunto de caracteres definido pelo usuário), escolher um tipo para o valor da mesma (tipo texto, data, número e valor booleano)

e preencher um valor referente ao tipo de valor escolhido. Quando essas anotações são efetuadas, o editor registra as mesmas na estrutura do documento, mas sem deixá-las visíveis durante a edição do texto, como ilustram as figuras 2.2 e 2.3.

Esse tipo de anotação é diretamente referente ao documento e pode trazer inúmeros benefícios, dentre eles a possibilidade de se efetuar a busca de documentos relativos a um assunto ou uma palavra-chave específica.

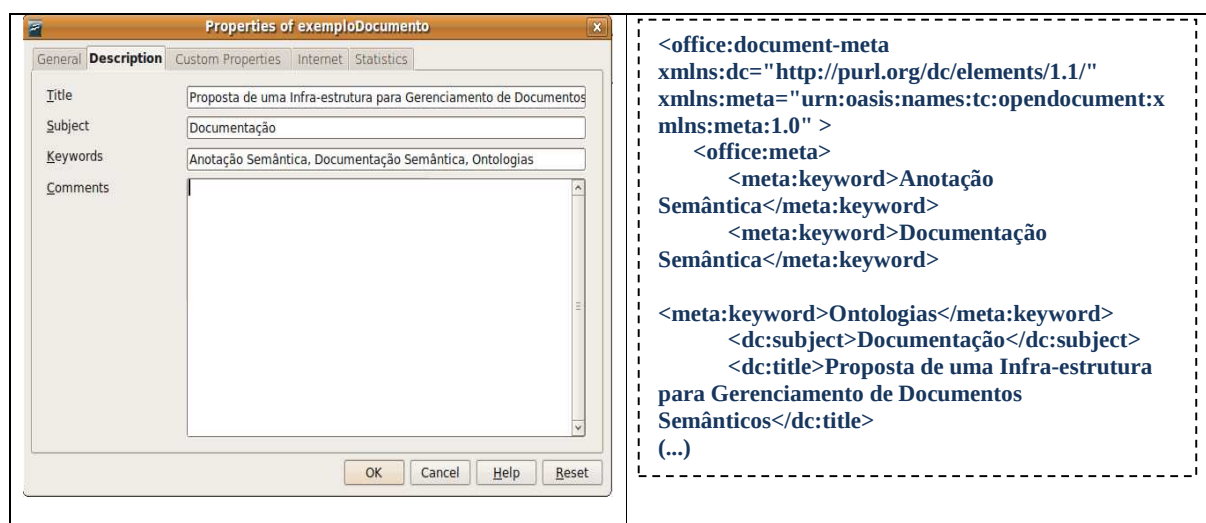


Figura 2.2 – (a)Edição de Metadados gerais do OpenOffice; (b) Código Gerado pelo OpenOffice (resumido)

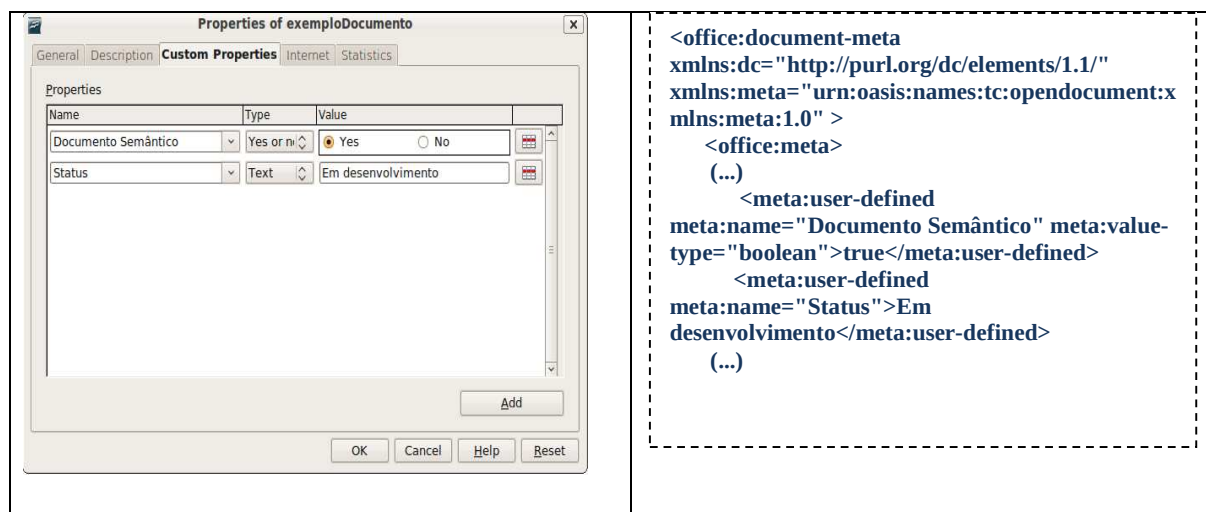


Figura 2.3 – (a)Edição de Metadados Customizáveis do OpenOffice; (b) Código Gerado pelo OpenOffice (resumido)

Embora esse tipo de artifício ajude a enriquecer documentos semanticamente, não basta apenas anotar o documento como um todo, pois existe uma gama considerável de informações

provenientes do conteúdo de um documento que podem ser extremamente úteis. Por esse motivo há também a necessidade de se anotar semanticamente o conteúdo de um documento (BRUGGEMANN et al., 2000). Em resumo, pode-se dizer que anotações semânticas podem ser feitas da seguinte forma: (i) anotação sobre o documento como um todo; (ii) anotação em torno de elementos do documento (trechos de texto, parágrafos, imagens, tabelas, células de tabela etc).

O uso de ferramentas para apoiar o processo de anotação semântica é fundamental, já que a anotação por via de edição da estrutura do documento é enfadonha e pode levar a erros. Nesse contexto várias ferramentas de anotação de documentos (sua maioria sendo voltada para anotação de páginas *web*) foram propostas. Uma análise do estado da arte das ferramentas dessa natureza foi feita por Uren et al. (2006). Os tipos de ferramentas de anotação são divididos da seguinte forma (ERIKSSON; BANG, 2006; UREN et al., 2006):

1. Ferramentas de Apoio à Anotação Manual: permitem que usuários editores de um documento possam criar manualmente anotações no mesmo, relacionando conceitos e indivíduos de uma ontologia a trechos de texto do documento. Apesar da grande utilidade e ganhos com a anotação manual, fica claro que fazê-la é uma tarefa tediosa. Dessa forma se faz necessária alguma forma de automatização desse processo;
2. Ferramentas de Anotação Automatizada: ferramentas deste tipo fazem uso de técnicas de processamento de linguagem natural com o intuito de minimizar a necessidade de anotações manuais. Duas categorias principais são contidas nesse conjunto de ferramentas: (a) ferramentas para sugestão de anotação durante a edição do texto e (b) ferramentas que anotam textos em larga-escala.

As ferramentas de anotação automatizada são notavelmente importantes, já que o processo de anotação consome um tempo considerável e pessoas responsáveis por criar anotações manualmente (com o uso de ferramentas de apoio à anotação manual) podem cometer erros durante o processo de anotação (ERIKSSON, 2007a; UREN et al., 2006).

Voltando a ideia inicial da *web* semântica, uma anotação semântica tem a finalidade inicial de prover entendimento para agentes realizarem tarefas automatizadas. Para que haja interoperação entre sistemas dessa natureza, o vocabulário utilizado tanto na realização das tarefas quanto nas possíveis conversações existentes entre esses sistemas deve ser padronizada e,

além disso, deve prover regras que impeçam que o conjunto de dados manipulado seja não lógico (BERNERS-LEE et al., 2001). As anotações semânticas oferecem, portanto, o ponto inicial para a captura de conhecimento contido em uma fonte e devem prover diretivas da localização do vocabulário usado para sua criação. Nesse contexto ontologias desempenham um papel fundamental. A próxima seção discute o que é uma ontologia e seu papel na *web* semântica.

2.4 - Ontologias

Segundo Guizzardi (2007), uma ontologia é uma especificação conceitual que descreve conhecimento sobre um domínio de forma independente de linguagem. Além disso, uma ontologia visa restringir as interpretações de um vocabulário para que os seus modelos lógicos se aproximem ao máximo do conjunto de estruturas da conceituação daquele domínio.

De acordo com a Filosofia Clássica, Ontologia é o estudo dos tipos de coisas que existem. Já no contexto de Inteligência Artificial e de algumas outras áreas da Ciência da Computação, como a Engenharia de Software, o termo é normalmente usado com duas finalidades (CHANDRASEKARAN, 1999):

- Representação de vocabulário comumente especializado para algum domínio;
- Referência a um corpo de conhecimento que descreve algum domínio.

Em resumo, as representações de vocabulário oferecidas pelas ontologias proveem um conjunto de termos que pode ser usado para descrever os acontecimentos que ocorrem dentro de um domínio, enquanto um corpo de conhecimento que usa esse vocabulário é uma coleção de fatos sobre o domínio (CHANDRASEKARAN, 1999).

Quatro tipos de ontologia são definidos na literatura, a saber (GUARINO, 1998):

- Ontologias Genéricas: descrevem conceitos gerais, como espaço, tempo, problema, objeto, evento, ação etc.
- Ontologias de Domínio: descrevem o vocabulário relacionado a um domínio genérico como, por exemplo, medicina, direito etc.
- Ontologias de Tarefa: descrevem o vocabulário relacionado a uma tarefa genérica, como, por exemplo, diagnose, venda etc.

- Ontologias de Aplicação: descrevem conceitos dependentes de um domínio e uma tarefa particulares, os quais são, frequentemente, especializações de ontologias relacionadas.

A Figura 2.4 mostra o relacionamento existente entre os tipos de ontologias apresentados. Ontologias genéricas são a base para a construção de ontologias de domínio e de tarefa, enquanto essas são combinadas de forma a apoiar a construção de ontologias de aplicação.

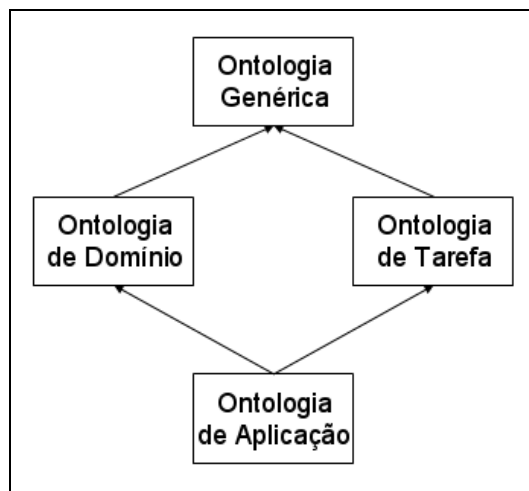


Figura 2.4 - Relacionamento entre os Tipos de Ontologia segundo Guarino (1998)

O uso de ontologias traz inúmeras vantagens, dentre elas a padronização de vocabulário e reúso do conhecimento acerca de um domínio para desenvolvimento de aplicações, dentre outras (CHANDRASEKARAN, 1999).

Ontologias desempenham um papel importante na *web* semântica. Segundo Berners Lee et al. (2001), ontologias podem ser usadas para aprimorar os mecanismos de busca existentes na *web*, resolvendo ambiguidades e imprecisões encontradas nos resultados de algumas pesquisas. Horrocks et al. (2003) apontam o papel fundamental das ontologias no cenário inicial elaborado na visão da *web* semântica (BERNERS-LEE et al., 2001), onde agentes buscam, manipulam e trocam informações acerca de um conjunto de entidades de um domínio. Sem o uso de ontologias para manter o vocabulário usado tanto para elaboração (anotação) das fontes de informação (páginas *web*, documentos e bancos de dados) quanto na manipulação e compartilhamento de dados, a realização desse cenário fica extremamente dificultada.

Em (FALBO et al., 2004) a importância de se prover semântica em aplicações de software é discutida. Embora o foco do trabalho seja voltado a ambientes de desenvolvimento de software,

a tendência atual nas empresas desenvolvedoras é o uso de ferramentas diversas, especializadas no apoio a atividades específicas da Engenharia de Software. Essas ferramentas, contudo, geralmente não trabalham de forma integrada, tornando a gerência dos dados gerados pelas mesmas uma tarefa árdua e suscetível a falhas.

No caso das ferramentas de documentação, é pequeno o apoio provido no sentido de enriquecer semanticamente os documentos gerados. As ferramentas são usadas para armazenar e disseminar o conhecimento, mas é necessário que os envolvidos leiam e interpretem os mesmos. Nesse contexto, se as ferramentas usadas para elaboração de documentos dispuserem de um ferramental para apoiar a elaboração de anotações semânticas, o processamento dos dados poderá ser feito também por computadores. Para que tal funcionalidade seja possível é fundamental que ontologias de domínio especifiquem o vocabulário utilizado nos documentos. 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. Assim, é possível obter uma visão mais abrangente sobre o conhecimento de uma organização. A próxima seção trata do tema Documentação Semântica.

2.5 - Documentação Semântica

Enquanto a documentação visa manter a comunicação e o entendimento efetivo entre envolvidos em um projeto (BRUGGEMANN et al., 2000; FORWARD; LETHBRIDGE, 2002), o uso de anotações semânticas visa permitir adição de metadados estruturados em torno de um tipo de mídia. O uso combinado dessas duas abordagens dá origem a uma outra, os Documentos Semânticos. A abordagem de Documentos Semânticos funciona de maneira análoga à abordagem da *Web Semântica* (ERIKSSON, 2007a), onde páginas *web* são anotadas com metadados semânticos. A diferença mais significativa é que o termo “documento semântico” é normalmente usado para referenciar arquivos renderizados por ferramentas de edição de texto *desktop*, diferentemente das páginas *web*, que necessitam de ferramentas de navegação para serem renderizadas.

Segundo Eriksson (2007a), documentos semânticos visam à combinação de documentos e ontologias, permitindo que usuários acessem os conhecimentos ali contidos de várias formas. Essa combinação entre documentos e ontologias se dá justamente pela adição de anotações semânticas nos documentos. A Figura 2.5 ilustra a estrutura de um documento semântico e seu relacionamento com ontologias. Basicamente, os metadados semânticos (retângulo em verde, na parte inferior esquerda da Figura) relacionam elementos do documento com ontologias (ERIKSSON; BANG 2006). Esses elementos podem ser: (i) elementos relativos às informações gerais do documento (retângulo laranja), geralmente descrevendo autores, palavras-chave, assunto e data de modificação do documento; e (ii) elementos do conteúdo do documento, representados pelo retângulo cinza (parágrafos, seções, tabelas, figuras etc).

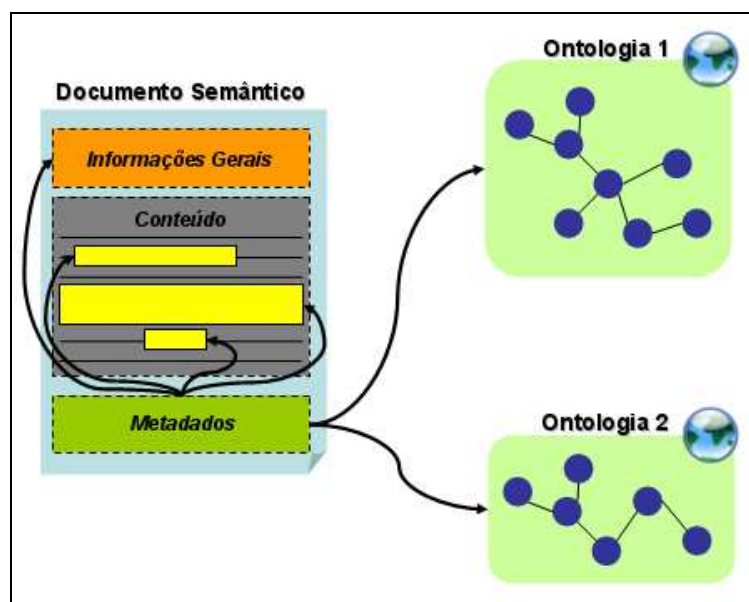


Figura 2.5 - Estrutura de um Documento Semântico e sua ligação com Ontologias - adaptado de Eriksson e Bang (2006) e Eriksson (2007b).

Adicionar anotações baseadas em ontologias em documentos eletrônicos viabiliza o oferecimento de novos serviços, tais como buscas baseadas em ontologias sobre grandes bases de dados de documentos (ERIKSSON, 2007a).

A criação de anotações semânticas deve ser assistida pelo uso de ferramentas, já que se trata de um processo custoso e suscetível a erros (ERIKSSON, 2007a; UREN et al., 2006). A Figura 2.6 ilustra o processo de anotação de um documento, gerando um documento semântico. Uma pessoa responsável pela adição de metadados faz uso de uma ferramenta de apoio à

anotação manual, com o intuito de enriquecer um documento semanticamente. A cada anotação efetuada, a ferramenta cria um metadado semântico, relacionando um dado do documento (referente a informações gerais ou conteúdo do documento) com um elemento da ontologia. Esses metadados são, normalmente, mantidos no próprio documento.

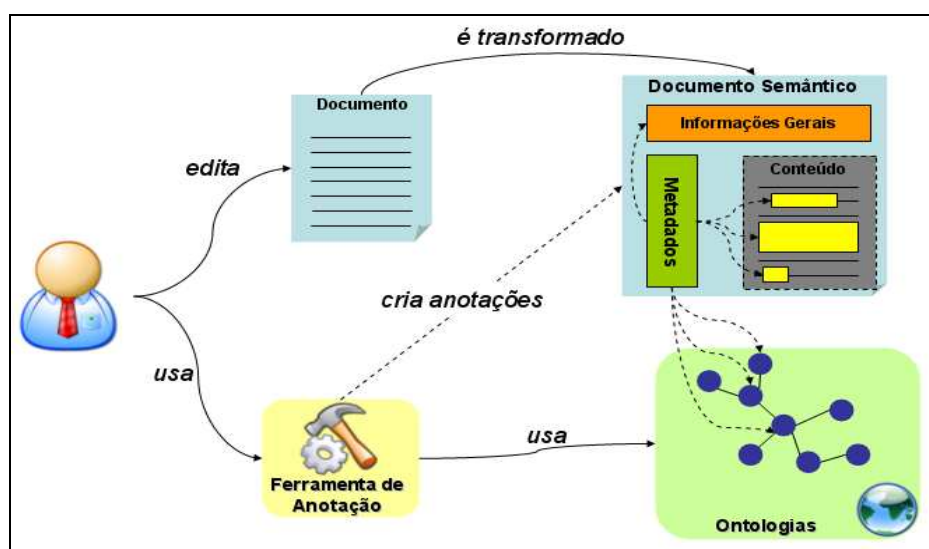


Figura 2.6 - Anotação Semântica de documento com apoio de Ferramenta - adaptado de Eriksson e Bang (2006) e Eriksson (2007b).

A adição de metadados semânticos pode ocasionar a criação, remoção ou atualização de alguns indivíduos, e relacionamentos entre eles, de uma ontologia do domínio sobre o qual versa o documento.

No âmbito desta dissertação o termo Documento Semântico se refere a documentos eletrônicos, disponíveis ou não através da *web*, em qualquer formato que permita a leitura humana e que disponha de metadados semânticos, possibilitando, dessa forma, processamento por máquina. Os metadados semânticos podem ser inseridos no próprio conteúdo do documento ou até mesmo no documento como um todo e esses são fundamentados em ontologias de domínio. Entretanto, os documentos semânticos considerados na proposta principal – uma plataforma de gerenciamento de documentos semânticos – são necessariamente gerados por ferramentas de edição de texto *desktop*.

A seguir, são apresentados alguns trabalhos relacionados à documentação semântica. A maior parte desses trabalhos está relacionada também com o tema *web* semântica, mostrando a estreita relação existente entre essas duas áreas de pesquisa. Os trabalhos apresentados a seguir

foram utilizados como base para a proposta principal desta dissertação: o desenvolvimento de uma plataforma de gerenciamento de documentos semânticos, apresentada no Capítulo 3.

SemanticWord

Tallis (2003) desenvolveu o *SemanticWord*, uma extensão do editor Microsoft Word para anotação semântica que provê funcionalidades para anotar trechos de texto de um documento. *SemanticWord* oferece uma interface para conteúdo da *web* semântica (definições de uma ontologia e descrições factuais) e traz esse conteúdo para dentro do *SemanticWord* com a finalidade de compor anotações. A ferramenta integra, ainda, o MSWord com AeroDAML, uma ferramenta de extração automática de informação. O AeroDAML analisa e anota o texto do documento em tempo de escrita, como um serviço paralelo (similar à checagem ortográfica automática).

SemanticWord permite que sejam elaborados *templates* de texto anotados, que podem ser reusados de forma a automatizar parcialmente a anotação de um documento. Neste contexto, *templates* são pequenos trechos de documento reutilizáveis previamente anotados semanticamente. A reutilização dos *templates* funciona da seguinte forma: o usuário que edita um documento pode adicionar (copiar e colar) um *template* semântico existente no conteúdo de seu documento.

O trabalho propõe duas formas de anotação: referência de instância e criação de triplas. A referência de instância associa um trecho de texto com uma instância existente. Uma tripla carrega a base de relação cunhada na *web* semântica: sujeito - predicado - objeto. O sujeito é sempre uma instância de uma classe. O predicado associa o sujeito com uma outra instância ou com um valor permitido para aquela propriedade.

No universo de *templates* semânticos permitidos pela ferramenta, existem duas formas de se produzir anotações: (i) criação de espaços reservados para instâncias e (ii) definição de repetição de grupos. A primeira permite que o texto do *template* reserve espaços pré-definidos onde uma referência para uma instância de uma ontologia será carregada quando o texto for preenchido. O usuário que define o *template* anota uma região de texto de forma que a mesma será preenchida posteriormente quando o *template* for utilizado. Já a repetição de grupos permite que um usuário delimite uma região do texto que é passível de anotações repetidas ou grupos de

anotações repetidas. Cada repetição possui seu próprio espaço reservado para instância e quando o grupo é replicado, todas triplas com referência para o mesmo são replicadas.

Uma Arquitetura para Gerenciamento de Documentos Semânticos

Eriksson e Bang (2006) propõem uma arquitetura para gerenciamento de documentos semânticos em torno de um repositório de documentos semânticos. Os autores discutem duas formas de se gerar documentos semânticos:

1. Geração automática de documentos semânticos: diz respeito à geração de documentos já anotados, fazendo uso de várias fontes de dados e ontologias para tal;
2. Uso de um analisador que investiga documentos previamente anotados e sugere anotações semelhantes em um documento não anotado.

Além de definir abordagens de criação de documentos semânticos, os autores mostram algumas funcionalidades que podem ser implementadas em torno de um repositório de documentos semânticos, dentre elas:

- “Clusterização” de Documentos: permite que haja navegação sobre os dados contidos no repositório semântico provenientes de documentos, oferecendo, dessa forma, uma visão gráfica dos documentos e dos seus respectivos dados;
- Verificação de Consistência: permite que sejam encontrados padrões e relacionamentos não desejáveis no repositório de dados, tais como relacionamentos contraditórios e inconsistência de dados;
- Raciocínio: permite que sejam gerados novos fatos a partir dos dados existentes na base de dados. Esses fatos podem ser utilizados em máquinas de resolução de problemas, sistemas de suporte à decisão e mecanismos de *workflow*.

A Figura 2.7 ilustra a arquitetura geral proposta em (ERIKSSON; BANG, 2006), exibindo as formas de criação de documentos semânticos, o repositório que comporta os mesmos, as ontologias de domínio que permeiam o universo semântico dos documentos e finalmente as funcionalidades que podem ser implementadas em torno do repositório semântico.

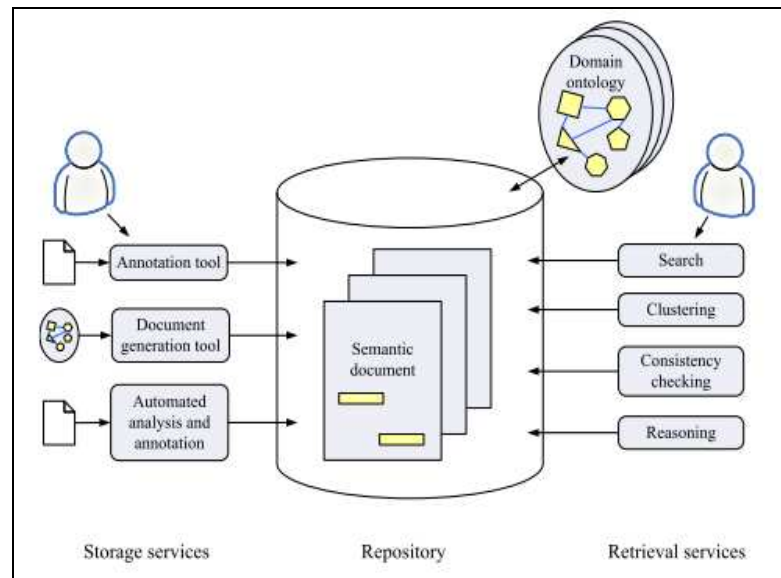


Figura 2.7 - Visão geral da arquitetura proposta por Eriksson e Bang (2006)

Segundo os autores, a criação de metadados em documentos não é suficiente para permitir o apoio à gerência dos documentos, sendo necessário o uso de ontologias para solidificar os conceitos e relacionamentos usados nos metadados e para manter relacionamentos entre os mesmos e as ontologias externas. Nesse contexto os autores propõem um conjunto de ontologias que permitem o estabelecimento de relações entre anotações semânticas, elementos encontrados em um documento e elementos de uma ontologia de domínio (conceitos, relacionamentos, indivíduos etc).

Em um primeiro nível os autores propõem uma ontologia de anotação semântica que liga elementos encontrados em um texto a uma ontologia de documento. A ontologia de documento descreve os conceitos relativos a documentos, tais como trechos de texto, seção, subseção, parágrafo, imagens, tabelas, diagramas etc, e faz uma ligação direta entre os seus elementos e os elementos de uma (ou mais de uma) ontologia de domínio. Uma ontologia de domínio, por sua vez, define o vocabulário comum acerca de um domínio específico sobre o qual versa um documento. A Figura 2.8 mostra as relações entre essas ontologias.

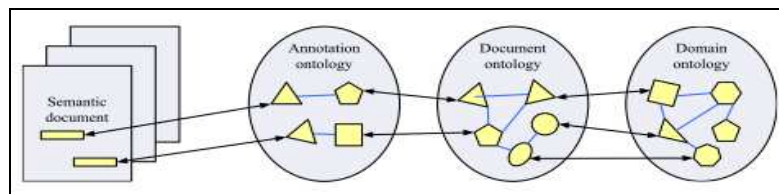


Figura 2.8 - Conjunto de ontologias e seu relacionamento com documentos semânticos (Eriksson e Bang 2006).

Os autores ainda citam que a abordagem pode ser modificada, permitindo, por exemplo, que a ontologia de anotação faça o relacionamento entre os elementos da ontologia de documento e da ontologia de domínio.

A evolução do trabalho de Eriksson e Bang (2006) resultou na construção de uma ferramenta voltada para criação de documentos semânticos – PDFTab (ERIKSSON, 2007a, 2007b). Em resumo a ferramenta proposta é uma extensão da ferramenta Protégé⁴, largamente utilizada para implementação de ontologias. PDFTab permite a importação de documentos em formato PDF⁵ (*Portable Document Format*) no próprio Protégé e a adição de anotações no documento importado. As anotações definidas no documento PDF carregam informações como: número da página em que a anotação se encontra, coordenadas textuais e texto demarcado. Adicionalmente, cada anotação possui um campo opcional que relaciona a mesma com algum conceito ou indivíduo de uma ontologia de domínio implementada em OWL. Nesse caso têm-se uma anotação semântica propriamente dita.

Uma das características de PDFTab é possibilitar a navegação de anotações semânticas textuais para os elementos da ontologia referenciados, como conceitos da ontologia de domínio referenciada e indivíduos provenientes da mesma (ERIKSSON, 2007b). A ferramenta PDFTab foi utilizada para anotar relatórios estatísticos na Suécia, providos pela empresa governamental *Statistics Sweden*⁶, que visa elaborar e fornecer publicamente tais relatórios na forma de documentos PDF. O cenário de uso foi melhorar a máquina de busca (fornecida pela mesma

⁴ The Protégé Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>, visitado em 12/07/2010.

⁵ Portable Document Format, <http://www.adobe.com/br/products/acrobat/adobe.pdf.html>, visitado em 12/07/2010.

⁶ Statistics Sweden, http://www.scb.se/default_2154.aspx, Visitado em 12/07/2010.

empresa) de documentos e informações, já que a mesma não considerava semântica nas buscas. Primeiramente os autores definiram ontologias de documento especializadas na estrutura dos relatórios estatísticos levantados. Em seguida, as instâncias dos relatórios tiveram alguns elementos de texto anotados (cabecinhos de tabelas e resumo de diagramas) e relacionados com conceitos de ontologias com o auxílio de PDFTab. Adicionalmente, uma ferramenta experimental para automatizar as anotações foi utilizada, agilizando o processo de anotação. Dessa forma foi possível adicionar um nível semântico às buscas efetuadas (ERIKSSON, 2007a).

Documentação Semântica apoiada por Tecnologias da Web Semântica

Em (KIM et al., 2006) foi desenvolvido um sistema colaborativo capaz de extrair, salvar e indexar dados de vários recursos da *web* (páginas, rss etc.) e recursos *desktop* (imagens, documentos etc.), com o intuito principal de permitir buscas semânticas, notavelmente mais poderosas que buscas textuais convencionais.

A infraestrutura criada funciona da seguinte forma: de posse de recursos *web* ou *desktop*, *wrappers* definidos para cada tipo de recurso são usados para extrair metadados baseados em ontologias de domínio. Após um processamento adequado, os dados são transformados em um grafo RDF e esse é devidamente persistido e indexado, permitindo que sejam executadas pesquisas mais significativas sobre o mesmo. Parte do aparato tecnológico conta com a plataforma XMP⁷ e o formato PDF (para documentos *desktop*). XMP (*eXtensible Metadata Platform*) permite a adição de metadados (escritos em RDF) sobre um arquivo no próprio arquivo. No caso do trabalho conduzido por Kim et. al (2006), os arquivos que contemplam metadados anotados por meio da XMP devem estar no formato PDF. A arquitetura proposta conta com os seguintes elementos, como mostra a Figura 2.9:

1. Recursos provenientes tanto de ambientes *desktop* quanto de ambientes *web*. Esses elementos são as fontes de dados do sistema;
2. *Wrappers* especializados para cada tipo de recurso (exemplo: *wrapper* voltados para páginas *web* em HTML ou para documentos PDF);
3. O sistema propriamente dito, o qual conta com os seguintes componentes:

⁷ Adobe XMP, <http://www.adobe.com/products/xmp/>, visitado em 12/07/2010.

- o Módulo Principal: responsável pela edição de ontologias de domínio, extração e geração de metadados e transformação de recursos XMP e RDF. Possui uma interface com o repositório de dados, permitindo que sejam elaboradas buscas refinadas acerca dos dados extraídos;
- o Repositório de dados: responsável por armazenar e indexar os dados extraídos pelo módulo principal.

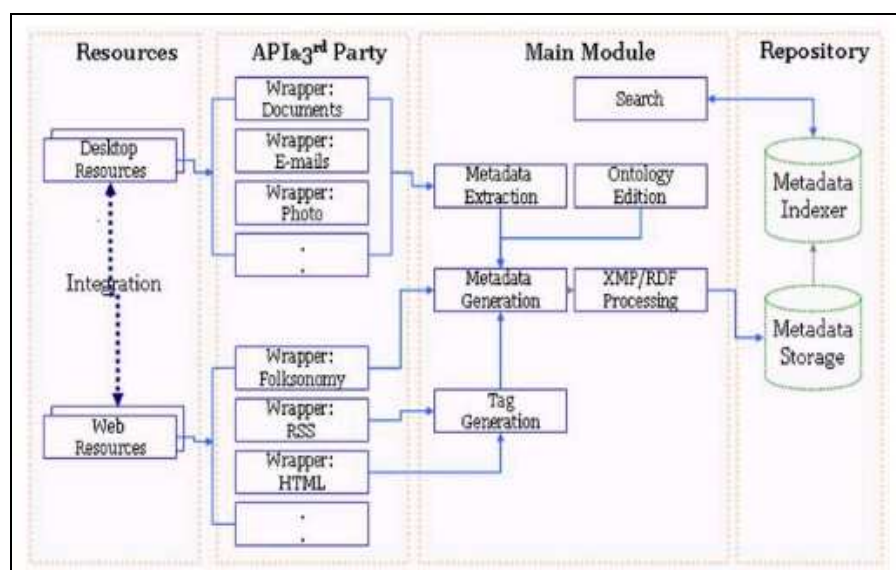


Figura 2.9 - Visão geral da arquitetura proposta em (Kim et al., 2006)

Os usuários podem anotar documentos PDF usando a ferramenta de anotação proposta pelos autores. Feitas as anotações, o módulo de extração de dados faz uso do *framework* Jena RDF para capturar os grafos RDF contidos nos documentos que, em seguida, são devidamente armazenados e indexados no repositório de dados.

Ambiente de Conteúdo Integrado

O trabalho realizado por Sefton et al. (2009) descreve algumas técnicas de anotação semântica usada no contexto do ambiente ICE (*Integrated Content Environment*⁸), que visa primariamente permitir que usuários possam criar páginas *web* ou outros documentos (apresentações, por exemplo) a partir de documentos de texto. As anotações semânticas

⁸ ICE: The integrated Content Environment, <http://ice.usq.edu.au/introduction/about.htm>, visitado em 13/07/2010.

produzidas nesse projeto são registradas em documentos por intermédio de ferramentas de edição de texto *desktop*, mais especificamente *Microsoft Word* e *OpenOffice*. Alguns dos requisitos para escolha da técnica e para elaboração das extensões nas ferramentas de apoio à anotação foram: (i) garantia de compatibilidade entre as duas principais ferramentas de edição de texto levantadas (Word e OpenOffice); (ii) aderência ao formato ODF (*Open Document Format*); e (iii) o trabalho deve ser agnóstico quanto à escolha de tecnologia de metadados, embora RDF tenha sido a tecnologia principal usada para anotações.

As técnicas proposta pelos autores se baseiam no uso de estilos de formatação de texto e tabelas de duas colunas nos documentos anotados. O uso dessas estruturas foi modificado, relacionando cada uma delas (instâncias de estilo de formatação e tabelas) com elementos RDF. Basicamente os autores usam essas estruturas para embutir metadados em torno do documento como um todo e, para alguns casos, para anotar o próprio conteúdo do documento. Essa escolha foi feita, pois ela permite a interoperabilidade de um documento pelas duas ferramentas de edição de texto levantadas. Um exemplo de tabela anotada pode ser observado na Tabela 2.1. Nesse exemplo, a intenção é capturar os seguintes dados relativos ao documento como um todo: título do documento e nome, filiação e email do autor.

No cabeçalho das tabelas anotadas os autores fazem uso de estilos de formatação com os nomes de estilo representados por *strings* contidas entre ‘{ }’s (no exemplo temos *{meta-document-information}*). Esses estilos são o ponto inicial para um procedimento interpretador que, em resumo, faz a leitura dos nomes de metadados, descritos na coluna à esquerda, e de seus respectivos valores, contidos na coluna à direita. Com essa técnica os autores conseguem anotar e extrair os metadados relativos ao documento como um todo de forma automatizada, mantendo a interoperabilidade entre as ferramentas levantadas.

Tabela 2.1- Exemplo de tabela anotada com metadados - retirado de Sefton et al., (2009)

<i>Metadata {meta-document-information}</i>	
Title	Metadata in ICE documentos
Author Name	Ian Barnes
Author Affiliation	ANU
Author Email	Ian.Barnes@anu.edu.au

Outra técnica usada consiste em apoiar a estruturação física do documento com estilos de formatação pré-definidos pelos autores. Alguns estilos definidos são: título do texto, cabeçalhos de texto (*heading*), citação (*quote*), parágrafo e listagem de itens. Essa técnica foi elaborada para manter a compatibilidade na transformação entre os formatos levantados (de ODF para HTML ou PDF), uma vez que um documento é publicado no ambiente ICE. Para facilitar a edição do texto usando tais estilos de formatação, *plug-ins*⁹ para as ferramentas *OpenOffice* e *Microsoft Word* oferecem uma barra de ferramentas com os estilos propostos pelos autores.

Sefton et al. (2009) também elaboraram uma técnica para estabelecer relacionamentos entre trechos de texto ou imagens e elementos externos ao documento. Nesse caso a anotação conecta o elemento do documento com o elemento externo pelo uso do endereço *web* do elemento externo (*links*). A visualização durante a edição do documento mantém o trecho de texto ou imagem anotada na forma estática. Uma vez que o documento é publicado, durante a visualização do mesmo via *browser*, os elementos externos referenciados no documento são embutidos na página de apresentação. Isso permite um aspecto dinâmico na apresentação do mesmo. Animações, por exemplo, são mantidas na forma estática durante a edição do documento e na forma dinâmica durante a sua visualização pelo ambiente ICE após sua publicação.

Em resumo, as técnicas propostas contam com o uso de estilos de formatação que são usados para manter um relacionamento com a entrada de dados feita pelo usuário. Com a

⁹ Toolbar and Templates, http://ice.usq.edu.au/instructions/templates/toolbars_and_templates.htm, visitado em 13/07/10.

publicação do documento anotado, o ambiente ICE é capaz de exibir o mesmo documento em diferentes formatos.

Sistema de Gerenciamento de Documentos Semânticos

O contexto do sistema de gerenciamento de documentos semânticos proposto por Nesic et al. (2008) é a Gerência de Objetos de Aprendizado (*Learning Objects*), tendo como objetivo principal apoiar, de forma colaborativa, a realização de tarefas na área pedagógica. Objetos de Aprendizado são entidades digitais reusáveis e auto contidas, voltadas para aprendizado humano, usadas para elaboração de aulas e treinamentos.

A proposta principal dos autores é prover um sistema de gerenciamento de documentos semânticos (SGDS) voltados para o registro de objetos de aprendizado. Os autores elaboraram um modelo diferenciado para edição de documentos semânticos: enquanto a maior parte dos trabalhos de edição de documentos semânticos persiste tanto as ontologias quanto os seus relacionamentos na própria estrutura do documento, os autores formalizaram uma ontologia (ACCM – *Abstract Compound Content Model core ontology*) que comporta as estruturas encontradas em objetos de representação de conteúdo (elementos de um documento, por exemplo) e permite o relacionamento com ontologias externas. Baseando-se no modelo abstrato de conteúdo composto (ACCM), numa ferramenta de anotação e no sistema SGDS, cada anotação feita no documento, com apoio da ferramenta, cria no próprio SGDS indivíduos e relacionamentos da ontologia e cada identificador desses itens (indivíduos e relacionamentos) é copiado no próprio documento. Dessa forma o relacionamento entre a anotação e os itens da ontologia é mantido.

O modelo ACCM, elaborado em (NESIC et al., 2007), define tanto unidades de conteúdo, elementos encontrados em documentos eletrônicos (seções, parágrafos, slides, tabelas, gráficos, vídeo etc.) em diferentes níveis de granularidade, quanto a forma pela qual eles podem se relacionar. O intuito final do modelo é abstrair a representação de conteúdo, provendo uma meta solução que, além de ser independente de plataforma, facilite a integração com ontologias de domínio.

Uma das utilidades da solução proposta é permitir que usuários desse sistema possam compartilhar e reusar objetos de aprendizado registrados no SGDS. O sistema de gerenciamento permite, ainda, que os usuários do mesmo efetuem buscas baseadas em ontologias, que facilitam

a elaboração de treinamentos. Dessa forma a disseminação de conhecimento e o reúso de objetos de aprendizado são facilitados.

2.6 - Semântica em Sistemas de Gerência de Configuração de Software

Segundo o padrão IEEE 610.12 (apud MURTA, 2006), a gerência de configuração é “uma disciplina que aplica procedimentos técnicos e administrativos para identificar e documentar as características físicas e funcionais de um Item de Configuração (IC), controlar as alterações nessas características, armazenar e relatar o processamento das modificações e o estágio da implementação e verificar a compatibilidade com os requisitos especificados”. Sob a ótica de desenvolvimento de software, um dos principais sistemas encontrados para apoiar a Gerência de Configuração de Software (GCS) são os sistemas de controle de versão (CV).

A maioria dos sistemas de GCS existentes trata o software como um conjunto de arquivos em um sistema de arquivos convencional. Isso normalmente acarreta uma diferença entre o domínio de desenvolvimento de software (nível semântico) e o domínio de gerência de configuração (nível de arquivos). Muitos sistemas de GCS tratam o software sendo construído como um conjunto de arquivos em diretórios dentro de um sistema de arquivos, em que configurações estáveis são definidas como conjuntos de versões de arquivos definidas em certas *baselines*¹⁰. Essa diferença é, em parte, ocasionada pelo propósito inicial dos sistemas de CV de arquivos, parte integrante dos sistemas de GCS. Sistemas de CV convencionais contam com um algoritmo de comparação sintática de arquivos, que atende muito bem às necessidades de comparação entre arquivos de texto. Nos casos em que existe uma estrutura diferenciada de formato (xml, doc) ou até mesmo de semântica, o mesmo algoritmo não é adequado. Nesses casos se faz necessária a existência de outros mecanismos de comparação (MURTA, 2006).

O *framework* Molhado, desenvolvido por Nguyen (2006), objetiva gerenciar um software de qualquer domínio sob uma abordagem lógica. Dessa forma os desenvolvedores devem efetuar atividades de GCS centradas nos objetos gerados por eles, sem se preocupar com sua representação em arquivos. Para que os desenvolvedores possam usufruir do *framework*, eles devem modelar seus domínios de aplicação usando o modelo de sistemas do próprio *framework*.

¹⁰ Conforme o padrão IEEE 610.12 (apud MURTA, 2006) o termo linha base (*baseline*) representa um conjunto de ICs formalmente aprovados que serve de base para as etapas seguintes de desenvolvimento.

Ou seja, eles devem representar seus domínios em termos de classes Java especializando as classes do *framework* Molhado. Consequentemente o sistema de versionamento dos objetos será uma especialização do próprio *framework*.

Para que o *framework* proposto funcione corretamente, existe uma forte interoperação entre os módulos de GCS e as ferramentas especializadas para objetos lógicos (editores). Para que essas ferramentas possam manipular as propriedades de um objeto e usufruir dos serviços de GCS, elas fazem uso da biblioteca provida pelo *framework*. Dessa forma o trabalho de Nguyen adiciona um nível semântico ao controle de versão de artefatos, funcionalidade altamente desejada na proposta principal dessa dissertação.

2.7 - Considerações Finais do Capítulo

A documentação desempenha um papel fundamental de apoiar a disseminação de conhecimento. Atualmente a maior parte das ferramentas usadas para geração de documentos considera apenas o conteúdo sintático dos mesmos e buscas a esses documentos são totalmente baseadas na análise sintática de caracteres contidos nos mesmos. Esse mesmo cenário é observado na *web*: páginas sem informações adicionais acerca do seu conteúdo não proveem formas de resolução de possíveis ambiguidades encontradas pelas atuais máquinas de busca. Nesse contexto a *web* semântica visa enriquecer semanticamente páginas publicadas na *web*, possibilitando, dessa forma, maneiras de se resolver ambiguidades e melhorar, portanto, resultados de buscas. O enriquecimento semântico de documentos é conhecido como anotação semântica. A criação de anotações semânticas demanda o uso de ontologias, que proveem a padronização dos possíveis vocabulários usados nas anotações.

A Documentação Semântica visa, basicamente, unir documentos e ontologias. Há, portanto, uma relação íntima com o conceito da *Web Semântica*, onde páginas *web* são anotadas, criando relacionamentos entre páginas e ontologias. A principal diferença existente é o fato de documentos semânticos serem normalmente tratados como arquivos anotados semanticamente, gerados e renderizados por ferramentas de edição de texto *desktop*, enquanto as páginas semanticamente anotadas na *web* são renderizadas por ferramentas de navegação *web*.

Apesar do crescente número de empresas que utilizam gerenciadores de documentos eletrônicos *on-line*, o uso de documentação eletrônica é, ainda, umas das soluções mais escolhidas por empresas que desenvolvem software. Segundo LethBridge et al. (2003),

ferramentas *desktop* de edição de texto (como o *Microsoft Word*) são as que possuem maior número de utilização para produção de documentos de software.

Em empresas de software maduras o desenvolvimento de projetos é apoiado pelo uso de modelos de documento de software. O uso de tais artefatos auxilia e padroniza a produção de documentos de software. Assim, o uso de modelos de documento semânticos pode apoiar ainda mais as organizações de software, uma vez que os documentos semânticos gerados a partir dos modelos permitem a extração do conteúdo semântico, o que possibilita efetuar pesquisas acerca do mesmo. Nesse contexto é importante não só gerenciar os documentos relativos à organização, mas também permitir certo nível de visibilidade do conteúdo dos mesmos, apoiando a condução de possíveis projetos de software que fazem uso de modelos semânticos. Faz-se necessário, então, uma plataforma que gerencie documentos semânticos.

Observando os trabalhos estudados é possível verificar um certo nível de sinergia. A Tabela 2.2 apresenta um comparativo dos trabalhos anteriormente apresentados, segundo alguns pontos considerados importantes para a construção de uma plataforma de gerenciamento de documentos semânticos. Dentre os pontos considerados têm-se: (i) o uso de ontologias no apoio à anotação semântica; (ii) automatização de anotações via modelos de documento; (iii) gerenciamento de documentos semânticos *desktop* e, finalmente, (iv) extração e busca semântica.

Tabela 2.2 - Comparativo entre os trabalhos estudados e características consideradas importantes

Trabalhos	Características			
	Uso de ontologias no apoio à anotação semântica	Automatização de anotações usando modelos de documentos	Gerenciamento de documentos semânticos <i>desktop</i>	Extração e busca semântica
<i>SemanticWord</i> (TALLIS, 2003)	X	X	X	X
Arquitetura para Gerenciamento de Documentos Semânticos (ERIKSSON, 2007a e 2007b; ERISSON;BANG, 2006)	X		X	X
Ambiente de Conteúdo Integrado (SEFTON et al., 2009)		X	X	X
Sistema de Gerenciamento de Documentos Semânticos (NESIC et al., 2007; NESIC et al., 2008)	X		X	X
Documentação Semântica apoiada por Tecnologias da <i>Web</i> Semântica (KIM et al., 2006)			X	X

Embora essa sinergia exista, foi observado que há uma lacuna no que se refere ao versionamento do conteúdo semântico extraído a partir dos documentos semânticos. Apesar de não tratar documentos semânticos propriamente ditos, Nguyen (2006) observou que o versionamento oferecido por sistemas de controle de versão convencionais não trata o versionamento lógico dos dados, fazendo apenas o versionamento sintático. O versionamento dos dados provenientes do conteúdo semântico de documentos é especialmente importante quando se deseja acompanhar a evolução dos dados ali contidos. Nesse sentido, a plataforma para gerenciamento de documentos semânticos poderia contar com o versionamento do conteúdo semântico de cada documento semântico considerado no sistema de controle de versão utilizado. Dessa forma, além de fornecer esse acompanhamento, há a possibilidade de integrar os conteúdos semânticos dos documentos e, com apoio de um ferramental apropriado, prover uma maior visibilidade dos dados contidos nos documentos.

No próximo capítulo é apresentada a Plataforma de Gerenciamento de Documentos Semânticos proposta neste trabalho que visa: (i) permitir a criação de modelos de documento semânticos; (ii) controlar a versão do conteúdo semântico de cada documento semântico gerado a partir desses modelos; (iii) integrar os dados extraídos dos diferentes documentos existentes sob controle de versão; (iv) notificar pessoas interessadas sobre as alterações ocorridas durante modificações nos dados contidos nos documentos semânticos gerenciados; e (v) permitir visibilidade dos dados integrados e da evolução dos mesmos.

Capítulo 3

Uma Plataforma para Gerenciamento de Documentos Semânticos

Este capítulo trata do objetivo geral desta dissertação: prover uma infraestrutura para gerenciamento de documentos semânticos que contemple desde a criação dos mesmos, fazendo uso de anotações semânticas semi-automatizadas, passando pelo seu armazenamento, extração e versionamento do seu conteúdo, até a integração de conteúdos gerados por diferentes documentos encontrados na plataforma, permitindo, assim, buscas especializadas.

3.1 - Introdução

Apesar dos avanços da documentação eletrônica juntamente com o aparecimento de ferramentas de edição colaborativa de texto (como alguns maquinários de edição colaborativa *online*, tais como os *wikis*), editores de texto para ambientes *desktop* ainda continuam sendo largamente utilizados pelas organizações durante o ciclo de vida de seus projetos (LETHBRIDGE et al., 2003). Tanto na abordagem moderna (uso de *wikis*, por exemplo) quanto na abordagem clássica (uso de editores de texto *desktop*), os documentos gerados por essas ferramentas ainda são a base para disseminação de conhecimento acerca de um domínio (ERIKSSON, 2007).

No contexto de um projeto de software esses documentos contêm um grande número de informações (tais como descrições de casos de uso, requisitos, *milestones*, alocações etc.) que são, na maioria das vezes, unicamente compreendidos por leitores humanos. Somado a este fato, o próprio processo de produção e manutenção dos documentos pode gerar inconsistências e imprecisões (ROCHA et al., 2001), dificultando a comunicação dos envolvidos em um projeto. Além disso, o acompanhamento da evolução dos dados contidos nos documentos só é possível mediante a leitura de cada versão do documento que contém um certo dado. Assim, essa atividade torna-se enfadonha e passível de erros.

A comunicação entre as partes envolvidas em um projeto pode ser facilitada se os dados contidos nos documentos forem expostos de forma a não ser necessária a leitura completa de um documento, ou de uma versão de um documento, para obter o entendimento do mesmo, dando,

portanto, visibilidade do seu conteúdo (dados e relacionamentos entre os mesmos) de forma automatizada. Já o acompanhamento da evolução dos dados pode ser efetuado com uma estratégia conjunta de anotação de metadados e versionamento do conteúdo semântico dos documentos. Para que haja tanto um quanto o outro, é fundamental que os documentos apresentem algum nível de riqueza semântica.

Para enriquecer documentos de forma que os mesmos sejam passíveis de leitura humana e, ao mesmo tempo, processáveis por máquina, é importante que metadados sejam adicionados aos documentos, preferencialmente de forma automatizada, já que o processo de anotação manual é tedioso e susceptível a erros (ERIKSSON, 2007; UREN et al., 2006). Adicionalmente, é fundamental que existam estratégias de extração dos metadados previamente anotados e que estes sejam expostos de modo a permitir buscas mais finas acerca das informações contidas nos documentos. Nesse contexto, uma infraestrutura de anotação semântica se faz necessária.

Dentre as abordagens analisadas no Capítulo 2, é possível observar alguns aspectos comuns, de forma que se podem estender essas abordagens visando aumentar os ganhos no uso de uma arquitetura voltada para o gerenciamento de documentos semânticos. As abordagens estudadas apresentam alguma das seguintes características:

1. Proposta de uma forma de se anotar documentos (a partir do uso de ferramentas de anotação manual, por exemplo) (TALLIS, 2003) (ERIKSSON, 2007a e 2007b; ERISSON; BANG, 2006) (SEFTON et al., 2009) (NESIC et al., 2007; NESIC et al., 2008) (KIM et al., 2006);
2. Uso de ontologias de domínio para guiar a criação e edição de documentos semânticos, ligando as estruturas encontradas nos documentos a elementos (conceitos, relações e indivíduos) encontrados na ontologia usada para efetuar as anotações nos mesmos (TALLIS, 2003) (ERIKSSON, 2007a e 2007b; ERISSON; BANG, 2006) (NESIC et al., 2007; NESIC et al., 2008);
3. Uso de uma ontologia de documento, que capture as estruturas tipicamente encontradas em documentos, e de um modelo de anotações utilizado juntamente com uma ontologia de domínio relativa ao domínio que o texto descreve. Dessa forma busca-se obter conformidade entre a tríade estrutura do texto, anotação e elementos da ontologia (TALLIS, 2003) (ERIKSSON, 2007a e 2007b; ERISSON; BANG, 2006);
4. Mecanismos especializados na extração dos dados contidos em documentos para capturar

seu conteúdo semântico (TALLIS, 2003) (ERIKSSON, 2007a e 2007b; ERISSON; BANG, 2006) (SEFTON et al., 2009) (NESIC et al., 2007; NESIC et al., 2008) (KIM et al., 2006);

5. Repositório de documentos semânticos ou de dados extraídos a partir dos mesmos (TALLIS, 2003) (ERISSON; BANG, 2006) (NESIC et al., 2007; NESIC et al., 2008) (KIM et al., 2006);
6. Sistemas especializados na busca de dados contidos em documentos (TALLIS, 2003) (ERISSON; BANG, 2006) (NESIC et al., 2007; NESIC et al., 2008) (KIM et al., 2006).

Essas características, direta ou indiretamente, foram levadas em consideração na definição da arquitetura da Plataforma para Gerenciamento de Documentos Semânticos apresentada neste capítulo. Para atender ao primeiro ponto, a abordagem definida foi utilizar um formato específico de documento, juntamente com uma linguagem para definição de anotações em modelos de documento. O ponto 3, em especial, tem relação direta com essa solução, já que se aproveitou o fato de usar o próprio formato escolhido para comportar elementos da linguagem de anotação em modelos. Nesse caso não foi utilizada uma ontologia de documentos. Com relação ao ponto 2, a estratégia foi similar aos trabalhos estudados. No atendimento ao ponto 4, usou-se a própria linguagem de anotação em modelos, documentos instanciados a partir desse modelo e mecanismos que fazem a leitura e transformação das anotações em conteúdo semântico. No que se refere ao ponto 5, foram utilizados tanto um Repositório de Documentos Semânticos, quanto um Repositório de Dados, o primeiro sendo um repositório de um sistema de controle de versão e o segundo um banco de dados relacional. Para efetuar buscas especializadas, visando atender ao ponto 6, optou-se por utilizar padrões definidos para pesquisa em grafos. Ao longo deste capítulo, cada um desses aspectos é discutido em detalhes. A seguir a visão geral da arquitetura é apresentada.

3.2 - Visão Geral da Arquitetura

Como foi observado anteriormente, os aspectos mais comumente tratados pelas arquiteturas propostas na literatura levam a um modelo satisfatoriamente robusto, não havendo necessidade de se fugir dele. O interessante é incrementá-lo, de forma a preencher as lacunas citadas por alguns autores, e prover mais facilidades para os usuários finais. Assim, em essência, a proposta de arquitetura sugerida neste trabalho apresenta as seguintes características:

1. Proposta de uma forma de se anotar modelos de documentos (a partir do uso de uma ferramenta, por exemplo), obtendo, assim, a automatização parcial do processo de anotação semântica. Essa abordagem está em linha com algumas das ideias de Tallis (2003);
2. Prover versionamento do conteúdo extraído de um documento semântico para apoiar a rastreabilidade de alterações no mesmo;
3. Prover visibilidade do conteúdo de um documento semântico, bem como da integração de diferentes conteúdos semânticos;
4. Prover um esquema de notificação automática mediante uma alteração de um elemento contido em algum documento semântico tratado na plataforma, mantendo uma forma proativa de conhecimento sobre elementos de interesse localizados na plataforma.

Na arquitetura proposta, o “conteúdo” considerado de um documento ou de sua versão é o conjunto de indivíduos que têm um mapeamento direto para conceitos e propriedades de uma ontologia de domínio e são passíveis de extração automatizada. Esses indivíduos e os valores de suas propriedades estão descritos no documento na forma de estruturas do mesmo, tais como trechos de texto, células de tabela, seções, etc. O conjunto de indivíduos e propriedades extraído de um documento é representado na forma de um grafo conectado. Os termos “conjunto de dados”, “dados” e “conteúdo semântico” são usados para denotar esse grafo.

No caso de leitura e entendimento de documentos por humanos, é a cognição humana que cria um grafo desse tipo na mente. Naturalmente cada pessoa pode ter um entendimento diferente do documento (dependendo inclusive do momento da leitura) e, portanto, grafos diferentes são criados. No caso de sistemas computacionais, a extração do conteúdo de um documento e, portanto do grafo, requer um processo automatizado e padronizado, já que a máquina não deve ter mais de uma interpretação sobre um mesmo documento. Nesse caso, o papel das ontologias é

fundamental para padronizar o vocabulário usado na criação do grafo, além de permitir inferências (baseadas nos axiomas descritos na ontologia) sobre os indivíduos extraídos. Esse é um aspecto fundamental para a criação das anotações semânticas contidas em documentos.

A Figura 3.1 mostra uma visão geral da arquitetura proposta, a qual leva em consideração as características descritas anteriormente. Nela podem-se observar os módulos encontrados na plataforma, bem como os elementos terceiros dos quais a plataforma depende, formando, portanto, a arquitetura do sistema.

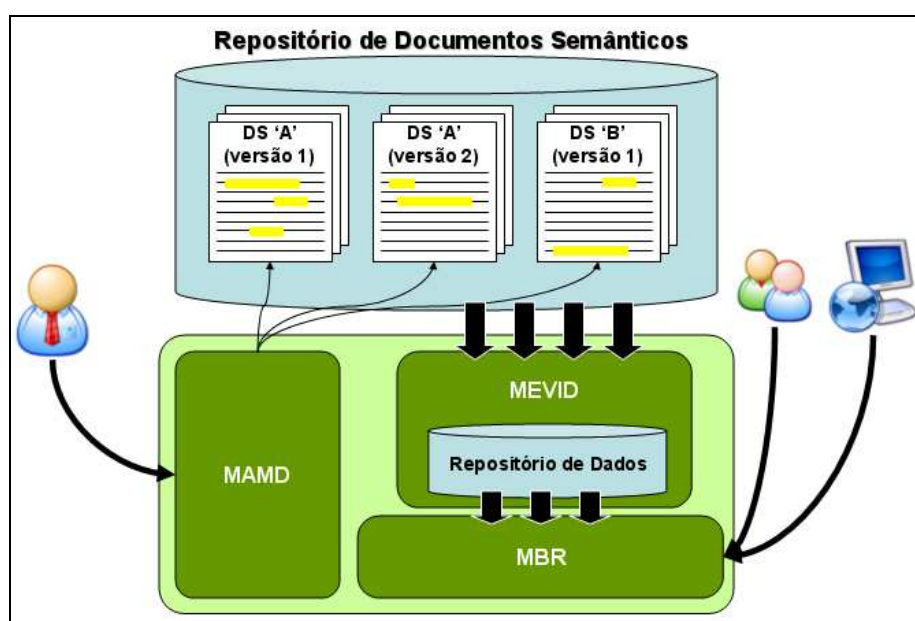


Figura 3.1 - Visão Geral da Plataforma de Gerenciamento de Documentos Semânticos

Em um nível mais alto, a arquitetura é composta por dois elementos principais, a saber:

- Repositório de Documentos Semânticos (RDS): responsável por armazenar os documentos previamente anotados e manter o histórico evolutivo dos mesmos. O RDS é um repositório de um sistema de controle de versões e a plataforma em si (mostrada em verde na Figura 3.1) depende desse elemento para receber notificações sobre a criação de novas versões de documentos. Assim, o RDS deve permitir a adição de ganchos (*hooks*), que serão utilizados para notificar a plataforma sobre a geração de novas versões de documentos semânticos;
- Plataforma de Gerenciamento de Documentos Semânticos, composta pelos seguintes

módulos:

- Módulo de Anotação em Modelo de Documento (MAMD): responsável por permitir a elaboração de anotações semânticas em modelos de documento;
- Módulo de Extração, Versionamento e Integração de Dados (MEVID): responsável por extrair o conteúdo semântico dos documentos persistidos em um RDS no momento em que uma nova versão de um documento semântico for registrada nesse RDS. Dessa forma, mantém-se a rastreabilidade da evolução do conteúdo de um documento semântico e se permite a notificação das alterações (diferenças entre versões de um conteúdo). Além disso, os serviços desse módulo se encarregam de persistir e integrar o conteúdo semântico relevante levantado a partir das extrações. Para registrar o conteúdo semântico extraído e versionado, esse módulo conta com um Repositório de Dados, implementado na forma de um banco de dados relacional;
- Módulo de Busca e Rastreabilidade (MBR): responsável por permitir pesquisas refinadas acerca dos dados contidos no Repositório de Dados do MEVID. Adicionalmente, esse módulo permite a visão da evolução de um indivíduo escolhido.

Em resumo, o engenheiro de documento (representado pelo ícone de ser humano à esquerda na Figura 3.1) usa o MAMD para anotar modelos de documento de sua organização. Posteriormente desenvolvedores e analistas podem instanciar documentos a partir dos modelos de documentos anotados, gerando, dessa forma, documentos semânticos. Em algum momento (fase de *checkin* de alterações) esse documento pode ser adicionado a um RDS. Quando uma nova versão de um documento semântico fica disponível nesse RDS, o MEVID extrai o conteúdo daquela versão e armazena o resultado no Repositório de Dados (RD), disponibilizando, portanto, o conteúdo extraído. Nesse ponto é possível efetuar a integração das informações extraídas por vários documentos semânticos disponíveis no RDS. Isso é feito a partir da união dos grafos que correspondem às últimas versões de cada documento contido nesse repositório. Finalmente, usuários e outros sistemas (mostrados à direita da Figura 3.1) podem interagir com o MBR, efetuando buscas sobre o grafo de união e podendo verificar o histórico evolutivo de um indivíduo.

3.2 – Requisitos da Plataforma de Gerenciamento de Documentos Semânticos

Na fase de concepção da plataforma, os seguintes requisitos foram levantados:

- req1. O sistema deve permitir a criação de modelos de documento semânticos;
- req2. O sistema deve permitir a adição e edição de anotações semânticas baseadas em ontologias de domínio em modelos de documento de texto;
- req3. O sistema deve ser capaz de capturar o registro de versões em um repositório de controle de versão previamente registrado;
- req4. O sistema deve ser capaz de efetuar a leitura das anotações semânticas contidas em um documento semântico e capturar o seu conteúdo semântico;
- req5. O sistema deve ser capaz de manter versões dos conteúdos extraídos a partir das versões de documentos semânticos;
- req6. O sistema deve permitir buscas acerca dos dados referentes a um documento semântico, baseando-se nas ontologias usadas para criar as anotações semânticas contidas no mesmo documento;
- req7. O sistema deve ser capaz de manter a rastreabilidade entre os dados capturados (conteúdo) e a versão de documento semântico usada para geração dos mesmos;
- req8. O sistema deve ser capaz de manter a diferença semântica existente entre duas versões consecutivas de conteúdos referentes ao mesmo documento semântico em versões consecutivas;
- req9. O sistema deve permitir que usuários se cadastrem para receber notificações sobre alterações de dados contidos no repositório de dados;
- req10. O sistema deve notificar as partes interessadas em uma informação particular, toda vez que a mesma é alterada;
- req11. O sistema deve ser capaz de integrar as informações relativas às anotações semânticas contidas em diferentes versões de diferentes documentos semânticos;
- req12. O sistema deve permitir a navegação entre os dados exibidos a partir de uma consulta e as versões de documentos semânticos utilizados para prover o resultado dessa consulta.

De posse dos requisitos, foram estabelecidos os casos de uso para atendê-los, representados na Figura 3.2, agrupados por módulo definido da plataforma.

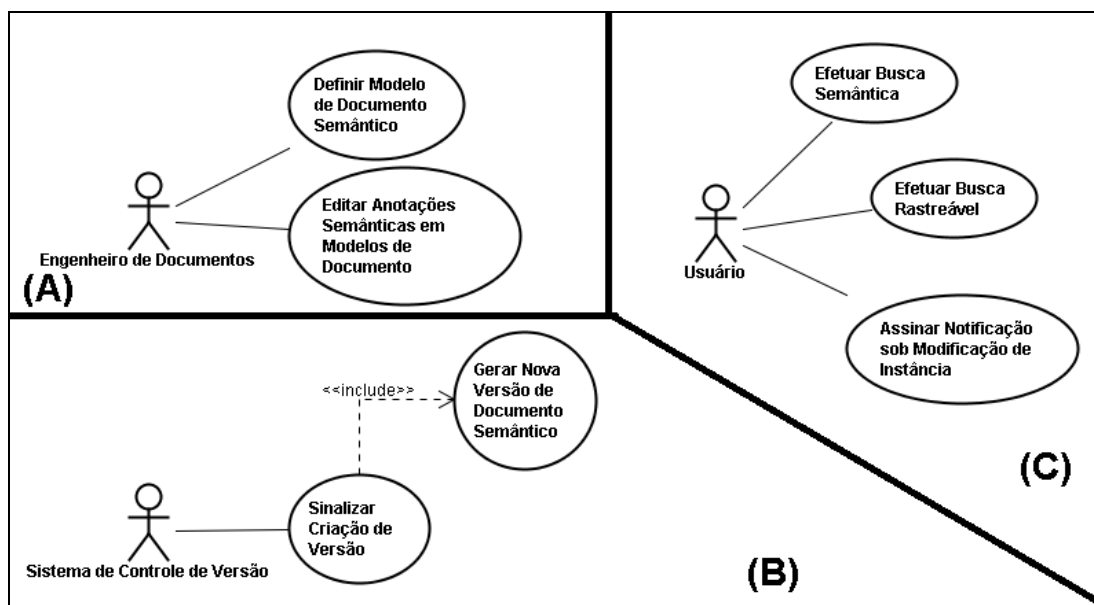


Figura 3.2 - Casos de Uso do Sistema - (A) Casos de uso de MAMD; (B) Casos de Uso de MEVID; (C) Casos de Uso de MBR

No MAMD foram definidos os seguintes casos de uso (Figura 3.2(A)):

- **Definir Modelo de Documento Semântico:** responsável por permitir que engenheiros de documentação criem Modelos de Documento Semânticos.
 - o Requisito atendido: req1
- **Editar Anotações Semânticas em Modelos de Documento:** responsável por permitir que usuários adicionem, editem e removam anotações semânticas baseadas em ontologias de domínio em modelos de documento.
 - o Requisito atendido: req2

No MEVID foram definidos os seguintes casos de uso (Figura 3.2(B)):

- **Sinalizar Criação de Versão:** responsável por integrar o sistema de controle de versão à plataforma, de forma que o mesmo sinalize a criação de versões, desencadeando a geração de novas versões de documentos semânticos (portanto, iniciando o caso de uso *Gerar Nova Versão de Documento Semântico*)

- o Requisito atendido: req3
- **Gerar Nova Versão de Documento Semântico:** responsável por: (i) registrar versões de documentos semânticos criados no repositório de documentos semânticos; (ii) extrair o conteúdo semântico de versões desses documentos semânticos; (iii) gerar as diferenças semânticas entre duas versões consecutivas de um documento semântico; (iv) identificar interessados em instâncias contidas no conteúdo semântico de alguma versão de documento semântico tratado e notificar os mesmos sobre as alterações; (v) integrar os conteúdos semânticos extraídos originados da geração de uma versão no repositório de documentos semânticos.
 - o Requisitos atendidos: req4, req5, req7, req8, req10 e req11.

Para o MEVID foram definidos os seguintes casos de uso (Figura 3.2(C)):

- **Efetuar Busca Semântica:** responsável por permitir que usuários efetuem consultas sobre os dados contidos no repositório de dados.
 - o Requisito atendido: req6
- **Efetuar Busca Rastreável:** responsável por: (i) permitir que usuários visualizem as modificações ocorridas em uma certa informação (instância) em um conjunto de versões, exibindo suas propriedades e valores, juntamente com as versões de documentos utilizadas para a geração do resultado; (ii) permitir que usuários verifiquem o histórico de evolução de um documento semântico, dado um conjunto de versões daquele documento, exibindo as modificações ocorridas e as versões que efetuaram essas modificações.
 - o Requisitos atendidos: req6 e req12
- **Assinar Notificação sobre Modificação em Instância:** responsável por permitir que usuários definam um conjunto de instâncias sobre as quais desejam ser notificados quando ocorrerem alterações.
 - o Requisito atendido: req9

A próxima seção trata de um aspecto fundamental para a plataforma: a anotação semântica de modelos de documento.

3.3 - Anotação Semântica em Modelos de Documento

Como discutido na seção 2.3 do Capítulo 2, existem algumas estratégias listadas na literatura para apoiar o processo de anotar documentos semanticamente. Notavelmente, a maior parte deles (vide Tabela 2.2) faz uso de ontologias em certo ponto. A padronização de vocabulário e geração dos metadados são os principais cenários de uso de ontologias no contexto de anotação semântica de documentos. A forma mais simples de se criar anotações em documentos é criá-las de forma manual, normalmente com o apoio de ferramentas de anotação. Esse cenário é reconhecidamente problemático por consumir um tempo considerável e permitir que a pessoa que efetua as anotações – o anotador – cometa erros durante o processo. Por esses motivos, Eriksson e Bang (2006) apontam a necessidade de automatizar o processo de anotação ao máximo. Adicionalmente, é interessante prover algum nível de transparência para o usuário final, fazendo com que ele não tenha conhecimento da geração do documento semântico.

Tallis (2003) propôs uma forma interessante de se anotar documentos editados na ferramenta Microsoft Word. Na abordagem proposta, modelos de documento eram anotados de maneira a permitir o reúso das anotações. Dessa forma, tanto a automatização quanto a transparência na criação das anotações foram, de alguma maneira, alcançadas. Considerando o fato que a maioria das organizações de software maduras faz uso de modelos de documentos para apoiar o processo de documentação, essa proposta pode ajudar empresas a organizar e controlar os dados armazenados em seus repositórios de documentos.

A proposta apresentada neste trabalho compartilha ideias de Tallis (2003) e consiste em prover anotações baseadas em ontologias de domínio embutidas em modelos de documento, permitindo que as instâncias criadas a partir desse modelo sejam semanticamente enriquecidas sem que haja a necessidade de intervenção humana para tal. Dessa forma oferece-se um certo nível de automatização na criação de documentos semânticos, trazendo transparência na criação dos mesmos. Para efeito de padronização e simplificação, o termo “modelo semântico”, usado neste trabalho, se refere a um modelo de documento anotado semanticamente.

Um modelo semântico deve guiar a criação de documentos semânticos de forma transparente. A Figura 3.2 apresenta um exemplo de modelo semântico e sua instanciação, no qual as anotações semânticas estão marcadas em amarelo.

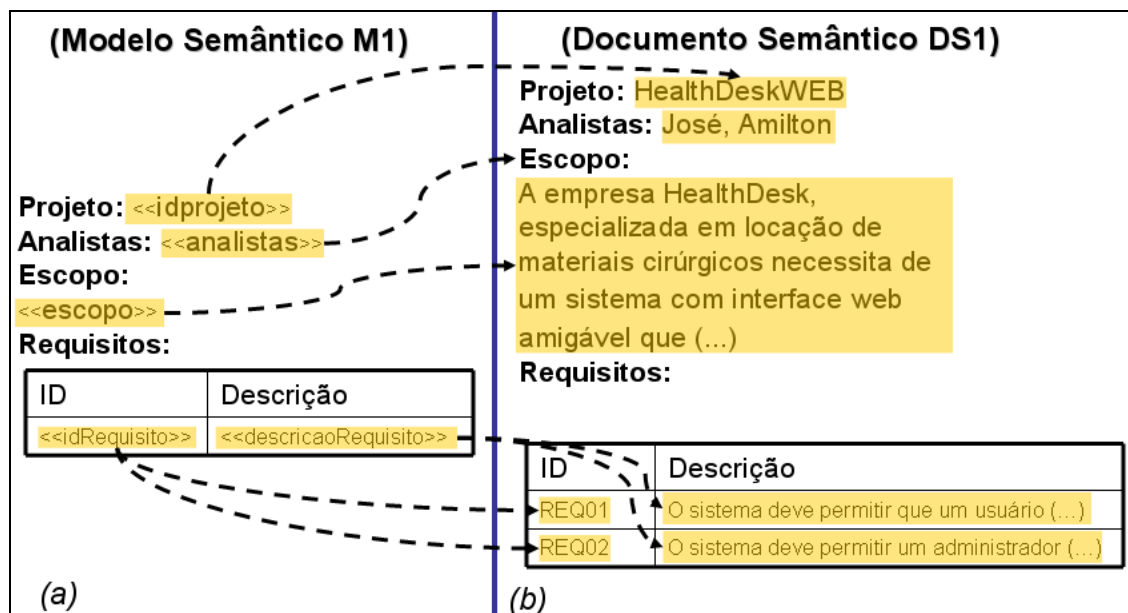


Figura 3.2 - (a) Modelo Semântico e (b) Documento semântico originado a partir do modelo

Primeiramente, é importante que um modelo de documento contemple anotações semânticas que permitam a criação de indivíduos e relacionamentos durante a instanciação de um documento, apoiando a automatização do processo de anotar documentos semanticamente. No exemplo, o modelo de documento com a estrutura apresentada na Figura 3.2(a) deve ser capaz de gerar o documento mostrado na Figura 3.2(b), contendo os indivíduos e relacionamentos apresentados na Figura 3.3. Dessa forma, ao instanciar o modelo *M1* e preencher os campos anotados, um usuário estará criando um documento semântico.

Em resumo, o modelo semântico *M1* apresentado na Figura 3.2(a) contempla o seguinte cenário de anotação e geração de conteúdo semântico (Figura 3.3) durante a instanciação de um documento:

- A anotação <<idprojeto>> corresponde a uma propriedade (*id*) de um conceito existente na ontologia de domínio (*Projeto*) usada para anotar semanticamente o modelo *M1*. Quando o modelo de documento *M1* é instanciado, dando origem ao documento semântico *DS1*, é criada uma instância de *Projeto* e o conteúdo da anotação (no caso, o valor “HealthDeskWeb”) é atribuído à propriedade (*id*) desse indivíduo, como mostra a Figura 3.3.

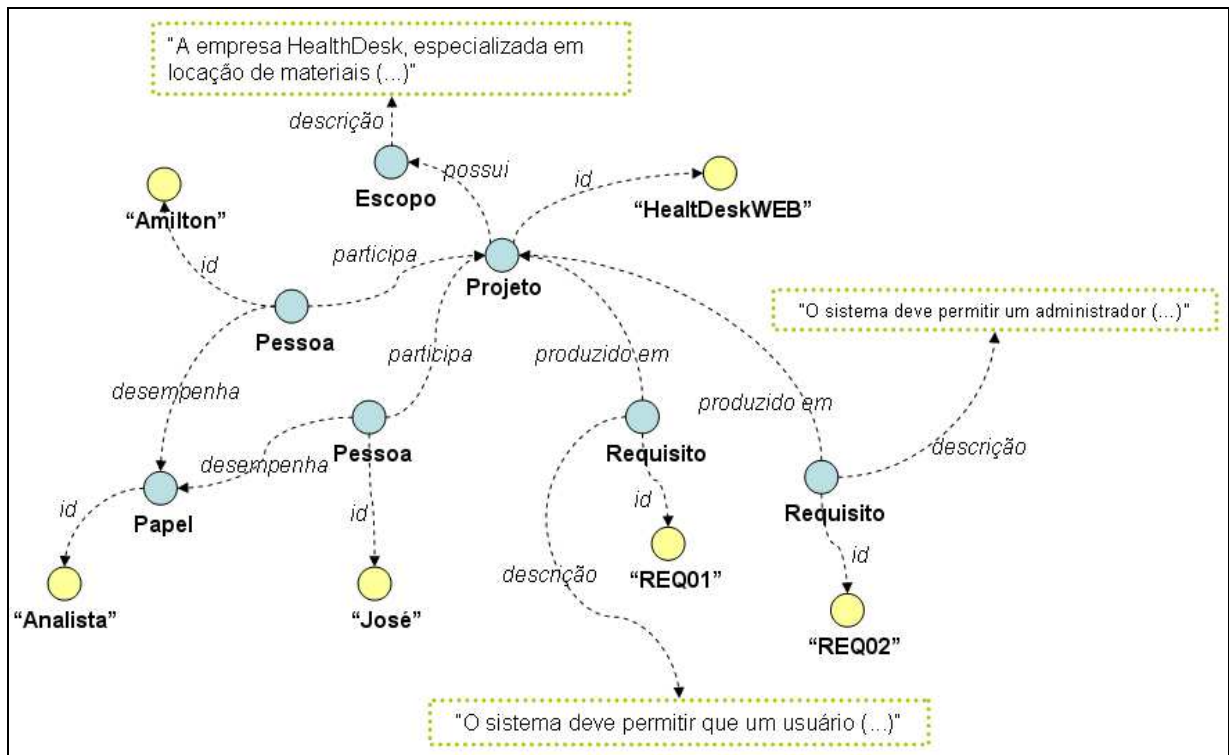


Figura 3.3 - Grafo gerado a partir da instanciação do modelo semântico da Figura 2.

- A anotação `<<analistas>>` gera, para cada nome encontrado, uma instância de *Pessoa* com identificador (*id*) igual ao conteúdo textual encontrado (no exemplo, José e Amilton). As instâncias geradas possuem relacionamentos com outros indivíduos, dados por uma ontologia de domínio. Esses relacionamentos são igualmente estabelecidos, indicando que José e Amilton (*Pessoa*) desempenham o papel (*Papel*) de analista (relacionamento *desempenha* na Figura 3.3) e que eles participam do projeto HealthDeskWeb (relacionamento *participa* na Figura 3.3);
- A anotação `<<escopo>>` gera uma instancia do conceito *Escopo*, contendo como descrição (propriedade *descrição* na Figura 3.3) conteúdo textual do parágrafo marcado pela anotação ("A empresa HealthDesk, especializada em locação de materiais..."). Esse indivíduo, por sua vez, é relacionado à instância do conceito *Projeto*, criada no passo anterior;
- As últimas anotações do modelo de documento *M1* anotam colunas de uma tabela: `<<idRequisito>>`, indica que a primeira coluna se refere ao identificador de um requisito, enquanto `<<descricaoRequisito>>` indica que a segunda coluna se refere à descrição de

um requisito. Durante a instanciação do modelo, cada linha criada na tabela gera uma instância do conceito *Requisito*, contendo como identificador (*id*) o conteúdo da primeira célula e como descrição (*descrição*) o conteúdo da segunda célula. Além disso, cada instância de *Requisito* criada é relacionada com a instância do conceito *Projeto* previamente criada, espelhando a relação *produzido em* descrita na ontologia de domínio. Assim, no caso da Figura 3.2b, têm-se duas instâncias de *Requisito* (*REQ01* e *REQ02*), cada uma com a descrição definida na coluna consecutiva ao seu nome, estando ambas relacionadas ao projeto HealthDeskWeb.

Um modelo semântico deve, portanto, guiar a criação de um documento semântico de forma transparente, de acordo com as anotações definidas pela pessoa responsável por criar o modelo semântico, tomando por base uma ontologia de domínio. Dessa forma, se faz necessária a criação de uma linguagem especializada para se efetuar anotações em modelos de documentos. Essa linguagem foi desenvolvida neste trabalho e a mesma é apresentada a seguir.

3.4 - Linguagem de Anotações em Modelos de Documentos

A primeira decisão tomada durante a elaboração da linguagem foi a escolha do formato de arquivos dos modelos de documentos. O formato ODF foi escolhido por ser um formato aberto, que vem sendo largamente usado, além de prover algumas formas de adição de metadados. Esse formato conta atualmente com duas versões principais: 1.1 e 1.2. Na primeira, o suporte a metadados é pequeno, se restringindo a anotações não alinhadas com os padrões fornecidos pela W3C e sendo totalmente focadas em adicionar metadados sobre o documento (metadados como título, assunto do documento, comentários, autores, palavras-chave, etc.). Anotações referentes ao corpo do texto não são possíveis nessa versão. Mais detalhes sobre essa versão do formato ODF podem ser encontrados em (DURUSAU; BRAUER, 2009).

O suporte extensível a metadados proposto na especificação ODF 1.2 traz inúmeras vantagens. Além de possibilitar anotações referentes ao documento como um todo, ela permite anotação de elementos do conteúdo do texto. Adicionalmente, o padrão RDF é usado, melhorando a possibilidade de integração com sistemas externos e preparando documentos para a *web* semântica (DURUSAU; BRAUSER, 2009). Entretanto, até o presente momento, a especificação não foi completamente implementada por nenhuma ferramenta livre. O pacote de aplicativos *OpenOffice* na versão 3.2 é aderente ao formato ODF 1.2 e prevê uma API de

metadados em RDF. Por fim, a especificação ODF 1.2 não exige a definição de uma ontologia a ser usada para criar anotações em um documento. Pode-se utilizar qualquer implementação de ontologia disponível.

Tanto a ferramenta *OpenOffice* 3.2 quanto seu kit de desenvolvimento (que contempla a API de metadados) foram liberados no início de 2010 (11 de fevereiro). Pela natureza recente tanto da especificação quanto do kit de desenvolvimento, o estado atual das implementações da API de metadados é inacabado e prevê a adição de metadados em alguns poucos elementos de um documento, a saber: parágrafos de texto, *bookmarks* e trechos de texto¹¹. Elementos como tabelas, colunas, linhas, células de tabelas, seções de texto e imagens ainda não são anotáveis. Tabelas, em especial, proveem uma forma facilitada de dinamizar a criação de conteúdo semântico. Por não prover esse tipo de recurso atualmente, não é possível conduzir anotações mais complexas com esse ferramental.

Em resumo, enquanto a especificação de ODF prevê anotações na maioria dos elementos encontrados em um documento desse tipo, o atual estado da implementação dessa API na ferramenta *OpenOffice* contempla um subconjunto mínimo de elementos anotáveis (trecho de texto, parágrafos e *bookmarks*). Além disso, em ODF 1.2, não existe uma forma nativa de usar o conteúdo textual de uma anotação para geração automática de indivíduos. Essa necessidade no âmbito de tabelas anotadas semanticamente fica evidente. No exemplo mostrado na Figura 3.2, cada linha da tabela instancia um indivíduo de um conceito específico, além de criar relacionamentos entre o indivíduo criado recentemente e outros indivíduos previamente criados. Por esses motivos não foi possível utilizar apenas o suporte a metadados provido pelo formato ODF 1.2.

Para suprir essa lacuna, a abordagem adotada faz uso de uma estrutura intermediária que define anotações especializadas para tratar elementos encontrados em um documento de texto, mais especificamente, elementos definidos pela especificação ODF. As anotações proveem diretivas para a realização de ações, tais como criação de indivíduos e relacionamentos, e para

¹¹ OpenOffice - RDF_Metadata, http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/OfficeDev/RDF_metadata, visitado em 17/07/2010

guiar a geração do conteúdo semântico do documento quando o mesmo for processado pela plataforma.

Duas formas de anotação são providas neste trabalho: anotações de fragmentos de texto e anotações em tabelas. Ambas permitem que o responsável pela criação da anotação possa adicionar um conjunto de ações a serem tomadas (criação de indivíduos e relacionamentos) quando o texto for analisado pelo módulo de extração, versionamento e integração de dados (MEVID), que usa essas ações para gerar o conteúdo semântico (grafo de indivíduos e relacionamentos) do documento. Cada ação (ou instrução) é também uma interface direta com o MEVID, permitindo que o mesmo crie variáveis durante a extração do conteúdo semântico de um documento e, portanto, relacione indivíduos criados a partir de uma instância de documento.

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**). Todo indivíduo é, portanto, instância de um conceito provido pela anotação. 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.

No caso da 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 um outro indivíduo ou 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. Nesta dissertação o termo “campo de referência” se refere a um campo de usuário que contém conteúdo semântico.

Conforme citado anteriormente, a adição de anotações semânticas é permitida em dois elementos: trechos de texto e tabelas. Para adicionar uma anotação semântica em um trecho de texto e definir seu conjunto de instruções, os seguintes passos devem ser realizados:

1. Criação de um estilo de formatação com um nome que siga o seguinte padrão: `SemanticAnnotation-ref-<campoReferencia>`. O valor de `<campoReferencia>` é usado para definir o campo de usuário com as instruções dessa anotação semântica;
2. Criação de um campo de referência com o nome de mesmo valor definido no passo anterior (`<campoReferencia>`), contendo uma das seguintes diretivas: (i) `[[textspan]]` seguido do conjunto de instruções separadas por vírgula ou (ii) `[[break with '<separador>']]`, também seguido do conjunto de instruções.

Durante a extração do conteúdo semântico contido em um documento, quando o MEVID encontra uma anotação referente à anotação semântica de texto, uma das seguintes ações é tomada:

- a. No caso de uma anotação do tipo `[[textspan]]`, o MEVID captura o texto completo referente à formatação usada (`SemanticAnnotation-ref-<campoReferencia>`) e armazena esse valor numa variável global acessível pelo nome `{content}`. Se alguma instrução referente a essa anotação contemplar o nome `{content}`, ele será automaticamente substituído pelo valor do conteúdo anotado. Seja o trecho de código abaixo, relativo ao exemplo da Figura 3.2, correspondente à anotação usada na primeira linha anotada da Figura 3.2(a), encontrada no conteúdo do campo de referência de nome `<campoReferencia>`:

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

O resultado de sua execução será a criação de uma instância de *Projeto*, conceito encontrado na URL `http://localhost/ontologies/SE/onto.owl#Projeto`, sendo o seu

identificador (*id*) o conteúdo demarcado por {content}. No caso do documento apresentado na Figura 3.2(b), esse conteúdo é o valor “HealthDeskWEB”. Em termos de RDF, o resultado da extração do documento da Figura 3.2(b) corresponde ao trecho de código 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>
```

- b. No caso de uma anotação do tipo `[[break with '<separador>']]`, o MEVID captura o texto delimitado pela ocorrência da formatação e divide esse valor textual usando um separador (<separador>). Para cada resultado da divisão efetuada, o conjunto de instruções é executado, considerando o valor do resultado da divisão dentro da variável global {slice}. Seja o trecho de código abaixo, relativo ao exemplo da Figura 3.2, correspondente à segunda anotação feita na Figura 3.2(a):

```
[[break with ', ']]
instance({slice},http://localhost/ontologies/SE/onto.owl#Pessoa,$pessoa);
```

Neste caso, tem-se a criação de instâncias do conceito *Pessoa*, cujos identificadores são os valores resultantes da divisão do conteúdo delimitado por vírgulas. No caso do documento apresentado na Figura 3.2(b), são criados dois indivíduos, com identificadores iguais a “José” e “Amilton”, respectivamente.

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 a mesma expressão de um estilo de formatação referente a uma anotação de texto (SemanticAnnotation-ref-<campoReferencia>), obviamente com um campo de referência diferente.

Para cada linha de uma tabela anotada semanticamente, são executados os conjuntos de instruções definidos nos campos de referência de cada coluna da tabela. Ou seja, 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 referência 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 referência não contemplar uma das colunas de uma tabela, essa 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.

Seja a anotação da Tabela de Requisitos da Figura 3.2, apontando para um campo de referência com o seguinte código:

```
[[at 0]]
instance({content},http://localhost/ontologies/SE/onto.owl#RequisitoFuncional,$req);
property($req,http://localhost/ontologies/SE/onto.owl#artefatoProduzidoEm,$projeto);
[[at 1]]
property($req,http://localhost/ontologies/SE/onto.owl#descricao,{content});
```

Ao efetuar a extração dos dados referentes à tabela anotada no documento apresentado na Figura 3.2(b), tem-se o seguinte comportamento:

1. O MEVID lê a primeira célula da primeira linha da tabela e executa as instruções referenciadas pela coluna 0, criando uma instância do conceito *RequisitoFuncional* com identificador igual ao conteúdo textual da célula e mantendo a referência a esse indivíduo na variável *\$req*. Adicionalmente um relacionamento *artefatoProduzidoEm* entre esse indivíduo e um indivíduo previamente criado (referenciado pelo nome *\$projeto*) é gerado.

O resultado dessa execução, em termos de RDF, é dado pelo código abaixo:

```
<rdf:Description rdf:about="http://localhost/ontologies/SE/onto.owl#RF001">
  <artefatoProduzidoEm
    rdf:resource="http://localhost/ontologies/SE/onto.owl#HealthDeskWEB"/>
  <rdf:type
    rdf:resource="http://localhost/ontologies/SE/onto.owl#RequisitoFuncional"/>
</rdf:Description>
```

2. O MEVID lê a segunda célula da primeira linha da tabela e, conforme a instrução fornecida pela coluna 1, cria um relacionamento entre o indivíduo referenciado por *\$req* e o conteúdo da célula, por meio da tripla que relaciona o indivíduo criado no passo anterior

(<http://localhost/ontologies/SE/onto.owl#REQ001>) com o valor “O sistema deve permitir que um usuário (...)”, através da propriedade <http://localhost/ontologies/SE/onto.owl#descricao>. Em termos de RDF, o resultado é o seguinte:

```
<rdf:Description rdf:about="http://localhost/ontologies/SE/onto.owl#RF001">
  <descricao rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    O sistema deve permitir que um usuário (...)</descricao>
  <artefatoProduzidoEm
    rdf:resource="http://localhost/ontologies/SE/onto.owl#HealthDeskWEB"/>
  <rdf:type
    rdf:resource="http://localhost/ontologies/SE/onto.owl#RequisitoFuncional"/>
</rdf:Description>
```

3. O MEVID lê a primeira célula da segunda linha, repetindo o processo apresentado no passo 1 e gerando o indivíduo <http://localhost/ontologies/SE/onto.owl#REQ002> também do tipo <http://localhost/ontologies/SE/onto.owl#RequisitoFuncional>. Adicionalmente o relacionamento entre o indivíduo criado e o indivíduo referenciado pela variável *\$projeto* é criado;
4. O MEVID lê a segunda célula da segunda linha e, similarmente ao passo 2, gera o relacionamento entre o indivíduo criado no passo anterior (<http://localhost/ontologies/SE/onto.owl#REQ002>) e o valor “O sistema deve permitir um administrador (...)” através da propriedade <http://localhost/ontologies/SE/onto.owl#descricao>.

Conforme citado, a linguagem e a estratégia de anotação aqui descritos são utilizados por diferentes módulos da arquitetura. A seguir é detalhado o funcionamento de cada módulo definido na Plataforma de Gerenciamento de Documento Semânticos.

3.5 - Funcionamento da Plataforma de Gerenciamento de Documentos Semânticos

Nesta seção cada módulo da Plataforma de Gerenciamento de Documentos Semânticos é detalhado, sem discutir, contudo, suas particularidades de implementação.

3.5.1 - Módulo de Anotação em Modelos de Documento (MAMD)

O Módulo de Anotação de Modelos de Documento (MAMD) provê um ferramental para efetuar anotações semânticas de forma transparente e semi-automatizada em documentos de texto *desktop*. Para tal, o MAMD se materializa como uma ferramenta para apoiar o processo de anotação em modelos de documento que usa como base os seguintes elementos: (i) a ferramenta de edição de texto *Open Office Writer*; (ii) a linguagem de anotação em modelos semânticos da plataforma e (iii) ontologias de domínio.

A Figura 3.4 mostra o funcionamento geral do MAMD. Em resumo, o engenheiro de documentos (1) usa o ferramental provido por MAMD (que utiliza a linguagem de anotação em modelos de documento juntamente com ontologias de domínio (2)) para anotar os modelos de documento definidos pela organização, gerando modelos enriquecidos semanticamente, ditos modelos semânticos (3). Posteriormente, os desenvolvedores responsáveis por criar e editar instâncias de documento da organização (4) usam os modelos semânticos gerados previamente para instanciar documentos, dando origem a documentos semânticos (5).

O MAMD se utiliza, portanto, de uma estrutura intermediária para anotação de modelos de documentos, a linguagem de anotação em modelos de documento, e não de anotações RDF nas instâncias de documento em si. Entretanto, por não trazer nenhuma estrutura externa à especificação ODF 1.2, a abordagem é compatível com a mesma, pois faz uso dos elementos definidos nessa especificação (estilos de formatação e campos de usuário) para permitir anotações semânticas de modelos de documento baseadas na linguagem proposta.

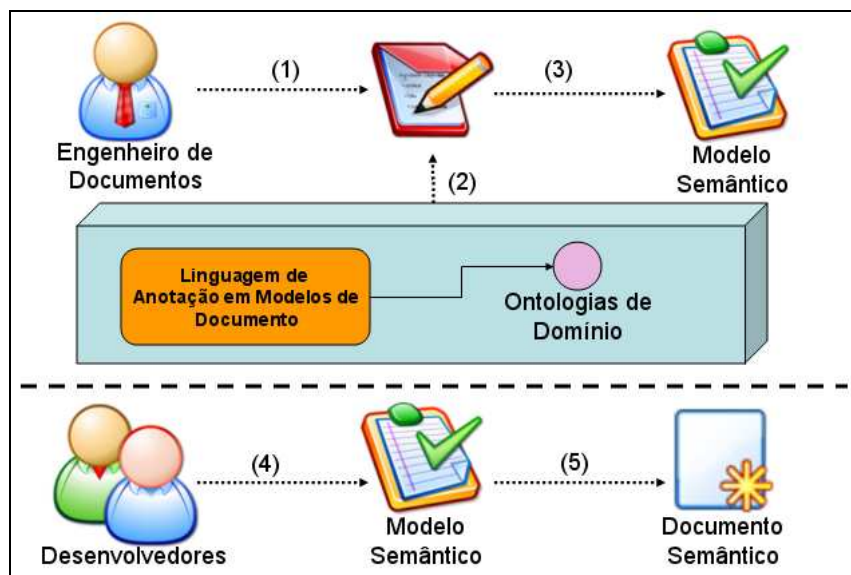


Figura 3.4 - Visão geral do Módulo de Anotação em Modelos de Documento (MAMD)

Vale destacar que, de posse da linguagem de anotação de modelos semânticos e da estratégia de utilizar elementos do formato ODF para criar as anotações, é possível desenvolver uma extensão do *OpenOffice Writer* para apoiar a elaboração de anotações semânticas em modelos de documento. Essa extensão, contudo, não foi desenvolvida neste trabalho.

3.5.2 - Módulo de Extração, Versionamento e Integração de Dados (MEVID)

O Módulo de Extração, Versionamento e Integração de Dados (MEVID) é responsável por extrair o conteúdo semântico das versões de documentos semânticos e registrar o mesmo no Repositório de Dados (RD). Os documentos semânticos a serem considerados para extração, versionamento e integração devem residir em um Repositório de Documentos Semânticos (RDS), que corresponde a um repositório de um sistema de controle de versão convencional, tal como o *Subversion*, utilizado na corrente implementação da plataforma. Vale ressaltar que o MEVID pode tratar diferentes RDSs. Dessa forma abre-se espaço para comparativos referentes aos dados provenientes de RDSs diferentes tratados pelo mesmo MEVID. Adicionalmente, para cada RDS tratado, há um RD que mantém os dados extraídos separados dos dados extraídos de outros RDSs.

A maior parte dos sistemas de controle de versão oferece meios de integração com sistemas externos a partir do uso de mecanismos conhecidos como “ganchos” (*hooks*). Esses mecanismos permitem a adição de um script de código a ser executado em algum evento tratado

pelo sistema de controle de versão. No *Subversion* vários eventos podem ser utilizados como marcos para a execução de ganchos (COLLINS-SUSSMAN et al., 2008), dentre elas o *post-commit*, que permite a execução do script de gancho exatamente após a geração de uma nova versão do repositório de controle de versão (ou seja, após a ocorrência de um *checkin*). Quando essa situação é atingida, duas informações são disponibilizadas para o script de gancho: (i) o caminho do repositório no sistema de controle de versões e (ii) a versão gerada a partir do *checkin*. Aproveitando essa funcionalidade de integração provida pelos sistemas de controle de versão, o gancho *post-commit* foi escolhido para notificar a Plataforma de Gerenciamento de Documentos Semânticos (PGDS) sobre a criação de uma nova versão. O RDS deve, portanto, dispor de um gancho de execução pós *checkin* que conheça a localização da PGDS para que esta possa ser comunicada sobre a geração de uma nova versão.

A Figura 3.5 apresenta o processo de geração de versão e execução do MEVID. Inicialmente, de posse da versão gerada e da localização do RDS, o gancho efetua uma chamada ao MEVID, informando a versão e a localização do repositório. Ao receber essa notificação, o MEVID verifica as alterações ocorridas na versão. No *Subversion*, essa informação é provida pela funcionalidade conhecida como *log* e contém informações sobre quais documentos foram adicionados, alterados ou removidos do repositório. Para cada documento referenciado no *log*, verifica-se se o mesmo é um documento semântico. Essa análise é feita verificando se o documento tratado (alterado, adicionado ou removido) possui um campo escondido (um *hidden field* definido na especificação ODF) com a seguinte chave e valor: “*SemanticDocument*”, “*true*”. Se sim, o procedimento continua, sendo iniciado o processo de extração e versionamento de dados.

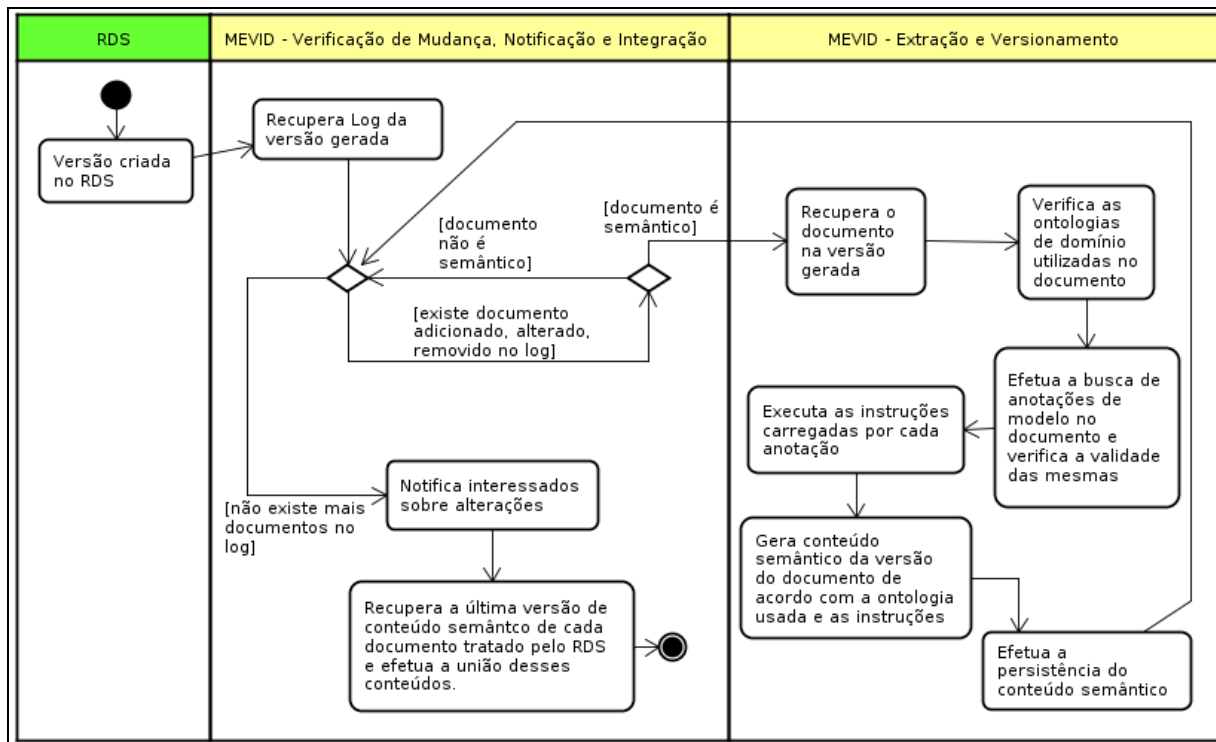


Figura 3.5 - Processo de geração de versão e execução do MEVID

O processo de extração se inicia pela análise de cada documento semântico referenciado no *log*. Há três situações possíveis: (i) um novo documento semântico é adicionado no repositório e, portanto, tem sua primeira versão criada no RDS; (ii) um documento semântico previamente tratado pelo MEVID é alterado no RDS; e (iii) um documento semântico previamente tratado pelo MEVID é excluído do RDS.

Nas duas primeiras situações o procedimento executado é similar (veja as figuras 3.6 e 3.7 para uma representação gráfica dos procedimentos de adição e de alteração de versão de documento semântico, respectivamente). Inicialmente, o MEVID recupera a versão do documento semântico (seja ele adicionado ou alterado). Em seguida as ontologias de domínio referenciadas no documento semântico (DS) são verificadas, ou seja, há uma tentativa de acesso à definição da mesma na *web*. Após a verificação, um mecanismo de extração faz a leitura das anotações contidas no documento semântico, verificando a validade das mesmas segundo a linguagem definida para anotações semânticas de modelos de documento.

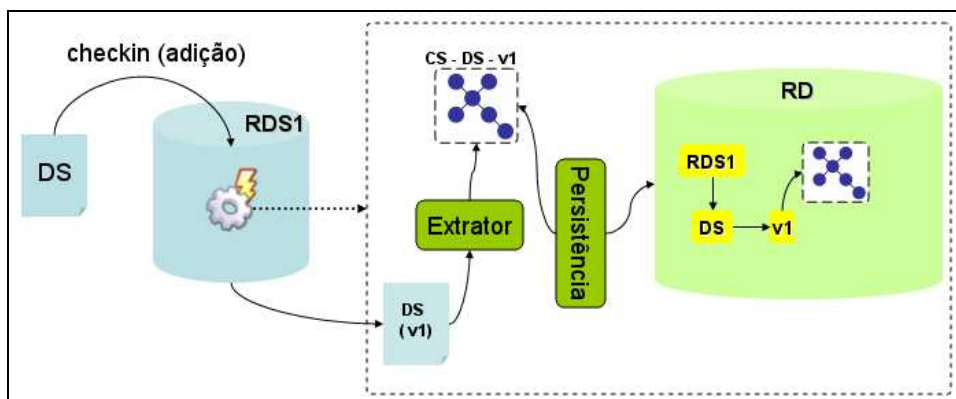


Figura 3.6 - Adição de Documento Semântico na Plataforma (MEVID)

Após a leitura das anotações de modelo, as mesmas são usadas para guiar a criação do conteúdo semântico (CS) do documento semântico, gerando, portanto, uma versão do conteúdo semântico do documento recuperado, como ilustra a Figura 3.6. Em seguida, o mecanismo de persistência de MEVID verifica se o RDS já está registrado no seu repositório de dados (RD) e em seguida checa o documento semântico e sua versão, registrando-os caso necessário. Finalmente o mecanismo de persistência registra o conteúdo semântico juntamente com a versão do documento semântico registrado.

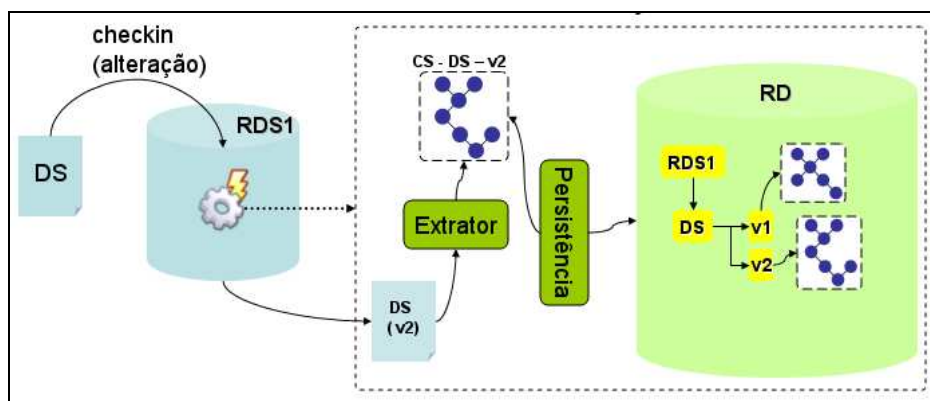


Figura 3.7 - Alteração de Documento Semântico na Plataforma (MEVID)

Ao adicionar uma nova versão de um documento semântico existente, o processo é similar, como mostra a Figura 3.7. A diferença é que o conteúdo semântico da correspondente versão de documento semântico é adicionada ao RD, mantendo, portanto, o histórico de alterações nos conteúdos semânticos de um mesmo documento em várias versões.

No caso de remoção de um documento semântico do RDS, o MEVID simplesmente faz a criação de uma versão do documento removido, sinalizando que o mesmo encontra-se excluído naquela versão e define um conteúdo semântico vazio para o mesmo, como ilustra a Figura 3.8.

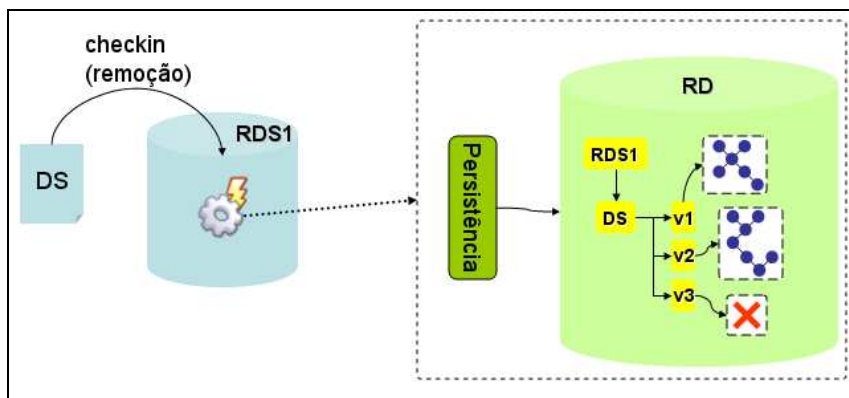


Figura 3.8 - Remoção de um Documento Semântico na Plataforma (MEVID)

Após a criação do conteúdo semântico de um documento semântico, inicia-se um processo adicional de identificação de diferenças entre a versão atual e a versão anterior do mesmo. Essa diferença aponta quais indivíduos foram adicionados, removidos ou alterados. Além disso, se houver alteração de algum indivíduo contido no conteúdo semântico da versão em relação ao conteúdo semântico da versão anterior, todas as alterações (valores de propriedades) são levantados, permitindo um detalhamento fino sobre as reais alterações ocorridas entre duas versões de um mesmo documento semântico. A Figura 3.9 ilustra a estratégia adotada para a geração da diferença entre duas versões

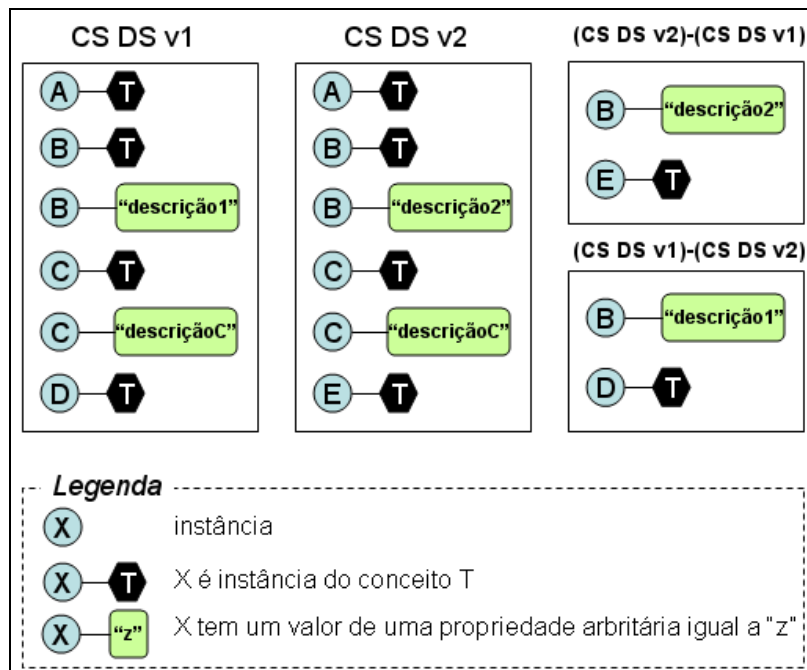


Figura 3.9 - Diferença no conteúdo semântico de duas versões de um mesmo DS (MEVID)

De posse do conteúdo semântico (CS) de um documento semântico em versões consecutivas (versão 1 - v1 e versão 2 - v2), dois conjuntos de diferenças são gerados: o conjunto diferença da versão atual para a versão anterior ($CS\ DS\ v2 - CS\ DS\ v1$) e o conjunto diferença da versão anterior para a versão atual ($CS\ DS\ v1 - CS\ DS\ v2$). Para efeito de simplicidade, considere os nomes D2-D1 e D1-D2 os conjuntos diferença da versão atual para a anterior e da versão anterior para a atual, respectivamente. A busca por indivíduos adicionados consiste em pesquisar qualquer indivíduo com definição de tipo (na figura representados por um círculo ligado a um hexágono) no conjunto D2-D1. Já indivíduos removidos na versão atual podem ser buscados da mesma forma, mas com a aplicação da mesma pesquisa sobre o conjunto D1-D2. No exemplo apresentado na Figura 3.9 o indivíduo "E" foi adicionado, enquanto o indivíduo "D" foi excluído.

A estratégia para análise de alterações de um indivíduo existente em ambas as versões de um documento semântico é a seguinte: primeiro, procura-se por indivíduos sem definição de tipo no conjunto D2-D1; para cada indivíduo encontrado nesse conjunto, listam-se as propriedades modificadas (essencialmente cada propriedade relacionada ao indivíduo no conjunto D2-D1); finalmente, para cada propriedade, é efetuada uma busca de seus valores no conteúdo semântico

da versão anterior (na figura, o conjunto CS DS v1) e uma busca dos valores da mesma propriedade no conteúdo semântico da versão atual (na figura, o conjunto CS DS v2).

No caso da primeira adição de um DS no RDS (claramente um DS pode ser removido e, posteriormente, re-adicionado ao mesmo RDS), a estratégia é sutilmente diferente: a aplicação da operação de diferença é feita unicamente sobre o conteúdo semântico da versão atual do DS e um conjunto vazio, buscando, apenas, os indivíduos com definição de tipo (indivíduos adicionados). A exclusão de um DS acarreta uma operação similar, embora a aplicação da operação de diferença seja feita sobre o conteúdo semântico da versão anterior e um conjunto vazio (representado a exclusão da versão atual do DS). Posteriormente são levantados todos indivíduos com definição de tipo encontrados no conjunto resultante, que representam aqueles que foram excluídos com a geração da nova versão do DS.

Ao final dessa fase de identificação de diferenças, o mecanismo de persistência do MEVID registra, para cada versão de documento semântico manipulado, as alterações ocorridas naquela versão. Dessa forma mantém-se o histórico de alterações ocorridas durante o ciclo de vida de um documento semântico.

Concluída a leitura do *log* gerado na criação da versão no RDS, MEVID busca as pessoas interessadas nos indivíduos alterados nessa versão. Essas pessoas devem estar previamente cadastradas na plataforma para recebimento de notificação de alteração de algum indivíduo. De posse da lista de interessados e das alterações referentes aos indivíduos de interesse, uma mensagem é enviada para os interessados, notificando-os sobre as mudanças ocorridas em cada indivíduo de interesse.

Finalmente, após a notificação dos interessados, o MEVID faz a busca das últimas versões dos conteúdos semânticos contidos no RD referentes ao RDS específico e, por meio de uma operação sequencial de união de conjuntos, realiza a integração dos mesmos.

A Figura 3.10 ilustra o processo de integração de dados, onde o conteúdo semântico da versão N (representado pela caixa nomeada CSR-N) é o resultado da união das últimas versões de conteúdos semânticos extraídos a partir dos documentos semânticos encontrados em um RDS. Ao final da integração, um mecanismo de raciocínio (conhecido também como máquina de inferência - *reasoner*) é executado sobre o grafo gerado. Se alguma ontologia utilizada contiver axiomas ou regras implementados, os mesmos serão executados nesse momento por intermédio do mecanismo de raciocínio, produzindo mais fatos baseados nas definições da ontologia,

enriquecendo ainda mais o grafo final. Tem-se, então, o conteúdo semântico do repositório (CSR) e, similarmente aos documentos semânticos registrados, esse conteúdo do repositório é também versionado.

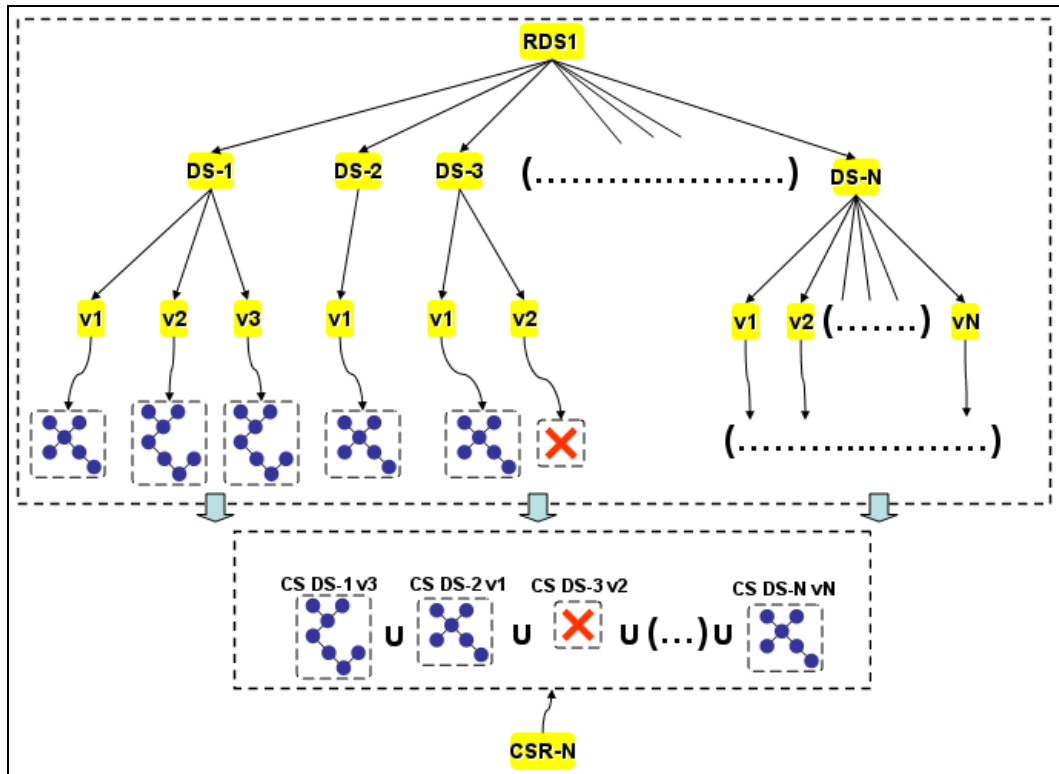


Figura 3.10 - Integração de Dados (MEVID)

Em resumo, o MEVID trabalha para que, a cada *checkin* bem sucedido efetuado em um RDS, os documentos semânticos tratados nesse *checkin* tenham seus conteúdos semânticos extraídos e persistidos, versionando o conteúdo semântico de cada documento semântico desse RDS. Adicionalmente, as alterações são persistidas e interessados em indivíduos envolvidos nessas alterações são notificados. Por fim, as últimas versões dos conteúdos semânticos de cada documento são utilizadas para a geração de um conteúdo semântico do repositório completo, provendo informações adicionais a respeito dos dados do repositório.

A seguir, é apresentado o Módulo de Busca e Rastreabilidade, que faz uso direto do repositório de dados mantido pelo MEVID.

3.5.3 - Módulo de Busca e Rastreabilidade (MBR)

O Módulo de Busca e Rastreabilidade (MBR) tem seu foco no oferecimento de serviços de acesso à plataforma para uso externo. Aplicações e usuários podem interagir com esse módulo para executar os seguintes serviços:

- Busca e rastreabilidade de dados de um repositório;
- Visualização da evolução de um indivíduo referenciado no repositório de dados;
- Visualização do histórico de alterações de um documento semântico; e
- Assinatura para notificação de alteração sobre um dado indivíduo.

As buscas são iniciadas com a escolha de um RDS. De posse da localização desse RDS e de uma versão de seu conteúdo semântico (CSR), o MBR permite a aplicação de pesquisas usando uma linguagem especializada em buscas sobre grafos RDF, a linguagem SPARQL (PRUD'HOMMEAUX; SEABORNE, 2008). A Figura 3.11 apresenta um cenário de busca no conteúdo semântico usando o MBR.

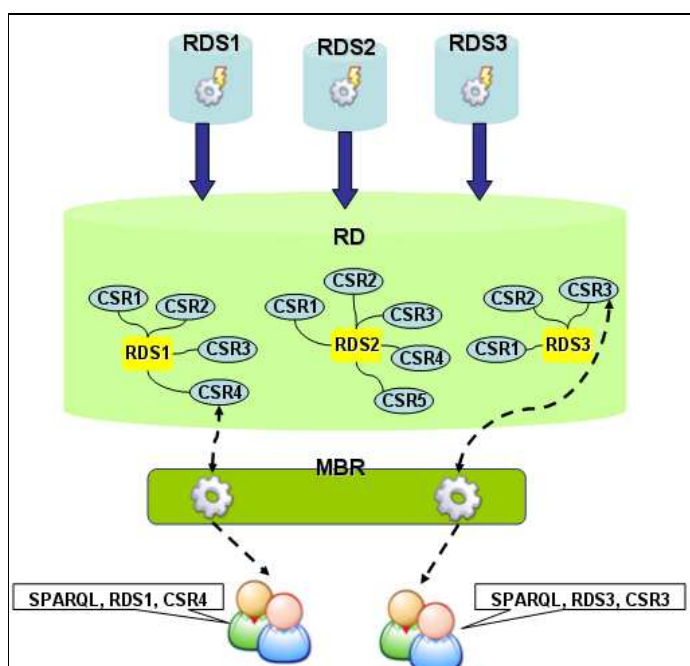


Figura 3.11 – Cenário de busca usando o MBR

Enquanto o MEVID garante a persistência e integração dos conteúdos semânticos provenientes de um RDS em um correspondente RD, usuários da plataforma podem usar o MBR

para efetuar pesquisas sobre os dados extraídos. No exemplo da Figura 3.11 a plataforma possui três RDSs (RDS1, RDS2 e RDS3), cada um deles devidamente registrado no RD juntamente com os conteúdos semânticos de suas versões (CSRn). Usuários definem a pesquisa a ser executada (SPARQL), o RDS e a sua versão a serem utilizados. O MBR, então, busca o conteúdo semântico correspondente (CSR) e aplica a pesquisa SPARQL definida pelo usuário, retornando os resultados. Como todo CSR registrado no RD possui uma referência para o conjunto de versões de documentos semânticos utilizados na geração do mesmo, esse conjunto é também apresentado, provendo, portanto, uma visão de alto nível da rastreabilidade dos dados. Por fim, com o resultado da consulta, o MBR possibilita, ainda, que o usuário selecione algum indivíduo do resultado e verifique em quais versões de documentos semânticos daquele RDS o indivíduo selecionado é referenciado.

Existem duas formas de analisar a evolução de um indivíduo: (i) listando, para um intervalo de versões escolhido e um documento semântico em que o indivíduo é referenciado, todos os conteúdos semânticos que envolvem o indivíduo em questão e levantando, para cada um deles, os relacionamentos registrados; (ii) listando, para um intervalo de versões escolhido, os CSRs que referenciam o indivíduo e efetuando uma pesquisa sobre cada um dos CSRs identificados. O termo “evolução” usado aqui se refere ao conjunto de diferentes relacionamentos que um dado indivíduo pode ter em diferentes versões registradas no RD. A Figura 3.12 apresenta um cenário relativo ao caso (ii).

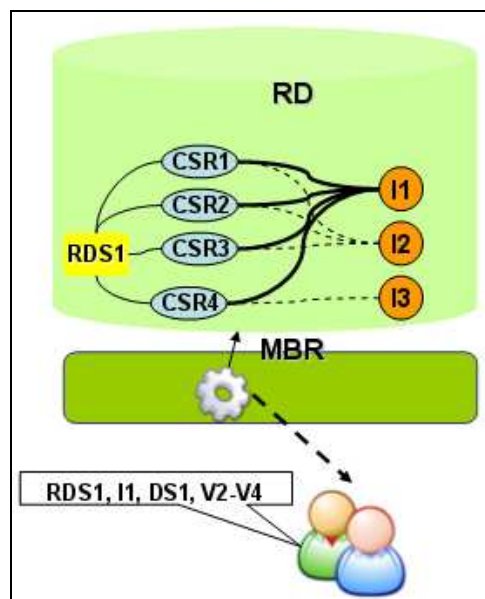


Figura 3.12 – Cenário de verificação da evolução de um indivíduo

Ao informar o RDS, o indivíduo, e a faixa de versões a serem analisadas (no balão da Figura 3.12, RDS1, I1 e V2-V4, respectivamente), o MBR faz uma busca dos CSRs dentro da faixa de versões requisitada (versões de 2 a 4 na figura) que possuem relação com o indivíduo informado (I1). Com os CSRs, o MBR efetua uma pesquisa em cada um deles, levantando os relacionamentos que o indivíduo possui e retorna esse resultado para o usuário.

A análise de evolução de um indivíduo em um documento semântico funciona de forma similar, com a diferença que as pesquisas são efetuadas sobre os conteúdos semânticos das versões do documento informado (DS1, na Figura 3.12).

A visualização da evolução de um documento semântico se dá a partir da recuperação das alterações realizadas em cada versão do documento. Toda vez que um documento semântico é modificado e registrado no RDS, o MEVID gera e registra as diferenças semânticas entre a versão gerada e a anterior. De posse do RDS onde se encontra o documento, é possível apresentar as modificações efetuadas no mesmo. Para se obter a visualização da evolução de um documento, o usuário informa o RDS, o documento semântico e a faixa de versões a serem analisadas. O MBR obtém os registros efetuados para cada versão contida na faixa de versões informada e apresenta o resultado ao usuário. A Figura 3.13 apresenta um cenário para esse caso. Nessa figura, um usuário usa o MBR para verificar as alterações existentes entre as versões 2 e 4 do

documento DS1. O MBR faz a busca das versões escolhidas desse documento e retorna as alterações registradas em cada versão (elementos A2, A3 e A4 da figura).

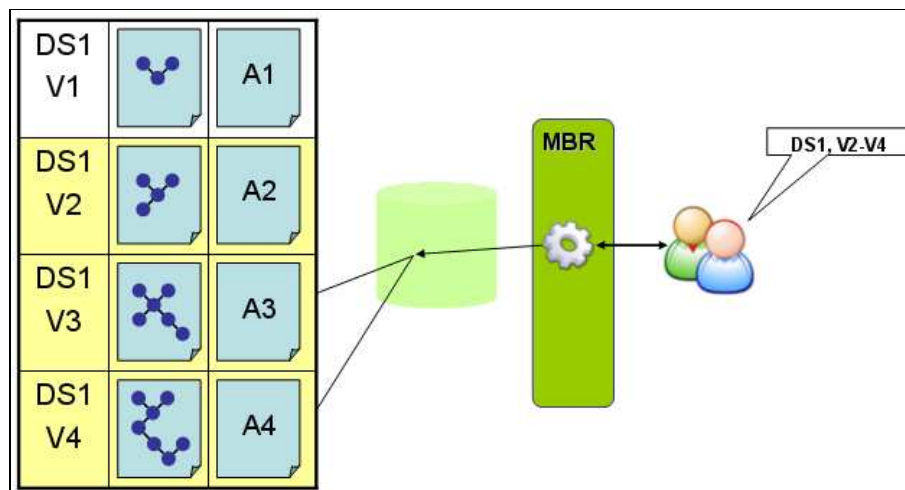


Figura 3.13 - Evolução de um documento semântico.

A última funcionalidade provida pelo MBR é a notificação de interessados. A partir do momento em que o assinante registra seu interesse por um indivíduo, qualquer alteração no RDS que referencie esse indivíduo gerará uma notificação ao interessado. Por ser uma forma pró-ativa de ciência sobre uma alteração em um dado indivíduo, essa característica pode apoiar a tomada de decisão das partes notificadas. Basicamente, o MBR provê um serviço de registro de interesse, no qual o usuário seleciona um RDS e um indivíduo contido nesses RDS e informa seu email.

3.6 – Projeto e Implementação da Plataforma de Gerenciamento de Documentos Semânticos

O projeto da arquitetura da PGDS foi conduzido segundo o padrão MVC (*Model-View-Controller*) (FOWLER, 2003) associado ao padrão Camada de Serviço (*Service Layer*) (FOWLER, 2003). O elemento *Model* – representado pelos pacotes com nome “*model*” – contempla as classes de domínio do problema (pacote “*domain*”) e regras de negócio (pacote “*service*”). O elemento *Controller* – representado pelos pacotes com nome “*controller*” – contempla as classes responsáveis por fazer a interface entre o modelo (*Model*) e a visão (elemento *View*). O elemento *View* – representado pelos pacotes com nome “*view*” – é composto pelas classes e arquivos de interface com o usuário. Outros padrões de projeto utilizados são o

padrão DAO (*Data Access Object*) e Injeção de Dependência (*Dependency Injection* ou *IoC - Inversion of Control*). O Apêndice A apresenta o projeto da PGDS na íntegra.

Para implementar a PGDS, as tecnologias listadas na Tabela 3.1 foram utilizadas. A Figura 3.14 apresenta a visão geral da Plataforma de Gerenciamento de Documentos Semânticos, destacando em que elementos (círculos em preto nomeados de A até J) as tecnologias foram aplicadas. A Tabela 3.2 relaciona os elementos da arquitetura às tecnologias empregadas.

Tabela 3.1 - Relação das Tecnologias utilizadas

Tecnologia	Descrição
Apache Tomcat	Servidor HTTP Java. Utilizado para comportar a plataforma como um todo. Foi utilizada a versão 6.0.26.
Linux curl	Ferramenta de linha de comando utilizada para efetuar requisições de rede de computadores. Foi utilizada a versão 7.18.2
Hibernate	<i>Framework</i> de mapeamento objeto-relacional (implementa JPA). Foi utilizada a versão 3.4.0GA
JavaMail	<i>Framework</i> para envio de <i>email</i> provido pela tecnologia Java EE.
Jena	<i>Framework</i> que oferece um ambiente de manipulação de grafos RDF, RDFS e OWL, além de permitir a execução de pesquisas usando a linguagem SPARQL. Foi utilizada a versão 2.6.2
JPA	A <i>Java Persistence API</i> (JPA) define padrões de <i>frameworks</i> de mapeamento objeto-relacional. Foi utilizada a versão 1.0
ODF	Especificação de Formato de documento. Define o formato Open Document Text utilizado nesse trabalho.
ODFDOM	<i>Framework</i> que provê uma forma facilitada de criar, acessar e manipular arquivos ODF. Foi utilizada a versão 0.7.5.
OpenOffice Writer	Ferramenta de Edição de Documentos de Texto. Foi utilizada a versão 3.2.
OWL	OWL (Ontology Web Language) é uma linguagem baseada em RDF (Resource Description Framework) usada para implementação de ontologias e modelos conceituais.
Pellet Reasoner	Máquina de inferência utilizada para inferir novos fatos baseados em um grafo instanciado a partir de uma ontologia.
PotgreSQL	Sistema de Gerenciamento de Bancos de Dados Relacionais utilizado para gerenciar os repositórios de dados (RDs), que essencialmente são bancos de dados relacionais.
SPARQL	Linguagem de pesquisa em grafos de triplas (padrão W3C).

Tabela 3.1 - Relação das Tecnologias utilizadas (Continuação)

Spring IoC	<i>Framework</i> de Injeção de Dependências. Utilizado para resolver as dependências entre as classes de PGDS. Foi utilizada a versão 3.0.2.
Spring MVC	<i>Framework</i> que permite o uso do padrão MVC em aplicações web. Foi utilizada a versão 3.0.2.
Subversion	Sistema de Controle de Versão. Foi utilizada a versão 1.5.4.
SVNKit	<i>Framework</i> que comporta funcionalidades referentes ao sistema de controle de versões <i>Subversion</i> . Foi utilizada a versão 1.3.1.
Xerces	<i>Framework</i> de pesquisa em XML (implementa a linguagem <i>XPath</i>). Foi utilizada a versão 2.9.1.
XPath	Linguagem de pesquisa em XML (padrão W3C).

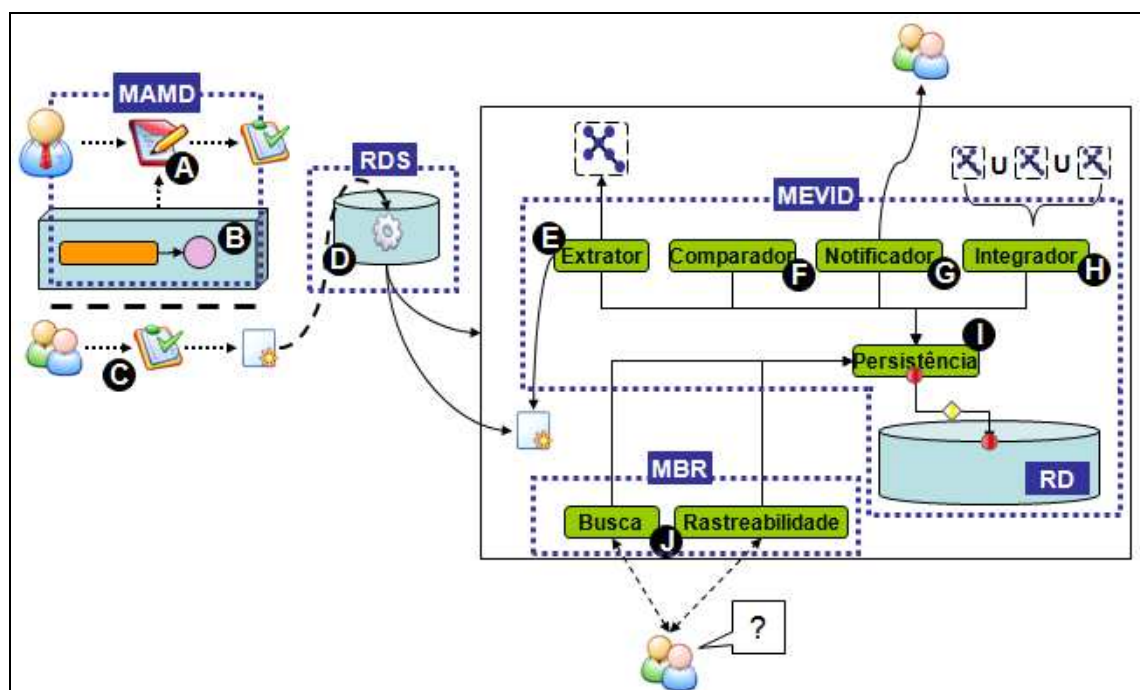


Figura 3.14 – Visão da Arquitetura Geral da PGDS (pontuada por elemento de arquitetura)

Tabela 3.2 - Relação de Pontos Arquiteturais e Tecnologias utilizadas

Ponto	Descrição	Tecnologias
A	Usando o OpenOffice e os artifícios definidos no MAMD, o engenheiro de documento faz anotações semânticas em modelos de documento.	OpenOffice Writer, ODF
B	A linguagem definida para anotação semântica de modelos de documento faz uso dos elementos definidos na especificação ODF. As Ontologias de Domínio utilizadas nas anotações são implementadas em OWL.	ODF, OWL
C	Desenvolvedores fazem uso dos modelos semânticos e geram documentos enriquecidos semanticamente. A edição dos documentos com o uso de modelos semânticos é processo convencional e o desenvolvedor deve utilizar a ferramenta OpenOffice Writer.	OpenOffice Writer
D	Uma vez que o desenvolvedor edita o documento, ele pode fazer o <i>checkin</i> do mesmo no repositório de controle de versões. Esse repositório assume o papel de Repositório de Documentos Semânticos (RDS) da arquitetura geral da plataforma. O sistema de controle de versões utilizados foi o <i>Subversion</i> . Assim que o <i>checkin</i> é efetuado, um gancho do tipo <i>post-commit</i> faz uma requisição HTTP ao servidor da plataforma via curl, notificando a geração de uma nova revisão do repositório. Os dados passados são: a revisão gerada pelo Subversion e o local do RDS. O gancho é implementado usando a linguagem definida pelo interpretador <i>bash</i> ¹² do sistema operacional Linux.	Subversion, Linux curl.
E	Uma instância do servidor web Apache Tomcat comporta a aplicação principal da plataforma que conta com os módulos MEVID e MBR. Um controlador <i>web</i> , definido usando o <i>framework</i> Spring MVC, recebe a requisição HTTP feita pelo gancho instalado no RDS. A requisição é então repassada para um interpretador, que faz a leitura do <i>log</i> gerado pela versão. A resolução das dependências existentes entre o controlador e o interpretador é feita com ajuda do Spring IoC. Para executar tal tarefa, o <i>framework</i> SVNKit foi utilizado. A funcionalidade de extração é dada em dois passos principais: (i) captura do documento semântico e (ii) extração do conteúdo semântico. O primeiro usa o SVNKit para obter o documento semântico em questão. O segundo faz a leitura do documento, que deve estar no formato ODT (<i>Open Document Text</i> , da especificação ODF), usando o <i>framework</i> ODFDOM, especializado na leitura do esquema XML provido pela especificação ODF. Dessa forma é possível fazer a leitura das anotações semânticas contidas no documento. A cada anotação encontrada a mesma é interpretada, transformando-a em código RDF fazendo o uso do <i>framework</i> Jena. No final do processo, de cada documento semântico, tem-se o conteúdo semântico do mesmo, que é essencialmente o grafo RDF resultante do processo de interpretação das anotações encontradas.	Apache Tomcat, Spring MVC, Spring IoC, SVNKit, ODFDOM, Jena
F	Para efetuar a diferença entre grafos, necessária para gerar o conjunto de diferenças entre duas versões de um documento semântico conforme discutido na seção 3.5.2, e para a busca de informações em RDF foram utilizados o <i>framework</i> Jena e o XPath (implementado pela biblioteca Xerces), respectivamente.	Jena, XPath, Xerces

¹² Bash, Wikipedia, <http://pt.wikipedia.org/wiki/Bash>, visitado em 17/07/2010.

Tabela 3.2 - Relação de Pontos Arquiteturais e Tecnologias utilizadas (Continuação)

G	A notificação das partes interessadas nas modificações ocorridas com a geração de uma versão no RDS é feita através do envio de emails para os interessados. Nesse caso foi utilizado o Spring MVC que faz uso da tecnologia JavaMail para configuração e envio dos mesmos.	JavaMail, Spring MVC
H	Ao final da leitura do <i>log</i> , vários conteúdos semânticos são unidos com o uso do <i>framework</i> Jena. Após essa operação, a máquina de raciocínio Pellet Reasoner faz a análise do grafo final, podendo gerar mais informações, em função das ontologias envolvidas.	Jena, Pellet Reasoner
I	As operações de persistência de dados (registro, remoção e busca) e os mapeamentos objeto-relacional usados são providos por uma camada de persistência desenvolvida usando a API de persistência Java JPA (Java Persistence API), implementada pelo <i>framework</i> Hibernate. O banco de dados relacional Postgresql é usado para a implementação de Repositório de Dados (RD).	JPA, Hibernate
J	O MBR conta com o Spring MVC para comportar os controladores que atendem as requisições de usuários ou outros sistemas interessados em executar buscas. Quando uma requisição de busca é executada, a versão do conteúdo semântico, o local do RDS e uma string com o conteúdo da busca escrito com a linguagem SPARQL são informados. De posse dessas informações o MBR obtém o conteúdo semântico inferido, previamente persistido no RD, e aplica a busca informada sobre o grafo, usando o <i>framework</i> Jena para tal, retornando os resultados levantados.	Spring MVC, Jena, SPARQL

3.7 - Comparação com Trabalhos Relacionados

Os trabalhos apresentados no Capítulo 2 foram de grande importância para a concepção da PGDS. Alguns deles guiaram algumas das escolhas feitas aqui. Assim, é importante comparar o presente trabalho com alguns trabalhos relacionados.

A arquitetura de gerenciamento de documentos semânticos proposta em (ERIKSSON, 2007a e 2007b; ERISSON; BANG, 2006) e o trabalho relativo à documentação semântica apoiada por tecnologias da *web* semântica de KIM et al. (2006) delineiam aspectos comuns, tais como: (i) uso de um repositório de documentos semânticos; (ii) uso de ferramentas de anotação semântica e (iii) buscas semânticas. De forma similar, a PGDS permite buscas com rastreabilidade para documentos. Adicionalmente a PGDS combina a ideia de manter modelos de documentos semânticos, proposta feita inicialmente por Tallis (2003), e a ideia de manter o histórico evolutivo dos dados extraídos. Além disso, para prover visibilidade nas modificações ocorridas durante a evolução de um indivíduo, foi elaborado um mecanismo de notificação da ocorrência de mudanças.

O trabalho conduzido por Tallis (2003) permite a construção de modelos de documento semânticos, usando uma ferramenta de anotação semântica. Uma diferença em relação ao trabalho conduzido nesta dissertação é que Tallis elaborou seu trabalho com base em um formato proprietário (*Microsoft Word doc*), dependendo desse formato para criação de documentos semânticos. Adicionalmente, seu trabalho não cobre a gerência de documentos semânticos como um todo, tratando basicamente de anotação semântica e extração automática de informação. PGDS compartilha da ideia de automatizar o processo de anotação semântica fazendo uso de modelos de documento semânticos, mas adiciona a gerência dos documentos gerados a partir desses modelos. Além disso, mantém o registro evolutivo dos dados extraídos. Em relação ao formato utilizado, PGDS usa o formato aberto ODF, que gradualmente vêm sendo usado em várias aplicações governamentais¹³.

O *framework Molhado* é especializado em diminuir a diferença de impedância existente entre arquivos versionados contemplados por um sistema de controle de versão e os objetos reais que aquele arquivo tenta representar. A maior diferença entre *Molhado* e o trabalho apresentado nesta dissertação é que *Molhado* faz uma especialização de um sistema de controle de versão, provendo capacidades semânticas ao mesmo. PGDS, por sua vez, provê maneiras de enriquecer documentos semanticamente a partir do uso de modelos de documento semânticos, extrai seus dados e controla a versão dos dados extraídos. Adicionalmente, não há intuito algum de modificar a arquitetura dos sistemas de controle de versão, já que os mesmos são um padrão *de-facto*.

Sefton et al. (2009) descreveram algumas técnicas de anotação semântica usadas no contexto do ambiente ICE (*Integrated Content Environment*¹⁴), que visa primariamente permitir que usuários possam criar páginas *web* ou outros documentos (apresentações, por exemplo) a partir de documentos de texto. Para tal, os autores fazem uso de ferramentas de edição de texto *desktop*, mais especificamente *Microsoft Word* e *OpenOffice*. A elaboração das anotações semânticas definidas pelos autores se baseia no uso de estilos de formatação de texto e no uso de tabelas. Enquanto Sefton et al. (2009) consideram anotações semânticas em documentos, o

¹³ ODF Annual Report 2008, <http://www.odfalliance.org/resources/Annual-Report-ODF-2008.pdf>, visitado em 13/07/2010.

¹⁴ ICE: The integrated Content Environment, <http://ice.usq.edu.au/introduction/about.htm>, visitado em 13/07/2010.

presente trabalho considera anotações semânticas em modelos de documentos, com o intuito de facilitar o processo de anotação semântica. Para tal, o desenvolvimento de uma linguagem especializada em anotação de modelos de documento foi elaborada, mantendo, portanto, um diferencial do trabalho no que se refere ao alvo da anotação. Entretanto, o processo de anotação dos modelos de documento faz uso de estilos de formatação de texto e tabelas, analogamente a (SEFTON et al., 2009).

Nesic et al. (2008) propõem um sistema de gerenciamento de documentos semânticos focado na gerência de objetos de aprendizado. O objetivo do sistema é permitir que usuários possam compartilhar e reusar objetos de aprendizado registrados. O sistema de gerenciamento permite, ainda, que os usuários efetuem buscas baseadas em ontologias, tal como os trabalhos de Eriksson (2007a e 2007b) e Kim et al. (2006). O diferencial em relação ao trabalho desta dissertação é o uso de modelos de documentos semânticos para prover anotação automatizada, o acompanhamento evolutivo do conteúdo semântico dos documentos e o mecanismo de notificação sob alteração.

3.8 - Conclusões do Capítulo

Neste capítulo foi apresentada a Plataforma para Gerenciamento de Documentos Semânticos desenvolvida neste trabalho. Em resumo, a plataforma contempla: (i) um módulo (MAMD) voltado para a criação de modelos de documento semânticos (modelos semânticos), que faz uso de anotações semânticas e ontologias de domínio; (ii) um módulo (MEVID) que permite a extração, versionamento, persistência e integração de dados provenientes dos documentos semânticos, além de prover um serviço de notificação aos interessados sobre modificações ocorridas entre versões; (iii) um módulo (MBR) que oferece uma interface para consultas variadas no repositório de dados da plataforma, além de permitir o acompanhamento da evolução dos mesmos.

Os detalhes referentes às tecnologias e padrões utilizados, bem como a materialização da arquitetura foram também discutidos (vide Apêndice A). O próximo capítulo apresenta o uso da plataforma para apoiar algumas atividades do processo de Engenharia de Requisitos.

Capítulo 4

Uso da Plataforma de Gerenciamento de Documentos Semânticos no Apoio à Gerência de Requisitos

Este capítulo apresenta o uso da Plataforma de Gerenciamento de Documentos Semânticos no apoio à Gerência de Requisitos de Software. Inicialmente, na seção 4.1, a motivação para o uso da plataforma é conduzida. Na seção 4.2, um cenário de uso da plataforma no apoio ao processo de engenharia de requisitos de uma organização de software é apresentado. A seção 4.3 apresenta brevemente as ontologias utilizadas e sua implementação. A seção 4.4 trata das anotações semânticas em modelos de documento. A seção 4.5 discute a execução e a evolução de um projeto de software usando a plataforma. Finalmente, a seção 4.6 apresenta as considerações finais do capítulo.

4.1 - Introdução

Na Engenharia de Software, um ramo de extrema importância é a Engenharia de Requisitos. Segundo NARDI (2006), a Engenharia de Requisitos visa tratar os requisitos de um software a ser construído, fazendo uso de um processo para tal e, adicionalmente, lidando com outras áreas de conhecimento com o intuito de executar a tarefa de engenhar requisitos.

O Processo de Engenharia de Requisitos envolve várias atividades, dentre elas: Levantamento, Análise e Negociação, Documentação, Validação e Gerência de Requisitos (KOTONYA et al., 1998). Em especial as atividades de Documentação de Requisitos e Gerência de Requisitos possuem alguns obstáculos que podem ser mais facilmente superados com o uso da Plataforma de Gerenciamento de Documentos Semânticos (PGDS) proposta.

A atividade de Documentação de Requisitos visa apresentar os resultados do processo de Engenharia de Requisitos em um documento que lista os requisitos do software a ser construído. Os requisitos contemplados nesse documento devem contar com um identificador único e uma descrição (IEEE 830-1998, 1998). Nesse contexto, um dos artefatos tipicamente produzidos na realização da atividade de Documentação de Requisitos é o Documento de Requisitos. O Documento de Requisitos visa, portanto, detalhar os requisitos levantados e acordados sobre o sistema. Segundo a norma IEEE 830-1998 (1998), um Documento de Requisitos deve obedecer a uma série de características, dentre elas a rastreabilidade, que demanda que esse documento

possua uma forma clara de se alcançar a origem de cada um de seus requisitos descritos, facilitando a referência dos mesmos no desenvolvimento futuro ou em documentos de apoio. Para se estruturar o conteúdo de um Documento de Requisitos de forma adequada, é necessário que a organização defina seus modelos de documentos de requisitos (NARDI, 2006).

A atividade de Gerência de Requisitos visa gerenciar de forma consistente e controlada as mudanças efetuadas sobre os requisitos contemplados durante o processo de Engenharia de Requisitos, já que os mesmos são passíveis de alteração durante o ciclo de vida de um projeto de software. Quando surge a necessidade de alteração de um requisito, deve ser feito o levantamento das dependências do requisito modificado. Essas dependências incluem outros requisitos do sistema e também outros artefatos que fazem referência ao requisito a ser modificado. Dessa forma, há a necessidade de se manter a rastreabilidade dos requisitos existentes em um projeto para outros artefatos (WIEGERS, 2003) (ROBERTSON et al., 1999) (KOTONYA et al., 1998)

Levando em conta que a rastreabilidade é considerada difícil, por envolver muitas informações espalhadas pelos artefatos relativos a um projeto (LOPES 2002), o uso de ferramentas de apoio à rastreabilidade é considerado um fator essencial para uma boa execução dessa tarefa (WIEGERS, 2003). Nesse contexto, o uso da PGDS para criar modelos de documentos semânticos que façam a captura tanto dos artefatos de um projeto de software quanto das relações existentes entre os mesmos pode trazer um valioso apoio à tarefa de rastreabilidade de requisitos. Adicionalmente, a visibilidade das informações acerca do projeto, bem como a possibilidade de acompanhar a evolução dessas informações, permite um melhor conhecimento sobre o projeto.

A seguir é colocado o cenário de aplicação da PGDS usado para avaliar a plataforma.

4.2 - Cenário de Aplicação

Seja o Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), uma organização que executa projetos de desenvolvimento de software. O NEMO faz o registro de seus artefatos em documentos, normalmente originados como produtos de trabalho de atividades dos processos definidos pelo mesmo. Os documentos gerados durante a execução de um projeto são centralizados em um diretório de arquivos, sendo que cada projeto conta com um diretório específico, onde são guardados os seus documentos. Apesar de registrar efetivamente seus artefatos em documentos e mantê-los nesse diretório, o NEMO não conta com ferramentas de

apoio e, portanto, encontra dificuldades no levantamento das informações acerca de um projeto. Em especial, as tarefas de rastreabilidade de requisitos e verificação da evolução dos artefatos de um projeto dependem do levantamento de informações espalhadas por uma série de versões de documentos, contidas no diretório de arquivos do projeto. Um tempo precioso é, portanto, despedido na leitura dos documentos e investigação dos relacionamentos entre as informações contidas nos mesmos.

Visando padronizar e melhorar o alcance às informações sobre um projeto de software, o NEMO possui vários modelos de documento, sendo dois deles modelos relativos a requisitos: Modelo de Documento de Requisitos e Modelo de Documento de Descrição de Caso de Uso.

O Modelo de Documento de Requisitos inclui com as seguintes informações: (i) nome do projeto; (ii) nome dos responsáveis pela elaboração do documento; (iii) descrição do escopo do projeto; (iv) lista dos requisitos funcionais e (v) lista dos requisitos não funcionais do projeto, juntamente com a descrição de cada um deles. Já o Modelo de Descrição de Caso de Uso conta com as seguintes informações: (i) nome do projeto; (ii) nome do caso de uso sendo descrito; (iii) nome do subsistema ao qual o caso de uso em questão está alocado; (iv) descrição sucinta do caso de uso; (v) cursos de eventos (cursos básicos e cursos alternativos) do caso de uso; (vi) requisitos atendidos pelo caso de uso em questão. Esses modelos estão apresentados no Anexo A desta dissertação.

O NEMO faz uso do sistema de controle de versões *Subversion*. Dessa forma o primeiro elemento da arquitetura contemplada pela PGDS, o Repositório de Documentos Semânticos (RDS), já estava disponível. Foi necessário apenas definir o gancho (*hook*) utilizado na notificação de geração de novas versões no RDS. Como comentado no Capítulo 3, o gancho deve prover ao Módulo de Extração, Versionamento e Integração de Dados (MEVID) informações acerca de uma versão gerada no RDS, mais exatamente a versão propriamente dita, e da localização física do repositório. O código abaixo mostra o conteúdo do gancho instalado no RDS:

```
#!/bin/bash
curl "http://localhost:8080/pgds/remoting/versioncontrol/vc?rev=$1&url=$2"
```

onde os argumentos \$1 e \$2 são a versão gerada e a localização do RDS, respectivamente, ambos providos pelo próprio *Subversion*. Vale ressaltar que cada RDS considerado na plataforma deve

contemplar um gancho como o descrito acima para que o RDS possa se comunicar de forma adequada com a plataforma.

O NEMO possui os modelos de documentos citados anteriormente, mas os mesmos não estavam anotados semanticamente. Assim, é imprescindível a realização de anotações semânticas nesses modelos, de modo a capturar as informações desejadas acerca do projeto. Para tal, foram selecionadas as ontologias a serem utilizadas pelos engenheiros de documentos durante o uso do Módulo de Anotação em Modelos de Documento (MAMD) da plataforma para a criação dos modelos semânticos.

A partir dos modelos de documento anotados semanticamente, documentos de requisitos e documentos de descrição de casos de uso podem ser elaborados e registrados por meio de operações de *checkin* no RDS, iniciando o processo principal efetuado pelo MEVID. Durante a execução do projeto, os engenheiros de requisitos, podem usar o Módulo de Busca e Rastreabilidade (MBR) para apoiar a tarefa de rastreabilidade do projeto em questão. Adicionalmente, a equipe de garantia da qualidade da organização pode fazer uso do MBR para levantar possíveis pontos de não conformidade encontrados no projeto.

Na próxima seção são apresentadas as ontologias utilizadas e sua implementação em OWL. Essa implementação foi utilizada para elaborar consultas sobre o domínio das ontologias e para anotar os modelos de Documento de Requisitos e de Descrição de Casos de Uso do NEMO.

4.3 - Ontologias Utilizadas

O uso de ontologias é fundamental para o funcionamento da plataforma. Primeiramente, os conceitos, propriedades e relacionamentos definidos nas ontologias são utilizados como parte das anotações semânticas. Com as anotações definidas, o processo de extração do MEVID usa as mesmas para guiar a criação de instâncias de conceitos e relações da ontologia. Por fim, os axiomas definidos nas ontologias são utilizados para manter a integridade da ontologia e, adicionalmente, para inferir mais fatos sobre a mesma.

No caso do cenário de aplicação apresentado na seção anterior, há a necessidade de se considerar ontologias que tratem do domínio da Engenharia de Requisitos. Conceitos como projetos, pessoas, requisitos funcionais e não-funcionais, casos de uso, fluxo de eventos de caso de uso e módulos são relevantes. Extratos de duas ontologias foram selecionados: Ontologia de Organizações de Software (BARCELLOS; FALBO, 2009) e Ontologia de Requisitos de

Software (FALBO; NARDI, 2008). A subseção 4.3.1 apresenta o fragmento da ontologia de organizações de software considerado, enquanto a subseção 4.3.2 apresenta o fragmento da ontologia de requisitos considerado. Entretanto, há aspectos importantes relativos a modelos de casos de uso que não são contemplados por essas ontologias, assim este último fragmento foi estendido para incorporar novos conceitos. Finalmente, a subseção 4.3.3 discute a implementação das ontologias.

4.3.1 - Ontologia de Organizações de Software

A Figura 4.1 mostra o fragmento da Ontologia de Organizações de Software (BARCELLOS; FALBO, 2009) considerado.

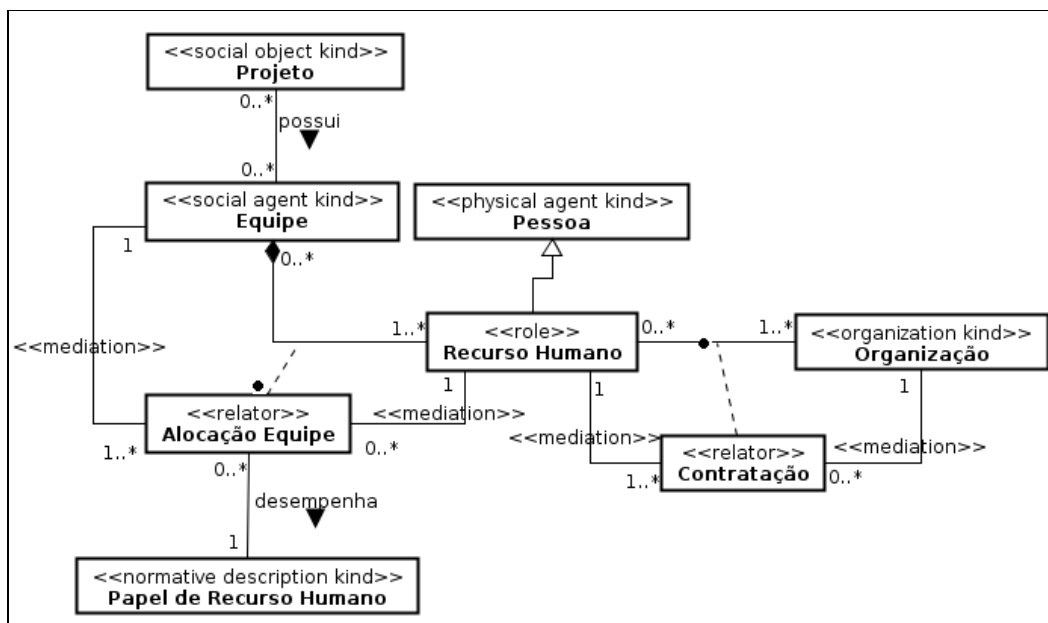


Figura 4.1 - Seção da Ontologia de Organização, adaptado de (BARCELLOS; FALBO, 2009)

De acordo com esse fragmento da ontologia, um *Projeto* pode possuir uma ou mais *Equipes*, que são compostas por um conjunto de *Recursos Humanos*. Um *Recurso Humano* é o papel desempenhado por uma *Pessoa* quando empregada em uma *Organização*. Cada recurso humano pertencente a uma equipe pode ter sido alocado à mesma (*Alocação Equipe*) para desempenhar um papel específico (*Papel de Recurso Humano*).

Com base no fragmento da ontologia apresentado na Figura 4.1 e analisando o modelo de Documento de Requisitos do NEMO (Anexo A), pode-se observar que os conceitos *Projeto*, *Pessoa* e *Recurso Humano* estão presentes no modelo de documentos. Assim, esses conceitos

juntamente com seus relacionamentos diretos foram considerados na fase de projeto e implementação da ontologia, discutida na subseção 4.3.3.

4.3.2 - Ontologia de Requisitos de Software

A Figura 4.2 apresenta o fragmento da Ontologia de Requisitos de Software (FALBO; NARDI, 2008) considerado.

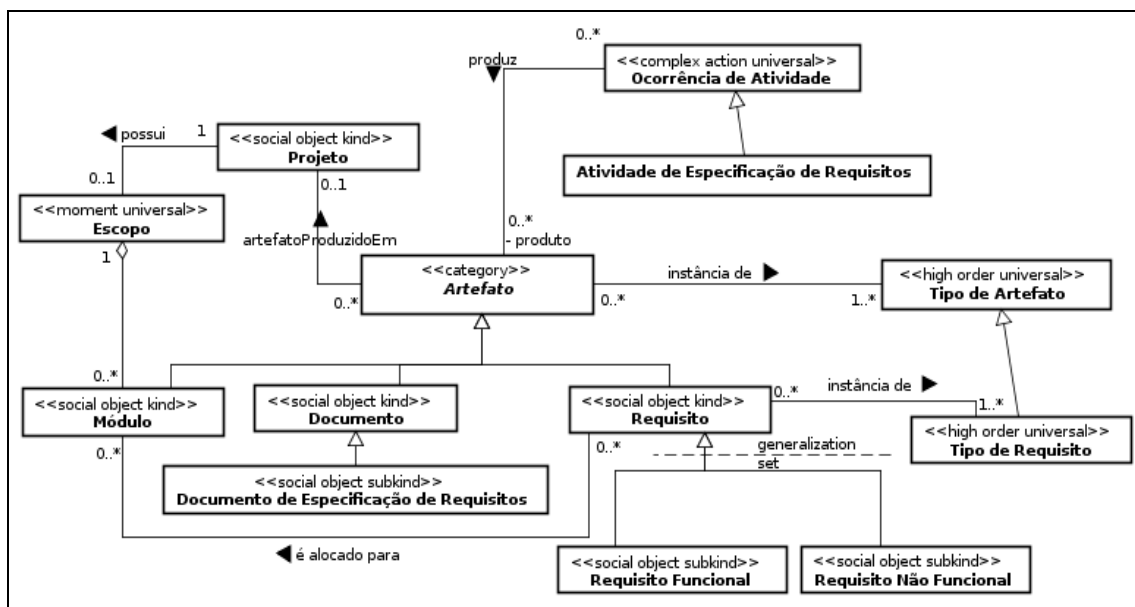


Figura 4.2 - Ontologia de Requisito, adaptado de (FALBO; NARDI, 2008)

Segundo esse fragmento da ontologia, um *Requisito de Software* é um *Artefato*, produzido em um *Projeto*. Requisitos podem ser, dentre outros, *Requisitos Funcionais* e *Requisitos Não Funcionais*. Cada requisito é alocado para um *Módulo* que faz parte do *Escopo* do *Projeto*. Dessa porção da ontologia, os seguintes conceitos, bem como as relações existentes entre eles, foram considerados na implementação da ontologia: *Artefato*, *Requisito*, *Requisito Funcional*, *Requisito Não Funcional*, *Escopo* e *Módulo*.

Um conceito estreitamente relacionado a requisitos é o conceito de casos de uso. Sommerville (2003) aponta casos de uso como sendo uma técnica baseada em cenários para obtenção de requisitos. Contudo a ontologia proposta em (FALBO; NARDI, 2008) não considera esse conceito. Assim, no contexto deste trabalho, foi realizada uma pequena extensão dessa

ontologia para considerar casos de uso. A Figura 4.3 apresenta essa extensão. As classes representadas em amarelo foram introduzidas na extensão da ontologia.

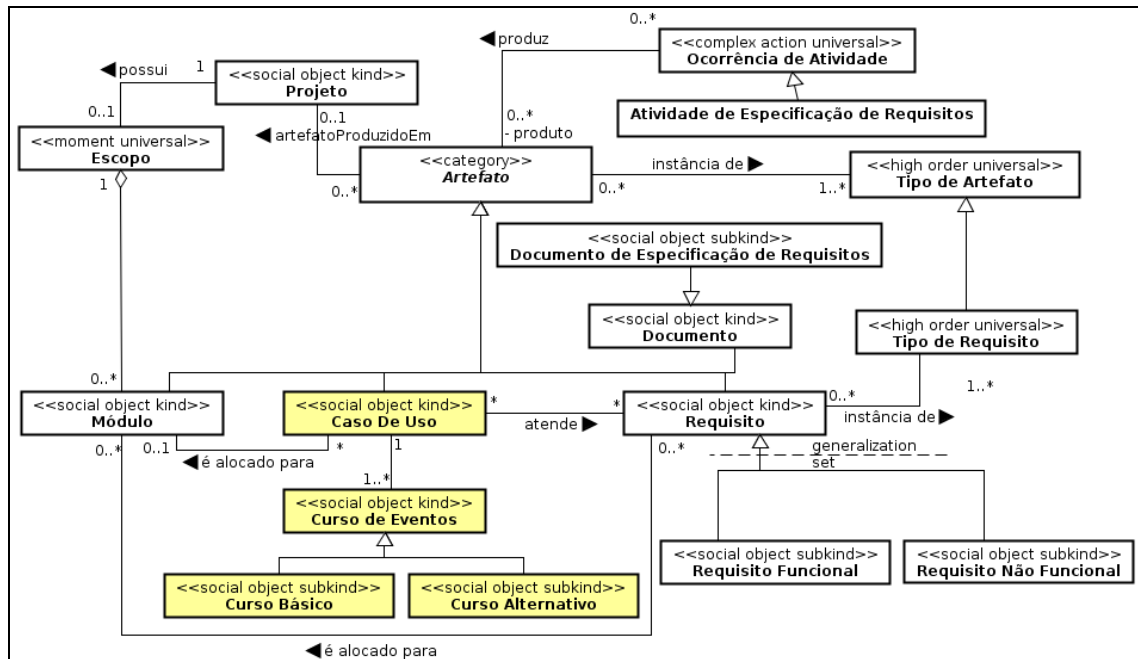


Figura 4.3 – Extensão do Fragmento da Ontologia de Requisitos de Software

Um *Caso de Uso* é um tipo de *Artefato* que descreve a interação entre um ator e o sistema. Um caso de uso possui um conjunto de *Cursos de Eventos*, descrevendo passos para se executar uma funcionalidade do sistema. Um curso de eventos pode ser um *Curso Básico*, que contempla a sequência de passos em que o ator envolvido no caso de uso consegue atingir seu objetivo sem interrupções, ou um *Curso Alternativo*, que lista exceções e erros que podem ocorrer durante a execução do caso de uso (ROSENBERG; STEPHENS, 2007; SHNEIDER; WINTERS, 2001; COCKBURN, 2001). *Casos de Uso* são alocados a *Módulos* e atendem *Requisitos*.

4.3.3 - Projeto e Implementação das Ontologias

Com base nos fragmentos das ontologias selecionadas e nos elementos das mesmas (conceitos e relacionamentos), foi elaborado um modelo de projeto contemplando a união desses fragmentos. Esse modelo, mostrado na Figura 4.4, foi a base para a implementação das ontologias em OWL.

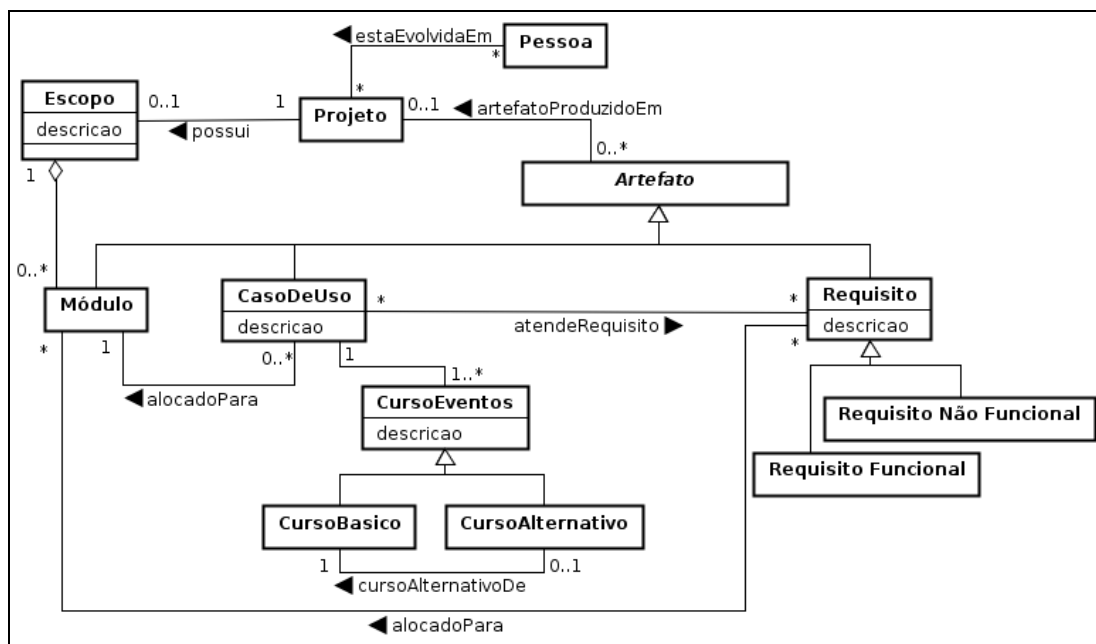


Figura 4.4 - Modelo de Projeto das Ontologias

Em resumo, um projeto possui um escopo que conta com uma descrição e com vários módulos. Casos de Uso são alocados a módulos e atendem a requisitos (funcionais ou não funcionais). Cada caso de uso possui uma descrição sucinta e um conjunto de cursos de eventos, os quais possuem também uma descrição, descrevendo os passos de interação entre ator e sistema. Cursos alternativos são referentes aos cursos básicos do mesmo caso de uso descrito.

Similar a um caso de uso, um requisito pode ser alocado a um módulo. Módulos, casos de uso e requisitos são artefatos produzidos em um projeto. Pessoas são envolvidas no projeto como parte do mesmo.

De posse desse modelo, foi conduzida a implementação do mesmo em OWL. O artefato gerado a partir dessa implementação foi utilizado no processo de anotação semântica de modelos de documento. A Figura 4.5 apresenta uma imagem da ferramenta Protégé¹⁵ na visão de classes da mesma, usada para implementar o modelo de projeto. A Figura 4.6 apresenta a visão de propriedades dessa mesma ferramenta.

¹⁵ The Protégé Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>, visitado em 28/07/2010.

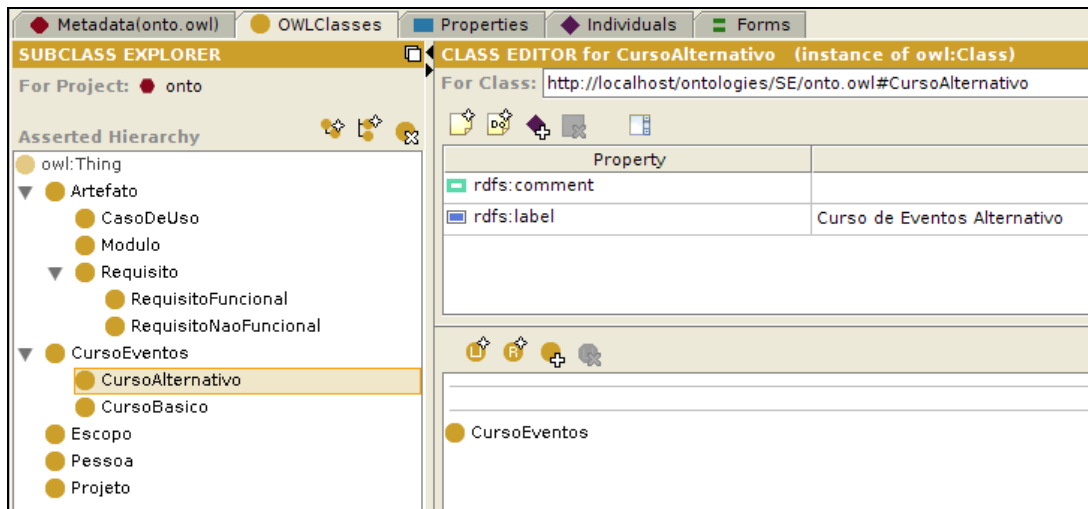


Figura 4.5 - Implementação do Modelo de Projeto na ferramenta Protégé (visão de classes)

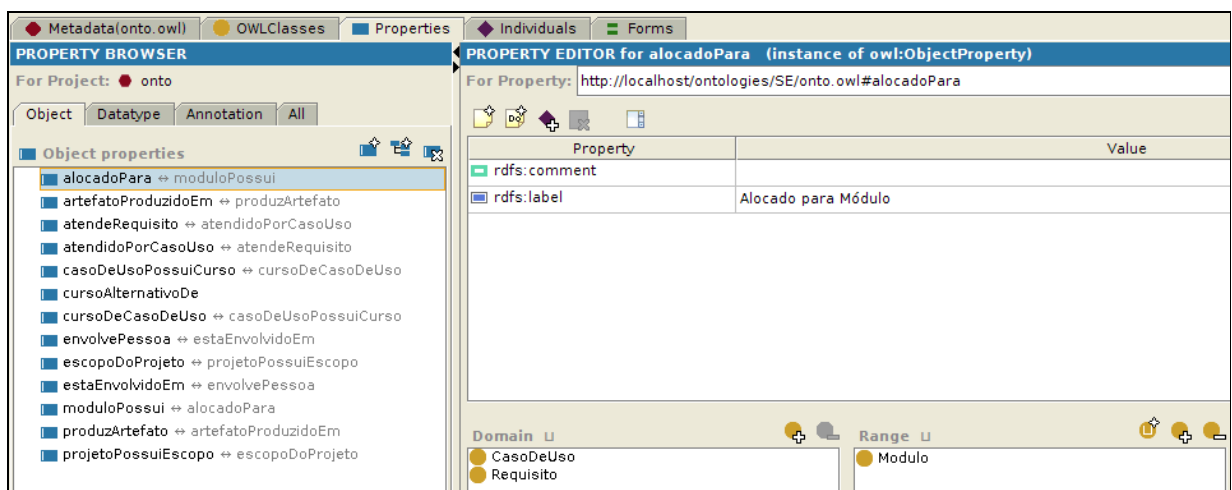


Figura 4.6 - Implementação do Modelo de Projeto na ferramenta Protégé (visão de propriedades)

Os relacionamentos existentes entre as classes foram implementados definindo-se o nome da propriedade como o nome da relação no modelo de projeto e como domínio (*Domain*) e alcance (*Range*) os conceitos relacionados. Na Figura 4.6 pode-se observar o relacionamento *alocadoPara*, o qual pode ser aplicado entre instâncias de *Caso de Uso / Requisito* (*Domain*) e *Módulo* (*Range*).

À exceção do relacionamento *cursoAlternativoDe*, que materializa a relação entre um *Curso Alternativo* e um *Curso Básico* de um caso de uso, todos os outros relacionamentos foram implementados de forma a possuírem um relacionamento inverso. Assim, se existe alguma

instância *A* com o valor da propriedade *p* igual a *B* e *p* possui a inversa *invp*, então *B* possui *A* como o valor da propriedade *invp*.

Tomando por base a implementação OWL elaborada, foram definidas duas consultas sobre o domínio de Engenharia de Requisitos, uma relativa à Matriz de Rastreabilidade de Requisitos e a outra com foco na identificação de não-conformidades.

A primeira consulta elaborada (referente à Matriz de Rastreabilidade de Requisitos) explora a relação existente entre um projeto, seus módulos, os casos de uso alocados para cada módulo e os requisitos atendidos pelos casos de uso. A Figura 4.7 apresenta o código SPARQL correspondente.

```

1. PREFIX onto: <http://localhost/ontologies/SE/onto.owl#>
2. SELECT DISTINCT ?projeto ?modulo ?casouso ?req
3. WHERE {
4.   ?projeto a onto:Projeto .
5.   OPTIONAL {
6.     ?projeto onto:produzArtefato ?req .
7.     ?req a onto:Requisito .
8.   }
9.   OPTIONAL {
10.    ?modulo onto:moduloPossui ?casouso .
11.    ?casouso onto:atendeRequisito ?req .
12.    ?projeto onto:produzArtefato ?casouso.
13.  }
14. }
15. GROUP BY ?projeto ?modulo ?casouso ?req
16. ORDER BY ?projeto ?modulo ?casouso

```

Figura 4.7 – Consulta SPARQL da Matriz de Rastreabilidade de Requisitos do Projeto

O resultado dessa consulta tem o formato de uma tabela contendo colunas para projeto, módulo, caso de uso e requisito (linha 2), como ilustra a Tabela 4.1, indicando, para cada projeto listado no resultado, seus módulos; para cada módulo do projeto, os casos de uso alocados para aquele módulo; e para cada caso de uso alocado ao módulo, os requisitos (funcionais e não funcionais) que o caso de uso alocado atende.

Tabela 4.1 - Exemplo de Resultados da Matriz de Rastreabilidade

Projeto	Módulo	Caso de Uso	Requisito
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF03
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF04

Em relação ao código da Figura 4.7, a primeira linha define o modelo (implementação OWL da ontologia) utilizado na consulta. A linha 2 indica que o resultado da consulta vai

apresentar linhas contendo projeto, módulo, caso de uso e requisito (*?projeto*, *?modulo*, *?casouso* e *?req*, respectivamente). A condicional da consulta, que se inicia na linha 3, define uma série de condições que as instâncias e valores das colunas devem atender para aparecer no resultado da consulta. A linha 4, estabelece a primeira condição, definindo de que a variável *?projeto*, mostrada no resultado, deve ser um projeto. Os blocos iniciados nas linhas 5 e 9 apresentam duas situações para composição do resultado da consulta.

O bloco opcional que se inicia na linha 5 define que o projeto da consulta deve produzir (*onto:produzArtefato*) um artefato referenciado por *?req* (linha 6), o qual deve ser do tipo *Requisito* (linha 7). Em essência, esse bloco opcional visa capturar os requisitos de um projeto, independente de qualquer outro relacionamento que os mesmos possuam.

Já o bloco opcional que se inicia na linha 9 visa capturar os módulos que possuem casos de uso alocados (linha 10). Cada caso de uso alocado deve atender a um requisito (linha 11) que é um artefato produzido no mesmo projeto (linha 12) definido na linha 4.

As duas últimas linhas da consulta efetuam, respectivamente, o agrupamento e a ordenação do resultado. Cenários de aplicação dessa consulta são apresentados na subseção 4.5.1.

A segunda consulta elaborada (referente à identificação de não conformidades) tenta capturar os elementos (requisitos, casos de uso e cursos de eventos) que não possuem uma descrição associada. Adicionalmente a consulta tenta capturar casos de uso que não atendem nenhum requisito. A Figura 4.8 apresenta o código SPARQL correspondente.

```

1.  PREFIX onto: <http://localhost/ontologies/SE/onto.owl#>
2.  SELECT DISTINCT ?ime ?classe {
3.    {
4.      ?ime a ?classe .
5.      NOT EXISTS { ?ime onto:descricao ?descricao } .
6.      FILTER (
7.        ?classe = onto:Requisito ||
8.        ?classe = onto:CasoDeUso ||
9.        ?classe = onto:CursoEventos
10.     )
11.    }
12.  UNION
13.  {
14.    ?ime a ?classe .
15.    NOT EXISTS { ?ime onto:atendeRequisito ?req } .
16.    FILTER (?classe = onto:CasoDeUso)
17.  }
18. }
19. ORDER BY ?classe ?ime

```

Figura 4.8 - Consulta SPARQL para identificação de não conformidades

Em relação ao código da Figura 4.8, a primeira linha define o modelo utilizado na consulta, no caso a implementação OWL das ontologias de domínio, conforme discutido anteriormente. A linha 2 define os elementos a serem apresentados no resultado da consulta, a saber: *?ime*, que representa o elemento que apresenta a não conformidade, e *?classe*, que indica qual o tipo do elemento *?ime*. O bloco de código que vai da linha 3 até a linha 11 define a consulta por instâncias de *Requisito*, *CasoDeUso* e *CursoEventos* que não possuem uma descrição associada. Na linha 4 a classe da instância *?ime* é restringida às classes *Requisito*, *CasoDeUso* e *CursoEventos* (definidas com o filtro – *FILTER* – nas linhas 7-9). A condição imposta na linha 5 define que essa instância (*?ime*) não deve ter valores para a propriedade descrição (*onto:descricao*). O bloco de código que vai da linha 13 a 17 captura instâncias de *Caso de Uso* que não atendem nenhum requisito. A linha 15 define essa condição, enquanto a linha 16 filtra as classes com essa condição. Os dois blocos são unidos através da operação UNION (linha 12). Um cenário de aplicação dessa consulta é apresentado na subseção 4.5.2.

Ambas consultas, a referente à Matriz de Rastreabilidade de Requisitos e a referente à identificação de não conformidades são realizadas no Módulo de Busca e Rastreabilidade (MBR), que provê uma interface para consultas em SPARQL sobre o Repositório de Dados, dado um Repositório de Documentos Semânticos e uma versão do mesmo. A Figura 4.9 apresenta a interface gráfica referente a essa consulta provida pelo MBR.

Union Graph Search:

```

PREFIX onto: <http://localhost/ontologies/SE/onto.owl#>
SELECT DISTINCT ?projeto ?modulo ?casouso ?req
WHERE {
?projeto a onto:Projeto .
  OPTIONAL {
    ?projeto onto:produzArtefato ?req .
    ?req a onto:Requisito .
  }
  OPTIONAL {
    ?modulo onto:moduloPossui ?casouso .
    ?casouso onto:atendeRequisito ?req .
  }
}

```

in repository/repository version:

SEARCH!

Repository Model Version
(/var/svn/analise_pgds) #1 is
composed by:

[/docRequisitos.odt in svn
revision\(1\)](#)

Figura 4.9 - Interface Gráfica do MBR - Consulta com SPARQL

A primeira área de texto da figura é utilizada para se definir a consulta desejada (nesse caso a própria consulta de matriz de rastreabilidade). Os outros dois campos definem o endereço do RDS e a versão do RDS a ser analisado, respectivamente. Como resultado, a consulta oferece adicionalmente um *link* para os documentos semânticos e versões dos mesmos usados na consulta (parte direita da Figura 4.9).

Com as ontologias implementadas, é possível enriquecer semanticamente os modelos de documento. Essa etapa é discutida a seguir.

4.4 - Anotação Semântica em Modelos de Documento

Conforme discutido anteriormente, dois modelos de documentos definidos pelo NEMO tem de ser anotados para tratar o cenário de aplicação descrito, são eles: o Modelo de Documento de Requisitos e o Modelo de Descrição de Caso de Uso.

Inicialmente foi feita uma análise dos modelos de documentos atuais. Alguns elementos usados nesses modelos não são ainda contemplados pela linguagem de anotação em modelos de documento proposta neste trabalho e, por esse motivo, foram modificados de forma a permitir a realização de anotações.

Tomando como base a implementação das ontologias em OWL descrita na seção anterior e usando a linguagem de anotações descrita no Capítulo 3 e a ferramenta *Open Office Writer*, foram efetuadas anotações semânticas nos modelos alterados. Os modelos de documento

originais e os modelos semanticamente anotados são apresentados nos Anexos A e Apêndice B, respectivamente.

Como o Módulo de Anotação em Modelos de Documento (MAMD) ainda não possui uma ferramenta de apoio à anotação de modelos de documentos, os engenheiros de documento devem efetuar as anotações de acordo com o processo definido no Capítulo 3. Em resumo, as anotações são efetuadas em dois elementos de documento: trechos de texto e tabelas. Para cada anotação em trecho de texto, os seguintes passos são realizados: (i) definir o trecho de texto anotado no documento; (ii) criar um estilo de formatação com um nome específico seguindo o padrão definido no Capítulo 3; (iii) criar um campo de usuário para comportar a anotação e (iv) definir a anotação textual e o conteúdo da mesma.

Para definir uma anotação textual, o engenheiro de documento deve escolher entre usar o conteúdo textual anotado por completo, usando a anotação `[[completeText]]`, ou dividir o conteúdo segundo um caractere, fazendo uso da anotação `[[break with <<caractere>> into <<variável>>]]`.

Anotações em tabelas seguem um processo similar, com a diferença que os passos (i) e (ii) são substituídos pela nomeação da tabela (segundo o padrão descrito no Capítulo 3). Além disso, a definição da anotação e do conteúdo da mesma deve ser feita de acordo com a coluna da tabela, usando a anotação `[[at <<número da coluna>>]]`.

As subseções seguintes detalham o processo de anotação semântica dos modelos de documento considerados.

4.4.1 - Anotação Semântica do Modelo de Documento de Requisitos

Para adicionar anotações semânticas no Modelo de Documento de Requisitos do NEMO, foi necessário, primeiro, fazer uma adaptação do mesmo, visto que esse modelo contava com elementos não tratados pela linguagem de anotação de modelos. Assim, o formato do Modelo de Documento de Requisitos foi modificado, de modo a utilizar apenas trechos de texto e tabelas. A Figura 4.10 mostra um comparativo entre o modelo original e o modelo modificado.

Projeto: <<Nome do Projeto>>	Projeto: <<Nome do Projeto>>		
Versão: <<Versão do documento>>	Versão: <<Versão do Documento>>		
Responsável: <<Lista de Responsáveis>>	Responsáveis: <<Nome dos Responsaveis, separados por vírgula>>		
1. Introdução <<Descrição da Organização do Documento>>	1. Introdução <<Descrição da Organização do Documento>>		
2. Descrição do Propósito do Sistema <<Descrição sucinta do propósito do sistema>>	2. Descrição do Propósito do Sistema <<Descrição sucinta do propósito do sistema>>		
3. Descrição do Mini-mundo <<Descrição do escopo do problema>>	3. Descrição do Mini-mundo <<Descrição do escopo do problema>>		
4. Requisitos Iniciais do Cliente Listar cada identificador de requisito, juntamente com sua descrição.	4. Requisitos Iniciais do Cliente		
Requisitos Funcionais RF1. <<Descrição do requisito RF1>>; RF2. <<Descrição do requisito RF2>>.	Requisitos Funcionais <table><tr><td><<id do Requisito>></td><td><<Descrição do requisito>></td></tr></table>	<<id do Requisito>>	<<Descrição do requisito>>
<<id do Requisito>>	<<Descrição do requisito>>		
Requisitos Não Funcionais RNF1. <<Descrição do requisito RNF1>>; RNF2. <<Descrição do requisito RNF2>>.	Requisitos Não Funcionais <table><tr><td><<id do Requisito>></td><td><<Descrição do requisito>></td></tr></table>	<<id do Requisito>>	<<Descrição do requisito>>
<<id do Requisito>>	<<Descrição do requisito>>		

Figura 4.10 - Comparativo entre o Modelo de Documento de Requisitos original (à esquerda) e sua nova formatação (à direita) para comportar anotações semânticas

Analisando os dois modelos apresentados na Figura 4.10, o modelo preparado para anotações semânticas é similar ao modelo original, contendo poucas modificações nos elementos do documento. As diferenças mais significativas são: (i) no item 3 – Descrição do Mini-Mundo – enquanto o texto original define um parágrafo textual para comportar a descrição, o modelo modificado define uma tabela de uma coluna para comportar a mesma descrição; (ii) a definição dos requisitos funcionais e não funcionais (item 4 dos modelos) foi modificada de listagem em texto para uso de tabelas de duas colunas.

Feitas as modificações de formato para que o Modelo de Documento de Requisitos tenha a estrutura necessária para comportar as anotações, o próximo passo é definir os estilos de formatação referentes aos trechos de texto e tabelas anotadas utilizando o *OpenOffice Writer*. A Figura 4.11 ilustra o Modelo de Documento de Requisitos anotado semanticamente. A composição de uma anotação é dada pela localização do elemento anotado, delimitado por um estilo de formatação de texto ou uma tabela, e um campo de usuário, que define o conjunto de instruções daquela localização. Os pontos destacados pelas letras de A a E indicam a localização onde houve anotação no modelo de documento. Esses pontos referem-se às seguintes anotações:

Projeto: <<Nome do Projeto>> A			
Versão: <<Versão do Documento>>			
Responsáveis: <<Nome dos Responsaveis, separados por vírgula>> B			
1. Introdução			
<<Descrição da Organização do Documento>>			
2. Descrição do Propósito do Sistema			
<<Descrição sucinta do propósito do sistema>>			
3. Descrição do Mini-mundo			
<table border="1"> <tr> <td colspan="2"><<Descrição do escopo do problema>> C</td> </tr> </table>		<<Descrição do escopo do problema>> C	
<<Descrição do escopo do problema>> C			
4. Requisitos Iniciais do Cliente			
Requisitos Funcionais			
<table border="1"> <tr> <td><<id do Requisito>></td> <td><<Descrição do requisito>></td> </tr> </table>	<<id do Requisito>>	<<Descrição do requisito>>	D
<<id do Requisito>>	<<Descrição do requisito>>		
Requisitos Não Funcionais			
<table border="1"> <tr> <td><<id do Requisito>></td> <td><<Descrição do requisito>></td> </tr> </table>	<<id do Requisito>>	<<Descrição do requisito>>	E
<<id do Requisito>>	<<Descrição do requisito>>		

Figura 4.11 - Modelo de Documento de Requisitos com Anotações Semânticas

- A: aponta o local de onde será capturada uma instância de *Projeto* (vide figura 4.4). O trecho de texto <<Nome do Projeto>>, apresentado nesse ponto, foi marcado com o estilo de formatação de nome “*SemanticAnnotation-ref-hf001*”.
- B: aponta o local das instâncias de *Pessoa* (envolvidos no projeto e responsáveis pelo documento). O trecho de texto referente a esse ponto, demarcado pela string <<Nome dos Responsáveis, separados por vírgula>>, foi formatado com o estilo de nome “*SemanticAnnotation-ref-h002*”.
- C: refere-se a uma instância de *Escopo* (mais especificamente a propriedade *descrição* do conceito *Escopo*) do projeto definido no documento. A tabela referente a esse ponto foi nomeada “*SemanticAnnotation-ref-hf004*”.
- D e E: definem os requisitos funcionais e não funcionais do projeto, juntamente com suas descrições. Os nomes dados às tabelas foram “*SemanticAnnotation-ref-hf005*” e “*SemanticAnnotation-ref-hf006*”, respectivamente.

As anotações semânticas citadas, referentes a trechos de texto ou a tabelas (*SemanticAnnotation-ref-hf<número>*), têm uma variável de referência (*hf<número>*) criada como campo de usuário. A Figura 4.12 apresenta a lista de campos de usuário do *OpenOffice*

Writer com as variáveis de referência contendo as anotações. Cada círculo mostrado na Figura 4.12 corresponde ao mesmo círculo mostrado na Figura 4.11.

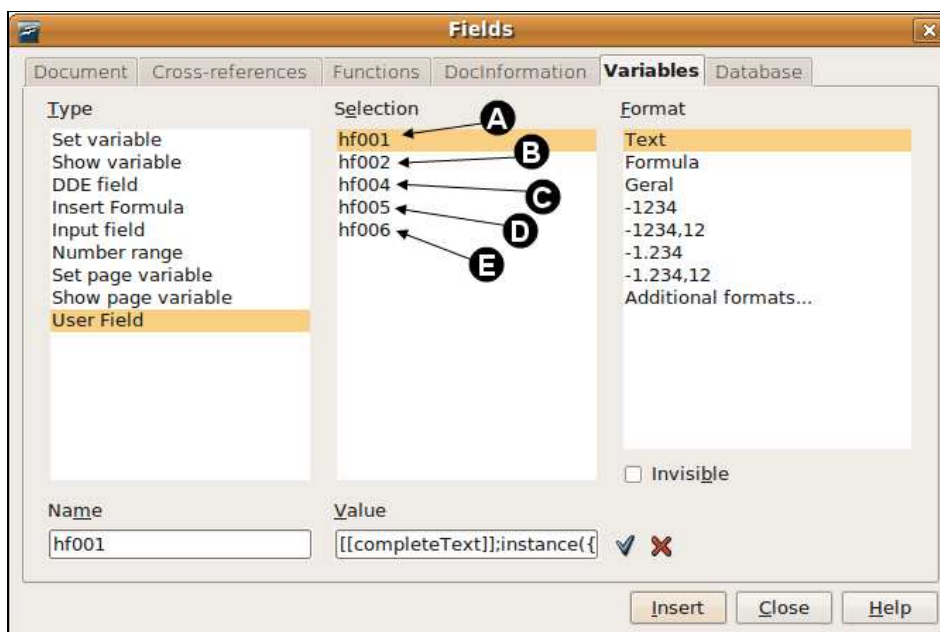


Figura 4.12 - Campos de usuário definidos no OpenOffice Writer para o Modelo de Documento de Requisitos

A Tabela 4.2 apresenta, para cada círculo da Figura 4.11, a anotação correspondente (contida no campo “Value” da Figura 4.12). Os pontos A e B são referentes a trechos de texto anotados. Já os pontos C, D e E são referentes a tabelas anotadas.

Tabela 4.2 - Conjunto de Anotações Semânticas do Modelo de Documento de Requisitos

Anotação	Conteúdo
A	[[completeText]]; instance({content},http://localhost/ontologies/SE/onto.owl#Projeto,\$projeto);
B	[[break with ',' into 'slice']]; instance({slice},http://localhost/ontologies/SE/onto.owl#Pessoa,\$pessoa); property(\$pessoa,http://localhost/ontologies/SE/onto.owl#estaEnvolvidoEm,\$projeto);
C	[[at 0]]; instance(escopo_\$projeto,http://localhost/ontologies/SE/onto.owl#Escopo,\$escopo); property(\$escopo,http://localhost/ontologies/SE/onto.owl#descricao,{content}); property(\$escopo,http://localhost/ontologies/SE/onto.owl#escopoDoProjeto,\$projeto);
D	[[at 0]]; instance({content},http://localhost/ontologies/SE/onto.owl#RequisitoFuncional,#req); property(#req,http://localhost/ontologies/SE/onto.owl#artefatoProduzidoEm,\$projeto); [[at 1]]; property(#req,http://localhost/ontologies/SE/onto.owl#descricao,{content});
E	[[at 0]]; instance({content},http://localhost/ontologies/SE/onto.owl#RequisitoNaoFuncional,#req); property(#req,http://localhost/ontologies/SE/onto.owl#artefatoProduzidoEm,\$projeto); [[at 1]]; property(#req,http://localhost/ontologies/SE/onto.owl#descricao,{content});

A anotação “A” da Tabela 4.2 captura o conteúdo do texto formatado com o estilo “*SemanticAnnotation-ref-hf001*” e cria uma instância do conceito *Projeto*, mantendo a variável *\$projeto* como uma referência para essa instância. A anotação “B” captura o conteúdo de texto informado e quebra esse conteúdo quando há uma ocorrência do caractere “,”, mantendo, para cada resultado da quebra, o conteúdo desse resultado na variável *{slice}*. Essa variável é utilizada, para criar uma instância do conceito *Pessoa*, referenciada por *\$pessoa*, que por sua vez é utilizada para definir o relacionamento *estaEnvolvidoEm* entre a instância de projeto previamente criada (*\$projeto*) e cada instância de pessoa definida.

A anotação “C” define a rotina de tratamento de instâncias do conceito *Escopo*. Primeiramente é criada uma instância com o nome definido pela string “*escopo_*” concatenada com o nome da instância de projeto (adquirida pela variável *\$projeto*). Essa string é guardada na variável *\$escopo*. Criada a instância, o conteúdo textual da célula da tabela é atribuído à descrição do escopo (propriedade *descricao*). Finalmente a instância de escopo criada é relacionada ao projeto através do relacionamento *escopoDoProjeto*.

As anotações “D” e “E” referem-se à instanciação de requisitos funcionais e não funcionais, respectivamente. Para cada linha da tabela apontada por “D”, as células referentes à primeira coluna têm seus conteúdos textuais usados para criar instâncias do conceito *Requisito Funcional* e relacionar as mesmas ao projeto referenciado no documento (*\$projeto*). A segunda coluna captura a descrição (propriedade *descricao*) de cada requisito funcional. A anotação da tabela apontada por “E” é análoga à apontada por “D”. A única diferença é que a primeira cria instâncias de Requisito Não Funcional (conceito *RequisitoNaoFuncional*) e a segunda de Requisitos Funcionais (conceito *RequisitoFuncional*).

4.4.2 - Anotação Semântica do Modelo de Descrição de Casos de Uso

O Modelo de Descrição de Casos de Uso do NEMO registra informações acerca de um caso de uso, a saber: o projeto ao qual o caso de uso pertence, o módulo ao qual o caso de uso está alocado, o nome do caso de uso, uma descrição sucinta do caso de uso, seus fluxos normais de execução, juntamente com as respectivas descrições, e os fluxos alternativos de cada fluxo normal citado (quando existir).

Para anotá-lo semanticamente, um processo similar ao apresentado anteriormente foi conduzido. Assim, inicialmente foi necessário alterar sua formatação, de modo a utilizar apenas

trechos de texto e tabelas. A Figura 4.13 apresenta o comparativo entre o modelo original e o modelo alterado para comportar as anotações semânticas.

Projeto : <<Nome do Projeto>> Subsistema : <<Nome do Subsistema ou Módulo>> Caso de Uso: <<Nome do caso de uso>>	Projeto : <<Nome do Projeto>> Subsistema : <<Nome do Subsistema ou Módulo>> Caso de Uso: <<Nome do Caso de Uso>>		
Descrição: <<Descrição sucinta do caso de uso>>	Descrição: <<Descrição sucinta do caso de uso>>		
Fluxos Normais <<Nome de cada Fluxo Normal>> 1. <<Descrição do Passo 1>>; 2. <<Descrição do Passo 2>>;	Fluxos Normais <table> <tr> <td><<nome de cada fluxo normal em cada linha>></td><td><<sequência de passos de interação>></td></tr> </table>	<<nome de cada fluxo normal em cada linha>>	<<sequência de passos de interação>>
<<nome de cada fluxo normal em cada linha>>	<<sequência de passos de interação>>		
Fluxos Alternativos de Exceção <<Nome de cada Fluxo Normal que possua Fluxo Alternativo de Exceção >> <<Número do passo com o fluxo alternativo>> <<Descrição do passo alternativo>>	Fluxos Alternativos de Exceção <table> <tr> <td><<nome do fluxo normal que possui fluxo alternativo>></td><td><<descrição do curso alternativo>></td></tr> </table>	<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>>
<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>>		
Fluxos Alternativos Variantes <<Nome de cada Fluxo Normal que possua Fluxo Alternativo Variante>> <<Número do passo com fluxo alternativo>> <<Descrição do passo alternativo>>	Fluxos Alternativos Variantes <table> <tr> <td><<nome do fluxo normal que possui fluxo alternativo>></td><td><<descrição do curso alternativo>></td></tr> </table>	<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>>
<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>>		
Requisitos: <<Listagem dos requisitos atendidos por esse caso de uso>>	Requisitos: <<Lista de Requisitos separados por vírgula>>		

Figura 4.13 - Comparativo entre o Modelo de Descrição de Caso de Uso original (à esquerda) e sua nova formatação (à direita) para comportar anotações semânticas

As diferenças entre os dois modelos são: (i) o item Descrição, referente à descrição do caso de uso em questão, passou a ser feito em uma tabela de uma única coluna; (ii) os fluxos normais e alternativos são definidos também em tabelas; (iii) a listagem de requisitos atendidos pelo caso de uso deve separar os identificadores dos requisitos por vírgula.

Feitas as devidas alterações de formatação, o modelo resultante foi anotado semanticamente. A Figura 4.14 apresenta os pontos em que anotações semânticas foram adicionadas. Essas anotações referem-se a:

Projeto : <<Nome do Projeto>> A	
Subsistema : <<Nome do Subsistema ou Módulo>> B	
Caso de Uso: <<Nome do Caso de Uso>> C	
Descrição: <<Descrição sucinta do caso de uso>> D	
Fluxos Normais	
<<nome de cada fluxo normal em cada linha>>	<<sequência de passos de interação>> E
Fluxos Alternativos de Exceção	
<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>> F
Fluxos Alternativos Variantes	
<<nome do fluxo normal que possui fluxo alternativo>>	<<descrição do curso alternativo>> G
Requisitos: <<Lista de Requisitos separados por vírgula>> H	

Figura 4.14 - Modelo de Descrição de Caso de Uso da Organização X (modificado)

- A: é a anotação que capturada uma instância de *Projeto*. O trecho de texto <<Nome do Projeto>> foi marcado com o estilo de formatação de nome “*SemanticAnnotation-ref-hf001*”.
- B: refere-se à anotação que captura o módulo que comporta o caso de uso em questão. Nesse caso foi definido também um estilo de formatação, dessa vez com o nome “*SemanticAnnotation-ref-hf002*”. Esse estilo foi usado no texto <<Nome do Subsistema ou Módulo>>.
- C: é a anotação referente à captura do caso de uso descrito no documento. O trecho de texto referente a esse ponto (<<Nome do Caso de Uso>>) foi formatado com o estilo de nome “*SemanticAnnotation-ref-hf003*”.
- D: é a anotação da tabela “Descrição”, nomeada de “*SemanticAnnotation-ref-hf004*”. Seu conteúdo corresponde à descrição sucinta do caso de uso.
- E, F e G: são as anotações relativas às tabelas de fluxos normais (nomeada “*SemanticAnnotation-ref-hf005*”), fluxos alternativos de exceção (nomeada

“*SemanticAnnotation-ref-hf006*”) e fluxos alternativos variantes (nomeada “*SemanticAnnotation-ref-hf007*”), respectivamente.

- H: anotação que aponta os requisitos atendidos pelo caso de uso descrito. Novamente criou-se um estilo de formatação, nomeado “*SemanticAnnotation-ref-hf008*”, para o trecho de texto anotado (<<Lista de Requisitos separados por vírgula>>).

Similarmente ao Modelo de Documento de Requisitos, cada anotação semântica efetuada para o Modelo de Descrição de Caso de Uso teve uma variável de referência criada como um campo de usuário do *OpenOffice Writer*. A Figura 4.15 ilustra esses campos e os pontos referentes ao documento.

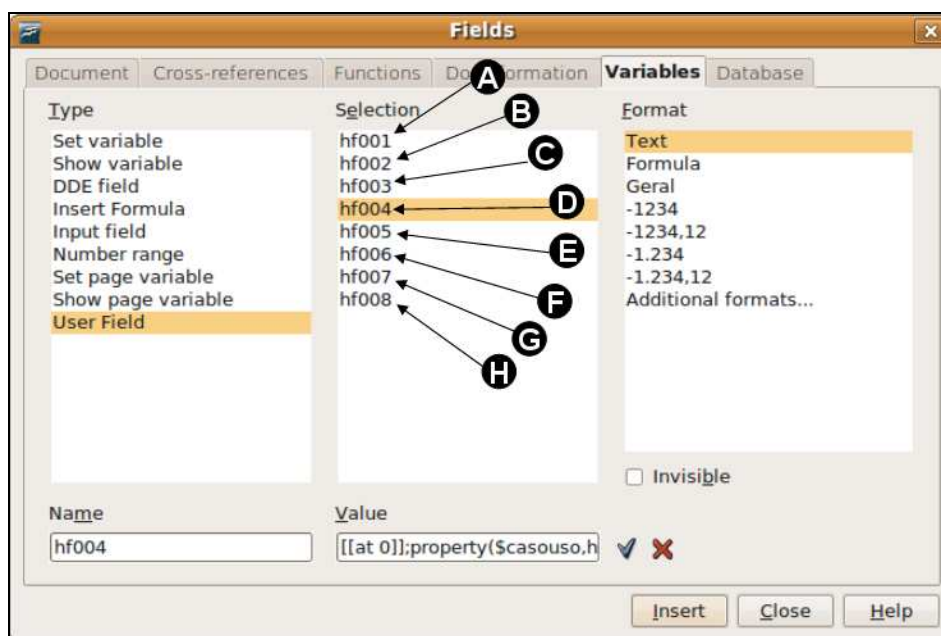


Figura 4.15 - Campos de usuário definidos no OpenOffice Writer para o Modelo de Descrição de Caso de Uso

A Tabela 4.3 apresenta, para cada círculo apontado na Figura 4.14, a anotação correspondente (definida no campo “*Value*” da Figura 4.15). Os pontos A, B, C e H são referentes a trechos de texto anotados. Já os pontos D, E, F e G são referentes a tabelas anotadas.

Tabela 4.3 - Anotações Semânticas do Modelo de Descrição de Caso de Uso

Anotação	Conteúdo
A	[[completeText]]; instance({content},http://localhost/ontologies/SE/onto.owl#Projeto,\$projeto);
B	[[completeText]]; instance({content},http://localhost/ontologies/SE/onto.owl#Modulo,\$modulo);
C	[[completeText]]; instance({content},http://localhost/ontologies/SE/onto.owl#CasoDeUso,\$casouso); property(\$casouso,http://localhost/ontologies/SE/onto.owl#alocadoPara,\$modulo); property(\$casouso,http://localhost/ontologies/SE/onto.owl#artefatoProduzidoEm,\$projeto);
D	[[at 0]]; property(\$casouso,http://localhost/ontologies/SE/onto.owl#descricao,{content});
E	[[at 0]]; instance({content},http://localhost/ontologies/SE/onto.owl#CursoBasico,#cursobasico); property(#cursobasico,http://localhost/ontologies/SE/onto.owl#cursoDeCasoDeUso,\$casouso); [[at 1]]; property(#cursobasico,http://localhost/ontologies/SE/onto.owl#descricao,{content});
F	[[at 0]]; instance({content},http://localhost/ontologies/SE/onto.owl#CursoBasico,#cursobasico); instance(altexc_#cursobasico,http://localhost/ontologies/SE/onto.owl#CursoAlternativo,#cursoalt); property(#cursoalt,http://localhost/ontologies/SE/onto.owl#cursoDeCasoDeUso,\$casouso); property(#cursoalt,http://localhost/ontologies/SE/onto.owl#cursoAlternativoDe,#cursobasico); [[at 1]]; property(#cursoalt,http://localhost/ontologies/SE/onto.owl#descricao,{content});
G	[[at 0]]; instance({content},http://localhost/ontologies/SE/onto.owl#CursoBasico,#cursobasico); instance(altnvar_#cursobasico,http://localhost/ontologies/SE/onto.owl#CursoAlternativo,#cursoalt); property(#cursoalt,http://localhost/ontologies/SE/onto.owl#cursoDeCasoDeUso,\$casouso); property(#cursoalt,http://localhost/ontologies/SE/onto.owl#cursoAlternativoDe,#cursobasico); [[at 1]]; property(#cursoalt,http://localhost/ontologies/SE/onto.owl#descricao,{content});
H	[[break with ',' into 'slice']]; instance({slice},http://localhost/ontologies/SE/onto.owl#Requisito,#req); property(\$casouso,http://localhost/ontologies/SE/onto.owl#atendeRequisito,#req);

A anotação “A” captura o conteúdo de texto e cria uma instância de *Projeto*, colocando uma referência para essa instância na variável *\$projeto*. A anotação “B” faz um processo similar, dessa vez criando uma instância de *Modulo* e colocando em *\$modulo* uma referência para a mesma. Na anotação “C”, primeiro é criada uma instância de *Caso de Uso*, referenciada pela variável *\$casouso*. Em seguida, a relação *alocadoPara* é definida entre a instância de caso de uso recém criada (*\$casouso*) e a instância de módulo referenciada nesse documento (*\$modulo*). Outra relação é ainda definida nessa anotação: a instância de caso de uso gerada é relacionada ao projeto (*\$projeto*) pela propriedade *artefatoProduzidoEm*. A anotação “D” captura a descrição (propriedade *descricao*) do caso de uso (*\$casouso*).

A anotação “E”, relativa à tabela descrevendo os fluxos normais do caso de uso, inicialmente captura o conteúdo da célula da primeira coluna de cada linha existente na tabela e

cria uma instância de *Curso Básico* e registra a mesma na variável *#cursobasico*. Adicionalmente essa instância é relacionada ao caso de uso do documento (*\$casouso*) pela propriedade *cursoDeCasoDeUso*. A segunda célula de cada linha é usada para capturar a descrição do respectivo curso através da propriedade *descricao*.

As anotações “F” e “G” realizam praticamente as mesmas ações: para cada linha (i) captura-se a instância de curso básico previamente criada; (ii) cria-se uma instância de curso alternativo (*#cursoalt*); (iii) relaciona a mesma ao caso de uso descrito no documento (*\$casouso*), por meio da propriedade *cursoDeCasoDeUso*; e (iv) relaciona a instância de curso alternativo ao curso básico capturado (*#cursobasico*), por meio da propriedade *cursoAlternativoDe*. A diferença entre “F” e “G” se encontra na segunda linha de cada anotação: enquanto “F” usa o identificador da instância de curso alternativo como sendo a concatenação de “altexc_” com o identificador do curso básico tratado na linha (*#cursobasico*), “G” usa a string “altvar_” concatenada com o mesmo identificador, diferenciando os cursos alternativos de exceção e variantes.

A última anotação (“H”) captura os requisitos atendidos pelo caso de uso descrito. Primeiramente, cada *Requisito* é instanciado e armazenado na variável *#requisito*. Em seguida, o caso de uso descrito no documento (*\$casouso*) é relacionado a cada requisito encontrado (*#requisito*) pelo uso da propriedade *atendeRequisito*.

4.5 - Uso da Plataforma no apoio à Engenharia de Requisitos

Para avaliar a plataforma no contexto de apoio à Engenharia de Requisitos, optou-se por utilizá-la para apoiar as atividades de Documentação de Requisitos, Gerência de Requisitos e Identificação de Não Conformidades relativas a requisitos de um projeto de desenvolvimento de um software para uma videolocadora. Dentre os produtos de trabalho gerados nesse projeto, as versões dos documentos de requisitos e de descrição de casos de uso foram gerenciadas usando a PGDS.

As várias versões desses documentos foram adicionadas ao Repositório de Documentos Semânticos (RDS), sendo que ao final do projeto, o RDS desse projeto contava com nove versões, contemplando um documento de requisitos e dez documentos de descrição de caso de uso. Todos esses documentos foram gerados a partir dos modelos semânticos definidos na seção 4.4. As versões finais desses documentos são apresentadas no Anexo B.

Esta seção está organizada da seguinte forma: na subseção 4.5.1, uma análise da matriz de rastreabilidade de requisitos do projeto é conduzida, apontando o estado da mesma em algumas versões do RDS; a subseção 4.5.2 apresenta a evolução do projeto sendo observada sob um foco de qualidade, onde os responsáveis tentam detectar situações de não-conformidade; na subseção 4.5.3, é apresentada uma situação de notificação de alteração de uma instância observada por um desenvolvedor; finalmente a seção 4.5.4 apresenta a funcionalidade de rastreabilidade de mudanças de documentos semânticos e de indivíduos em documentos semânticos.

4.5.1 - Matriz de Rastreabilidade de Requisitos do Projeto

A consulta referente à matriz de rastreabilidade de requisitos do projeto, discutida na seção 4.3.3, foi utilizada em todas as versões do RDS. Os resultados da aplicação da consulta sobre cada versão do RDS são apresentados na íntegra no Apêndice C.

Para ilustrar o uso dessa consulta durante a evolução do projeto foram considerados três diferentes versões do RDS, a saber: a primeira versão, a versão de número 4 e a última versão (9).

A primeira versão do RDS contava apenas com um documento semântico e, portanto, uma versão do mesmo (*docRequisitos.odt* na versão 1, como mostra a parte direita da Figura 4.9). Esse documento apresenta os requisitos funcionais e não funcionais do sistema. Todo requisito funcional definido pela organização segue o padrão de identificação “RF<número>” e os não funcionais o padrão “RNF<número>”. O termo <número> é atribuído pelo próprio analista de sistemas ao descrever o documento, mas por convenção do NEMO deve possuir dois dígitos. A Tabela 4.4 apresenta o resultado da consulta à matriz de rastreabilidade de requisitos relativa a esse estado do RDS (versão 1).

Tabela 4.4 - Resultados da Matriz de Rastreabilidade na versão 1 do Projeto

Projeto	Módulo	Caso de Uso	Requisito
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08

Como se pode perceber, a versão 1 do RDS continha apenas um documento semântico, que em seu conteúdo definia oito requisitos.

Na versão de número 4 do RDS, o número de documentos aumenta. Um documento de requisitos (ainda docRequistos.odt) e cinco documentos de descrição de caso de uso (casoUso_Acervo_Cadastrar_Filme.odt, casoUso_Acervo_Cadastrar_Distribuidora.odt, casoUso_Acervo_Consultar_Acervo.odt, casoUso_Acervo_Cadastrar_Item.odt, casoUso_Acervo_Tipo_de_Midia.odt) compõem essa versão do RDS. A Tabela 4.5 apresenta o resultado da mesma consulta para a versão 4 do RDS.

Tabela 4.5 - Resultados da Matriz de Rastreabilidade na versão 4 do Projeto

Projeto	Módulo	Caso de Uso	Requisito
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RF12
Video_Locadora_Passatempo			RF13
Video_Locadora_Passatempo			RF14
Video_Locadora_Passatempo			RF15
Video_Locadora_Passatempo			RF16
Video_Locadora_Passatempo			RNF04
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF06
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09

Tabela 4.5 - Resultados da Matriz de Rastreabilidade na versão 4 do Projeto (Continuação)

Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02

Como pode ser observado, o número de requisitos aumentou de 8 para 16 entre a versão 1 e 4 do RDS. Adicionalmente a matriz de rastreabilidade de requisitos começa a ser apresentada de forma mais detalhada. Na primeira versão (Tabela 4.4) ainda não existiam casos de uso ou módulos atendendo requisitos. Já na versão 4 do RDS há um módulo (Acervo), que possui cinco casos de uso alocados (Cadastrar_Distribuidora, Cadastrar_Filme, Cadastrar_Item, Cadastrar_Tipo_de_Midia e Consultar_Acervo) e cada um deles define uma série de requisitos atendidos.

Ao final do projeto, uma consulta sobre a última versão do RDS (versão 9) aponta a existência dos onze documentos citados anteriormente. A Tabela 4.6 mostra o resultado da consulta de matriz de rastreabilidade de requisitos para essa versão, efetuada sobre o grafo de união dos conteúdos semânticos dos onze documentos semânticos envolvidos em suas últimas versões.

O resultado aponta que todos os requisitos funcionais levantados (vide Anexo B) foram atendidos por algum caso de uso que está devidamente alocado a um módulo do projeto. O projeto Video_Locadora_Passatempo conta, então, com os módulos Acervo e Atendimento. O módulo Acervo contém os casos de uso Cadastrar_Distribuidora, Cadastrar_Filme, Cadastrar_Item, Cadastrar_Tipo_de_Midia e Consultar_Acervo. Cada caso de uso define uma série de requisitos atendidos. O módulo Atendimento conta com os casos de uso Cadastrar_Cliente, Efetuar_Locacao, Efetuar_Devolucao, Efetuar_Pagamento e Efetuar_Reserva.

Os requisitos não relacionados a casos de uso foram os requisitos não funcionais RNF07 e RNF08 (primeira e segunda linhas da tabela). Isso não significa necessariamente que há um problema, já que, em termos de completude de funcionalidades de sistema, o conjunto de casos de uso do mesmo devem atender necessariamente aos requisitos funcionais.

Tabela 4.6 - Resultados da Matriz de Rastreabilidade na versão 9 do Projeto

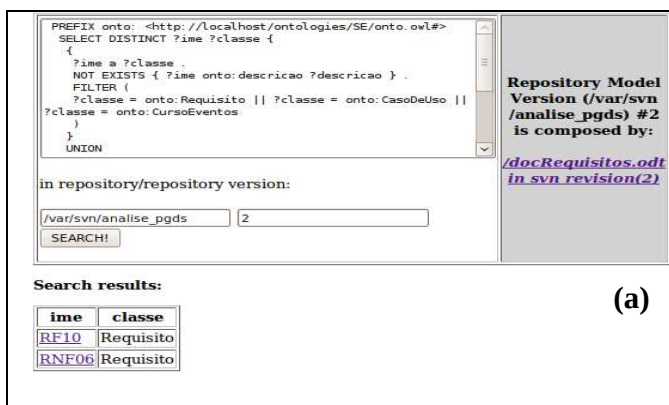
Projeto	Módulo	Caso de Uso	Requisito
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF12
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF13
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF14
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF15
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF16
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF03
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RF05
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF05
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF07
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF08
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RNF04

4.5.2 - Identificação de Não Conformidades relacionadas a Requisitos

Durante a evolução do projeto, a consulta voltada para a identificação de não conformidades foi aplicada para encontrar requisitos e cursos de eventos sem descrição associada e casos de uso sem descrição ou sem relação com requisitos (vide seção 4.3.3). Essa consulta foi

efetuada sobre todas as versões do RDS, procurando por possíveis não conformidades na documentação do projeto. Foram encontradas não conformidades em cinco versões do RDS, a saber: 2, 3, 4, 6 e 8. As versões 2 e 4 são discutidas a seguir.

A versão 2 do RDS contava com apenas 1 documento: /docRequisitos.odt na versão 2, um Documento de Requisitos. A Figura 4.16 apresenta o resultado dessa consulta.

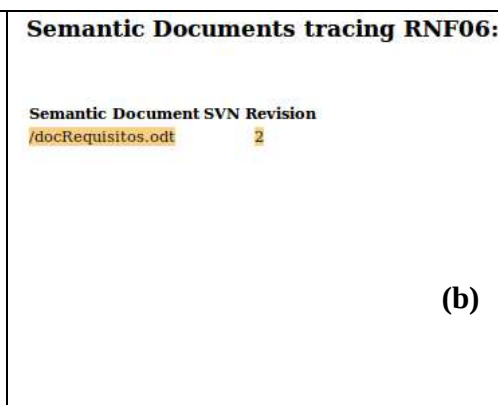


Repository Model Version (/var/svn/analise_pgds) #2 is composed by:
/docRequisitos.odt in svn revision(2)

Search results:

ime	classe
RF10	Requisito
RNF06	Requisito

(a)



(b)

Figura 4.16 – Resultado da Consulta de Não Conformidades na versão 2 (a) e rastreabilidade para os documentos relativos a um dos itens do resultado (b).

O resultado apresentado na Figura 4.16(a) mostra dois requisitos sem descrição nessa versão do RDS, a saber: RF10 e RNF06. Ao visualizar o resultado, o usuário pode selecionar uma instância qualquer para visualizar a rastreabilidade do mesmo para os documentos semânticos e as versões específicas dos mesmos que se referem à instância selecionada. A Figura 4.16(b) ilustra essa funcionalidade aplicada ao requisito não funcional RNF06. Com isso a plataforma consegue oferecer uma rastreabilidade de instância para documento físico, além de facilitar o alcance ao documento e versão onde foi originado o problema.

Na versão 4 do RDS, um problema dessa mesma natureza foi encontrado. A Figura 4.17 apresenta o resultado encontrado. O formulário de consulta do MBR foi removido dessa figura para deixá-la mais simples.

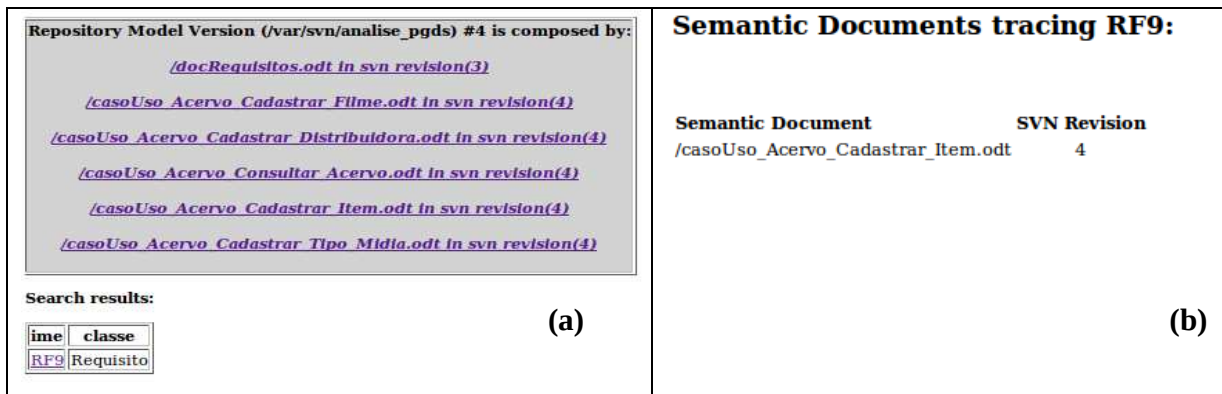


Figura 4.17 - Resultado da Consulta de Não Conformidades na versão 4 (a) e rastreabilidade para os documentos relativos a um dos itens do resultado (b).

O resultado dessa consulta aponta a existência de um requisito que possui uma não conformidade, a saber, RF9. Aparentemente o problema é que esse requisito não possui uma descrição. Contudo, observando o identificador do requisito, é possível notar que o mesmo não está dentro do padrão estabelecido pelo NEMO, que pede dois dígitos para numeração do identificador.

Na parte superior da Figura 4.17(a) observa-se a lista das versões de documentos semânticos usados na consulta, onde, são listados seis itens. Ou seja, a consulta foi aplicada ao grafo de união referentes ao conteúdo semântico dessas 6 versões. Para saber qual desses documentos introduziu o problema, o usuário deve selecionar o item na lista de resultados (RF9) e a página resultante apresenta o documento semântico e a versão do mesmo no RDS que referenciou esse item. Nesse caso, o documento em questão é o documento de descrição de caso de uso *casoUso_Acervo_Cadastrar_Item.odt* da versão 4 do RDS.

Fazendo uma análise do documento apresentado, é possível dizer que o responsável pela elaboração do documento referenciou o requisito de forma errada: não existe requisito RF9 no documento de requisitos do projeto e sim um requisito com identificador RF09, ou seja, houve um erro do responsável pela elaboração desse documento no momento em que ele informou o identificador do requisito. Dessa forma foi possível corrigir o erro, editando o documento *casoUso_Acervo_Cadastrar_Item.odt* listado no resultado.

4.5.3 - Notificação de Mudança

À medida que um projeto de software evolui, a quantidade de informações acerca do mesmo aumenta. As alterações ocorridas durante sua evolução podem trazer novas informações que, por conseguinte, podem mudar a direção de alguma atividade dentro desse mesmo projeto. Nesse contexto, a visibilidade de alterações ocorridas é fundamental e pode apoiar tanto a resolução de possíveis problemas que surgem à medida que o projeto é executado quanto a melhoria do entendimento do projeto como um todo. O MBR provê funcionalidades para assinatura sobre as alterações de uma instância existente no RD. Qualquer recurso humano pode usar esse mecanismo para se manter atualizado quanto às alterações ocorridas em uma instância de interesse no decorrer do projeto.

No cenário de aplicação conduzido no NEMO, um dos desenvolvedores da equipe que executa o projeto em questão é alocado para implementar o caso de uso Efetuar_Pagamento. Esse caso de uso possui um curso de eventos denominado Incluir_Pagamento. Para acompanhar a evolução desse caso de uso, o desenvolvedor deve utilizar o MBR para assinar seu interesse por alterações ocorridas tanto no caso de uso quanto nesse curso de eventos.

Até a versão 8 as descrições e relações dessas instâncias de caso de uso e curso de eventos eram as seguintes:

- Efetuar_Pagamento:
 - o Descrição: “Este caso de uso é responsável pela realização de um pagamento em dinheiro e consulta a dados de pagamentos.”;
 - o Requisitos Atendidos: RF05, RNF01, RNF04.
- Incluir_Pagamento:
 - o Descrição:
 - “1- De posse do valor a ser pago, o atendente informa a forma de pagamento (somente dinheiro);
 - 2- O pagamento é registrado.”

Com a evolução do projeto, o documento em questão foi modificado, mais precisamente as instâncias citadas acima. A nova versão do documento passou a ter os seguintes dados para as instâncias em questão:

- Efetuar_Pagamento:
 - o Descrição: “Este caso de uso é responsável pela realização de um pagamento e consulta a dados de pagamentos.”
 - o Requisitos: RF05, RNF01, RNF04, RNF05.

- Incluir_Pagamento:
 - o Descrição:
 - “1. De posse do valor a ser pago, o atendente informa a forma de pagamento.
 - 2. Efetuar o pagamento
 - 2a. Em dinheiro
 - 2b. Em cheque
 - 2c. Em cartão
 - 3. O pagamento é registrado.”

Ou seja, as descrições dos cursos de eventos foram modificadas e o caso de uso passou a atender a um novo requisito (RNF05). Por ocasião do *checkin* da versão que efetivou essas modificações (versão 9), um email é enviado pelo MBR aos desenvolvedores que assinam a notificação de mudança nas instâncias contempladas naquela versão. A Figura 4.18 ilustra parte do corpo da mensagem de email recebido pelo desenvolvedor que assinou a notificação sobre mudanças nos elementos acima descritos.

```
(CHANGED) http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento
property: http://localhost/ontologies/SE/onto.owl#atendeRequisito
from: http://localhost/ontologies/SE/onto.owl#RF05 http://localhost/ontologies/SE/onto.owl#RNF01 http://localhost/ontologies/SE/onto.owl#RNF04
to: http://localhost/ontologies/SE/onto.owl#RF05 http://localhost/ontologies/SE/onto.owl#RNF01 http://localhost/ontologies/SE/onto.owl#RNF04 http://localhost/ontologies/SE/onto.owl#RNF05

(CHANGED) http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento
property: http://localhost/ontologies/SE/onto.owl#descricao
from: Este caso de uso é responsável pela realização de um pagamento em dinheiro e consulta a dados de pagamentos.
to: Este caso de uso é responsável pela realização de um pagamento e consulta a dados de pagamentos.

(CHANGED) http://localhost/ontologies/SE/onto.owl#Incluir_Pagamento
property: http://localhost/ontologies/SE/onto.owl#descricao
from: De posse do valor a ser pago, o atendente informa a forma de pagamento (somente dinheiro) O pagamento é registrado.
to: De posse do valor a ser pago, o atendente informa a forma de pagamento. Efetuar o pagamento:2a. Em dinheiro2b. Em cheque2c. Em cartãoO pagamento é registrado.
```

Figura 4.18 – Corpo da Mensagem de Notificação de Mudança de Instância

A mensagem de notificação contém informações importantes acerca das mudanças realizadas nos indivíduos modificados. Os novos valores (da versão mais recente – 9) e os valores antigos (da versão anterior – 8) das propriedades descrição (*descricao*) e atendimento a requisitos (*atendeRequisito*) são listados. Essas informações podem ajudar o desenvolvedor a entender as modificações da nova versão.

4.5.4 - Rastreabilidade Evolutiva de Instância e de Documento Semântico

Os artefatos produzidos durante a execução de um projeto de software evoluem à medida que o projeto é executado. Em relação aos documentos produzidos nesse projeto, a visibilidade da evolução dos mesmos pode ser uma tarefa enfadonha, já que os sistemas de controle de versão convencionais não oferecem mecanismos para visualizar modificações de um documento, exceto arquivos de texto. Nesse contexto, avaliar as diferenças capturadas provê exatamente essa visibilidade. O MBR oferece um serviço de rastreabilidade evolutiva para os documentos semânticos contemplados na plataforma.

Durante a execução do projeto, um desenvolvedor envolvido no projeto pode desejar conhecer todos os estados pelos quais um certo elemento de um documento passou até o momento. Para tal, o desenvolver pode realizar uma consulta referente à evolução desse elemento a partir do uso do MBR. A Figura 4.19 mostra o formulário para essa consulta.

O formulário, intitulado "Individual Traceability (Graph):", contém os seguintes campos e botões:

- Label: "Individual:"
- Input field: "to.owl#Efetuar_Pagamento"
- Label: "Repository:"
- Input field: "/var/svn/analise_pgds"
- Label: "(from revision / to revision):"
- Input field: "8"
- Input field: "9"
- Button: "TRACE!"

Figura 4.19 - Formulário de pesquisa de rastreabilidade evolutiva de um indivíduo.

O primeiro campo do formulário deve ser preenchido com o indivíduo que se deseja conhecer a evolução, no exemplo o identificador do caso de uso Efetuar_Pagamento. O segundo campo se refere ao RDS em que os documentos serão pesquisados (/var/svn/analise_pgds). O terceiro e quarto campos são, respectivamente, as versões inicial e final do RDS, no exemplo a pesquisa é aplicada ao conteúdo das versões 8 e 9 do RDS. Os resultados da consulta apresentam dois conjuntos de informações para cada versão encontrada que referencia a instância: (i) propriedades e valores da instância na versão e (ii) instâncias e propriedades que possuem como valor a instância pesquisada. Os resultados da consulta da Figura 4.19 são sumarizados na Figura 4.20.

Properties and Values of http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento		Properties and Values of http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento	
property	value	property	value
alocadoPara	Atendimento a Cliente	alocadoPara	Atendimento a Cliente
artefatoProduzidoEm	Video Locadora Passatempo	artefatoProduzidoEm	Video Locadora Passatempo
atendeRequisito	RF05	atendeRequisito	RF05
atendeRequisito	RNF01	atendeRequisito	RNF01
atendeRequisito	RNF04	atendeRequisito	RNF04
atendeRequisito	RNF05	atendeRequisito	RNF05
casoDeUsoPossuiCurso	Consultar Pagamento	casoDeUsoPossuiCurso	Consultar Pagamento
casoDeUsoPossuiCurso	Incluir Pagamento	casoDeUsoPossuiCurso	Incluir Pagamento
casoDeUsoPossuiCurso	altexc Incluir Pagamento	casoDeUsoPossuiCurso	altexc Incluir Pagamento
casoDeUsoPossuiCurso	altvar Incluir Pagamento	casoDeUsoPossuiCurso	altvar Incluir Pagamento
descricao	Este caso de uso é responsável pela realização de um pagamento em dinheiro e consulta a dados de pagamentos.	descricao	Este caso de uso é responsável pela realização de um pagamento e consulta a dados de pagamentos.
type	Artefato	type	Artefato
type	CasoDeUso	type	CasoDeUso
type	Thing	type	Thing
sameAs	Efetuar Pagamento	sameAs	Efetuar Pagamento

(a)

Objects and Properties referring to http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento		Objects and Properties referring to http://localhost/ontologies/SE/onto.owl#Efetuar_Pagamento	
object	property	object	property
Atendimento a Cliente	moduloPossui	Atendimento a Cliente	moduloPossui
Consultar Pagamento	cursoDeCasoDeUso	Consultar Pagamento	cursoDeCasoDeUso
Efetuar Pagamento	sameAs	Efetuar Pagamento	sameAs
Incluir Pagamento	cursoDeCasoDeUso	Incluir Pagamento	cursoDeCasoDeUso
RF05	atendidoPorCasoUso	RF05	atendidoPorCasoUso
RNF01	atendidoPorCasoUso	RNF01	atendidoPorCasoUso
RNF04	atendidoPorCasoUso	RNF04	atendidoPorCasoUso
RNF05	atendidoPorCasoUso	RNF05	atendidoPorCasoUso
Video Locadora Passatempo	produzArtefato	Video Locadora Passatempo	produzArtefato
altexc Incluir Pagamento	cursoDeCasoDeUso	altexc Incluir Pagamento	cursoDeCasoDeUso
altvar Incluir Pagamento	cursoDeCasoDeUso	altvar Incluir Pagamento	cursoDeCasoDeUso

(c)

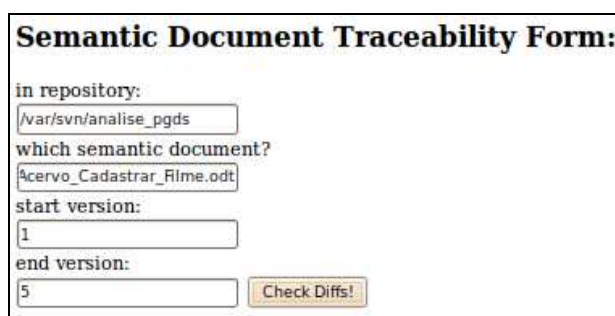
(b)

(d)

Figura 4.20 – Resultado da Pesquisa de rastreabilidade evolutiva do caso de uso Efetuar_Pagamento: (a) e (b) propriedades e valores da instância nas versões 8 e 9 respectivamente; (c) e (d) instâncias e propriedades que se relacionam com o indivíduo nas versões 8 e 9 respectivamente.

As figuras 4.20(a) e 4.20(b) apresentam as propriedades e valores do indivíduo Efetuar_Pagamento nas versões 8 e 9, respectivamente. Entre as duas versões pode-se observar que a quantidade de requisitos atendidos aumentou e que a propriedade *descrição* foi modificada. Além disso, o resultado mostra a hierarquia de classes a que essa instância pertence: a mesma é um *Artefato*, um *CasoDeUso* e um *Thing* (classe básica definida como superclasse de todas as classes em OWL). Já as Figuras 4.20(c) e 4.20(d) apresentam as instâncias e propriedades que referenciam o indivíduo Efetuar_Pagamento nas versões 8 e 9, respectivamente. Com essa pesquisa o desenvolvedor pode verificar as informações sob um contexto de evolução de um indivíduo específico, detalhando como o mesmo se relaciona com outros indivíduos e quais os valores das propriedades que o mesmo possui no decorrer do projeto.

Outra forma de consulta que possui aspectos de rastreabilidade evolutiva aponta as diferenças existentes entre versões de um mesmo documento. Nesse contexto, um desenvolvedor pode querer conhecer as modificações referentes a um documento semântico contemplado pela PGDS. A Figura 4.21 apresenta o formulário de consulta correspondente, preenchido considerando o documento de descrição de caso de uso *casoUso_Acervo_Cadastrar_Filme.odt*, nas versões de 1 a 5 do repositório informado (/var/svn/analise_pgds). Os resultados dessa consulta são apresentados na Figura 4.22.



Semantic Document Traceability Form:

in repository:

which semantic document?

start version:

end version:

Figura 4.21 – Formulário de Pesquisa de Rastreabilidade Evolutiva de Documento Semântico



Results:

Repository: file:///var/svn/analise_pgds, **File:** /casoUso_Acervo_Cadastrar_Filme.odt, **Revision Start, End: (1,5)**

Revision 4

(CHANGED) http://localhost/ontologies/SE/onto.owl#Alterar_Filme

property: <http://localhost/ontologies/SE/onto.owl#descricao>
from: Similar a inclusao.
to: 1- Sistema exibe os dados do filme;2- Usuario informa os novos dados;3- Sistema verifica e registra os novos dados do filme;

Revision 3

(ADDED) <http://localhost/ontologies/SE/onto.owl#Acervo>
(ADDED) http://localhost/ontologies/SE/onto.owl#Alterar_Filme
(ADDED) http://localhost/ontologies/SE/onto.owl#altexc_Excluir_Filme
(ADDED) http://localhost/ontologies/SE/onto.owl#Cadastrar_Filme
(ADDED) http://localhost/ontologies/SE/onto.owl#Consultar_Filme
(ADDED) http://localhost/ontologies/SE/onto.owl#Excluir_Filme
(ADDED) http://localhost/ontologies/SE/onto.owl#Incluir_Filme
(ADDED) <http://localhost/ontologies/SE/onto.owl#RF09>
(ADDED) <http://localhost/ontologies/SE/onto.owl#RNF01>
(ADDED) http://localhost/ontologies/SE/onto.owl#Video_Locadora_Passatempo

Figura 4.22 - Resultados para a de Rastreabilidade Evolutiva de Documento Semântico

Dentro do intervalo de versões pesquisado (da versão 1 à versão 5), o documento em questão estava presente em duas delas: versão 3, quando foi criado; e versão 4, quando foi

modificado. Na primeira versão (*Revision 3* na Figura 4.22), pode-se observar uma série de instâncias adicionadas ao documento, que são representadas pelas linhas iniciadas com a *string* “(ADDED)”. Na versão 4, o documento foi modificado, mudando um valor da propriedade de uma instância previamente existente no documento (*Alterar_Filme*). Alterações dessa natureza são precedidas com a *string* “(CHANGED)”.

Essa pesquisa permite, portanto, um acompanhamento evolutivo do documento semântico como um todo. As instâncias referenciadas no documento são listadas quando adicionadas, removidas ou modificadas no documento.

4.6 - Conclusões do Capítulo

Este capítulo apresentou uma aplicação da Plataforma de Gerenciamento de Documentos Semânticos (PGDS) no apoio à Engenharia de Requisitos e o uso dessa aplicação durante a execução de um projeto de software conduzido no Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO). As diversas atividades relativas a esse uso foram discutidas, a saber: (i) instalação do gancho de chamada ao MEVID no RDS; (ii) instalação do servidor web com os módulos MEVID e MBR; (iii) escolha das ontologias de domínio; (iv) projeto e implementação em OWL das ontologias escolhidas; (v) reestruturação dos modelos de documento considerados para permitir anotações semânticas; (vi) anotação semântica dos modelos de documento resultantes da atividade anterior e, por fim, (vii) execução e acompanhamento de um projeto com o apoio da PGDS.

Como forma de mostrar o potencial da plataforma, foram apresentados cenários referentes ao apoio à Engenharia de Requisitos, apresentando a matriz de rastreabilidade dos requisitos do projeto em questão, bem como o apoio à identificação de algumas não conformidades durante a evolução do projeto. Adicionalmente, pesquisas relativas ao acompanhamento evolutivo tanto de indivíduos pertencentes a um projeto quanto dos documentos semânticos também foram apresentados.

O próximo capítulo apresenta as conclusões e perspectivas futuras deste trabalho, levantando algumas limitações e apontando possíveis melhorias.

Capítulo 5

Conclusões e Perspectivas Futuras

Este capítulo discute as conclusões do trabalho conduzido nesta dissertação. Os obstáculos encontrados durante a sua realização, bem como as soluções definidas são comentadas. Adicionalmente, as contribuições e limitações do trabalho são descritas. Por fim, são apresentadas perspectivas de trabalhos futuros.

5.1 – Conclusões

A documentação de um projeto vem sendo um dos principais veículos para disseminação de conhecimento acerca do mesmo entre as pessoas (ERIKSSON, 2007), auxiliando os envolvidos a buscar e trocar informações (BRUGGEMANN et al., 2000) (FORWARD; LETHBRIDGE, 2002). Quando utilizada em projetos de software, em especial aplicada junto a um processo apropriado, a documentação provê várias vantagens, tais como: melhoria da produtividade no desenvolvimento, facilidade de entendimento e apoio às atividades referentes à localização das informações acerca de projetos de software (ROCHA et al., 2001).

A aplicação de um processo de documentação de forma apropriada não é tarefa simples e envolve uma série de obstáculos, dentre eles (LETHBRIDGE et al., 2003) (ROCHA et al., 2001): vencer barreiras culturais dentro da organização, que, por vezes, levam seus membros a considerar o processo como sendo burocrático; alcançar precisão e consistência no conteúdo dos documentos gerados; facilitar a busca de informações contidas nos documentos; estabelecer relações existentes entre documentos e seus conteúdos. Adicionalmente, as informações de um projeto estão, em muitas ocasiões, espalhadas em diferentes documentos, o que dificulta tanto a integração das informações quanto a visão do projeto como um todo.

Esse conjunto de obstáculos é, por vezes, fruto de um tratamento puramente sintático dado aos documentos. Ferramentas relativas à edição de texto e sistemas de controle de versão (utilizados para centralizar e acompanhar a evolução de documentos) normalmente contemplam pouco ou nenhum conhecimento acerca do conteúdo semântico dos documentos. Nesse contexto, a visão da *web* semântica (BERNERS-LEE et al, 2001) considera uma ideia que pode auxiliar a superar esses obstáculos: unir metadados baseados em ontologias de domínio a documentos, anotando os mesmos com informação semântica. Neste contexto, o ferramental aplicado deve ser

preferencialmente transparente e automatizado, já que o processo de anotação de documentos é considerado tedioso e suscetível a erros (ERIKSSON, 2007) (UREN et al., 2006).

Uma maneira de automatizar parcialmente o processo de anotação pode ser obtida pela anotação semântica de modelos de documento (TALLIS, 2003). Uma vez anotados os modelos de documento, a instanciação de documentos baseados nesses modelos gera documentos semânticos. Organizações de software maduras normalmente dispõem de um conjunto de modelos de documento para padronizar seus documentos de software. A estratégia de anotação em modelos de documento pode ser, portanto, aplicada em modelos de documentos de software de forma a apoiar o processo de produção de documentos de software semânticos.

Somando essa estratégia (uso de documentos semânticos) ao uso de repositórios de controle de versão e a mecanismos que permitem buscas baseadas em ontologias, é possível prover vários serviços de apoio ao processo de documentação e usufruir de uma série de vantagens, apoiando a visibilidade e a rastreabilidade de informações.

Este trabalho discutiu diversos desses aspectos e, de forma geral, sua principal contribuição foi a integração de sistemas de controle de versão a um ferramental de gerenciamento de documentos semânticos, promovendo uma sinergia entre ambos, na medida em que, de um lado, adiciona-se semântica a sistemas de controle de versão; de outro, trata-se versionamento e acompanhamento de evolução ao conteúdo de documentos semânticos. Outras contribuições incluem:

- Definição de um mecanismo de anotação e extração de conteúdo semântico de documentos instanciados a partir de modelos de documento semânticos. Para tal, foi necessária a definição de uma linguagem de anotação semântica em modelos de documento;
- Associação de sistemas de controle de versão a mecanismos de estabelecimento de conteúdo semântico, unindo a visão sintática dos sistemas de controle de versão com a visão semântica dos documentos semânticos;
- Definição de uma Plataforma de Gerenciamento de Documentos Semânticos, capaz de: apoiar o processo de anotação semântica; controlar a evolução do conteúdo semântico de documentos; integrar informações contidas em diversos documentos; notificar partes envolvidas sobre alterações ocorridas durante a evolução de um repositório de

documentos semânticos e finalmente permitir consultas sobre informações extraídas desse repositório;

- Desenvolvimento, usando a Plataforma de Gerenciamento de Documentos Semânticos, de serviços de apoio a atividades da Engenharia de Requisitos de Software.

5.2 – Limitações e Perspectivas Futuras

Como qualquer trabalho de pesquisa, o presente trabalho apresenta limitações. Nesta seção, algumas limitações são discutidas e são apresentadas oportunidades de melhoria para contorná-las. Outras perspectivas futuras tratam de novas possibilidades proporcionadas pela elaboração deste trabalho, visando dar continuidade ao mesmo.

O processo de anotação semântica em modelos de documento no formato ODF é, até o momento, manual e depende de uma série de passos suscetíveis a erros. Uma forma de se contornar esse cenário é desenvolver uma extensão da ferramenta *OpenOffice Writer* para apoiar parte do processo de anotação proposto.

O uso da linguagem de anotação em modelos, juntamente com o formato ODF, foi suficiente para o propósito do trabalho. Contudo, é importante efetuar uma nova análise do formato ODF para anotações semânticas, já que a implementação do *kit* de desenvolvimento provido pelo *Open Office* está em evolução e, por isso, pode prover novas facilidades para anotações, as quais podem ser incorporadas à Plataforma de Gerenciamento de Documentos Semânticos (PGDS). Essa análise deve ser conduzida ainda levando em consideração a automatização das anotações semânticas. Uma das vantagens de se atender esse cenário é que, segundo a especificação ODF, será possível permitir que agentes externos à plataforma possam usufruir dos dados contidos nos documentos sem necessariamente acessarem a plataforma através do Módulo de Busca e Rastreabilidade (MBR).

Ainda no que se refere a anotações semânticas, a linguagem de anotação de modelos semânticos possui um grupo sintático relativamente pequeno, tratando apenas anotações em trechos de texto e tabelas. Anotações em imagens, seções e subseções de texto, por exemplo, não são contempladas na linguagem. Assim, a extensão da linguagem, provendo um ferramental de anotação mais robusto e aumentando as possibilidades de uso da Plataforma, é uma possibilidade de trabalho futuro. Neste contexto, é importante avaliar a incorporação de uma ontologia de

documento ao trabalho, com o propósito de descrever os elementos comuns a documentos de texto. Para o desenvolvimento dessa ontologia, é útil considerar as várias especificações de documento existentes (Microsoft, ODF, RDF, etc) e assim facilitar a evolução da linguagem de anotação em modelos de documento. Adicionalmente, formatos de outros tipos de documento, tais como modelos UML e gráficos, devem ser considerados em futuras extensões da ferramenta.

Apesar do crescente uso da ferramenta *Open Office*, há outros editores de texto, em especial o *Microsoft Word*, que possuem um vasto conjunto de usuários. Nesse contexto, a possibilidade de se estender outros editores, sobretudo o *Microsoft Word*, para geração de modelos semânticos é também uma perspectiva futura que aumenta o leque de oportunidades com relação ao uso de diferentes ferramentas de edição de texto.

No que se refere ao mecanismo de notificação de alterações, a Plataforma de Gerenciamento de Documentos Semânticos (PGDS) pode ser estendida de forma a prover integração com outros sistemas, tal como uma ferramenta de definição de processos, de modo subscrever itens automaticamente para desenvolvedores. Nesse contexto, o mecanismo de assinatura de alterações poderia ser incrementado, provendo assinatura automática para notificação de mudanças baseando-se no papel de um recurso humano. Por exemplo, se um desenvolvedor trabalha em um projeto, então ele deve subscrever itens no escopo do projeto. Se o desenvolvedor estiver alocado a uma atividade de design ou implementação de um caso de uso, ele pode ter subscrito automaticamente a notificação daquele caso de uso, bem como dos requisitos que o caso de uso resolve.

Com relação aos aspectos não-funcionais, o sistema deve ser revisado a fim de se compreender melhor os impactos de segurança da aplicação como um todo, bem como da eficiência da mesma sob estresse. Adicionalmente, um estudo sobre a escalabilidade da plataforma pode ser considerado em conjunto.

No estágio atual, quando um usuário assina a notificação de alteração de um indivíduo, a notificação será feita apenas quando as propriedades diretas do indivíduo assinado forem alteradas. Se um valor de uma propriedade referenciada por um indivíduo que consta como valor de uma propriedade do indivíduo assinado for alterado, existe grande possibilidade de que o usuário querer ser notificado da alteração do indivíduo intermediário também. Uma possibilidade para atender esse caso é sugerir a assinatura de indivíduos referenciados pelo indivíduo assinado. Seja o seguinte exemplo. Um usuário assina para receber notificações de alterações no caso de

uso *X*. Qualquer alteração em uma propriedade desse caso de uso gerará uma notificação ao usuário. Por exemplo, se a lista de requisitos atendidos por esse caso de uso for modificada, o usuário receberá uma notificação. Contudo, se esse caso de uso, em algum momento, possui a propriedade *atendeRequisito* com os valores RF1, RF2 e RF3 e a descrição do requisito RF3 for modificado, o usuário não receberá notificação sobre essa alteração.

Sistemas de Controle de Versão normalmente possuem mecanismos conhecidos como linhas de desenvolvimento. Em essência, as linhas de desenvolvimento permitem que desenvolvedores trabalhem nos artefatos versionados de um mesmo repositório de controle de versões de forma independente. Até o presente momento não foi considerado o tratamento de diferentes linhas de desenvolvimento dentro de um Repositório de Documento Semântico. Por ser largamente utilizado por organizações de software, o tratamento desse tipo de mecanismo deve ser objeto de trabalhos futuros.

Foi analisada a possibilidade de incrementar a plataforma no sentido de integrá-la a uma ontologia de Gerência de Configuração de Software e a uma de artefatos. No caso, a ontologia de artefatos especializaria o conceito de documento, descrevendo os possíveis elementos contidos em um documento e as relações existentes entre eles. Unindo essas duas ontologias com as ontologias de domínio e efetuando uma integração das mesmas com a plataforma seria possível elaborar consultas e inferências ainda mais ricas.

Ainda no contexto de escolha de uma ontologia que defina elementos de um documento, é importante manter o relacionamento entre os indivíduos e relacionamentos gerados a partir de uma extração e os elementos de um documento. Nesse contexto, seria possível descobrir a localização, em uma dada versão de documento semântico, que gerou efetivamente uma instância ou relacionamento específico, melhorando o nível de granularidade das consultas da plataforma.

É importante gerenciar a configuração dos modelos de documento semânticos e manter uma relação deles com os documentos semânticos instanciados a partir deles. Alterações em um modelo semântico podem acarretar não conformidades falsas em documentos semânticos elaborados a partir de versões anteriores do modelo. Se um desenvolvedor *A* usa um modelo semântico na versão *v1* e outro *B* usa o mesmo modelo na versão *v2*, há a possibilidade de inconsistências e perda de informação.

A linguagem usada para efetuar buscas, SPARQL, não é trivial e necessita de conhecimento técnico elevado. Atualmente o MBR conta apenas com uma interface para lidar

com pesquisas escritas nessa linguagem, demandando que o usuário conheça SPARQL. Um ponto de melhoria na plataforma é aperfeiçoar o MBR de forma a prover uma interface para a elaboração de consultas independente de SPARQL.

Referências Bibliográficas

- ARTZ, D., GIL, Y., A survey of trust in computer science and the Semantic Web, **Web Semantics: Science, Services and Agents on the World Wide Web archive**, Volume 5 , Issue 2, p.58-71. 2007.
- BARCELLOS, M.A., FALBO, R.A., Using a Foundational Ontology for Reengineering a Software Enterprise Ontology, In: **Proceedings** of the ER 2009 WORKSHOPS ON ADVANCES IN CONCEPTUAL MODELING - Challenging Perspectives, Gramado, Brazil, 2009. p.179-188.
- BERNERS-LEE, T., HENDLER, J., LASSILA, O., 2001. The semantic web. **Scientific American** 284 (5), 34–43.
- BREITMAN, K., **Web Semântica: A internet do futuro**. Rio de Janeiro. LTC. 2005.
- BRUGGEMANN B. M., HOLZ K.P., MOLKENTHIN F. (2000), Semantic Documentation in Engineering, In: **Proceedings** of the Eighth INTERNATIONAL CONFERENCE ON COMPUTING IN CIVIL AND BUILDING ENGINEERING, California, August 14-16, 2000.
- CAPLAN, P., **Metadata Fundamentals for all Librarians**. Chicago: American Library Association. 2003.
- CHANDRASEKARAN, B.; JOSEPHSON J.R., BENJAMINS V.R., What Are Ontologies, and Why Do We Need Them?, **IEEE Intelligent Systems**, Volume 14, Issue 1. 1999.
- COCKBURN A., **Writting Effective Use Cases**, 1st Edition, Addison-Wesley, 2001.
- COLLINS-SUSSMAN, B., FITZPATRICK, B., PILATO, C. M., **Version Control With Subversion**, 2008. Disponível em: <<http://svnbook.red-bean.com/en/1.5/svn-book.pdf>>. Acesso em: 21 de julho de 2010.

- DURUSAU, P., BRAUER, M., **Open Document Format for Office Applications (OpenDocument) Version 1.2**. 2009. Disponível em: <<http://docs.oasis-open.org/office/v1.2/part1/cd04/OpenDocument-v1.2-part1-cd04.pdf>>. Acesso em: 21 de julho de 2010.
- ERIKSSON H., BANG M., Towards document repositories based on semantic documents, **Proceedings** of Sixth INTERNATIONAL CONFERENCE ON KNOWLEDGE MANAGEMENT AND KNOWLEDGE TECHNOLOGIES, I-KNOW '06. 2006.
- ERIKSSON, H., An annotation tool for semantic documents. In: **Proceedings** of the 4th EUROPEAN CONFERENCE ON THE SEMANTIC WEB: RESEARCH AND APPLICATIONS. p: 759 – 768. 2007b.
- ERIKSSON, H., The semantic-document approach to combining documents and ontologies, **International Journal of Human-Computer Studies** Volume 65 , Issue 7. 2007a.
- FALBO, R. A., NARDI, J.C., Evolving a Software Requirements Ontology, 2008, Santa Fe. **Anales** de XXXIV CONFERENCIA LATINOAMERICANA DE INFORMÁTICA - CLEI'2008, p. 300-309. 2008.
- FALBO, R.A., RUY, F.B., PEZZIN, J., DAL MORO, R.; Ontologias e Ambientes de Desenvolvimento de Software Semânticos, **Actas** de las IV JORNADAS IBEROAMERICANAS DE INGENIERÍA 2004.
- FENSEL, D., HENDLER, J.A., LIEBERMAN, H., WAHLSTER W. **Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential**. Mit Press. 2003
- FORWARD, A., LETHBRIDGE, T. C. The relevance of software documentation, tools and technologies: a survey. In: **Proceedings** of the 2002 ACM SYMPOSIUM ON DOCUMENT ENGINEERING. DocEng '02 (2002)
- GRUBER, T.R., Towards Principles for the Design of Ontologies used for Sharing. **International Journal of Human-Computer Studies**, v. 43, n. 5/6, 1995.

- GUARINO, N. Formal Ontology and Information Systems. In: Formal Ontologies in Information Systems, N. Guarino (Ed.), **IOS Press**, 3 -15, 1998
- GUIZZARDI G. On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta) Models. In: **Proceeding** of the 2007 CONFERENCE ON DATABASES AND INFORMATION SYSTEMS IV. 2007.
- GUIZZARDI, G., **Ontological Foundations for Structural Conceptual Models**, Universal Press, The Netherlands, 2005.
- HAPPEL, H.; SEEDORF, S.; Applications of Ontologies in Software Engineering, In: **Proceedings** of 2nd INTERNATIONAL WORKSHOP ON SEMANTIC WEB ENABLED SOFTWARE ENGINEERING (SWESE 2006), held at the 5th International Semantic Web Conference (ISWC 2006). 2006.
- HORROCKS, I., PATEL-SCHNEIDER, P.F., BECHHOFFER, S., TSARKOV, D., OWL rules: A proposal and prototype implementation, **Journal of Web Semantics: Science, Services and Agents on the World Wide Web** 3. 2005.
- HORROCKS, I., PATEL-SCHNEIDER, P.F., Harmelen, F.V., From SHIQ and RDF to OWL: The Making of a Web Ontology Language, **Journal of Web Semantics**. Volume 1. 2003.
- IEEE, **IEEE/ANSI 830-1998**, IEEE Recommended Practice for Software Requirements Specifications, 1998.
- INTERNATIONAL STANDARDIZATION ORGANIZATION, **ISO 10007**, Sistemas de gestão da qualidade – Diretrizes para a gestão de configuração. 2005.
- INTERNATIONAL STANDARDIZATION ORGANIZATION, **ISO/IEC 12207**, Systems and software engineering – Software life cycle processes. 2008.
- INTERNATIONAL STANDARDIZATION ORGANIZATION, **ISO/IEC 26300**, Open Document Format for Office Applications (OpenDocument) v1.0. 2006.

- KANELLOPOULOS D. N., KOTSIANTIS S. B., Semantic Web: a state of the art survey, **International Review on Computer and Software (I.RE.CO.S)**, Vol 2, n. 5, September 2007.
- KIRYAKOV, A., POPOV, B., OGNJANOFF, D., MANOV, D., KIRILOV, A. e GORANOV, M. Semantic annotation, indexing, and retrieval. In **Proceedings** of the 2nd INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC2003). 2003.
- KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering: Process and Techniques. 1st Edition, Wiley, 1998.
- LETHBRIDGE, T. C., SINGER, J., FORWARD, A. How Software Engineers Use Documentation: The State of the Practice. **IEEE Software**, vol. 20, no. 6, pp. 35-39, Nov./Dec. 2003, doi:10.1109/MS.2003.1241364
- MCGUINNESS, D.L., HARMELEN, F. **OWL Web Ontology Language**, Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em 22/04/2010.
- MURTA, Leonardo Gresta Paulino, **Gerência de Configuração no Desenvolvimento Baseado em Componentes**. Tese de Doutorado, COPPE/UFRJ, 2006.
- NARDI, Júlio César. **Apoio de Gerência de Conhecimento à Engenharia de Requisitos em um Ambiente de Desenvolvimento de Software**. Dissertação de Mestrado. Universidade Federal do Espírito Santo, Vitória, 2006.
- NEŠIĆ, S., GAŠEVIĆ, D., JAZAYERI, M.. Semantic Document Management for Collaborative Learning Object Authoring. In **Proceedings** of the 8th IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, Santander, Spain, 2008.
- NEŠIĆ, S., JAZAYERI, M., JOVANOVIĆ, J., GAŠEVIĆ, D. Ontology-Based Content Model for Scalable Content Reuse, In **Proceedings** of the 4th ACM SIGART INTERNATIONAL CONFERENCE ON KNOWLEDGE CAPTURE, Whistler, Canada, 2007.

- NGUYEN, T. N. Object-Oriented Software Configuration Management. In: **Proceedings** of the 22nd IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE (ICSM'06). 2006.
- NUNES, Vanessa Battestin. **Integrando Gerência de Configuração de Software, Documentação e Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software**. Dissertação de Mestrado. Universidade Federal do Espírito Santo, Vitória, 2005.
- PRESSMAN R.S., **Engenharia de Software**, 6ª Edição, McGraw Hill, 2006.
- PRUD'HOMMEAUX, E., SEABORNE, A., SPARQL Query Language for RDF. 2008. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 21 de julho de 2010.
- PULIDO, J.R.G., RUIZ, M.A.G., HERRERA, R., CABELLO, E., LEGRAND, S., ELLIMAN, D., Ontology languages for the semantic web: A never completely updated review. **Knowledge-Based Systems** 19 (2006) 489–497.
- RDF, Resource Description Framework. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em 22/04/2010.
- ROBERTSON, S.; ROBERTSON, J. **Mastering the Requirements Process**. 1st Edition. ACM Press, Addison Wesley, 1999.
- ROCHA, A.R., Maldonado, J.C., Weber, K.C., **Qualidade de Software: Teoria e Prática**, 1.a Edição, Prentice Hall, 2001.
- ROSENBERG D., STEPHENS M., **Use Case Driven Object Modeling with UML: Theory and Practice**, Apress, 2007.
- SÁNCHEZ-FERNÁNDEZ, L., FERNÁNDEZ-GARCÍA, N., The Semantic Web: Fundamentals and A Brief State-of-the-Art. **Upgrade Journal**. Vol. VI, Issue n. 6, December. 2005.
- SCHNEIDER G., WINTERS J. P., **Applying Use Cases, Second Edition: A Practical Guide**, Addison Wesley, 2001.

- SEFTON, P., BARNES, I., WARD, R., DOWNING, J., Embedding Metadata and Other Semantics in Word processing Documents, **The International Journal of Digital Curation**, Issue 2, Volume 4. 2009.
- SILVA, P. P., MCGUINNESS, D., FIKES, R., A proof markup language for semantic web services, **Information Systems**, Vol. 31, n. 4-5, pp. 381-395, 2006.
- SOMMERVILLE I., **Engenharia de Software**, 6a Edição, Addison-Wesley, 2003.
- UREN, V., CIMIANO, P., IRIA, J., HANDSCHUH, S., VARGAS-VERA, M., MOTTA, E., CIRAVEGNA, F., Semantic Annotation for Knowledge Management: Requirements and a survey of the state of the art. **Journal of Web Semantics: Science, Services and Agents on the World Wide Web** 4, 14-28, 2006.
- WIEGERS, K. E. **Software Requirements: Pratical techniques for gathering and managing requirements throughout the product development cycle**. Microsoft Press, Second Edition, Redmond, Washington, 2003.

Apêndice A – Projeto da Plataforma de Gerenciamento de Documentos Semânticos

O projeto da arquitetura da PGDS foi conduzido segundo o padrão MVC (*Model-View-Controller*) (FOWLER, 2003) associado ao padrão Camada de Serviço (*Service Layer*) (FOWLER, 2003). O elemento *Model* – representado pelos pacotes com nome “*model*” – contempla as classes de domínio do problema (pacote “*domain*”) e regras de negócio (pacote “*service*”). O elemento *Controller* – representado pelos pacotes com nome “*controller*” – contempla as classes responsáveis por fazer a interface entre o modelo (*Model*) e a visão (elemento *View*). O elemento *View* – representado pelos pacotes com nome “*view*” – é composto pelas classes e arquivos de interface com o usuário. Outros padrões de projeto utilizados são o padrão DAO (*Data Access Object*) e Injeção de Dependência (*Dependency Injection* ou *IoC - Inversion of Control*).

Seguindo a orientação do padrão MVC, os diferentes módulos da plataforma contém classes relacionadas às camadas desse padrão, segundo o esquema mostrado na Figura A.1.

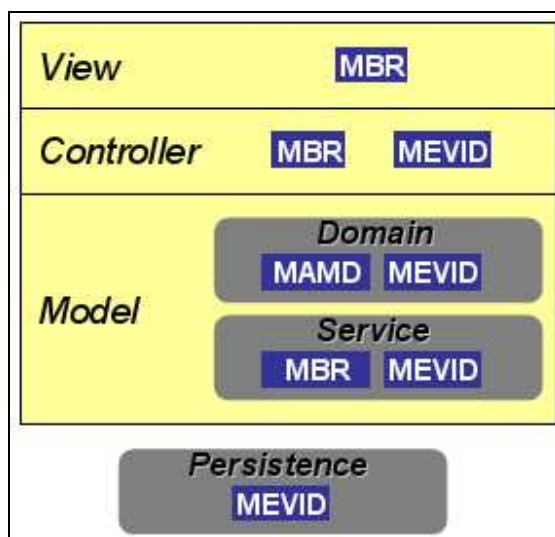


Figura A.1 – Relação do Padrão MVC e os Módulos da Plataforma

A camada de visão (*View*) é provida pelo MBR, uma vez que, na versão corrente, é o único módulo por meio do qual usuários podem acessar os serviços da plataforma. A camada de persistência é provida pelo MEVID, uma vez que o repositório de dados é parte desse módulo. No que se refere à camada de modelo (*Model*), uma vez que o padrão Camada de Serviço

(FOWLER, 2003) é utilizado, essa camada possui dois tipos de classes: classes de domínio do problema e classes provedoras de serviços. Tanto o MAMD quanto o MEVID possuem classes de domínio do problema. Já os serviços providos pela plataforma são disponibilizados pelo MEVID e pelo MBR. Por fim, o MBR provê controladores (*Controller*) para controlar a interação, enquanto o MEVID possui um controlador responsável por atender requisições efetuadas pelo gancho instalado no RDS.

Os módulos da plataforma se comunicam entre si, estabelecendo, portanto, relações de dependência entre eles. A Figura A.2 apresenta essas relações de dependência. O Módulo de Anotação em Modelo de Documento (pacote *mamd*) define a estrutura necessária para o tratamento de documentos semânticos. Essa estrutura é utilizada pelo módulo central do sistema – o Módulo de Extração, Versionamento e Integração de Dados (pacote *mevid*) – com o intuito de registrar informações acerca dos documentos semânticos tratados pelo RDS. O Módulo de Busca e Rastreabilidade (pacote *mbr*), por sua vez, utiliza um componente do MEVID – o Repositório de Dados (RD) – para efetuar buscas escritas na linguagem SPARQL sobre o conteúdo semântico do RD

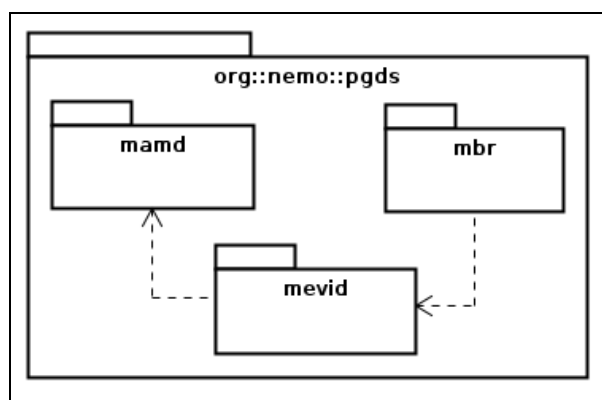


Figura A.2 - Diagrama de dependência entre os módulos da PGDS

A seguir, o projeto de cada um desses módulos é detalhado.

3.6.1 - Módulo de Anotação em Modelos de Documento (MAMD)

O Módulo de Anotações em Modelos de Documento (MAMD) é responsável por manter informações sobre os documentos semânticos tratados na plataforma. Uma vez que neste trabalho não foi desenvolvida uma ferramenta efetivamente para anotar os modelos de documento, o pacote *mamd* conta apenas com o pacote de modelo (*mamd::model*). Esse pacote é subdividido em

dois: (*mamd::model::domain* e *mamd::model::odt*). O primeiro se responsabiliza por definir contratos (interfaces) relativos aos documentos semânticos da plataforma. O pacote *odt* se responsabiliza por implementar o domínio, focando em arquivos de texto no formato ODT (*Open Document Text*, o formato de documentos de texto de ODF). A Figura A.3 apresenta o diagrama de classes do domínio do problema do pacote *domain*.

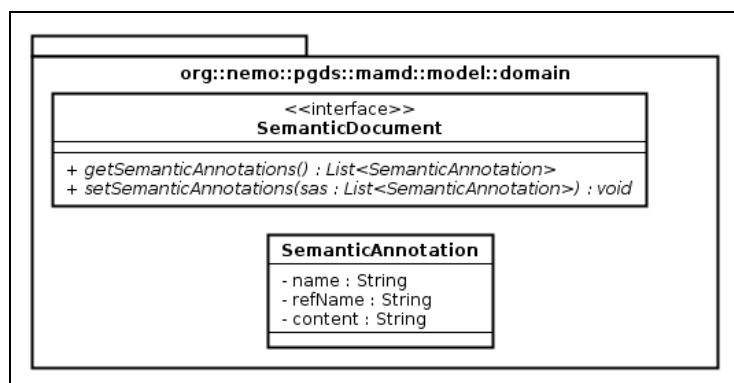


Figura A.3 - Diagrama de Classes do pacote *mamd::model::domain*

Basicamente, a interface *SemanticDocument* provê o contrato de representação de documentos semânticos na PGDS e de acesso às suas anotações semânticas (representadas pela classe *SemanticAnnotation*). Cada instância de *SemanticAnnotation* registra o nome da anotação semântica utilizada no documento (atributo *name*), a variável de referência para o campo escondido (atributo *refName*) e o conteúdo da anotação semântica (atributo *content*), que contém as instruções utilizadas para a geração do conteúdo semântico.

Por ser implementada como uma interface, cada formato de arquivo considerado para extração de conteúdo semântico deve implementar a interface *SemanticDocument* para usufruir das funcionalidades da plataforma.

Como citado anteriormente, neste trabalho foi utilizado o formato ODT para criação de modelos de documentos semânticos. A Figura A.4 apresenta o diagrama de classes do pacote *mamd::domain::odt*, responsável por definir a implementação de documentos semânticos no formato ODT, representado pela classe *OdtSemanticDocument*.

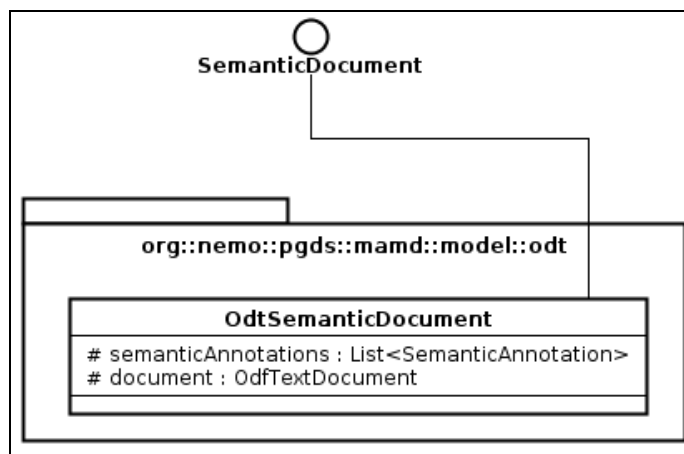


Figura A.4 - Diagrama de Classes de Domínio do pacote *mamd::model::domain::odt*

A classe *OdtSemanticDocument* implementa o protocolo fornecido pela interface *SemanticDocument* e serve de ponte entre o MAMD e documentos ODT. Essa relação é feita através do atributo *document* do tipo *OdfTextDocument*, pertencente ao *framework* ODFDOM, responsável por prover ferramental de leitura e escrita em documentos ODT.

3.6.2 - Módulo de Extração, Versionamento e Integração de Dados (MEVID)

As classes que modelam o domínio de MEVID são encontradas no pacote *mevid::model::domain*. Para efetuar o registro e acesso em banco de dados das classes definidas para esse módulo, há classes que seguem o padrão DAO no pacote *mevid::model::persistence*. Já os serviços oferecidos por esse módulo foram organizados no pacote *mevid::model::service* e a fronteira com o mundo externo é provida por elementos do pacote *mevid::controller*. A Figura A.5 mostra o diagrama de classes de domínio do pacote *mevid::model::domain*.

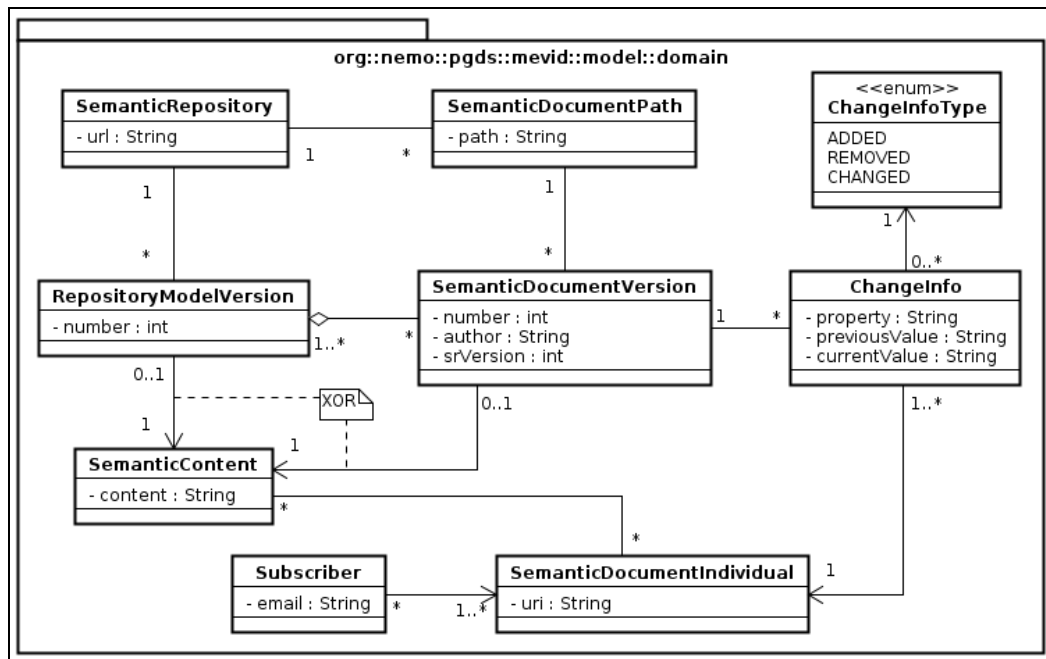


Figura A.5 - Diagrama de Classes de Domínio do pacote *mevid::model::domain*

A classe *SemanticRepository* representa o Repositório de Documentos Semânticos (RDS) tratado na plataforma. O atributo *url* define a localidade física do RDS oferecida pelo gancho *post-commit* definido no próprio RDS. Um RDS contém uma série de documentos semânticos, relacionados por caminhos de arquivo de sistema, representados pela classe *SemanticDocumentPath*. Durante o ciclo de vida de um RDS, a evolução de cada documento semântico contemplado pelo RDS deve ser tratada. A classe *SemanticDocumentVersion* comporta as informações relativas à versão de um documento semântico contido no RDS. O autor responsável pela geração da versão (fornecido pelo *Subversion*), a numeração da versão dentro do Repositório de Dados e a versão de referência no RDS são registradas pelos atributos *author*, *number* e *srVersion*, respectivamente.

O resultado da extração do conteúdo semântico de uma versão de documento semântico é registrado pela classe *SemanticContent* que, em essência, registra o grafo RDF gerado pelo processo de extração de dados do documento semântico. Cada versão de documento semântico gerado, quando essa versão não é relativa à remoção do documento do RDS, produz, portanto, um grafo RDF, que, por sua vez, é composto por triplas provenientes da instanciação de conceitos e relações definidas nas ontologias usadas no documento semântico. Para acompanhamento de alterações referentes a um indivíduo e para permitir acessá-lo facilmente, é

importante manter uma referência para o mesmo em cada versão de documento semântico que referencia aquele indivíduo. Para atender tal cenário, a classe *SemanticDocumentIndividual* foi definida. Todo indivíduo de um grafo RDF é representado por um endereço *web*, representado pelo atributo *url* dessa classe.

Instâncias da classe *ChangeInfo* representam as alterações ocorridas na geração de uma versão de documento semântico. Cada instância dessa classe representa um fato referente a uma alteração ocorrida em um indivíduo contemplado naquela versão. Uma alteração pode ser referente à adição e remoção do indivíduo naquela versão ou referente a uma alteração de propriedades daquele indivíduo. Quando ocorre uma alteração de propriedade referente a um indivíduo existente em duas versões consecutivas, registram-se a propriedade modificada (atributo *property*), os valores anteriores dessa propriedade (atributo *previousValue*) e os valores da versão corrente (atributo *currentValue*). Os possíveis tipos de modificação feitos em um indivíduo em uma versão de documento semântico são registrados pela classe de enumeração *ChangeInfoType*.

Ao final do processo de extração de cada versão de documento semântico relacionado na versão gerada no RDS, o conjunto de indivíduos modificados (relacionados a alguma instância de *ChangeInfo*) é guardado para referências posteriores. De posse desses indivíduos, os interessados (representados pela classe *Subscriber*) são notificados com base em seus endereços de *email*. Dessa forma pode-se apresentar a modificação (instância de *ChangeInfo*) e a versão de documento semântico que gerou a modificação (instância de *SemanticDocumentVersion*).

A última funcionalidade efetuada com a geração de uma nova versão no RDS é a de integração dos conteúdos semânticos do RDS. Para tal, deve-se agregar os conteúdos semânticos do RDS em suas últimas versões, unir os mesmos e inferir novos fatos baseados no grafo final. A classe *RepositoryModelVersion* comporta essa informação.

O pacote *mevid::model::persistence* comporta as classes responsáveis por registrar e acessar instâncias das classes definidas em *mevid::model::domain* em um mecanismo de persistência. Na materialização da PGDS, o mecanismo utilizado foi o de banco de dados relacional, usando o sistema de gerenciamento de banco de dados PostgreSQL.

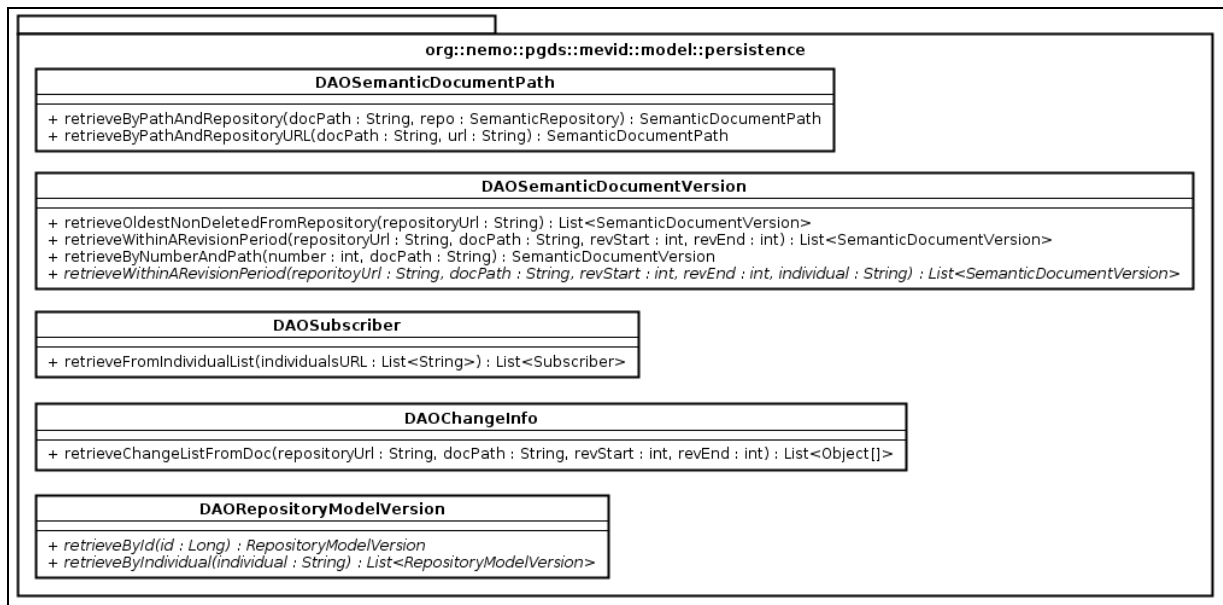


Figura A.6 - Modelo de Classes do pacote *mevid::model::persistence*

Conforme advoga o padrão DAO, cada classe definida nesse pacote possui uma interface que define o contrato de métodos que devem ser implementados para o funcionamento efetivo da persistência de dados da plataforma. A Figura A.6 exhibe as classes de persistência definidas no MEVID. Para efeito de simplicidade as interfaces relativas a cada classe e métodos de registro (inserção no banco de dados) foram omitidas.

A classe *DAOSemanticDocumentPath* implementa os métodos relativos à busca de instâncias de *SemanticDocumentPath* registrados no RD. As buscas podem ser feitas com base no caminho do documento (atributo *path* da classe) e no repositório de documentos semânticos (acessíveis pelo atributo *url* da classe *SemanticRepository*). A classe *DAOSemanticDocumentVersion* implementa métodos relativos à busca de instâncias de *SemanticDocumentVersion*, efetuando pesquisas como recuperação das últimas versões de cada versão de documento semântico não removido, recuperação de listagem de conteúdo semântico dentro de uma faixa de versões dos mesmos e, finalmente, recuperação com base na versão e caminho do documento referenciado pela versão de documento semântico. *DAOSubscriber* é responsável por obter uma lista de interessados com base em uma lista de indivíduos modificados em uma versão gerada no RDS. A classe *DAOChangeInfo* permite a recuperação de instâncias de *ChangeInfo* com base em um documento semântico (*SemanticDocumentPath*) e em uma faixa de versões do mesmo. Dessa forma obtêm-se o histórico de modificações ocorridas em um

documento semântico. Por fim, *DAORepositoryModeVersion* é responsável por recuperar instâncias da classe *RepositoryModelVersion* segundo um indivíduo (*SemanticDocumentIndividual*). Com esse mecanismo é possível verificar a evolução do indivíduo no RDS.

Todos os serviços (extração, versionamento, notificação e integração de dados) do MEVID são providos pelo pacote *mevid::model::service*. A Figura A.7 mostra os subpacotes contidos nesse pacote e suas relações de dependência.

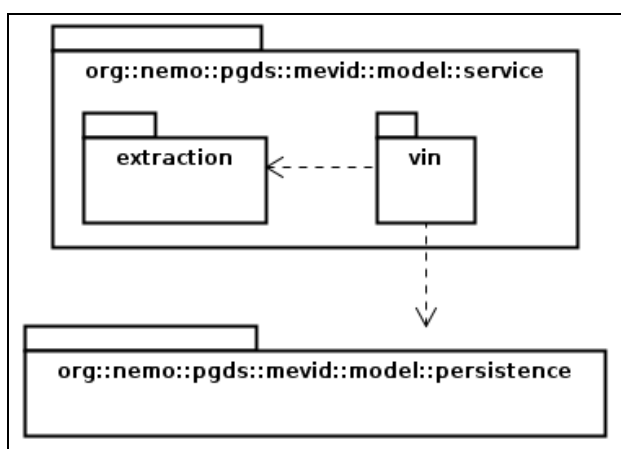


Figura A.7 - Diagrama de Pacotes do pacote *mevid::model::service*

Quando uma versão é gerada no RDS, o primeiro ponto a ser tratado no MEVID é a extração do conteúdo semântico de cada documento semântico referenciado nessa versão. Para tal o subpacote *extraction* é responsável por comportar as interfaces que definem o contrato para extração do conteúdo semântico, além de classes usadas na implementação dessas interfaces. O subpacote *vin* é o pacote central desse módulo e é responsável pelas funcionalidades gerais de versionamento de documentos semânticos, notificação de alterações e integração dos dados levantados em uma versão.

A Figura A.8 apresenta o diagrama de classes do subpacote *extraction*. A interface *SemanticContentExtractor* define o contrato que permite a verificação de um documento semântico e a extração do conteúdo semântico do mesmo. A classe que implementa essa interface, *AbstractSemanticDocumentExtractor*, faz a implementação de parte do comportamento comum às suas subclasses. Para geração do conteúdo semântico, essa classe depende da interface *SemanticAnnotationInterpreter*, que define o contrato das instruções de geração de conteúdo semântico definidas no MAMD.

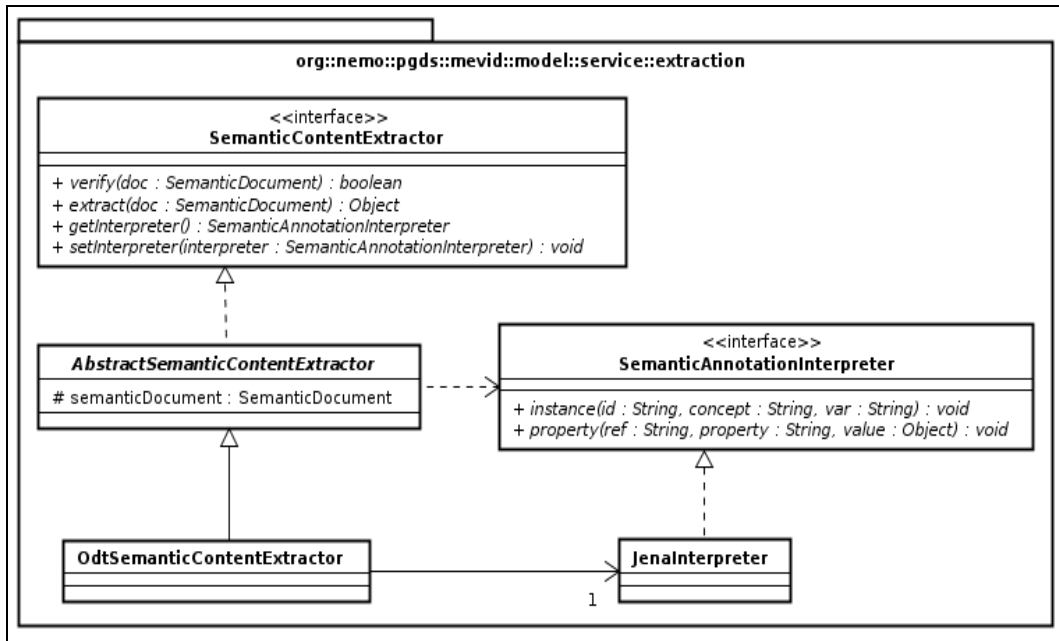
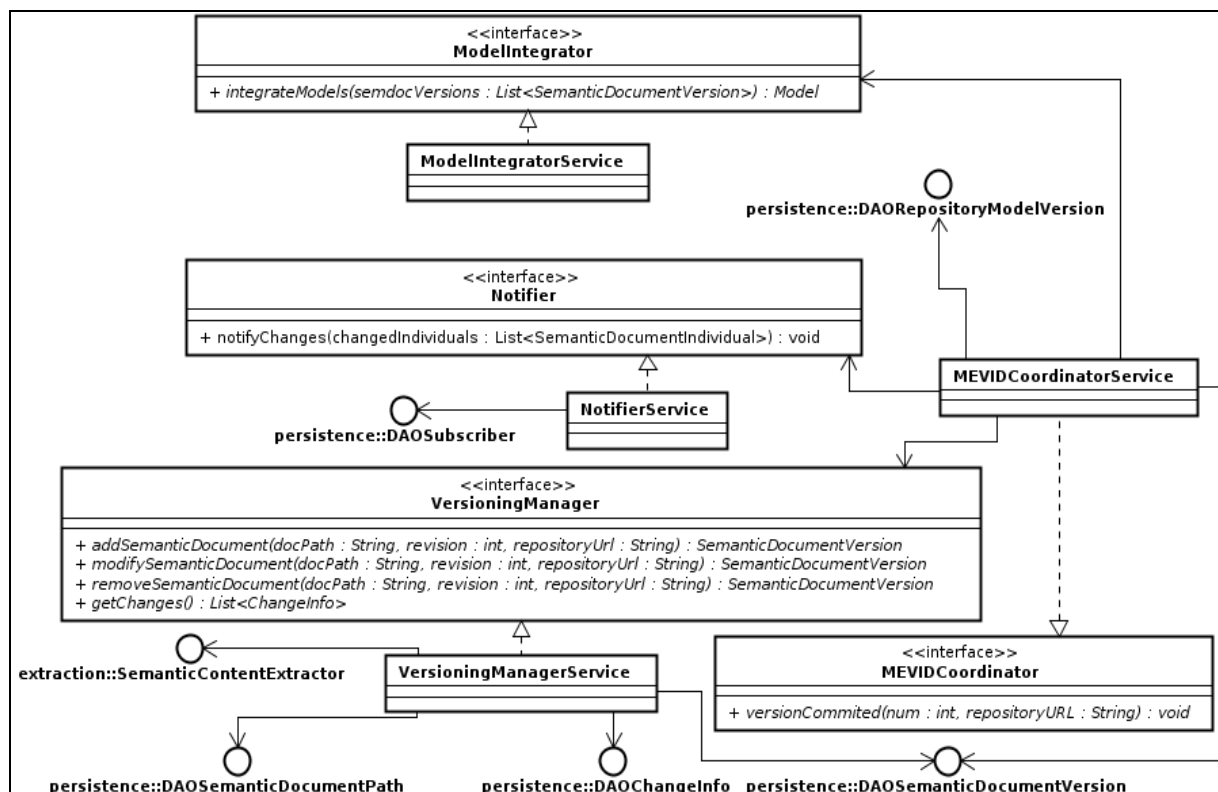


Figura A.8 - Diagrama de Classes do pacote *mevid::model::service::extraction*

A classe *OdtSemanticContentExtractor* faz a implementação dos métodos de extração de conteúdo semântico para documentos semânticos no formato ODT. A implementação de *SemanticAnnotationInterpreter* utilizada por essa classe é a classe *JenaInterpreter*, relativa ao *framework* Jena. Dessa forma quando um documento semântico no formato ODT é processado por esse maquinário, a cada anotação semântica de modelo encontrada no documento, a análise e o processamento da mesma monta, gradativamente, o grafo final do documento.

A Figura A.9 apresenta o diagrama de classes do subpacote *vin*. A interface *MEVIDCoordinator* define o método a ser chamado quando uma versão for gerada no RDS. A implementação dessa interface (a classe *MEVIDCoordinatorService*), por sua vez, faz a orquestração das funcionalidades de extração, versionamento, notificação e integração de dados desse módulo, baseando-se na leitura do *log* gerado pelo sistema de controle de versões *Subversion*. Por esse motivo ela depende das interfaces que definem o contrato para cada funcionalidade.



A interface *VersioningManager* define os métodos que vão conduzir o versionamento de um documento semântico, a saber: operações de adição, modificação e remoção de documentos semânticos. Sua classe de implementação, *VersioningManagerService*, faz uso da interface *SemanticContentExtractor* para realizar as operações de extração do conteúdo semântico dos documentos.

Ao finalizar a geração do conteúdo semântico da versão e o versionamento do mesmo, o processo de notificação é iniciado. Esse processo é implementado pela classe *NotifierService*. Por fim, o processo é finalizado com a integração dos conteúdos semânticos contidos no RD em suas últimas versões e gera mais fatos usando uma máquina de raciocínio (Pellet Reasoner). Esse passo final é apoiado pela classe *ModelIntegratorService*.

O pacote *mevid::controller* é responsável por atender requisições efetuadas pelo gancho de *post-commit* instalado no RDS. A implementação desse gancho faz uso da funcionalidade cURL que é capaz de efetuar requisições HTTP. O conteúdo do script utilizado deve seguir o seguinte padrão (em sistemas Linux):

```
#!/bin/bash
curl <url_instalacao_pgds>/remoting/versioncontrol/vc?rev=$1&url=$2
```

onde <url_instalacao_pgds> é o caminho onde a plataforma foi instalada (exemplo: <http://meuservidor.com.br/pgds>) e os itens “\$1” e “\$2” representam a revisão recém gerada e o caminho do repositório *Subversion*, respectivamente.

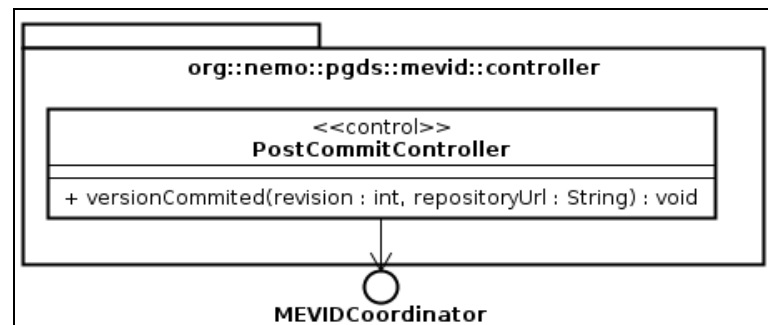


Figura A.10 - Diagrama de Classes do pacote *mevid::controller*

O comportamento de atender à requisição HTTP feita pelo gancho é implementado pela classe *PostCommitController* (Figura A.10), que recebe uma instância de *MEVIDCoordinatorService* do pacote *mevid::model::service::vin*, via injeção de dependências, e repassa a requisição para a mesma.

3.6.3 - Módulo de Busca e Rastreabilidade (MBR)

O Módulo de Busca e Rastreabilidade (MBR) é responsável por prover os serviços para os usuários da plataforma, dentre eles consultas aos conteúdos semânticos providos pelo MEVID, rastreamento evolutivo de um indivíduo contido no RD do MEVID e assinatura de notificação segundo alteração de um indivíduo.

Os serviços contemplados no pacote *mbr::model::service* são responsáveis por efetuar pesquisas sobre os grafos armazenados pelo RD do MEVID. Adicionalmente, o serviço de assinatura de notificação de alteração de indivíduo é também contemplado nesse pacote.

A interface *SearchAndTraceabilityEngine* é responsável por manter o contrato dos métodos para executar as funcionalidades referentes a pesquisas. Sua implementação faz uso do aparato de persistência do MEVID (pacote *mevid::model::persistence*) para obter os grafos referentes a um indivíduo. Para buscas, usa-se a linguagem de pesquisas SPARQL (voltada para pesquisa de triplas, elementos que compõem os grafos RDF) e uma instância de *RepositoryModelVersion*. A Figura A.11 apresenta as classes e dependências desse pacote.

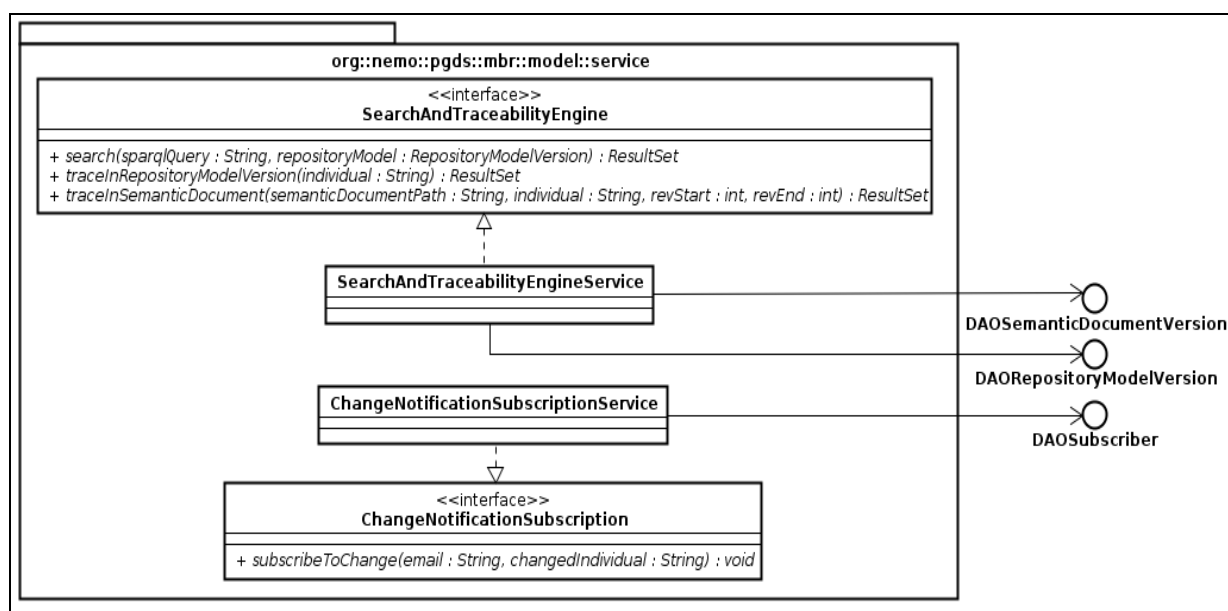


Figura A.11 - Diagrama de Classes do pacote *mbr::model::service*

As funcionalidades de rastreabilidade (métodos *traceInRepositoryModelVersion* e *traceInSemanticDocument*) fazem as listagens dos grafos e executam consultas capturando as triplas em que o indivíduo em questão é o objeto da mesma, em cada um dos grafos, retornando o resultado. A classe *ResultSet*, que representa o resultado de uma consulta, é provida pelo *framework* Jena, utilizado para efetuar as buscas sobre grafos. Finalmente, a interface *ChangeNotificationSubscription* define o contrato para assinatura de alterações em um indivíduo.

O pacote *mbr::controller* conta com um único controlador, *SearchTraceabilityAndSubscriptionController*, responsável por atender as requisições de pesquisa em geral (busca ou rastreabilidade) e de assinatura de alterações. A Figura A.12 apresenta as classes do pacote *mbr::controller* e, adicionalmente, do pacote *mbr::view*.

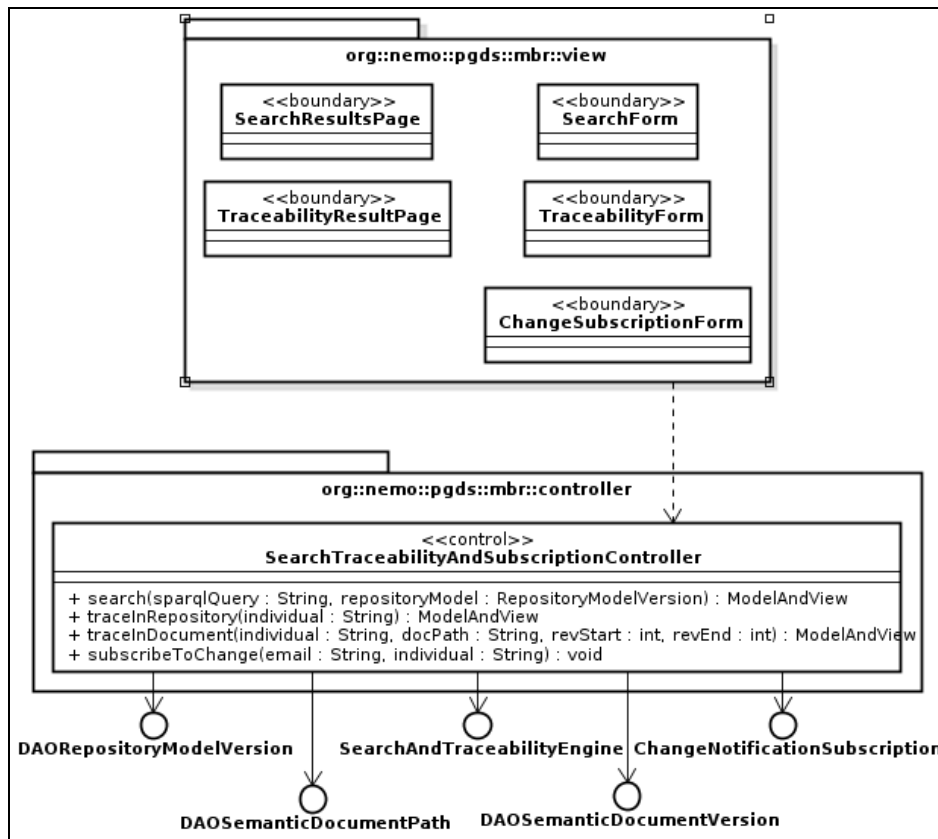


Figura A.12 - Diagrama de Classes do pacote *mbr::controller* e *mbr::view*

Todas as requisições recebidas por essa classe são repassadas aos serviços responsáveis, contemplados no pacote *mbr::service*. As relações entre o controlador e as classes DAO são referentes à busca e listagem de itens a serem apresentados na interface gráfica.

O pacote *mbr::view* contém as classes e arquivos necessários para a realização da interface com o usuário do MBR. Todas as classes contempladas nesse pacote se comunicam com o controlador principal do MBR – *SearchTraceabilityAndSubscriptionController*.

SearchForm e *SearchResultsPage* representam o formulário de busca e a página de resultados da mesma, respectivamente. *TraceabilityForm* e *TraceabilityResultsPage* representam o formulário de pesquisa de rastreabilidade evolutiva e a página de resultados da mesma, respectivamente. Finalmente a classe *ChangeSubscriptionForm* é responsável por comportar o formulário de assinatura sobre alterações em um indivíduo.

Apêndice B – Modelos de Documento do NEMO Anotados Semanticamente

Documento de Requisitos

Projeto: <<Nome do Projeto>>

Versão: <<Versão do Documento>>

Responsáveis: <<Nome dos Responsaveis, separados por vírgula>>

1. Introdução

<<Descrição da Organização do Documento>>

2. Descrição do Propósito do Sistema

<<Descrição sucinta do propósito do sistema>>

3. Descrição do Mini-mundo

<<Descrição do escopo do problema>>

4. Requisitos Iniciais do Cliente

Requisitos Funcionais

<<id do Requisito>>	<<Descrição do requisito>>
---------------------	----------------------------

Requisitos Não Funcionais

<<id do Requisito>>	<<Descrição do requisito>>
---------------------	----------------------------

Projeto : <<Nome do Projeto>>

Subsistema : <<Nome do Subsistema ou Módulo>>

Caso de Uso: <<Nome do Caso de Uso>>

Descrição:

<<Descrição sucinta do caso de uso>>

Fluxos Normais

<< <i>nome de cada fluxo normal em cada linha</i> >>	<<sequência de passos de interação>>
--	--------------------------------------

Fluxos Alternativos de Exceção

<< <i>nome do fluxo normal que possui fluxo alternativo</i> >>	<<descrição do curso alternativo>>
--	------------------------------------

Fluxos Alternativos Variantes

<< <i>nome do fluxo normal que possui fluxo alternativo</i> >>	<<descrição do curso alternativo>>
--	------------------------------------

Requisitos: <<Lista de requisitos separados por vírgula>>

Apêndice C – Evolução de uma Matriz de Rastreabilidade de Requisitos

Este anexo apresenta os resultados das consultas referentes à matriz de rastreabilidade de requisitos do projeto em cada uma das versões do Repositório de Documentos Semânticos (RDS) do projeto da Vídeo Locadora Passatempo. Essa matriz informa o projeto, os módulos do mesmo, os casos de uso alocados em cada módulo, os requisitos definidos no projeto e, se for o caso, a relação entre os requisitos e os casos de uso. Os resultados das consultas realizadas em cada versão são apresentados a seguir.

C.1 Resultados da Matriz na Versão 1:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08

Documentos Semânticosda Versão 1:
/docRequisitos.odt in svn revision(1)

C.2 Resultados da Matriz na Versão 2:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RF09
Video_Locadora_Passatempo			RF11
Video_Locadora_Passatempo			RF12
Video_Locadora_Passatempo			RF13
Video_Locadora_Passatempo			RF14
Video_Locadora_Passatempo			RF15

Video_Locadora_Passatempo			RF16
Video_Locadora_Passatempo			RNF01
Video_Locadora_Passatempo			RNF02
Video_Locadora_Passatempo			RNF03
Video_Locadora_Passatempo			RNF04
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08

Documentos Semânticos da Versão 2:
/docRequisitos.odt na versão (2)

C.3 Resultados da Matriz na Versão 3:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RF10
Video_Locadora_Passatempo			RF11
Video_Locadora_Passatempo			RF12
Video_Locadora_Passatempo			RF13
Video_Locadora_Passatempo			RF14
Video_Locadora_Passatempo			RF15
Video_Locadora_Passatempo			RF16
Video_Locadora_Passatempo			RNF02
Video_Locadora_Passatempo			RNF03
Video_Locadora_Passatempo			RNF04
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF06
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01

Documentos Semânticos da Versão 3:
/docRequisitos.odt na versão (3)
/casoUso_Acervo_Cadastrar_Filme.odt na versão (3)
/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (3)

C.4 Resultados da Matriz na Versão 4:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RF12
Video_Locadora_Passatempo			RF13
Video_Locadora_Passatempo			RF14
Video_Locadora_Passatempo			RF15
Video_Locadora_Passatempo			RF16
Video_Locadora_Passatempo			RNF04
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF06
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02

Documentos Semânticos da Versão 4:

/docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)

/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)

/casoUso_Acervo_Consultar_Acervo.odt na versão (4)

/casoUso_Acervo_Cadastrar_Item.odt na versão (4)

/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)

C.5 Resultados da Matriz na Versão 5:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF01
Video_Locadora_Passatempo			RF02
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RF12
Video_Locadora_Passatempo			RF13
Video_Locadora_Passatempo			RF14
Video_Locadora_Passatempo			RF15
Video_Locadora_Passatempo			RF16
Video_Locadora_Passatempo			RNF04
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF06
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02

Documentos Semânticos da Versão 5:

/docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)

/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)

/casoUso_Acervo_Consultar_Acervo.odt na versão (4)

/casoUso_Acervo_Cadastrar_Item.odt na versão (5)

/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)

C.6 Resultados da Matriz na Versão 6:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF03
Video_Locadora_Passatempo			RF04
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF06
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF12
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF13
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF14
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF15
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF16
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF04

Documentos Semânticos da Versão 6:

/docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)

/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)

/casoUso_Acervo_Consultar_Acervo.odt na versão (4)

/casoUso_Acervo_Cadastrar_Item.odt na versão (5)

/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)

/casoUso_Atendimento_Efetuar_Locacao.odt na versão (6)

/casoUso_Atendimento_Cadastrar_Cliente.odt na versão (6)

C.7 Resultados da Matriz na Versão 7:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF05
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF12
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF13
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF14
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF15
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF16
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF03
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF04

Documentos Semânticos da Versão 7:

/docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)

/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)

/casoUso_Acervo_Consultar_Acervo.odt na versão (4)

/casoUso_Acervo_Cadastrar_Item.odt na versão (5)

/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)

/casoUso_Atendimento_Efetuar_Locacao.odt na versão (7)

/casoUso_Atendimento_Cadastrar_Cliente.odt na versão (6)

/casoUso_Atendimento_Efetuar_Devolucao.odt na versão (7)

C.8 Resultados da Matriz na Versão 8:

projeto	modulo	casouso	req
Video_Locadora_Passatempo			RF06
Video_Locadora_Passatempo			RF07
Video_Locadora_Passatempo			RF08
Video_Locadora_Passatempo			RNF05
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF12
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF13
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF14
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF15
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF16
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF03
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RF05
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF04

Documentos Semânticos da Versão 8:

/docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)

/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)

/casoUso_Acervo_Consultar_Acervo.odt na versão (4)

/casoUso_Acervo_Cadastrar_Item.odt na versão (5)

/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)

/casoUso_Atendimento_Efetuar_Locacao.odt na versão (7)

/casoUso_Atendimento_Cadastrar_Cliente.odt na versão (6)
 /casoUso_Atendimento_Efetuar_Devolucao.odt na versão (8)
 /casoUso_Atendimento_Efetuar_Pagamento.odt na versão (8)
 /casoUso_Atendimento_Efetuar_Reserva.odt na versão (8)

C.9 Resultados da Matriz na Versão 9:

projeto	modulo	casoUso	req
Video_Locadora_Passatempo			RNF07
Video_Locadora_Passatempo			RNF08
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RF10
Video_Locadora_Passatempo	Acervo	Cadastrar_Distribuidora	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Filme	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF01
Video_Locadora_Passatempo	Acervo	Cadastrar_Item	RNF03
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RF09
Video_Locadora_Passatempo	Acervo	Cadastrar_Tipo_de_Midia	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RF11
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF01
Video_Locadora_Passatempo	Acervo	Consultar_Acervo	RNF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF12
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF13
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF14
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF15
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RF16
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Cadastrar_Cliente	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF03
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Devolucao	RNF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RF02
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Locacao	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RF05
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF04
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Pagamento	RNF05
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF06
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF07
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RF08
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RNF01
Video_Locadora_Passatempo	Atendimento_a_Cliente	Efetuar_Reserva	RNF04

Documentos Semânticos da Versão 9:
 /docRequisitos.odt na versão (3)

/casoUso_Acervo_Cadastrar_Filme.odt na versão (4)
/casoUso_Acervo_Cadastrar_Distribuidora.odt na versão (4)
/casoUso_Acervo_Consultar_Acervo.odt na versão (4)
/casoUso_Acervo_Cadastrar_Item.odt na versão (5)
/casoUso_Acervo_Cadastrar_Tipo_Midia.odt na versão (4)
/casoUso_Atendimento_Efetuar_Locacao.odt na versão (7)
/casoUso_Atendimento_Cadastrar_Cliente.odt na versão (6)
/casoUso_Atendimento_Efetuar_Devolucao.odt na versão (8)
/casoUso_Atendimento_Efetuar_Pagamento.odt na versão (9)
/casoUso_Atendimento_Efetuar_Reserva.odt na versão (9)

Anexo A – Modelos de Documento do NEMO

Documento de Requisitos

Projeto: <<Nome do Projeto>>

Versão: <<Versão do documento>>

Responsável: <<Lista de Responsáveis>>

1. Introdução

<<Descrição da Organização do Documento>>

2. Descrição do Propósito do Sistema

<<Descrição sucinta do propósito do sistema>>

3. Descrição do Mini-mundo

<<Descrição do escopo do problema>>

4. Requisitos Iniciais do Cliente

Listar cada identificador de requisito, juntamente com sua descrição.

Requisitos Funcionais

RF1. <<Descrição do requisito RF01>>;

RF2. <<Descrição do requisito RF02>>.

Requisitos Não Funcionais

RNF01. <<Descrição do requisito RNF01>>;

RNF02. <<Descrição do requisito RNF02>>.

Projeto : <<Nome do Projeto>>

Subsistema : << Nome do Subsistema ou Módulo>>

Caso de Uso: <<Nome do caso de uso>>

Descrição: <<Descrição sucinta do caso de uso>>

Fluxos Normais

<<Nome de cada Fluxo Normal>>

1. <<Descrição do Passo 1>>;
2. <<Descrição do Passo 2>>.

Fluxos Alternativos de Exceção

<<Nome de cada Fluxo Normal que possua Fluxo Alternativo de Exceção >>

<<Número do passo com o fluxo alternativo>> <<Descrição do passo alternativo>>

Fluxos Alternativos Variantes

<<Nome de cada Fluxo Normal que possua Fluxo Alternativo Variante>>

<<Número do passo com fluxo alternativo>> <<Descrição do passo alternativo>>

Requisitos: <<Listagem dos requisitos atendidos por este caso de uso>>

Anexo B – Documentação Final

Documento de Requisitos

Projeto: Video Locadora Passatempo

Versão: 1.0

Responsáveis: Ricardo de Almeida Falbo, Lucas de Oliveira, Daniel de Oliveira

1. Introdução

Este documento apresenta os requisitos iniciais para a informatização da Vídeo Locadora Passatempo. Essa atividade foi conduzida por meio da realização de entrevistas com cliente e usuários. Este documento está organizado da seguinte forma: a seção 2 contém uma descrição do propósito do sistema; a seção 3 apresenta uma descrição do mini-mundo apresentando o problema; e a seção 4 apresenta a lista de requisitos iniciais levantados junto ao cliente;.

2. Descrição do Propósito do Sistema

A vídeo-locadora necessita de um sistema de informação para apoiar a realização de suas atividades principais, a saber: empréstimo e devolução de itens e reserva de filmes. Para que essas atividades sejam apoiadas, é necessário controlar as informações acerca de filmes e usuários, dentre outros. Além disso, devem ser fornecidas facilidades de consulta ao acervo da locadora, permitindo consultas por diversas informações dos filmes.

3. Descrição do Mini-mundo

A Vídeo Locadora *Passatempo* deseja um sistema de informação para gerenciar o atendimento aos seus clientes. O negócio principal da locadora é a locação de vídeos (ou filmes). De um dado filme, a locadora pode possuir vários itens, em diferentes mídias, tais como DVD, VHS, Blu-Ray, HD-DVD. Os filmes são classificados em gêneros, tais como drama, comédia, documentário, ação, terror etc. Além disso, a locadora faz distinção entre filmes de catálogo e lançamentos.

Os valores padrão das locações são dados pelo tipo de mídia do item sendo locado. Atualmente, são cobrados os seguintes valores: DVD, VHS e HD-DVD – R\$ 5,00; Blu-Ray – R\$ 7,50, sendo que lançamentos têm um acréscimo de 50% nos valores acima mencionados. O prazo para devolução é de um dia para lançamentos e três dias para filmes do catálogo. Contudo, o valor efetivamente cobrado por uma locação ou a sua data de devolução prevista podem ser alterados pelo funcionário da locadora para aplicar

descontos individualizados ou ampliar prazos de devolução.

Sobre um filme, deseja-se saber: título original, título em português, país, ano, direção, elenco, sinopse, duração, gênero, distribuidora, informações de áudio (tipo, p.ex., Dolby Digital 2.0, e língua, p.ex., Português) e informações de legenda (língua). Os filmes são fornecidos por distribuidoras. De uma distribuidora deseja-se saber a razão social, CNPJ, endereço, telefone e pessoa de contato. De um item, deseja-se saber a data de aquisição, número de série e tipo de mídia.

Clientes locam itens. Um cliente pode ser um cliente titular ou um de seus dependentes. Quando uma pessoa faz sua inscrição na locadora como titular, lhe é dado o direito de indicar até três dependentes, pelos quais será responsável. Para a locadora, é fundamental identificar exatamente quem locou uma fita, se o titular ou um de seus dependentes. Contudo, para efeito de controle, a locadora deseja ter mais informações sobre o titular do que sobre seus dependentes. Sobre um titular, deseja-se saber nome, email, endereço, telefone residencial, local onde trabalha, telefone comercial, telefone celular, sexo, CPF e data de nascimento. Apenas maiores de idade podem ser titulares. De um dependente, são necessários apenas o nome, email, sexo e data de nascimento. Tanto titulares quanto dependentes têm um número de inscrição, o qual é único por cliente.

Clientes podem também reservar filmes. É importante registrar a data e a hora em que a reserva foi feita e o tipo de mídia que o cliente deseja. Assim, é possível atender as reservas por ordem de chegada, por tipo de mídia. Uma locação só pode ser feita para um item, se não houver uma reserva não atendida para o seu filme e mídia. Quando um item de um filme e tipo de mídia reservado é devolvido, comunica-se o cliente interessado por email e, a partir desse momento, o cliente tem 24 horas para retirá-lo; caso contrário, expira-se a reserva e o item é liberado. Não são aceitas reservas para filmes que têm itens do tipo de mídia requerido disponíveis na locadora, nem reservas para datas previamente especificadas.

Quando a devolução de um item é feita com atraso, cobra-se multa. Caso a locação do item não tenha sido paga no ato da locação, terá de ser paga obrigatoriamente na devolução. Não são aceitos pagamentos mensais ou em outros momentos que não a locação ou a devolução. Pagamentos podem ser feitos em dinheiro, cheque ou cartão, sendo que para pagamentos com cheque deseja-se saber: banco, agência, conta e número do cheque. Para pagamentos em cartão, registra-se apenas o valor, a operadora do cartão e o número e a data da autorização.

Consultas ao acervo da locadora devem poder ser feitas pela Internet. Um cliente pode consultar os dados de um filme específico, informando o título (ou parte dele), original ou em português. Também devem ser possíveis consultas por gênero, tipo de mídia disponível, ator, diretor, nacionalidade e lançamentos, bem como combinações dessas informações.

4. Requisitos Iniciais do Cliente

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos do cliente:

Requisitos Funcionais

RF01	O sistema deve registrar locações, indicando o cliente e os itens locados, bem como deve registrar a data e o valor da locação e a data de devolução prevista de cada item.
RF02	O sistema deve permitir que sejam dados descontos nas locações, bem como que sejam ampliados os prazos de devolução de itens, em função da política da empresa.
RF03	O sistema deve registrar devoluções, indicando os itens devolvidos e a data de devolução.
RF04	O sistema deve cobrar multa para itens devolvidos com atraso, de acordo com política da empresa.
RF05	O sistema deve registrar os pagamentos de locações, os quais poderão ser feitos no ato da locação ou da devolução de um item. Pagamentos poderão ser feitos em dinheiro, cheque ou cartão.
RF06	O sistema deve registrar a reserva de filmes a clientes, permitindo indicar, ainda, o tipo de mídia desejado.
RF07	O sistema deve permitir que o atendimento às reservas se dê pela ordem temporal das reservas, considerando os tipos de mídia solicitados.
RF08	O sistema deve permitir o cancelamento de uma reserva, tanto pelo usuário, quanto automaticamente pelo sistema, quando expirado o prazo para retirada do item, de acordo com política da empresa. Reservas canceladas pelo sistema não deverão ser efetivamente excluídas pelo sistema, mas sim marcadas como expiradas.
RF09	O sistema deve efetuar o controle do acervo da locadora.
RF10	O sistema deve efetuar o controle de distribuidoras de filmes.
RF11	O sistema deve permitir consultas ao acervo. As consultas poderão ser feitas informando uma (ou uma combinação) de informações dos filmes, dentre elas: título (ou parte dele), gênero, tipo de mídia disponível, ator.
RF12	O sistema deve permitir o gerenciamento do conjunto de clientes da locadora, abrangendo tanto clientes titulares quanto seus dependentes.
RF13	O sistema deve manter o histórico de locações e, portanto, clientes que tenham feito locações não poderão ser excluídos.

RF14	O sistema deve restringir a três o número de dependentes de um titular em um certo ponto no tempo.
RF15	O sistema deve permitir desativar e reativar clientes. Quando um cliente titular for desativado, todos os seus dependentes também devem ser desativados.
RF16	O sistema deve gerar um número de inscrição único para cada cliente.

Requisitos Não Funcionais

RNF01	O sistema deve controlar o acesso às funcionalidades. Funcionalidades para controlar o acervo da locadora devem ser restritas a administradores. Funcionalidades de atendimento a clientes devem estar restritas a atendentes. Funcionalidades de consulta ao acervo devem estar disponíveis em formulários <i>html</i> na Internet e acessíveis via <i>web services</i> .
RNF02	A consulta ao acervo deve estar disponível pela Internet, a partir dos principais navegadores disponíveis no mercado.
RNF03	Os itens devem ser identificados por um código de barras, sendo possível a leitura dos mesmos usando dispositivos de leitoras de código de barras.
RNF04	O tempo para a realização das funções de atendimento ao cliente deve ser inferior a um segundo, a partir da correta entrada de dados.
RNF05	O sistema deve estar integrado ao Sistema de Operadoras de Cartão de Crédito para enviar e receber informações para pagamento com cartão de crédito.
RNF06	A comunicação de disponibilidade de uma reserva deverá ser feita automaticamente por email.
RNF07	A persistência das informações deve ser implementada, em um primeiro momento, em um Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR) livre (Postgres ou MySQL). Contudo, no futuro deve ser possível utilizar outros SGBDRs ou até mesmo outra tecnologia de bancos de dados.
RNF08	Um serviço de manutenção da base de dados deverá ser provido, eliminando clientes desativados e suas locações. Esse serviço será rodado automaticamente em um intervalo de tempo definido pela locadora ou disparado manualmente pelo administrador do sistema. Em ambos os casos, uma cópia de restauração da base de dados deverá ser feita.

Projeto : Vídeo Locadora Passatempo

Subsistema : Acervo

Caso de Uso: Cadastrar Distribuidora

Descrição:

Caso de Uso responsável pelo cadastro de distribuidoras no sistema

Fluxos Normais

<i>Incluir Distribuidora</i>	Informar: razão social, CNPJ, endereço, telefone e pessoa de contato.
<i>Alterar Distribuidora</i>	1- Exibe os mesmos dados da inclusão para o usuário; 2- O mesmo informa os novos dados; 3- O sistema efetua o registro
<i>Consultar Distribuidora</i>	Consulta por cada atributo informado na inclusão.
<i>Excluir Distribuidora</i>	Exclusão por seleção e confirmação.

Fluxos Alternativos de Exceção

<i>Excluir Distribuidora</i>	Não é permitido excluir uma distribuidora que tenha filmes associados.
-------------------------------------	--

Fluxos Alternativos Variantes

--	--

Requisitos: RF10,RNF01

Projeto : Vídeo Locadora Passatempo

Subsistema : Acervo

Caso de Uso: Cadastrar Filme

Descrição:

Caso de Uso responsável pelo cadastro de filmes no sistema

Fluxos Normais

<i>Incluir Filme</i>	Informar: título original, título em português, país, ano, diretores, atores, sinopse, duração, gênero, distribuidora, tipo de áudio (p.ex., Dolby Digital 2.0), idioma do áudio e idioma da legenda.
<i>Alterar Filme</i>	1- Sistema exibe os dados do filme; 2- Usuário informa os novos dados; 3- Sistema verifica e registra os novos dados do filme;
<i>Consultar Filme</i>	Consulta por cada atributo informado na inclusão.
<i>Excluir Filme</i>	Exclusão por seleção e confirmação.

Fluxos Alternativos de Exceção

<i>Excluir Filme</i>	1. Não é permitida a exclusão de filmes que tenham itens associados. 2. Ao excluir um filme, devem-se excluir as reservas associadas.
-----------------------------	--

Fluxos Alternativos Variantes

--	--

Requisitos: RF09,RNF01

Projeto : Vídeo Locadora Passatempo

Subsistema : Acervo

Caso de Uso: Cadastrar Item

Descrição:

Caso de Uso responsável pelo cadastro de itens no sistema

Fluxos Normais

<i>Incluir Item</i>	Informar: filme, tipo de mídia, data de aquisição e número de série.
<i>Alterar Item</i>	Similar a incluir
<i>Consultar Item</i>	Consulta por cada atributo informado na inclusão.
<i>Excluir Item</i>	Exclusão por seleção e confirmação.

Fluxos Alternativos de Exceção

<i>Excluir Item</i>	Não é permitido excluir um item que tenha locações associadas.
---------------------	--

Fluxos Alternativos Variantes

--	--

Requisitos: RF09,RNF01,RNF03

Projeto : Vídeo Locadora Passatempo

Subsistema : Acervo

Caso de Uso: Cadastrar Tipo de Mídia

Descrição:

Caso de Uso responsável pelo cadastro dos diferentes tipos de mídia no sistema

Fluxos Normais

<i>Incluir Tipo de Mídia</i>	Informar: nome e valor de locação.
<i>Alterar Tipo de Mídia</i>	Similar a incluir
<i>Consultar Tipo de Mídia</i>	Consulta por cada atributo informado na inclusão.
<i>Excluir Tipo de Mídia</i>	Exclusão por seleção e confirmação.

Fluxos Alternativos de Exceção

<i>Excluir Tipo de Mídia</i>	1.Não é permitido excluir um tipo de mídia que tenha itens associados. 2.Ao excluir um tipo de mídia, devem-se excluir as reservas que especificam apenas esse tipo de mídia.
-------------------------------------	--

Fluxos Alternativos Variantes

--	--

Requisitos: RF09,RNF01

Projeto : Vídeo Locadora Passatempo

Subsistema : Acervo

Caso de Uso: Consultar Acervo

Descrição:

As consultas ao acervo poderão ser feitas informando uma (ou uma combinação) das seguintes informações: título (ou parte dele), original ou em português, gênero, tipo de mídia disponível, ator, diretor, nacionalidade e lançamentos.

Fluxos Normais

--	--

Fluxos Alternativos de Exceção

--	--

Fluxos Alternativos Variantes

--	--

Requisitos: RF11,RNF01,RNF02

Projeto : Vídeo Locadora Passatempo

Subsistema : Atendimento a Cliente

Caso de Uso: Cadastrar Cliente

Descrição:

Este caso de uso é responsável pela inclusão de um novo titular e seus dependentes, bem como alteração de dados, consulta e exclusão de clientes em geral (titulares e dependentes).

Fluxos Normais

<i>Incluir Novo Titular</i>	1. O atendente informa os dados do novo titular, a saber: nome, sexo, data de nascimento, endereço, telefones residencial, celular e comercial e local de trabalho. 2. Um número de inscrição é gerado pelo sistema. 3. O novo cliente titular é registrado como sendo um cliente ativo. 4. Caso o cliente titular deseje inscrever dependentes, realizar o fluxo de eventos “Incluir Dependente”.
<i>Incluir Dependente</i>	1. O atendente informa o titular que deseja incluir um novo dependente. 2. O atendente informa os dados do novo dependente, a saber: nome, sexo e data de nascimento. 3. Um número de inscrição é gerado pelo sistema. 4. O novo dependente é registrado como sendo um cliente ativo.
<i>Alterar Dados de Cliente</i>	5. O atendente informa o cliente do qual deseja alterar dados. 6. O atendente informa os novos dados, sendo que o número de inscrição de um cliente não pode ser alterado. 7. As alterações são registradas.
<i>Desativar Cliente</i>	7. O atendente informa o cliente ativo a ser desativado. 8. O cliente é desativado.
<i>Reativar Cliente</i>	2. O atendente informa o cliente inativo a ser reativado. 3. O cliente é reativado.
<i>Consultar Dados de Cliente</i>	1. O atendente informa o cliente que deseja consultar. 2. Os dados do cliente são apresentados. 3. Se o cliente for um titular e possuir dependentes ativos, apresentar, também, os nomes dos mesmos.
<i>Excluir Cliente</i>	1. O atendente informa o cliente que deseja excluir. 2. Os dados do cliente são apresentados e é solicitada confirmação. 3. As reservas feitas pelo cliente e por seus dependentes são excluídas. 4. Caso o cliente seja um titular, seus dependentes são também excluídos. 5. O cliente é excluído.

Fluxos Alternativos de Exceção

<i>Incluir Novo Titular</i>	1a – Dados inválidos: uma mensagem de erro é exibida, retornando ao passo 1 para correção da informação inválida.
<i>Incluir Dependente</i>	1a – O titular já possui três dependentes ativos: uma mensagem de erro é exibida, informando que o titular já possui três dependentes ativos e o fluxo de eventos é abortado. 2a – Dados inválidos: uma mensagem de erro é exibida, retornando ao passo 2 para correção da informação inválida.
<i>Alterar Dados de Cliente</i>	2a – Dados inválidos: uma mensagem de erro é exibida, retornando ao passo 2 para correção da informação inválida.
<i>Reativar Cliente</i>	2a – O cliente a ser reativado é dependente de um cliente titular inativo: uma mensagem de erro é exibida, informando que cliente é dependente de um titular inativo e o fluxo de eventos é abortado.
<i>Excluir Cliente</i>	1a – O cliente ou um de seus dependentes possui locações 1a.1: Uma mensagem de erro é exibida, indicando que o cliente ou um de seus dependentes possui locações e, portanto, não pode ser excluído, e perguntando se deseja desativar o cliente. 1a.2 - Se o atendente deseja desativar o cliente, realizar o fluxo de eventos “Desativar Cliente”, senão abortar o fluxo de eventos corrente. 2a – Solicitação de confirmação de exclusão negada: abortar o fluxo de eventos.

Fluxos Alternativos Variantes

<i>Desativar Cliente</i>	2a – O cliente é um cliente titular e possui dependentes: O cliente titular e seus dependentes são desativados.
---------------------------------	---

Requisitos: RF12, RF13, RF14, RF15, RF16, RNF01, RNF04

Projeto : Vídeo Locadora Passatempo

Subsistema : Atendimento a Cliente

Caso de Uso: Efetuar Devolução

Descrição:

Este caso de uso é responsável pela realização de devolução de itens da locadora.

Fluxos Normais

Devolver Item	1. O atendente informa cada um dos itens que estão sendo devolvidos. 2. Para cada item devolvido 2.1 - Se o item não tiver sido pago no ato da locação, adicionar o seu valor de locação ao valor a ser pago na devolução. 2.2 - Caso o item locado esteja em atraso (data de devolução > data de devolução prevista do item), calcular a multa devida e adicionar o valor da multa ao valor a ser pago na devolução. A multa é calculada como sendo o número de dias em atraso (data devolução - data de devolução prevista) vezes o valor de locação do item. 3. Caso deseje, o atendente poderá aplicar um desconto no valor a ser pago na devolução. 4. Se o valor a ser pago na devolução for maior do que zero, realizar caso de uso “Efetuar Pagamento”. 5. Registrar a devolução, indicando os itens locados que foram devolvidos e atribuindo a data corrente como data de devolução. 6. Para cada item devolvido, caso não haja uma reserva pendente para o filme do item que especifique o mesmo tipo de mídia, registrar o item como estando disponível.

Fluxos Alternativos de Exceção

Devolver Item	1a – Item não possui uma locação vigente. 1a.1 - Uma mensagem de erro é exibida, informando que o item informado não está locado no momento. 1a.2 – Retornar ao passo 1 sem incluir esse item no conjunto de itens sendo devolvidos.

Fluxos Alternativos Variantes

Devolver Item	6a – Há reserva pendente para o filme do item devolvido (especificando o mesmo tipo de mídia do item devolvido): 6a.1 – Registrar o item como estando reservado.
----------------------	---

	6a.2 – Enviar um email automaticamente para o cliente da reserva, indicando que há um item disponível para locação e que, a partir daquele momento, o cliente tem 24 horas para efetuar a locação. 6a.3 – Registrar a reserva como tendo sido comunicada ao cliente. 6a.4 – Prosseguir o passo 6 com o próximo item que está sendo devolvido.
--	--

Requisitos: RF02, RF03, RF04, RNF01, RNF04, RNF06

Projeto : Vídeo Locadora Passatempo

Subsistema : Atendimento a Cliente

Caso de Uso: Efetuar Locação

Descrição:

Este caso de uso é responsável pela realização de uma nova locação, bem como pela consulta e cancelamento de uma locação.

Fluxos Normais

<i>Efetuar Nova Locacao</i>	1. O atendente informa o cliente que deseja efetuar a locação. 2. O atendente informa cada um dos itens a serem locados. 3. Para cada item locado 3.1 - Calcular o valor da locação do item. O valor da locação de um item é dado pelo tipo de mídia do item. Cada tipo de mídia tem um valor de locação associado. Um acréscimo de 50% do valor da locação do tipo de mídia deve ser aplicado no caso do filme do item ser um lançamento. 3.2 - Adicionar o valor de locação do item locado ao valor da locação. 3.3 – Calcular a data de devolução prevista. A data de devolução prevista é definida em função do filme do item ser lançamento ou não. Lançamentos têm prazo de um dia; filmes do catálogo têm três dias de prazo. 9. - Caso deseje, o atendente poderá alterar a data de devolução prevista de um item locado. 4. Caso deseje, o atendente poderá aplicar um desconto no valor total da locação. 5. A locação é registrada com a data corrente como data de locação. 6. Caso o cliente deseje pagar, realizar o caso de uso “Efetuar Pagamento”.
Consultar Dados de Locacao	1. O atendente seleciona a locação que deseja consultar os dados. 2. Os dados da locação são apresentados.
<i>Cancelar Locacao</i>	1. O atendente seleciona a locação que deseja cancelar. 2. Os dados da locação são apresentados e é solicitada confirmação do cancelamento. 3. A locação selecionada é excluída.

Fluxos Alternativos de Exceção

<i>Efetuar Nova Locacao</i>	1a – Cliente está em débito. 1a.1 - Uma mensagem de erro é exibida, indicando a(s) locação(ões) em atraso do cliente e perguntando se o cliente deseja quitar o débito. 1a.2 – Caso o cliente deseje quitar seu débito, realizar o caso de uso
------------------------------------	---

	“Efetuar Pagamento” ; caso contrário, abortar o fluxo de eventos. 2a – Existe uma reserva pendente para um item do filme e tipo de mídia sendo locado. 2a.1 - Uma mensagem de erro é exibida, indicando o item não está disponível para locação. 2a.2 – Retornar ao passo 2 sem incluir esse item na locação.
<i>Cancelar Locacao</i>	1a – Locação paga: uma mensagem de erro é exibida, indicando que a locação já foi paga e, portanto, não pode ser cancelada. O fluxo de eventos é abortado. 2a – Solicitação de confirmação de cancelamento negada: abortar o fluxo de eventos.

Fluxos Alternativos Variantes

--	--

Requisitos: RF01,RF02,RNF01,RNF04

Projeto : Vídeo Locadora Passatempo

Subsistema : Atendimento a Cliente

Caso de Uso: Efetuar Pagamento

Descrição:

Este caso de uso é responsável pela realização de um pagamento e consulta a dados de pagamentos.

Fluxos Normais

Incluir Pagamento	1. De posse do valor a ser pago, o atendente informa a forma de pagamento. 2. Efetuar o pagamento: 2a. Em dinheiro 2b. Em cheque 2c. Em cartão 3. O pagamento é registrado.
Consultar Pagamento	1- O atendente seleciona o pagamento que deseja consultar os dados. 2- Os dados do pagamento são apresentados.

Fluxos Alternativos de Exceção

Incluir Pagamento	1a – Item não possui uma locação vigente. 1a.1 - Uma mensagem de erro é exibida, informando que o item informado não está locado no momento. 1a.2 – Retornar ao passo 1 sem incluir esse item no conjunto de itens sendo devolvidos. 2a – Pagamento em Dinheiro: 2a.1 – O atendente informa a quantia em dinheiro entregue pelo cliente. 2a.2 – O sistema informa o valor do troco a ser dado ao cliente. 2b – Pagamento em Cheque: 2b.1 – O atendente informa os dados do cheque, a saber: banco, agência, conta e valor. 2c – Pagamento em Cartão: 2c.1 – O atendente envia as informações do cartão, valor da compra e identificação da loja para o serviço de autorização do Sistema de Operadoras de Cartão de Crédito. 2c.2 – O Sistema de Operadoras de Cartão de Crédito envia o código da autorização.

Fluxos Alternativos Variantes

<i>Incluir Pagamento</i>	2a.1 – O valor da quantia paga em dinheiro é inferior ao valor a ser pago: . 2a.1.1 - Uma mensagem de erro é exibida, informando que a quantia paga deve ser maior ou igual ao valor a ser pago. 2a.1.2 – Retornar ao passo 2a.1. 2b.1 – O valor do cheque é diferente do valor a ser pago: . 2b.1.1 - Uma mensagem de erro é exibida, informando que o valor do cheque deve ser igual ao valor a ser pago. 2b.1.2 – Retornar ao passo 2b.1. 2b.1 – O Sistema de Operadoras de Cartão de Crédito não autoriza a transação: 2b.1.1 - Uma mensagem de erro é exibida, informando que a transação não foi autorizada. 2b.1.2 – Retornar ao passo 1.
---------------------------------	---

Requisitos: RF05, RNF01, RNF04, RNF05

Projeto : Vídeo Locadora Passatempo

Subsistema : Atendimento a Cliente

Caso de Uso: Efetuar Reserva

Descrição:

Este caso de uso é responsável pela realização de uma nova reserva, consulta e cancelamento de uma reserva existente.

Fluxos Normais

<i>Efetuar Nova Reserva</i>	1. O atendente informa o cliente que deseja efetuar a reserva. 2. O atendente informa o dados do item desejado, a saber o filme e o tipo de mídia. 3. A reserva é registrada, com a data e a hora do sistema, além das informações de cliente e filme e tipo de mídia solicitados.
<i>Consultar Dados de Reserva</i>	1. O atendente seleciona a reserva que deseja consultar os dados. 2. Os dados da reserva são apresentados.
<i>Cancelar Reserva</i>	4. O atendente seleciona a reserva que deseja cancelar. 5. Os dados da reserva são apresentados e é solicitada confirmação. 6. Se a exclusão for confirmada, a reserva é excluída.
<i>Cancelar Reserva Automaticamente</i>	Diariamente, o sistema verifica que reservas já expiraram, i.e, foram comunicadas a mais de 24h para o cliente. Para cada reserva expirada, registrá-la como cancelada. A reserva não é efetivamente excluída, mas apenas marcada como cancelada.

Fluxos Alternativos de Exceção

<i>Efetuar Nova Reserva</i>	1a. – Cliente está em atraso: Uma mensagem de erro é exibida, informando que há itens locados pelo cliente em atraso e apresentando dados desses itens. O fluxo de eventos é abortado. 2a. – Há itens que atendem à reserva (filme e tipo de mídia solicitados) disponíveis para locação: uma mensagem é exibida, informando que não é possível efetuar a reserva, pois há itens que atendem à reserva disponíveis na locadora. O fluxo de eventos é abortado.

Fluxos Alternativos Variantes

<i>Cancelar Reserva</i>	1a. – A reserva foi cancelada pelo sistema: Uma mensagem de erro é exibida, informando que a reserva já foi cancelada automaticamente
--------------------------------	---

	pelo sistema por ter expirado seu prazo. O fluxo de eventos é abortado.
--	---

Requisitos: RF06, RF07, RF08, RNF01, RNF04