

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA E GESTÃO DO CONHECIMENTO**

Marcio Napoli

**APLICAÇÃO DE ONTOLOGIAS PARA APOIAR OPERAÇÕES
ANALÍTICAS SOBRE FONTES ESTRUTURADAS E NÃO
ESTRUTURADAS**

Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia e Gestão do Conhecimento.

Orientador: Denilson Sell, Dr.

Florianópolis
2011

Catálogo na fonte elaborada pela biblioteca
da
Universidade Federal de Santa Catarina

N216a Napoli, Marcio

Aplicação de ontologias para apoiar operações analíticas sobre fontes estruturadas e não estruturadas [dissertação] / Marcio Napoli ; orientador, Denilson Sell. - Florianópolis, SC, 2011.

133 p.: grafs., tabs.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento.

Inclui referências

1. Engenharia e gestão do conhecimento. 2. Inteligência empresarial. 3. Ontologia. 4. Planejamento experimental.
I. Sell, Denilson. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento. III. Título.

CDU 659.2

Marcio Napoli

**APLICAÇÃO DE ONTOLOGIAS PARA APOIAR OPERAÇÕES
ANALÍTICAS SOBRE FONTES ESTRUTURADAS E NÃO
ESTRUTURADAS**

Esta dissertação foi julgada adequada para a obtenção do Título de Mestre em Engenharia e Gestão do Conhecimento, Especialidade em Engenharia do Conhecimento, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina.

Florianópolis, 03 de março de 2011.

Prof. Paulo Maurício Selig, Dr.
Coordenador do Programa - UFSC

Banca Examinadora:

Prof. Denilson Sell, Dr.
Universidade Federal de Santa Catarina
Orientador

Membro Prof. Renato Fileto, Dr.
Universidade Federal de Santa Catarina

Membro Prof. Aran Bey Tcholakian Morales, Dr.
Universidade Federal de Santa Catarina

Membro Prof. Alexandre Leopoldo Gonçalves, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus pais,
Léo e Mariza, a minha irmã Mariana e
principalmente ao amor da minha vida
Adriane

AGRADECIMENTOS

Ao orientador e amigo Denilson Sell que além de acreditar na materialização desse trabalho permitiu a inspiração em sua pesquisa na área de *Business Intelligence*.

Aos membros da banca pela gentileza de participar, revisar e comentar sobre esse trabalho, permitindo uma melhor apresentação da aplicação proposta.

Ao amor da minha vida, Adriane, pelo apoio e carinho não apenas nesse momento de criação, mas em todo o trajeto de vida.

A família pelo apoio, carinho e dedicação constantes, buscando a excelência na educação e, também, pela compreensão de minha ausência em certos momentos.

Aos amigos e colegas que de alguma forma contribuíram direta ou indiretamente para o trabalho.

Aos amigos, do Instituto Stela, que acompanharam e contribuíram para o meu crescimento profissional, em especial ao grupo de *Business Intelligence*.

Ao Instituto Stela pela confiança e por apoiar a formação continuada permitindo ausentar-me nos momentos de dedicação a esse trabalho.

Ao programa EGC por acreditar duas vezes no meu potencial e pela oportunidade de participar nesse programa inovador.

À Universidade Federal de Santa Catarina por permitir a criação de um programa de pós-graduação inovador.

RESUMO

Apesar da importância das ferramentas de processamento analítico on-line (OLAP) para a gestão estratégica, verifica-se que a sua aplicação está restrita às fontes de dados estruturados das organizações. Para analisar os dados não estruturados é necessário tratá-los não apenas como um conjunto de caracteres isolados, mas sim extrair informação desse conteúdo e incluí-la no processo decisório, dessa forma é fundamental trazer o universo textual para as ferramentas de processamento analítico, permitindo realizar as operações OLAP sobre todas as fontes de informação da organização. Com o advento da Web Semântica, surgem novas possibilidades de tratar a integração e exploração dos dados organizacionais. O presente trabalho descreve uma aplicação baseada em ontologias que permite o uso de recursos de processamento analítico como: a análise conjunta de dados estruturados e não estruturados; o acesso transparente as fontes de informação e a exploração dessas fontes por meio de conceitos atrelados a um domínio. Um estudo de caso é apresentado para demonstrar os benefícios da aplicação proposta no domínio da gestão de Ciência & Tecnologia. Ao analisar os resultados da aplicação notamos o enriquecimento das análises, possibilitando uma melhor compreensão do contexto explorado, ao complementá-las com informações oriundas das fontes não estruturadas e, também, a facilidade do usuário ao requisitar informações de forma transparente por meio um modelo de consulta baseado nos conceitos de um domínio.

Palavras-chave: Business Intelligence. OLAP. Ontologia. Dados estruturados. Dados não estruturados.

ABSTRACT

Despite their significant use in strategic management's decision making process, online analytical processing (OLAP) tools are restricted to structured data sources in their application. To analyze unstructured data and it is necessary to not just treat such data as an isolated set of characters, but to extract information from its content and include it in the decision-making process, thus bringing the universe is crucial to textual analytic processing tools allowing perform OLAP operations on all sources of information of the organization. The advent of the Semantic Web, brings new possibilities of dealing with the integration, exploitation and organization of data. This work describes an ontology-based application which allows the use of analytical processing features such as: joint analysis of structured and unstructured data; the transparent access of information sources and browsing sources through concepts linked to a domain. A case study is presented to demonstrate the benefits of implementing proposal Management Science & Technology. When analyzing the results of the application noted the enrichment of analyses, enabling a better understanding of context explored to complement them with data from unstructured sources, and also the ease of the user to request information through a query model based on the concepts of a domain.

Keywords: Business Intelligence. OLAP. Ontology. Structured data. Unstructured data.

LISTA DE FIGURAS

Figura 1 Visão da arquitetura de Business Intelligence	29
Figura 2 Exemplo de um cubo de dados OLAP	32
Figura 3 Exemplo de um processo de extração, transformação e carga.....	33
Figura 4 Ilustração mostrando à tripla RDF.....	44
Figura 5 Apresentando as camadas da Web Semântica	47
Figura 6 Camadas da arquitetura do framework PARMENIDES.....	51
Figura 7 Camadas da arquitetura SBI	52
Figura 8 Arquitetura dos componentes da aplicação proposta.....	55
Figura 9 Hierarquia dos conceitos presentes na ontologia de MAPEAMENTO	59
Figura 10 Hierarquia dos conceitos presentes na ontologia ANALÍTICA..	62
Figura 11 Diagrama explicando a geração das mensagens de requisição (RequestMessage)	64
Figura 12 Diagrama explicando os passos executados pelo Driver	66
Figura 13 Principais módulos de preenchimento do Currículo Lattes	70
Figura 14 Documentos temáticos disponíveis no aplicativo Coleta.....	71
Figura 15 Visualização dos conceitos e propriedades da ontologia de DOMÍNIO.....	73
Figura 16 Visualização de instâncias da classe DBCollection presentes na ontologia de MAPEAMENTO.....	74
Figura 17 Visualização de instâncias da classe TextAttribute presentes na ontologia de MAPEAMENTO.....	75
Figura 18 Fluxo representando o acesso aos dados estruturados	76
Figura 19 Visualização de uma tabela multidimensional com dados de uma base relacional.....	77
Figura 20 Fluxo simplificado representando o combinação entre fontes de dados estruturados e não estruturados textuais.....	78
Figura 21 Visualização do retorno de uma consulta combinando dados de ambas as bases	80
Figura 22 Requisição aplicando as operações drillDown e sort sobre a combinação de dados estruturados e não estruturados textuais.....	81
Figura 23 Visualização do retorno de uma consulta aplicando as operações drillDown e sort	83

LISTA DE QUADROS

Quadro 1 Principais operações OLAP	38
Quadro 2 Comparação entre MOLAP vs ROLAP	41
Quadro 3 Descrição dos principais conceitos reunidos na ontologia de MAPEAMENTO	58
Quadro 4 Descrição dos principais conceitos reunidos na ontologia ANALÍTICA.....	61

LISTA DE ABREVIATURAS

- BI - Business Intelligence
- CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
- CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
- CRM - Customer Relationship Management
- C&T – Ciência & Tecnologia
- DM – Data Mart
- DW - Data Warehouse
- ERP - Enterprise Resource Planning
- OIL – Ontology Inference Layer
- OLAP - On-Line Analytical Processing
- OWL - Web Ontology Language
- RDF – Resource Description Framework
- SBI – Semantic Business Intelligence
- SQL – Structured Query Language
- URI – Uniform Resource Identifier

SUMÁRIO

1 INTRODUÇÃO	21
1.1 Definição do problema.....	23
1.2 Objetivos.....	23
1.3 Justificativa.....	23
1.4 Aderência ao objeto de pesquisa do programa.....	24
1.5 Metodologia.....	24
1.6 Escopo do trabalho.....	25
1.7 Estrutura do trabalho.....	26
2 FUNDAMENTAÇÃO TEÓRICA	27
2.1 Introdução.....	27
2.2 Tipos de Dados.....	27
2.3 Business Intelligence.....	28
2.4 OLAP (On Line Analytical Processing).....	35
2.4.1 Metodologia de Armazenamento	38
2.5 Universo Textual na abordagem OLAP.....	41
2.6 Ontologia.....	43
2.6.1 RDF (Resource Description Framework).....	44
2.6.2 OIL (Ontology Inference Layer).....	45
2.6.3 DAML (DARF Agent Markup Language).....	46
2.6.4 OWL.....	46
2.7 Integração semântica.....	48
2.8 Abordagens de Business Intelligence Baseados em Tecnologias Semânticas	49
2.8.1 Semântica de fontes de notícias para aplicações de BI.....	50
2.8.2 PARMENIDES	50
2.8.3 SBI (Semantic Business Intelligence).....	51
2.9 Considerações sobre o capítulo	53
3 APLICAÇÃO PROPOSTA	55
3.1 Apresentação da Aplicação.....	56
3.2 Ontologias	56
3.3 Gerenciador de Ontologias.....	63
3.4 Requisições de informação	63
3.5 Mensagem de requisição (RequestMessage).....	65
3.6 Gerenciador de Drivers	65
3.7 Fluxos de Mensagens	67
3.8 Operações OLAP	67
3.9 Considerações sobre o capítulo	68

4 ESTUDO DE CASO APLICADO NO DOMÍNIO DE GESTÃO DA CIÊNCIA & TECNOLOGIA.....	69
4.1 Currículo Lattes.....	69
4.2 Aplicativo Coleta de Dados da CAPES.....	71
4.3 Preparação do ambiente.....	72
4.4 Acesso Transparente às fontes de informação.....	75
4.5 Realizando uma consulta combinando dados de ambas as bases	77
4.6 Aplicando operações OLAP.....	81
4.7 Considerações sobre o capítulo	84
5 CONCLUSÃO	85
5.1 Contribuições	86
5.2 Limitações e trabalhos futuros	86
REFERÊNCIAS	88
APÊNDICE A - Ontologia de Domínio	97
APÊNDICE B - Ontologia de Análise.....	99
APÊNDICE C - Ontologia de Mapeamento.....	114

1 INTRODUÇÃO

O crescimento das organizações é acompanhado da proliferação de fontes de dados isoladas, ocasionando dificuldades no processo decisório e criando um hiato sobre todo o conhecimento organizacional (WANG et al., 2008).

As instituições estão investindo em tecnologia para gerenciar artefatos de informação, um dos objetivos é quebrar a barreira entre dados estruturados¹ e não estruturados², permitindo distribuir todas as informações relevantes, não importando o formato ou sua localização (EVELSON, BROWN, 2008, p. 14). Os dados estruturados possuem seu tipo definido e podem ser geridos de forma eficiente. O dado não estruturado, em contraste, é de fluxo livre e sem tipo, geralmente consultado utilizando um conjunto de palavras-chave (ROY et al., 2005). Esta diferença inerente à maneira como as duas classes de dados são gerenciadas e consultadas dificulta o processo decisório haja vista que tais classes são complementares (ROY et al., 2005).

As ferramentas de processamento analítico online (OLAP) são importantes no processo decisório, permitindo analisar os dados de forma flexível e interativa, ajudando os analistas a sintetizarem informações sobre a organização, por meio de comparações, visões personalizadas, análises históricas e projeções de dados (KIMBALL et al., 1998).

Lamentavelmente as ferramentas OLAP tradicionais são capazes de manipular apenas dados estruturados, portanto é importante estender essa visão e permitir analisar dados não estruturados e combiná-los entre si, possibilitando executar operações analíticas sobre todas as fontes de informação da organização (ZHANG et al., 2009; NAPOLI et al., 2006; NAPOLI; SELL, 2010).

Para analisar os dados não estruturados é necessário tratá-los não apenas como um conjunto de caracteres isolados, mas sim extrair informação desse conteúdo e incluí-la no processo decisório, dessa forma é fundamental trazer o universo textual para as ferramentas de processamento analítico (ÁVILA, 2010; FOLGOSO, 2008).

¹ Dados estruturados são dados que vem repetidamente no mesmo formato e leiaute (INMON et al., 2008)

² Os dados não estruturados existem na forma textual e não textual (INMON et al., 2008)

Para tornar o processo de tomada de decisão efetivo é necessário acessar a informação na sua totalidade e as organizações estão rapidamente percebendo a necessidade de superar este hiato. A partir do advento da Web Semântica (BERNERS-LEE; HENDLER; LASSILA, 2001) verifica-se na literatura um crescente registro de iniciativas baseadas em ontologias para a integração dos dados estruturados e não estruturados (ROY et al., 2005; IZZA; VICENT; BURLAT, 2008; WANG et al., 2008; KETTER et al., 2008). Verifica-se, no entanto, pouca ênfase na literatura para a criação de abordagens para processamento analítico sobre dados não estruturados.

O uso de ontologias também pode apoiar a recuperação da informação por meio da reescrita de consultas, buscando otimizar, decompor e traduzir as demandas informacionais através do uso de inferências e/ou processamento de regras (CORRENDO et al., 2010).

Nesse trabalho apresentamos uma aplicação do uso de ontologias para integrar dados não estruturados com estruturados e permitir à realização de operações analíticas sobre esses dados.

Para demonstrar o resultado da aplicação proposta utilizou-se o cenário da Gestão de Ciência & Tecnologia, por meio de duas bases, uma representando o histórico curricular, pertencente ao Currículo Lattes (CNPQ, 2011), e outra proveniente do aplicativo Coleta da CAPES (CAPES, 2009).

Nesse cenário as fontes não estruturadas textuais são representadas por meio de dados como a descrição de cada produção e o *resumé* de cada pessoa, disponível na base do Currículo Lattes. Enquanto campos como a grande área de atuação e o ano da produção foram utilizados como dados estruturados. Para permitir a combinação entre ambas as bases utilizou-se o campo comum referente ao nome da pessoa.

A estratégia utilizada para inserir na análise o universo de dados não estruturados textuais foi permitir realizar operações analíticas por meio de filtros textuais. O campo estruturado “nome da pessoa” responsável pela ligação entre os dois tipos de dados foi associado com os dados não estruturados textuais, permitindo através de uma busca textual retornar informações estruturadas.

Como características dessa aplicação pode-se destacar o acesso transparente à informação; a realização de operações analíticas sobre dados estruturados e não estruturados; e a exploração dessas fontes por meio dos conceitos definidos numa ontologia de domínio.

1.1 DEFINIÇÃO DO PROBLEMA

Segundo relatório da IDC (2007) as fontes não estruturadas representam 80% de todas as fontes disponíveis por uma instituição. O modelo atual para realizar operações OLAP limita-se ao uso de fontes estruturadas (ZHANG et al., 2009; LIN et al. 2008), impedindo que o tomador de decisão tenha acesso à informação na sua totalidade. Com base nessa premissa há a necessidade de buscar na engenharia do conhecimento os recursos para realizar o processamento analítico sobre dados estruturados e não estruturados. Para isso é necessário tratar o texto não apenas como um conjunto de caracteres isolados, mas sim extraindo informação desse conteúdo para melhorar o processo decisório. Dessa forma é fundamental trazer o universo textual para as ferramentas de processamento analítico, permitindo realizar as operações OLAP sobre todas as fontes de informação da organização.

1.2 OBJETIVOS

O objetivo geral do presente trabalho é utilizar ontologias para prover a integração entre fontes de dados estruturados e não estruturados, permitindo realizar operações analíticas sobre essas fontes.

Para alcançar esse objetivo geral, são estabelecidos os seguintes objetivos específicos:

- Propor uma descrição semântica padrão para mapear fontes de informação de diferentes tipos;
- Identificar estratégias baseadas em ontologias para apoiar a navegação sobre fontes de informação e o apoio à execução de operações analíticas;
- Criar uma aplicação que permita analisar e combinar as fontes estruturadas com as não estruturadas;
- Desenvolver um protótipo que demonstre a viabilidade da aplicação proposta.

1.3 JUSTIFICATIVA

As ferramentas de processamento analítico online (OLAP) são importantes no processo decisório, permitindo analisar os dados de forma flexível e interativa, ajudando os analistas a sintetizarem informações sobre a organização, por meio de comparações, visões

personalizadas, análises históricas e projeções de dados (KIMBALL et al., 1998).

Apesar da importância das ferramentas de processamento analítico on-line (OLAP) para a gestão estratégica, verifica-se que a sua aplicação está restrita às fontes de dados estruturados das organizações (ZHANG et al., 2009; DING et al., 2010).

Segundo relatório da IDC (2007) as fontes não estruturadas representam 80% de todas as fontes disponíveis por uma instituição, portanto é importante estender essa visão e permitir analisar também os dados não estruturados e combiná-los entre si, possibilitando executar operações analíticas em todas as fontes de informação da organização, objetivando melhor apoiar o processo decisório (ZHANG et al., 2009; BHITE et al., 2008).

1.4 ADERÊNCIA AO OBJETO DE PESQUISA DO PROGRAMA

Os sistemas de conhecimento são destinados a apoiar as decisões de forma mais efetiva (SCHREIBER et al., 1999). A aplicação proposta utiliza a representação explícita de conhecimento na sua arquitetura para prover meios mais eficientes de explorar as fontes de informação para apoiar a tomada de decisão.

Sendo assim, a interdisciplinaridade da pesquisa é fundamenta em áreas e métodos tais como a Gestão Estratégica, Recuperação de Informação, *Business Intelligence* e Web Semântica. Considera-se este projeto aderente a área da Engenharia do Conhecimento por utilizar a formalização e codificação de conhecimento na aplicação proposta (EGC 1, 2010). Além do uso de ontologias, esta proposta viabiliza a descoberta do conhecimento através da realização de operações analíticas sobre as fontes de informação estruturadas e não estruturadas.

Verifica-se também, ao analisar os objetivos específicos, a similaridade com a linha de pesquisa denominada de Engenharia do Conhecimento aplicada às organizações, que visa à concepção, desenvolvimento e implantação de soluções da Engenharia do Conhecimento em organizações (EGC 2, 2010).

1.5 METODOLOGIA

Quanto à natureza, este trabalho é considerado uma pesquisa aplicada, pois tem como objetivo criar uma aplicação prática dirigida à solução de problemas específicos, demonstrado na justificativa desta pesquisa. Quanto à abordagem do problema, este trabalho é considerado

qualitativo, pois está baseado em um modelo de desenvolvimento em que não é necessária uma análise estatística para qualquer comprovação (SILVA; MENEZES, 2001, p.20).

Do ponto de vista de seus objetivos, a pesquisa é exploratória, pois visa proporcionar uma maior relação com o problema por meio do levantamento bibliográfico e de exemplos já existentes. Do ponto de vista dos procedimentos técnicos, esta pesquisa pode ser classificada como bibliográfica e de estudo de caso (SILVA; MENEZES, 2001, p.21).

O presente trabalho possui como visão de mundo o paradigma funcionalista, pois procura examinar regularidade e relações que levam a generalizações e princípios universais (BURREL; MORGAN, 1979) e, como modalidade de pesquisa a pesquisa tecnológica na qual não há teorização, haja vista que o objetivo do trabalho visa à criação de um instrumento tecnológico visando uma solução para uma demanda ou necessidade preestabelecida.

Para tratar o problema de pesquisa da dissertação, adotaram-se as seguintes etapas:

- Estudo teórico sobre negócios inteligentes (Business Intelligence).
- Estudo sobre o processamento analítico (OLAP).
- Estudo sobre processamento analítico (OLAP) aplicado sobre o universo textual.
- Estudo sobre ontologias, tipos de ontologias e o uso de ontologias como meio integrador.
- Apresentação da aplicação proposta.
- Apresentação de um estudo de caso no contexto de Ciência & Tecnologia.

1.6 ESCOPO DO TRABALHO

O presente trabalho limita-se a propor uma solução capaz de acessar diversos repositórios, inclusive heterogêneos. Porém o trabalho não se preocupa em como os dados, sejam estruturados ou não estruturados, são indexados ou armazenados e sim como podem ser acessados de forma transparente. Uma forma de armazenamento é explicada em detalhes em Beppler et al. (2005).

No caso das fontes não estruturadas o trabalho se limita as fontes textuais e não aborda como é realizado o processo de análise e reconhecimento de entidades presentes no conteúdo haja vista a

complexidade desse problema. Uma forma de extração e reconhecimento de entidades é apresentada em detalhes em Ceci (2010).

A combinação das fontes de informação se limita às operações básicas de conjunto como união, interseção e diferença. Sobre a característica de acesso transparente às fontes de informação, a aplicação proposta abstrai a localização, ou seja, o usuário solicita a informação sem se preocupar quais fontes devem ser acessadas para retornar a informação solicitada.

1.7 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em cinco capítulos organizados de acordo com os temas relacionados ao desenvolvimento deste trabalho. Após este primeiro capítulo organizaram-se os seguintes em:

- Capítulo 2 (Fundamentação Teórica) – Discorre sobre os principais conceitos utilizados nessa pesquisa como Business Intelligence, Ontologia e Integração Semântica. Complementamos essa revisão teórica com pesquisas similares na tratativa do universo não estruturado nas operações OLAP e destacamos algumas pesquisas que influenciaram o desenvolvimento desse trabalho;
- Capítulo 3 (Aplicação Proposta) – Apresenta a interação entre os principais componentes da aplicação proposta, bem como suas funcionalidades;
- Capítulo 4 (Estudo de Caso Aplicado no Domínio da Gestão da Ciência & Tecnologia) – Esse capítulo descreve um cenário de uso da aplicação proposta no contexto C&T do Currículo Lattes e, também, do aplicativo Coleta da CAPES;
- Capítulo 5 (Conclusão) – Por fim, esse capítulo apresenta os resultados obtidos da aplicação proposta, além dos benefícios e sugestões para possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Diante dos objetivos apresentados na seção anterior é necessário vislumbrar como construir uma aplicação capaz de realizar as operações OLAP sobre as fontes estruturadas e não estruturadas. Nesse sentido exploramos o conceito de *Business Intelligence* para entender melhor seus componentes como: Data Warehouse, a modelagem dimensional e a exploração analítica por meio dos cubos multidimensionais. Visitamos trabalhos que fazem uso das operações OLAP em abordagens textuais para verificar o estado da arte e entender o universo não estruturado. Em seguida destacamos as ontologias como forma de integração entre os dados e finalizamos o tópico com algumas abordagens que utilizam a semântica no contexto de *Business Intelligence*.

2.1 INTRODUÇÃO

Nesse tópico definem-se os tipos de dados e apreentam-se os conceitos vinculados a *Business Intelligence*, destacando as operações analíticas oferecidas pela tecnologia OLAP. Como meio para a criação de descrições semânticas cita-se o uso de ontologia. Para permitir a visão unificada dos dados comenta-se sobre a integração de dados e, por último, destaca-se a arquitetura SBI que inspirou a criação desse trabalho.

2.2 TIPOS DE DADOS

Existem dois tipos básicos de dados, os estruturados e não estruturados. Dados estruturados são dados que vem repetidamente no mesmo formato e leiaute, geralmente oriundos dos sistemas transacionais (INMON et al., 2008). Enquanto dados não estruturados refere-se ao fato de que nenhuma estrutura identificável está disponível, seja por meio de metadados ou pelo seu próprio conteúdo (SINT et al., 2009).

Dados estruturados são convenientemente armazenados em registros na base de dados, onde existem atributos, chaves, índices, tabelas e assim por diante. Na verdade, esse universo dos dados estruturados é servido por tecnologias já padrões de mercado (INMON et al., 2008).

Exemplos típicos de dados estruturados incluem dados gerados por transações bancárias, transações de reserva aérea, as operações de

seguros, operações de fabricação, operações de varejo e assim por diante (INMON et al., 2008).

Segundo Inmon et al (2008) existe o universo dos dados não estruturados. Os dados não estruturados existem na forma textual e não textual. Dados não estruturados textuais ocorrem em vários locais como e-mails, conversas via *chat*, apresentações e muito mais. Dados não estruturados que não são textuais são geralmente elementos gráficos e imagens, incluindo, mas não se limitando a, fotografias, radiografias, ressonâncias magnéticas, diagramas e ilustrações.

Enquanto a tecnologia atual ainda é restrita em lidar com dados não textuais de forma simples, os dados não estruturados textuais podem ser capturados e manipulados. Tecnologias para manipular dados textuais possuem um grande desafio, haja vista que esse tipo de dados não são repetitivos como os dados estruturados. No entanto, os dados textuais possuem um grande valor agregado (INMON et al., 2008).

De fato os dados não estruturados textuais se dividem em duas classes: dados não estruturados e semiestruturados. Dados não estruturados são representados pelo texto escrito de forma livre. Um livro, um manual ou um curso de formação tem muitas vezes grandes quantidades de texto não estruturado. Dados semiestruturados são dados textuais, porém existem repetições do formato do texto. Por exemplo, um livro de receitas é um documento único, mas dentro do livro de receitas existem muitas receitas. Cada receita tem seus próprios ingredientes e suas próprias instruções. Cada receita é uma forma de dados semiestruturados (INMON et al., 2008).

Sint et al. (2009) define dados semiestruturados como dados sem esquemas ou que são autodescritos, pois não há nenhum metadados externo do tipo ou estrutura de dados. Os dados semiestruturados não requerem uma definição por meio do esquema, mas isso não significa que sua definição não seja possível. Nesse tipo de dado as instâncias continuam existindo mesmo em que haja mudanças no esquema.

2.3 BUSINESS INTELLIGENCE

O conceito de BI (*Business Intelligence*), de forma mais ampla, pode ser entendido como a utilização de várias fontes de informação para definir estratégias de competitividade nos negócios da empresa. O objetivo maior das técnicas de BI está na definição de regras e técnicas para a formatação adequada dos dados, visando transformá-los em depósitos estruturados de informações, independentes de sua origem. Esse objetivo pode ser atingido por meio da definição de estruturas

modeladas dimensionalmente, armazenadas em *Data Warehouse* ou *Marts* e interpretadas pela ótica analítica das ferramentas OLAP (*OnLine Analytical Processing*) ou pela visão das ferramentas de *Data Mining* (BARBIERI, 2001).

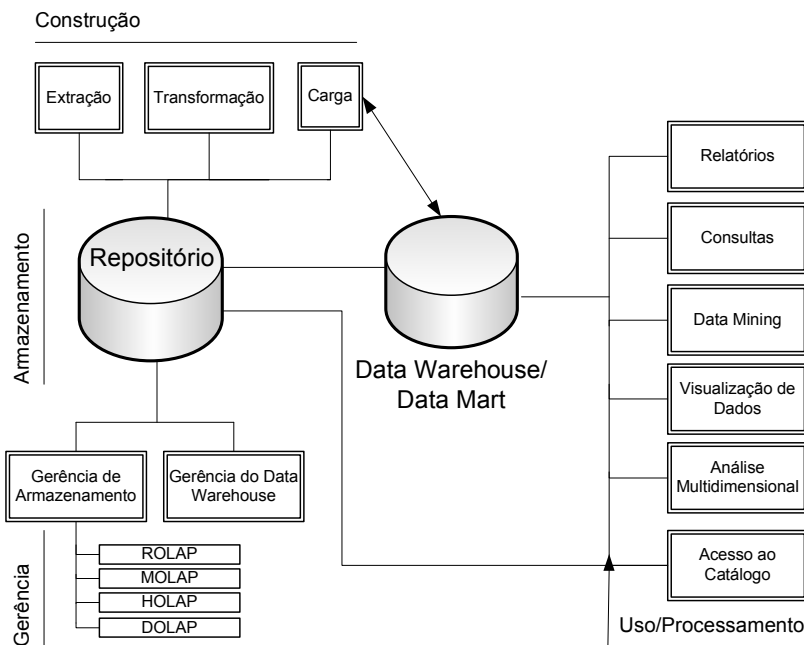


Figura 1 Visão da arquitetura de Business Intelligence

Fonte: Adaptado de Barbieri (2001)

Depois de muito progresso e desenvolvimento, Business Intelligence tem sido capaz de provar sua importância nos negócios modernos. Como resultado dos avanços e de seu impacto, o ano de 2006 é reconhecido como o surgimento da versão 2.0. Esta nova forma de Business Intelligence permite construir um software analítico mais desenvolvido, sendo mais pró-ativo do que reativo em relação à versão anterior (CHAIDEZ, 2008; NELSON, 2010).

Enquanto a primeira versão do Business Intelligence trata principalmente a análise de informações em decisões de negócios e o apoio há decisões já tomadas, esta nova versão busca auxiliar a tomada de decisões *ad hoc*. A versão 2.0 inclui recursos de visualização que permitem aos usuários analisar e manipular complexas relações de

dados, uma maior interatividade e uma nova forma de trabalho que se adapta conforme o raciocínio do usuário de negócios (NELSON, 2010; CZERNICKI, 2010).

BI 2.0 visa trazer informação para o usuário rapidamente sem ter que exigir uma profunda compreensão das ferramentas analíticas. Os usuários estão aptos para adquirir conhecimento e tomar decisões. Nessa versão o conhecimento e a informação certa estão disponíveis num formato simples e o uso efetivo do conhecimento permite aos usuários realizarem decisões estratégicas. Por meio dos modelos semânticos a nova versão facilita encontrar a informação necessária, não importando se as informações são estruturadas ou não. Para isso, a aplicação visual deve ser simples e fácil de entender (CZERNICKI, 2010).

Portanto, essas características permitem um novo nível de colaboração nas empresas que reduz o espaço entre a análise e a ação. Sendo assim, mesmo que os atuais recursos de BI são uma boa opção para muitas tarefas atuais, esta nova versão ajudará estendê-lo a um horizonte totalmente diferente (CHAIDEZ, 2008).

2.3.1 Data Warehouse

Inmon (1997) define o *Data Warehouse* como sendo um conjunto de dados integrado, não volátil e variável em relação ao tempo. Esses dados são organizados por assuntos e a análise desses dados apóia a tomada de decisões gerenciais.

De acordo com Kimball e Ross (2002), os principais objetivos de um Data Warehouse são:

- permitir fácil acesso à informação, com conteúdo intuitivo e significado evidente para os usuários, utilizando-se nomenclatura correta e total legibilidade das informações armazenadas;
- apresentar informações consistentes e confiáveis que, mesmo tendo sido originadas de diversas fontes de dados, tenham passado por um processo de limpeza e verificação de qualidade, sendo disponibilizadas aos usuários somente quando estiverem apropriadas para o consumo;
- ser adaptável a mudanças, originadas de alterações das necessidades dos usuários, das condições de negócio, dos dados ou da própria tecnologia;

- garantir a segurança ao acesso das informações armazenadas, já que muitas vezes são consideradas confidenciais;
- armazenar dados que servirão de base para o processo de tomada de decisão;
- ser aceito pela comunidade de usuários, pois, uma vez que difere de um sistema de informação operacional, cujo uso torna-se praticamente obrigatório, o DW é uma ferramenta opcional.

Assim como Business Intelligence evoluiu em sua versão houve a necessidade de criar um novo paradigma para a construção de armazéns de dados, denominado DW 2.0. Este novo paradigma para armazenagem de dados que incide sobre os tipos básicos de dados (estruturados e não estruturados), sua estrutura e como eles se relacionam, visa formar um poderoso armazém de dados (INMON et al., 2008).

Na primeira geração houve ênfase em obter o armazém de dados construído sobre os dados que geram valor para os negócios, porém eram restritos aos dados estruturados, nessa nova versão entende-se que para agregar maior valor existe há necessidade de incluir dados textuais não estruturados na armazenagem (INMON et al., 2008).

Segundo Kimball (1998), a tradicional abordagem relacional altamente normalizada, utilizada com sucesso no projeto de sistemas para o ambiente operacional, não é adequada para a construção de um *data warehouse*. No modelo relacional, a estrutura é otimizada para recuperar, criar e atualizar registros individuais em tempo real e preservar a integridade dos dados. No *data warehouse*, a utilização dos dados é muito diferente. As consultas realizadas recuperam um grande número de registros e os resume, durante o processamento. Neste caso, a estrutura é denominada modelo dimensional que normalmente não é normalizada para evitar junções de muitas tabelas, obtendo-se assim uma performance superior.

2.3.1.1 Modelo Dimensional

Kimball e Ross (2002) definem a modelagem dimensional como uma técnica estruturada desenvolvida para a obtenção de um modelo de dados dimensional. Na sua composição típica esse modelo possui uma grande entidade central (fato) que reflete a evolução dos negócios do dia-a-dia de uma organização e um conjunto de entidades menores

(dimensões) arranjadas ao redor da entidade central e utilizadas de forma combinada como variáveis de análise do fato.

Os fatos são tabelas que armazenam medidas numéricas totalizáveis relacionadas a um determinado assunto ou processo de negócio. Cada registro em uma tabela de fato está relacionado a um conjunto de dimensões que determinam a granularidade dos fatos armazenados e definem qual o escopo dessas medidas. Quanto menor a granularidade de um fato, maior será o nível de detalhe armazenado (KIMBALL; ROSS, 2002). Já as dimensões armazenam descrições textuais de um processo de negócio, podendo cada uma ser formada por um conjunto de atributos, denominados hierarquias, que servem como base para determinar regras de agrupamento, quebras e filtros em consultas a uma tabela de fato (KIMBALL; ROSS, 2002).

Segundo Datta e Thomas (1999) um cubo de dados é um elemento fundamental na construção de banco de dados multidimensionais e serve como uma unidade básica de entrada e saída para todos os operadores definidos em um banco de dados multidimensional. Um cubo contém células que são medidas ou valores baseados num conjunto de dimensões.



Figura 2 Exemplo de um cubo de dados OLAP
Fonte: Nardi (2010).

A visão multidimensional dos dados oferece vantagens importantes como a possibilidade de visualizar as medidas de seu negócio a partir de diferentes perspectivas; é um modo intuitivo de trabalhar com dados agregados ou sumarizados; facilita identificar padrões e tendência nos dados e permite o usuário explorar a informação apresentada (HOPE et al., 2000).

2.3.1.2 Extração, Transformação e Carga (ETL)

As fontes de dados para aplicações BI são oriundas de uma variedade de sistemas operacionais e aplicações. A finalidade do processo ETL é fundir os dados destas plataformas heterogêneas em um formato padrão no ambiente de suporte a decisão. Esse processo é definido em três etapas: extração, transformação e carga (MOSS; SHAKU, 2003).

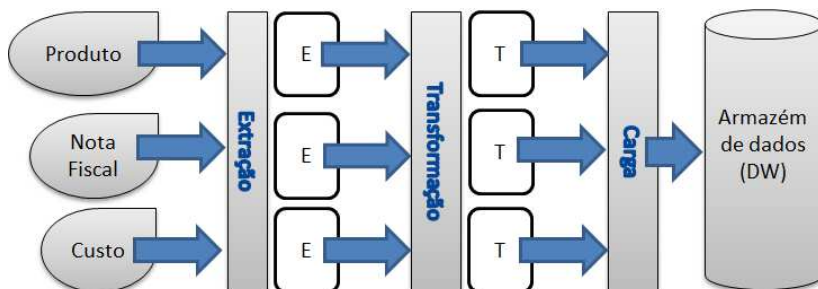


Figura 3 Exemplo de um processo de extração, transformação e carga
Fonte: Adaptado de Moss e Shaku (2003).

Da perspectiva do projeto de BI, a maneira mais favorecida é criar extrações que podem ser classificadas, filtradas, limpas e agregadas em apenas um passo. Entretanto, em algumas organizações isso impactaria tal nível que as operações do negócio teriam que ser suspensas por diversas horas (MOSS; SHAKU, 2003). A solução é criar programas de extração e projetá-los o mais eficiente possível. O objetivo é evitar que as funções diárias do negócio sejam afetadas.

Selecionar e fundir as fontes de dados pode ser uma tarefa difícil devido ao alto nível de redundância nos sistemas. Os programas de extração devem saber quais dados são redundantes, por exemplo, o mesmo elemento de dado pode existir em dezenas de bases.

Os programas de transformação podem se tornar complexos quando os dados são extraídos de um ambiente heterogêneo. Alguns problemas são descritos abaixo:

- Inconsistência nas chaves primárias: A chave primária dos registros de origem não combinam com a chave primária das tabelas do BI;
- Inconsistência nos dados: Os elementos de dados tais como datas e moedas podem ser armazenadas em formatos completamente diferentes que devem ser armazenados na base de dados do BI;
- Dados inválidos: A lógica de limpeza tem que ser definida para corrigir valores inapropriados e deve ser aplicada a cada ciclo desse processo;
- Sinônimos e homônimos: Os dados redundantes não são sempre fáceis de reconhecer porque o mesmo elemento de dados pode ter nomes diferentes. Tanto os sinônimos quanto os homônimos não devem existir num ambiente de BI, esses elementos devem ser renomeados na migração para um termo específico;
- Lógica de processo embutido: Alguns sistemas são extremamente antigos, eles rodam, mas ninguém sabe como. Frequentemente contêm códigos que identificam lógica de processo, por exemplo, o valor “0” significa que a mercadoria voltou, enquanto o valor “1” identifica a entrega correta. Então, o processo de transformação deve refletir esta lógica.

Segundo Moss e Shaku (2003) a etapa final no processo de ETL em comparação com o processo de transformação é bem mais simples, entretanto ainda é necessário tomar decisões sobre a integridade referencial e a indexação.

Por causa do grande volume de dados, muitas organizações preferem desligar a checagem da integridade referencial para aumentar a velocidade no processo de carga. Contudo, devem executar as verificações necessárias, se não, as bases de dados podem tornar-se inconsistentes rapidamente. A verificação da integridade referencial não é necessária para aplicações BI porque nenhum relacionamento novo dos dados é criado e somente os dados operacionais existentes são carregados. A corrupção ocorre principalmente porque os dados operacionais não são relacionados corretamente em primeiro lugar,

especialmente quando os dados operacionais não estão numa base relacional. Contudo deve-se ligar a integridade referencial após o processo de carga para verificar violações entre as tabelas dependentes (MOSS; SHAKU, 2003).

Baixa performance no processo de carga na base de dados pode ser problema no uso da indexação, pois para construir os índices enquanto o processo de carga é realizado, existem muitas operações a serem realizadas no cálculo desse índice. Assim, é aconselhável excluir todos os índices antes do processo de carga e, ao final, recriá-los (MOSS; SHAKU, 2003).

2.4 OLAP (On Line Analytical Processing)

Segundo Kimball et al. (1998), OLAP constitui-se das atividades gerais e específicas de consulta e apresentação de dados numéricos e textuais provenientes do DW. Os sistemas OLAP utilizam a visão multidimensional dos dados, sendo ela natural, fácil e intuitiva, permitindo a visualização desses dados em diferentes perspectivas de negócios de uma organização.

Basicamente, as informações são representadas de forma a permitir seu livre manuseio. Essa representação deve oferecer informações estratégicas para o analista de negócio omitindo a preocupação com detalhes tecnológicos do sistema. Os sistemas OLAP ajudam analistas a sintetizarem informações sobre a organização, por meio de comparações, visões personalizadas, análises históricas e projeções de dados, oferecendo respostas rápidas e consistentes às consultas ad-hoc.

Numa seção típica OLAP, um usuário deseja obter informações (medidas de um fato) de várias perspectivas (dimensões) e em diferentes níveis de detalhe. Por exemplo, o valor de vendas (medidas de um fato) pode ser analisado por filial, por produto e por mês (dimensões). Um OLAP pode oferecer diversas funcionalidades para visualizar as informações de diferentes formas, sendo cada uma delas padronizada com um nome. Entre as principais funcionalidades disponibilizadas pelas ferramentas OLAP, destacam-se as apresentadas no Quadro a seguir.

OPERAÇÃO	DESCRIÇÃO	EXEMPLO
<i>Drill Down</i>	Aumentar o nível de detalhe da informação e, conseqüentemente, diminuir o grau de granularidade (PARENTE, 2000).	Uma análise de vendas por Estado é alterada para uma análise de vendas das cidades de um Estado específico.
<i>Drill Up</i>	Diminuir o nível de detalhe e aumentar o grau de granularidade (PARENTE, 2000).	Uma análise de vendas é alterada de uma cidade para seu Estado correspondente.
<i>Slice</i>	Cortar o cubo, mas manter a mesma perspectiva de visualização dos dados. Funciona como um filtro restringindo uma dimensão a somente algum ou alguns de seus valores (PARRINI, 2002).	Em uma dimensão tempo de um modelo, é selecionado somente o ano de 2000.
<i>Dice</i>	Mudar a perspectiva da visão multidimensional. Mostrar o inter-relacionamento entre unidades de informação distintas, na qual cada unidade de informação representa um eixo de análise (BALLARD et all, 2006).	A análise é alterada acrescentando mais um atributo de uma dimensão no eixo da coluna.

<i>Drill Across</i>	É a ligação de duas ou mais tabelas de fato em uma mesma granularidade, ou seja, tabelas com o mesmo grupo de medidas e dimensões (KIMBALL, 1996).	Passar a visualizar a medida “total de vendas” de uma tabela de fatos chamada “Fato_Vendas” para outra “Fato_Pedido_Vendas” (através de uma tabela de dimensão Dim_Tempo).
<i>Drill Through</i>	Ocorre quando o usuário passa de uma informação contida em uma dimensão para uma outra (PARENTE, 2000).	O usuário está realizando uma análise na dimensão de tempo e no próximo passo analisa a informação por região (PARENTE, 2000).
<i>Drill Out</i>	É o detalhamento para informações externas, como fotos, som, arquivos-texto, tabelas.	O usuário seleciona um cliente ou fornecedor para visualizar um resumo sobre seus dados.
<i>Sort</i>	Tem a função de ordenar a informação (PARENTE, 2000), podendo ser aplicada a qualquer tipo de informação, não somente a valores numéricos.	Ordenar as instituições em ordem alfabética.
<i>Ranking</i>	Permite agrupar resultados numéricos por ordem de tamanho (PARENTE, 2000), refletindo somente na apresentação do resultado e não no resultado em si.	Ordenar a relação de filiais de acordo com os maiores volumes de vendas.

<i>Paging</i>	Apresentação dos resultados de uma consulta em várias páginas, permitindo a navegação do usuário.	Permitir um máximo de 10 resultados por página.
<i>Filtering</i>	Apresentação de consultas com restrições sobre atributos ou fatos, ou seja, restringe um fato a somente algum ou alguns de seus valores.	Em um fato Vendas, apenas os valores das vendas a prazo são apresentados.
<i>Alerts</i>	Utilizados para indicar situações de destaque em elementos dos relatórios, baseados em condições envolvendo objetos e variáveis (PARENTE, 2000).	Definir que os valores das vendas mensais inferiores a R\$ 50.000,00 devem aparecer com destaque em vermelho.
<i>Break</i>	Permite separar o resultado de uma análise em grupos de informações (PARENTE, 2000), possibilitando também a subtotalização de valores para cada grupo.	O usuário tem a necessidade de visualizar a informação por cidades, então ele solicita um <i>break</i> . O relatório será automaticamente agrupado por cidades, somando os valores mensuráveis por cidades (PARENTE, 2000).

Quadro 1 Principais operações OLAP

Fonte: Adaptado de Gonzaga (2005).

2.4.1 Metodologia de Armazenamento

Segundo Golfarelli e Rizzi (2009) existem três acrônimos para as principais abordagens para implementar datawarehouses e eles estão relacionados com o modelo lógico para representar os dados.

MOLAP é acrônimo para Multidimensional Online Analytical Processing. MOLAP armazena os valores agregados numa estrutura multidimensional. Um valor agregado é pré-calculado melhorando significativamente o tempo de resposta, pois possuem as repostas das perguntas antes que sejam feitas. Por exemplo, quando um fato do datawarehouse contiver centenas dos milhares de registros, uma consulta que pede os totais semanais das vendas para um produto em particular pode gastar muito tempo se tiver que realizar uma varredura na tabela. Entretanto, a resposta pode ser quase imediata se os dados para responder essa consulta já tenham sido pré-calculados. As tabelas de fato são tabelas centrais que contêm todos os registros da transação num modelo multidimensional (RASMUSSEN, 2002).

A maioria dos valores na célula de um cubo OLAP não existe na fonte de dados. A fonte de dados possui a informação somente no nível do detalhe. A estrutura OLAP permite a agregação (pré-calculada) dos valores ao longo de cada dimensão no cubo OLAP. Essa agregação resulta num fenômeno conhecido como explosão dos dados, pois o cubo OLAP pré-calcula cada valor agregado possível quando o cubo é construído e armazena esse valor na estrutura do cubo (RASMUSSEN, 2002).

A maior vantagem dos sistemas MOLAP é que as operações multidimensionais podem ser executadas de uma maneira fácil e natural, sem a necessidade de realizar operações complexas de junção (GOLFARELLI e RIZZI, 2009).

ROLAP é acrônimo para *Relational Online Analytical Processing*. ROLAP armazena as agregações numa estrutura relacional compartilhada com a estrutura relacional pré-existente. Esta arquitetura exige um alto processamento na base de dados relacional onde as tabelas de origem para os cubos de OLAP são armazenadas. Cada consulta emitida é processada pelo RDBMS³ antes de retornar o resultado. Isto parece um processo custoso com considerações sérias de desempenho e levanta o questionamento do por quê o ROLAP seria considerado um mecanismo de persistência (RASMUSSEN, 2002).

Com o ROLAP a explosão dos dados não pode ser realizada, pois a informação não é pré-agregada como no armazenamento MOLAP.

³ Relational Database Management System

Isso é importante quando os dados originais colocados no mecanismo OLAP são grandes.

Para reduzir o número de junções a desnormalização necessária, ou seja, uma violação consciente da terceira forma normal para maximizar o desempenho. Utilizando a redundância podem-se minimizar os custos de execução, como a materialização de algumas tabelas derivadas (*views*) que armazenam os dados agregados usados nas consultas típicas (GOLFARELLI; RIZZI, 2009).

HOLAP é acrônimo para *Hybrid Online Analytical Processing*. HOLAP armazena as agregações numa estrutura multidimensional e deixa a fonte de dados na estrutura relacional existente. No modelo híbrido, os valores agregados (ou pré-calculado) são armazenados na seção MOLAP e as consultas que podem ser respondidas pelos agregados possuem respostas mais rápidas. Quando a consulta solicita informação dos valores agregados e acessa a seção ROLAP, o tempo de resposta é mais lento (RASMUSSEN, 2002).

O uso da estratégia HOLAP implica que a maior quantidade de dados deve ser armazenada em um RDBMS, a fim de evitar problemas causados pela dispersão, enquanto o sistema multidimensional armazena apenas as informações mais frequentes que os usuários precisam acessar (GOLFARELLI; RIZZI, 2009).

HOLAP implies that the largest amount of data should be stored in an RDBMS to avoid the problems caused by sparsity, and that a multidimensional system stores only the information users most frequently need to access.

BENEFÍCIO	MOLAP	ROLAP
Visão multidimensional	X	X
Excelente desempenho	X	
Flexibilidade analítica	X	
Acesso a dados em tempo real		X
Grande capacidade de dados		X
Alavanca o DW		X
Fácil desenvolvimento	X	

Baixa manutenção de estrutura		X
Baixa manutenção de agregados	X	

Quadro 2 Comparação entre MOLAP vs ROLAP

Fonte: Adaptado de Gonzaga (2005).

2.5 UNIVERSO TEXTUAL NA ABORDAGEM OLAP

No tópico anterior apresentou-se a importância da modelagem dimensional e das operações OLAP em cima dos cubos para apoiar o processo de tomada de decisão, porém existe um hiato na tratativa das ferramentas analíticas OLAP sobre os dados não estruturados. Para investigar uma solução que estenda essa visão para o universo das fontes não estruturadas realizou-se uma busca por pesquisas científicas correlatas que trazem o universo textual para as ferramentas de processamento OLAP, de forma a tratar o texto não apenas como um conjunto de caracteres isolados, mas sim extraindo informação desse conteúdo para melhorar o processo decisório. Nesse sentido destacamos alguns trabalhos relevantes que são apresentados abaixo.

A visão mais simples de analisar um cubo contendo dados textuais é modelá-lo como um campo de caracteres no modelo dimensional, permitindo realizar uma consulta de palavra-chave, onde os registros que possuem o texto mais relevante no campo serão retornados como resultado. Por exemplo, a consulta (local = "Florianópolis", palavra-chave = "Praia") buscará os registros com localização igual ao local "Florianópolis" e o texto que contém o termo "Praia". Essa forma apenas amplia a capacidade de consulta de um banco de dados relacional para oferecer suporte à pesquisa sobre um campo de texto. No entanto, essa abordagem não apoia a realização de operações OLAP sobre o universo textual (ZHANG et al., 2009).

Lee et al. (2000) descreve um método para usar as técnicas OLAP em uma coleção de texto utilizando como base meios oriundos da recuperação de informação, como a combinação de busca textual e o ranqueamento, e outros advindos do processamento analítico online como as operações *slice*, *dice* e *drilldown*. A abordagem integra dados estruturados e não estruturados por meio das hierarquias disponibilizadas nos dados estruturados.

Segundo Bhide et al. (2008) para combinar as fontes não estruturadas com as estruturadas há a necessidade de ligar as duas fontes pois não existe uma forma simples de combinar os resultados retornados

individualmente. Nesse sentido promovem o uso de um novo sistema denominado EROCS (*Entity Recognition in Context of Structured data*). Esse sistema permite ligar um documento textual com os dados estruturados armazenados numa base de dados relacional, tratando os dados estruturados como um conjunto de entidades predefinidas e identificando as entidades desse conjunto que melhor correspondem ao determinado documento.

Inokuchi e Takeda (2007) propõe em seus estudos um modelo para representação de dados e o uso de operações algébricas sobre esse modelo para integrar ontologias com sistemas OLAP. O objetivo é analisar uma grande quantidade de documentos textuais. Salientam que a visão de tratar um documento textual como fato e os termos desse documento como valores de dimensão, a tendência é que o número de nós na hierarquia se torne muito grande e há uma relação complexa entre os nós. Segundo Inokuchi e Takeda (2007) não se pode, por meio de uma implementação simples, armazenar os dados num esquema estrela e agregá-los de forma eficiente, haja vista a quantidade de elementos armazenados numa estrutura hierárquica.

Zhang et al. (2009) apresentam o cubo de tópicos (*topic cube*) para combinar OLAP com o modelo de tópicos probabilístico, permitindo realizar as operações OLAP na dimensão de dados textuais por meio de uma hierarquia de tópicos e o armazenamento de medidas probabilísticas oriundas dos documentos textuais. Essa abordagem adota uma árvore de tópicos hierárquica para definir um tópico na dimensão a fim de explorar a informação textual e armazena a distribuição de palavras como conteúdo de medidas da informação textual.

Ding et al. (2010) sugerem a criação de um modelo denominado cubo textual (*text cube*), na qual a célula do cubo textual agrega um conjunto de documentos com valores de atributo oriundos de um subconjunto de dimensões, ou seja, apresentam o mecanismo *TopCells* que busca solucionar o problema de encontrar as células mais relevantes no cubo textual dado uma consulta de palavras-chave. O objetivo é descobrir relações existentes entre as dimensões estruturadas e o dado textual, retornando as células relevantes (agregações das tuplas) que representam certos objetos.

O uso das operações OLAP no universo não estruturado ainda carece de pesquisa. Nesse sentido discorreu-se sobre algumas abordagens que fazem o uso do texto e permitem sua exploração por meio da análise de cubos. Visando a abstração e integração das fontes de informação por meio da semântica apresenta-se o próximo tópico destinado ao uso de ontologias.

2.6 ONTOLOGIA

Uma ontologia é uma especificação formal explícita de uma conceitualização compartilhada (GRUBER, 1993). Uma ontologia está ligada ao estudo das categorias de vocabulários que existem ou podem existir em algum domínio. O produto desse estudo é um conjunto dos tipos que pertencem a um domínio “D” pela perspectiva de uma pessoa que usa a língua “L” com a finalidade de falar sobre “D”. Na ontologia os tipos representam entidades, classes, propriedades, predicados e funções utilizadas na língua “L” para discutir tópicos do domínio “D” (SOWA, 2000, p. 492).

Castoldi (2003, p. 34) relata que uma ontologia facilita o intercâmbio de informações sobre um domínio específico e geralmente, a ontologia de um domínio não tem fim próprio. Desenvolver uma ontologia é como definir uma estrutura e um conjunto de dados que serão usados por outros programas.

Albuquerque (2003) em seu trabalho cita as classificações de ontologias conforme a base do seu conteúdo em genéricas, de domínio, de tarefas, de aplicação e de representação. As ontologias genéricas expressam teorias básicas do mundo, de caráter bastante abstrato, aplicáveis a qualquer domínio, conhecimento de senso comum. As ontologias de domínio expressam conceituações de domínios particulares, descrevendo o vocabulário relacionado a um domínio genérico, tal como medicina e direito. As ontologias de tarefas expressam conceituações sobre a resolução de problemas, independentemente do domínio em que ocorram, isto é, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, tal como, diagnose ou vendas. As ontologias de aplicação expressam conceitos dependentes do domínio e da tarefa particulares. Estes conceitos freqüentemente correspondem a papéis desempenhados por entidades do domínio quando da realização de certa atividade. As ontologias de representação expressam os compromissos ontológicos embutidos em formalismos de representação de conhecimento.

Segundo Guarino (1998) existem quatro tipos de ontologia: “top-level ontology”; “domain ontology”; “task ontology” e “application ontology”. A ontologia “top-level” descreve conceitos genéricos que são independentes de um domínio ou problema particular; enquanto as ontologias de “domain” e “task” descrevem respectivamente o vocabulário relacionado a um domínio genérico ou tarefa/atividade genérica, especializando os termos introduzidos na ontologia “top-level”

e, por último, as ontologias do tipo “application” descrevem conceitos dependentes das ontologias do tipo “domain” e “task”, esse tipo de ontologia é responsável em expressar entidades de um domínio que realizam alguma atividade ou tarefa.

2.6.1 RDF (Resource Description Framework)

O RDF define uma infra-estrutura que permite a codificação, troca e reuso de metadados estruturados. RDF é uma aplicação do XML que impõe restrições estruturais para prover métodos não-ambíguos a fim de expressar a semântica (D-LIB MAGAZINE, 1998).

O fundamento do RDF estabelece um modelo base para descrição de dados que consiste em três tipos de objeto: 1) Recursos: um recurso é qualquer coisa descrita em expressões do RDF e são sempre especificadas por URI⁴; 2) Propriedades: uma propriedade é uma característica, um atributo ou uma relação utilizada para descrever o recurso e 3) Declarações: uma declaração é um recurso específico com uma propriedade definida mais o valor desta propriedade. Essas três partes individuais (triplas) são chamadas respectivamente de sujeito, predicado e objeto (SILVA; LIMA, 2002) e podem ser visualizadas na figura abaixo.



Figura 4 Ilustração mostrando à tripla RDF

Fonte: Adaptado de Silva e Lima (2002).

O RDF Schema (RDFS, 2006) compreende primitivas para a organização de hierarquias e para a definição de restrições sobre RDF. O RDF Schema é um sistema de classes extensível e genérico que pode ser utilizado como base para esquemas de um domínio específico. Esses esquemas podem ser compartilhados e estendidos através de refinamento de subclasses. Além disso, definições de metadados podem ser reutilizadas através do compartilhamento de esquemas

⁴ URI (*Uniform Resource Identifier*) é uma cadeia de caracteres usada para identificar um recurso.

(BRAGANHOLO; HEUSER, 2001). As classes podem estar organizadas em uma hierarquia de subclasses. Isso significa que qualquer recurso de um tipo que é subclasse de outro, é também considerado como sendo do tipo da superclasse.

2.6.2 OIL (Ontology Inference Layer)

A OIL é uma linguagem para a especificação de ontologias que reúne as seguintes características: provê primitivas de modelagem normalmente utilizadas em ontologias baseadas em *frames*; possui uma semântica bem definida, simples e clara baseada na lógica descritiva; e apresenta suporte para dedução automática (HORROCKS et al., 2000).

Uma ontologia OIL contém descrições para classes, relacionamentos (*slots*) e instâncias. Classes podem se relacionar com outras classes através de uma hierarquia (classes/subclasses) e através de relações binárias estabelecidas entre duas relações. Além disso, restrições de cardinalidade podem ser atribuídas aos relacionamentos.

A definição de uma ontologia em OIL é constituída de dois componentes: o primeiro, que descreve as características da ontologia (*ontology container*) utilizando-se de descritores do padrão Dublin Core⁵; e o segundo (*ontology definitions*) que define o vocabulário particular daquela ontologia (HORROCKS et al., 2000).

Uma característica importante dessa linguagem é que a mesma pode ser utilizada com a linguagem XML, muito embora esquemas XML não capturem totalmente a semântica embutida no OIL. Porém, sua integração com RDF/RDFS é bastante promissora. Do mesmo modo que um RDF Schema é utilizado para se auto definir, o mesmo também pode ser utilizado para definir outras linguagens de ontologia. Dessa forma utilizou-se o RDF/RDFS para definir o OIL básico, onde os elementos do seu vocabulário foram mapeamentos para termos do RDF/RDFS (HORROCKS et al., 2000).

⁵ Dublin Core é um esquema de metadados que visa descrever objetos digitais, tais como vídeos, sons, imagens, textos e sites web.

2.6.3 DAML (DARP AGENT MARKUP LANGUAGE)

A linguagem DAML é uma iniciativa da agência DARPA⁶ que está sendo desenvolvida como uma extensão da XML e RDF. A sua iniciativa mais recente é oriunda da combinação de DAML e OIL, uma linguagem padrão para representação de ontologias e metadados pela W3C.

A combinação de DAML e OIL, denominada DAML+OIL, sofre muita influência do OIL original, embora não utilize o seu conceito original de *frames*. A linguagem DAML é constituída de uma coleção de classes, propriedades e objetos que são adicionados ao RDF e RDFS. Assim, declarações (*statements*) em DAML+OIL também são declarações RDF (HENDLER, 2001).

Segundo Hendler (2001) a linguagem DAML irá fornecer uma série de vantagens sobre os métodos de marcação atual. Ela permitirá a interoperabilidade semântica no mesmo nível que temos a interoperabilidade sintática em XML. Os objetos na Web podem ser marcados (manualmente ou automaticamente) para incluir informações e descrições de dados. Isso permitirá que as páginas da web, bases de dados, programas, modelos e todos os sensores sejam ligados entre si, por meio de agentes que usam DAML para reconhecer os conceitos que estão procurando. Se for bem sucedido, a fusão de informações de diversas fontes se tornará uma realidade.

2.6.4 OWL

O grupo da W3C (W3CSEMANTICWEB, 2004) vem desenvolvendo e evoluindo uma série de linguagens para ontologias, possuindo como padrão a linguagem OWL (*Web Ontology Language*). Esta linguagem é influenciada por formalismos estabelecidos, por paradigmas de representação do conhecimento e pela existência de outras linguagens para ontologias e para a Web Semântica (HORROCKS et al., 2003).

A linguagem OWL é uma nova linguagem para ontologias. Porém, esta linguagem respeita a arquitetura da Web Semântica (veja figura abaixo), evoluindo suas linguagens bases: XML (XML, 2004),

⁶ DARPA é uma organização central de P&D do departamento de defesa norte-americano.

- d) indicar equivalência e disjunção entre conceitos e propriedades;
- e) indicar igualdade e diferença entre indivíduos;
- f) outras informações fundamentais para fornecer o suporte semântico necessário para os primeiros passos da Web Semântica.

O uso de ontologias facilita a troca de informações sobre um domínio, nosso objetivo ao propor o uso de descrições semânticas nessa aplicação é permitir intercambiar informação entre os dois universos de fontes de informação: estruturadas e não estruturadas. Outro objetivo do trabalho é permitir o acesso uniforme e transparente sobre as fontes de informação, para isso o próximo tópico descreve as formas de integração de dados e, também, a necessidade de tratar a semântica dos dados no processo de integração.

2.7 INTEGRAÇÃO SEMÂNTICA

A integração de dados envolve formas de combinar dados oriundos de diversas fontes e prover ao usuário uma visão unificada desses dados. A razão para integração de dados é facilitar o acesso e reutilizar informações através de um ponto de acesso à informação e, também, complementar os dados e combiná-los para obter uma base mais ampla para satisfazer uma necessidade (ZIEGLER, DITTRICH, 2004; LENZERINI, 2002).

Segundo Ziegler e Dittrich (2004) as abordagens mais comuns de integração são: **Integração manual**: Usuários interagem diretamente com os sistemas de informação e manualmente integram os dados selecionados, isso é o usuário manipula diversas interfaces e linguagens de consultas, adicionalmente o usuário precisa conhecer a localização, a representação lógica do modelo e a semântica dos dados; **Interface comum de usuário**: Usuário utiliza uma interface padrão que provê uma visão uniforme. Os dados necessários ainda são apresentados de forma separada para a homogeneização e a integração dos dados ainda é realizada pelos usuários; **Integração pelas aplicações**: Nessa abordagem a aplicação acessa vários repositórios e retorna os resultados de forma integrada para o usuário. Essa solução é prática para um pequeno número de componentes no sistema. Entretanto, na medida que a aplicação aumenta o número de interface dos sistemas e de formato de dados para homogeneizar e integrar também cresce; **Integração por middleware**: *Middleware* provê funcionalidades que podem ser reutilizadas, são geralmente usadas para resolver aspectos

dedicados aos problemas de integração. Enquanto a aplicação está isenta de codificar a funcionalidade os esforços de integração ainda são necessários nas aplicações. Além disso, diferentes ferramentas de *middleware* precisam ser combinadas para construir sistemas integrados; **Acesso uniforme aos dados:** Nesse caso a integração lógica dos dados é acoplada no nível de acesso aos dados. Aplicações globais são providas com uma visão global onde os dados estão fisicamente distribuídos, embora apenas os dados virtuais estejam disponíveis a este nível. No entanto, a disponibilização de todos os dados físicos pode ser demorada, pois os dados de acesso, homogeneização e integração têm que ser feito durante a execução; **Armazenamento de dados comum:** A integração de dados físicos é realizada pela transferência de dados para um novo repositório de dados. Em geral, a integração de dados físicos proporciona acesso rápido aos dados, porém se as fontes forem locais, periodicamente a atualização dos dados no repositório global deve ser realizada.

As abordagens iniciais de integração foram baseadas num modelo de dados relacional ou funcional, porém com o advento da internet e das tecnologias web, o foco ampliou incorporando não somente os dados estruturados, mas também os dados não estruturados. Entretanto, integração é mais que um problema estrutural ou técnico é um problema de heterogeneidade semântica (KZAZ et al., 2010).

A integração semântica é a tarefa de agrupar, combinar ou completar dados de diferentes fontes, tendo em vista a semântica explícita dos dados a fim de evitar que os dados semanticamente incompatíveis sejam estruturalmente mesclados, ou seja, a integração semântica tem de garantir que apenas os dados relacionados com a mesma entidade sejam fundidos, para isso é necessário resolver a ambigüidade semântica sobre os dados por meio de metadados (ZIEGLER, DITTRICH, 2004; KZAZ et al., 2010).

Ontologias podem contribuir para resolver o problema de heterogeneidade semântica. Comparada com outros esquemas clássicos como taxonomia, tesouro, palavras-chave, as ontologias são mais completas e precisas ao modelar um domínio (ZIEGLER, DITTRICH, 2004).

2.8 ABORDAGENS DE BUSINESS INTELLIGENCE BASEADOS EM TECNOLOGIAS SEMÂNTICAS

Nos tópicos anteriores observamos a importância da semântica ao manipular a informação. Com base nessa premissa apresentamos três

abordagens que aplicam a semântica no contexto de negócios inteligentes (Business Intelligence).

2.8.1 Semântica de fontes de notícias para aplicações de BI

Yeh e Kass (2008) apresentam em seu artigo uma abordagem baseada em conhecimento para capturar as representações semânticas a partir de fontes de notícias na web para Business Intelligence. Essa abordagem captura a semântica das notícias através da geração de frases contra as notícias, para isso fazem o uso de transformações sintáticas e semânticas. No seu algoritmo a representação que melhor corresponder a uma notícia, é selecionada como o significado dessa notícia.

Yeh e Kass (2008) citam a necessidade de cinco fases para o tratamento de uma notícia, são elas: 1) Substituição léxica: Troca um termo por outro que tenha o mesmo conceito, exemplo: “descansar” por “repousar”; 2) Substituição sintática: Troca um termo por outro que tenha a mesma relação semântica, exemplo: “assunto” por “tema”; 3) Voz ativa para voz passiva: Mudar a relação sintática para uma relação semântica, exemplo: “assuntu” por “assunto”; 4) Modificador de conversão: Um conceito é um atributo de outro se eles estão diretamente relacionados a uma relação semântica; e 5) Especialização de realização: Substituir um conceito para uma de suas subclasses ou instâncias, exemplo: Substituir o termo “mover” para o conceito “caminhar”, uma subclasse de “mover” (YEH; KASS, 2008).

Segundo Yeh e Kass (2008) sua abordagem não trata em conformidade casos de eufemismo e metáfora, além de acontecer algumas manipulações indevidas em casos de negação, porém, ao aplicarem essa abordagem em dois casos reais obtiveram uma precisão em torno de 85,87% de acerto.

2.8.2 PARMENIDES

Mikroyannidis et al. (2006) introduzem um framework orientado ao negócio para gestão da informação na web denominado PARMENIDES. O framework permite que a informação semântica possa ser extraída de dados da web e armazenada de forma que permita, além de uma recuperação eficiente, o raciocínio para apoiar as tarefas relacionadas aos negócios inteligentes (Business Intelligence).

A figura abaixo ilustra a arquitetura adotada em PARMENIDES. A arquitetura consiste de componentes para a construção e armazenamento de documentos (DW). Primeiro, os documentos são

capturados a partir da web e convertidos num esquema de anotação baseado em XML denominado CAS (*Common Annotation Scheme*). O CAS permite uma ampla gama de anotações: estruturais, léxicas e semânticas ou conceituais. Em seguida, um anotador de conceitos extrai a informação semântica temporal dos documentos, mediante a aplicação de técnicas de extração de informação. Durante esse processo o conhecimento de domínio é fornecido pela ontologia de domínio. A análise do documento está organizada em fluxos de trabalho que podem ser totalmente personalizado pelo usuário. O DW contém documentos e metadados com base na semântica extraída no conteúdo dos documentos (MIKROYANNIDIS et al., 2006).

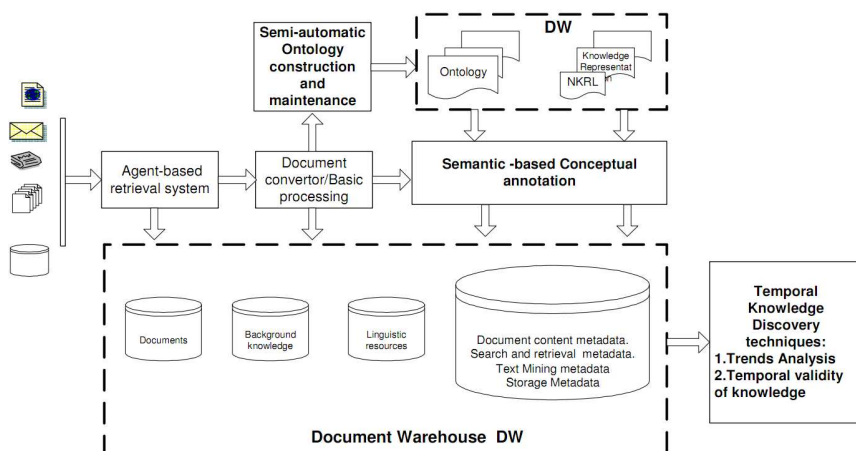


Figura 6 Camadas da arquitetura do framework PARMENIDES
Fonte: Mikroyannidis et al. (2006).

A aplicação do PARMENIDES permite executar consultas e raciocínios sobre os metadados, isso pode auxiliar no acompanhamento evolucionário de uma organização, a fim de oferecer respostas aos gestores para suas tarefas comuns Mikroyannidis et al. (2006).

2.8.3 SBI (Semantic Business Intelligence)

Sell (2006), em sua tese, descreve a arquitetura SBI (*Semantic Business Intelligence*) formada por cinco camadas (veja figura abaixo): Fontes de Dados, Repositório de Ontologias, Mecanismos de Inferência, Módulos Funcionais e Clientes.

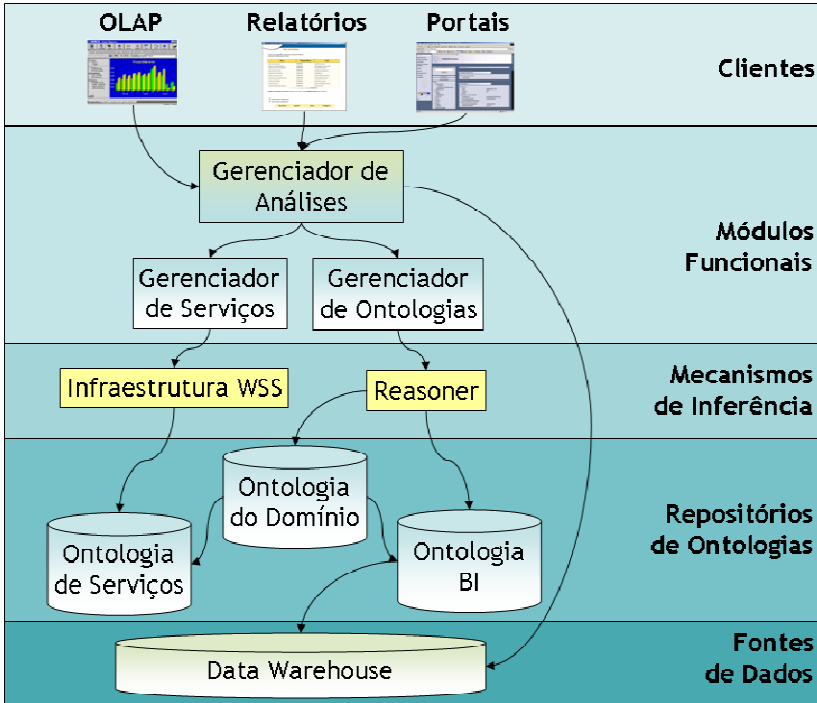


Figura 7 Camadas da arquitetura SBI

Fonte: Sell (2006).

As fontes de dados em sua arquitetura são restritas aos data marts da organização, limitando as análises em repositórios já consolidados. Nessa camada há um hiato sobre o conhecimento organizacional, pois se restringe as fontes estruturadas e já consolidadas. Um dos objetivos desse trabalho é prover uma visão integrada das fontes estruturadas e não estruturadas, permitindo até combiná-las numa mesma análise, sem perder a capacidade de aplicar as operações analíticas online (OLAP).

O repositório de ontologias permite o mapeamento da semântica do negócio, dos dados da organização e dos serviços necessários para o apoio ao processo decisório. Nesse caso Sell (2006) trabalha com três ontologias: ontologia de domínio responsável em modelar a semântica do negócio; ontologia BI que permite a navegação sobre as fontes de dados usando a semântica do negócio e a utilização das regras de negócios para apoiar o processamento analítico; e a ontologia de serviços necessária para descrever semanticamente os serviços web e como localizar serviços de acordo com a análise.

Os mecanismos de inferência permitem a realização de processamento de regras sobre as ontologias e Sell (2006) aproveitou a capacidade de inferência para permitir a realização de cortes semânticos durante a análise. Tal fato propiciou a capacidade de filtrar os dados reunidos na análise ou a expansão do escopo dos dados por meio da definição de regras.

Os módulos funcionais são responsáveis pela articulação entre a integração de serviços e o suporte a inferências às aplicações clientes dessa arquitetura. Nessa camada a arquitetura proposta por Sell (2006) oferece a possibilidade de montar fluxos de processamento de informação capazes de trazer dados adicionais ao contexto da análise por meio de serviços web.

A camada da aplicação cliente representa as ferramentas analíticas que proverão aos tomadores de decisão acesso às fontes de dados para a obtenção das informações desejadas.

2.9 CONSIDERAÇÕES SOBRE O CAPÍTULO

Dentre os três trabalhos supracitados destaca-se o trabalho proposto por Sell (2006) pela similaridade da pesquisa e seu contexto inserido na aplicação proposta por esse trabalho.

O presente trabalho faz o uso da arquitetura SBI e a estende para permitir a as ferramentas analíticas incluir em sua análise a possibilidade de executar suas operações sobre as fontes não estruturadas e estruturadas, além de permitir combiná-las.

Um dos módulos funcionais descritos por Sell (2006) denominado Gerenciador de Análises é estendido para permitir o acesso transparente às fontes estruturadas ou não estruturadas; a execução de operações analíticas sobre essas fontes; e sua combinação por meio de operações matemáticas como união, interseção e diferença.

Nesse trabalho a ontologia de BI proposta por Sell (2006) é representada por duas, a de MAPEAMENTO responsável em mapear as fontes de informação e a ANALÍTICA capaz de disponibilizar conceitos para guiar a execução das operações analíticas. O objetivo dessa separação é permitir estender a ontologia de MAPEAMENTO para possibilitar o mapeamento também das fontes não estruturadas e para atender outros formatos específicos de dados sem alterar a ontologia ANALÍTICA.

Nesse capítulo apresentamos uma revisão teórica dos principais assuntos relacionados com essa pesquisa. A seguir detalhamos a aplicação e por último apresentamos um estudo de caso aplicado no

domínio da gestão em Ciência & Tecnologia para verificar se os objetivos foram atingidos.

3 APLICAÇÃO PROPOSTA

Para atender os objetivos desse trabalho que visa o enriquecimento das análises ao combinar as fontes não estruturadas com as estruturadas; o acesso transparente a essas fontes de informação; e a exploração dessas fontes por meio de conceitos atrelados a um domínio, é apresentada os seguintes componentes da aplicação proposta.

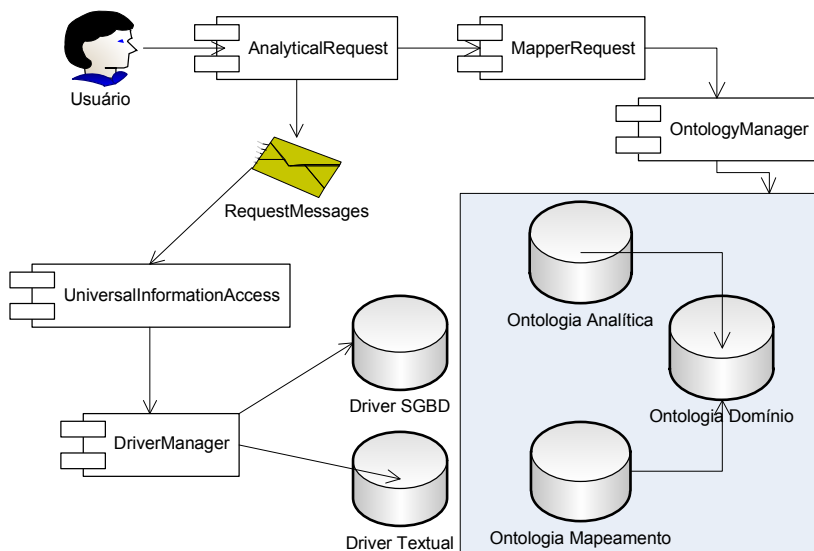


Figura 8 Arquitetura dos componentes da aplicação proposta

A aplicação é formada por cinco componentes: AnalyticalRequest é responsável por requisitar as informações para exploração das fontes de informação; MapperRequest é responsável em traduzir os elementos da ontologia de DOMÍNIO oriundos da requisição para os elementos da ontologia de MAPEAMENTO e complementar a requisição com outras informações como as de junção; OntologyManager é o gestor das três ontologias; UniversalInformationAccess é responsável em receber a mensagem de requisição invocar o DriverManager e, se necessário, realizar a junção entre os retornos; e, por último, o DriverManager responsável em receber uma requisição e instanciar o driver correspondente, solicitando ao driver o seu processamento para então retornar o resultado num

formato padrão. Para um melhor entendimento a aplicação é detalhada nos próximos tópicos.

3.1 APRESENTAÇÃO DA APLICAÇÃO

A aplicação introduz a necessidade de três ontologias, denominadas MAPEAMENTO, ANALÍTICA e DOMÍNIO, sendo esta última variável (conceitos e instâncias) de acordo com o domínio a ser analisado.

Uma API foi criada para possibilitar a recuperação da informação por meio da ontologia de DOMÍNIO. Esta API de requisição também permite aplicar operações analíticas sobre as fontes de informação e, inclusive, combiná-las.

Nessa aplicação a transparência para acesso a informação é obtida utilizando ontologias; padronizando a mensagem de requisição; distribuindo cada mensagem ao seu respectivo processador (*Driver*); e, se necessário, unindo o retorno num formato padrão. A aplicação é detalhada a seguir.

3.2 ONTOLOGIAS

A presente proposta faz o uso de três ontologias denominadas de MAPEAMENTO, necessária para descrever os repositórios, ANALÍTICA responsável em permitir as operações de análise sobre as fontes de informação e de DOMÍNIO, cuja finalidade é permitir uma visão integradora dos diferentes repositórios e atuar como facilitadora na navegação entre os conceitos da organização.

A ontologia de DOMÍNIO é necessária para possibilitar uma visão integrada das diferentes entidades distribuídas nos mais diversos repositórios e servir como entrada para o processo de exploração e análise das fontes de informação. Os conceitos dessa ontologia variam conforme o domínio a ser analisado.

A ontologia de DOMÍNIO faz parte da arquitetura proposta pela aplicação por meio de seu uso, porém essa arquitetura não define conceitos ou instâncias para ontologia de DOMÍNIO, haja vista sua reutilização.

A ontologia de MAPEAMENTO possibilita mapear os metadados dos repositórios, além de permitir representar os mais variados tipos de repositórios. O mapeamento se dá por meio de instâncias de conceitos definidos na ontologia de MAPEAMENTO.

Um vínculo é necessário entre as ontologias, para permitir a navegação entre o conceito descrito na ontologia de DOMÍNIO e os repositórios disponíveis na ontologia de MAPEAMENTO, para tanto a classe InformationSource possui a propriedade hasDomainClass, que armazena o ponteiro para a classe da ontologia de domínio, enquanto a classe Attribute, possui a propriedade hasDomainProperty, com objetivo de referenciar uma propriedade de classe.

CONCEITO	DESCRIÇÃO
Element	A classe Element abstrai qualquer entidade da ontologia de mapeamento. Todas as demais classes dessa ontologia herdam de Element
Function	A classe Function representa a tarefa de associar um valor de saída a cada valor de entrada. Têm-se dois tipos: Aggregate Function e Scalar Function
Aggregate Function	AggregateFunction é um tipo de Function usada para auxiliar a sumarização de grandes volumes de dados
Scalar Function	ScalarFunction é um tipo de Function que retorna um único valor conforme o valor de entrada
Merge	Descreve como uma coleção (Collection) pode ser unida a outra coleção. Identifica quais atributos são utilizados e aponta o método para junção de coleções. Para mapear coleções onde existam relacionamentos de dependência o elemento Merge é descrito de forma recursiva, ou seja, um Merge pode encadear outro Merge e assim sucessivamente
Locale	Locale representa os aspectos relacionados a uma região geográfica, política ou cultural
InformationSource	Abstrai a representação de qualquer elemento das fontes de informação, como por exemplo, uma tabela, visão (view), atributo (banco de dados); tags e propriedades (XML); colunas e linhas de uma planilha ou arquivo csv, etc
Expression	Representa uma combinação de valores, variáveis, operadores e funções que são

	interpretadas (avaliadas) de acordo com regras específicas de precedência, na qual os valores são processados e retornados. Existem duas classes derivadas a UnaryExpression e BinaryExpression
Collection	Representa uma coleção em uma determinada fonte de dados. É derivada de conceitos específicos para representar tabelas de banco de dados, documentos XML, entre outros
Attribute	Representa uma unidade básica de informação a ser utilizada como projeção de um campo ou filtragem. Pode corresponder a um campo de tabela de um banco de dados, um elemento do XML ou uma entidade extraída de textos e outras fontes não estruturadas

Quadro 3 Descrição dos principais conceitos reunidos na ontologia de MAPEAMENTO

A ontologia de MAPEAMENTO permite extensão por meio de subclasses, portanto para cada tipo de coleção/atributo, deve-se criar uma subclasse que mapeie os metadados necessários, visando representar o tipo de repositório em questão, por exemplo, um arquivo do tipo CSV possuiria como atributo o delimitador entre as colunas.

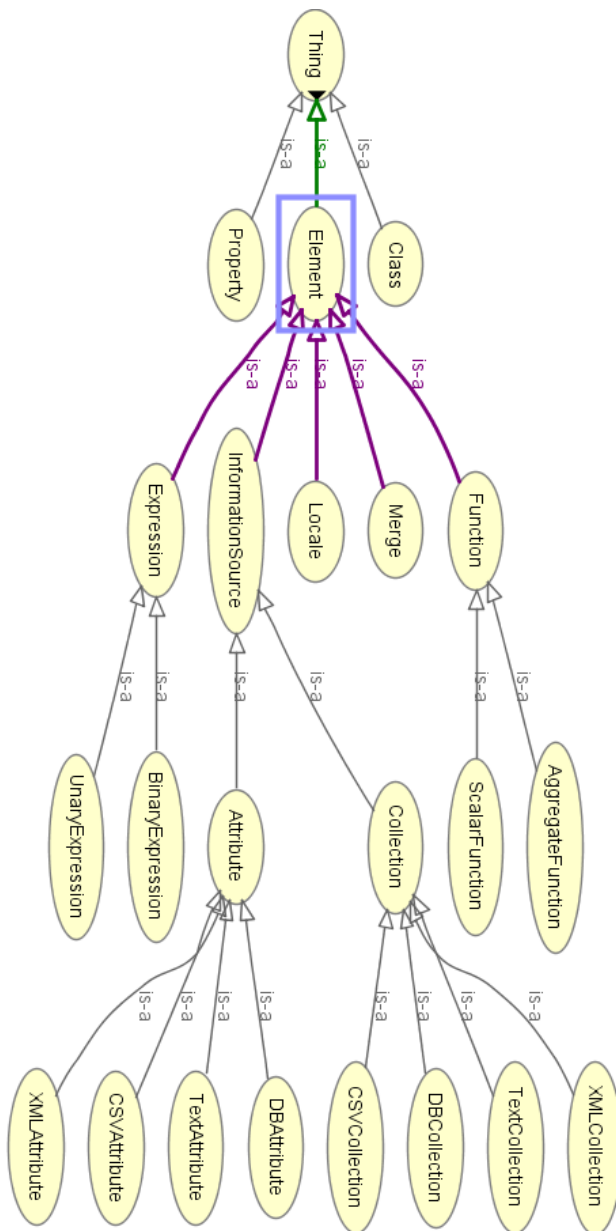


Figura 9 Hierarquia dos conceitos presentes na ontologia de MAPEAMENTO

A ontologia ANALÍTICA descreve conceitos para permitir a realização das operações de análise. Cada instância da ontologia ANALÍTICA possui um ponteiro para um conceito ou instância armazenada na ontologia de DOMÍNIO. Assim como a ontologia de MAPEAMENTO a ontologia ANALÍTICA é preenchida por meio de instâncias de conceitos prédefinidos.

CONCEITO	DESCRIÇÃO
Theme	Classe que representa um tema ou um assunto usado na análise ou na busca de informação. Possui referencia para uma ou mais unidades de tema (ThemeUnit). Ex. um datamart ou um arquivo de índice. Possui a propriedade slices que permite a definição de filtros.
ThemeUnit	Classe que representa uma unidade do tema, que pode ser uma unidade de análise ou ainda, uma unidade de busca. Ex: uma fato de análise ou um campo de busca. Possui a propriedade slices que permite a definição de filtros.
Grouping	Classe que representa um grupo de informações com a finalidade de classificar ou agrupar em categorias. Possui um ou mais GroupingUnits. Ex. As dimensões pertinentes ao fato ou um conjunto de cortes que podem ser aplicados numa busca
GroupingUnit	Representa uma unidade de informação que pode ser agrupada pelos seus valores distintos. Também possui as propriedades hasHierarchicalParent e hasHierarchicalChild responsável em mapear hierarquias. A propriedade hasSortGroupingUnit pode ser configurada para utilizar um campo de ordenação diferente do campo de apresentação (exemplo: num índice textual o campo de pesquisa é diferente de um campo de armazenamento).
Measure	Representa uma unidade de informação associada a uma função de agregação (Aggregate Function), possui referência para detalhar a informação por meio da classe Detail. A Measure pode apontar para inúmeros Detail.

Detail	Representa o detalhamento de uma informação. Um Detail é usado para mostrar em maior nível de detalhe a informação
Slice	Representa uma restrição no conjunto de dados

Quadro 4 Descrição dos principais conceitos reunidos na ontologia ANALÍTICA

A ontologia ANALÍTICA auxilia a ferramenta-cliente em verificar quais medidas, agrupamentos e filtros podem ser aplicados para um conceito ou instância de uma ontologia de DOMÍNIO.

É importante salientar que num cenário onde existam elementos na ontologia de DOMÍNIO sem referência para um elemento na ontologia de MAPEAMENTO não será possível realizar uma requisição de informação. Para evitar essa situação a ferramenta-cliente pode fazer uso da ontologia ANALÍTICA para verificar quais elementos de domínio são cabíveis de análise e suas respectivas operações disponíveis.

Os conceitos definidos na ontologia ANALÍTICA são apenas elementos lógicos para facilitar tanto a organização dos elementos oriundos da ontologia de DOMÍNIO quanto a exploração da informação pela ferramenta-cliente.

A utilização das classes Theme e ThemeUnit também podem ser utilizadas para definir filtros antes de executar uma requisição. Por exemplo, em casos onde existam mais de um conceito da ontologia de DOMÍNIO para uma tabela de dados, pode existir a necessidade de realizar um filtro prévio na tabela. Por exemplo, em uma base de dados está disponível uma tabela denominada “pessoa”, responsável em armazenar dados cadastrais de docentes e discentes. Na ontologia de domínio estão mapeados dois conceitos: “Docente” e “Discente”. Na tabela “pessoa” existe um atributo denominado “bol_docente”, informando para cada tupla se a pessoa é um docente. Ao realizar uma requisição para calcular a quantidade de docentes cadastrados, um filtro prévio é necessário a fim de retornar somente registros onde o valor do atributo “bol_docente” seja verdadeiro. Para mapear esse exemplo podem-se criar duas instancias de Theme, uma apontando para o conceito “Docente” e a outra para “Discente”, sendo que na instância para “Docente” configura-se um slice que deve ser aplicado no atributo “bol_docente” com o valor “true”. Então, a ferramenta-cliente, ao requisitar uma informação sobre “Docente” pode verificar na ontologia ANALÍTICA se existem temas associados a esse elemento da ontologia de DOMÍNIO. Caso exista e a propriedade *slices* esteja populada,

automaticamente as informações de restrições podem ser aplicadas na requisição por solicitação da ferramenta-cliente.

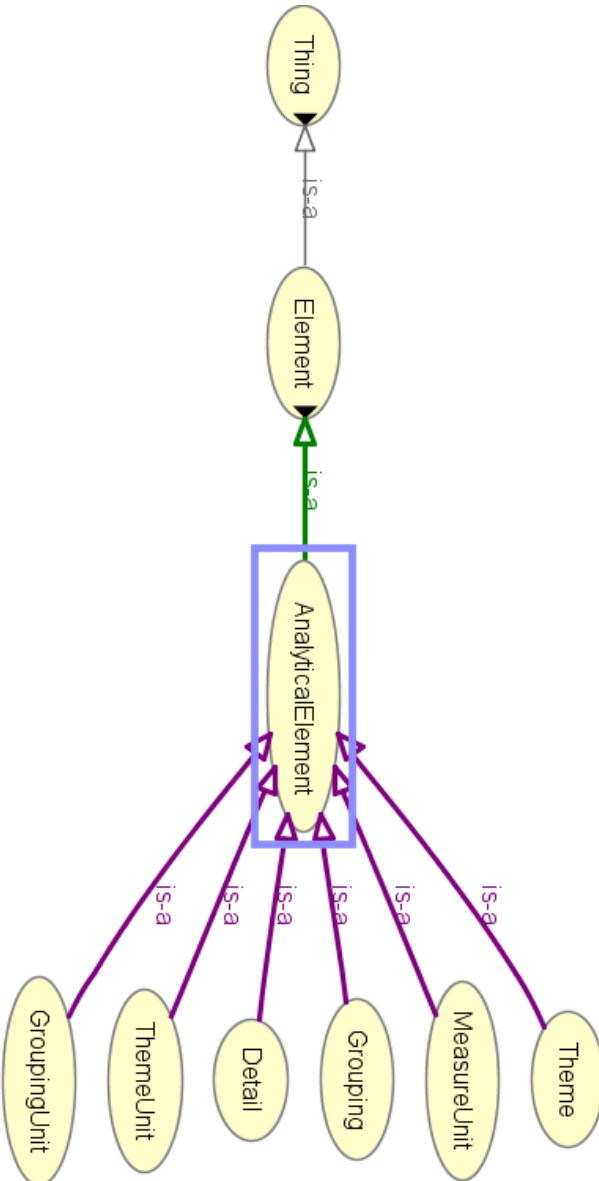


Figura 10 Hierarquia dos conceitos presentes na ontologia ANALÍTICA

3.3 GERENCIADOR DE ONTOLOGIAS

O componente gestor de ontologias (OntologyManager) é responsável em carregar as três ontologias, MAPEAMENTO, ANALÍTICA e de DOMÍNIO, permitindo um acesso central aos elementos e instâncias disponíveis em cada ontologia.

O OntologyManager é uma classe que permite acessar as definições mantidas nas ontologias. Os conceitos representados nas ontologias de MAPEAMENTO E ANALÍTICA são disponibilizados na forma de JavaBeans através de um conjunto de métodos implementados por essa classe.

Já a ontologia de DOMÍNIO, como não se conhece a estrutura a priori, pode-se navegar utilizando apenas estruturas genéricas capazes de mapear classes, propriedades, relações e instâncias.

As definições recuperadas são utilizadas pela ferramenta-cliente para guiar o tomador de decisão na seleção das informações desejadas. O OntologyManager também orienta a recuperação de informações necessárias para a formatação de requisições de dados.

3.4 REQUISIÇÕES DE INFORMAÇÃO

O componente de requisição (AnalyticalRequest) é responsável por traduzir as demandas informacionais em requisições. As ferramentas-cliente invocam métodos para definir quais informações devem ser recuperadas, além de filtros a serem aplicados. A paginação dos dados pode ser solicitada por métodos distintos (setStartPosition, setEndPosition). As operações analíticas podem ser invocadas por meio de métodos padronizados (veja abaixo) como slice, drillDown, drillUp, drillAcross, drillThrough, e pivot.

Na montagem da requisição, o componente complementa as definições da requisição adicionando informações, como por exemplo, quais coleções são envolvidas e como deve ser realizada a junção entre as mesmas. Essas informações são recuperadas da ontologia de MAPEAMENTO e são utilizadas para formar uma mensagem de requisição padrão (RequestMessage).

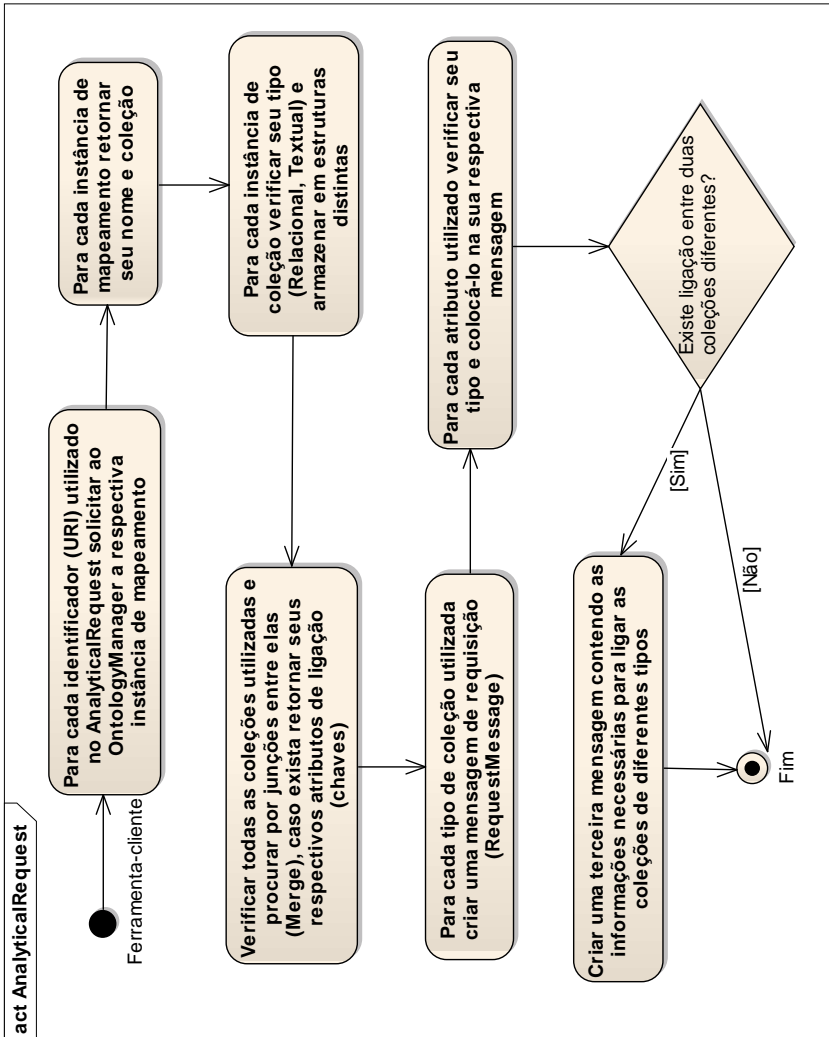


Figura 11 Diagrama explicando a geração das mensagens de requisição (RequestMessage)

Para executar a requisição é necessário utilizar o componente de acesso à informação universal (UniversalInformationAccess), cuja finalidade é disponibilizar a mensagem de requisição ao respectivo Driver.

Caso seja uma requisição envolvendo múltiplos repositórios, esse componente é responsável por fazer as chamadas necessárias e devolver ao solicitante uma estrutura padrão contendo todas as informações solicitadas (Response).

3.5 MENSAGEM DE REQUISIÇÃO (REQUESTMESSAGE)

A troca de informação entre o componente de requisição e o gerenciador de Drivers se dá por meio de uma estrutura padrão capaz de ser representada no formato XML. Essa estrutura armazena informações como o tipo de fonte de dados (relacional ou textual), as coleções e atributos a serem acessados e, também, como as junções entre as coleções devem ser realizadas.

Conforme a requisição pela ferramenta-cliente há a necessidade de armazenar informações adicionais nessa mensagem como o eixo de projeção, a fim de indicar se um determinado atributo deve ser projetado na linha ou na coluna de um cubo.

Optou-se por uma estrutura flexível e extensível, haja vista que as linguagens de consulta OLAP como o MDX⁷ são específicas para acessar um repositório multidimensional.

Como a aplicação proposta estende essa visão, permitindo também o acesso às fontes não estruturadas, podem existir diferenças inerentes ao tipo de fonte a ser acessada. Nesse caso a mensagem de requisição pode ser modificada para incluir atributos e/ou elementos específicos para tratar uma particularidade da fonte de dados.

3.6 GERENCIADOR DE DRIVERS

O processamento das requisições é realizado pelos Drivers, os quais são acionados pelo gerenciador conforme o tipo de requisição enviada pelo componente de requisição de informação.

Os Drivers são responsáveis por processar a mensagem de requisição (*RequestMessage*), gerar uma solicitação de dados na sua linguagem nativa e retornar as informações num formato padrão. Para cada *Driver* codificado é necessário seu registro no gerenciador de Drivers (*DriverManager*).

⁷ Multidimensional Expression (MDX) é uma linguagem para manipular dados multidimensionais.

Cada Driver possui a capacidade de receber os resultados do processamento de outros Drivers, de maneira a permitir a interseção, união e diferença de resultados entre Drivers de diferentes tipos ou pertencentes a repositórios heterogêneos.

Cada Driver deve implementar uma interface padrão, responsável por definir os métodos básicos para troca de mensagens com o gerenciador de Drivers e o componente de acesso universal à informação (*UniversalInformationAccess*).

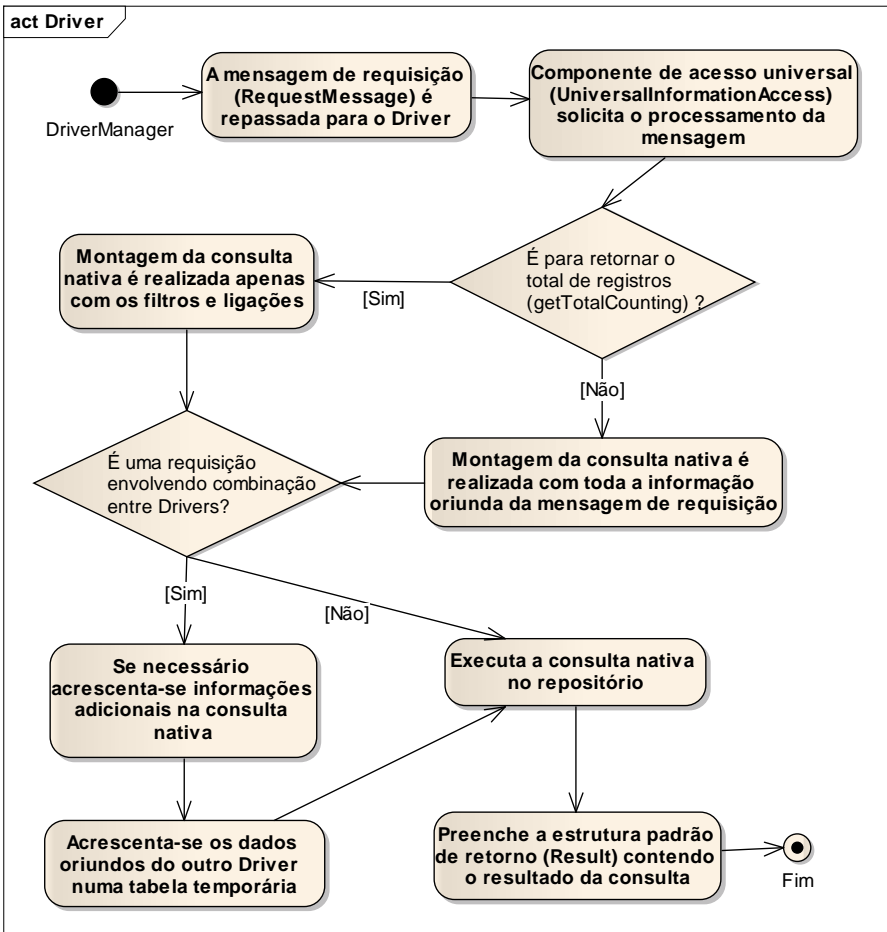


Figura 12 Diagrama explicando os passos executados pelo Driver

Como entrada recebe a mensagem de requisição (*RequestMessage*). Destacamos os seguintes métodos: método principal para execução (*execute*), para realizar a junção entre Drivers (*merge*), outro para retornar o tipo de requisição tratada pelo respectivo Driver (*getType*), o método de retorno (*getResponse*), outro para retornar o número de registros total sem levar em conta a paginação dos dados (*getTotalCounting*) e por fim o método *close*, destinado à liberação de recursos como uma conexão com o banco de dados.

3.7 FLUXOS DE MENSAGENS

A ferramenta-cliente inicializa o componente *AnalyticalRequest* que possui referencia para o componente *OntologyManager*, internamente o *AnalyticalRequest* conversa com o *MapperRequest*.

Nesse momento as ontologias já estão disponíveis para acesso e o usuário pode preencher quais conceitos são do seu interesse e quais filtros devem ser aplicados na requisição.

O *AnalyticalRequest* com auxílio do *OntologyManager* traduz os elementos de domínio oriundos da requisição nos respectivos elementos de mapeamento.

O próximo passo é a invocação do componente *UniversalInformationAccess* a fim de solicitar a execução da requisição, para então retornar as informações oriundas da requisição.

Nesse processo o *AnalyticalRequest* gera um ou mais objetos de mensagem (*RequestMessage*) para atender a requisição de informação. Cada mensagem é repassada para seu respectivo Driver. O Driver processa a mensagem, transformando na linguagem de consulta nativa, por exemplo, o Driver SGBD transforma a requisição no formato SQL, enquanto o Driver Textual traduz numa consulta textual, similar as realizadas nos sites de buscas da Web.

Após o processamento, o Driver gera a resposta num formato padrão. Se necessário o componente *UniversalInformationAccess* possui a capacidade de unir o retorno de mais de um Driver, retornando para a ferramenta-cliente apenas uma resposta com as informações consolidadas. Essa resposta (*Response*) é codificada no formato XML capaz de representar uma estrutura de cubo.

3.8 OPERAÇÕES OLAP

As operações *Drill Down*, *Drill Up* e *Drill Across* estão mapeadas no conceito *GroupingUnit* por meio das respectivas

propriedades `hasHierarchicalChild` e `hasHierarchicalParent`. Assim como o *Slice*, *Dice* e *Filtering* podem ser aplicados diretamente no seu ponteiro para o elemento da ontologia de DOMÍNIO, filtrando um ou mais valores de um atributo. A operação de ordenação *Sort* também pode ser definida sobre um `GroupingUnit` ou `Measure`. As operações *Drill Through* e *Drill Out* são representadas pelo conceito `Detail`. O *Paging* é possível via métodos específicos disponíveis no componente `AnalyticalRequest`.

3.9 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo introduzimos a necessidade de componentes para permitir a requisição transparente da informação, a manipulação de ontologias e a abstração das fontes de informação. Também se apresenta dois mapeamentos semânticos necessários para realizar o mapeamento das fontes de informação e outro responsável em mapear as operações analíticas possíveis sobre os conceitos do domínio.

Com o objetivo de mostrar um cenário que permita aplicar operações analíticas sobre as fontes estruturadas e não estruturadas, a seguir é apresentado um estudo de caso aplicado na Gestão da Ciência & Tecnologia.

4 ESTUDO DE CASO APLICADO NO DOMÍNIO DE GESTÃO DA CIÊNCIA & TECNOLOGIA

Nesse capítulo são apresentados exemplos de uso da aplicação. O objetivo é mostrar o acesso a um repositório com dados estruturados; o acesso a um repositório com dados não estruturados textuais; a combinação entre esses dados de diferentes tipos; e a execução de operações analíticas sobre os mesmos.

No estudo de caso elegeram-se o uso de duas bases: uma representa o contexto da gestão em Ciência & Tecnologia do país denominada Currículo Lattes e outra voltada à análise de cursos de pós-graduação chamada Aplicativo Coleta de Dados CAPES.

Para o experimento foram utilizados apenas os currículos de pessoas que estejam de alguma forma vinculada a Universidade Federal de Santa Catarina (UFSC). O aplicativo Coleta da CAPES foi preenchido com dados do programa de Pós-Graduação em Engenharia e Gestão do Conhecimento (EGC)⁸ também vinculado a UFSC.

Representando os dados não estruturados textuais, utilizaram-se informações textuais incluídas pelo autor de forma livre, como a descrição de cada produção e seu *resumé*. Os campos “grande área de atuação” e “área de atuação” também são oriundos do Currículo Lattes, porém representam dados estruturados. O atributo “nome da pessoa” é utilizado como junção entre as duas fontes.

4.1 CURRÍCULO LATTES

A Plataforma Lattes representa a experiência do CNPq⁹ na integração de bases de dados de Currículos, de Grupos de pesquisa e de Instituições em um único Sistema de Informações. Sua dimensão atual se estende não só às ações de planejamento, gestão e operacionalização do fomento do CNPq, mas também de outras agências de fomento federais e estaduais, das fundações estaduais de apoio à ciência e tecnologia, das instituições de ensino superior e dos institutos de pesquisa. Além disso, se tornou estratégica não só para as atividades de planejamento e gestão, mas também para a formulação das políticas do

⁸ O EGC tem como objetivo a pesquisa de novos modelos, métodos e técnicas de engenharia, de gestão e de mídias do conhecimento, para as organizações e para a sociedade em geral.

⁹ CNPq é o Conselho Nacional de Desenvolvimento Científico e Tecnológico

Ministério de Ciência e Tecnologia e de outros órgãos governamentais da área de ciência, tecnologia e inovação (CNPQ, 2011).

A figura abaixo apresenta os principais módulos de preenchimento do Currículo Lattes: Identificação, Endereço, Formação, Atuação, Produção e Outros.

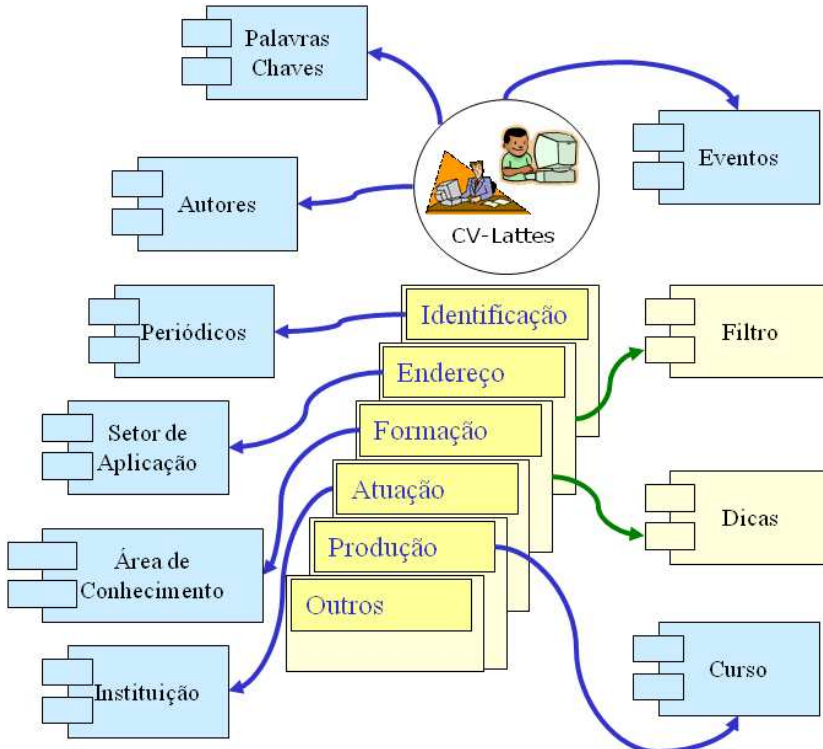


Figura 13 Principais módulos de preenchimento do Currículo Lattes

O Currículo Lattes se tornou um padrão nacional no registro da vida pregressa e atual dos estudantes e pesquisadores do país, e é hoje adotado pela maioria das instituições de fomento, universidades e institutos de pesquisa do País. Por sua riqueza de informações e sua crescente confiabilidade e abrangência, se tornou elemento indispensável e compulsório à análise de mérito e competência dos pleitos de financiamentos na área de ciência e tecnologia (CNPQ, 2011).

A disponibilização pública dos dados na internet dá maior transparência e mais confiabilidade às atividades de fomento do CNPq e

das agências que a utilizam, fortalecem o intercâmbio entre pesquisadores e instituições e é fonte inesgotável de informações para estudos e pesquisas. Na medida em que suas informações são recorrentes e cumulativas, têm também o importante papel de preservar a memória da atividade de pesquisa no país (CNPQ, 2011).

4.2 APLICATIVO COLETA DE DADOS DA CAPES

A aplicação Coleta de Dados é um sistema informatizado da Fundação CAPES, desenvolvido com objetivo de coletar informações dos programas de pós-graduação *stricto sensu* do país. Os dados coletados prestam-se principalmente à avaliação dos programas de pós-graduação e para constituição da chamada “memória da pós-graduação”, que é o acervo de informações consolidadas sobre o Sistema Nacional de Pós-Graduação - SNPG. A coleta de dados objetiva ainda prover a CAPES, informações necessárias ao planejamento dos seus programas de fomento e delineamento de suas políticas institucionais (CAPES, 2009).

Segundo a CAPES (2009) as informações relativas às atividades e à produção de cada programa de pós-graduação são agrupadas em documentos temáticos, denominados: Programas, Disciplinas, Cursos, Turmas, Cadastros, Trabalhos de Conclusão, Proposta do Programa, Produção Intelectual, Linhas de Pesquisa, Fluxo Discente e Projetos.

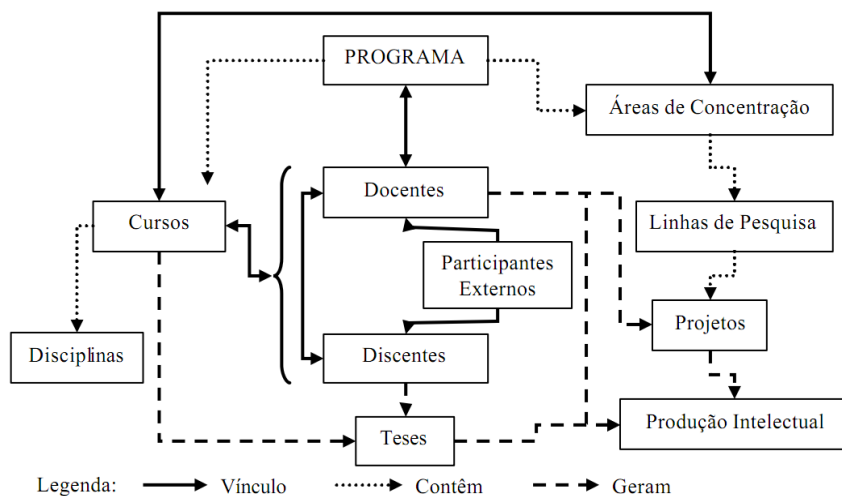


Figura 14 Documentos temáticos disponíveis no aplicativo Coleta de Dados

Fonte: CAPES (2009)

Não obstante a abrangência da coleta de dados, não é intenção atribuir-lhe um caráter censitário. As informações são classificadas segundo um modelo necessariamente genérico. Ele é adequado, porém, à grande variedade de situações existentes e à terminologia empregada no SNPG. Eventualmente, as especificidades de um programa ou IES deverão ser traduzidas para os termos da classificação empregada (CAPES, 2009).

4.3 PREPARAÇÃO DO AMBIENTE

Os currículos da UFSC foram exportados no formato textual XML, logo após processados e indexados com o uso da biblioteca Lucene¹⁰, enquanto a base de dados relacional (HSQLDB¹¹) utilizada pelo aplicativo Coleta foi acessada diretamente.

Desenvolveram-se dois Drivers de acesso às fontes de informação, o primeiro para acessar repositórios relacionais, representando as fontes estruturadas, e outro para acessar índices textuais, representando as fontes não estruturadas. O Driver relacional utiliza a API JDBC¹² disponibilizado pela tecnologia Java para acessar diferentes tipos de banco de dados, enquanto o Driver Textual faz uso do Lucene, ambos de código aberto e codificados em Java.

As ontologias foram criadas com o software Protégé¹³, de código aberto, que permite a modelagem de ontologias. Após esse processo, as ontologias, foram exportadas para o formato OWL em arquivo. Para manipulação desse arquivo, utilizamos o framework JENA¹⁴ de código aberto, o qual disponibiliza APIs para manipulação de RDF e OWL, além de contar com um mecanismo de suporte a consultas no formato SPARQL. Esse framework possibilitou o componente OntologyManager manipular o arquivo anteriormente exportado.

No estudo de caso são apresentados exemplos de acesso às fontes de dados demonstrando as operações mais comuns de análise. Nos exemplos são trabalhados os conceitos Pesquisador e Produção, descritos na ontologia de DOMÍNIO. Os dados que representam as

¹⁰ Lucene para mais informações acesse o endereço <http://lucene.apache.org/>

¹¹ HSQLDB (HyperSQL DataBase) é um banco de dados relacional escrito em Java

¹² JDBC é um acrônimo para *Java Database Connectivity*

¹³ Protégé para mais informações acesse o endereço <http://protege.stanford.edu/>

¹⁴ Jena para mais informações acesse o endereço <http://jena.sourceforge.net/>

instâncias dessa entidade sejam estruturados ou não permanecem no repositório. A ontologia de DOMÍNIO representa o conceito Pesquisador, que publicou nenhuma ou “n” Produção. As ontologias de MAPEAMENTO e ANALÍTICA foram devidamente preenchidas para demonstrar os exemplos.

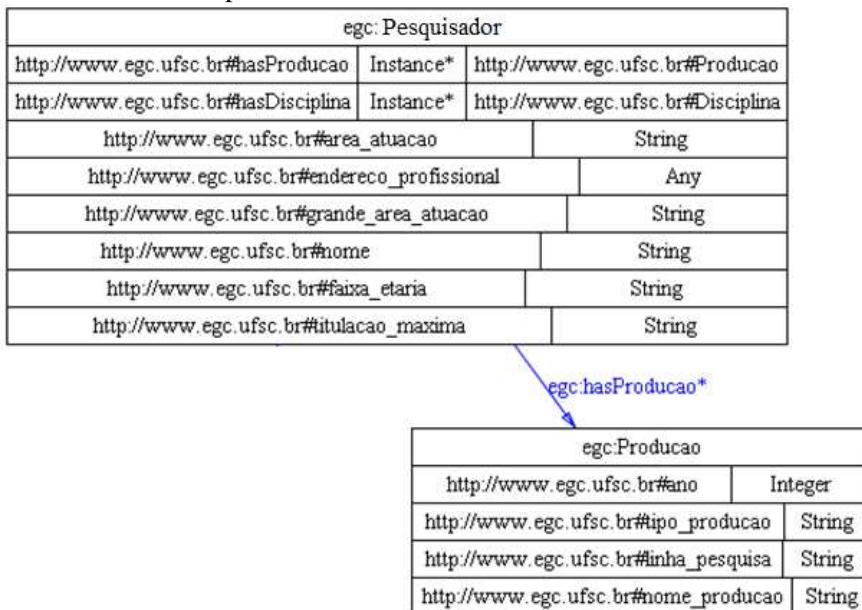


Figura 15 Visualização dos conceitos e propriedades da ontologia de DOMÍNIO

No mapeamento dos elementos oriundos da base relacional temos duas classes principais DBCollection e DBAttribute representando respectivamente as tabelas e os atributos de cada tabela, veja abaixo uma figura representando as instancias DBCollection.

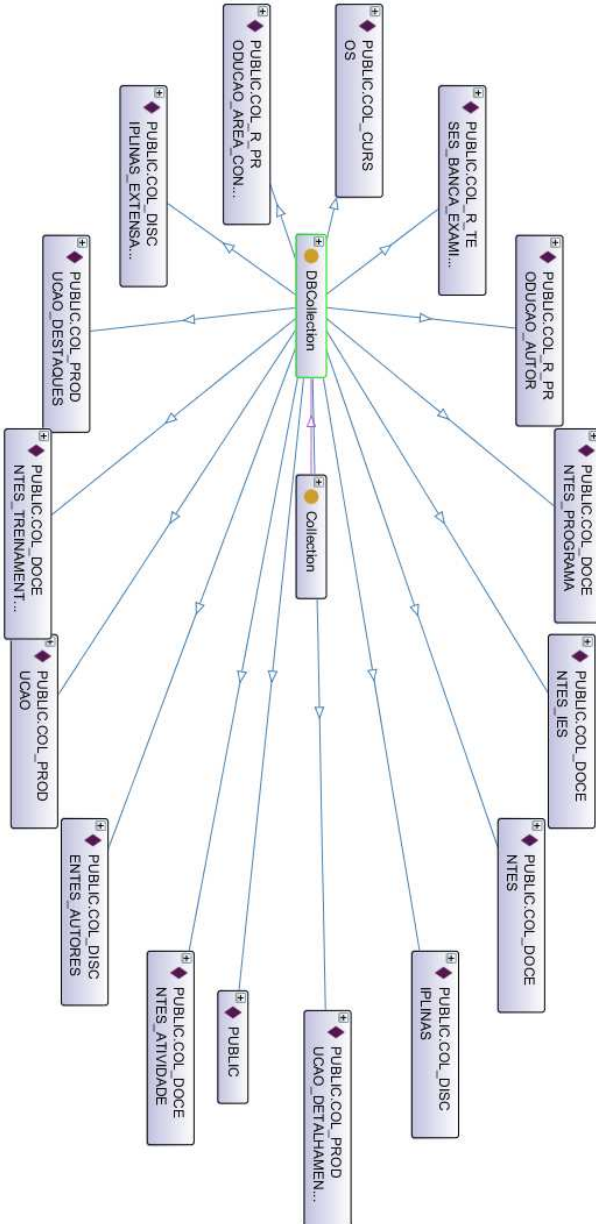


Figura 16 Visualização de instâncias da classe `DBCollection` presentes na ontologia de MAPEAMENTO

No mapeamento dos elementos oriundos da base textual temos duas classes principais TextCollection e TextAttribute representando respectivamente um índice e os atributos disponíveis nesse índice, veja abaixo uma figura representando as instancias do tipo TextCollection e TextAttribute.

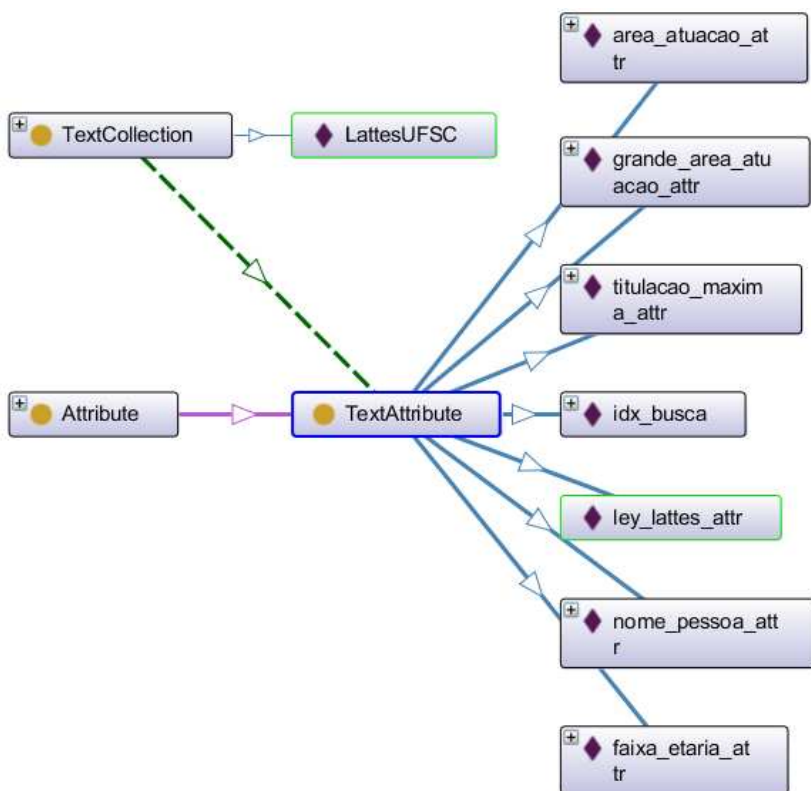


Figura 17 Visualização de instâncias da classe TextAttribute presentes na ontologia de MAPEAMENTO

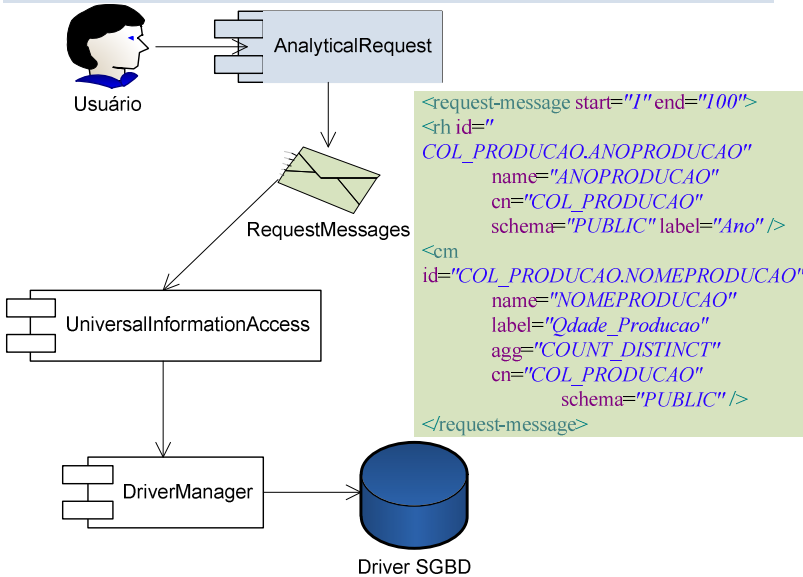
4.4 ACESSO TRANSPARENTE ÀS FONTES DE INFORMAÇÃO

Um dos objetivos da aplicação é permitir o acesso transparente às fontes de informação, para isso, a classe AnalyticalRequest abstrai as informações de quais repositórios devem ser acessados e como será realizada a ligação entre eles.

A ferramenta-cliente apenas solicita os dados sem se preocupar em quais bases serão acessadas para retornar a informação desejada. Conforme cada solicitação de dado, seja um campo de projeção ou filtro, as informações de quais bases, coleções (tabelas) e junções são automaticamente descobertas por meio de inferências sobre a ontologia de MAPEAMENTO.

Essa característica permite facilitar a montagem da requisição pela ferramenta-cliente, haja vista que a ferramenta-cliente não necessita conhecer o metadados das bases e suas ligações.

```
IAAnalyticalRequest request = new AnalyticalRequest(ontologyManager);
request.setStartPosition(1);
request.setEndPosition(100);
request.addGroupingUnit(new URI(ns+"#ano"), "Ano", AxisType.ROW_AXIS);
request.addMeasure(new URI(ns+"#nome_producao"), Aggregator.COUNT_DISTINCT,
"Qdade_Producao", AxisType.COLUMN_AXIS);
```



```
SELECT t0.ANOPRODUCAO AS "Ano",
COUNT(DISTINCT t0.NOMEPRODUCAO) AS "Qdade_Producao"
FROM PUBLIC.COL_PRODUCAO t0
GROUP BY t0.ANOPRODUCAO
```

Figura 18 Fluxo representando o acesso aos dados estruturados

Nesse exemplo o objetivo é realizar uma consulta utilizando apenas dados oriundos da base relacional, ou seja, utilizado exclusivamente um repositório estruturado. Desejamos verificar a quantidade de produções agrupada pelo seu ano de publicação.

O código desenvolvido para responder a questão está representado na figura acima. As URIs representam identificadores oriundos da ontologia de DOMÍNIO.

Ao invocar o método doRequest da interface AnalyticalRequest o componente retorna a mensagem padrão para invocação dos dados.

A mensagem é repassada para o *driver* responsável em manipular requisições para bases relacionais. O mesmo transforma a mensagem padrão na requisição nativa, nesse caso, no formato SQL (*Structured Query Language*).

Nesse momento é realizada uma conexão com a base de dados, executado o comando propriamente dito e o retorno é gerado seguindo um formato padrão capaz de representar um cubo de dados.

Após o retorno gerado apresentamos um componente visual capaz de interpretar esse retorno e renderizá-lo numa tabela multidimensional OLAP, o objetivo é simular uma ferramenta-cliente consumindo a informação solicitada.

Ano	Qdade_Producao
2005	284
2006	297
2007	266
2008	309
2009	477
Total	1633

Figura 19 Visualização de uma tabela multidimensional com dados de uma base relacional

4.5 REALIZANDO UMA CONSULTA COMBINANDO DADOS DE AMBAS AS BASES

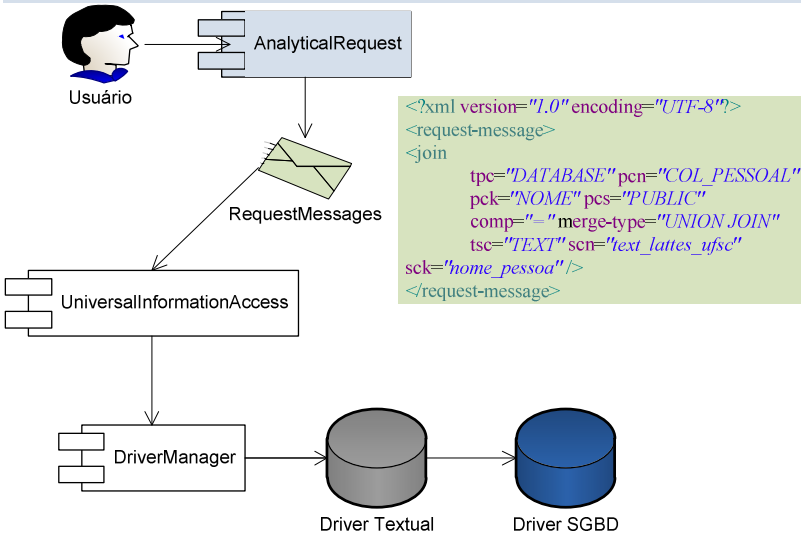
A combinação entre as fontes de informação é outro objetivo do trabalho. Permitir aplicar operações de conjuntos como união,

intersecção e a diferença são formas de combinar as fontes de informação e retornar uma visão consolidada.

```

IUniversalInformationAccess uia = UIAFactory.create(om);
IAAnalyticalRequest request = new AnalyticalRequest(ontologyManager);
request.addGroupingUnit(new URI(ns+"#nome"), AxisType.ROW_AXIS);
request.addGroupingUnit(new URI(ns+"#grande_area_atuacao"), AxisType.COLUMN_AXIS);
request.addMeasure(new URI(ns+"#nome_producao"),
Aggregator.COUNT_DISTINCT, "Qdade_Producao", AxisType.COLUMN_AXIS);
request.slice(new URI(ns+"#Producao"), RelationalOperator.SEARCH, new Object[] { "inovacao" } );
byte[] response = uia.doRequest(rms, CombineType.INTERSECT);

```



```

SELECT t0.NOME AS "iso-PUBLIC.COL_PESSOAL.NOME",
COUNT(DISTINCT t1.NOMEPRODUCAO) AS "Qdade_Producao"
FROM PUBLIC.COL_R_PRODUCAO_AUTOR t2
INNER JOIN PUBLIC.COL_PRODUCAO t1 ON ((t2.ANOBASE = t1.ANOBASE)
AND ((t2.IDPROGRAMA = t1.IDPROGRAMA)
AND (t2.IDPRODUCAO = t1.SEQUENCIAL)))
INNER JOIN PUBLIC.COL_PESSOAL t0 ON ((t2.ANOBASE = t0.ANOBASE)
AND ((t2.IDPROGRAMA = t0.IDPROGRAMA)
AND (t2.IDAUTOR = t0.SEQUENCIAL)))
WHERE (EXISTS (SELECT t3.nome_pessoa FROM PUBLIC.TMP921332904215683845 t3
WHERE (t0.NOME = t3.nome_pessoa) ))
GROUP BY t0.NOME

```

Figura 20 Fluxo simplificado representando o combinação entre fontes de dados estruturados e não estruturados textuais

Nesse exemplo o objetivo é retornar a quantidade de produção bibliográfica dos pesquisadores agrupada pelo seu respectivo nome e pela grande área de atuação, que produziram algo que contenha o termo

“inovação”. Os dados sobre “nome do pesquisador” estão na base relacional e textual, enquanto “grande área de atuação” e a pesquisa por “inovação” é realizada na base textual.

A mensagem de requisição textual solicita os atributos “nome_pessoa” e “grande_area_atuacao” do documento. O índice deve ser filtrado pelas produções nos últimos dez anos que contenham a palavra “inovacao”.

A mensagem de requisição relacional solicita trazer os atributos “nome” e “nomeproducao”, sendo que o atributo “nomeproducao” deve ser contabilizado de forma distinta, ou seja, se possuírem produções que contenham o mesmo nome deve ser contabilizada uma vez. Nesse caso como o banco de dados já implementa operações de agregação utilizou-se o próprio formalismo do SQL.

As informações de “nome” e “nomeproducao” estão em tabelas distintas e, para fazer as junções entre elas é preciso relacionar outras tabelas como a “COL_PESSOAL” e “COL_R_PRODUCAO_AUTOR”. Na mensagem de requisição podemos observar que as junções foram inferidas de forma automática, essas informações de como as coleções se ligam estão representadas na ontologia de MAPEAMENTO.

Para combinar as bases é necessário um campo de ligação para realizar a união, junção ou intersecção, evitando que o retorno seja um produto cartesiano pela combinação dos dois conjuntos. Nesse caso como se deseja apresentar resultados presentes em ambos os repositórios, utilizou-se a intersecção.

Quando há a necessidade de utilizarmos *drivers* de diferentes tipos para atender uma requisição, o componente UniversalInformationAccess prioriza a requisição textual, pois a aplicação dos filtros num índice invertido é de forma extremamente rápida.

Após o retorno do *driver* textual, é inicializado o *driver* relacional, com a informação da requisição de junção. A abordagem para junção no *driver* relacional com outros *drivers* é a seguinte: 1) cria-se em tempo de execução uma tabela temporária na base de dados contendo os registros oriundos do *driver* de origem, nesse caso textual; 2) o *driver* relacional acrescenta em sua consulta restrições conforme o tipo de união. No exemplo de intersecção utilizou-se o comando “EXISTS”; 3) O comando SQL é executado e 4) No final do processamento do *driver* a tabela temporária é removida da base, mantendo sua estrutura original.

iso-PUBLIC_COOL_PESSOALNOME	[.] Qdade_Producao	Clências Biológicas	Clências Exatas e da Terra	Clências Hum	Clências Sociais Aplic	Clências Engenharia
Robert Carlos dos Santos Pacheco	62,00					62,00
Lisiane Geisler	1,00					1,00
Silvestre Labiak Junior	3,00					
Raimundo Nonato Macedo dos San	1,00					3,00
Aline França de Abreu	29,00					1,00
Eliza Coral	1,00					1,00
Oscar Dalfovo	27,00					
Charles Anderson Prada	3,00					
Fabiano Duarte Beppler	5,00					5,00
Louise de Lira Roedel Botelho	10,00					10,00
Cristiane da Silva Santos Villala	1,00					1,00
Maurício Fernandes Pereira	2,00					2,00
Renato Balancieri	5,00					5,00
Dorzell Salete Tzedlak	5,00					5,00
Valmir Emil Hoffmann	2,00					2,00
Cristiano Jose Castro de Almeida C	33,00					33,00

Subtotal

Base Relacional

Base Textual

Subtotal

Clências Biológicas

Clências Exatas e da Terra

Clências Hum

Clências Sociais Aplic

Clências Engenharia

Ambas as bases (campo utilizado para junção)

Figura 21 Visualização do retorno de uma consulta combinando dados de ambas as bases

Após a execução da consulta textual e da consulta relacional, é realizado o processo para complementar as informações de retorno, esse processo só é executado se houverem campos projetados no *driver* textual. Nesse momento é gerado o pacote final de dados com todos os atributos de projeção pertencentes aos dois *drivers*. Para esse exemplo a tabela multidimensional é representada na figura acima.

4.6 APLICANDO OPERAÇÕES OLAP

Permitir realizar operações analíticas, como *drillDown*, *drillUp*, *slice*, entre outras e, também, agregar informações através de operadores como sumarização, contagem e média são premissas desse trabalho. Para demonstrar a execução da operação OLAP denominada *drillDown*, reponsável em detalhar os dados, utilizou-se o exemplo anterior solicitando a API de requisição a execução dos métodos *drillDown* e *sort*, em seguida realizou-se uma nova requisição.

```
IUniversalInformationAccess uia = UIAFactory.create(om);
IA analyticalRequest = new AnalyticalRequest(ontologyManager);
request.addGroupingUnit(new URI(ns+"#nome "), AxisType.ROW_AXIS);
request.addGroupingUnit(new URI(ns+"#grande_area_atuacao"),
AxisType.COLUMN_AXIS);
request.addMeasure(new URI(ns+"#nome_producao"),
Aggregator.COUNT_DISTINCT, "Qdade_Producao",
AxisType.COLUMN_AXIS);
request.slice(new URI(ns+"#Producao"), RelationalOperator.SEARCH, new
Object[] { "inovacao" } );
request.drillDown(new URI(ns+"#grande_area_atuacao"),
AxisType.ROW_AXIS);
request.sort(new URI(ns+"#nome "), Order.ASC);
byte[] response = uia.doRequest(rms, CombineType.INTERSECT);
```

Figura 22 Requisição aplicando as operações *drillDown* e *sort* sobre a combinação de dados estruturados e não estruturados textuais

Nesse exemplo o *drillDown* faz referência a mesma URI utilizada como projeção na coluna denominada “*grande_area_atuacao*”, porém o componente *AnalyticalRequest* nesse momento verifica na ontologia ANALÍTICA se existe uma instância de *GroupingUnit* que faz referência para essa propriedade da ontologia de DOMÍNIO. Por meio dos métodos *hasHierarchicalParent* e *hasHierarchicalChild* disponíveis

no GroupUnit é possível realizar as operações drillUp e drillDown respectivamente, então configurou-se na ontologia ANALÍTICA um GroupingUnit que referencia a instância da propriedade “grande_area_atuacao” oriundo da ontologia de DOMÍNIO, e na sua propriedade hasHierarchicalChild, criou-se uma nova instância de GroupingUnit porém apontando para a propriedade “area_atuacao” da ontologia de DOMÍNIO.

As mensagens de requisição foram modificadas ao executar a requisição para os drivers, porém a requisição de junção continuou equivalente. A mensagem para o driver textual foi acrescida de mais um campo de projeção relativo à “area_atuacao”, respeitando a mesma posição de coluna do seu campo pai “grande_area_atuacao” e a informação para ordenação (sort) do campo “nome_pessoa”.

A mensagem para o *driver* relacional foi acrescida apenas da informação para ordenação (sort) do campo “nome_pessoa”.

Escolheu-se colocar o elemento do drillDown na posição da linha para facilitar a visualização do resultado, a seguir segue o resultado projetado na tabela multidimensional.

epc-area_atuacao_atr	iso-PUBLIC.COL_PESSOAL.NOME	[-] Qdade_Producao						
		SubTotal	Ciências Bio	Ciências Exatas	Ciências Humanas	Ciências Sociais Aplicadas	Ciências da Saúde	Engenharias
	Valmir Emil Hoffmann	2					2	
	Vera Luci de Almeida	7					7	
[-] Arquitetura e Urbanis		3					3	
	Luiz Roberto Mayr	3					3	
[-] Ciência da Computaç		141		90			51	
	Alexandre Leopoldo Gonçalves	12		12				
	Aline França de Abreu	29		29				
	Aran Bey Tcholakian Morales	9						9
	Cátia dos Reis Machado	2		2				
	Danilo Pereira	7		7				
	Fábio Alexandrini	1		1				
	José Leomar Todesco	42						42
	Luiz Alberto Schmitz	3		3				
	Oscar Dalfovo	27		27				
	Pedro Felipe de Abreu	5		5				
	Pedro Sidnei Zanchett	3		3				
	Tatiana Sousa Gonzaga	1		1				
[-] Ciência da Informaçã		2					2	
	Luiz Fernando Gonçalves de Figuei	1						1
	Raimundo Nonato Macedo dos San	1						1
[-] Comunicação		13					13	

Figura 23 Visualização do retorno de uma consulta aplicando as operações drillDown e sort

4.7 CONSIDERAÇÕES SOBRE O CAPÍTULO

Apesar do suporte na arquitetura para registrar múltiplos *drivers* a fim de atender diferentes tipos de fontes de informação, apenas dois *drivers* foram codificados para demonstrar a viabilidade da aplicação proposta.

Quando uma requisição envolve o uso de múltiplos *drivers* poderia ser controlada a ordem do sequenciamento, pois na codificação atual, se existir uma necessidade de utilizar o *driver* textual, o mesmo terá privilégio sobre os demais.

Na estratégia de junção entre *drivers* optou-se em utilizar uma tabela temporária como facilitadora do processo, em certas situações que manipulam poucos registros, pode ser interessante realizar todo o processo em memória, então a forma de junção poderia ser configurável.

No estudo de caso aplicado à gestão de Ciência & Tecnologia apresentaram-se exemplos de requisições que fazem o uso de uma base de dados relacional e de um índice textual e, também, requisições que combinam informações entre ambas as fontes objetivando a execução de operações analíticas.

Nesse estudo de caso utilizou-se um computador com processador intel® modelo I5 2500K com 4GB de memória RAM, porém apenas 512MB estava disponível para a máquina virtual JAVA. O disco rígido armazenava 120GB de dados e possuía a velocidade de 7200 rpm.

Ao executar os exemplos em requisições na qual envolve apenas um *driver* a sobrecarga (*overhead*) foi de até três milissegundos, enquanto requisições que envolviam os dois *drivers* variavam entre três a quatro milissegundos. Em ambos os casos não foi considerado o tempo de carga das ontologias, haja vista que podem ser lidas apenas no início da aplicação e disponibilizadas em memória devido ao seu baixo tamanho (1,3 MB).

5 CONCLUSÃO

Tecnologias semânticas oferecem o potencial para a criação de novas alternativas de exploração das fontes de dados para a formação de conhecimento útil. A aplicação proposta incorpora várias características que permitem a visão integrada de todas as fontes de informação disponíveis na organização. Esta aplicação visa realizar operações analíticas alinhadas a um domínio e acessar repositórios com fontes de dados estruturados e/ou fontes de dados não estruturados do tipo textual.

A aplicação demonstrou, por meio de exemplos, ser uma alternativa viável para a construção de soluções integradas mais flexíveis e alinhadas à lógica do negócio. O uso das três ontologias permite esta aplicação ser aplicada em diferentes domínios, possibilitando as ferramentas analíticas serem configuradas e não reescritas para cada domínio enfrentado. Resumidamente, as seguintes características foram disponibilizadas pela aplicação apresentada: a informação pode ser apresentada e explorada pelos usuários utilizando conceitos de domínio; a realização de operações analíticas sobre dados estruturados e não estruturados textuais e a combinação em uma mesma análise de dados estruturados e não estruturados textuais.

O uso das ontologias de análise e de mapeamento permite a camada visual ser configurada de acordo com o domínio a ser analisado, sem a necessidade de recodificar toda a aplicação para cada domínio enfrentado, possibilitando criar ferramentas de análise genéricas.

A realização de operações analíticas sobre dados não estruturados é um dos propósitos desse trabalho. Tal fato pôde ser observado no estudo de caso, na qual se verifica a aplicação da operação analítica denominada *slice* (corte) sobre os dados não estruturados textuais, permitindo assim restringir o retorno da informação.

Ao analisar os tempos no estudo de caso observa-se uma sobrecarga (*overhead*) baixa, em torno de três a quatro milissegundos por requisição. Ao inferir essa informação conclui-se que o uso da solução proposta é passível de adoção em aplicações comerciais.

A utilização de uma ontologia de domínio como modelo de consulta permite uma análise mais intuitiva para o usuário, pois a navegação se dá por meio de conceitos mais lógicos ao domínio do usuário.

5.1 CONTRIBUIÇÕES

A principal contribuição da aplicação é o enriquecimento das análises possibilitando uma melhor compreensão do contexto explorado, ao complementá-lo com informações oriundas das fontes não estruturadas. Também podemos destacar a facilidade do usuário ao requisitar informações de forma transparente por meio de um modelo de consulta baseado nos conceitos de um domínio.

A aplicação pode apoiar o número crescente de aplicações que são facilitadoras da gestão do conhecimento como localização de competências, inteligência competitiva, painéis de indicadores (dashboards), BSC (Balanced Scorecard), CRM (Custom Relationship Management), entre outras que necessitam analisar e combinar informações de fontes estruturadas e não estruturadas.

5.2 LIMITAÇÕES E TRABALHOS FUTUROS

Os próximos passos compreendem a utilização de repositórios externos a organização, como exemplo, utilizar a ontologia de DOMÍNIO para invocar serviços web e estender a ontologia de MAPEAMENTO para permitir a configuração de serviços web dinâmicos. Outro item importante a ser trabalhado é o suporte ao uso de inferências pelo componente de requisição de informações, por exemplo, por meio da definição de regras podem-se aplicar *slices* e *drills* semânticos conforme a arquitetura proposta por Sell (2006).

A aplicação proposta limita-se a utilização de uma ontologia de domínio, então novas pesquisas podem ser realizadas a fim de incluir uma visão multiontológica e buscar uma solução de como realizar a integração semântica de conceitos similares ou parecidos em ontologias distintas.

Utilizando a ontologia de domínio como um modelo de consulta limita-se a leitura de dados é interessante buscar soluções que estendam essa visão e permitam realizar operações como inclusão, alteração e exclusão de dados, possibilitando realizar todas as operações de persistência de dados por meio de uma *API*¹⁵ padrão.

¹⁵ *Application Programming Interface* ou Interface de Programação de Aplicações

Nesse trabalho os dados não estruturados textuais são representados por campos de conteúdo aberto. Esses campos permitiram realizar cortes (*slices*) nas análises, porém há a necessidade de buscar novas abordagens que permitam aplicar outras operações analíticas sobre esses dados não estruturados textuais.

Essa sugestão seria interessante aplicar no processo de extração, transformação e carga dos dados para um repositório, permitindo realizar o processo de ETL por meio de uma ontologia de domínio.

REFERÊNCIAS

ALBUQUERQUE, Nikolai Dimitrii Braga de. **Uma Arquitetura para o Compartilhamento do Conhecimento em Bibliotecas Digitais**. 2003. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina.

ÁVILA, Elizabet Tejada. **Data Warehousing con procesamiento de datos textuales**. 2010. Tese (Doutorado em Informática) – Universidade de Granada. Departamento de Ciências da Computação e Inteligência Artificial. Espanha – Granada.

D-LIB MAGAZINE. **An Introduction to the Resource Description Framework**. United States: D-lib Magazine. maio 1998.

BALLARD, Chuck; FARRELL, Daniel; GUPTA, Amit; MAZUELA, Carlos; VOHNIK, Stanislav. **Dimensional Modeling: In a Business Intelligence Environment**. 670p. ISBN 0738496448. Vervante. 2006.

BARBIERI, Carlos. **Business Intelligence: Modelagem e Tecnologia**. 2001. Axcel Books. ISBN: 8573231483.

BRAGANHOLO, Vanessa; HEUSER, Carlos. **XML Schema, RDF(S) e UML: uma comparação**. In: IDEAS 2001 - 4th Iberoamerican Workshop on Requirements Engineering and Software Environments, Santo Domingo, Heredia, Costa Rica, 2001. p. 78-90.

BEPPLER, Fabiano; TODESCO, José Leomar; GONÇALVES, Alexandre; SELL, Denilson; MORALES, Aran Bey Tcholakian; PACHECO, Roberto Carlos dos Santos. 2005. **Uma Arquitetura para Recuperação de Informação Aplicada ao Processo de Cooperação Universidade-Empresa**, In: KM BRASIL, São Paulo.

BURRELL, Gibson; MORGAN, Gareth. **Sociological paradigms and organizational analysis**. London: Heinemann Educational Books, 1979.

CAPES. Coleta de Dados 12.0: **Manual do usuário**: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). 2009.

CASTOLDI, André Vinícius. **Uma ontologia para enlaces de unidades de informação em plataformas de governo eletrônico**. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina. 2003.

CECI, Flávio. **Um Modelo Semiautomático para a Construção e Manutenção de Ontologias a partir de Bases de Documentos não estruturados**. Dissertação (Mestrado em Engenharia de Conhecimento) – Universidade Federal de Santa Catarina. 2010.

CHAIDEZ, Jovany. **Business Intelligence & IT Governance: The current trend and its implication on modern businesses**. University of Illinois at Urbana-Champaign. 2008.

CNPQ. Conselho Nacional de Desenvolvimento Científico e Tecnológico. **A Plataforma Lattes**. Disponível em: <<http://lattes.cnpq.br/conteudo/aplataforma.htm>>. Acesso em Janeiro de 2011.

CORRENDO, Gianluco; SALVADORES, Manuel; MILLARD, Ian; GLASER, Hugh; SHADBOLT, Nigel. **SPARQL Query Rewriting for Implementing Data Integration over Linked Data**. Proceedings of the 1st International Workshop on Data Semantics DataSem 10. 2010.

CZERNICKI, Bart. **Silverlight 4: Business Intelligence Software**. ISBN10: 1-4302-3060-6. 576 pags. 2010.

DATTA, Anindya; THOMAS, Helen. **The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses.** Decision Support Systems, volume 27, p.289-301, ISSN 0167-9236. 1999.

DING, Bolin; ZHAO, Bo; LIN, Cindy Xide; HAN, Jiawei; ZHAI, ChengXiang. **TopCells: Keyword-based search of top-k aggregated documents in text cube.** Data Engineering (ICDE), IEEE 26th International Conference, p.381-384.2010.

EGC 1. **Engenharia do Conhecimento.** Disponível em: <http://www.egc.ufsc.br/index.php?option=com_content&view=article&id=37:engenharia-do-conhecimento&catid=26:areas&Itemid=45&lang=pt>. Acessado em: 20/09/2010.

EGC 2. **Engenharia do Conhecimento aplicada às organizações.** Disponível em: <http://www.egc.ufsc.br/index.php?option=com_content&view=article&id=44:engenharia-do-conhecimento-aplicada-as-organizacoes&catid=27:linhas-de-pesquisa&Itemid=46&lang=pt>. Acessado em: 20/09/2010.

EVELSON, Boris; BROWN Matthew. **Search + BI: Unified Information Access: Combining Unstructured And Structured Info Delivers Business Insight.** Forrester. 2008.

FOLGOSO, Sandro Martínez. **Tratamiento semántico de atributos textuales en un Modelo Relacional Orientado a Objetos: Implementación en Software Libre.** 2008. Tese (Doutorado em Informática) – Universidade de Granada. Departamento de Ciências da Computação e Inteligência Artificial. Espanha – Granada.

GOLFARELLI, Matteo; RIZZI, Stefano. **Data Warehouse Design: Modern Principles and Methodologies**. McGraw-Hill. ISBN-10: 9780071610391.2009.

GONZAGA, Tatiana. **Uma metodologia para o desenvolvimento de instrumentos de análise multidimensional da informação em projetos de governo eletrônico voltado ao cidadão**. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina. 2005.

GUARINO, Nicola. **Formal Ontology and Information Systems**. Proceedings of FOIS'98, Trento, Itália.1998.

GRUBER, Thomas. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**. International Journal Human-Computer Studies Vol. 43. Padova, Itália. 1993.

HENDLER, James. **Agents and the Semantic Web**. IEEE Intelligent Systems. Volume 16, número 2. 2001.

HORROCKS, Ian; FENSEL, Dieter; BROEKSTRA, Jeen; DECKER, Stefan; ERDMANN, Michael; GOBLE, Carole; HARLEMEN, Frank van; KLEIN, Michel; STAAB, Steffen; STUDER, Rudi; MOTTA, Enrico. **The Ontology Inference Layer OIL**. 2000.

HORROCKS, Ian; PATEL-SCHNEIDER, Peter F.; HARMELEN, Frank Van. **From SHIQ and RDF to OWL: The Making of a Web Ontology Language**. In: Journal of Web Semantics. p. 7-26. 2003.

IDC. **The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010**. IDC White paper - EMC. 2007.

INOKUCHI, Akihiro; TAKEDA, Koichi. **A method for online analytical processing of text data. Conference on information and knowledge management.** CIKM. ACM, New York, USA. 2007.

INMON, William Harvey. **Como construir o Data Warehouse.** 2 ed., Rio de Janeiro: Campus, 1997.

INMON, William Harvey; STRAUSS, Derek; NEUSHLOSS, Genia. **DW 2.0: The Architecture for the Next Generation of Data Warehousing.** 2008.

KIMBALL, Ralph et al. **The Data Warehouse Lifecycle Toolkit: Expert Methods For Designing, Developing and Deploying Data Warehouses.** New York: John Wiley & Sons Inc., 1998.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit: The complete guide to dimensional modeling.** 2. ed. USA: Wiley, 2002.

KZAZ, Larbi, ELASRI, Hicham, SEKKAKI Abderrahim. **A model for semantic integration of business components. Internacional Journal of Computer Science & Information Technology.** IJCSIT, vol. 2, p. 1-12, 2010.

LEE, Jinho; GROSSMAN, David; FRIEDER, Ophir; MCCABE, M. Catherine. 2000. **Integrating Structured Data and Text: A Multi-Dimensional Approach.** In **Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00).** IEEE Computer Society, Washington, DC, USA.

LENZERINI, Maurizio. **Data integration: a theoretical perspective.** In **Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '02).** ACM, New York, NY, USA, p. 233-246. 2002.

LIN, Cindy Xide; et al. **Text Cube**: Computing IR Measures for Multidimensional Text Database Analysis. ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, p.905–910, Washington, DC, USA, 2008.

MIKROYANNIDIS, Alexander; THEODOULIDIS, Babis; PERSIDIS, Andreas. **PARMENIDES**: Towards Business Intelligence Discovery from Web Data. In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI '06). IEEE Computer Society, Washington, DC, USA, 2006.

MOSS, Larissa; SHAKU, Atre. **Business Intelligence Roadmap**: The Complete Project Lifecycle for Decision-Suport Applications. Addison Wesley. 2003. ISBN 0-201-78420-3.

MSDN. **XML for Analysis**. 2001. Disponível em:
<<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnxmlspec/html/xmlanalysis.asp>>. Acesso em: 25 maio 2006.

NAPOLI, Marcio; SELL, Denilson; LIDUÁRIO, Luciano; BORGES, Rodrigo Ferreira; TODESCO, José Leomar; PACHECO, Roberto Carlos dos Santos. **Um Framework para Concepção de Ferramentas de Apoio a Decisão Baseadas em Ontologias**. In: Simpósio Brasileiro de Banco de Dados (SBBD), Florianópolis. Anais, 2006.

NAPOLI, Marcio; SELL, Denilson. **Uma abordagem baseada em ontologias para realizar operações TextOLAP**. In: Ontobras, 2010, Florianópolis. 3º Seminário de Pesquisa em Ontologia no Brasil, 2010.

NARDI, Alexandre Ricardo. Fundamentos e Modelagem de Banco de dados Multidimensionais. Publicado em 2007. Acesso em: 03 janeiro 2010.

NELSON, Gregory S. **Business Intelligence 2.0: Are we there yet?**. The SAS Global Forum. Seattle. 2010.

NOWACK, Benjamin. **Semantic Web Layer Cake**. Disponível em: <http://smiy.files.wordpress.com/2011/01/sw_layercake.png>. Acesso em: 22 janeiro 2011.

RASMUSSEN, Nils H.; GOLDY, Paul S.; SOLLI, Per O. **Financial Business Intelligence: Trends, Technology, Software Selection, and Implementation**. John Wiley & Sons Inc., 2002. ISBN 0-471-15555-1.

RDFS. **RDF Vocabulary Description Language 1.0: RDF Schema**. 2004. Disponível em: <<http://www.w3.org/TR/rdf-schema/>>. Acesso em: 26 maio 2006.

RDF. **Resource Description Framework (RDF)**. 1999. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em: 26 maio 2006.

SCHREIBER, Guss; AKKERMANS, Hans; ANJEWIERDEN, Anjo; HOOG, Roberto de; SHADBOLT, Nigel; VELDE, Walter Van de; WIELINGA, Bob. **Knowledge Engineering and Management: the CommonKADS Methodology**. MIT Press. Cambridge. Massachussets. 1999.

SELL, Denilson. **Uma arquitetura para Business Intelligence baseada em tecnologias semânticas para suporte a aplicações analíticas**. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2006.

SILVA, Edna Lúcia da; Menezes, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3º edição, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção. Laboratório de Ensino a Distância, Florianópolis, 2001.

SILVA, Geiza C.; LIMA, Tarcízio de Souza. **RDF e RDFS na Infra-estrutura de suporte à Web Semântica**. Revista Eletrônica de Iniciação Científica. Ano II. Volume II. Número 1. 2002. ISSN 1519-8219. Acesso em: 24 de maio de 2004.

SINT, Rolf; SCHAFFERT, Sebastian; STROKA, Stephanie; FERSTL, Roland. **Combining Unstructured, Fully Structured and Semi-Structured Information in Semantic Wikis**. In: 4th Workshop on Semantic Wikis (SemWiki2009) at ESWC09, Heraklion, Greece. 2009.

SOWA, John. **Knowledge Representation: Logical, Philosophical and Computational Foundations**. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

XML. **Extensible Markup Language (XML)**. Disponível em: <<http://www.w3.org/XML/>>. Acesso em Junho de 2006.

XMLA. **XML for Analysis**. Disponível em: <<http://www.xmlforanalysis.com/>>. Acesso em: 24 maio. 2005

YEH, Peter Z; KASS, Alex. **Capturing the Semantics of Online News Sources for Business Intelligence Applications**. Proceedings of the IEEE International Conference on Tools with Artificial Intelligence. ICTAI. 2008.

ZHANG, Duo; Zhai, ChengXiang; Han, Jiawei; Srivastava, Asholk; Oza Nikunk. **Topic Modeling for OLAP on Multidimensional Text Databases: Topic Cube and its Applications**. Statistical Analysis and Data Mining vol. 2. 378-395. 2009.

ZIEGLER, Patrick; DITTRICH, Klaus R. **Three Decades of Data Integration: All Problems solved ?**. In 18th IFIP World Computer Congress (WCC 2004), Volume 12, Building the Information Society. 2004.

W3CSEMANTICWEB. **W3C Semantic Web Activity**. Disponível em: <<http://www.w3.org/2001/sw/>>. Acesso em Junho de 2006.

APÊNDICE A - Ontologia de Domínio

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY egc "http://www.egc.ufsc.br#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
  <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
  <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY daml "http://www.daml.org/2001/03/daml+oil#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
  <!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
]>
<rdf:RDF xmlns="http://www.egc.ufsc.br#"
  xml:base="http://www.egc.ufsc.br"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:egc="http://www.egc.ufsc.br#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <owl:ObjectProperty rdf:about="&egc;hasDisciplina">
    <rdfs:range rdf:resource="&egc;Disciplina"/>
    <rdfs:domain rdf:resource="&egc;Pesquisador"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="&egc;hasProducao">
    <rdfs:domain rdf:resource="&egc;Pesquisador"/>
    <rdfs:range rdf:resource="&egc;Producao"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="&egc;ano">
    <rdfs:type rdf:resource="&owl;FunctionalProperty"/>

```

```

    <rdfs:domain rdf:resource="&egc;Producao"/>
    <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;area_atuacao">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Pesquisador"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&egc;endereco_profissional">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Pesquisador"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;faixa_etaria">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Pesquisador"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;grande_area_atuacao">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Pesquisador"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;linha_pesquisa">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Producao"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;nivel">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Disciplina"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;nome_disciplina">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Disciplina"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;nome ">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&egc;Pesquisador"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&egc;nome_producao">

```

```

    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="&egc;Producao"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="&egc;numero_creditos">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="&egc;Disciplina"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="&egc;sigla">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="&egc;Disciplina"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="&egc;tipo_producao">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="&egc;Producao"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="&egc;titulacao_maxima">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="&egc;Pesquisador"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:Class rdf:about="&egc;Disciplina"/>
  <owl:Class rdf:about="&egc;Pesquisador"/>
  <owl:Class rdf:about="&egc;Producao"/>
</rdf:RDF>

```

APÊNDICE B - Ontologia de Análise

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
  <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
  <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
  <!ENTITY iso "http://www.egc.ufsc.br/iso#" >
  <!ENTITY vio "http://www.egc.ufsc.br/vio#" >
  <!ENTITY ano "http://www.egc.ufsc.br/ano#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY pro "http://www.egc.ufsc.br/profile#" >
  <!ENTITY daml "http://www.daml.org/2001/03/daml+oil#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >

```

```

<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
<!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
<!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
]>
<rdf:RDF xmlns="http://www.egc.ufsc.br/base#"
  xml:base="http://www.egc.ufsc.br/base"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:iso="http://www.egc.ufsc.br/iso#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:vio="http://www.egc.ufsc.br/vio#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:pro="http://www.egc.ufsc.br/profile#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ano="http://www.egc.ufsc.br/ano#">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
    <owl:Ontology>
    <owl:ObjectProperty rdf:about="&ano;hasAggregateFunction">
      <rdfs:domain rdf:resource="&ano;MeasureUnit"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&ano;hasElement">
      <rdfs:range rdf:resource="&ano;Element"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&ano;hasDetail">
      <rdfs:range rdf:resource="&ano;Detail"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&ano;hasDetailDeviance">
      <rdfs:domain rdf:resource="&ano;Detail"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&ano;hasFilter">
      <rdfs:comment xml:lang="en"
>Defines a fixed filter to be applied in the queries</rdfs:comment>
      <rdfs:comment xml:lang="pt"
>Define um filtro fixo a serem aplicados nas consultas ou
buscas</rdfs:comment>
      <rdfs:domain rdf:resource="&ano;AnalyticalElement"/>
    </owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:about="&ano;hasFilterDetail">
  <rdfs:range rdf:resource="&ano;Detail"/>
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasGroupDetail">
  <rdfs:range rdf:resource="&ano;Detail"/>
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasGrouping">
  <rdfs:range rdf:resource="&ano;Grouping"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&ano;Criteria"/>
        <rdf:Description rdf:about="&ano;ThemeUnit"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasGroupingParent">
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
  <owl:inverseOf rdf:resource="&ano;hasGroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasGroupingUnit">
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHierarchicalChild">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHierarchicalFilter">
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHierarchicalFilterLevel">
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHierarchicalFilterParent">
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHierarchicalParent">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>

```

```

    <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
    <rdfs:range rdf:resource="&ano;GroupingUnit"/>
    <owl:inverseOf rdf:resource="&ano;hasHierarchicalChild"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasHintGroupingUnit">
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasMeasureDeviance">
  <rdfs:domain rdf:resource="&ano;MeasureUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasMeasureUnit">
  <rdfs:range rdf:resource="&ano;MeasureUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasSearchField">
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
  <rdfs:domain rdf:resource="&ano;SearchThemeUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasSortGroupingUnit">
  <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
  <rdfs:range rdf:resource="&ano;GroupingUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasThemeParent">
  <rdfs:range rdf:resource="&ano;Theme"/>
  <rdfs:domain rdf:resource="&ano;ThemeUnit"/>
  <owl:inverseOf rdf:resource="&ano;hasThemeUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&ano;hasThemeUnit">
  <rdfs:comment xml:lang="en"
    >Associates a ThemeUnit's instances to an Theme's
instance</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Associa uma instancia de ThemUnit a uma instancia de
Theme</rdfs:comment>
  <rdfs:domain rdf:resource="&ano;Theme"/>
  <rdfs:range rdf:resource="&ano;ThemeUnit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasParent">
  <rdfs:comment xml:lang="en"
    >Represents the element's parent, in case it exist</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Representa o elemento-pai, caso exista um</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&vio;hasIcon">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment xml:lang="en"

```

```

    >Defines an icon for the analytical element</rdfs:comment>
    <rdfs:comment xml:lang="pt"
    >Define um cone para o elemento
anal&#237;tico</rdfs:comment>
    <rdfs:domain rdf:resource="&ano;AnalyticalElement"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&vio;hasLabel">
    <rdfs:comment xml:lang="en"
    >Defines a label for the analytical element or information
source</rdfs:comment>
    <rdfs:comment xml:lang="pt"
    >Define um label para o elemento anal&#237;tico ou elemento de fonte
de informa&#231;&#227;o</rdfs:comment>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&vio;hasOrderedDetail">
    <rdfs:domain>
    <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
    <rdf:Description rdf:about="&ano;MeasureUnit"/>
    <rdf:Description rdf:about="&ano;ThemeUnit"/>
    </owl:unionOf>
    </owl:Class>
    </rdfs:domain>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&vio;hasOrderedGrouping">
    <rdfs:domain rdf:resource="&ano;ThemeUnit"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:about="&vio;hasOrderedGroupingUnit">
    <rdfs:domain rdf:resource="&ano;Grouping"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&vio;hasOrderedMeasure">
    <rdfs:domain rdf:resource="&ano;BIThemeUnit"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&vio;hasOrderedThemeUnit">
    <rdfs:domain rdf:resource="&ano;Theme"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:about="&ano;result">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:about="&ano;groupingUnitType">
    <rdfs:domain rdf:resource="&ano;GroupingUnit"/>
    <rdfs:range>
    <rdf:Description>
    <rdf:type rdf:resource="&owl;DataRange"/>

```

```

<owl:oneOf>
  <rdf:Description>
    <rdf:type rdf:resource="&rdf;List"/>
    <rdf:first rdf:datatype="&xsd;string">COMPOSED</rdf:first>
    <rdf:rest>
      <rdf:Description>
        <rdf:type rdf:resource="&rdf;List"/>
        <rdf:first rdf:datatype="&xsd;string">HIRARCHICAL</rdf:first>
        <rdf:rest>
          <rdf:Description>
            <rdf:type rdf:resource="&rdf;List"/>
            <rdf:first rdf:datatype="&xsd;string">LOOK_UP</rdf:first>
            <rdf:rest>
              <rdf:Description>
                <rdf:type rdf:resource="&rdf;List"/>
                <rdf:first rdf:datatype="&xsd;string">PLAIN_TEXT</rdf:first>
                <rdf:rest>
                  <rdf:Description>
                    <rdf:type rdf:resource="&rdf;List"/>
                    <rdf:first rdf:datatype="&xsd;string">RANGE</rdf:first>
                    <rdf:rest rdf:resource="&rdf:nil"/>
                  </rdf:Description>
                </rdf:rest>
              </rdf:Description>
            </rdf:rest>
          </rdf:Description>
        </rdf:rest>
      </rdf:Description>
    </rdf:rest>
  </owl:oneOf>
</rdf:Description>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&ano;order">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&ano;Theme"/>
  <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&vio;backgroundColor">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&vio;country">

```



```

<rdfs:range>
  <rdf:Description>
    <rdf:type rdf:resource="&owl;DataRange"/>
    <owl:oneOf>
      <rdf:Description>
        <rdf:type rdf:resource="&rdf;List"/>
        <rdf:first rdf:datatype="&xsd;string">BR</rdf:first>
        <rdf:rest>
          <rdf:Description>
            <rdf:type rdf:resource="&rdf;List"/>
            <rdf:first rdf:datatype="&xsd;string">CA</rdf:first>
            <rdf:rest>
              <rdf:Description>
                <rdf:type rdf:resource="&rdf;List"/>
                <rdf:first rdf:datatype="&xsd;string">CN</rdf:first>
                <rdf:rest>
                  <rdf:Description>
                    <rdf:type rdf:resource="&rdf;List"/>
                    <rdf:first rdf:datatype="&xsd;string">DE</rdf:first>
                    <rdf:rest>
                      <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">ES</rdf:first>
                        <rdf:rest>
                          <rdf:Description>
                            <rdf:type rdf:resource="&rdf;List"/>
                            <rdf:first rdf:datatype="&xsd;string">FR</rdf:first>
                            <rdf:rest>
                              <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first rdf:datatype="&xsd;string">GB</rdf:first>
                                <rdf:rest>
                                  <rdf:Description>
                                    <rdf:type rdf:resource="&rdf;List"/>
                                    <rdf:first rdf:datatype="&xsd;string">IT</rdf:first>
                                    <rdf:rest>
                                      <rdf:Description>
                                        <rdf:type rdf:resource="&rdf;List"/>
                                        <rdf:first rdf:datatype="&xsd;string">JP</rdf:first>
                                  </rdf:rest>
                                </rdf:rest>
                              </rdf:rest>
                            </rdf:rest>
                          </rdf:rest>
                        </rdf:rest>
                      </rdf:rest>
                    </rdf:rest>
                  </rdf:rest>
                </rdf:rest>
              </rdf:rest>
            </rdf:rest>
          </rdf:rest>
        </rdf:rest>
      </rdf:rest>
    </rdf:rest>
  </rdf:rest>
</rdf:rest>
<rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">KR</rdf:first>
  <rdf:rest>
    <rdf:Description>

```



```

</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&vio;language">
  <rdfs:range>
    <rdf:Description>
      <rdf:type rdf:resource="&owl;DataRange"/>
      <owl:oneOf>
        <rdf:Description>
          <rdf:type rdf:resource="&rdf;List"/>
          <rdf:first rdf:datatype="&xsd;string">de</rdf:first>
          <rdf:rest>
            <rdf:Description>
              <rdf:type rdf:resource="&rdf;List"/>
              <rdf:first rdf:datatype="&xsd;string">default</rdf:first>
              <rdf:rest>
                <rdf:Description>
                  <rdf:type rdf:resource="&rdf;List"/>
                  <rdf:first rdf:datatype="&xsd;string">en</rdf:first>
                  <rdf:rest>
                    <rdf:Description>
                      <rdf:type rdf:resource="&rdf;List"/>
                      <rdf:first rdf:datatype="&xsd;string">es</rdf:first>
                      <rdf:rest>
                        <rdf:Description>
                          <rdf:type rdf:resource="&rdf;List"/>
                          <rdf:first rdf:datatype="&xsd;string">fr</rdf:first>
                          <rdf:rest>
                            <rdf:Description>
                              <rdf:type rdf:resource="&rdf;List"/>
                              <rdf:first rdf:datatype="&xsd;string">it</rdf:first>
                              <rdf:rest>
                                <rdf:Description>
                                  <rdf:type rdf:resource="&rdf;List"/>
                                  <rdf:first rdf:datatype="&xsd;string">ja</rdf:first>
                                  <rdf:rest>
                                    <rdf:Description>
                                      <rdf:type rdf:resource="&rdf;List"/>
                                      <rdf:first rdf:datatype="&xsd;string">ko</rdf:first>
                                      <rdf:rest>
                                        <rdf:Description>
                                          <rdf:type rdf:resource="&rdf;List"/>
                                          <rdf:first rdf:datatype="&xsd;string">nl</rdf:first>
                                          <rdf:rest>
                                            <rdf:Description>
                                              <rdf:type rdf:resource="&rdf;List"/>
                                              <rdf:first rdf:datatype="&xsd;string">pt</rdf:first>

```



```

</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&vio;order">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
<owl:Class rdf:about="&ano;Element">
  <rdfs:comment xml:lang="en"
    >The Element class abstracts any ontology entity. All classes must
inherit from Element class</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >A classe Element abstrai qualquer entidade da ontologia. Todas as
demais classes dessa ontologia herdam de Element.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="&ano;ElementSequence">
  <rdfs:subClassOf rdf:resource="&ano;Element"/>
</owl:Class>
<owl:Class rdf:about="&ano;AnalyticalElement">
  <rdfs:subClassOf rdf:resource="&ano;Element"/>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="&ano;hasFilter"/>
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="&vio;hasIcon"/>
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en"
    >This class represents the ontolgy's elements related to search and
analytical operations. An AnalyticalElement is used to guide the creating of
graphic intefaces, to assist the information navigability and acessibility and also
to guide the queries on information sources</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Essa classe representa os elementos da ontologia relacionados &#224;s
opera&#231;&#245;es de buscas ou an&#225;lises. Um AnalyticalElement
&#233; usado para guiar a montagem das interfaces anal&#237;ticas, auxiliar
na navegabilidade e acessibilidade das informa&#231;&#245;es e guiar as
consultas sobre as fontes de informa&#231;&#227;o;</rdfs:comment>
  </owl:Class>
<owl:Class rdf:about="&ano;BIFThemeUnit">

```

```

    <rdfs:subClassOf rdf:resource="&ano;ThemeUnit"/>
  </owl:Class>
  <owl:Class rdf:about="&ano;Criteria">
    <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&ano;hasMeasureUnit"/>
        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="&ano;hasGrouping"/>
          <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment xml:lang="en"
          >Class that represents a criteria, i.e., a grouping applied to a
measure.</rdfs:comment>
        <rdfs:comment xml:lang="pt"
          >Classe que representa um crit&#233;rio, sendo esse, um agrupamento
aplicado a uma medida.</rdfs:comment>
        </owl:Class>
        <owl:Class rdf:about="&ano;Detail">
          <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
          <rdfs:comment xml:lang="en"
            >Represents an attribute that details an information. A Detail is used to
show an information from search or analysis&#39; result.</rdfs:comment>
          <rdfs:comment xml:lang="pt"
            >Representa um atributo que detalha uma informa&#231;&#227;o. Um
Detail &#233; usado para mostrar uma informa&#231;&#227;o do objeto de
resultado de uma busca ou an&#225;lise</rdfs:comment>
          </owl:Class>
          <owl:Class rdf:about="&ano;Edge">
            <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
          </owl:Class>
          <owl:Class rdf:about="&ano;Graph">
            <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
          </owl:Class>
          <owl:Class rdf:about="&ano;Grouping">
            <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
            <rdfs:comment xml:lang="en"

```

>Grouping represents a group of information in order to classify or
agregater informations</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Grouping representa um grupo de informações com a
finalidade de classificar ou agrupar as informações.

(ex: Geografia, Área de conhecimento, Tempo, Setor CNAE
etc.)</rdfs:comment>

```
</owl:Class>
```

```
<owl:Class rdf:about="&ano;GroupingUnit">
```

```
<rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&ano;groupingUnitType"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&ano;hasHierarchicalFilterLevel"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&ano;hasGroupingParent"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&ano;hasFilterDetail"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty
```

```
rdf:resource="&ano;hasHierarchicalFilterParent"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```

</rdfs:subClassOf>
</rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&ano;hasGroupDetail"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&ano;hasHierarchicalFilter"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&ano;hasSortGroupingUnit"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</rdfs:comment xml:lang="en"
  >GroupingUnit is an information unit associated to a
Grouping</rdfs:comment>
  <rdfs:comment xml:lang="pt"
  >GroupingUnit &#233; uma unidade de informa&#231;&#227;o dentro
de um Grouping.
(ex:Dado o Grouping Geografia podemos ter como GroupingUnit as UFs, as
Regi&#245;es e os Pa&#237;ses)</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="&ano;MeasureUnit">
  <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&ano;hasAggregateFunction"/>
      <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en"
  >Class that represents a measure unit or an indicator related to a
ThemUnit</rdfs:comment>
  <rdfs:comment xml:lang="pt"

```


>Classe que representa uma unidade de medida ou um indicador relacionado a ThemeUnit.</rdfs:comment>

```
</owl:Class>
<owl:Class rdf:about="&ano;SearchThemeUnit">
  <rdfs:subClassOf rdf:resource="&ano;ThemeUnit"/>
</owl:Class>
```

```
<owl:Class rdf:about="&ano;Theme">
  <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&ano;hasThemeUnit"/>
      <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en"
```

>Class that represents a theme or um subject related to an analysis or an search</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Classe que representa um tema ou um assunto usado na análise ou na busca</rdfs:comment>

```
</owl:Class>

<owl:Class rdf:about="&ano;ThemeUnit">
  <rdfs:subClassOf rdf:resource="&ano;AnalyticalElement"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&ano;hasThemeParent"/>
      <owl:minCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en"
```

>Class that represents a theme unit, that may be an analysis unit or search unit</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Classe que representa uma unidade do tema, que pode ser uma unidade de análise ou ainda, uma unidade de busca.</rdfs:comment>

```
</owl:Class>
<owl:Class rdf:about="&owl;Thing"/>
<owl:Thing rdf:about="&ano;Locale_de_DE">
  <vio:country rdf:datatype="&xsd:string">DE</vio:country>
  <vio:language rdf:datatype="&xsd:string">de</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_default">
```

```

    <vio:country rdf:datatype="&xsd:string">default</vio:country>
    <vio:language rdf:datatype="&xsd:string">default</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_en_CA">
    <vio:country rdf:datatype="&xsd:string">CA</vio:country>
    <vio:language rdf:datatype="&xsd:string">en</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_en_GB">
    <vio:country rdf:datatype="&xsd:string">GB</vio:country>
    <vio:language rdf:datatype="&xsd:string">en</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_en_US">
    <vio:country rdf:datatype="&xsd:string">US</vio:country>
    <vio:language rdf:datatype="&xsd:string">en</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_es_ES">
    <vio:country rdf:datatype="&xsd:string">ES</vio:country>
    <vio:language rdf:datatype="&xsd:string">es</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_fr_CA">
    <vio:country rdf:datatype="&xsd:string">CA</vio:country>
    <vio:language rdf:datatype="&xsd:string">fr</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_fr_FR">
    <vio:country rdf:datatype="&xsd:string">FR</vio:country>
    <vio:language rdf:datatype="&xsd:string">fr</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_it_IT">
    <vio:country rdf:datatype="&xsd:string">IT</vio:country>
    <vio:language rdf:datatype="&xsd:string">it</vio:language>
</owl:Thing>
<owl:Thing rdf:about="&ano;Locale_pt_BR">
    <vio:country rdf:datatype="&xsd:string">BR</vio:country>
    <vio:language rdf:datatype="&xsd:string">pt</vio:language>
</owl:Thing>
</rdf:RDF >

```

APÊNDICE C - Ontologia de Mapeamento

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <ENTITY ekp "http://www.egc.ufsc.br/ekp#" >
    <ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
    <ENTITY dc "http://purl.org/dc/elements/1.1/" >

```

```

<!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
<!ENTITY iso "http://www.egc.ufsc.br/iso#" >
<!ENTITY vio "http://www.egc.ufsc.br/vio#" >
<!ENTITY ano "http://www.egc.ufsc.br/ano#" >
<!ENTITY base "http://www.egc.ufsc.br/base#" >
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
<!ENTITY pro "http://www.egc.ufsc.br/profile#" >
<!ENTITY daml "http://www.daml.org/2001/03/daml+oil#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
<!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
<!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
]
<rdf:RDF xmlns="http://www.egc.ufsc.br/base#"
  xml:base="http://www.egc.ufsc.br/base"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:iso="http://www.egc.ufsc.br/iso#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:vio="http://www.egc.ufsc.br/vio#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:base="http://www.egc.ufsc.br/base#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:pro="http://www.egc.ufsc.br/profile#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ano="http://www.egc.ufsc.br/ano#">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <owl:ObjectProperty rdf:about="&iso;hasAggregateFunction">
    <rdfs:range rdf:resource="&iso;AggregateFunction"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="&iso;hasElement">
    <rdfs:range rdf:resource="&iso;Element"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="&iso;hasExpression">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="&iso;Expression"/>

```

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasFirstExpression">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;BinaryExpression"/>
  <rdfs:range rdf:resource="&iso;Expression"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasFunction">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&iso;Function"/>
  <rdfs:domain rdf:resource="&iso;UnaryExpression"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasSecondExpression">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;BinaryExpression"/>
  <rdfs:range rdf:resource="&iso;Expression"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasAttribute">
  <rdfs:range rdf:resource="&iso;Attribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasChild">
  <rdfs:comment xml:lang="en"
    >Represents the element child contained in the
collection</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Represetna um elemento-filho contido na
cole&#231;&#227;o</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;Collection"/>
  <rdfs:range rdf:resource="&iso;InformationSource"/>
  <owl:inverseOf rdf:resource="&iso;hasParent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasDomainClass">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment xml:lang="en"
    >Represents the domain ontolgy&#39;s class related to information
source&#39;s instance</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Representa a classe da ontologia de dom&#237;nio relacionada
&#224; inst&#226;ncia da fonte de informa&#231;&#227;o</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;InformationSource"/>
  <rdfs:range rdf:resource="&rdfs;Class"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&iso;hasDomainProperty">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment xml:lang="en"

```

>Represents a domain ontology's property related to information source's instance</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Representa uma propriedade da ontologia de domínio relacionada a instância da fonte de informação;</rdfs:comment>

```
<rdfs:domain rdf:resource="&iso;Attribute"/>
```

```
<rdfs:range rdf:resource="&rdf;Property"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasInformationSource">
```

```
<rdfs:domain rdf:resource="&iso;UnaryExpression"/>
```

```
<rdfs:range rdf:resource="&iso;InformationSource"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasKeyAttribute">
```

```
<rdfs:type rdf:resource="&owl;FunctionalProperty"/>
```

```
<rdfs:range rdf:resource="&iso;TextAttribute"/>
```

```
<rdfs:domain rdf:resource="&iso;TextCollection"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasMergeUsage">
```

```
<rdfs:comment xml:lang="pt"
```

>Propriedade da classe Merge que define outros merges necessários para a ligação entre a collection de origem (source) e a collection de destino (target)</rdfs:comment>

```
<rdfs:range rdf:resource="&iso;Merge"/>
```

```
<rdfs:domain rdf:resource="&iso;Merge"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasParent">
```

```
<rdfs:comment xml:lang="en"
```

>Represents the element's parent, in case it exist</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Representa o elemento-pai, caso exista um</rdfs:comment>

```
<rdfs:range rdf:resource="&iso;Collection"/>
```

```
<rdfs:domain rdf:resource="&iso;InformationSource"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasRelation">
```

```
<rdfs:range rdf:resource="&iso;BinaryExpression"/>
```

```
<rdfs:domain rdf:resource="&iso;Merge"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&iso;hasSourceCollection">
```

```
<rdfs:type rdf:resource="&owl;FunctionalProperty"/>
```

```
<rdfs:comment xml:lang="pt"
```

>Propriedade da classe Merge que define qual collection o ponto de partida (origem) para ligação entre collections.</rdfs:comment>

```
<rdfs:range rdf:resource="&iso;Collection"/>
```



```

</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</owl:oneOf>
</rdf:Description>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;functionType">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;Function"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;isBracketed">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;Expression"/>
  <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;rawExpression">
  <rdfs:domain rdf:resource="&iso;Expression"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;rawFunction">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;Function"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;result">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&iso;Expression"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&ano;order">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;attributeSize">
  <rdfs:comment xml:lang="en"
  >Represents the maximum size of an attribute</rdfs:comment>
  <rdfs:comment xml:lang="pt"
  >Representa o tamanho máximo de um atributo</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;Attribute"/>
  <rdfs:range rdf:resource="&xsd;int"/>

```



```

</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;attributeType">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment xml:lang="en"
    >Defines the data type of an Attribute&#39;s instance</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Define o tipo de dado de uma inst&#226;ncia de
Attribute</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;Attribute"/>
  <rdfs:range>
  <rdf:Description>
  <rdf:type rdf:resource="&owl;DataRange"/>
  <owl:oneOf>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">BINARY</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">BOOLEAN</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">DATE</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">DATETIME</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">FLOAT_NUMBER</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">INTEGER_NUMBER</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">NULL</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd;string">OTHER</rdf:first>
  <rdf:rest>

```

```

<rdf:Description>
<rdf:type rdf:resource="&rdf;List"/>
<rdf:first rdf:datatype="&xsd:string">TEXT</rdf:first>
<rdf:rest>
<rdf:Description>
<rdf:type rdf:resource="&rdf;List"/>
<rdf:first rdf:datatype="&xsd:string">XML</rdf:first>
<rdf:rest rdf:resource="&rdf:nil"/>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</owl:oneOf>
</rdf:Description>
</rdfs:range>
  <owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:about="&iso;bidirectional">
      <rdfs:comment xml:lang="pt"
        >Define se a posi&#231;&#227;o das collections n&#227;o tem
        relev&#226;ncia no merge. Logo, quando o valor da propriedade for verdadeira
        (true), significa que o merge entre uma collection C1 e uma collection C2 pode
        ser aplicado entre C2 e C1 visto que implicam no mesmo resultado. Portanto, a
        posi&#231;&#227;o da collection (source ou target) &#233;
        ignorada.</rdfs:comment>
      <rdfs:domain rdf:resource="&iso;Merge"/>
      <rdfs:range rdf:resource="&xsd:boolean"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:about="&iso;isPrimaryKey">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <rdfs:comment xml:lang="en"

```

>Flag that defines whether the DBAttribute's instance is part of primary key or not.</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
  >Define se a instancia de DBAttribute faz parte da chave
  primária ou não.</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;DBAttribute"/>
  <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;mergeType">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment rdf:datatype="&xsd:string"
    >Propriedade que representa o tipo de ligação
    entre as diversas collections da fonte de
    informação.</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;Merge"/>
  <rdfs:range>
  <rdf:Description>
  <rdf:type rdf:resource="&owl;DataRange"/>
  <owl:oneOf>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">FULL_OUTER_JOIN</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">INDIRECT</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">INNER_JOIN</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">LEFT_OUTER_JOIN</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">RIGHT_OUTER_JOIN</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:datatype="&xsd:string">UNION</rdf:first>
  <rdf:rest>
  <rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
```

```

<rdf:first rdf:datatype="&xsd:string">UNION_ALL</rdf:first>
<rdf:rest rdf:resource="&rdf:nil"/>
</rdf:Description>
  </rdf:rest>
</rdf:Description>
  </rdf:rest>
  </rdf:Description>
  </rdf:rest>
  </rdf:Description>
  </rdf:rest>
  </rdf:Description>
  </rdf:rest>
  </rdf:Description>
  </rdf:rest>
  </owl:oneOf>
</rdf:Description>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;separator">
  <rdfs:comment xml:lang="en"
    >Represents the character used to separate the values of the
    CSVCollection's instance</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Representa o caracter used para separar os valores da inst&#226;ncia
    de CSVCollection</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;CSVCollection"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&iso;source">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment xml:lang="en"
    >Represents the element's name.</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Representa o nome do elemento</rdfs:comment>
  <rdfs:domain rdf:resource="&iso;InformationSource"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&vio;mask">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&vio;order">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

    <rdfs:range rdf:resource="&xsd:int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="&vio;path">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </owl:DatatypeProperty>
  <owl:Class rdf:about="&iso;AggregateFunction">
    <rdfs:subClassOf rdf:resource="&iso;Function"/>
    <rdfs:comment rdf:datatype="&xsd:string"
      >AggregateFunction &#233; um tipo de Function usada para auxiliar a
sumariza&#231;&#227;o de grandes volumes de dados</rdfs:comment>
    <rdfs:comment xml:lang="en"
      >AggregateFunction is type of Function used to assist with the
summarization of large volumes of data</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="&iso;BinaryExpression">
    <rdfs:subClassOf rdf:resource="&iso;Expression"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&iso;expressionOperator"/>
        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&iso;hasSecondExpression"/>
        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&iso;hasFirstExpression"/>
        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:comment xml:lang="en"
      >BinaryEspression is a type of expression in which are necessary two
expressions and an operator that associates both</rdfs:comment>
    <rdfs:comment xml:lang="pt"
      >BinaryExpression &#233; um tipo de express&#227;o na qual
s&#227;o necess&#225;rios duas outras express&#245;es e um operador que
associa ambas</rdfs:comment>

```

```
</owl:Class>
```

```
<owl:Class rdf:about="&iso;Element">
```

```
<rdfs:comment xml:lang="en"
```

>The Element class abstracts any ontology entity. All classes must inherit from Element class</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>A classe Element abstrai qualquer entidade da ontologia. Todas as demais classes dessa ontologia herdam de Element.</rdfs:comment>

```
</owl:Class>
```

```
<owl:Class rdf:about="&iso;Expression">
```

```
<rdfs:subClassOf rdf:resource="&iso;Element"/>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&iso;rawExpression"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&iso;result"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
<rdfs:comment xml:lang="en"
```

>Expression is a combination of values, variables, operators, and functions that are interpreted (evaluated) according to the particular rules of precedence, which computes and then produces (returns, in a stateful environment) another value.</rdfs:comment>

```
<rdfs:comment xml:lang="pt"
```

>Expression é uma combinação de valores, variáveis, operadores e funções que são interpretadas (avaliadas) de acordo com regras específicas de precedência, na qual valores são processados e retornados</rdfs:comment>

```
</owl:Class>
```

```
<owl:Class rdf:about="&iso;Function">
```

```
<rdfs:subClassOf rdf:resource="&iso;Element"/>
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="&iso;functionType"/>
```

```
<owl:maxCardinality
```

```
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```

<rdfs:comment xml:lang="en"
  >Function represents a task that associates a single output to each input
  element drawn from a fixed set</rdfs:comment>
<rdfs:comment xml:lang="pt"
  >Function representa a tarefa de associar um valor de sa&#237;da a
  cada valor de entrada</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="&iso;ScalarFunction">
  <rdfs:subClassOf rdf:resource="&iso;Function"/>
  <rdfs:comment xml:lang="en"
    >ScalarFunction is a type of Function that returns a single value based
    on the input value.</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >ScalarFunction &#233; um tipo de Functio quer retorna um
    &#250;nico valor conforme o valor de entrada</rdfs:comment>
  </owl:Class>
<owl:Class rdf:about="&iso;UnaryExpression">
  <rdfs:subClassOf rdf:resource="&iso;Expression"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;hasInformationSource"/>
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;hasFunction"/>
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en"
    >UnaryExpression is a kind of Expression with just one operating
    present in the whole expression. Moreover, it may have an operator or a
    function associated.</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >UnaryExpression &#233; um tipo de Expression com apenas um
    operando presente em toda a express&#227;o. Al&#233;m disso, pode ter um
    operador ou uma fun&#231;&#227;o associada.</rdfs:comment>
  </owl:Class>
<owl:Class rdf:about="&iso;Attribute">
  <rdfs:subClassOf rdf:resource="&iso;InformationSource"/>
  <rdfs:subClassOf>
    <owl:Restriction>

```

```

    <owl:onProperty rdf:resource="&iso;attributeType"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&iso;attributeSize"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&iso;hasDomainProperty"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment xml:lang="en"
  >Attribute class represents any property or relationship between others
ontology elements, such as, database table fields; XML document elements;
spreadsheet columns or values of CSV file; etc.</rdfs:comment>
  <rdfs:comment xml:lang="pt"
  >Classe que representa qualquer propriedade ou rela&#231;&#227;o
entre outros elementos da ontologia, tais como, atributos de tabelas ou
vis&#245;es (banco de dados); atributos de tags (XML); colunas de uma
planilha ou de um arquivo CSV; etc.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="&iso;CSVAttribute">
    <rdfs:subClassOf rdf:resource="&iso;Attribute"/>
  </owl:Class>
  <owl:Class rdf:about="&iso;CSVCollection">
    <rdfs:subClassOf rdf:resource="&iso;Collection"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&iso;separator"/>
        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="&iso;Collection">
    <rdfs:subClassOf rdf:resource="&iso;InformationSource"/>

```



```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&iso;hasChild"/>
    <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment xml:lang="en"

```

>Collection class represents any kind of elements' collection. It could be used to represent, for instance: the database tables, dimensions and fact tables XML document or nodes, in which owns some elements (XML); a set of comma-separated values (flat file); etc.</rdfs:comment>

```

<rdfs:comment xml:lang="pt"

```

>Representa qualquer tipo de coleção de elementos. Isto poderia ser usado para representar, por exemplo: tabelas, dimensões ou tabelas de fato (banco de dados); nodos ou tags (XML). linhas de uma planilha ou de uma arquivo CSV; etc.</rdfs:comment>

```

</owl:Class>

```

```

<owl:Class rdf:about="&iso;DBAttribute">

```

```

  <rdfs:subClassOf rdf:resource="&iso;Attribute"/>

```

```

  <rdfs:comment xml:lang="en"

```

>DBAttribute is a type of Attribute that represents a table's column defined in the database</rdfs:comment>

```

  <rdfs:comment xml:lang="pt"

```

>DBAttribute é um tipo de Attribute que representa um coluna de uma tabela definida na base de dados</rdfs:comment>

```

</owl:Class>

```

```

<owl:Class rdf:about="&iso;DBCcollection">

```

```

  <rdfs:subClassOf rdf:resource="&iso;Collection"/>

```

```

  <rdfs:comment xml:lang="en"

```

>DBCcollection is a type of Collection that contains a set of database objects, as catalog, schema or tables</rdfs:comment>

```

  <rdfs:comment xml:lang="pt"

```

>DBCcollection é um tipo de Collection que contém um conjunto de objetos da base de dados, como catálogos, schemas ou tabelas</rdfs:comment>

```

</owl:Class>

```

```

<owl:Class rdf:about="&iso;InformationSource">

```

```

  <rdfs:subClassOf rdf:resource="&iso;Element"/>

```

```

  <rdfs:subClassOf>

```

```

    <owl:Restriction>

```

```

      <owl:onProperty rdf:resource="&iso;hasParent"/>

```

```

      <owl:maxCardinality

```

```

rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>

```

```

    </owl:Restriction>

```

```

</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&iso;hasDomainClass"/>
    <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&iso;source"/>
    <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&iso;Merge">
  <rdfs:subClassOf rdf:resource="&iso;Element"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;mergeType"/>
      <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;hasTargetCollection"/>
      <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;hasSourceCollection"/>
      <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&iso;bidirectional"/>
      <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>

```

```

</rdfs:subClassOf>
<rdfs:comment xml:lang="en"
  >Class that defines the relations between the Attributes belongs to
  Collections&#39; instances. This class allows know how the collections are
  related by its attributes and how access the information source&#39;s
  elements.</rdfs:comment>
  <rdfs:comment xml:lang="pt"
    >Classe que define as rela&#231;&#245;es entre os attributtes
    pertencentes &#224;s diversas inst&#226;ncias de Collection. Essa classe
    permite saber como as collections est&#227;o relacionadas pelos seus atributos
    e como acessar ou navegar sobre os elementos da fonte de
    informa&#231;&#227;o.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="&iso;TextAttribute">
    <rdfs:subClassOf rdf:resource="&iso;Attribute"/>
  </owl:Class>
  <owl:Class rdf:about="&iso;TextCollection">
    <rdfs:subClassOf rdf:resource="&iso;Collection"/>
    <rdfs:comment xml:lang="en"
      >TextCollection is a type of Colelction that contains a set of textual
      entities. This class is used to represent a set of textual dabase&#39;s
      elements</rdfs:comment>
      <rdfs:comment xml:lang="pt"
        >TextCollection &#233; um tipo de Collection que cont&#233;m um
        conjunto de entidades textuais. Esta classe &#233; usada para representar um
        conjunto de elementos de uma base de dados textual</rdfs:comment>
      </owl:Class>
      <owl:Class rdf:about="&iso;XMLAttribute">
        <rdfs:subClassOf rdf:resource="&iso;Attribute"/>
      </owl:Class>
      <owl:Class rdf:about="&iso;XMLCollection">
        <rdfs:subClassOf rdf:resource="&iso;Collection"/>
      </owl:Class>
      <owl:Class rdf:about="&vio;Locale">
        <rdfs:subClassOf rdf:resource="&iso;Element"/>
        <rdfs:comment xml:lang="en"
          >Locale represents the visual aspects related to specific geographical,
          political, or cultural region</rdfs:comment>
          <rdfs:comment xml:lang="pt"
            >Locale representa os aspectos visuais relacionados a uma regi&#227;o
            geogr&#225;fica, pol&#237;tica ou cultural</rdfs:comment>
          </owl:Class>
          <owl:Class rdf:about="&rdf;Property"/>
          <owl:Class rdf:about="&rdf;Class"/>
          <owl:Class rdf:about="&owl;Thing"/>

```

```

<iso:AggregateFunction rdf:about="&iso;AVERAGE">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType rdf:datatype="&xsd:string">AVG</iso:functionType>
</iso:AggregateFunction>
<owl:Thing rdf:about="&iso;CONCAT">
  <rdf:type rdf:resource="&iso;ScalarFunction"/>
  <iso:functionType
rdf:datatype="&xsd:string">CONCAT</iso:functionType>
</owl:Thing>
<owl:Thing rdf:about="&iso;COUNTING">
  <rdf:type rdf:resource="&iso;AggregateFunction"/>
  <iso:functionType
rdf:datatype="&xsd:string">COUNT</iso:functionType>
</owl:Thing>
<owl:Thing rdf:about="&iso;COUNTING_DISTINCT">
  <rdf:type rdf:resource="&iso;AggregateFunction"/>
  <iso:functionType
rdf:datatype="&xsd:string">COUNT_DISTINCT</iso:functionType>
</owl:Thing>
<iso:AggregateFunction rdf:about="&iso;FREQUENCY">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType rdf:datatype="&xsd:string">FREQ</iso:functionType>
</iso:AggregateFunction>
<owl:Thing rdf:about="&iso;LENGTH">
  <rdf:type rdf:resource="&iso;ScalarFunction"/>
  <iso:functionType
rdf:datatype="&xsd:string">LENGTH</iso:functionType>
</owl:Thing>
<iso:ScalarFunction rdf:about="&iso;LOWER_CASE">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType
rdf:datatype="&xsd:string">LOWERCASE</iso:functionType>
</iso:ScalarFunction>
<owl:Thing rdf:about="&iso;MAXIMUM">
  <rdf:type rdf:resource="&iso;AggregateFunction"/>
  <iso:functionType rdf:datatype="&xsd:string">MAX</iso:functionType>
</owl:Thing>
<iso:AggregateFunction rdf:about="&iso;MINIMUM">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType rdf:datatype="&xsd:string">MIN</iso:functionType>
</iso:AggregateFunction>
<owl:Thing rdf:about="&iso;NONE">
  <rdf:type rdf:resource="&iso;AggregateFunction"/>
  <iso:functionType
rdf:datatype="&xsd:string">NONE</iso:functionType>

```

```

</owl:Thing>
<iso:ScalarFunction rdf:about="&iso;SUBSTRING">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType
rdf:datatype="&xsd:string">SUBSTRING</iso:functionType>
</iso:ScalarFunction>
<owl:Thing rdf:about="&iso;SUM">
  <rdf:type rdf:resource="&iso;AggregateFunction"/>
  <iso:functionType rdf:datatype="&xsd:string">SUM</iso:functionType>
</owl:Thing>
<iso:ScalarFunction rdf:about="&iso;TRIM">
  <rdf:type rdf:resource="&owl;Thing"/>
  <iso:functionType rdf:datatype="&xsd:string">TRIM</iso:functionType>
</iso:ScalarFunction>
</rdf:RDF >

```