

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

Dhiogo Cardoso da Silva

**UMA ARQUITETURA DE BUSINESS INTELLIGENCE PARA
PROCESSAMENTO ANALÍTICO BASEADO EM
TECNOLOGIAS SEMÂNTICAS E EM LINGUAGEM NATURAL**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia do Conhecimento.

Orientador: Prof. Dr. Denilson Sell

Florianópolis

2011

Catologação na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

S586a Silva, Dhiogo Cardoso da
Uma arquitetura de business intelligence para processamento analítico baseado em tecnologias semânticas e em linguagem natural [dissertação] / Dhiogo Cardoso da Silva ; orientador, Denilson Sell. – Florianópolis, SC, 2011.
161 p.: il., 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. Sistemas de consultas e respostas. 4. Tecnologias semânticas. I. Sell, Denilson. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento. III. Título.

CDU 659.2

Dhiogo Cardoso da Silva

**UMA ARQUITETURA DE BUSINESS INTELLIGENCE PARA
PROCESSAMENTO ANALÍTICO BASEADO EM
TECNOLOGIAS SEMÂNTICAS E EM LINGUAGEM NATURAL**

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia do Conhecimento, e aprovada em sua forma final pelo Programa Pós-Graduação em Engenharia e Gestão do Conhecimento.

Florianópolis, 17 de Fevereiro de 2011.

Prof. Paulo Maurício Selig, Dr.
Coordenador do Curso

Banca Examinadora:

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

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

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

Prof., Dr. José Leomar Todesco,
Universidade Federal de Santa Catarina

RESUMO

A necessidade de obtenção e uso de conhecimento para apoio à tomada de decisão motiva a convergência das novas gerações de *Business Intelligence* (BI) com os instrumentos da Engenharia do Conhecimento. Não obstante a aplicação de tecnologias semânticas e métodos de representação de conhecimento, as pesquisas de BI pouco exploram o uso de linguagem natural para a condução das análises. A metáfora de busca de informações conjecturada na Web Semântica revela-se como tendência para a área de BI. Assim, propõe-se uma arquitetura de BI em que a estratificação das informações estratégicas das fontes de dados corporativas é conduzida por meio da interpretação semântica de perguntas declaradas em linguagem natural. Esta arquitetura aproxima a área de BI da disciplina de Question Answering (QA) e dos formalismos oriundos da Web Semântica em uma abordagem interdisciplinar. Alguns recursos de representação de conhecimento, como ontologia, regras de inferência, padrões idiomáticos e heurísticas auxiliam os módulos funcionais da arquitetura na interpretação de perguntas e na obtenção de cubos OLAP. A demonstração da viabilidade da arquitetura é verificada em um estudo de caso relacionado ao domínio de C&T da Plataforma Lattes Institucional da UFSC. Uma interface analítica foi construída para permitir a entrada de perguntas em idioma português, a interação com o tomador de decisão para a resolução de ambigüidades e a visualização de *hipercubos*. Assim, tal como o modo de localização de informações já familiarizado por bilhões de usuários da Web, essa pesquisa proporciona um método inovador para auxiliar o processo decisório.

Palavras-chave: Business Intelligence. Question Answering. Tecnologias semânticas.

ABSTRACT

The need to obtain and use knowledge to support the decision making motivates the convergence of the new generations of Business Intelligence (BI) solutions with the Knowledge Engineering tools. Despite application of semantic technologies and methods of knowledge representation, BI research still lacks the use of natural language to conduct analysis. The metaphor of information searching conjectured on the Semantic Web is becoming a trend in the area of BI. Thus, a BI architecture is hereby proposed in which the gathering of strategic information from corporate data sources is driven by means of the semantic interpretation of natural language questions. This architecture brings to the BI area of the discipline of Question Answering (QA) and the Semantic Web formalisms through an interdisciplinary approach. Some resources of knowledge representation, such as ontology, inference rules, idiomatic patterns and heuristics aid the architecture's function modules with the interpretation of question and the return of the OLAP cube. The demonstration of the viability of this proposal is verified in a case study related to the domain of Science and Technology of the Plataforma Lattes Institucional of UFSC. An analytical interface was constructed to allow the entry of questions in the Portuguese language, the interaction with the decision maker to resolve ambiguities and the visualizing *hypercubes*. As well as the way millions of users search for information on the Web, this research provides an innovative method to aid in the decision making process.

Keywords: Business Intelligence. Question Answering. Semantic technologies.

LISTA DE FIGURAS

Figura 1 – Ilustração do método de pesquisa	30
Figura 2 - Arquitetura clássica de Business Intelligence.....	34
Figura 3 - Ilustração de um modelo dimensional e snowflaking.....	39
Figura 4 – Tipos de análises e ferramentas em proporcionalidade de uso	41
Figura 5 - Diagrama em camadas da Web Semântica.....	46
Figura 6 – Ilustração de uma arquitetura típica de Question Answering	57
Figura 7 - Arquitetura de Business Intelligence proposta	71
Figura 8 - Ilustração das possibilidades de caminhos para uma ontologia.	86
Figura 9 - Modelo tripla para persistência de conclusões de regras de inferência.....	100
Figura 10 – Modelo dimensional da Plataforma Lattes construído.....	105
Figura 11 - Ilustração da ontologia de domínio.....	108
Figura 12 - Ilustração da matriz de caminhos obtida da ontologia de domínio	110
Figura 13 - Ilustração do mapeamento parcial da Ontologia BI.....	112
Figura 14 - Arquitetura tecnológica do protótipo.....	118
Figura 15 - Ilustração do protótipo da interface analítica.....	121
Figura 16 - Ilustração de resolução de ambigüidades.	126
Figura 17 - Consulta SQL e Cubo OLAP projetado na interface analítica.	127
Figura 18 - Visualização do resultado da pergunta com aplicação de inferência.....	130
Figura 19 - Consulta e resultado gerados para uma pergunta com função.	134

LISTA DE QUADROS

Quadro 1 - Exemplos de regras de inferência	51
Quadro 2 - Regra de Inferência que define a relação Formando entre pessoa e instituição.....	84
Quadro 3 - Regra de inferência para identificar os alunos com baixo desempenho.....	99
Quadro 4 - Sintaxe de função para associação entre terminologias e cálculos.....	111
Quadro 5 - Consulta SQL gerada a partir da pergunta do domínio de C&T.	124
Quadro 6 - Regra de inferência para explicitar o conceito Calouro. ...	128
Quadro 7 - Consulta gerada no processo de inferência on-the-fly.	131

LISTA DE TABELAS

Tabela 1 - Diferenciação entre os sistemas e tipos de dados operacionais e analíticos.....	32
Tabela 2 - Classificação dos elementos textuais da pergunta geradas pelo Analisador Lingüístico	76
Tabela 3 - Tipos de stop-words para a tradução OLAP	91
Tabela 4 - Descrição do modelo dimensional do DW.....	106
Tabela 5 - Padrões e heurísticas utilizados.....	114
Tabela 6 - Lista de stop-words utilizada conforme o tipo	116

LISTA DE ABREVIATURAS E SIGLAS

BI - Business Intelligence
CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
CRM - Customer Relationship Management
C&T – Ciência & Tecnologia
DM – Data Mart
DW - Data Warehouse
EI – Extração de Informação
EIS - Enterprise Information Systems
ERP - Enterprise Resource Planning
ETL – Extraction, Transformation and Loading
IRI – International Resource Identifier
MER - Modelo Entidade-Relacionamento
MUSING – MUlti-industry, Semantic-based next generation business INtelliGence
NER – Named-Entity Recognition
OLAP - On-Line Analytical Processing
OLTP - On-Line Transaction Processing
OWL - Web Ontology Language
PLI - Plataforma Lattes Institucional
POS-Tagging – Part-of-Speech Tagging
QA – Question Answering
RDF – Resource Description Framework
RI – Recuperação de Informação
SBI – Semantic Business Intelligence
SQL – Structured Query Language
SWRL – Semantic Web Rule Language
URI – Uniform Resource Identifier

SUMÁRIO

1	INTRODUÇÃO.....	17
1.1	PROBLEMÁTICA DA PESQUISA	20
1.2	OBJETIVOS	22
1.2.3	Objetivo Geral.....	22
1.2.4	Objetivos Específicos	23
1.3	JUSTIFICATIVA.....	23
1.4	ADERÊNCIA AO OBJETO DE PESQUISA DO PROGRAMA	25
1.5	DELIMITAÇÃO DE ESCOPO	26
1.6	MÉTODO DE PESQUISA	28
1.7	ORGANIZAÇÃO DA DISSERTAÇÃO	30
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	BUSINESS INTELLIGENCE.....	31
2.1.1	Data Warehouse e Data Mart	35
2.1.2	Extração, Transformação e Carga	36
2.1.3	Modelagem dimensional.....	38
2.1.4	Área de Apresentação.....	40
2.1.5	OLAP.....	42
2.1.6	Metadados.....	43
2.2	TECNOLOGIAS SEMÂNTICAS.....	45
2.2.1	Web Semântica.....	45
2.2.2	Ontologia.....	48
2.2.3	Raciocínio e regras de inferência	50
2.2.4	Iniciativas de Business Intelligence baseadas em tecnologias semânticas.....	52
2.3	QUESTION ANSWERING	54
2.3.1	Recuperação e Extração de Informação	55
2.3.2	Arquitetura típica de Question Answering.....	56
2.3.3	Tipos de Question Answering	60
2.3.4	Iniciativas de Question Answering baseadas em tecnologias semânticas.....	62
2.3.5	Iniciativas de Question Answering no contexto de Business Intelligence.....	65
3	ARQUITETURA PROPOSTA	69
3.1	VISÃO GERAL	69
3.2	MODELO E BASE DE CONHECIMENTO	72
3.3	ANALISADOR LINGÜÍSTICO	75
3.4	REFORMULADOR.....	79
3.4.1	Reformulação por hierarquia de classes e sinônimos	80
3.4.2	Reformulação por regras de inferência.....	83
3.5	MOTOR DE BUSCA POR SIMILARIDADE	85
3.6	TRADUTOR OLAP.....	89

3.7	GERENCIADOR DE CONSULTAS.....	94
3.7.1	Consultas na abordagem on-the-fly.....	95
3.7.2	Consultas na abordagem in-batch.....	96
3.8	GERENCIADOR DE ONTOLOGIAS.....	97
3.9	MECANISMO DE INFERÊNCIA.....	98
3.10	DATA WAREHOUSE.....	100
4	DEMONSTRAÇÃO DA VIABILIDADE DA	
	ARQUITETURA	103
4.1	INTRODUÇÃO AO DOMÍNIO DE APLICAÇÃO DO PROTÓTIPO	103
4.2	DATA WAREHOUSE DA PLATAFORMA LATTES	104
4.3	MODELO E BASE DE CONHECIMENTO UTILIZADO.....	107
4.3.1	Ontologia de Domínio.....	107
4.3.2	Regras de Inferência e Funções	111
4.3.3	Ontologia BI	112
4.3.4	Padrões léxico-sintáticos e Heurísticas.....	114
4.4	CONSTRUÇÃO DOS MÓDULOS FUNCIONAIS DA ARQUITETURA	117
4.5	INTERPRETAÇÃO DE PERGUNTAS E RESULTADOS OBTIDOS	120
4.5.1	Pergunta com conceitos do domínio de C&T	121
4.5.2	Pergunta com ambigüidades de conceitos e caminhos.....	124
4.5.3	Pergunta com inferência	128
4.5.4	Pergunta com função.....	132
4.6	AVALIAÇÃO SOBRE OS RESULTADOS OBTIDOS NO DOMÍNIO DE C&T.....	135
4.7	DISCUSSÃO SOBRE OS TRABALHOS RELACIONADOS	138
5	CONCLUSÃO.....	143
5.1	LIMITAÇÕES E TRABALHOS FUTUROS.....	145
	REFERÊNCIAS.....	149
	APÊNDICE A – Especificação das regras de inferência	159
	APÊNDICE B – Especificação das funções	161

1 INTRODUÇÃO

A revolução causada pela era do conhecimento traz novas oportunidades para gestão organizacional. Reconhece-se que o conhecimento é um importante insumo para a inovação e para a economia (SMITH, 2002). Nessa nova economia, para que as organizações obtenham sucesso é essencial cada vez mais a aquisição de informações estratégicas sobre o seu negócio para a tomada de ações. O mercado competitivo faz com que as empresas aprimorem suas competências e seus processos para agregar valor aos seus serviços e produtos. Conseqüentemente, o conhecimento revela-se como um dos principais ativos nas organizações (RAO, 2005; ROTHBERG; ERICKSON, 2005). Desde então, o termo mão-de-obra, na qual está associado às características da era industrial, perde espaço para uma nova abordagem, em que as atividades intensivas em conhecimento são necessárias. Nesse cenário, a gerência organizacional possui grandes desafios relacionados à obtenção, uso e gestão do conhecimento adquirido e, dessa forma, o papel da Engenharia do Conhecimento, como meio auxiliador, se faz necessário (SCHREIBER et. al., 2002).

Com a mudança de enfoque trazida pela era do conhecimento, a Engenharia do Conhecimento, na qual preconiza que o conhecimento pode ser adquirido, modelado e codificado em sistemas de conhecimento, tem emergido para atendimento às demandas da gestão organizacional (STUDER; BENJAMINS; FENSEL, 1998; RAO, 2005). Hoje, além dos sistemas transacionais e sistemas de informação, há necessidade também de instrumentos de coleta e localização de informações que colaborem na transformação dessas informações em conhecimento, de modo a embasar e efetivar a tomada de decisões das organizações (CODY, et. al., 2002). Com isso, tais instrumentos e tecnologias ganham importância e contribuem para a evolução dos sistemas de apoio à decisão e conseqüentemente, para o crescimento da área de inteligência de negócio – Business Intelligence (BI).

Dentre os processos de gestão de conhecimento, a área de BI está atrelada principalmente aos relacionados à aquisição, criação e uso de conhecimento (RAO, 2005). Isto porque, fornece meios para que o conhecimento seja gerado, tornado explícito e ainda aplicado para a tomada de ações dentro das organizações. BI consiste em métodos, processos, ferramentas e tecnologias necessárias para transformar dados em informação e, por sua vez, informação em conhecimento a fim de apoiar o planejamento e a direção efetiva das atividades de negócios das

organizações (ECKERSON, 2006). Para Kimball e Ross (2002), BI pode descrever em âmbito geral os recursos informacionais internos e externos da organização necessários para propiciar melhores decisões sobre o negócio.

Observa-se na literatura que as arquiteturas tradicionais de BI contemplam ferramentas e processos tais como, o processo ETL (*Extraction, Transformation and Loading*); repositórios de dados como Data Warehouse (DW) e Data Marts (DM); ferramentas de análises, como ferramentas OLAP (On-line Analytical Processing) e; uma camada de metadados para orquestrar todos os componentes e processos (INMON, 2005; KIMBALL; ROSS, 2002). Além da dificuldade de gerir a grande quantidade de dados que diariamente é produzida interna e externamente nas organizações, a necessidade por informações estratégicas exige ainda mais o uso de tecnologias analíticas, tais como os componentes das arquiteturas de BI supracitados. Assim, as soluções de BI desempenham um papel fundamental e indispensável para os processos analíticos.

Apesar dos esforços para oferecer um ambiente propício para coleta de informações, para análises e para apoio à tomada de decisão, verifica-se que as soluções de BI ainda apresentam algumas limitações referentes às novas necessidades analíticas. Algumas pesquisas já apontam as lacunas e os principais problemas das arquiteturas clássicas de BI (SELL, 2006; CODY et. al., 2002; BÖHRINGER et. al., 2010), que além da falta de integração semântica entre fontes de dados heterogêneas, destacam-se: a falta do uso da semântica do negócio para guiar às análises e; a ausência de mecanismos de inferência para estender as funcionalidades exploratórias e explicitar novas informações ao gestor. Tais pesquisas já demonstram evoluções que direcionam para estratégias de aplicação de recursos semânticos e formas de representação e utilização de conhecimento. Segundo essas pesquisas, esses recursos podem oferecer maior expressividade e poder de raciocínio para assistir ao processo decisório no ambiente de negócios (LAVBIC; VASILECAS; RUPNIK, 2010).

Ainda que essas limitações semânticas sejam solúveis, sabe-se que a exploração das fontes de dados e a obtenção de informação nas soluções de BI se dão geralmente por meio de operações de consulta, denominadas operações OLAP, tais como *slice and dice*, *drill-down*, *drill-up*, etc. (KIMBALL; ROSS, 2002; INMON, 2005). Na prática, essas operações costumam ser guiadas pelos estímulos do tomador de decisão sem qualquer interação ou auxílio para condução de análises ou interpretação de resultados. Outrossim, são consideradas complexas à

medida que requerem conhecimento técnico e esforço para efetuar consultas e navegar sobre o conteúdo do DW (ZENG, et. al., 2006).

O público usuário das ferramentas analíticas comumente é constituído por especialistas do negócio que são preparados em longas sessões de capacitação e treinamento, envolvendo a apresentação da estrutura das fontes de dados da organização e o uso dos recursos padrão de uma ferramenta OLAP específica de um fabricante (THOMSEN, 2002; CONLON; CONLON; JAMES, 2004). Portanto, além de se preocupar com a interpretação de resultados e a tomada de decisão, torna-se dificultoso para o gestor ter que compreender também as formas de manuseio e operação das distintas ferramentas OLAP disponíveis no mercado (ZENG, et. al., 2006). Reduzir a curva de aprendizado tornando as soluções de BI mais naturais, simplistas e fáceis de usar e ao mesmo tempo acessíveis a todas as pessoas da organização tem sido discutido pela comunidade acadêmica e comercial. Com isso, novas tendências e terminologias da área como *BI for Masses* e *Pervasive BI* ganharam notoriedade nos últimos anos (COMPUTER WEEKLY, 2002; SWOYER, 2010).

É fato que a simplicidade das buscas na Web atual contribui continuamente para o aumento de usuários e para o crescimento de sua popularidade. A facilidade de uso nessas interfaces de busca permite que com poucas palavras informadas em texto livre seja possível encontrar quase todo tipo de conteúdo de modo rápido e ubíquo. Pelo seu modo intuitivo e natural com que os mecanismos de buscas provêem acesso à informação para pessoas de praticamente todas as idades, a mesma metáfora de busca na Web deve ser considerada para as próximas gerações de soluções de BI. A tendência para o futuro da área de BI leva em conta a sua aproximação com os recursos e serviços da Web, tanto no uso de fontes heterogêneas quanto no modo de localizar informações (HOWSON, 2008; BÖHRINGER, et. al., 2010).

Embora sejam fáceis e ágeis, as buscas na Web são sensíveis às palavras-chave, podem ocasionar baixa precisão nas respostas e há possibilidade de poucas ocorrências encontradas no resultado (ANTONIOU, HARMELEN, 2008). Com os ideais da Web Semântica, inúmeras pesquisas lançaram luz sobre a área de recuperação e extração de informação, culminando em novas técnicas e abordagens baseadas em ontologias e raciocínio. Além de melhorar a precisão das buscas, esses métodos visam adicionar significado ao conteúdo da Web para colaborar de modo automático e mais inteligente com as pessoas (BERNEERS-LEE; HENDLER; LASSILA, 2001).

Percebe-se, então, que na conjugação entre as novas pesquisas de BI e os anseios da Web Semântica há focos de estudo que podem ser abordados mais profundamente. Para atender aos diferentes *stakeholders*, torna-se necessário às ferramentas analíticas contar com estratégias para a representação do conhecimento do negócio e mecanismos que possibilitem o uso intensivo deste conhecimento nas atividades de exploração das fontes de dados. Do mesmo modo que a Web Semântica prevê formas ágeis e interfaces de navegação com alta expressividade semântica para localizar o conteúdo relevante na Internet, as arquiteturas de BI devem também fazer uso desses recursos para dar suporte ao processamento analítico. Falta às soluções de BI o uso de métodos efetivos de exploração de conteúdo tais como os já familiarizados e consolidados pelos bilhões de usuários da Web atual, ainda assim, sem perder o potencial conjecturado pela Web Semântica (SMALLTREE, 2006).

Ainda com a necessidade de integração entre as áreas de estudo da Engenharia do Conhecimento, como a recuperação e a extração de conhecimento, o foco desta pesquisa concentra-se no modo como as arquiteturas de BI são projetadas para a realização de consultas sobre os repositórios da organização. Na visão de mundo desta pesquisa, as mesmas operações OLAP, que hoje são efetuadas de modo distinto pelo grande número de ferramentas de análises, podem ser realizadas de um modo único e análogo como é feito pelo usuário Web, sem grande esforço e sem necessidade de treinamentos a priori.

O esforço e o custo de treinamento em ferramentas analíticas levam em conta vários fatores como: o número potencial de usuários; a disponibilidade de tempo desses usuários; a complexidade das interações com as ferramentas; as habilidades de cada usuário e; ainda a relação do usuário com a organização. Para reduzir esses custos, o uso de linguagem natural é considerado um dos meios mais adequados e viáveis (CONLON; CONLON; JAMES, 2004). Portanto, a capacidade de se expressar por meio de linguagem natural deve ser introduzida nas novas arquiteturas de BI e é o objeto de estudo deste trabalho. A seção seguir detalha o problema da pesquisa.

1.1 PROBLEMÁTICA DA PESQUISA

Não obstante o crescente número de pesquisas direcionadas para a integração entre as disciplinas de BI, Web Semântica e Processamento

de Linguagem Natural a abordagem adotada por este trabalho ainda é pouco explorada. Em sua grande maioria, as pesquisas que inter-relacionam essas áreas de conhecimento são destinadas a integrar fontes de dados estruturadas e não estruturadas a partir dos dados internos e externos da organização para suportar a inteligência competitiva. Normalmente, essas pesquisas de BI são baseadas em recursos de representação de conhecimento, tais como ontologia, para que o conteúdo das bases de dados textuais seja extraído, armazenado e combinado com os repositórios estruturados da organização (CODY, et. al., 2002; CHUNG; CHEN; NUNAMAKER, 2002; SAGGION et. al., 2007; BENEVENTANO, et. al., 2007). O processamento de linguagem natural dessas pesquisas está centrado na recuperação e extração de informação para preparação de repositórios de dados integrados e em mecanismos de buscas baseados em palavras-chave. Essas pesquisas não dão enfoque a meios mais expressivos de consulta para obtenção de informações estratégicas por meio de linguagem natural.

A problemática da pesquisa foca nos métodos de engenharia do conhecimento para integrar as funcionalidades semânticas e analíticas das novas tendências de BI com as áreas de processamento de linguagem natural. Neste trabalho as análises submetidas às fontes de dados da organização, em vez de serem guiadas por meio das operações OLAP convencionais, são efetuadas por meio da interpretação semântica de uma pergunta expressa em linguagem natural. Isto é, através de uma pergunta declarada na linguagem habitual do usuário, na qual os conceitos e terminologias do negócio são expressos de forma descritiva e livre, visa-se à obtenção do cubo OLAP¹. Logo, surge o problema da pesquisa: *É possível realizar consultas multidimensionais por meio da interpretação semântica de perguntas expressas em linguagem natural em uma arquitetura de Business Intelligence?*

A condução de consultas OLAP por meio da interpretação de perguntas para suporte à tomada de decisão constitui-se em uma linha de pesquisa em estado da arte. Diferentemente das estratégias de buscas por palavras-chave, a resolução do problema baseia-se na criação de uma arquitetura de BI que alinha os métodos da Engenharia do Conhecimento com aqueles mais próximos da disciplina de *Question*

¹ Cubo OLAP – nome da estrutura dimensional criada a partir do processamento multidimensional ou OLAP sobre a fonte de dados, na qual as informações podem ser combinadas, sumarizadas ou mesmo detalhadas para apoio à tomada de decisão (KIMBALL; ROSS, 2002)

Answering (QA). Esta disciplina caracteriza-se por adotar meios mais significativos de uso da linguagem natural ou de perguntas para o retorno de uma resposta única e mais precisa. Ao longo da fundamentação teórica, essa subárea do Processamento de Linguagem Natural é detalhada. As seções posteriores descrevem os objetivos deste trabalho.

1.2 OBJETIVOS

1.2.3 Objetivo Geral

No contexto da Engenharia do Conhecimento, como objetivo geral desta proposta visa-se à criação de uma arquitetura de apoio à tomada de decisão capaz de realizar o processamento analítico por meio da interpretação semântica de perguntas expressas em linguagem natural.

A interpretação semântica da pergunta consiste na tarefa em que todas as relações entre os conceitos e as terminologias do domínio informados livremente pelo tomador de decisão são identificados e formalizados em uma estrutura que representa o significado da pergunta. A interpretação semântica completa-se após essa estrutura, que modela o significado da pergunta, ser transformada ou traduzida em operações OLAP. Para subsidiar essa tarefa, este trabalho adota métodos e recursos baseados em conhecimento como o uso de ontologias, aplicação de padrões idiomáticos, heurísticas e inferências.

Entende-se por processamento analítico a tarefa de aplicar as operações OLAP sobre o *data warehouse*, tais como filtros e agrupamentos de conteúdo, combinação entre as dimensões e tabelas de fato, com o objetivo de recuperar as informações estratégicas sumarizadas para o tomador de decisão. Deste modo, essa tarefa estabelece as medidas, os fatos quantitativos e os atributos de dimensão que devem ser projetados para a criação do cubo OLAP. Embora a problemática da pesquisa permeie os métodos de processamento de linguagem natural e *Question Answering*, as respostas obtidas são sumarizações das informações oriundas do data warehouse. Portanto, o retorno da informação é destinado a responder às perguntas que requeram quantificações e estratificações de conteúdo.

1.2.4 Objetivos Específicos

Os objetivos específicos são:

- Analisar e identificar as tecnologias semânticas, os componentes ou subsistemas das arquiteturas relacionadas ao contexto de *Business Intelligence* e Processamento de Linguagem Natural para compor a arquitetura;
- Identificar as estratégias para aplicar o conhecimento do domínio da organização no apoio à interpretação semântica de perguntas informadas em linguagem natural;
- Permitir o uso de inferências na arquitetura a fim de apoiar a descoberta de conhecimento a partir das fontes de dados da organização;
- Identificar os métodos para a execução de operações OLAP sobre as fontes de dados para obtenção de quantificações e informações sumarizadas a partir da pergunta já interpretada;
- Desenvolver um protótipo para demonstrar a viabilidade da arquitetura;

1.3 JUSTIFICATIVA

Os métodos e ferramentas de BI previnem de certa forma a perda de conhecimento nas organizações. Isto porque, desde as primeiras gerações, as soluções de BI oferecem uma base rica em informações para que as análises e ações sejam efetuadas de maneira fundamentada no vasto conteúdo acumulado ao longo do tempo (PONNIAH, 2001). Assim, dada a contribuição que essas soluções provêm para as empresas, não é estranho notar que o setor de BI possui a maior taxa de crescimento comparada a todos os outros segmentos de mercado. Segundo revela a pesquisa da IDC Brasil (2010), em 2009 a área de BI apresentou um investimento de 504 milhões de dólares só na América Latina, sendo que no Brasil esse valor foi de 251 milhões de dólares. Já para 2010, a IDC espera que a América Latina cresça em torno de 12% e o mercado brasileiro aproximadamente 14%.

Embora os valores tangíveis sejam claramente percebidos nas organizações, como o retorno sobre os investimentos; aumento dos lucros e economia de tempo/custo; esses ainda não são os principais

fatores que impulsionam a grande procura por ferramentas analíticas. A importância e os benefícios oferecidos pelas soluções de BI são em sua maioria intangíveis e muitas vezes são difíceis de justificar em termos de custo. Um estudo do TDWI (ECKERSON, 2003) indica que os benefícios intangíveis tais como a melhoria na qualidade de planejamentos e estratégias; melhores táticas e decisões; mais eficiência nos processos organizacionais; satisfação de clientes e empregados são mais esperados e complexos de serem alcançados em proporção aos benefícios tangíveis citados.

As tecnologias de BI são comumente direcionadas aos gestores da organização na condução de análises sobre o negócio. A necessidade de prover informações para tomada de decisões, monitoração e execução de ações nos diferentes níveis da organização dá também oportunidade de crescimento para outras formas de BI, como o *Operational BI* (HOWSON, 2008; SWOYER, 2010). Porém, dentre os principais motivos por que é difícil oferecer instrumentos de BI a todas as pessoas da organização está a dificuldade de uso e controle sobre as ferramentas analíticas. O uso de linguagem natural, proposto pela disciplina de *Question Answering*, consiste num dos métodos mais convenientes e intuitivos para acesso à informação (KATZ; LIN; FELSHIN, 2001). Dessa forma, a aproximação da área de BI com esses métodos de localização de informações revela-se como tendência cada vez mais necessária, dada à facilidade de obter informações de modo simples e mais natural (ECKERSON, 2010; COMPUTER WEEKLY, 2002).

Em matéria divulgada pela InformationWeek (HENSCHEN, 2008), o uso de linguagens naturais pode propiciar aos gestores mais facilidade para realização de consultas *ad hoc*² sem a necessidade de elevados custos em treinamento de uma solução analítica específica. Pela fácil usabilidade, tais como as já trazidas pelas ferramentas de busca na Web, na qual as consultas podem ser livremente declaradas, toda a logística de exploração das fontes de dados torna-se mais transparente ao usuário. Cabe a ele informar apenas o conhecimento o qual deseja extrair utilizando as terminologias na linguagem que está habituado. Eckerson (2003), diretor *do The Data Warehouse Institute*, afirma que historicamente a maioria das ferramentas de análise atende a

² Consulta *ad hoc* – Segundo Inmon (2005), são os acessos casuais que manipulam dados conforme parâmetros ainda não utilizados, geralmente executados de forma iterativa e heurística.

um público reduzido constituído de usuários especialistas. Devido a isso, muitas empresas têm dificuldade em atrair clientes para suas soluções e acabam perdendo mercado. Assim, o uso de linguagem natural, como objeto de estudo deste trabalho, é justificado também como contribuição para as próximas gerações de BI.

1.4 ADERÊNCIA AO OBJETO DE PESQUISA DO PROGRAMA

De acordo com Studer, Benjamins e Fensel (1998), a nova Engenharia do Conhecimento pressupõe a existência de processos de modelagem de conhecimento em sistemas, nos quais devem ser aplicados em atendimento às demandas da gestão. Os sistemas de conhecimento são destinados a apoiar as decisões de modo mais rápido e com maior qualidade, e também aumentar a produtividade das organizações. Dentre as principais distinções entre outros tipos de sistemas de software é que nos sistemas de conhecimento assume-se que há alguma representação explícita de conhecimento inclusa no sistema. Daí a necessidade por técnicas especiais conjuntas para modelagem de conhecimento e de sistemas, tal como é empregado neste estudo (SCHREIBER, et. al., 2002).

Sendo assim, dada a interdisciplinaridade da pesquisa, na qual se fundamenta em áreas e métodos tais como *Natural Language Processing*, *Question Answering*, *Gestão Estratégica*, *Business Intelligence* e *Web Semântica*, considera-se este projeto aderente a área da Engenharia de Conhecimento pelo seu âmbito de formalização e codificação de conhecimento na arquitetura proposta. Além do uso de ontologia, esta arquitetura viabiliza a representação do conhecimento por meio de especificação de padrões e heurísticas baseados no idioma e ainda, regras de negócio para aplicação de inferências no processo decisório.

Além disso, este trabalho está relacionado à linha de pesquisa do programa denominada *Engenharia de Conhecimento aplicada às organizações*, já que visa a auxiliar os processos de gestão de conhecimento relacionados à aquisição e uso de conhecimento no apoio à tomada de decisão.

1.5 DELIMITAÇÃO DE ESCOPO

Devido à complexidade do tema, alguns aspectos resultantes da combinação das áreas de *Question Answering* e BI não são totalmente aprofundados e fogem ao escopo deste trabalho. Esses aspectos são detalhados a seguir:

- **Quanto ao perfil do usuário:** no contexto desta pesquisa, considerar o perfil do usuário implica em afirmar que uma mesma pergunta será tratada de modo distinto e possivelmente terá uma resposta diferente para cada usuário. Embora seja uma característica importante, a arquitetura proposta não visa à coleta de informação a respeito do usuário para identificar perfis. A semântica de uma pergunta será a mesma dentro do domínio tratado independente do perfil do usuário, e por isso não afeta o método de interpretação e retorno de informações. Assim, como o método único de consulta, não é escopo deste trabalho prover meios diferenciados de visualização de informações conforme o perfil do usuário.
- **Quanto à integração de fontes de dados heterogêneas:** ainda que se reconheça o grande valor gerado pelo cruzamento de informações dispersas em fontes estruturadas e não estruturadas, não é objetivo desta pesquisa resolver o problema da integração de fontes de dados heterogêneas. A arquitetura delimita-se a aplicação de consultas diretamente sobre o Data Warehouse ou Data Marts da organização, sem estender para bases textuais. Assim, este trabalho propõe-se a consumir a informação anteriormente produzida por processos ETL. Logo, cabe a equipe de especialistas da organização integrar as informações oriundas de fontes de dados textuais ou mesmo estruturadas no DW.
- **Quanto à sensibilidade ao idioma:** o processamento da linguagem natural é sem dúvida uma tarefa complexa que está obtendo grandes avanços pela comunidade acadêmica. Cada idioma ainda que possua variantes quanto à cultura do povo e ao modo de escrita, normalmente apresenta alguns padrões lingüísticos que auxiliam a sua interpretação automática ou semi-automática. Assim, boa parte das pesquisas utiliza técnicas de descoberta de aspectos léxicos, sintáticos e semânticos para explicitar modelos e criar bases de

conhecimento para processamento de linguagem natural. A arquitetura proposta não restringe quanto ao uso de um idioma específico, uma vez que a base e os modelos de representação de conhecimento podem ser preparados e adaptados para atender aos conceitos e termos da organização. A arquitetura prevê o uso de ontologia de domínio, dicionários de sinônimos, configuração de heurísticas e outras formas para a modelagem da base de conhecimento e explicitar padrões idiomáticos. Porém, os módulos da arquitetura foram baseados em estudos que utilizam o sistema de escrita do alfabeto latino, abrangendo principalmente idiomas como o inglês, italiano, português, espanhol e francês. Logo, a extensão para outros sistemas de escritas não são contemplados nesse trabalho, embora a arquitetura possa dar suporte.

- **Quanto à interação com usuário e processo de desambiguação:** durante a fase de interpretação da pergunta, na qual é descrita em detalhes no capítulo 3, é comum que um ou mais termos tenham mais de um significado no contexto. A resolução de ambigüidades não é enfoque deste trabalho. Quando uma ambigüidade não é resolvida pelos módulos da arquitetura, o usuário sempre deve intervir em um processo de desambiguação. Para tal, presume-se que a ferramenta OLAP, além de possibilitar a entrada da pergunta, possa interagir com o tomador de decisão até não haver ambigüidades. Também não faz parte do escopo da arquitetura propor mecanismos para aprendizagem dos conceitos e terminologias que já passaram pelo processo de desambiguação.
- **Quanto ao tipo de pergunta e resposta:** nesta proposta, todas as perguntas são interpretadas para a posterior construção de consultas multidimensionais. As perguntas devem ser direcionadas na arquitetura para a obtenção de informações sumarizadas e agrupadas conforme o conteúdo do *data warehouse*. Diferentemente da área de *QA*, que trata de perguntas pontuais e muitas vezes com exigências de respostas detalhadas, as perguntas devem ser direcionadas para o contexto de BI e respostas sob o formato de cubo OLAP. Isto é, as perguntas servem como guia para mensuração, resumo ou estratificação de informações e, portanto, não são destinadas a recuperar relatórios operacionais detalhados.

- **Quanto à construção e manutenção da base de conhecimento:** para acompanhar as próprias evoluções da linguagem e também as mudanças de regras de negócio e incorporação de novos jargões e conceitos do domínio, a base de conhecimento deve sofrer manutenções ao longo do tempo. Este trabalho não dá enfoque nos métodos de engenharia de ontologias. Dessa forma, outros trabalhos que usam de meios automáticos ou semi-automáticos devem ser agregados à arquitetura e auxiliar o engenheiro do conhecimento nesta tarefa.

1.6 MÉTODO DE PESQUISA

Os passos para a consolidação desta pesquisa foram executados em ciclos iterativos semelhante aos métodos espirais. Considera-se esse método mais aderente à pesquisa porque permite avaliar as evoluções incrementais do trabalho e identificar os pontos de melhoria com antecedência, tal como sugerido por Schreiber (et. al., 2002). Além disso, esse método possibilita que lições aprendidas sejam obtidas já nas primeiras iterações do ciclo do projeto, relacionando os aspectos teóricos e práticos das áreas de conhecimento envolvidas (*Question Answering e Business Intelligence*). Levando em conta a delimitação de escopo deste trabalho, os passos iterativos são enumerados a seguir:

- 1) **Análise das arquiteturas de BI** - Há inúmeras possibilidades de arquiteturas de BI na literatura que podem ser utilizadas como referência para concepção dos componentes da arquitetura proposta. Por isso, este trabalho restringe-se à análise sobre os trabalhos que levam em conta o uso de tecnologias semânticas, como as seguintes características descritas abaixo:
 - a. Abordagens de BI que incorporam formas de representação explícita de conhecimento baseadas em ontologias para a exploração de fontes de dados;
 - b. Iniciativas que se beneficiam da aplicação de regras de inferência e raciocínio a partir de bases de conhecimento a fim de apoiar ao processo decisório.
- 2) **Análise das arquiteturas de Question Answering e de Processamento de Linguagem Natural** – tal como no passo 1, uma análise sobre as arquiteturas de QA também é efetuada. Neste caso foca-se nos estudos que utilizam ontologias como modelo para a interpretação das perguntas em linguagem natural. Tratam-se aqui

os trabalhos que codificam e formalizam a semântica da pergunta na estruturas de representação de conhecimento fornecida pelo modelo de ontologia de domínio. Isto é, identificam os elementos textuais da pergunta na forma de classes, relacionamentos, propriedades e instâncias de classes. Nessa etapa compreende também o estudo de métodos, modelos, tarefas e algoritmos utilizados por essas soluções.

- 3) **Análise das abordagens que integram o uso de linguagem natural e BI** – Consiste em analisar as pesquisas e as arquiteturas relacionadas a este trabalho que já exploram as tecnologias semânticas para unir as áreas de QA e BI em uma única solução.
- 4) **Engenharia da arquitetura propriamente dita** – Compreende a etapa de engenhar a arquitetura e levantar os métodos e tarefas a serem aplicados por cada componente identificado. Tal etapa, que é evoluída no decorrer dos ciclos iterativos, fundamenta-se nos passos descritos anteriormente em atendimento ao objetivo geral do trabalho.
- 5) **Desenvolvimento de um protótipo da arquitetura** – Nesta fase uma instância da arquitetura é construída e aplicada sobre um DW específico. Essa fase compreende a construção das ontologias e base de conhecimento conforme o domínio tratado pelo DW; o uso de ferramentas e de frameworks para a construção dos módulos da arquitetura e; a adaptação de uma ferramenta OLAP para interação com o tomador de decisão. Neste método, essa etapa consiste também na identificação de lacunas, pontos de melhoria e limitações do modelo proposto que são aperfeiçoados à medida que as etapas avançam dentro do ciclo em espiral.

A Figura 1 a seguir demonstra o ciclo em espiral com o conjunto de passos adotados por este trabalho.

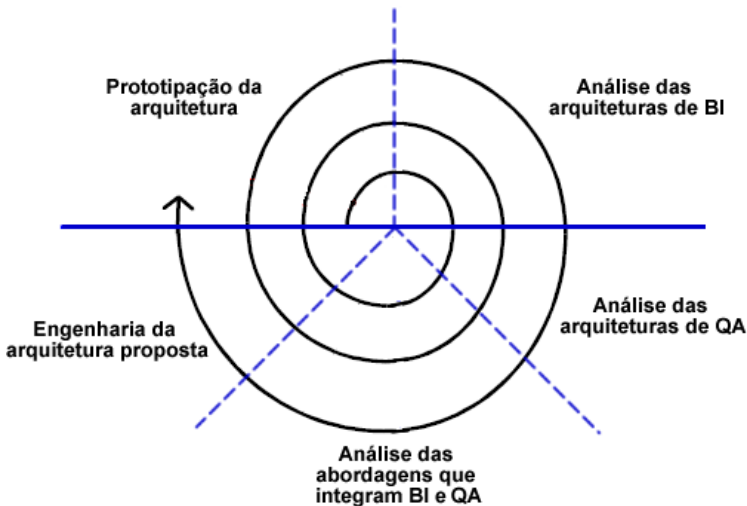


Figura 1 – Ilustração do método de pesquisa

1.7 ORGANIZAÇÃO DA DISSERTAÇÃO

Após este primeiro capítulo introdutório que contempla a problemática, objetivos, justificativa, delimitação de escopo e o método de pesquisa adotado, o trabalho organiza-se em:

- Capítulo 2 (Fundamentação Teórica) – trata dos fundamentos teóricos e também de pesquisas aplicadas das soluções de engenharia do conhecimento específicas das áreas de *Business Intelligence* e *Question Answering*. Esse capítulo apresenta ainda como as tecnologias semânticas inspiradas na Web Semântica auxiliam ambas as áreas de conhecimento e embasam este trabalho.
- Capítulo 3 (Arquitetura proposta) – apresenta a interação entre os componentes da arquitetura de Business Intelligence proposta;
- Capítulo 4 (Demonstração da viabilidade da arquitetura) – esse capítulo descreve uma instância da arquitetura aplicada no contexto de C&T da Plataforma Lattes Institucional da UFSC.
- Capítulo 5 (Conclusões) - por fim, esse último capítulo apresenta as conclusões, problemas encontrados e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Dada a grande rivalidade mercadológica, manter-se estrategicamente bem posicionado e competitivo em sua área de negócio é um dos grandes objetivos da gestão organizacional. Neste contexto, o conhecimento é considerado um dos principais ativos, na qual deve ser aplicado para interpretação de informações e para a tomada de decisões corretas na condução do sucesso da organização (RAO, 2005; ROTHBERG; ERICKSON, 2005). Evidentemente, a aquisição e o uso de conhecimento são atividades complexas e em face ao grande volume de informação disponível, normalmente necessitam de apoio de tecnologias e de sistemas de suporte às análises da área de Inteligência de Negócio (em inglês *Business Intelligence* - BI).

As seções a seguir descrevem a área de BI sob duas perspectivas. Primeiramente, a visão clássica das abordagens de BI é apresentada com os seus principais componentes e processos contidos nas arquiteturas tradicionais. Logo após, são descritas as abordagens que aproximam BI das tecnologias semânticas e do uso de linguagem natural.

2.1 BUSINESS INTELLIGENCE

Os sistemas voltados às atividades diárias das empresas são essenciais para o atendimento de seus processos operacionais e de suas políticas de negócio. Normalmente esses sistemas focam no processamento transacional (denominados sistemas OLTP – *On-Line Transactional Processing*), são estáticos por natureza, e só são alterados em detrimento a mudanças não intencionais nos processos da organização ou por razões técnicas (IMHOFF, GALEMMO, GEIGER, 2003). Tais sistemas não visam a auxiliar ao processo decisório e à gestão estratégica empresarial. Deste modo, cabe uma distinção entre os sistemas de âmbito operacional e os sistemas de BI de apoio à tomada de decisão. Conforme explica Imnon (2005) a separação entre os dois universos de sistemas tem vários motivos: a forma física como os dados são tratados; as diferenças entre tecnologias usadas para o suporte operacional e suporte informacional; a diferença entre as necessidades e o público alvo das comunidades de usuários e; as características de processamento dos ambientes operacional e analítico. As empresas que demoraram a compreender essas diferenças no passado sofreram com

crises de informação, já que recorriam a sistemas operacionais para obter informações e respostas estratégicas (PONNIAH, 2001).

O propósito de investir em soluções de BI é criar um ambiente pró-ativo para tomada de decisões com base nos sistemas OLTP das empresas. BI consiste na transformação metódica e consciente dos dados provenientes de quaisquer fontes de dados (estruturadas e não estruturadas) em novas formas de proporcionar informação e conhecimento dirigidos aos negócios e orientados aos resultados (BIERE, 2003). Conforme declara Howson (2008), a área de BI é centrada nas pessoas da organização e não diretamente aos recursos tecnológicos e deve permitir que elas acessem, interajam e façam análises para gerir o negócio, para melhorar a produtividade e ainda para descobrir novas oportunidades de mercado. Logo, as tecnologias de processamento analítico (em inglês *OLAP - On-Line Analytical Processing*) devem proporcionar uma visão diferenciada e integrada dos dados oriundos das fontes da organização. A Tabela 1 relaciona os principais diferenciais entre os sistemas e dados operacionais e analíticos.

Tabela 1 - Diferenciação entre os sistemas e tipos de dados operacionais e analíticos.

OLTP	OLAP
Dados são organizados e baseados nas aplicações da organização	Dados são organizados em assuntos e baseados nas áreas de negócio
Sistemas desenvolvidos para estruturar o negócio e reagir aos eventos	Sistemas desenvolvidos para adaptar o negócio e antecipar eventos
Sistema projetado para a eficiência e alto desempenho	Sistema projetado para a efetividade em que o alto desempenho é tolerável
Dados sempre detalhados de forma bruta	Dados podem ser obtidos em detalhe, sumarizados e derivados
Dados são continuamente atualizados e modificados ao longo do tempo	Dados após serem integrados não sofrem atualizações
Exatos em relação ao momento do acesso	Representam valores de momentos já decorridos e também instantâneos

Sistemas atendem à comunidade e processos operacionais	Sistemas atendem à necessidade analítica da comunidade gerencial
Dados são voltados às transações	Dados são voltados à tomada de decisão
São processados repetitivamente	São processados de forma heurística
Os requisitos dos sistemas são conhecidos com antecedência	Os requisitos de análise não são conhecidos a priori
Pequena quantidade de dados utilizada em um processo	Grande quantidade de dados utilizada em um processo
Dados com alta probabilidade de acesso	Dados com baixa ou modesta probabilidade de acesso ao longo do tempo
Dados não contemplam redundância	A redundância de dados é comum
Estrutura fixa, conteúdo variável	Estrutura flexível

Fonte adaptada: INMON, 2005; ECKERSON, 2003; THOMSEN, 2002.

As diferenças citadas das soluções de BI não visam à substituição dos sistemas operacionais de apoio às atividades diárias da empresa. Elas devem atuar de forma paralela, suprindo às demandas analíticas da gestão. Logo, o ambiente para a tomada de decisão não requer qualquer alteração nos sistemas operacionais, tais como ERP (Enterprise Resource Planning), CRM operacional (Customer Relationship Management) e SCM (Supply Chain Management). As arquiteturas de BI são geralmente desenvolvidas para coletar e integrar os dados originados a partir desses sistemas fontes. Gradualmente essas arquiteturas ganharam espaço e a crescente construção de Data Warehouses (DW) bem como o surgimento de metodologias de desenvolvimento tornaram-se familiar nas organizações (KIMBAL; ROSS, 2002; MOSS, 2003).

As arquiteturas tradicionais de BI contemplam vários elementos e técnicas para transformação de dados em informação que são percebidos pelo o número famigerado de siglas e terminologias da área. Destacam-se os processos de Extração, Transformação e Carga de dados (em

inglês, ETL – *Extraction, Transformation and Loading*); a integração de dados nos repositórios como Data Warehouse (DW), Data Mart (DM) e Operational Data Store (ODS); ferramentas de análise de informação conhecidas como ferramentas *OLAP* e os metadados que auxiliam e catalogam o fluxo de dados e processos (KIMBALL; ROSS, 2002; ZENG; et. al., 2006). Conforme cada definição encontrada na literatura, essas arquiteturas podem ser consideradas como refinarias de dados (ECKERSON, 2003), ou como fábricas de informações corporativas (CIF - Corporate Information Factory) (INMON, 2005) e normalmente possuem o DW como componente central. A Figura 2 a seguir demonstra a disposição dos principais componentes encontrados em uma plataforma de BI tradicional.

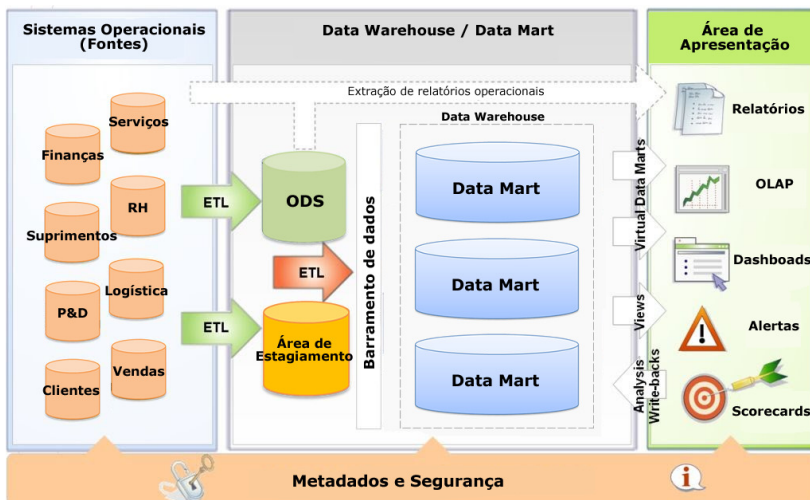


Figura 2 - Arquitetura clássica de Business Intelligence

Fonte adaptada: Hodge (2011)

As arquiteturas podem ser divididas de acordo com os componentes produtores de informação, situados mais à esquerda da Figura 2 em uma região denominada Back-End ou Back-Room, e conforme os componentes consumidores de informação, situados na região à direita denominada Front-End ou Front-Room (KIMBALL; ROSS, 2002; INMON, 2005). Ambas as divisões possuem o DW como limiar, cuja construção deve-se aos componentes e processos de Back-End para o uso por meio de ferramentas analíticas de Front-End. As seções adiante detalham os principais elementos que constituem as

arquiteturas de BI descrevendo as suas evoluções conforme as mudanças tecnológicas e necessidades analíticas.

2.1.1 Data Warehouse e Data Mart

A integração de dados no DW permite que a instituição tenha uma visão coletiva e sumariada do que foi e do que está sendo produzido em suas fontes. Assim, isso cria um ambiente para melhor obtenção de conhecimento por meio de análises e combinação de informações sobre todo o conteúdo disponibilizado ao longo do tempo. Dessa forma, o DW é um elemento chave e núcleo em qualquer arquitetura de BI.

Conforme advogam Inmon, Strauss e Neushloss, (2007), desde suas versões iniciais, o DW é conceituado como um repositório base de apoio à tomada de decisões, orientado a assuntos, integrado, não volátil e variável em relação ao tempo. A criação de um repositório único, na qual reúne as informações de negócio, requer sem dúvida muito esforço, segurança e investimento. A fim de reduzir este custo, muitos projetos de BI iniciam pela criação de repositórios para atendimento a uma área de negócio ou departamento específico. Esses repositórios menores, denominados Data Marts (DM), consistem em coleções de dados para satisfazer às necessidades analíticas de um grupo específico de usuários (INMON; STRAUSS; NEUSHLOSS, 2007).

Dependendo da abordagem na qual é construído, o DW pode ser formado pelo conjunto somatório de Data Marts da organização (desenvolvimento *bottom-up*) ou ainda; ser construído por inteiro e ter cada Data Mart gerado a partir dele (desenvolvimento *top-down*) (BIERE, 2003). Kimball e Ross (2002) adotam a primeira abordagem e argumentam que os dados devem ser armazenados e detalhados em Data Marts individuais e conectados logicamente usando dimensões em conformidade. Essas dimensões são estruturadas em uma arquitetura em que tais autores denominam de *Arquitetura de Barramento* (ou inglês, *Bus Architecture*). Apesar disso, uma pesquisa do TDWI revela que as organizações preferem a segunda abordagem em multicamadas que é proposta por Inmon, nos quais os Data Marts são criados de modo descentralizado a partir do DW (ECKERSON, 2006).

Há evoluções substanciais para a segunda geração de DW - a qual é denominada DW 2.0 por Inmon, Strauss e Neushloss (2007). A primeira geração clássica de DW dava ênfase para a integração de dados

estruturados, principalmente de bases relacionais transacionais. Hoje, sabe-se que o DW é mais efetivo quando combina os dados também das fontes não estruturadas. A gestão sobre o ciclo de vida dos dados não era tão bem reconhecida como agora, já que boa parte das soluções pouco tratava a tarefa de manipular grandes volumes dos dados sem considerar a diminuição de sua probabilidade de acesso e seu envelhecimento. Com a proposta de Inmon, Strauss e Neushloss (2007), a segunda geração do DW apresenta distintos setores conforme a necessidade de acesso e a temporalidade da informação. Esses setores são: *Interactive*, *Integrated*, *Near line* e *Archival*. Outra mudança importante é que os metadados, tanto técnicos quanto de negócio, precisam de um ambiente comum e uma estrutura local para cada componente das plataformas de BI. Esses metadados, nos quais são descritos na seção 2.1.6, são conjuntamente mantidos no DW com as informações de negócio. Isto se deve a facilidade de gerir melhor as mudanças de regras de negócio e ter uma memória do significado dos dados conforme essas mudanças (INMON; STRAUSS; NEUSHLOSS, 2007).

Praticamente, todas as metodologias utilizadas para a construção de DW ou DM, prevêem etapas relacionadas ao planejamento da integração de dados nesses repositórios. Essa integração, na prática, é executada pela equipe de especialistas técnicos por meio de processos denominados Extração, Transformação e Carga de dados, e são descritos a seguir.

2.1.2 Extração, Transformação e Carga

Integrar os dados procedentes dos sistemas operacionais e legados da empresa de uma forma unificada e homogênea é talvez a tarefa mais dispendiosa. Com as mudanças sinalizadas para o *DW 2.0*, o custo necessário para a construção e manutenção do DW deve ser revisto e possivelmente pode ultrapassar os 70% previstos por Kimball e Casserta (2004). Isto porque novos tipos de fontes (heterogêneas, internas e externas) são ainda mais exigidos, os metadados das arquiteturas de BI devem ser estendidos ao uso de recursos semânticos (SELL, 2006) e o volume de informação produzida é cada vez maior. Com isso, os processos de Extração, Transformação e Carga dos Dados (em inglês, ETL - Extraction, Transformation and Loading) no DW são

previstos em praticamente todas as arquiteturas de BI para a consolidação da informação.

Os processos ETL basicamente consistem na captura dos dados das fontes da organização para serem transformados e carregados no DW de acordo com o escopo e nível de granularidade exigidos. Algumas equipes executam esses processos em uma ordem diferente e os chamam de processos ELT (Extraction, Loading and Transformation), que na prática cumprem o mesmo objetivo (HOWSON, 2008). A seguir, esses processos são descritos sumariamente:

- **Extração** (*Extraction*): destina-se a coletar os dados em sua forma bruta tal como estão armazenados nas fontes da organização. Kimball e Casserta (2004) citam que a extração é a leitura dos dados transacionais e a sua cópia em uma área de trabalho anterior ao DW, conhecida como área de estagiamento de dados (ou em inglês – *Data Area Staging*). Essa área serve como um repositório provisório em que os dados devem ser tratados até a sua transição para o DW.
- **Transformação** (*Transformation*): são rotinas de limpeza, validação e preparação dos dados anteriormente extraídos. Para garantir a qualidade das informações do DW, filtros e eliminações de inconsistências devem ser realizados por este processo. As funções de transformação envolvem, por exemplo, a conversão lógica de dados, verificação de domínio e criação de valores padrão. Essas rotinas na maioria das vezes são executadas na área de estagiamento para que os dados fiquem prontos para serem integrados ao DW (INMON; STRAUSS; NEUSHLOSS, 2007; KIMBAL; ROSS, 2004; MOSS, ATRE, 2003).
- **Carga** (*Loading*): ao final de todo o processo de transformação, os dados devem ser consolidados e inseridos no Data Warehouse. Com a nova visão do DW 2.0, o processo de carga deve considerar o ciclo de vida e envelhecimento dos dados e garantir que as cargas incrementais possam distinguir os requisitos de acessibilidade dos dados conforme o tempo (INMON; STRAUSS; NEUSHLOSS, 2007).

Antes mesmo de efetuar a uniformização e inclusão dos dados no DW, é necessário definir qual o modelo de dados e os tipos de informações de negócio que serão extraídas. Os processos ETL, portanto, estão atrelados a características da estrutura de dados adotada para o DW. A tarefa de estruturar e representar os dados em um modelo

adequado para a exploração de informação no DW é conhecida como modelagem dimensional e é explanada a seguir.

2.1.3 Modelagem dimensional

Entre as principais diferenças entre o DW e as fontes de dados transacionais está o modo como os dados estão estruturados. Normalmente, os dados das fontes de dados operacionais encontram-se num modelo desenvolvido para evitar redundâncias e possíveis inconsistências geradas por meio de inserções ou atualizações. Esse modelo, denominado Modelo Entidade-Relacionamento (Modelo E-R ou MER), tem como característica uma alta normalização dos dados em que conceitos, relações e regras do domínio estão organizados em estruturas concisas que obedecem a formas normais (KIMBALL; ROSS, 2002).

O modelo E-R, embora seja eficiente para os sistemas transacionais, não é propício às análises que envolvam um grande volume de dados. As consultas quando aplicadas diretamente nas bases operacionais podem apresentar baixo desempenho, já que muitas estruturas normalizadas precisam ser relacionadas no cruzamento de dados. Além disso, as fontes operacionais costumam ser modificadas concorrentemente pelos sistemas OLTP, e além da perda de desempenho, os resultados das análises podem oscilar de acordo com o momento da consulta. Por outro lado, essa oscilação indica as alterações instantâneas do negócio, que podem ser úteis para a tomada de decisão operacional e monitoração direta dos processos da empresa. Porém, ainda sim as metodologias recomendam que esses tipos de análises devem envolver uma rápida integração de dados e serem disponibilizadas em uma ambiente mais adequado para consultas, tal como o região interativa do DW 2.0 sugerida por Inmon, Strauss e Neushloss (2007).

Como alternativa para o Modelo E-R, o método de modelagem adotado tradicionalmente para o DW é a modelagem dimensional. O modelo dimensional, também chamado de esquema estrela, organiza os dados em uma estrutura padrão e intuitiva que é direcionada ao alto desempenho de consultas e orientada a estratificação de informações. Este modelo baseia-se na denormalização da estrutura de dados, e por isso não se preocupa com a redundância de dados, combinando dados em dimensões e tabelas de fato, em vez de entidades e relacionamentos. Quando o esquema estrela possui dimensões normalizadas, tem-se o

chamado *Snowflake*, que, ainda que perfeitamente legal, deve ser evitado (KIMBALL; ROSS, 2002; THOMSEN, 2002). A Figura 3 a seguir exibe um modelo dimensional e um exemplo de snowflake que tratam sobre o domínio de vendas.

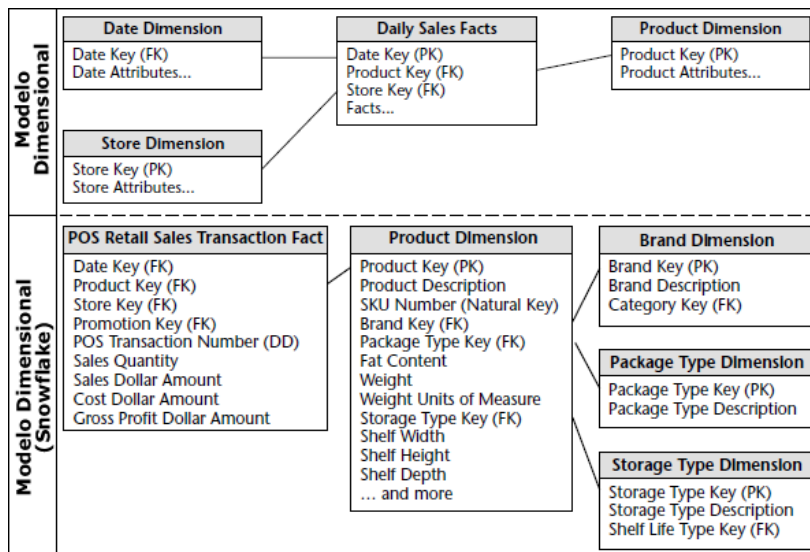


Figura 3 - Ilustração de um modelo dimensional e snowflaking
 FONTE Adaptada: Kimbal e Ross (2002).

Nota-se na parte superior da Figura 3 que as dimensões (com os nomes *Date Dimension*, *Store Dimension*, *Product Dimension*) são estruturas não normalizadas enquanto que a tabela de fato (*Daily Sales Facts*) relaciona cada uma das dimensões em uma estrutura com alta normalização. Já na parte inferior da figura, ocorre um snowflake dado que a dimensão *Product Dimension* relaciona-se diretamente com outras dimensões do modelo.

De acordo com a definição do escopo nas fases iniciais do ciclo de construção do DW, deve-se planejar qual será a granularidade a ser considerada na modelagem dimensional (KIMBALL; ROSS, 2002). A granularidade representa como o dado é organizado com relação ao seu nível de detalhamento. Uma alta granularidade significa que o dado possui um nível de detalhe menor ao modo como está estruturado nas fontes de origem (PONNIAH, 2001). Por exemplo, embora as transações nos sistemas fontes registrem os minutos e segundos de uma operação, pode-se determinar que o modelo dimensional organize os

dados em sumarizações mensais, sem se preocupar com dias, horas ou minutos. Dessa forma, o modelo dimensional organizaria o dado com uma granularidade maior e com menor nível de detalhe em relação ao tempo. Já o contrário, ou seja, uma granularidade baixa denota um maior nível de detalhe e possivelmente análises com mais informações.

Após a conclusão da integração dos dados no modelo dimensional planejado, todo o conteúdo do DW pode ser acessado e apresentado aos stakeholders da organização. Assim, a área de apresentação, que contempla os métodos e ferramentas de análise de informação, é descrita na seção ulterior.

2.1.4 Área de Apresentação

Posteriormente à construção de toda a infraestrutura de informações, é necessário prover meios de análise sobre conteúdo integrado aos usuários da organização. Assim sendo, uma variedade de ferramentas de consulta, geração de relatórios e métodos de visualização de informações devem ser projetados conforme os requisitos de análise definidos. A área de apresentação de informações, também chamada de *Front-End* ou *Front-Room*, é onde essas ferramentas atuam em interação com a comunidade de usuários. Ela destina-se a tornar acessível toda a informação contida no DW, Data Marts, ou *ODS* para dar suporte ao processo de descoberta e uso de conhecimento (CODY, et. al., 2002; KIMBALL; ROSS, 2002; INMON, 2005).

Conforme a necessidade de informação e perfil do usuário, diferentes ferramentas e métodos de análises de dados podem ser desenvolvidos (THOMSEN, 2002). A área de apresentação de dados costuma aplicar, além das ferramentas de consultas ou ferramentas OLAP, técnicas de mineração de dados (*Data Mining*) que buscam entender características e padrões de eventos ocorridos, levantar indicadores estatísticos e fazer análises preditivas (HOWSON, 2008).

Em virtude da demanda por métodos ágeis para a tomada de decisões, a área de apresentação deve oferecer meios para obter informações tanto atuais (presente próximo) quanto informações históricas (passado), tal como apontado por Inmon, Strauss e Neushloss (2007) na segunda geração de DW. Com o surgimento de data warehousing interativos e dispositivos visuais e analíticos em tempo real (por exemplo, *dashboards*, *mecanismos de alertas*, *entre outros*), a tarefa de analisar as informações instantâneas dentro do contexto

histórico e integrado de BI foi facilitada (ECKERSON, 2003). Assim, as análises e ferramentas da área de apresentação podem ser classificadas e desenvolvidas de acordo com esses critérios de acessibilidade e uso. A Figura 4 apresenta um estudo feito pelo TDWI que identifica os tipos de análises e ferramentas conforme a sua proporção de utilização pelos usuários da organização.

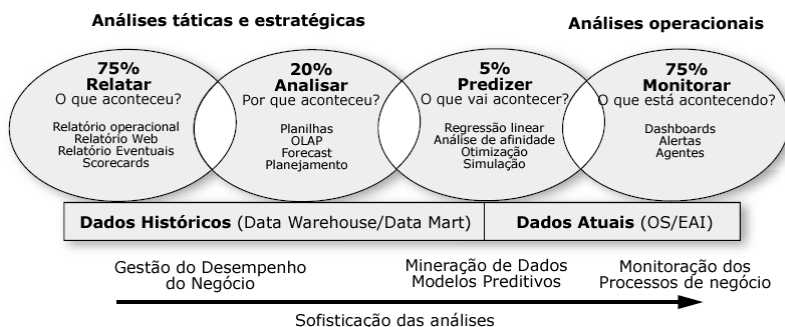


Figura 4 – Tipos de análises e ferramentas em proporcionalidade de uso

Fonte adaptada: Eckerson (2003).

As análises que se baseiam em fatos históricos são utilizadas principalmente para gestão estratégica da organização. Isto porque as decisões estratégicas envolvem análise de dados para propósitos de planejamento por longo tempo (semestres ou anos) em alinhamento com a visão e a missão da empresa. Análises táticas voltam-se para as ações que devem ser tomadas num futuro próximo (semanas ou meses) e são mais focadas aos processos do que as análises estratégicas. Por fim, as análises e decisões operacionais precisam ser feitas imediatamente, e dessa forma, contam com a ajuda de *dashboards*, agentes, alertas e outros elementos da área de apresentação. Conforme o estudo do TDWI, 75% dos usuários, representados pelos extremos da Figura 4, utilizam relatórios estáticos ou parametrizados, buscas pré-definidas baseadas em valores históricos, ou ainda, monitoram e reagem aos indicadores de desempenho. Já as análises e as previsões (ao centro da Figura 4) são utilizadas por 20% e 5%, respectivamente, por pessoas que se ocupam em verificar os dados em detalhes para explorar a causa de problemas e criar previsões ou tendências (ECKERSON, 2003).

Dentre as funções de consulta e análises de informações desempenhadas pelas ferramentas analíticas, estão as operações OLAP. Neste trabalho estas operações, nas quais são executadas de diferentes modos de acordo com a ferramenta, são realizadas de modo único por

meio de linguagem natural. A seção a seguir descreve o processamento OLAP.

2.1.5 OLAP

O termo OLAP (*On-Line Analytical Processing*) refere-se à tecnologia de processamento analítico que é designada a obter novas informações de negócio por meio de um conjunto de transformações e cálculos executados sobre as fontes de dados (MOSS, ATRE, 2003). O processamento OLAP ou processamento analítico, aplicado pelas ferramentas de apoio à decisão, possibilita a navegação de forma amigável pelo modelo multidimensional do DW. Esse processamento constitui uma importante etapa para aquisição de conhecimento e a transformação desse conhecimento em ações (PONNIAH, 2001).

Por meio da combinação entre as diferentes dimensões e tabelas de fato, é possível sumarizar o conteúdo em uma estrutura denominada *cubo OLAP* ou ainda *hipercubo* (KIMBALL; ROSS, 2002; IMNON, 2005). Essa estrutura estabelece um formato em que perspectivas de visualização de informações podem ser facilmente criadas conforme a interação com o usuário.

Dentre as principais características compreendidas nas ferramentas OLAP estão: apresentação de uma visão multidimensional dos dados de forma intuitiva ao tomador de decisão (Cubo OLAP, gráficos, etc.); sumarização e agregação de dados; capacidade de consultas e análises interativas sobre o retorno dos dados e suporte para que os analistas de negócio customizem suas próprias consultas e cálculos (MOSS, ATRE; 2003).

As formas de processamento OLAP variam principalmente conforme o tipo de armazenamento de dados utilizado, sendo: ROLAP (*Relational OLAP*) – utilizado quando as dimensões e tabelas de fato são modeladas nas estruturas dos bancos de dados relacionais; MOLAP (*Multidimensional OLAP*) – quando armazena os dados em bases de dados multidimensionais; HOLAP (*Hybrid OLAP*) – quando combina as duas formas ROLAP e MOLAP e; DOLAP (*Dynamic OLAP*) – utiliza estruturas de armazenamento temporário de acesso rápido (*cache* de dados) automaticamente de acordo com a execução de consultas (HOWNSON, 2008).

Com o intuito de navegar e localizar informações a partir do DW, as ferramentas OLAP fornecem diversas funcionalidades na quais, em resumo, destacam-se:

- **slice-dice:** capacidade de acessar o DW por meio de qualquer de suas dimensões de maneira igual. É o processo de separação e combinação de dados com várias possibilidades de cruzamento de informações. (KIMBALL; ROSS, 2002);
- **drill-up ou roll-up:** permitem navegar até um nível ou hierarquia de detalhe imediatamente superior (mais granular) a partir de uma dimensão. Normalmente associado à ação de remover um cabeçalho de linha ou coluna para resumir um conjunto de dados (INMON, 2005; KIMBALL; ROSS, 2002);
- **drill-down:** ao contrário de roll-up, refere-se a ação de percorrer uma hierarquia de nível superior de agregação para níveis inferiores de detalhamento (IMHOFF, GALEMMO, GEIGER, 2003);
- **drill-across:** possibilita a combinação de dados entre duas ou mais tabelas de fatos em um única análise, quase sempre envolvendo consultas separadas que são posteriormente unidas (KIMBALL; ROSS, 2002);
- **drill-through:** ocorre quando o usuário faz análises de distintas visões proporcionadas por troca de informações entre dimensões, por exemplo, o usuário realiza análises de indicadores pela dimensão geografia e posteriormente passa a analisar sobre a dimensão tempo (SELL, 2006).

Todos os componentes e os processos das arquiteturas tradicionais de BI descritos utilizam como base uma camada de metadados que é explanada na seção seguinte.

2.1.6 Metadados

A descrição de como e onde os dados estão organizados nos repositórios, bem como o que representam conforme o contexto do negócio são alguns dos motivos por que as plataformas de BI possuem uma camada de metadados (MOSS; ATRE, 2003). Metadados são muito conhecidos por serem dados que descrevem dados (dados sobre dados) similarmente a dicionários ou catálogos de informações. Consoante afirma Ponniah (2001), os metadados vão além de meros dicionários de dados ou catálogos, e atuam em conjunto com os componentes das

arquiteturas de BI auxiliando na interação e fornecendo informações para governança dos processos.

Para Inmon, Strauss e Neushloss (2007) o metadado é o componente mais importante e crítico para a construção do DW. Esses autores declaram que na primeira versão do DW, paradoxalmente os metadados eram negligenciados e dificilmente tinham um lugar de destaque dentro das metodologias. Embora estejam inicialmente previstos nas arquiteturas, o seu uso não era reconhecido. Geralmente, os metadados eram separados da estrutura dos dados, no entanto, com a proposta do DW 2.0 os metadados ganham um local de armazenamento mais próximo aos dados. Agora, eles passam a ser armazenados em uma área fim juntamente com os dados da organização. Isso porque, as regras de negócio mudam constantemente, novos requisitos tanto de análise quanto administrativos surgem e a informação descritiva dos dados precisa ser documentada. Outra justificativa para a importância dos metadados é a crescente necessidade da aplicação dos processos ETL sobre as fontes de dados não estruturadas (INMON; STRAUSS; NEUSHLOSS, 2007).

Encontram-se inúmeras classificações para os metadados na literatura. Moss e Atre (2003) classificam os metadados conforme as três divisões de uma arquitetura tradicional de BI exibida pela Figura 2. São eles: metadados operacionais, metadados de ETL e metadados da área de apresentação ou de usuário final. Imhoff (et. al., 2003) divide-os em: metadados técnicos, metadados de negócio e metadados administrativos. Segundo Inmon, Strauss e Neushloss (2007), os metadados existem em diversos níveis hierárquicos, tais como empresariais, que são formados pelos metadados locais, que por sua vez distinguem-se em metadados de negócio e técnicos.

Normalmente, observa-se que os metadados consistem de descrições sintáticas dos processos e componentes das arquiteturas de BI convencionais e por isso, não oferecem expressão semântica para a realização de raciocínios e colaborar com as análises feitas pelo gestor (SELL, 2006). Nesse sentido, as ontologias possuem um papel importante no desenvolvimento de metadados sendo usadas para a criação de visões inteligentes sobre os recursos de informação, para consultas sobre bases de dados e documentos textuais (LAVBIC; VASILECAS; RUPNIK, 2010). A seção a seguir descreve como o uso de tecnologias semânticas e ontologias, além de funcionar como metadados mais adequado, podem assistir às aplicações analíticas nesta pesquisa.

2.2 TECNOLOGIAS SEMÂNTICAS

A fim de atender aos processos da gestão do conhecimento, inúmeras pesquisas tem se voltado para o uso de tecnologias semânticas baseadas nos fundamentos propostos pela Web Semântica (DAVIES; FENSEL; VAN HARMELEN, 2003). A base teórica da Web Semântica oferece oportunidades às empresas para que possam localizar e compartilhar informações corporativas, descobrir novos conhecimentos para apoio à tomada de decisão por meio de agentes de software, melhorar a sua visão estratégica e aperfeiçoar os seus processos de negócio (DACONTA; OBRST; SMITH, 2003). Além disso, a Web Semântica prevê mecanismos para representação de domínios e para aplicação de inferências sobre bases de conhecimento que podem gerar conhecimento útil a quem precisa (FENSEL, 2001). Neste cenário, a aplicação de tecnologias semânticas ganhou notoriedade em iniciativas de BI, tanto para a integração semântica de fontes de dados quanto para apoio ao processamento OLAP. Dentre as tendências para o futuro da área, o uso das tecnologias semânticas já demonstra grandes avanços para que os sistemas baseados em conhecimento possam apoiar ao processo decisório (SAGGION, et. al., 2007; SELL et. al., 2008; BÖHRINGER et. al., 2010; LAVBIC; VASILECAS; RUPNIK, 2010).

Esta pesquisa fundamenta-se nas abordagens de BI que incorporam tecnologias semânticas para assistir às análises. Assim, as próximas subseções descrevem os recursos e bases conceituais dessas abordagens que são utilizados também por este trabalho. Os formalismos de representação de conhecimento previstos na Web Semântica e formas de aplicação de inferências são discutidos.

2.2.1 Web Semântica

A Web Semântica chamou a atenção para o uso de mecanismos de representação de conhecimento e formas mais inteligíveis por máquinas para tratamento de informações. Em resumo, a Web Semântica é uma extensão da Web atual que permite a criação de um ambiente para que os sistemas possam processar e compreender o seu conteúdo a fim de colaborar automaticamente com as pessoas (BERNEERS-LEE, 2001). Essa extensão se deve a vários motivos, dentre eles estão: a baixa exatidão das buscas da Web atual;

sensibilidade dos resultados às palavras-chave; possibilidade do retorno de grande quantidade de conteúdo não relacionado ou ainda nenhum documento encontrado; e o esforço empregado em muitas consultas para analisar o conteúdo que está disperso em vários documentos (ANTONIOU; HARMELEN, 2008). Esses problemas agravam-se ainda mais com o aumento do volume de documentos publicados na Web e a forma atual como estão estruturados.

Para tornar solúveis os problemas da Web atual, a Web Semântica propõe uma arquitetura dividida em camadas na qual é exibida pela Figura 5. A estrutura da pilha de formalismos da Web Semântica vem sendo revista aos longos dos anos pelo W3C (2007) e a Figura 5 representa a versão de 2007.

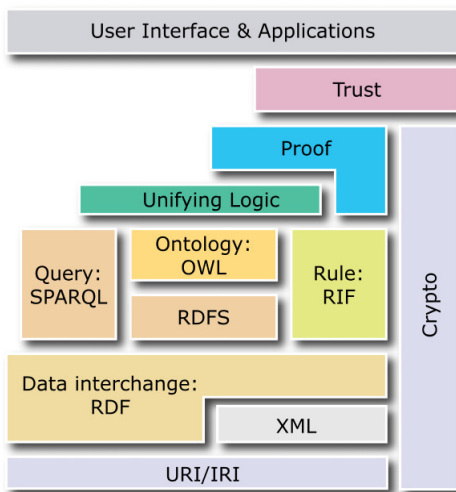


Figura 5 - Diagrama em camadas da Web Semântica

Fonte: W3C(2007).

A arquitetura multicamadas da Web Semântica, exposta no diagrama acima, está organizada de modo que cada camada inferior ou adjacente provê suporte para que as demais camadas cumpram seus objetivos. Visto que a arquitetura ainda encontra-se em evolução, os formalismos das camadas mais superiores *Unifying Logic*, *Crypto*, *Proof* e *Trust*, ilustradas no diagrama da Figura 5, encontram-se em proposição pelo W3C (2007).

Em sua camada mais base, URI/IRI, a arquitetura da Web Semântica trata da codificação de caracteres e também do

endereçamento e nomeação de recursos na Web. Para tal, ela utiliza os padrões URI³ (*Uniform Resource Identifier*) e IRI⁴ (*Internationalized Resource Identifier*) que são baseados na formatação de caracteres *Unicode*. Na camada imediatamente superior encontra-se XML⁵ (*eXtensible Markup Language*) que estabelece a linguagem de estruturação dos recursos para o intercâmbio de dados entre os sistemas na Web e; RDF⁶ (*Resource Description Framework*), modelo baseado na sintaxe XML que representa os recursos da Web na forma da tripla Sujeito-Predicado- Objeto. Logo acima, estão os formalismos RDFS (*Resource Description Framework Schema*) que orienta para a formatação de um documento RDF válido; OWL⁷ (*Web Ontology Language*) que define o modelo formal de representação de ontologias; SPARQL⁸ que define a sintaxe de consulta e recuperação de recursos sobre o modelo RDF e; RIF⁹ (*Rule Interchange Format*) que impõe a formatação para o intercâmbio de regras de inferência (W3C, 2004; W3C, 2008; W3C, 2009). As demais camadas são: *Unifying Logic* - prevê a adição de regras de inferências e a lógica de predicado em complementação ao uso de ontologias; *Proof* - tem como papel a comprovação dos processos dedutivos e validação das camadas inferiores; *Trust* – responsável por avaliar a veracidade e integridade no uso e no compartilhamento de ontologias; *Crypto* – responsável por garantir o intercâmbio de informações sigilosas e por fim; *User Interface & Application* – que representa a interação entre os agentes e serviços da web semântica em colaboração com os usuários (DAVIES; FENSEL; VAN HARMELEN, 2003; ANTONIOU; HARMELEN, 2008).

Com o suporte tecnológico fornecido pelas camadas inferiores e adjacentes, o uso de ontologia é recomendado para a modelagem e representação de conhecimento e na prática, já é aplicado no auxílio aos processos de gestão do conhecimento (DAVIES; FENSEL; VAN HARMELEN, 2003). Os trabalhos de BI relacionados a esta pesquisa utilizam ontologias como meio de prover semântica às análises e elaborar metadados mais expressivos nas plataformas de apoio à

³ URI - <<http://tools.ietf.org/html/rfc3986>>

⁴ IRI - <<http://tools.ietf.org/html/rfc3987>>

⁵ XML - <<http://www.w3.org/XML>>

⁶ RDF - <<http://www.w3.org/TR/rdf-concepts>>

⁷ OWL - <<http://www.w3.org/TR/owl2-overview>>

⁸ SPARQL - <<http://www.w3.org/TR/rdf-sparql-query>>

⁹ RIF - <<http://www.w3.org/TR/rif-overview>>

decisão. A seguir, a próxima subseção comenta a respeito de ontologias e a sua importância no contexto das arquiteturas de BI.

2.2.2 Ontologia

Fensel (2001) declara que o uso de ontologias é um tópico de pesquisa atuante em diversas áreas tais como Engenharia e Gestão do Conhecimento, Processamento de Linguagem Natural, Integração de fontes de dados e Sistemas de Informação. O termo ontologia possui origem na Filosofia e sua definição é bastante disseminada pela comunidade da área de computação como uma especificação formal e explícita de uma conceituação compartilhada (GRUBER, 1993; BORST, 1997). Essa definição pode ser detalhada como:

“Conceituação se refere a um modelo abstrato de algum fenômeno do mundo que identifique seus conceitos relevantes. Explícito significa que os tipos de conceitos utilizados e as restrições sobre seu uso são explicitamente definidos. Formal refere-se ao fato de que a ontologia deve ser legível por máquinas. Compartilhada reflete a noção de que uma ontologia captura o conhecimento consensual, ou seja, ele não é privado de algum indivíduo, mas aceito por um grupo.” (STUDER; BENJAMINS; FENSEL, 1998).

Podem-se distinguir as ontologias que são mais próximas de taxonomias das ontologias que modelam o domínio de uma forma mais profunda com mais restrições sobre a semântica do domínio. As primeiras, denominadas de ontologias *lightweight*, incluem taxonomia de conceitos, relações entre conceitos, e as propriedades que descrevem esses conceitos. Já a outra classe de ontologias, chamada de ontologias *heavyweight*, adiciona axiomas e restrições às ontologias *lightweight* para esclarecer os significados das suas terminologias (PÉREZ; FERNÁNDEZ-LÓPEZ, CORCHO, 2004). Guarino (1998) classifica as ontologias quanto à sua generalidade ou nível de representação em: ontologia de alto nível (*top-level*); ontologia de domínio; ontologia de tarefa e ontologia de aplicação.

Por apresentar estruturas formais como classes, propriedades, relacionamentos, instâncias de classes e restrições sobre esses elementos (DACONTA; OBRST; SMITH, 2003), a ontologia pode ser comparada

ao desenvolvimento orientado a objetos ou ainda aos esquemas de bancos de dados. Fensel (2001) argumenta que as ontologias diferenciam-se dos esquemas de bases de dados uma vez que: a) a linguagem para a definição da ontologia é semanticamente mais rica que os modelos de bancos de dados; b) a informação que é descrita pela ontologia consiste em textos em linguagem natural semi-estruturados e não se encontra no formato tabular; c) as terminologias da ontologia devem ser compartilhadas e cada significado deve ter consenso para que as informações sejam trocadas em um grupo de usuários; d) uma ontologia provê uma teoria do domínio e não uma estrutura de armazenamento de dados. Noy e McGuinness (2001) afirmam que a construção de ontologias baseia-se nas estruturas semânticas das classes e relacionamentos, já o desenvolvimento orientado a objetos é centrado nos métodos ou nas operações que as classes têm que exercer.

A ontologia possui um conjunto de conceitos, propriedades e relacionamentos que dão semântica ao universo de discurso tratado. Ela pode ser usada para o compartilhamento de conhecimento e modelagem do negócio da organização sob a forma de ontologia de domínio. Dentre outros benefícios, ela serve também como metadados mais expressivos para o tratamento de informações e podem ser aplicada para guiar os processos analíticos e componentes de uma arquitetura de BI. Além das ontologias de domínio, outras ontologias auxiliares, consoante o que Guarino (1998) classifica como ontologia de tarefa e ontologia de aplicação, são aplicadas para a anotação semântica das fontes de dados e em complementação aos metadados das soluções de BI (SAGGION; et. al., 2007; SELL, 2006; et. al., 2008; PRIEBE; PERNUL, 2003).

Quanto ao uso de linguagem natural as iniciativas comumente aplicam ontologias: para formalização e representação conceitual das perguntas dos sistemas de *Question Answering*; para identificação ou reconhecimento de conceitos ou classes; como auxílio na criação de índices textuais e buscas sobre bases de conhecimento; (NIRENBURG; RASKIN, 2004; MCGUINNESS, 2004; LOPEZ; et. al., 2007).

A subseção a seguir trata do processo de raciocínio e aplicação de regras de inferência necessários para auxiliar o processo decisório e retorno de informações nesta pesquisa.

2.2.3 Raciocínio e regras de inferência

Pearl (1988, apud BEPLER, 2008) comenta que ambos os conceitos *raciocínio* e *inferência* se confundem, sendo raciocínio um processo de inferir um novo conhecimento, e inferência diferencia-se por tratar da derivação de fatos ou conhecimentos a partir de um conjunto de dados. No contexto da Web Semântica, a inferência pode ser caracterizada pela descoberta automática de novos conceitos ou relacionamentos entre conceitos da ontologia a partir de regras, vocabulários ou axiomas aplicados sobre os dados. As ontologias, nas quais são formalizadas na linguagem OWL, concentram-se em métodos de classificação nos quais dão ênfase para a definição da estrutura de classes (e subclasses) e como os recursos (ou instâncias) são associados a essas classes. Por outro lado, as regras, estabelecem um mecanismo geral para a descoberta e geração de novos relacionamentos baseados nos já existentes (W3C, 2010).

Embora seja expressiva suficiente para representar um domínio de conhecimento por meio de classes, propriedades e relacionamentos, OWL (Web Ontology Language), não é destinada a produzir regras ou sentenças axiomáticas entre classes e propriedades para o raciocínio com base em regras de inferência. As inferências que podem ser feitas com os construtores¹⁰ de OWL são baseadas na Lógica Descritiva (*Description Logic*), na qual representa o conhecimento com declaração de classes e propriedades com semântica bem definida por meio de disjunções, uniões, etc. (ANTONIOU; HARMELEN, 2008; W3C, 2010).

Para dar mais poder de inferência aos sistemas computacionais, o W3C propõe e ainda estuda formalismos para compor a camada *Unifying Logic* da Web Semântica. Nesse contexto um conjunto de linguagens e frameworks vem sendo desenvolvidos para a aplicação de regras de inferência e realização de raciocínio por sistemas. Essas linguagens, como por exemplo, SWRL (Semantic Web Rule Language) oferecem um conjunto de declarações antecedentes – que definem as premissas das regras e; um conjunto de fatos conseqüentes, caso essas regras sejam satisfeitas (HORROCKS, et. al.; 2004). Assim, a partir de um conjunto de regras, fatos ou axiomas definidos em uma linguagem

¹⁰ Elementos formais da linguagem utilizados para criar ou definir um conceito (como classe, propriedade, relacionamento, instância).

formal, os ditos mecanismos de inferência (ou em inglês, *reasoners*) podem realizar o processo de inferência sobre a ontologia e bases de conhecimento. Dois exemplos de regras de inferência aplicadas sobre uma ontologia que trata das relações familiares¹¹ é ilustrado no Quadro 1 a seguir.

Regra 1: $(?x \text{ possuiPai } ?y) \wedge (?y \text{ possuiIrmão } ?z) \rightarrow (?x \text{ possuiTio } ?z)$

Regra 2: $(?x \text{ possuiIrmão } ?y) \wedge (?y \text{ possuiFilho } ?z) \rightarrow (?x \text{ possuiSobrinho } ?z)$

Quadro 1 - Exemplos de regras de inferência

No Quadro 1, os elementos $?x$, $?y$ e $?z$, escritos com a Notação 3 (N3)¹², representam variáveis que remetem a uma instância da classe *Pessoa* da ontologia tratada. Já os itens *possuiPai*, *possuiIrmão*, *possuiTio*, *possuiFilho* e *possuiSobrinho* são auto-relacionamentos da classe *Pessoa* neste exemplo. Em resumo, a *Regra 1* estabelece em termos formais que se uma pessoa X possui um pai Y e se Y possui um irmão Z conclui-se que a pessoa X possui como tio a pessoa Z. Da mesma forma, a *Regra 2* define que se uma pessoa X possui um irmão Y e Y possui um filho Z, implica dizer que a pessoa X possui um sobrinho que é Z. Note que ambas as relações *possuiSobrinho* e *possuiTio* do exemplo são derivadas e explícitas a partir de relações prévias como *possuiPai*, *possuiIrmão* e *possuiFilho*. Assim, a aplicação de regras de inferência possibilita que informações ou relações antes implícitas sejam descobertas a partir de bases de conhecimento. Como prática da Engenharia do Conhecimento, as regra de inferência, tal como a ontologia, são formas de representação de conhecimento que, no contexto de BI deste trabalho, são usadas também como meio para geração de novos conhecimentos a partir das informações do DW.

A fim de aperfeiçoar ainda mais o processo de inferência, alguns estudos já estendem as propostas de linguagem de definição de regras de inferência para suportar o uso de funções matemáticas (PATEL-SCHNEIDER, 2005). São os casos em que é necessário o uso de operações e parâmetros computáveis como, dados sobre o tempo

¹¹ O exemplo utiliza como base a ontologia de relações familiares localizada em: <http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl>

¹² Notação N3 - <<http://www.w3.org/DesignIssues/Notation3.html>>

(passado, atual e futuro), dados preferenciais do usuário (localização geográfica, idioma, etc.), dentre outros em conjunto com a sintaxe de regras. Este trabalho adota também essa extensão para definir funções e cálculos na escrita das regras de inferência. Devido à ausência de mecanismos de inferência que manipulem por completo a linguagem SWRL, as regras definidas na prototipação da arquitetura no capítulo 4 têm como base a sintaxe do framework utilizado chamado Jena (JENA, 2011).

Dentre os trabalhos relacionados a esta pesquisa, Sell (et. al., 2008) adota duas abordagens para o suporte ao processo de criação de conhecimento e consideração das novas informações inferidas nas análises. Essas duas abordagens, *on-the-fly* e *in-batch* são usadas neste trabalho e são brevemente descritas abaixo.

- **On-the-fly:** nessa abordagem as regras de negócio definidas sobre o modelo da ontologia são aplicadas durante o processamento analítico. Com base nas regras de negócio e nos conceitos representados na ontologia de domínio, os resultados das inferências são usados para a construção das consultas sobre o DW. Para mais informações, vide Sell (2006; et. al., 2008).
- **In Batch:** a fim de melhorar o desempenho do processo de inferência da abordagem *on-the-fly*, as derivações semânticas obtidas pela aplicação das regras são armazenadas previamente no DW, tal como nos processos ETL. Uma estrutura similar ao framework RDF, denominada *Modelo Tripla*, é utilizada para associar quaisquer duas dimensões ao resultado da inferência. As ferramentas analíticas podem então combinar as informações do modelo dimensional com o Modelo Tripla para apresentar o resultado da inferência. Uma explicação mais detalhada desse modelo pode ser vista na seção 3.10 ou nos trabalhos de Sell (et. al., 2008) e Silva (2006).

2.2.4 Iniciativas de Business Intelligence baseadas em tecnologias semânticas

Dentro do contexto de BI e assim como em outras áreas de conhecimento, verifica-se na literatura que as tecnologias semânticas citadas podem ser aplicadas para diversos propósitos. Em meio aos

tópicos abordados por essas pesquisas e considerando como base a proposta deste trabalho, destaca-se:

- **Trabalhos de BI que utilizam tecnologias semânticas para extração de informação e descoberta de conhecimento em bases textuais.** São as soluções baseadas em conhecimento que focam em bases não estruturadas ou semi-estruturadas como documentos, informações publicadas na Web, emails e outros. Dentre os projetos desse tópico de pesquisa menciona-se o MUSING¹³ (*MULTI-industry, Semantic-based next generation business INtelliGence*) que integra as tecnologias da Web semântica com o processamento de linguagem natural. Essa pesquisa combina métodos baseados em regras e abordagens estatísticas para a aquisição de conhecimento e auxílio ao processo de raciocínio para integração e exploração de conteúdo sob os contextos: gestão financeira, riscos operacionais de TI e Internacionalização (SAGGION, et. al., 2007). Os métodos e técnicas de extração de informação utilizados por esses estudos são úteis neste trabalho principalmente para o processo de interpretação de linguagem natural.
- **Abordagens de BI baseados em ontologias, taxonomias e anotação semântica para a integração de fontes de dados heterogêneas.** Contemplam as iniciativas mencionadas por Cody (et. al., 2002) para combinar fontes de dados estruturadas e não estruturadas para atender a inteligência competitiva, contudo utilizam como meio ontologias, thesaurus, taxonomias e outros. Cita-se, como exemplo, a arquitetura SEWASIE¹⁴ (*Semantic Webs and AgentS in Integrated Economies*) que, por meio de uma rede de agentes mediadores baseados em ontologias, une os conteúdos dispersos em múltiplas fontes de dados (BENEVENTANO; et. al., 2007). Faz-se referência também à proposta de Priebe e Pernul (2003) que cria um ambiente que integra o processamento OLAP e as funcionalidades de recuperação de informação a partir documentos e de informações do DW através de ontologias. No contexto desta proposta, esses estudos são importantes uma vez que unem as áreas de recuperação de informação e BI, e ainda outras correlatas.
- **Trabalhos que além de ontologias aplicam raciocínio sobre bases de conhecimento para estender às funcionalidades OLAP.**

¹³ MUSING - <<http://www.musing.eu>>

¹⁴ SEWASIE - <<http://www.sewasie.org>>

São as linhas de pesquisa que incorporam mecanismos de inferência normalmente baseados em regras declaradas sobre os conceitos da ontologia de domínio. O resultado das inferências, isto é, as derivações semânticas são utilizadas para explicitar novas variáveis de interesse nas análises. Cita-se aqui o framework SBI (*Semantic Business Intelligence*) proposto por Sell (2006; et. al., 2008), na qual possui módulos baseados em ontologias que flexibilizam o processamento analítico por permitir que o conteúdo do DW seja explorado e combinado com o resultado do processo de inferência. Outro trabalho relevante denominado DSS-MAS (Decision Support Systems - Multi-Agent Systems) que a partir da integração de dados heterogêneos no DW, permite a aplicação de regras para inferir novos conhecimentos em colaboração com o processo de decisão (LAVBIC; VASILECAS; RUPNIK, 2010). Esta pesquisa adota ontologias e as técnicas de raciocínio tratadas por esses frameworks em conjunto com o uso de linguagem natural. Tais frameworks são os principais trabalhos de BI relacionados a esta pesquisa.

A seção posterior introduz os aspectos teóricos para a interpretação semântica de perguntas com base na disciplina de *Question Answering*.

2.3 QUESTION ANSWERING

Por considerar que o conhecimento pode ser explícito a partir de meios como documentos, livros, revistas, artigos científicos e mais recentemente, em blogs, emails, dentre outras mídias, a Engenharia do Conhecimento beneficia-se também de tecnologias de processamento de linguagem natural (RAO, 2005). Devido à necessidade de busca por informações dispersas na Internet e Intranets organizacionais, essas tecnologias, tais como as ferramentas de busca, de fato já fazem parte do cotidiano da sociedade da informação. No entanto segundo Katz, Lin e Felshin (2001), o potencial de conhecimento disponibilizado nesses variados meios, principalmente na Web, ainda não foi obtido por causa da falta de métodos efetivos de acesso à informação. Esses métodos referem-se ao uso de linguagens naturais para a condução de análises mais precisas a partir das fontes de dados. Dessa forma, a tarefa de *Question Answering* (QA) tem sido discutida em vários eventos desde

então, como TREC¹⁵ (*Text REtrieval Conference*); MUC¹⁶ (*Message Understanding Conference*); CLEF¹⁷ (*Cross-Language Evaluation Forum*) e NTCIR¹⁸ (*NII Test Collection for IR Systems*).

Question Answering (QA) é uma tarefa proveniente da combinação das áreas Recuperação de Informação (RI) e de Extração de Informação (EI) que visa à obtenção de uma resposta exata e precisa com base em uma pergunta formulada em linguagem natural (BILOTTI, 2004; QUARTERONI, 2007). Em vez de palavras-chave e retorno de listas de documentos, em QA espera-se que o usuário informe uma pergunta sintaticamente e semanticamente coerente para que seja possível extrair uma resposta efetiva e completa (HIRSCHMAN, GAIZAUSKAS; 2001).

Em geral, a combinação de RI e EI deve-se ao fato de que os sistemas de Question Answering tipicamente aplicam métodos de busca sobre bases textuais indexadas para localizar a lista de documentos mais relacionada à pergunta informada (métodos associados às tarefas de RI). A partir dessa lista, métodos de extração de parágrafos ou trechos de informações relevantes (em inglês, *passage extraction*) são executados para identificar a resposta correta à pergunta informada (métodos associados às tarefas de EI). A seção a seguir detalha ambas as áreas e aponta os principais modelos e técnicas aplicados pelas pesquisas da área de Question Answering.

2.3.1 Recuperação e Extração de Informação

A área de RI trata da representação, armazenamento, organização e da acessibilidade aos itens de informação de maneira fácil de acordo com a necessidade do usuário (BAYEZA-YATES, RIBEIRO-NETO, 1999). Essa área desenvolve modelos e algoritmos para estruturar e organizar repositórios de documentos, em sua maioria em índices textuais, com o intuito de localizar informações rapidamente. Diferentemente dos sistemas de QA que apresentam uma única resposta ao usuário, nos sistemas de RI o usuário entra com um conjunto de palavras-chave para que o sistema retorne uma lista de documentos

¹⁵ TREC - <<http://trec.nist.gov>>

¹⁶ MUC - <http://www-nlpir.nist.gov/related_projects/muc/>

¹⁷ CLEF - <<http://www.clef-campaign.org>>

¹⁸ NTCIR - <<http://research.nii.ac.jp/ntcir>>

relevantes aos termos informados (MANNING; RAGHAVAN, SCHÜTZE, 2008).

Os sistemas de RI utilizam diversos modelos, técnicas e métricas que podem ser conjugados para a melhoria dos resultados. Dentre as principais medidas estão *precision* (relação entre o número de itens retornados e considerados relevantes com o número total de itens retornados) e *recall* (razão entre a quantidade de itens relevantes retornados e a quantidade total de itens relevantes presente na fonte de dados) (KOWALSKI, MAYBURY, 2000). Com base principalmente nessas medidas, diversos modelos de RI são desenvolvidos e comparados. Dentre os modelos encontrados na literatura cita-se: modelo booleano, vetorial, probabilístico, difuso, indexação de semântica latente, dentre outros (BEPPLER, 2008).

Para poder identificar exatamente o trecho de informação que está disperso nos repositórios de dados e documentos, os sistemas de QA utilizam também técnicas e algoritmos da área de EI. Segundo Moens (2006), EI é a identificação, classificação e estruturação de informações textuais específicas em classes semânticas tornando-as mais apropriadas para o processamento de informações. Nesse contexto, a área de EI analisa as partes dos documentos que potencialmente contém informação relevante segundo os critérios de extração definidos (KOWALSKI, MAYBURY, 2000).

Para gerar ou explicar conhecimento, a área de EI aplica inúmeras técnicas de análise intradocumental que visam obter *fatos* (blocos de informação relevante sobre o contexto tratado) ou ainda integrá-los e relacioná-los a fim de derivar novos fatos (GRISHMAN, 1997). Dentre os processos comuns presentes na disciplina de EI estão o reconhecimento de entidades, resolução de co-referências e as técnicas de análises léxica, sintática e semântica sobre o conteúdo textual (MOENS, 2006; BILOTTI, 2004).

A seção a seguir apresenta como os componentes e técnicas de Recuperação e Extração de Informação são organizados em conjunto nas arquiteturas tradicionais de *QA*.

2.3.2 Arquitetura típica de Question Answering

Em síntese, os componentes das arquiteturas de QA podem ser agrupados em três módulos: 1) módulo de processamento da pergunta – compreende as ferramentas e recursos para a análise da pergunta

conforme o idioma; 2) módulo de recuperação e extração de informação – com base na pergunta interpretada, destina-se a localizar os parágrafos ou fragmentos de informação que atendem à pergunta nas fontes de dados. Para tal, os sistemas de QA partem do princípio que as respostas estão armazenadas e encontram-se implícitas nas bases de dados; 3) Módulo de processamento da resposta – define os critérios de priorização, ordenação e *ranking* para determinar qual a melhor resposta dentre as possíveis encontradas. A Figura 6 demonstra como esses módulos são organizados nas arquiteturas de *QA* e ainda apresenta alguns dos principais métodos e técnicas sobre a área.

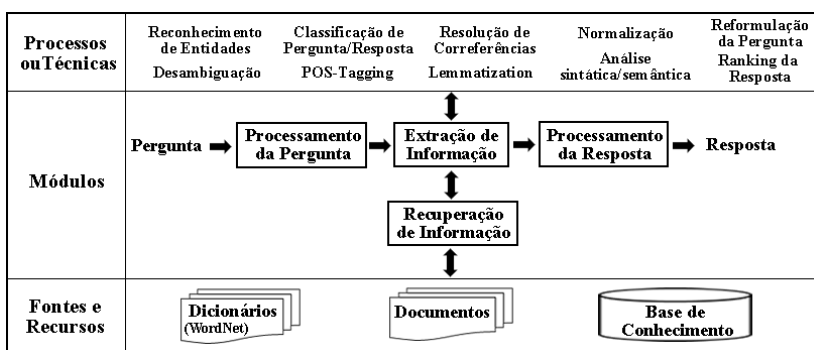


Figura 6 – Ilustração de uma arquitetura típica de Question Answering (Fonte adaptada: HIRSCHMAN e GAIZAUSKAS (2001); MOLDOVAN; TATU e CLARK (2009).

Na arquitetura exibida pela Figura 6 acima, as *Fontes e Recursos* representam os dicionários de sinônimos, fontes de dados textuais, bases de conhecimento, ontologias, *thesaurus*, etc. que são utilizados pela maioria das pesquisas para análise da pergunta e processamento da resposta. Os *Processos* ou *Técnicas*¹⁹ em destaque na figura são conceituados a seguir:

Tokenization e Normalização: a tarefa denominada *Tokenization* consiste em dividir uma ou mais sentenças contidas num documento em *Tokens* (*token* é uma instância de caracteres que representam uma unidade semântica útil para processamento) e também desprezar caracteres considerados não relevantes para o contexto de aplicabilidade, tais como sinais de pontuação (vírgula, ponto-e-vírgula,

¹⁹ O termo *Processos ou Técnicas* é usado para abranger as tarefas, algoritmos, processos e técnicas da área.

etc.). A partir da tarefa da *Tokenization*, cada unidade textual identificada pode sofrer ainda um processo chamado *Normalização* (em inglês, *Normalization*). Esse processo transforma tokens com diferenças superficiais (como acentos, hífens, etc.) tal como *anti-discriminatório* e *antidiscriminatório*, em uma forma equivalente e única (KOWALSKI, MAYBURY, 2000; MANNING; RAGHAVAN, SCHÜTZE, 2008). Note que esses processos são sensíveis ao idioma e também devem acompanhar as alterações, evoluções e acordos ortográficos de cada língua.

Consideração de Stop-words: *Stop-words* são termos bastante comuns e com alta frequência de ocorrência nos textos. São os exemplos das preposições, dos artigos e dos pronomes. Embora esses termos sejam desconsiderados em boa parte dos métodos de RI (MANNING; RAGHAVAN, SCHÜTZE, 2008), eles podem ser utilizados para: a) identificação de idiomas (i.e. stop-words como *what, the, of, why, from* podem identificar a língua inglesa assim como *quoi, il, des, pourquoi*, identificam o idioma francês, e assim por diante); b) para classificação de perguntas e respostas nas tarefas de QA; c) para identificação de padrões e aplicação de heurísticas com base na língua (HIRSCHMAN, GAIZAUSKAS; 2001; LOPEZ, et. al.; 2007; MOLDOVAN; TATU; CLARK, 2009).

Reconhecimento de entidades (em inglês, Named-Entity Recognition): técnica de EI que visa à identificação ou classificação de entidades de diferentes tipos em classes predefinidas como Pessoa, Tempo, Cidade, Organização, dentre outras. Normalmente conta com o auxílio de dicionários, taxonomias e também ontologias para a categorização e desambiguação de entidades textuais. Vale ressaltar a complexidade dessa técnica em identificar o início e fim (fronteiras) das entidades compostas por mais de um *token*, por exemplo, nas palavras “New York” ou “New York Times” (MAYNARD; BONTCHEVA; CUNNINGHAM, 2003).

Resolução de co-referências (em inglês, Coreference Resolution): tarefa que busca a identificação de expressões anafóricas no texto, na qual é usualmente executada após a aplicação da técnica de Reconhecimento de Entidades. Ou seja, essa tarefa reconhece as terminologias escritas de modo distinto que, no entanto, referenciam a uma mesma entidade. Por exemplo, na oração “O presidente dos EUA viaja a Londres amanhã. A capital da Inglaterra receberá Barack Obama com festa.” os termos “*presidente dos EUA*” e “Barack Obama” representam a mesma entidade no texto, assim como “*Londres*” e “*capital da Inglaterra*”. O objetivo principal da tarefa de resolução de

co-referências é identificar essa relação entre os termos. (QUARTERONI, 2007).

Part-Of-Speech Tagging (POS-Tagging): também denominado *Word-category disambiguation*, consiste na tarefa de classificar os *tokens* segundo as classes gramaticais do idioma. Dentre essas classificações cita-se: substantivo, artigo, pronome, verbo, advérbio, numeral, etc (MOENS, 2006).

Reformulação e Expansão de Perguntas (em inglês, *Reformulation and Question Expansion*): técnica também utilizada por sistemas de RI que propõe alternativas às consultas do usuário por meio de uso de sinônimos, hierarquia e relações semânticas entre conceitos a fim de expandir a pergunta, explicitar informações omitidas e converter em termos formais o que o usuário procura. O uso de relações de sinonímia, hiponímia, meronímia e hierarquia de conceitos são tarefas usuais em QA baseados em ontologias. Isto porque o usuário nem sempre informa termos exatamente iguais aos conceitos modelados na base de conhecimento. A reformulação e expansão de perguntas podem ser também tarefas interativas que requerem em alguns casos a participação do usuário para melhorar a pergunta inicialmente imposta (LOPEZ; et. al., 2007; WANG; et. al., 2007; MANNING; RAGHAVAN, SCHÜTZE, 2008).

Desambiguação (em inglês, *Word-Sense Disambiguation*): constitui a tarefa de identificar e resolver as ambigüidades presentes nas entidades textuais conforme o domínio. A ambigüidade é gerada quando a entidade reconhecida possui dois ou mais significados (como homônimos) e; portanto, pode ser associado a mais de uma classe dentro do contexto tratado (MOLDOVAN; TATU; CLARK, 2009). Como exemplo cita-se a entidade “São Paulo” na qual pode estar associada a um time de futebol, a uma cidade ou ainda um estado brasileiro. O uso de ontologias já é aplicado nas pesquisas de QA para a classificação de termos com bases nas classes de domínio e também para a resolução de ambigüidades (NIRENBURG; RASKIN, 2004; KAUFMANN; BERNSTEIN, 2007; LOPEZ; et. al., 2007).

Lematization e Stemming: ambas as técnicas visam à redução das flexões ou formas relativas de palavras para uma base comum ou para o seu radical (MANNING; RAGHAVAN, SCHÜTZE, 2008). Por exemplo: em vez de usar as palavras *estudante*, *estudando*, *estudavam* poderia considerar apenas a forma *estudar*.

Classificação de perguntas/respostas: No âmbito de QA é comum que haja um tratamento diferenciado das respostas de acordo com o tipo da pergunta. Deste modo, consoante o tipo ou classe da pergunta informada

resultará em uma resposta distinta. Por exemplo, na pergunta “*Onde Abraham Lincoln nasceu?*” espera-se como resposta um local geográfico (cidade, estado ou país) já na variante “*Quando Abraham Lincoln nasceu?*”, a resposta deveria ser uma entidade diferente que representa o tempo. Vide os diferentes tipos de classificações na seção 2.3.3. (HIRSCHMAN, GAIZAUSKAS; 2001; LOPEZ; et. al., 2007)

Análises sintática e semântica: normalmente são tarefas que atuam em conjunto com as descritas anteriormente para determinar as características dos relacionamentos entre os entidades textuais ou *tokens* das sentenças. Na análise sintática tenta-se identificar as árvores ou funções sintáticas dos termos nas orações como o sujeito, predicado, objeto direto, indireto, dentre outros e ainda os padrões lingüísticos. Na análise semântica, os relacionamentos de sinonímia, hipônimos, hiperônimos, antônimos, merônimos, associativos, etc., são estabelecidos. Devido à expressividade de representação de relacionamentos entre conceitos, novamente as iniciativas demonstram o direcionamento para o uso de ontologias (KAUFMANN; BERNSTEIN, 2007; LOPEZ; et. al., 2007).

Ao longo do tempo, percebe-se que os estudos de Question Answering evoluíram de acordo com a necessidade de obtenção de informações além de domínios específicos de conhecimento. Inicialmente, essas pesquisas eram aplicadas somente sobre os repositórios estruturados e focavam em um determinado contexto de negócio da organização (HIRSCHMAN, GAIZAUSKAS; 2001). Com o surgimento da Web, boa parte dessas pesquisas não mais se limitou apenas ao uso de bases estruturadas e de domínios fechados de conhecimento. Houve uma expansão do universo de consultas sobre os mais variados tipos de conteúdo, que hoje vão além de textos publicados na Internet e nas intranets corporativas, chegando até mesmo a recuperação de imagens, áudios e vídeos. Com isso, os sistemas de QA evoluíram para diversos tipos de aplicação. A seção posterior explana a respeito dos tipos de QA e situa onde este trabalho classifica-se.

2.3.3 Tipos de Question Answering

A classificação dos sistemas de Question Answering ajuda a compreender as principais diferenças e complexidades entre as técnicas e abordagens utilizadas. Em geral, eles podem ser classificados segundo os mais variados critérios como:

Tipos de fontes de dados: dividem-se em sistemas de QA que atuam sobre fontes estruturadas (mais conhecidos como interfaces de *front-end para banco de dados*); fontes semi-estruturadas e não estruturadas (tais como documentos textuais, páginas da Web, etc). Sobre os tipos de fontes de dados, subdividem-se ainda quanto a sua heterogeneidade de conteúdo, que pode concentrar-se em informações puramente textuais a abranger também imagens, vídeos dentre outros formatos (HIRSCHMAN, GAIZAUSKAS; 2001). Este trabalho adota as bases estruturadas, mais especificamente Data Warehouse (ou Data Marts), como fontes de consulta conforme declarado na delimitação de escopo. Deste modo, as informações dispersas em outros tipos de fontes de dados devem ser integradas ao DW para que sejam combinadas para a tomada de decisão.

Domínio de aplicação: distinguem-se em domínio fechado (em inglês, *closed-domain question answering*), que dão enfoque a responder sobre uma determinada área específica de interesse e; domínio aberto (em inglês, *open-domain question answering*), que se tornaram mais evidentes a partir da década de 90 com os serviços Web e tratam de responder a um universo maior de perguntas sem limitação de escopo a priori (QUARTERONI, 2007). A partir dos formalismos propostos na Web Semântica, as pesquisas de domínio fechado, na qual se situa este trabalho, evoluíram para o uso de ontologias (MCGUINESS, 2004). Porém, há propostas também que utilizam ontologias em rede para abranger um contexto maior em domínios abertos (LOPEZ, et. al., 2007).

Classes de perguntas e respostas: são os sistemas classificados conforme os tipos ou taxonomias de perguntas e tipos de respostas nos quais atendem. É comum um mesmo sistema atender mais de um tipo de pergunta e resposta. Os tipos de perguntas e respostas subdividem-se em: *booleanas*, que tratam de responder sim/não ou certo/errado; *factuais* (perguntas que normalmente iniciam com os pronomes interrogativos do tipo *Qu*, tais como *Qual*, *Quem*, *Quando*, *Quanto*, etc.), em que o retorno está dentro de um categoria previsível relacionado a uma entidade do domínio (pessoa, localidade, tempo, etc.); *opiniões* ou *definições* – correspondem a perguntas que direcionam a respostas normalmente descritivas ou a um parágrafo sobre um dado assunto; *comandos* - declaram instruções imperativas por exemplo nas expressões *mostre-me* e *liste-me*, etc (HIRSCHMAN, GAIZAUSKAS; 2001). Salienta-se que a arquitetura de BI proposta neste trabalho aborda as perguntas e respostas factuais. As perguntas e respostas *factuais* visam à obtenção de informações sumarizadas e fatos

quantitativos ou medidas do DW conforme as entidades identificadas na ontologia de domínio.

Perfil do usuário: Conforme declara Burger (et. al., 2001) os usuários em sistemas de *QA* variam de usuários casuais – que o utilizam de forma similar a buscas textuais baseadas em palavras-chave; e vão até analistas profissionais – que o utilizam para obter informações em detalhes com perguntas complexas. Por limitação de escopo, este trabalho não considera os diferentes perfis dos tomadores de decisão e, portanto as consultas e resultados não sofrem alterações conforme o usuário solicitante.

2.3.4 Iniciativas de Question Answering baseadas em tecnologias semânticas

Assim como em BI, as tecnologias semânticas inspiradas na Web Semântica são adotadas como forma mais sofisticada para desenvolver frameworks de *QA* que permitem: a) oferecer proativamente informações adicionais a respeito da resposta; b) prover medidas de confiabilidade e melhorar as medidas *precision* e *recall*; c) explicar como a resposta foi derivada (LOPEZ; et. al., 2007; MCGUINNESS, 2004). Além do suporte para as tarefas de classificação de perguntas e reconhecimento de entidades, tais tecnologias são empregadas também para: expandir e refinar perguntas; anotação semântica de documentos; criação de índices textuais para a recuperação e extração de respostas; desambiguação de conceitos com base no domínio e; descoberta de relacionamentos entre conceitos e raciocínio sobre bases de conhecimento (KAUFMANN; BERNSTEIN, 2007; LOPEZ; et. al., 2007; WANG, et. al., 2007; MOLDOVAN; TATU; CLARK, 2009). Logo a seguir, algumas iniciativas de *QA* que se fundamentam nas tecnologias semânticas tomadas como base desta pesquisa são resumidas:

PowerAnswer: esse sistema de *QA* utiliza um conjunto de sete camadas hierárquicas (*Concepts; Semantic Relation; Contexts; Event Structure; Event Relation; MacroEvent*) para a representação de conhecimento a partir de textos. A arquitetura do *PowerAnswer* é formada pelos seguintes componentes semânticos organizados nos três módulos clássicos das arquiteturas tradicionais de *QA* vistos na seção 2.3.2: módulo para resolução de ambigüidades; bases de conhecimento baseados em *WordNet, EventNet e Wikipedia*; módulo de identificação

de relações (temporais ou não) entre conceitos e eventos com base no domínio definido pela ontologia; módulo de criação e manutenção de ontologias a partir de textos (chamado de *Jaguar*); mecanismo de inferência (denominado *Cogex*) para identificação de respostas e raciocínio (MOLDOVAN; TATU; CLARK, 2009).

Aqua, AquaLog e PowerAqua: esses três sistemas de QA são versões sucessivas propostas por Lopez (et. al., 2007). As duas primeiras versões, *Aqua* e *AquaLog*, são semelhantes em termos arquiteturais. No entanto, *AquaLog* estende as funcionalidades da versão anterior e aprimora os métodos de desambiguação e a conversão da pergunta em uma representação aderente ao modelo da ontologia. Ambas as versões, utilizam ontologias para: a) classificação conforme os tipos de perguntas declarados na seção 2.3.3 b) reformulação e expansão da pergunta; c) em processos de raciocínio específicos no retorno da resposta; d) na busca por similaridade entre relacionamentos dos termos identificados na pergunta e os conceitos presentes na base de conhecimento. *Aqualog* possui ainda módulos adicionais para integração com Web Services. Já em sua última instância, *PowerAqua* propõe-se a responder as perguntas em um domínio aberto em que múltiplas ontologias são utilizadas em rede. As necessidades dos usuários são mapeadas em uma ontologia conforme o contexto da pergunta para que recursos possam ser localizados na Web Semântica (LOPEZ; et. al., 2007).

NLP-Reduce, Querix, Ginseng e Semantic Crystal: todos são sistemas de QA baseados nos formalismos da Web Semântica (RDF, OWL, SPARQL) propostos por Kaufmann e Bernstein (2007). NLP-Reduce constitui a abordagem de QA mais simples proposta por esses autores. Ela permite que usuários entrem como palavras-chave, fragmentos de sentença ou perguntas completas e emprega uso de sinônimos e técnicas de *stemming* para refinar essa entrada. Triplas com as relações dos termos da pergunta são criadas, tal como no modelo RDF, e comparadas via consultas na base de conhecimento por meio de SPARQL. Já o sistema Querix adota os mesmos procedimentos do NLP-Reduce, no entanto usa perguntas completas, técnicas de *POS-tagging* e incorpora um método interativo com o usuário para a resolução de ambigüidades. *Ginseng* (Guided input natural language search engine) aplica uma linguagem controlada (forma que impõe vocábulos específicos e limita o uso de linguagem natural) baseada em menus sobre bases de conhecimento em OWL. *Ginseng* utiliza métodos de reformulação e expansão para sugerir complementações durante a entrada da pergunta do usuário e ainda, possibilita a anotação semântica dinâmica para adição e sinônimos dos conceitos da ontologia. Esse

sistema também traduz as perguntas em consultas SPARQL que são executadas diretamente sobre as bases de conhecimento para encontrar as respostas. Por fim, *Semantic Crystal* permite ao usuário manipular e selecionar os conceitos e relacionamentos da ontologia graficamente. Nessa abordagem apesar de aplicar as técnicas dos demais sistemas de QA, os autores propõem uma interface em que as consultas podem ser criadas de modo interativo a partir do modelo da ontologia, em vez de uma entrada em linguagem natural (KAUFMANN; BERNSTEIN, 2007). Esses autores concluem que entre os sistemas, o uso de linguagem natural presente no sistema Querix é em geral o melhor aceito pelos usuários.

PANTO (Portable nAtural laNguage inTerface to Ontologies): tal como as propostas de Lopez (et. al., 2007) descritas acima, *PANTO* é um sistema de QA que formaliza as perguntas numa estrutura semelhante ao modelo de triplas de RDF que é comparada ao modelo da ontologia de domínio. A ontologia é utilizada por alguns componentes da arquitetura principalmente para reconhecimento de entidades; determinar as projeções, filtros e relacionamentos entre conceitos para extrair a resposta a partir de consultas em SPARQL (WANG, et. al., 2007).

QuestIO e FREyA: o primeiro sistema, proposto por Damljanovic, Agatonovic e Cunningham (2010), *QuestIO* (acrônimo de **Q**uestion-based **I**nterface to **O**ntologies) atua sobre domínios fechados definidos por ontologias para converter a pergunta em linguagem natural em uma consulta SPARQL. Essa consulta é executada sobre a base de conhecimento para que a resposta envolvendo os conceitos do domínio seja retornada. As consultas são executadas diretamente sobre o conjunto de instâncias do modelo da ontologia e; portanto, os dados das fontes de dados (documentos, bases de dados) precisam ser carregados para o modelo. *FreyA* (Feedback, Refinement and Extended Vocabulary Aggregation) toma como base os métodos de *QuestIO* e em síntese, aplica os três seguintes passos: identificação e verificação dos conceitos da ontologia a partir da pergunta em linguagem natural; geração de uma consulta na sintaxe SPARQL e; identificação do tipo de resposta e apresentação dos resultados ao usuário. *FreyA* supre algumas deficiências de *QuestIO* como: melhora na compreensão da semântica da pergunta; fornece uma resposta mais concisa e exata às perguntas do usuário e provê maior interação e mecanismo de aprendizado para auxílio a interpretação das perguntas (DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2010).

ORAKEL: pesquisa de QA semelhante aos sistemas de Lopez (et. al., 2007) e Wang (et. al., 2007) que se baseia em ontologias para criar uma

estrutura formal de representação da pergunta e para obter respostas factuais diretas ou ainda derivá-las por meio de processos de inferências a partir de bases de conhecimento. Adota as linguagens de consulta SPARQL e ainda F-Logic para obter os fatos e conceitos a partir da interpretação da pergunta (CIMIANO; et. al., 2007).

Personalized QA Framework: organiza os módulos semânticos em uma arquitetura dividida conforme os três módulos clássicos das arquiteturas de QA. Com base na ontologia, utiliza formas de anotação semântica sobre documentos, classificação e expansão de perguntas, e recuperação de informação sobre índices do motor de busca *Lucene* (THAI; et. al., 2006). Também se baseia em respostas factuais em estudo de caso no contexto de BI.

2.3.5 Iniciativas de Question Answering no contexto de Business Intelligence

Comparado à área de Business Intelligence (BI), as abordagens de Question Answering (QA) podem facilitar também o acesso a informações contidas nas fontes de dados corporativas para apoio à gestão do conhecimento e à tomada de decisão. Considerando as abordagens convencionais, embora possam usar métodos distintos, essas áreas de conhecimento distinguem-se na forma como a informação é solicitada e também como o seu retorno é disposto para o usuário. Como visto na seção 0, tradicionalmente quando se trata de BI, todas as informações da organização devem ser integradas no DW para melhor serem obtidas a partir dos ferramentais OLAP, nos quais apresentam a informação resumida em diferentes formatos (tabulares, hipercubo, sob gráficos, etc.). Em geral, para que fiquem disponíveis ao gestor, todas as possibilidades de cruzamentos de informações sobre o DW e as prováveis operações OLAP sobre o seu conteúdo devem ser previamente conhecidas e pré-configuradas nas ferramentas OLAP. Além disso, cabe ao gestor saber lidar com todas as funções e capacidades de uma ferramenta OLAP específica para poder tirar proveito de seu potencial e realizar as análises.

Já na área de QA, em vez de ter que conhecer como manusear uma ferramenta específica, o usuário deve apenas informar uma única pergunta em linguagem natural para obter a informação que precisa. Conforme a definição conceitual clássica de QA, o retorno esperado é uma única sentença textual simples e exata na qual responde a uma

pergunta. No entanto, sabe-se que além dessa modalidade de Question Answering baseado em texto, existem outras que tratam do uso não apenas de sentenças textuais para responder às perguntas em linguagem natural. Alguns trabalhos acadêmicos e principalmente comerciais já oferecem outras formas de resposta para o usuário tais como imagens, áudios e até vídeos relacionados com a pergunta tal como proposto no sistema *Wolfram Alpha*²⁰. Ressalta-se que diferentemente da recuperação de uma lista de documentos textuais, áudios ou vídeos obtidos por meio de buscas por palavras-chave, esses trabalhos abordam o uso de perguntas para obter uma resposta mais precisa. Assim, a integração de Question Answering com a área de *Business Intelligence* baseia-se nessas outras modalidades para obter novas visões de análise para o gestor de modo mais simples e amigável do que por meio de ferramentas OLAP convencionais.

Em análise empírica sobre os trabalhos que unificam QA e BI observa-se que há cenários tanto para a utilização de fontes heterogêneas (estruturadas ou não estruturadas) e ainda a integração destas. No âmbito mais próximo a esta pesquisa, na qual a fonte de dados é o *data warehouse*, existem algumas soluções que já permitem a exibição de informações como o usuário tomador de decisão está acostumado nas interfaces analíticas. Isto é, a informação, após ser obtida das fontes de dados por meio de uma pergunta, pode ser visualizada em formatos tabular, *hipercubo* ou ainda por meio de gráficos (em forma de gráfico de pizza, de barras, séries históricas, dentre outros). Nesses casos, a resposta não necessariamente é uma sentença textual, e normalmente contém informações sumarizadas e agrupadas conforme a pergunta informada pelo usuário. No caso do formato tabular, a resposta não precisa conter uma única informação em uma coluna ou em uma linha. Conforme as operações OLAP, a resposta da tarefa de Question Answering pode ter várias informações agrupadas de inúmeras dimensões do DW para serem apresentadas em forma de um hiper-cubo ao tomador de decisão. *Semantra* e *EasyAsk* (EVELSON; BROWN, 2008; ECKERSON, 2010) são dois exemplos de trabalhos comerciais relacionados em que as características dos sistemas de QA e BI citadas podem ser observadas na prática.

Conforme a análise sobre as iniciativas mencionadas, ao longo do capítulo 3, apresenta-se a arquitetura proposta. Nesse próximo capítulo, pontuam-se onde os trabalhos relacionados de QA e BI são enquadrados

²⁰ Wolfram Alpha - <<http://www.wolframalpha.com>>

conforme as etapas e os módulos funcionais da arquitetura. Para facilitar a compreensão das tarefas e dos componentes propostos, alguns exemplos de perguntas são apresentados nas seções respectivas de cada módulo funcional adiante.

3 ARQUITETURA PROPOSTA

A fim de realizar o processamento analítico, a arquitetura proposta baseia-se na conjugação entre as pesquisas de *QA* e os frameworks semânticos de *BI* descritos anteriormente. Isto é, a arquitetura integra as diferentes técnicas, processos e sistemas já desenvolvidos nos trabalhos relatados para compor uma nova abordagem interdisciplinar para obtenção de conhecimento. No entanto, alguns componentes e técnicas utilizados foram adaptados para atender ao contexto de aplicação desta pesquisa, na qual envolve a recuperação de informações estratégicas a partir de linguagem natural sobre fontes estruturadas para o apoio à tomada de decisão.

3.1 VISÃO GERAL

O processamento das requisições em linguagem natural é realizado a partir dos módulos da arquitetura em três etapas principais: 1) uma etapa associada à construção e manutenção do modelo e base de conhecimento, na qual é fundamental para as etapas subsequentes; 2) uma segunda etapa relacionada à interpretação da pergunta e a sua formalização em uma estrutura que represente o seu significado; 3) uma terceira e última etapa responsável por retornar a resposta, que nesta pesquisa está sob a forma de um cubo OLAP.

A primeira etapa ocorre previamente ao processo decisório e deve ser executada regularmente conforme a evolução da ontologia de domínio e o crescimento e as alterações das fontes de dados da organização. Ela visa à preparação das ontologias e da base de conhecimento utilizadas tanto no processo de análise e interpretação da pergunta quanto no retorno das informações estratégicas do DW. A segunda etapa envolve em sua maior parte os estudos da área de *QA*. Nessa etapa, a pergunta informada em linguagem natural é analisada e processada por um conjunto de métodos e tecnologias semânticas conforme o contexto definido pela ontologia do domínio da organização. Aqui, algumas tarefas de *QA* são aplicadas para a interpretação da pergunta visando à tradução da linguagem natural em uma linguagem formal. Essa representação formal da pergunta resultante possui a definição das medidas quantitativas, os agrupamentos descritivos e os filtros para a execução das operações OLAP (e.g. *drill-*

down, roll-up, slice, dice, etc.). Uma vez construída e formalizada a consulta, a última etapa executa o processamento OLAP sobre a fonte de dados para obter a sumarização das informações e responder a pergunta. Dado que esta pesquisa é aplicada sobre bases estruturadas – *data warehouses ou data marts*, o resultado da consulta já é a resposta final. Assim, esse resultado não necessita passar por métodos de RI ou EI como nos estudos de QA tradicionais, restando a sua formatação e visualização na ferramenta OLAP.

As três etapas são detalhadas conforme a descrição e a interação entre os componentes da arquitetura adiante. Ao longo das seções deste capítulo, são usados alguns exemplos de perguntas relacionadas ao contexto de C&T. Essas perguntas introduzem algumas terminologias e conceitos sobre esse tema, tais como as atividades acadêmicas de discentes, docentes, seus vínculos com as instituições de ensino, seus níveis de escolaridade, produções bibliográficas, dentre outros. Os exemplos servem também para familiarizar o leitor sobre os principais termos desse domínio que é tratado e modelado no capítulo 4 do protótipo da arquitetura.

A Figura 7 a seguir exibe a disposição dos elementos constituintes da arquitetura proposta. Os elementos distinguem-se em: **processos e técnicas** – representam as tarefas, procedimentos e processos desempenhados pelos módulos funcionais da arquitetura; **entradas e saídas** – são os dados de entrada e resultados dos processos e técnicas; **módulos funcionais propostos** – embora não inéditos, são os subsistemas inerentes à arquitetura ou componentes desenvolvidos por terceiros que possuem peculiaridades em alguns papéis que desempenham e; **repositórios e fontes** – compreendem os repositórios de ontologias, modelos e base de conhecimento, itens de configuração e ainda as fontes de dados da arquitetura (DW).

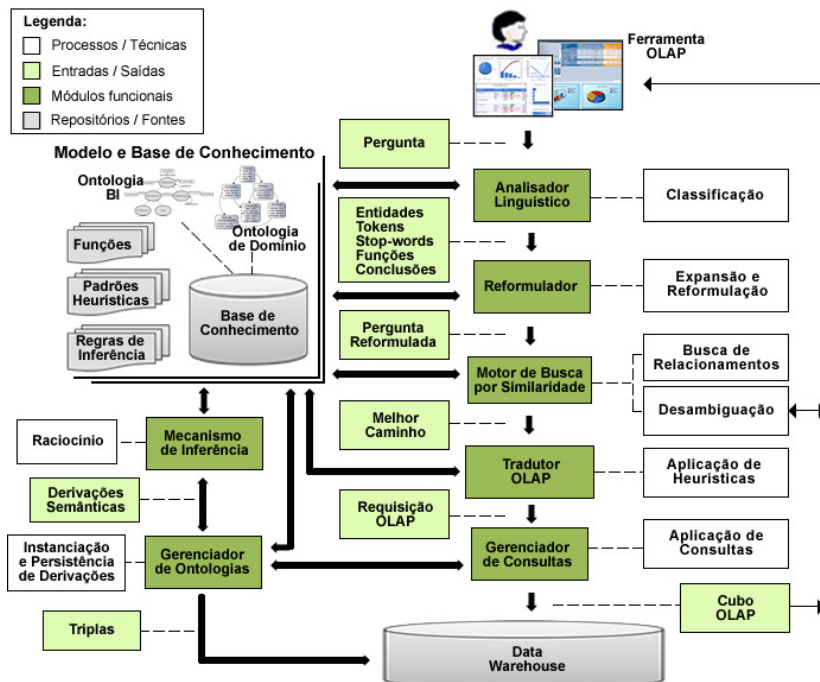


Figura 7 - Arquitetura de Business Intelligence proposta

Conforme ilustra a Figura 7, a partir de uma ferramenta analítica, na qual o usuário expressa uma necessidade de informação em linguagem natural, um conjunto de módulos sucessivos interagem até a obtenção do cubo OLAP. Em resumo, a pergunta é inicialmente avaliada pelo *Analizador Linguístico* que identifica as entidades textuais, *stop-words* e outras classificações declaradas adiante na seção 3.3. A partir disso, o módulo *Reformulador* emprega as técnicas de *Query Expansion* e *Reformulation* para detalhar e expandir a pergunta inicialmente proposta. Com base na pergunta reformulada, os relacionamentos entre os conceitos da pergunta são comparados de acordo com o modelo da ontologia do domínio pelo *Motor de Busca por Similaridade*. Uma vez descoberto o melhor caminho (conjunto de relacionamentos) entre os conceitos na ontologia de domínio, o *Tradutor OLAP* converte a pergunta em uma requisição formal, contendo as definições de filtros, projeções e medidas quantitativas a serem consideradas na consulta. Então, o *Gerenciador de Consultas*, com o auxílio do *Gerenciador de Ontologias*, executa a requisição sobre

o DW da organização para retornar o cubo OLAP com as informações estratégicas ao tomador de decisão. Nesse retorno as derivações semânticas provenientes do *Mecanismo de Inferência* podem ser combinadas com as informações do DW. O *Modelo e Base de Conhecimento* funciona como elemento central na qual os módulos funcionais citados são dependentes para concluir cada tarefa.

Embora a pergunta original sofra alterações ao longo do processo de interpretação, todas as informações usadas como entrada ou saída de uma tarefa anterior não são descartadas, dado que podem ser utilizadas a frente por outros módulos. A seção a seguir descreve os componentes da arquitetura proposta em detalhes.

3.2 MODELO E BASE DE CONHECIMENTO

O Modelo e Base de Conhecimento é o núcleo essencial para o correto funcionamento dos módulos e processos da arquitetura. Sua má construção implica em falhas em praticamente todos os processos e por consequência, pode produzir análises imprecisas e tomadas de decisão erradas. Esse elemento ou módulo da arquitetura é um repositório de recursos de representação de conhecimento na qual é composto por:

- a) **Ontologia de Domínio** – utilizada por todos os trabalhos relacionados a esta pesquisa, é a principal forma de representação do contexto do negócio da organização. A ontologia de domínio, que neste trabalho é codificada na linguagem OWL, modela os principais conceitos e os relacionamentos necessários para a condução das análises e posterior navegação sobre as fontes de dados da organização. Essa ontologia deve descrever os possíveis sinônimos, hierarquias ou taxonomias, propriedades e os relacionamentos das classes para a construção de instâncias dos conceitos do domínio. Tais instâncias são criadas a partir do conteúdo dos repositórios de dados (DW) através de alguns módulos detalhados adiante e são mantidas na base de conhecimento da arquitetura. Os conceitos dessa ontologia servem também para a criação de regras de inferência, nas quais são descritas logo adiante.
- b) **Ontologia BI** – utilizada no trabalho de Sell (2006; et. al., 2008) e Silva (2006), é responsável por dar suporte ao mapeamento entre os conceitos estabelecidos na ontologia de domínio e a estrutura das fontes de dados. Essa ontologia trata a correspondência de cada

classe ou propriedade modelada na ontologia de domínio com as tabelas de fato, as dimensões e seus atributos presentes no DW. Por meio desse mapeamento é possível que as instâncias da ontologia de domínio sejam criadas de forma automática na base de conhecimento, já que os valores das propriedades das classes necessários para instanciação podem ser extraídos das fontes de dados. Além disso, esse mapeamento permite que a exploração sobre as fontes de dados sejam conduzidas por meio da ontologia de domínio. Sell (et. al., 2008) divide essa ontologia em duas partes: 1) uma parte com conceitos analíticos utilizados para auxiliar a navegação da ferramenta OLAP (classes como *Theme*, *AnalysisUnit*, *Measure*, *Dimension*, dentre outras) e; 2) parte relacionada aos conceitos que modelam a estrutura das fontes de dados e sua relação com os conceitos da ontologia de domínio (como *Collection*, *Attribute*, *CollectionJoin*, etc.). Apenas a segunda parte da Ontologia BI citada é utilizada neste trabalho. Todas as instâncias dessa parte do modelo da ontologia possuem construtores para associar: a) classes da ontologia de domínio às dimensões (ou às tabelas de fato) do DW; b) propriedades de classes aos atributos de dimensão; c) relacionamento entre classes às junções entre dimensões e tabelas. No contexto desta pesquisa, essa ontologia é estendida para que determinadas propriedades de classes sejam definidas como padrão para a criação da consulta. Conforme é explanado na seção 3.6 adiante, é necessário identificar as medidas, agrupamentos e filtros para a construção da consulta OLAP. Dessa forma, a Ontologia BI mapeia também as propriedades padrão de cada classe de domínio que devem ser utilizadas na identificação desses elementos da consulta. Além disso, essa ontologia indica os métodos de quantificação para as medidas (similar às funções *SQL*), como somatório, contagem, média, valor máximo ou mínimo. Para mais informações sobre a Ontologia BI, vide Sell (2006; et. al., 2008) e Silva (2006).

- c) **Base de Conhecimento** – possui o conjunto de instâncias da Ontologia de Domínio e da Ontologia BI deste trabalho. As instâncias de ambas as ontologias podem ser armazenadas em locais ou formatos distintos do modelo OWL. Assim, a Base de Conhecimento pode ser construída fisicamente, por exemplo, em bases de dados relacionais, o que é útil quando há uma grande quantidade de instâncias. Mesmo com todas as instâncias mantidas em locais diversos, elas devem ainda obedecer ao modelo de ontologia a qual estão associadas. A Base de Conhecimento é

utilizada principalmente nos processos de inferências, em que um conjunto de regras de inferência sobre as instâncias de classes são aplicados para apoiar às análises.

- d) **Regras de inferência** - são as regras de negócio da organização definidas com base na ontologia de domínio. As regras de inferência visam à explicitação de novos relacionamentos ou conceitos a partir da ontologia de domínio como proposto nos trabalhos de Labvic, Vasilecas, Rupnik (2010), Sell (2006; et. al., 2008) e Silva (2006). A arquitetura não limita quanto ao uso de uma linguagem específica para a formalização de regras. O Apêndice A expõe alguns exemplos da linguagem adotada para a definição de regras de negócio com base no mecanismo de inferência do framework Jena (MCBRIDGE, 2002). As regras de inferência são utilizadas neste trabalho para apoio ao processo de raciocínio conforme as duas abordagens *on-the-fly* e também *in-batch* discutidas na seção 0.
- e) **Funções** – consistem em bibliotecas de funções ou cálculos que são diretamente associados a alguns termos específicos da linguagem utilizada pelo tomador de decisão e também a um conceito do modelo da ontologia de domínio. Como na definição matemática, as funções destinam-se a produzir um resultado a partir de uma entrada (um termo da pergunta). Por exemplo, as palavras *Hoje*, *Amanhã*, *Ontem* podem ser vinculadas a funções relativas à classe *Tempo*. Essas funções podem ser calculadas com base no dia atual para que o seu valor resultante seja usado na pergunta. Neste caso, o resultado da função seria uma data (dia, mês ou ano) que representaria uma instância ou valor de uma propriedade de classe (e.g. *Tempo*). Outro exemplo, o termo *Aqui* pode ser relacionado a uma função para obter as informações da localização ou endereço da máquina do usuário (no exemplo, esta função poderia ser associada à classe *Geografia*). Neste trabalho as funções são definidas segundo uma sintaxe própria a fim de demonstrar a viabilidade dessa prática. Todas as funções são aplicadas pelo módulo funcional Tradutor OLAP apresentado na seção 3.6.
- f) **Heurísticas e Padrões** – representam as expressões regulares, padrões léxico-sintáticos, heurísticas determinadas por um especialista conforme o idioma e distância entre termos da pergunta e stop-words. Esses recursos devem ser definidos e codificados no Modelo e Base de Conhecimento para que o módulo Tradutor OLAP possa reconhecer quais são as medidas, agrupamentos e filtros a partir da pergunta. Tal como adotado por Lopez (et. al., 2007) no desenvolvimento dos módulos de análise linguística, neste

trabalho os padrões sintáticos e as stop-words identificadas na pergunta são úteis para reconhecer os elementos da consulta (medidas, filtros, projeções, etc.). Vide a seção 4.3.4 adiante para compreender como tais padrões e heurísticas são aplicadas no protótipo da arquitetura.

Nesta pesquisa, a primeira etapa, responsável pela criação do Modelo e Base de Conhecimento, caracteriza-se por ser um processo semi-automático. Isto porque ela requer trabalhos manuais, como a modelagem de ontologias, definição de regras de inferência, funções e heurísticas para reconhecimento de medidas, filtros e agrupamentos nas consultas OLAP. Contudo, também há possibilidade de trabalhos automáticos, como a criação de instâncias da base de conhecimento a partir do conteúdo das fontes de dados. Não é escopo desta pesquisa desenvolver mecanismos para a criação ou manutenção automática da ontologia de domínio e de bases de conhecimento. Assim, ela necessita de intervenção por parte do engenheiro do conhecimento para que seja constituída e evoluída ao longo de tempo. Para mais detalhes sobre trabalhos que lidam com a manutenção automática ou semi-automática de bases de conhecimento vide Ceci (et. al., 2010) e Ghisi (2008).

Uma vez construído o Modelo e Base de Conhecimento, o gestor já pode utilizar os processos e demais módulos da arquitetura para conduzir as análises sobre o repositório de dados a partir da ferramenta OLAP. A ferramenta OLAP funciona como componente externo à arquitetura e, portanto, este trabalho apenas aponta as características e as funcionalidades que ela deve contemplar para que o processamento analítico seja feito por meio de linguagem natural. Basicamente, a ferramenta OLAP participa de três momentos no fluxo do processo decisório: no início como interface para consulta, com a particularidade de permitir a entrada de uma pergunta; na interação com o usuário em alguns processos, como a desambiguação de conceitos e de relacionamentos do domínio e; por fim, na exibição do cubo OLAP, na qual já é sua função inerente. A seção posterior descreve o módulo de análise lingüística que atua após a entrada da pergunta na ferramenta OLAP.

3.3 ANALISADOR LINGÜÍSTICO

Após a pergunta ser informada, ela passa por um processo de análise léxica, sintática e semântica realizado pelo Analisador

Lingüístico ilustrado na Figura 7. O Analisador Lingüístico efetua um conjunto de tarefas para análise dos elementos textuais da pergunta a fim de classificá-los (com base em tipos e classes conhecidos a priori). A tarefa de classificação dos termos ou dos *tokens* da pergunta é bastante comum em sistemas de Question Answering. No entanto, este trabalho apresenta outras classificações para auxiliar a interpretação da pergunta e para aplicar as consultas OLAP sobre o DW. As classes podem ser agrupadas conforme as características léxico-sintáticas e semânticas conforme mostra a Tabela 2.

As propriedades ou características de cada elemento textual da pergunta determinantes na classificação são: a posição ou distância em relação a outro termo na pergunta, se é ou não stop-word ou ainda conceitos da ontologia de domínio. Os conceitos da ontologia de domínio distinguem-se em classes, instâncias de classes, propriedades ou relacionamentos. Este trabalho introduz ainda outras classificações aplicadas quando os termos não são identificados quanto a essas categorias citadas, como *Funções* ou *Conclusões de regras de inferência*. A Tabela 2 detalha as classificações utilizadas pelo Analisador Lingüístico.

Tabela 2 - Classificação dos elementos textuais da pergunta geradas pelo Analisador Lingüístico

Análise	Classificação	Descrição
Léxico-sintática	Stop-words	Representa as palavras com alta frequência de ocorrência em textos. Normalmente são preposições, artigos ou pronomes que devem ser mapeados anteriormente conforme o idioma. Neste trabalho, as stop-words possuem classificações específicas para auxiliar a identificação dos construtores das operações OLAP. Essas classificações são usadas pelo módulo Tradutor OLAP e são descritas na seção 3.6.
	Posição ou Ordem	Identifica a posição numérica ou ordem de cada termo da pergunta em relação a outro. Esses dados são

		úteis para a formalização de padrões e heurísticas que devem ser mantidos no Modelo e Base de Conhecimento.
	Função	Conforme explanado na seção 3.2, a função associa uma terminologia a um cálculo definido para um conceito da ontologia de domínio. É uma classificação proposta neste trabalho.
	Tokens não reconhecidos	Categoria que possui os <i>tokens</i> que não são identificados pelo <i>Analizador Lingüístico</i> , visto que nem sempre há uma classe determinada para os termos informados.
Semântica	Conclusão de regra de inferência	Categoria que determina se o termo está presente na implicação de uma regra de inferência, isto é, se o elemento textual está associado ao fato conseqüente ou a conclusão da regra de inferência. Veja exemplos na seção 0 que trata sobre os processos de inferência.
	Entidade ou Conceito do domínio	Define se o elemento textual representa uma classe específica da ontologia de domínio, a uma propriedade ou relacionamento entre classes, ou ainda, se o elemento textual é uma instância de uma classe.

As classificações dos termos da pergunta pelo Analisador Lingüístico contribuem posteriormente para que os módulos funcionais possam identificar as relações desses termos e o significado da pergunta com base no modelo da ontologia de domínio. Além disso, essa classificação auxilia no reconhecimento de medidas, filtros e agrupamentos e ligações entre elementos para a construção da consulta OLAP.

Para realizar a classificação dos termos da pergunta, o Analisador Lingüístico deve realizar algumas atividades comuns do processamento de linguagem natural, tais quais, POS-Tagging, Lemmatisation ou Stemming, Named-Entity Recognition, Coreference, e buscas a dicionários ou thesaurus na base de conhecimento, como explanado na seção 2.3. Por isso, os frameworks desenvolvidos e consolidados por outros estudos podem ser utilizados para conceber o Analisador Lingüístico. Não é escopo deste trabalho propor algoritmos, ou inovações no processamento de linguagem natural e sim utilizar as abordagens de engenharia do conhecimento que melhor se adaptam a esta problemática.

Durante a classificação quanto aos conceitos da ontologia de domínio, os elementos textuais da pergunta podem apresentar ambigüidades. Isto é, o Analisador Lingüístico pode identificar duas ou mais classificações para o mesmo termo da pergunta. Essas ambigüidades não são eliminadas pelo Analisador Lingüístico e são tratadas posteriormente pelo módulo Motor de Busca por Similaridade. O Analisador Lingüístico realiza um processo com ênfase em cada termo específico da pergunta sem focar na semântica das relações entre as palavras conforme o contexto do domínio. Já o Motor de Busca por Similaridade, ao verificar a relação entre as palavras e obter informações de contexto, pode reduzir ou até mesmo eliminar as ambigüidades presentes. Portanto, a desambiguação é postergada e realizada somente uma única vez por meio do Motor de Busca por Similaridade. Na prática, o Analisador Lingüístico identifica uma ambigüidade na pergunta quando:

1) A entidade textual é uma instância de duas ou mais classes da ontologia de domínio. Por exemplo, na pergunta “*Quantos alunos nasceram em São Paulo?*”. A entidade “*São Paulo*” poderia estar associada a um conceito que representa o local geográfico (cidade ou estado) ou ainda, ser um clube de futebol. Ou seja, “*São Paulo*” poderia ser hipoteticamente uma instância da classe *Cidade* ou *Estado* ou ainda da classe *ClubeEsportivo*. Note que a ambigüidade poderia ser resolvida pela semântica das relações entre as terminologias da pergunta caso, por exemplo, a ontologia de domínio explicita-se que os alunos possuem ligação somente com cidades por meio do relacionamento do nascimento (e.g. considerando a tripla *Aluno nasceramEm Cidade*). Ressalta-se que o Analisador Lingüístico apenas identificaria as ambigüidades que seriam resolvidas semi-automaticamente pelo Motor de Busca por Similaridade na análise de relacionamentos entre os conceitos.

2) A entidade textual é uma classe e possui semelhança com duas ou mais classes da ontologia. Este caso aparece quando duas ou mais classes tem o mesmo nome ou sinônimos em comuns e são mencionadas na pergunta. Por exemplo, na pergunta “*Quantos artigos foram publicados em 2009?*”, a palavra “*artigo*” pode se referir tanto a uma produção bibliográfica ou quanto a uma norma específica de uma lei ou estatuto. Logo, a palavra artigo pode ser um sinônimo para a classe *Produção* ou sinônimo para a classe *Lei*. Outrossim, o contexto definido pela ontologia de domínio deve ser usado para reconhecer qual o conceito do negócio a que o termo se refere.

3) A entidade textual é uma propriedade ou relacionamento e pertence a duas ou mais classes envolvidas na pergunta. Esse caso é comumente encontrado dado que os conceitos podem compartilhar as mesmas propriedades ou possuírem relacionamentos equivalentes para um dado contexto. Um exemplo claro é a propriedade *nome* que poderia ser compartilhada entre as classes *Pessoa* e *Organização*.

4) A entidade textual tem similaridade entre classes, instâncias, propriedades ou relacionamentos da ontologia de domínio. Isto ocorre quando o termo possui a mesma descrição textual de uma classe, uma propriedade ou também de uma instância de classe.

Uma vez executado o processo de análise lingüística e obtida todas as classificações para os termos, a pergunta pode ser reformulada ou ainda expandida por meio do módulo *Reformulador* descrito na seção a seguir.

3.4 REFORMULADOR

Após a obtenção das classificações léxico-sintáticas e semânticas dos elementos textuais, a pergunta passa por um processo de reformulação. Esse processo visa ao enriquecimento e possivelmente à expansão da pergunta original para que ela contenha todas as informações necessárias para a criação da requisição OLAP posteriormente. A reformulação também é um processo característico de sistemas de Question Answering. Ela trata de encontrar fatos importantes relacionados ao domínio que foram omitidos ou declarados de modo diverso pelo usuário e que devem ser incorporados para completar e formalizar a pergunta. Esse trabalho adota dois tipos de reformulação que podem ser aplicados sucessivamente. Um trata da reformulação baseada na hierarquia de classes e no uso de relações de

sinonímia e outro atenta para a reformulação baseada em regras de inferência.

3.4.1 Reformulação por hierarquia de classes e sinônimos

Dentre as técnicas empregadas para expandir e reformular uma pergunta, o uso de dicionário de sinônimos revela-se uma prática muito comum na literatura. Observa-se que os sinônimos na reformulação da pergunta são usados principalmente de duas maneiras. Ora os sinônimos podem substituir totalmente um ou mais termos ora podem ser incluídos na pergunta original para melhorar as buscas sobre fontes de dados. No primeiro caso, uma dada terminologia que foi informada na pergunta é trocada por outra que está mais aderente ou mais próxima ao contexto tratado. Como neste trabalho o contexto é modelado por meio da ontologia de domínio, o sinônimo utilizado seria aquele mais relacionado a uma classe, uma propriedade, um relacionamento ou até mesmo uma instância. Por exemplo, na pergunta “*Quantos professores trabalhavam na UFSC em 2008?*”, poder-se-ia fazer uma reformulação ao ponto de substituir as palavras “*professores*”, “*trabalhavam*”, “*UFSC*” pelos respectivos sinônimos, “*docentes*”, “*lecionavam*”, “*Universidade Federal de Santa Catarina*”. Este exemplo traz o uso de sinônimos para classes (e.g. *Professor* ou *Docente*), sinônimos para relacionamentos (e.g. relação *trabalhaEm* ou *lecionaEm*) e ainda, sinônimos para instâncias de classe (e.g. *UFSC* ou “*Universidade Federal de Santa Catarina*” como instância da classe *Instituição*). Assim, considerando neste exemplo que os sinônimos utilizados são os mais apropriados para o contexto, a pergunta resultante do processo de reformulação seria “*Quantos docentes lecionavam na Universidade Federal de Santa Catarina em 2008?*”.

Já no segundo caso, em vez de substituir os termos pelos seus sinônimos, inclui-se o sinônimo preservando a palavra original na pergunta. A inclusão de sinônimos é característica de sistemas que utilizam motores de busca para recuperar documentos mais relacionados com a pergunta. Essa inclusão normalmente tem a função de reduzir o universo de possíveis respostas a fim de retornar somente aquelas mais próximas do contexto da pergunta. Então, no exemplo anterior a pergunta seria expandida para “*Quantos professores docentes trabalhavam lecionavam na UFSC Universidade Federal de Santa Catarina em 2008?*“. A expansão da pergunta também traz vantagens

quanto à desambiguação de termos. Como nesse caso a pergunta reformulada produz um texto retórico, é possível que as ambigüidades sejam resolvidas ou reduzidas com a adição de sinônimos no momento da recuperação das respostas.

Além do uso de sinônimos, outra prática adotada pelos estudos que utilizam ontologias para modelar o domínio é o uso de hierarquias ou herança de classes e relacionamentos. Assim como os sinônimos, a hierarquia de classes pode ser usada tanto para substituir uma classe por outra quanto para incluir a classe pai juntamente com a classe filha na pergunta. Tomando-se como exemplo a pergunta anterior, pode-se substituir o termo “*docente*” por “*pessoa*”, considerando que *docente* é subclasse da classe *Pessoa*. Pode-se substituir uma classe filha por uma classe pai ou vice-versa. Conceitualmente as superclasses abrangem as classes filhas e todas as definições, propriedades ou relacionamentos são herdados às subclasses. Assim, quando ocorre a substituição de uma classe por outra superior na hierarquia, o contexto de aplicação da pergunta é ampliado para um universo maior. Isto é, ao substituir o termo “*docente*” por “*pessoa*”, por exemplo, a pergunta reformulada seria “*Quantas pessoas trabalham na UFSC em 2008?*”. Nesse caso, como a classe *Docente* é subclasse de *Pessoa* e herda todas as suas propriedades e relacionamentos, ao substituí-la o contexto do domínio é ampliado, pois qualquer *Pessoa* (classe pai) não somente *Docente* (classe filha) será considerada.

Do mesmo modo, ao trocar um conceito pai por um conceito filho na hierarquia de classes tende-se a restringir o contexto de aplicação da pergunta. Normalmente, essa substituição acontece quando há relações, propriedades ou termos específicos das classes filhas porém o autor faz menção somente às superclasses na pergunta. Por exemplo, considerando a existência da classe *Discente* como subclasse de *Pessoa*, na pergunta “*Quantas pessoas estudam Filosofia?*” a expressão “*estudam*” estaria mais relacionada à subclasse *Discente* do que a classe *Pessoa* propriamente. Nesse sentido, como somente a classe *Discente* possui um relacionamento do tipo “*estuda*”, a classe *Pessoa* poderia ser trocada por sua subclasse sem grandes perdas. Pode-se também incluir a classe pai juntamente com a classe filha na pergunta. Essa abordagem tem praticamente os mesmos objetivos citados na inclusão de sinônimos. A única diferença é que a inclusão de subclasses ou superclasses geralmente pode ampliar ou restringir o escopo da pergunta como mencionado.

Além da hierarquia de classes, a ontologia de domínio conta também com hierarquias de relacionamentos. Ou seja, pode-se modelar

relacionamentos-pai que generalizam outros relacionamentos-filho em uma hierarquia. Do mesmo modo que as classes, os relacionamentos-filho herdam todas as definições dos relacionamentos-pai podendo ainda definir suas próprias peculiaridades ou restrições. Tomando como exemplo uma relação tripla sujeito-predicato-objeto da ontologia – “*Pessoa temFormacao Organizaçao*”, pode-se criar um relacionamento filho do relacionamento *temFormacao* denominado *temGraduacao*. Esse novo relacionamento filho é caracterizado por ser uma relação mais específica e herda todas as definições do relacionamento *temFormacao*. Portanto, isso significa que *temGraduacao* é também um relacionamento *temFormacao* que do mesmo modo que a hierarquia de classes, pode ser usado na reformulação da pergunta.

Este trabalho emprega a hierarquia de classes e de relacionamentos supracitados bem como o uso de sinônimos para reconhecer o significado da pergunta. No entanto, o uso de hierarquias de conceitos e sinônimos é aplicado em atuação conjunta dos módulos Reformulador e Motor de Busca por Similaridade. O capítulo 4 mostra como a reformulação por hierarquia de classes e relações de sinonímia podem ser realizadas na prática.

Uma vez utilizada a hierarquia de classes e sinônimos, o módulo Reformulador deve transformar a pergunta original levando em conta apenas o modelo da ontologia de domínio. Isto é, somente classes, relacionamentos ou propriedades devem estar contidos na pergunta reformulada. Deste modo, caso a pergunta contenha instâncias de classes ou valores de propriedades, seus termos são substituídos respectivamente por suas classes e propriedades correspondentes. Para esclarecer cita-se o exemplo de pergunta: “*Quantos alunos do Rio de Janeiro estudam Matemática na UFSC?*”. O resultado do processo realizado pelo Reformulador para essa pergunta seria: “*Quantos alunos do estado estudam Disciplina na Instituição?*”. Esse exemplo considera que os termos “Rio de Janeiro”, “*Matemática*” e “*UFSC*” são respectivamente um valor de propriedade de classe (*estado*), uma instância da classe *Disciplina* e uma instância da classe *Instituição*. Apesar de serem substituídos pelo módulo Reformulador, esses termos são utilizados como critérios de filtros posteriormente pelo módulo Tradutor OLAP. Veja a seção 3.6 para compreender como as instâncias de classes e valores de propriedades são utilizados como filtros.

No final do processo de reformulação, a pergunta resultante é comparada diretamente ao modelo da ontologia de domínio pelo Motor de Busca por Similaridade. Na demonstração da aplicabilidade da arquitetura no capítulo 4, os sinônimos e a hierarquia de conceitos são

armazenados e indexados na própria base de conhecimento e, portanto são analisados nesse processo de busca. Assim, a pergunta reformulada seria usada como vetor de busca posteriormente pelo módulo Motor de Busca por Similaridade para encontrar os relacionamentos na ontologia que atendem à pergunta. Logo, o Reformulador e o Motor de Busca por Similaridade são necessários para que a hierarquia de classes e os sinônimos sejam levados em conta na interpretação da pergunta.

Devido às ambigüidades possivelmente existentes e identificadas pelo Analisador Lingüístico, o Reformulador pode produzir mais de uma pergunta resultante a partir da pergunta inicial. Isto é, dado que uma mesma entidade textual pode ter duas ou mais classes reconhecidas pelo Analisador Lingüístico, diferentes reformulações da pergunta podem ser geradas. Cada reformulação contempla um significado possível para a entidade ambígua. Por exemplo, na pergunta “*Quantos pesquisadores trabalham com Stela?*”, considerando que o token *Stela* pode ser uma instância da classe *Pessoa* ou *Instituição*, pode-se ter duas perguntas reformuladas: 1) “*Quantos pesquisadores trabalham com Pessoa?*” e; 2) “*Quantos pesquisadores trabalham com Instituição?*”. Essas perguntas são usadas como entradas para a realização de buscas e comparações com o modelo da ontologia no passo seguinte pelo Motor de Busca por Similaridade. Todavia apenas uma resposta recuperada deve ser utilizada para a construção da consulta final. Veja a seção 3.5 que descreve a tarefa do módulo Motor de Busca por Similaridade com mais detalhes.

3.4.2 Reformulação por regras de inferência

Este trabalho utiliza também as relações definidas em regras de inferência como prática para a reformulação da pergunta. Essa prática é aplicada quando os termos da pergunta são classificados pelo Analisador Lingüístico como sendo *Conclusões de regras de inferência* (veja as classificações na Tabela 2 na seção 3.3). Neste trabalho, a reformulação é realizada da mesma maneira independente da abordagem adotada para a aplicação de regras de inferência: *on-the-fly* ou *in-batch*.

Para compreender como esse segundo tipo de reformulação é realizado, conjectura-se a seguinte pergunta: “*Quantos formandos estudam na UDESC?*”, onde o token “*formandos*” é classificado pelo Analisador Lingüístico como uma conclusão de regra de inferência. A regra de inferência que define a relação *Formando* é exibida a seguir no

Quadro 2. Essa regra é regida pela sintaxe do mecanismo de inferência do framework Jena.

<i>(?pessoa</i>	<i>rdf:type</i>	<i>Pessoa)</i>
<i>(?pessoa</i>	<i>temFormacao</i>	<i>?formacao)</i>
<i>(?formacao</i>	<i>rdf:type</i>	<i>Formacao)</i>
<i>(?formacao</i>	<i>cursadaEm</i>	<i>?instituicao)</i>
<i>(?instituicao</i>	<i>rdf:type</i>	<i>Instituicao)</i>
<i>(?formacao</i>	<i>anoTermino</i>	<i>currentYear(?ano)</i>
→		
<i>(?pessoa</i>	<i>formando</i>	<i>?instituicao)</i>

Quadro 2 - Regra de Inferência que define a relação Formando entre pessoa e instituição.

Obs.: No exemplo acima, *currentYear(?ano)* representa um construtor (*build-in*) criado para que o mecanismo de inferência do framework Jena possa determinar que o valor a ser considerado é o ano atual.

A regra de inferência do Quadro 2 considera que *formando* é toda relação entre pessoa (instância da classe *Pessoa*) e instituição (instância da classe *Instituição*) em que essa pessoa possui uma formação (instância da classe *Formação*) cursada nessa instituição cujo ano de término é o ano atual.

A reformulação por regras de inferência utiliza sempre os fatos conseqüentes contidos nas implicações das regras para reformular a pergunta. Isto é, a tripla (ou conjunto de triplas) que forma a própria conclusão de uma regra é utilizada em substituição ao termo classificado como conclusão de regra de inferência pelo Analisador Lingüístico. Assim, considerando o exemplo da pergunta anterior o token “*formandos*” seria expandido conforme a conclusão da regra denotada apenas pela tripla (*Pessoa formando Instituição*). Portanto, a pergunta reformulada seria: “*Quantos [Pessoa formando Instituição] estudam na Instituição?*”. Nesse exemplo, os sinais [e] são usados apenas para evidenciar os conceitos inseridos na pergunta. Note que além do uso dos conseqüentes da regra, o Reformulador deve do mesmo modo substituir a instância *UDESC* (instância da classe *Instituição*) por sua classe direta.

Cada fato conseqüente da implicação de uma regra possui os relacionamentos entre os conceitos do modelo da ontologia que estão envolvidos na pergunta (no caso *Pessoa formando Instituição*). Embora a reformulação da pergunta seja feita com base nesses fatos conseqüentes das regras de inferência, a aplicação da regra em si só é feita adiante pelo Mecanismo de Inferência. Os conceitos da regra são considerados na pergunta reformulada somente para auxiliar a descoberta do significado da pergunta na etapa posterior, tal como o uso de sinônimos e hierarquia de classes.

A partir das tarefas de classificação e de reformulação efetuadas pelos módulos Analisador Lingüístico e Reformulador respectivamente, pode-se então tentar encontrar qual o caminho ou o conjunto de relacionamentos entre os conceitos que melhor atendem a pergunta. Esse objetivo é cumprido por um módulo proposto denominado Motor de Busca por Similaridade na qual é apresentado na seção a seguir.

3.5 MOTOR DE BUSCA POR SIMILARIDADE

Assim como nos trabalhos de Lopez (et. al., 2007) e Wang (et. al., 2007) o modelo da ontologia de domínio oferece a estrutura entre os conceitos do negócio usada neste trabalho para a interpretação semântica da pergunta. Os elementos textuais usados na pergunta reformulada são confrontados com o modelo da ontologia para saber se estão em conformidade com o contexto do domínio tratado. O intuito dessa comparação é descobrir qual o melhor caminho (ou o conjunto de relações entre os conceitos) que pode resolver a pergunta. Dessa forma, com base na pergunta de saída do Reformulador, o Motor de Busca por Similaridade realiza uma pesquisa sobre o modelo da ontologia de domínio para descobrir qual o caminho que mais se aproxima ao contexto da pergunta. Portanto, a semântica descrita no modelo da ontologia, juntamente com os sinônimos e hierarquia de classes auxiliam na compreensão da pergunta.

Similar ao significado empregado na teoria dos grafos, chama-se de *caminho* o trajeto único formado pela seqüência de conceitos ou classes (vértices) interligados pelos seus relacionamentos (arestas). A fim de esclarecer essa definição, considere a ilustração apresentada na Figura 8.

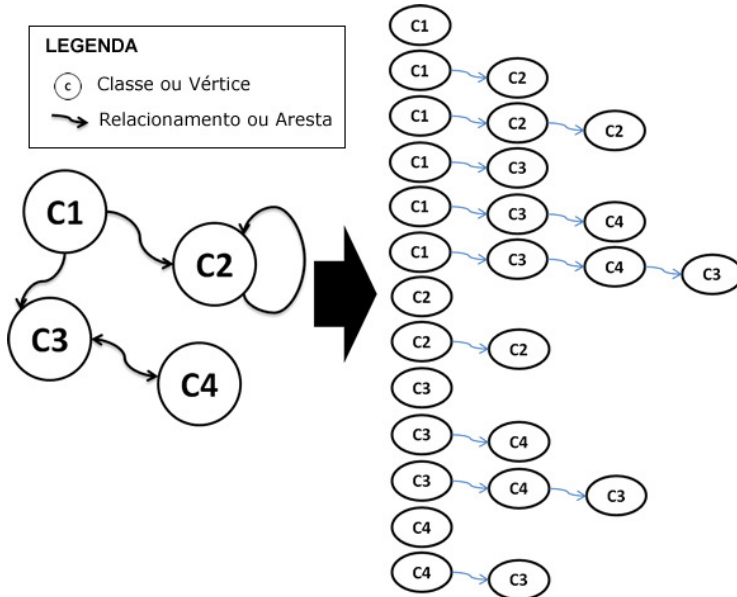


Figura 8 - Ilustração das possibilidades de caminhos para uma ontologia.

A Figura 8 mostra alguns possíveis caminhos para um dado modelo de ontologia. Considere que as classes são as elipses numeradas de C1 a C4 e que as setas indicam a direção de cada relacionamento entre as classes (de quem possui o relacionamento para quem o recebe). Este exemplo apenas preocupa-se em evidenciar as combinações dos caminhos da estrutura da ontologia sem descrever a semântica das relações e conceitos envolvidos. No lado esquerdo da Figura 8, apresenta-se o modelo da ontologia sob a forma de um grafo. Nesse modelo, C1 possui um relacionamento com C2 e C3, C2 possui um auto-relacionamento e C3 relaciona-se com C4 por meio de um relacionamento bidirecional. Já no lado direito têm-se 13 possíveis caminhos que se pode obter a partir desse grafo.

Diferentemente de uma estrutura hierárquica ou em árvore, o modelo da ontologia não possui um conceito ou nodo raiz. Isto é, qualquer classe pode iniciar ou terminar um caminho, respeitando a direção dos relacionamentos do modelo. O menor caminho possível sempre é aquele formado por apenas uma das classes da ontologia isoladamente, que no exemplo é visualizado à direita na Figura 8 pelas elipses C1, C2, C3 ou C4 sem a presença de ligações. Nesse caso, o próprio conceito ou classe forma um caminho em que o vértice inicial é

também o vértice final. Além disso, observe à direita da Figura 8 que os auto-relacionamentos e os relacionamentos bidirecionais foram desmembrados para perceber melhor os novos caminhos formados por esses tipos de relações. Devido ao auto-relacionamento presente na classe C2, há uma recursividade que deve ser tratada como um novo caminho. O mesmo ocorre com C3 e C4, que possuem um relacionamento bidirecional. Na prática, esses tipos de relacionamentos devem ser considerados como se houvesse uma nova ligação com um novo conceito. Ou seja, pode existir um caminho formado apenas por dois vértices com a presença de dois conceitos $C1 \rightarrow C2$, ou ainda um diferente caminho formado por três vértices e por dois conceitos $C1 \rightarrow C2 \rightarrow C2'$. Esse tipo de estrutura assemelha-se ao de um autômato finito e é bastante encontrado nos modelos OWL.

Os diferentes caminhos possíveis da ontologia de domínio representam todas as possibilidades de interpretações e consultas para um determinado contexto de pergunta. Cada caminho estabelece toda a semântica de relacionamentos entre um ou mais conceitos que deve ser usado como guia para a formalização das consultas. A semântica ou o que o tomador de decisão quer dizer com uma pergunta é na prática associado a um único caminho. Cabe ao Motor de Busca por Similaridade definir o melhor caminho que representa o significado da pergunta considerando toda a terminologia empregada na pergunta.

Nesta pesquisa, semelhante como é adotado por Lopez (et. al., 2007), o melhor caminho é caracterizado por ser aquele que apresenta a maior quantidade de conceitos e relacionamentos relevantes identificados a partir da pergunta. Ou seja, todos os conceitos da pergunta, após serem identificados, devem ser considerados para determinação do caminho, na qual o mais completo em comparação ao modelo da ontologia deve ser o utilizado. Logo, aqueles que, quando comparados, possuem um maior número de conceitos e relacionamentos afins ou semelhantes são avaliados como candidatos mais favoráveis a atender a pergunta do que aqueles com menor quantidade de elementos relacionados. Visto que o resultado da reformulação desconsidera as instâncias de classes em substituição pelas próprias classes diretas, apenas o modelo de classes e relacionamentos é usado pelo Motor de Busca por Similaridade.

Na descoberta do melhor caminho é provável que o Motor de Busca por Similaridade encontre mais de um caminho possível relacionado à pergunta. Quando isto ocorre, tem-se uma ambigüidade entre os caminhos que podem atender a uma dada pergunta. Assim, além das ambigüidades anteriormente identificadas pelo Analisador

Lingüístico, o Motor de Busca por Similaridade é responsável também pela resolução das ambigüidades entre caminhos candidatos. Por isso, dois tipos de desambiguação são possíveis de serem efetuados pelo Motor de Busca de Similaridade: desambiguação de conceitos (classes, propriedades) e desambiguação de caminhos (ou relacionamentos). Esses dois tipos são detalhados a seguir:

- 1) A primeira etapa da desambiguação ocorre quando o Analisador Lingüístico reconhece mais de uma classificação para as entidades textuais da pergunta. Mesmo com a presença de ambigüidades já identificadas pelo Analisador Lingüístico, o Motor de Busca por Similaridade executa uma busca para localizar o melhor caminho no modelo da ontologia de domínio. Isto porque o contexto da ontologia pode auxiliar a resolver a ambigüidade caso o resultado da busca retorne um único caminho, ou ainda, auxiliar a diminuir a ambigüidade considerando que o resultado da busca retorne poucos caminhos candidatos. Como mencionado antes, o Reformulador pode produzir uma ou mais perguntas como saída de seu processo. Para cada pergunta gerada na reformulação deve ser feita uma busca para localizar a existência do caminho na ontologia. Caso o somatório das buscas retorne mais de um caminho candidato, o tomador de decisão deve interagir iterativamente com o Motor de Busca por Similaridade até identificar um único caminho que atenda à pergunta.
- 2) Mesmo que não exista ambigüidade nas entidades classificadas pelo Analisador Lingüístico ou ainda que apenas uma pergunta resultante da tarefa de reformulação seja gerada, o Motor de Busca por Similaridade pode retornar mais de um caminho. A ambigüidade entre caminhos sempre ocorre quando nenhum termo informado na pergunta contribui para a descoberta de um único relacionamento entre os conceitos. Dessa forma, caso haja dois ou mais caminhos retornados na comparação do modelo da ontologia, deve ocorrer o processo de desambiguação desses caminhos candidatos. Novamente o tomador de decisão participa diretamente da escolha do melhor caminho para resolver a ambigüidade e atender à pergunta.

Portanto, ambos os processos de desambiguação podem necessitar da presença do usuário para serem concluídos. Conforme a escolha do usuário nesse processo, os elementos textuais da pergunta são identificados e a pergunta é iterativamente refinada até que não haja dúvidas sobre qual é o melhor caminho e o significado dos elementos.

O Motor de Busca por Similaridade, como o próprio nome indica, efetua buscas sobre o modelo da ontologia. O termo *Similaridade* desse módulo refere-se ao fato que sinônimos, hierarquia de classes, dentre outros tipos de relações são usadas para a localização das terminologias no modelo da ontologia. Como fonte de busca, pode-se estruturar a ontologia para que os sinônimos e hierarquias de classes sejam armazenados, por exemplo, em índices textuais para facilitar a recuperação do caminho, tal como é construído no protótipo da arquitetura no capítulo 4.

Após o melhor caminho ser identificado bem como a semântica dos termos da pergunta, a consulta sobre as fontes de dados pode ser construída. O trabalho de traduzir o melhor caminho em uma requisição para explorar o DW é executado pelo Tradutor OLAP descrito na próxima seção.

3.6 TRADUTOR OLAP

A partir do caminho mais aderente ao modelo da ontologia de domínio, o Tradutor OLAP determina os elementos construtores das operações OLAP que são: medidas, agrupamentos, filtros e ligações entre conceitos. Como o próprio nome indica, ele realiza uma tradução do melhor caminho encontrado pelo Motor de Busca por Similaridade em uma requisição OLAP, que será posteriormente executada sobre o DW pelo módulo Gerenciador de Consultas. O objetivo de determinar esses elementos da consulta OLAP (medidas, agrupamentos, filtros e ligações) é motivado no Gerenciador de Consultas oriundo do trabalho de Sell (et. al., 2008).

Com base no melhor caminho, o Tradutor OLAP especifica uma expressão (ou requisição) que é formada somente pelos conceitos da ontologia de domínio. Nessa expressão, os conceitos do domínio (classes, propriedades, relacionamentos e instâncias de classes) são organizados em medidas, agrupamentos ou projeções, filtros e suas ligações. Logo, a requisição remete diretamente às classes, propriedades e aos relacionamentos da ontologia e não aos objetos (tabelas de fato, dimensões e atributo de dimensões) do esquema estrela do DW. A associação dos conceitos da ontologia com as tabelas e dimensões do DW está definida na Ontologia BI e é aplicada pelo Gerenciador de Consultas em seguida. Note que a consulta para obtenção da resposta final não é executada sobre a base de conhecimento, tal como alguns

trabalhos relacionados fazem por meio de SPARQL. As consultas OLAP são propriamente realizadas sobre o DW ou Data Marts da organização, geralmente por meio de *SQL* (Structured Query Language).

Dentre os construtores das operações OLAP, as medidas representam quantificações numéricas (somatório, média, mínimo, máximo, etc.) sobre um determinado conceito do domínio. Na prática, os conceitos classificados como medidas pelo Tradutor OLAP são associados aos fatos aditivos, indicadores ou medidas propriamente ditas do modelo dimensional.

Já os conceitos traduzidos como agrupamentos provêm as informações descritivas utilizadas para agrupar ou categorizar as medidas nas consultas. Esses conceitos geralmente devem ser associados aos atributos das dimensões do DW e são propriedades de classes.

Quando os conceitos são traduzidos como filtros, os valores relacionados a esses conceitos são usados como critérios de seleção na requisição OLAP. Normalmente, o elemento da pergunta é traduzido como filtro quando se refere a uma instância de classe ou quando um valor identificador de uma propriedade (*nome, sigla, data, etc.*) é informado. Além disso, o Tradutor OLAP encarrega-se de traduzir aqueles termos da pergunta classificados pelo Analisador Lingüístico como *Função*. Nesses casos, o resultado da função é usado também como valor para a criação dos filtros na requisição OLAP. A função relaciona um ou mais termos a um conceito do domínio e ao seu respectivo cálculo ou valor. Quando um termo é substituído pelo resultado da função é como se o valor resultante estivesse sido informado na pergunta. Então, o Tradutor OLAP pode tanto identificar valores como filtros diretamente na pergunta quanto aplicar funções para substituir os termos por seus valores para a criação da requisição OLAP.

Além das medidas, agrupamentos e filtros, o Tradutor OLAP deve também definir na requisição de consulta como esses itens devem ser relacionados. Cada relacionamento entre as classes denota qual a junção ou ligação que deve ser utilizada entre os objetos das fontes de dados. Todos os relacionamentos entre conceitos, inclusive aqueles provenientes dos conseqüentes de regras de inferência na fase de reformulação, devem estar presentes no caminho oriundo do Motor de Busca por Similaridade. Isso serve para que o Tradutor OLAP possa considerar por completo as entidades do modelo da ontologia de domínio que devem ser relacionadas para obtenção do cubo OLAP adiante.

A fim de realizar a tradução do caminho em uma consulta, o Tradutor OLAP utiliza um conjunto de padrões e heurísticas baseadas na distância ou posição entre os conceitos da pergunta e stop-words. Embora as palavras classificadas como stop-words sejam desprezadas pela maioria dos sistemas de RI, elas são essenciais nessa fase de tradução. Geralmente, as pesquisas de QA baseadas em respostas textuais utilizam stop-words para classificar o tipo de pergunta e também para identificar o padrão sintático para respondê-la corretamente. Neste trabalho, as stop-words além de serem úteis para determinar o tipo de pergunta, elas auxiliam também a descoberta dos elementos da consulta OLAP, como medidas, agrupamentos, filtros e junções.

Todos os padrões sintáticos e as heurísticas juntamente com a listagem de stop-words usados pelo Tradutor OLAP devem ser configurados de acordo com o idioma na base de conhecimento. Tal como nos frameworks propostos por Lopez (et. al.; 2007) e Meng e Chu (1999), essa configuração permite que expressões regulares e critérios baseados na posição ou distância entre tokens e stop-words sejam aplicados pelo Tradutor OLAP. Com isso, há uma maior flexibilidade no reconhecimento dos elementos da consulta de acordo com os padrões idiomáticos ou ainda o modo de escrita dos tomadores de decisão. Essa proposta não determina um conjunto fixo de padrões ou heurísticas para a obtenção da consulta OLAP, uma vez que há uma variedade de formas idiomáticas possíveis que podem ser utilizadas nessa tarefa. Entretanto, o capítulo 4 detalha algumas adaptações do trabalho de Meng e Chu (1999) e alguns padrões e heurísticas baseadas no estudo de Lopez (et. al., 2007) que podem ser aprimorados ou modificados conforme a necessidade.

Para realizar a tradução e configurar os padrões no Modelo e Base de Conhecimento, este trabalho adota três tipos de *stop-words* organizadas conforme a Tabela 3 a seguir.

Tabela 3 - Tipos de stop-words para a tradução OLAP

Tipo de Stop-word	Elemento da consulta	Descrição
Quantificação	Medida	Stop-words que tratam da sumarização ou cálculos de valores numéricos e quantificação de

		dados. Cita-se como exemplo as expressões <i>quanta(s)</i> , <i>quanto(s)</i> , e suas variantes como <i>qual a quantidade de</i> , <i>qual o total de</i> , etc.
Projeção	Agrupamento	São aquelas utilizadas para categorizar ou agrupar conteúdos normalmente descritivos sem necessidade de quantificá-los, tais como os atributos de dimensão. São exemplos as stop-words: <i>por</i> , <i>conforme</i> , <i>agrupado por</i> , <i>segundo</i> , etc. em perguntas como “Quantos alunos <u>agrupados por</u> idade e <u>por</u> cidade estudam no Sul?”.
Seleção	Filtro	São as stop-words que participam de critérios de seleção para filtrar um conjunto de dados. Distinguem-se aqui as stop-words relacionais (operadores relacionais), como <i>acima de</i> , <i>igual a</i> , <i>maior que</i> , <i>abaixo de</i> , etc. e ainda lógicas (operadores lógicos) como, <i>E</i> e <i>OU</i> . Exemplo de pergunta com stop-words relacionais: “Quantos especialistas <u>acima de</u> 40 anos e <u>abaixo de</u> 50 anos publicaram artigos em 2010?”. Exemplo de pergunta com stop-words lógicas: “Quantos professores <u>e</u> discentes estudam Fisiologia Humana <u>ou</u> Saúde Ocupacional?”.

Conforme assinalado na delimitação de escopo deste trabalho, as perguntas, e conseqüentemente as respostas, são interpretadas para atender ao contexto de BI, onde apenas sumarizações e quantificações são possíveis. Por isso, os tipos de *stop-words* apontados na Tabela 3 apresentam os elementos visando à construção de consultas multidimensionais.

A requisição de consulta gerada na tradução não exige a presença de filtros, porém espera-se que ela possua pelo menos uma medida ou

um agrupamento que determine qual informação deverá ser extraída das fontes de dados. Por exemplo, na pergunta “Quantos alunos?” embora seja simples, a pergunta é considerada válida e, por conseguinte, a consulta resultante apresentaria apenas uma única medida (e.g. *total de alunos*), sem a presença de agrupamentos ou filtros.

No exemplo citado na Tabela 3 para as stop-words lógicas, note que os *tokens E* e *OU* podem ser usados não somente como critérios de filtros mas também para listar mais de uma projeção (no caso, *Quantos professores e discentes*). Isto é, ambos tokens podem ser utilizados na combinação dos elementos que devem ser retornados (medidas e agrupamentos), como podem ser também utilizados na criação de filtros na consulta. Dessa forma, as configurações de padrões e heurísticas devem ser diferenciadas no Modelo e Base de Conhecimento para esses tipos de *stop-words*.

Outro fato observado sobre as stop-words lógicas *E* e *OU* é que quando são utilizadas para especificar os elementos de retorno da consulta (como projeções), elas podem ser usadas indistintamente uma vez que produzem o mesmo resultado. No exemplo a pergunta “Quantos professores e discentes...” tem o mesmo efeito de “Quantos professores ou discentes...”, pois a semântica dos *tokens E* e *OU* é a mesma e por isso, ambas as informações dos conceitos (professores e discentes) devem ser projetados. O problema ocorre quando essas stop-words participam de filtros. Isto porque o uso na linguagem natural dos tokens *E* e *OU* confunde-se com os respectivos operadores lógicos. Por exemplo, na pergunta “*Quantos professores estudam Fisiologia Humana e Saúde Ocupacional?*” o token *E* pode ter sua semântica interpretada como: a) intersecção – somente os professores que estudam ambas as áreas simultaneamente ou; b) união – todos os professores que estudam tanto as duas áreas quanto apenas uma delas. Essa variação de semântica é devida ao *E* lingüístico ser muitas vezes diferente do operador lógico *E*, sendo que o seu sentido na pergunta pode estar associado na verdade ao operador lógico *OU*. Por motivos de simplificação, este trabalho considera a semântica dos operadores lógicos para a criação de filtros. Assim, o tomador de decisão tem as opções de intersecção ou união na seleção de conteúdo.

Assim como as funções e conclusões de regras de inferência, novas stop-words podem ser especificadas no Modelo e Base de conhecimento para que sejam utilizadas na determinação dos elementos da consulta. Logo, novas stop-words ainda não mapeadas podem ser incorporadas à arquitetura para que os padrões e heurísticas sejam configurados e tratados pelo Tradutor OLAP. A seção a seguir

apresenta o módulo Gerenciador de Consultas que faz interface com o Tradutor OLAP.

3.7 GERENCIADOR DE CONSULTAS

O Gerenciador de Consultas é um módulo inspirado nas pesquisas de BI propostas por Sell (2006; et. al., 2008) e por Beneventano (et. al., 2007) e por isso herda as funcionalidades desses estudos. Ele é responsável por criar a consulta OLAP e por retornar as informações estratégicas ao tomador de decisão. A partir da especificação da consulta OLAP formalizada pelo Tradutor OLAP, o Gerenciador de Consulta constrói uma consulta conforme a linguagem da fonte de dados para obter o cubo OLAP. Na grande maioria dos casos, o DW é construído sobre banco de dados relacionais (*ROLAP*) e, conseqüentemente a linguagem de consulta gerada é SQL.

Para aplicar a requisição gerada pelo Tradutor OLAP, é necessário que os conceitos e os relacionamentos da ontologia de domínio sejam mapeados aos objetos ou esquemas das fontes de dados anteriormente. Isto é, deve existir uma correspondência entre cada classe, propriedade e relacionamento da ontologia de domínio com as tabelas de fato, dimensões e atributos do Data Warehouse. Todo esse mapeamento deve ser feito previamente na criação das instâncias da Ontologia BI, conforme comentado na seção 3.2. Para obter esse mapeamento a partir da Ontologia BI, o Gerenciador de Consultas conta com o auxílio do módulo Gerenciador de Ontologias. Assim, todas as estruturas e objetos das fontes de dados são obtidos a partir dos conceitos de domínio especificados na requisição gerada pelo Tradutor OLAP. A seção 4.3.3 apresenta como a Ontologia BI faz a associação da ontologia de domínio ao modelo dimensional do estudo de caso deste trabalho.

Dependendo da abordagem de inferência utilizada, *in-batch* ou *on-the-fly*, o Gerenciador de Consultas deve agir de modo diferente para a criação da consulta. As subseções a seguir explicam em detalhes as diferenças para as duas abordagens.

3.7.1 Consultas na abordagem *on-the-fly*

Em comparação ao processo *in-batch*, a abordagem *on-the-fly* é o método mais dispendioso com relação ao desempenho para a obtenção do cubo OLAP. Isto porque ela sempre depende da aplicação de regras de inferência realizadas em tempo de execução e a criação da consulta com base no resultado dessas inferências. Essa técnica é a mesma utilizada por Sell (2006, et. al., 2008) para a elaboração das operações que envolvem filtros semânticos (*slices* e *dices*) sobre o conteúdo do DW.

Como exposto no exemplo da seção 3.4.2, quando um termo da pergunta é classificado pelo Analisador Lingüístico como *conclusão de regra de inferência*, a pergunta reformulada possui os fatos (triplas) da implicação da regra. Para que a inferência seja considerada na análise, a regra de inferência deve ser processada pelo módulo Mecanismo de Inferência e o conjunto de dados resultante deve ser usado pelo Gerenciador de Consultas na criação da consulta OLAP.

Dado a mesma pergunta da seção 3.4.2, que possui o conceito *formando*, a regra de inferência que deriva esse conceito deve ser aplicada antes da consulta ser escrita pelo Gerenciador de Consultas. Então na abordagem *on-the-fly*, ao identificar que o termo *formando* é uma conclusão de regra de inferência, o Gerenciador de Consultas solicita a aplicação da referida regra. Essa solicitação se dá por meio do módulo Gerenciador de Ontologias, que interage diretamente com o Mecanismo de Inferência da arquitetura. O Mecanismo de Inferência executa a regra de inferência sobre o conjunto de instâncias contidas na Base de Conhecimento. Todas as instâncias devem ter sido criadas anteriormente na base de conhecimento pelo Gerenciador de Ontologia (veja a seção 3.8 para mais informações). Durante esse processo de inferência, caso as premissas da regra sejam satisfeitas, as conclusões do conceito *formando* são retornadas pelo Mecanismo de Inferência. Como o resultado dessa conclusão associa as instâncias da classe *Pessoa* às instâncias da classe *Instituicao*, os valores retornados para esses conceitos são usados como filtros na criação da consulta. Com isso, o Gerenciador de Consultas reconhece quais os valores de filtros (no caso, as informações de pessoas e instituições) que satisfazem a regra e que devem ser usados na escrita da consulta. No caso, as respectivas dimensões mapeadas na Ontologia BI desses conceitos seriam usadas pelo Gerenciador de Consultas.

A consulta gerada pelo Gerenciador de Ontologias na abordagem *on-the-fly* possui sempre o resultado das inferências como critérios de filtros. Um exemplo mais detalhado de consulta na abordagem *on-the-fly* pode ser vista em Sell (2006; et. al., 2008) e também é exemplificada na seção 4.5.3 deste trabalho.

3.7.2 Consultas na abordagem in-batch

Caso a abordagem *in-batch* seja a adotada, todas as derivações semânticas ou conclusões das regras devem ser armazenadas no modelo tripla projetado no DW. Neste caso, os resultados das inferências armazenados no modelo tripla são combinados com as informações das dimensões e tabelas de fato diretamente na consulta pelo Gerenciador de Consultas.

O modelo tripla sempre interliga as dimensões do DW com as derivações semânticas das regras de inferência feitas a priori. Por isso, o cubo OLAP pode ser retornado pelo Gerenciador de Consultas sem a necessidade de reescrever a consulta com os filtros oriundos do processo de inferência, diferentemente da abordagem *on-the-fly*. O modelo tripla é detalhado na seção 3.10 adiante.

Para combinar o modelo tripla e o modelo dimensional em uma única consulta, a pergunta reformulada e posteriormente traduzida pelo Tradutor OLAP deve contemplar a relação tripla presente no fato conseqüente (conclusão) da regra. Neste caso, as relações dos conceitos presentes nesses fatos conseqüentes ou conclusões das regras devem ser mapeadas também na Ontologia BI. Assim, o Gerenciador de Consultas realiza o mesmo procedimento em conjunto com o Gerenciador de Ontologias para obter a estrutura do DW e construir a consulta usando o modelo tripla. Na seção 4.5.3 apresenta-se na prática como essa consulta pode ser criada pelo Gerenciador de Consultas.

Dada a complexidade das tarefas, os sistemas de QA tradicionais podem apresentar erros durante as etapas de interpretação e retorno da resposta. No entanto, considera-se que a etapa desempenhada exclusivamente pelo Gerenciador de Consultas neste trabalho não produz resultados incorretos. Isto porque a obtenção da resposta constitui-se na exploração de repositórios de dados estruturados com base em uma pergunta formal e, no geral, essa exploração é uma atividade comum, bastante disseminada entre especialistas e possui diversos mecanismos e ferramentas já consolidados para sua execução.

Assim, a resposta só produzirá um resultado indesejado ou inválido caso a requisição OLAP, interpretada até a etapa de tradução anterior, estiver incorreta. A seção a seguir detalha o módulo Gerenciador de Ontologias que auxilia o Gerenciador de Consultas na aplicação de consultas.

3.8 GERENCIADOR DE ONTOLOGIAS

Também motivado na iniciativa de Sell (et. al., 2008), o Gerenciador de Ontologias é designado à manipulação das ontologias utilizadas neste trabalho que fazem parte do Modelo e Base de conhecimento. Nesta pesquisa, esse módulo é responsável por: a) criação e recuperação das instâncias da ontologia de domínio para aplicação de regras de inferência na base de conhecimento; b) suporte ao Gerenciador de Consultas na localização das instâncias da Ontologia BI que contém a associação entre os elementos das fontes de dados e os conceitos da ontologia de domínio; c) comunicação com o Mecanismo de Inferência para obtenção das conclusões semânticas das regras de inferência no processo analítico e; d) armazenamento das conclusões das regras de inferência em uma estrutura própria no DW (modelo tripla), quando adotada a abordagem de raciocínio *in-batch*.

Dado que as instâncias já estejam construídas na Ontologia BI, a criação das instâncias da ontologia de domínio pode ser feita automaticamente. Para tal, pressupõe que todas as informações necessárias para popular a ontologia de domínio estejam presentes no DW. Na prática, como mencionado por Sell (et. al., 2008), o conteúdo para a criação de instâncias do domínio se encontra principalmente nos atributos de dimensão, já que estas possuem as informações textuais sobre o negócio. Nessa tarefa, o mapeamento previsto na Ontologia BI, obtido pelo Gerenciador de Ontologias, fornece as informações necessárias para a formulação das consultas sobre o DW. Assim, o Gerenciador de Ontologias pode atuar com o Gerenciador de Consultas para recuperar os dados do DW necessários para a criação das instâncias da ontologia de domínio e formar a base de conhecimento.

Uma vez que a base de conhecimento contém as instâncias da ontologia domínio, o Mecanismo de Inferência pode aplicar as regras de inferência definidas e obter as deduções semânticas (veja a seção 3.9 a seguir). O Gerenciador de Ontologias interage com o Mecanismo de Inferência para que o resultado do processo de inferência seja utilizado pelo Gerenciador de Consultas na criação das análises na abordagem de

raciocínio *on-the-fly*. Para que ocorram as derivações semânticas *na abordagem on-the-fly*, todas as instâncias devem estar presentes na base de conhecimento previamente a entrada da pergunta.

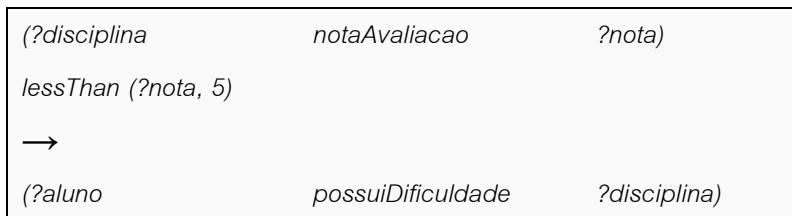
Em especial, o Gerenciador de Ontologias desempenha uma função específica para a abordagem *in-batch*. Nesse caso, o resultado das inferências é armazenado no modelo tripla, que complementa o esquema estrela. Os resultados obtidos pela aplicação das regras de inferência encontram-se sob o formato de triplas *sujeito-predicado-objeto*. O Gerenciador de Ontologias armazena cada resultado (triplas) no modelo tripla conforme é apresentado na seção 3.10.

3.9 MECANISMO DE INFERÊNCIA

O módulo Mecanismo de Inferência tem o papel de explicitar as informações para apoio ao processo decisório e descoberta de conhecimento. Essa tarefa é executada por meio de aplicação das regras de inferência definidas conforme o negócio da organização. Tal como nos formalismos previstos na Web Semântica, essas regras são definidas tendo como base o modelo conceitual da ontologia de domínio. Com as instâncias criadas nesse modelo, as conclusões ou derivações semânticas podem ser obtidas por descoberta de relacionamentos implícitos entre essas instâncias. Assim, as análises sobre o conteúdo do DW podem ser combinadas com essas derivações semânticas e conseqüentemente visões inéditas podem ser produzidas ao tomador de decisão.

Para ilustrar como o processo de inferência pode auxiliar as análises e gestão do conhecimento, cita-se um exemplo hipotético dentro do cenário acadêmico: um coordenador de curso precisa identificar os alunos com potenciais dificuldades e com baixo desempenho em disciplinas cursadas a fim de propor políticas para melhorar o ensino e evitar a evasão desses alunos. Considerando esse universo de discurso, a análise para a identificação desses alunos poderia ser auxiliada pela seguinte regra de inferência do Quadro 3.

<i>(?aluno</i>	<i>rdf:type</i>	<i>Discente)</i>
<i>(?aluno</i>	<i>estuda</i>	<i>?disciplina)</i>
<i>(?disciplina</i>	<i>rdf:type</i>	<i>Disciplina)</i>



Quadro 3 - Regra de inferência para identificar os alunos com baixo desempenho.

O Quadro 3 traz uma regra de inferência que já considera a sintaxe do framework Jena usado no protótipo da arquitetura. Essa regra declara em termos formais que se um aluno ao cursar uma disciplina tiver pelo menos uma nota de avaliação abaixo de 5, ele possui potenciais dificuldades na referida disciplina. Assim, uma vez inferidas, as relações triplas (no exemplo, *aluno possuiDificuldade disciplina*) podem ser integradas ao retorno das consultas OLAP. Neste exemplo, consultas como “*Quantos estudantes por curso e por gênero possuem dificuldades em Física?*” requerem informações de alunos, curso e disciplina. A relação tripla explicitada determina quais alunos possuem as características de baixo desempenho definidas pela regra de inferência acima. Deste modo, com a aplicação dessa regra, somente as informações desses alunos devem ser retornadas na consulta ao coordenador do curso. Assim, o Gerenciador de Ontologias obtém do Mecanismo de Inferência as informações que devem ser consideradas nas consultas pelo Gerenciador de Consultas conforme a abordagem de raciocínio *on-the-fly* ou *in-batch*.

É evidente que consultas e relatórios podem ser desenvolvidos exclusivamente para atender à pergunta do exemplo acima e que as ferramentas OLAP dão suporte a localização dessas informações. No entanto, dada uma mudança de requisitos de análise ou necessidade de novas variáveis de interesse, novamente tais relatórios devem ser desenvolvidos e muitas vezes necessitam de suporte de especialistas técnicos (SELL, 2006). Nesse ponto, as regras de inferência, como forma de representação de conhecimento, possuem a vantagem de dar flexibilidade tanto às alterações quanto inclusive às adições de novos requisitos de informação. O conhecimento de como obter os alunos com dificuldades em uma disciplina pode então ser alterado conforme a regra da organização. Toda a semântica estabelecida na regra fornece a flexibilidade às soluções de BI para que os sistemas ETL, o modelo

dimensional e cubos OLAP não necessitem ser modificados (SELL, 2006; SILVA, 2006).

3.10 DATA WAREHOUSE

Nessa proposta, o Data Warehouse é considerado um elemento externo à arquitetura e não necessita sofrer alterações no modelo dimensional. No entanto, este trabalho adota o modelo tripla abordado por Silva (2006) e Sell (et. al., 2008) usado como extensão ao modelo dimensional para a persistência de conclusões de regras de inferência, conforme comentado na seção 0. Deste modo, o DW além de integrar as informações das fontes operacionais da empresa, também pode ser utilizado para armazenamento do resultado do processo de inferência feito pelo Mecanismo de Inferência. A Figura 9 a seguir ilustra como o modelo tripla é organizado com o modelo dimensional.

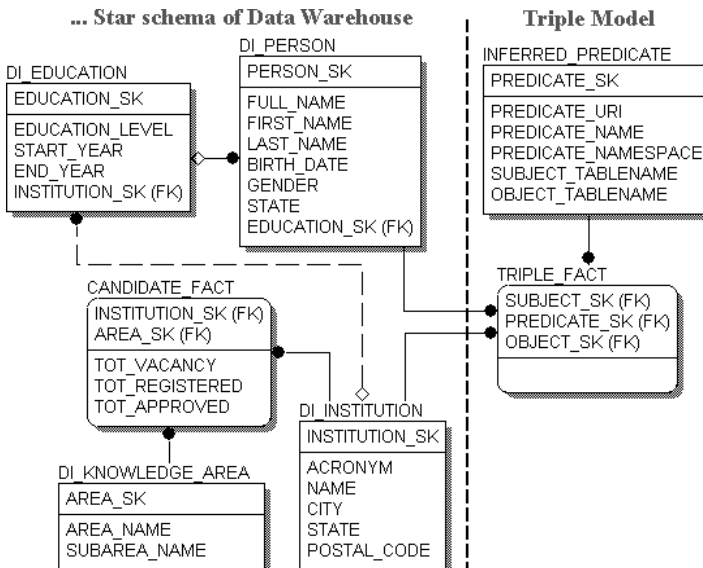


Figura 9 - Modelo tripla para persistência de conclusões de regras de inferência. Fonte: Sell (et. al., 2008).

O modelo tripla, representado na Figura 9, estende o esquema estrela por interligar duas de suas dimensões a uma dimensão chamada

INFERRED_PREDICATE. Essa interligação se dá através de uma tabela associativa denominada *TRIPLE_FACT*. Assim como no framework RDF que organiza os recursos em sujeito-predicado-objeto, o modelo tripla exemplificado na Figura 9 possibilita que as dimensões *DI_PERSON* (*sujeito da relação* - com dados de *pessoa*) e *DI_INSTITUTION* (*objeto da relação* - com dados de instituição) possam se relacionar por um predicado definido na dimensão *INFERRED_PREDICATE*.

O modelo tripla é usado na abordagem de raciocínio *in-batch* para que o Gerenciador de Consultas possa combinar as informações das dimensões com as informações derivadas e armazenadas no modelo tripla. Dado que mais de um conceito (predicado) pode ser armazenado nesse modelo, o Gerenciador de Consultas deve filtrar o conceito desejado por meio da dimensão *INFERRED_PREDICATE*. O atributo *PREDICATE_NAME* dessa dimensão é usado no critério de seleção dos predicados conforme o termo da pergunta classificado como conclusão de regra de inferência. A seção 4.5.3 traz um exemplo de consulta envolvendo esse modelo.

4 DEMONSTRAÇÃO DA VIABILIDADE DA ARQUITETURA

A arquitetura apresentada no capítulo anterior define os objetivos e responsabilidades de cada componente, suas interações e as tarefas que cada um desempenha de acordo com os conjuntos de entradas e saídas esperados. Ela não detalha o modo como cada componente deve ser desenvolvido em termos de algoritmos ou de tecnologias e nem especifica frameworks de terceiros que devem ser utilizados na realização das tarefas. Isto porque não se deseja limitar o uso de novos métodos ou tecnologias para conceber uma instância da arquitetura. Deste modo, cabe ao engenheiro do conhecimento adotar os recursos e frameworks que melhor atendem aos requisitos e ao contexto da organização com base na arquitetura proposta.

Este capítulo mostra uma aplicação da arquitetura para o domínio relacionado às produções intelectuais e atividades acadêmicas e profissionais dos pesquisadores, docentes e discentes da UFSC. Toda a amostra de dados utilizada como fonte de dados é oriunda da Plataforma Lattes Institucional²¹ (PLI) e foi concedida pela Universidade Federal de Santa Catarina (UFSC). A seção a seguir introduz o contexto de Ciência e Tecnologia (C&T) utilizado como estudo de caso.

4.1 INTRODUÇÃO AO DOMÍNIO DE APLICAÇÃO DO PROTÓTIPO

O cenário de aplicabilidade da arquitetura concentra-se no âmbito de C&T da Plataforma Lattes do CNPq. A Plataforma Lattes é uma plataforma de governo eletrônico destinada à gestão curricular que contempla informações de pesquisadores, de grupos de pesquisa de C&T e instituições de ensino de todo o Brasil (CNPq, 2010). Por meio dos currículos presentes na base de dados dessa plataforma, é possível realizar análises sobre o cenário histórico e a evolução das produções intelectuais do país, avaliar tendências e indicadores de C&T conforme as áreas de conhecimento e também, estabelecer redes sociais de atuação profissional e acadêmica entre especialistas.

²¹ Plataforma Lattes Institucional - <http://lattes.ufsc.br>

Em particular, este trabalho utiliza uma amostra de dados da PLI da UFSC relacionada a dois assuntos:

- Atividades acadêmicas e profissionais: relacionado às atividades profissionais e acadêmicas dos pesquisadores, docentes e discentes da UFSC exercidas nos laboratórios, grupos de pesquisa, departamentos e demais órgãos da universidade e também em outras instituições de ensino.
- Produção Intelectual: trata das produções bibliográficas e técnicas (artigos publicados, participações em eventos, dentre outros) da comunidade acadêmica da UFSC.

Ambos os assuntos estão organizados em dois respectivos Data Marts que compõem o DW da PLI adotado neste trabalho. Esse DW é descrito na próxima seção com as descrições das tabelas de fato e dimensões projetadas para o estudo de caso.

4.2 DATA WAREHOUSE DA PLATAFORMA LATTES

A partir dos dados provenientes do DW da Plataforma Lattes, um modelo dimensional foi construído para abranger um cenário mais completo de utilização dos módulos da arquitetura. Esse modelo foi criado para que o conteúdo do esquema estrela seja combinado às conclusões de regras de inferência do modelo tripla. Logo adiante, esse modelo dimensional já integrado ao modelo tripla é apresentado na Figura 10.

Tabela 4 - Descrição do modelo dimensional do DW.

Dimensão ou Tabela de Fato	Nº de Registros	Descrição
di_pessoa	2.945	Dimensão que possui os dados de identificação das pessoas, como nome, data e país de nascimento, gênero, nacionalidade, etc.
di_formacao	6.983	Dimensão que contempla as titulações, formações ou escolaridades das pessoas.
di_instituicao	16.413	Dimensão que contém as informações dos grupos de pesquisa, departamento, centros e demais órgãos da UFSC e outras instituições de ensino.
di_natureza_atividade	18	Dimensão que possui as descrições dos tipos de atividades acadêmicas ou profissionais
di_area_conhecimento	55	Dimensão com os dados que identificam as áreas de conhecimento.
ft_atividade	7249	Tabela de fato que determina a quantidade de atividades acadêmicas ou profissionais das pessoas nos órgãos e instituições.
ft_producao	7995	Tabela de fato com a quantidade de produções bibliográficas e técnicas anuais dos especialistas.

O banco de dados utilizado para a criação do DW é o *PostgreSQL*²² versão 9.0. Como visto na Tabela 4 e na Figura 10, esse

²² PostgreSQL - <http://www.postgresql.org>

banco possui uma quantidade pequena de registros e junções entre tabelas. Isso facilita a conferência de dados e validação de consultas conforme as perguntas informadas.

4.3 MODELO E BASE DE CONHECIMENTO UTILIZADO

Dado que o DW da organização já está construído, a primeira etapa do desenvolvimento da arquitetura é a preparação do módulo Modelo e Base de Conhecimento. Conforme visto na seção 3.2, essa etapa envolve: a modelagem da ontologia de domínio; a definição das funções e regras de inferência específicas para o domínio; o mapeamento entre a ontologia de domínio e o modelo dimensional do DW na Ontologia BI; e configuração de padrões sintáticos e heurísticas para identificação das operações OLAP a partir da pergunta.

Todos os recursos de representação de conhecimento e as terminologias presentes no módulo Modelo e Base de Conhecimento, como a ontologia de domínio, lista de stop-words, dicionários, e padrões idiomáticos estão projetados em conformidade ao idioma português do Brasil.

4.3.1 Ontologia de Domínio

O modelo da ontologia de domínio foi construído com base no modelo dimensional DW da PLI descrito na seção 4.1. Logo, os principais conceitos e terminologias tais como pessoa, formação, instituição de ensino, atividade profissional e acadêmica, produção dentre outros estão presentes nessa ontologia. Por isso, as classes, propriedades e relacionamentos do modelo de domínio assemelham-se respectivamente às dimensões, atributos de dimensão e relacionamentos entre tabelas do DW. Tal ontologia consiste de um modelo parcial da Plataforma Lattes e contempla somente os conceitos relacionados ao referido DW. Vale ressaltar que não é foco deste trabalho descrever ou aprofundar sobre métodos para a criação ou manutenção de ontologias. Cabe ao engenheiro de ontologias conceber o modelo que melhor represente o negócio da organização. A ontologia de domínio, ilustrada na Figura 11, foi modelada com o auxílio do editor de ontologias *Protégé* (STANFORD, 2011) e encontra-se no formato OWL.

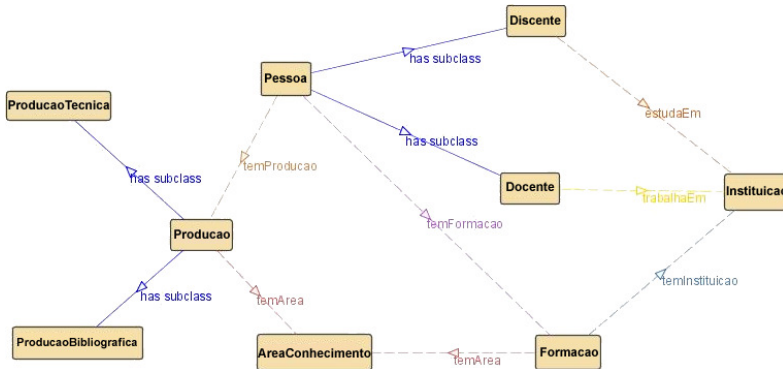


Figura 11 - Ilustração da ontologia de domínio

O modelo da ontologia de domínio ilustrado na Figura 11 possui todos os conceitos e relacionamentos utilizados neste trabalho para a interpretação semântica de perguntas e apoiar a exploração do DW. O contexto de C&T, a qual essa ontologia se refere, possui os seguintes conceitos:

- *Pessoa*: classe que representa as pessoas da PLI. Essa classe possui as propriedades de identificação das pessoas, como nome, data de nascimento, sexo, estado de nascimento, etc. A classe *Pessoa* relaciona-se diretamente com outras duas classes *Formacao* e *Producao* descritas a seguir.
- *Discente*: é uma subclasse de *Pessoa* que possui as propriedades e relacionamentos peculiares dos estudantes da PLI. Conforme visualizado na Figura 11, esta classe relaciona-se com a classe *Instituicao* por meio de um relacionamento denominado *estudaEm* (simboliza a instituição na qual o aluno estuda).
- *Docente*: subclasse de *Pessoa* que modela os docentes propriamente ditos. Essa classe possui um relacionamento chamado *trabalhaEm* que associa o docente à sua instituição em que ministra aula.
- *Formacao*: classe que modela as escolaridades e titulações dos alunos e professores da PLI. Possui propriedades que identificam o curso, o ano de início e término da formação, o seu nível (especialização, graduação, mestrado, doutorado, etc.). Essa classe possui relacionamento com a classe *AreaConhecimento*, que determina a área de conhecimento do curso de formação e relaciona-se também com a classe

Instituicao que denota a instituição de ensino em que a pessoa estuda ou obteve o grau.

- *Producao*: são as produções tanto dos alunos quanto dos professores da PLI. Ela subdivide-se em duas subclasses: *ProducaoBibliografica* e *ProducaoTecnica*. Essa classe se relaciona com a classe *AreaConhecimento* e tal relacionamento identifica as áreas ou campos de estudo referentes às publicações.
- *ProducaoBibliografica*: são as produções bibliográficas como artigos, livros, capítulos de livros das pessoas que participam como autores ou co-autores.
- *ProducaoTecnica*: classe que representa as produções técnicas como patentes de softwares, produtos tecnológicos, processos e técnicas dentre outros.
- *Instituicao*: classe que conceitua as instituições de ensino superior ou ainda os órgãos e unidades internas como departamentos, centro acadêmicos, grupos de pesquisa, dentre outros.
- *AreaConhecimento*: são as áreas de conhecimento da PLI, normalmente associadas às formações acadêmicas e às produções.

Com o intuito de utilizar também relações de sinonímia para a hierarquia de classes, propriedades de classes e relacionamentos, um dicionário é utilizado em conjunto com o formato OWL da ontologia de domínio. A partir do dicionário de sinônimos e da ontologia de domínio, uma estrutura similar a um índice textual foi criada para que o melhor caminho seja localizado pelo Motor de Busca por Similaridade. Tal como nos modelos de RI, esse índice forma uma matriz com o conjunto de termos e seus sinônimos extraídos de cada caminho da ontologia de domínio como mostra a Figura 12 a seguir.

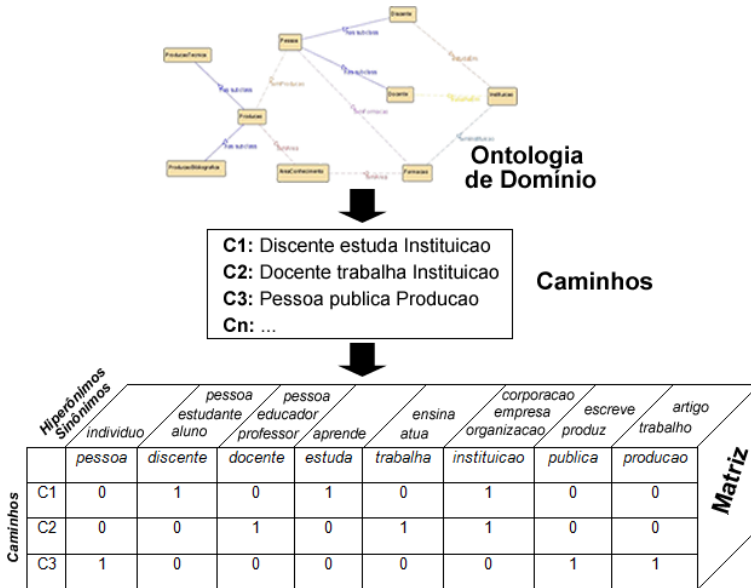


Figura 12 - Ilustração da matriz de caminhos obtida da ontologia de domínio

A Figura 12 ilustra como que alguns dos caminhos (conceitos e relacionamentos da ontologia de domínio) são organizados em um modelo de RI booleano. Este tipo de modelo é adotado na construção da arquitetura para demonstrar como os caminhos podem ser descobertos na prática. No entanto, a arquitetura não limita quanto à adoção de outras formas e estruturas para a organização e localização de caminhos. Assim, outros mecanismos e métodos podem ser utilizados para auxiliar a construção do módulo Modelo e Base de conhecimento com vistas a cumprir o objetivo do módulo Motor de Busca por Similaridade.

Devido à presença de sinônimos e hierarquias de classes na matriz de caminhos, o processo de reformulação baseado na hierarquia de classes e sinônimos é tratado de uma única vez pelo Motor de Busca por Similaridade. Dessa forma, o Motor de Busca por Similaridade desenvolvido atua também como Reformulador, exceto para os casos de reformulação por regras de inferência.

Com base na matriz criada, a pergunta, após passar pelas fases de análise lingüística e reformulação, é usada como vetor de entrada pelo Motor de Busca por Similaridade para a realização de uma busca. Aqui, todas as stop-words são ignoradas e somente são aproveitadas posteriormente pelo Tradutor OLAP. Tal como explicado na seção 3.5, caso o resultado da busca retornar mais de um caminho, o usuário deve

participar do processo de desambiguação até um único caminho ser definido.

4.3.2 Regras de Inferência e Funções

Uma vez que os conceitos do negócio estão modelados na ontologia de domínio, pode-se então definir o conjunto de regras de inferência e também as funções baseadas nos vocábulos do domínio. Este trabalho adota o mecanismo de inferência do framework *Jena* (JENA, 2011) e por isso, todas as regras de inferência estão definidas segundo a sintaxe desse framework. Para mais informações, vide McBride (2002). Já as funções encontram-se em um formato XML desenvolvido especificamente para o protótipo da arquitetura. Um exemplo de função no formato XML que associa algumas terminologias do domínio a um cálculo é ilustrado no Quadro 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<functions>
  <function>
    <input>
      <term> hoje </term>
      <term> atualmente </term>
      <term> agora </term>
      <term> ano atual </term>
    </input>
    <output> ${current.year} </output>
    <concept> lattes:ano </concept>
  </function>
</functions>
```

Quadro 4 - Sintaxe de função para associação entre terminologias e cálculos.

O cálculo representado pela expressão $\${current.year}$ é executado pelo Tradutor OLAP para retornar o ano vigente quando quaisquer um dos termos *hoje*, *atualmente*, *agora* ou *ano atual* estiverem na pergunta e forem classificados como *Função*. Note também que a função acima se refere à propriedade denominada *lattes:ano* (onde, *lattes* representa o *namespace* e o *ano* denota o nome da propriedade) da ontologia de domínio. Logo, se um dos termos (*hoje*, *atualmente*, *agora* ou *ano atual*) for qualificado na pergunta como

Função, este terá o seu valor substituído pelo resultado do cálculo, (no caso o ano atual) sendo que a propriedade *ano* (identificado na ontologia por *lattes:ano*) será usada para formar o filtro da consulta.

O Analisador Lingüístico classifica um termo da pergunta como *Função* caso esse termo esteja contido no conjunto de entradas (no exemplo, tag *<input>*), como ilustra o Quadro 4. O conjunto de cálculos e expressões foram desenvolvidos pontualmente para demonstrar sua aplicação na arquitetura. Tanto as regras quanto as funções não são itens obrigatórios da arquitetura e, portanto, dependendo das análises e necessidade de raciocínio, não precisam ser especificadas. Os Apêndices A e B deste trabalho contêm respectivamente o conjunto de regras de inferência e as funções utilizados para a interpretação de perguntas e retorno de informações a partir do DW.

4.3.3 Ontologia BI

A Ontologia BI, utilizada para relacionar os conceitos da ontologia domínio às dimensões e tabelas do esquema estrela, também foi modelada com o auxílio do editor *Protégé*. Conforme mencionado na seção 3.2, apenas a parte da Ontologia BI que trata da modelagem das estruturas do DW é utilizada neste trabalho.

Para ilustrar como o mapeamento é realizado na prática, a Figura 13 apresenta graficamente os principais construtores das classes *DBCcollection*, *DBAttribute* e *CollectionJoin* da Ontologia BI.

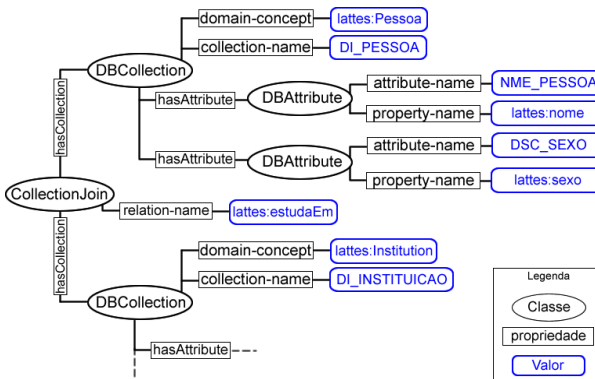


Figura 13 - Ilustração do mapeamento parcial da Ontologia BI

A Figura 13 mostra como uma instância da classe *DBCcollection* da Ontologia BI faz o mapeamento entre a classe *Pessoa* da ontologia de domínio (*lattes:Pessoa*) e a dimensão *DI_PESSOA* do modelo dimensional. Para o mapeamento entre os atributos de dimensão e as propriedades de classe, a classe *DBCcollection* possui um relacionamento com a classe *DBAttribute*. Essa última classe é responsável respectivamente pelo mapeamento entre as propriedades *nome* e *sexo* da classe *Pessoa* com os atributos *NME_PESSOA* e *DSC_SEXO* da dimensão *DI_PESSOA*.

Tal como é realizado o mapeamento entre classes e tabelas e também entre propriedades e atributos, a Ontologia BI prevê ainda a associação entre os relacionamentos de classes e as junções entre tabelas. Esse mapeamento é realizado pela classe *CollectionJoin*. Na Figura 13, uma instância da classe *CollectionJoin* determina que o relacionamento *estudaEm* entre as classes *Pessoa* e *Instituicao* se dá pela relação entre as tabelas *DI_PESSOA* e *DI_INSTITUICAO* do DW. Embora as chaves primárias e estrangeiras da associação não estejam simbolizadas na Figura 13, elas devem ser definidas nos construtores da classe *CollectionJoin*. Para mais informações sobre a Ontologia BI vide Sell (2006, et. al., 2008).

Este trabalho adota os mesmos construtores para a Ontologia BI proposta por Sell, e acrescenta três novos relacionamentos (contidos na classe *DBCcollection* para que esta seja associada a três instâncias da classe *DBAttribute*). Esses novos relacionamentos, denominados *hasMeasure*, *hasGrouping* e *hasFilter* identificam as propriedades padrão de cada classe da ontologia de domínio que devem ser utilizadas como medida, agrupamento ou filtro, respectivamente. Isto serve para reconhecer qual a propriedade a ser usada de uma dada classe quando esta for classificada na pergunta como medida, agrupamento ou filtro. Por exemplo, considerando somente a classe *Pessoa* na pergunta “*Quais pessoas estudam no CTC?*”, para saber que a propriedade *nome* deve ser aplicada como agrupamento (em vez da propriedade *sexo*), a instância da classe *DBCcollection* deve ter o relacionamento *hasGrouping* com a instância de *DBAttribute* que está associada à referida propriedade. Além da propriedade *hasGrouping*, as propriedades *hasMeasure* e *hasFilter* devem ser definidas para a classe *Pessoa*, para quando este conceito ser traduzido pelo Tradutor OLAP como medida e filtro respectivamente. Deste modo, todas as classes da ontologia de domínio possuem propriedades definidas como padrão para os três tipos de elementos da consulta (medida, agrupamento e filtro) na Ontologia BI.

Exclusivamente para as medidas, além de estabelecer a instância da classe *DBAttribute* do relacionamento *hasMeasure*, deve-se ainda especificar qual o método a ser aplicado em sua quantificação. Esses métodos são usados posteriormente pelo Gerenciador de Consultas para executar a requisição OLAP sobre o DW. Tais métodos são semelhantes às funções SQL: *SUM* (somatório), *COUNT* (contagem), *MIN* (valor mínimo), *MAX* (valor máximo) ou *AVG* (média). Neste cenário de aplicação, as medidas das tabelas de fato (exibidas na Figura 10 pelos atributos *TOT_PRODUCAO_BIBLIOGRAFICA*, *TOT_PRODUCAO_TECNICA* e *TOT_ATIVIDADE*) são quantificadas com o método de somatório (função *SUM* da linguagem SQL). Já para a quantificação de conceitos (classes) o valor se dá por meio da contagem (função *COUNT* da linguagem SQL) sobre as dimensões associadas a esses conceitos. Neste caso, os atributos chave das dimensões são usados para a contagem de valores e especificados como medidas na Ontologia BI.

4.3.4 Padrões léxico-sintáticos e Heurísticas

Com base no contexto do domínio e no idioma em questão, o Modelo e Base de Conhecimento deve conter também o conhecimento de como identificar os elementos das consultas OLAP, como medidas, agrupamentos, filtros e junções. Esse conhecimento, armazenado no módulo Modelo e Base de Conhecimento, é formado pelos padrões léxico-sintáticos, posições relativas dos *termos* da pergunta e por heurísticas baseadas nos tipos de stop-words anteriormente citados na Tabela 3. As representações desses padrões e heurísticas são formalizadas no Modelo e Base de Conhecimento por meio de expressões regulares que combinam as posições das entidades reconhecidas da pergunta e os tipos de stop-words. Os padrões e heurísticas utilizados pelo módulo Tradutor OLAP são detalhados na Tabela 5 segundo cada elemento da consulta (medida, agrupamento, filtro ou junções) associado.

Tabela 5 - Padrões e heurísticas utilizados

Nº	Elemento da consulta	Descrição do padrão ou heurística
1	Medida	Todos os termos classificados como classes e

		propriedades de classes posicionados imediatamente à direita de stop-words de quantificação, que estejam seguidos ou não dos tokens <i>E</i> ou <i>OU</i> , são classificados como medidas.
2	Agrupamento	Todos os termos classificados como classes e propriedades de classes posicionados à direita de stop-words de projeção, que estejam seguidos ou não dos tokens <i>E</i> ou <i>OU</i> , são classificados como agrupamentos.
3	Agrupamento	As classes diretas dos termos classificados como instâncias de classes sempre são utilizadas como agrupamentos.
4	Filtro	Todos os termos classificados como instâncias de classes ou identificados como valores de propriedades de classe são utilizados como filtros. Obs.: Caso o termo não esteja à direita de um stop-word de seleção (vide classificação na Tabela 3), o critério de igualdade (=) é utilizado para filtrar o conteúdo, caso contrário, a stop-word de seleção será considerada. Os tokens <i>E</i> e <i>OU</i> presentes entre os valores de propriedades ou instâncias de classes são usados como operadores lógicos para o critério de filtro.
5	Junção ou relacionamento	Todos os relacionamentos entre os conceitos da ontologia de domínio são usados como junções ou relacionamento na requisição OLAP. No próprio mapeamento da Ontologia BI esses relacionamentos devem ter correspondência com as junções ou ligações entre as tabelas e dimensões na consulta.

Para esclarecer como as heurísticas e padrões são aplicados, considere a seguinte pergunta: “*Quantos alunos¹ por sexo² e formação² estudam⁵ no CSE^{3,4} ou CFH^{3,4} ?*”. Os números sobrescritos aos termos remetem ao número do respectivo padrão ou heurística da Tabela 5. Neste exemplo, o termo “*alunos*” refere-se à classe “*Discente*” e é classificado como medida porque possui proximidade à direita da stop-word *Quantos* (pela da regra 1). O atributo configurado na Ontologia BI como medida padrão da classe “*Discente*” deve ser utilizado para

quantificar a informação na consulta. Já os dois termos “*sexo*” e “*formação*”, embora estejam com a mesma numeração (número 2), o primeiro representa uma propriedade da classe “*Pessoa*” e o segundo representa diretamente a classe “*Formacao*” da ontologia de domínio. Neste caso, o primeiro (o termo “*sexo*”) é usado diretamente como agrupamento para a projeção da consulta, já o segundo (o termo “*formação*”), o atributo padrão da classe “*Formacao*” definido como agrupamento é que deve ser aplicado. Já os termos “*CSE*” e “*CFH*”, visto que são instâncias da classe “*Instituicao*” neste exemplo, são usados como filtros (pela regra 4) e também projetados no retorno da consulta (pela regra 3). Novamente, os atributos da classe “*Instituicao*” definidos na Ontologia BI devem ser empregados como filtro e como agrupamento pelo Tradutor OLAP. Como os termos “*CSE*” e “*CFH*” não estão envolvidos com as stop-words de seleção, o critério de igualdade (=) é utilizado no filtro de comparação dos dados na consulta. O operador lógico *OU* (em inglês, *OR*) é utilizado para a construção do critério de filtro dado que o token “*ou*” é informado na pergunta entre os termos. Por fim, por aplicação do padrão número 5 da Tabela 5, o termo “*estuda*”, que simboliza um relacionamento na ontologia de domínio, seria usado como junção para a requisição OLAP. Uma vez que esse relacionamento é dado no caminho retornado pelo Motor de Busca por Similaridade, o Tradutor OLAP reconhece que o termo “*estuda*” relaciona os conceitos “*Pessoa*” e “*Instituição*” neste exemplo.

Tal como o dicionário que auxilia a classificação de termos, as stop-words são organizadas conforme o seu tipo no Modelo e Base de Conhecimento. Essas stop-words são definidas conforme a classificação anteriormente dada pela Tabela 3. A seguir, o conjunto de stop-words utilizado para os exemplos de pergunta no cenário de C&T é exibido na Tabela 6.

Tabela 6 - Lista de stop-words utilizada conforme o tipo

Quantificação	Projeção	Seleção			
		Operador Relacional		Operador Lógico	
		Termo	Operador	Termo	Operador
quantas; quanta; quantos; quanto; quantidade de; qual a quantidade de; qual o total de;	por; segundo; agrupado por; conforme; consoante	acima de; maior que	>	e	AND

número de		abaixo de; menor que;	<		
		a partir de; a partir do; a partir da	>=	ou	OR
		igual a	=		

As stop-words de seleção utilizadas para filtros são relacionadas uma a uma com um operador específico. Por exemplo, a stop-word formada pelos tokens “*Acima de*” está associada ao operador > (maior que); a stop-word “*a partir de*” e suas variantes está associada ao operador >= (maior que ou igual a); e assim por diante. Outras stop-words poderiam ser adicionadas conforme a necessidade e padrão de escrita da organização. Entretanto, esse trabalho limita-se ao conjunto de stop-words exposto na

Tabela 6.

4.4 CONSTRUÇÃO DOS MÓDULOS FUNCIONAIS DA ARQUITETURA

Todos os módulos funcionais da arquitetura são construídos com a linguagem Java (ORACLE, 2011), sendo que alguns desses módulos são compostos por frameworks e bibliotecas desenvolvidos por terceiros. Todas as interfaces de comunicação entre cada módulo desenvolvido estão acopladas conforme as etapas sequenciais da arquitetura. Assim, neste protótipo as entradas e saídas dos módulos são codificadas e representadas nas próprias estruturas de classes Java. A Figura 14 a seguir exhibe a arquitetura tecnológica construída com os componentes e frameworks utilizados no protótipo.

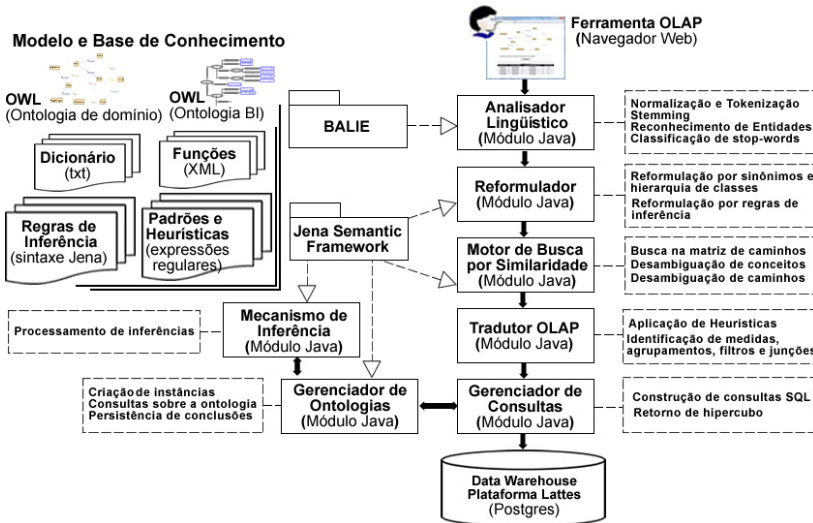


Figura 14 - Arquitetura tecnológica do protótipo

Para auxiliar a análise léxica, sintática e semântica da pergunta, o Analisador Lingüístico foi projetado e integrado ao framework de reconhecimento de entidades denominado BALIE²³ - Baseline Information Extraction (NADEAU, 2011). Esse framework foi adaptado para que as stop-words, dicionário de sinônimos, hierarquias de termos e conceitos do domínio fossem classificados corretamente consoante ao cenário de C&T e ao idioma português. Todos os termos utilizados para identificar os conceitos do domínio bem como valores de propriedades (nome, ano, datas) foram extraídos manualmente do conteúdo das dimensões do DW.

Para a tarefa de reformulação e expansão da pergunta, o módulo Reformulador utiliza tanto o dicionário de sinônimos quanto o conjunto de regras de inferência contidos no Modelo e Base de Conhecimento. Tal como delineado na seção 3.4, o dicionário de sinônimos é a base para a reformulação por hierarquia de classes e sinônimos, já as regras de inferência são usadas para a reformulação baseada nas regras de inferência propriamente ditas. A fim de realizar a leitura das regras de inferência, o Reformulador é combinado ao mecanismo de inferência do framework Jena conforme ilustrado na Figura 14.

²³ BALIE - <http://balie.sourceforge.net>

Exceto para a construção automática da matriz de caminhos descrita na seção 4.3.1, o Motor de Busca por Similaridade foi projetado sem qualquer uso de componentes de terceiros. Neste trabalho, a matriz de caminhos foi criada com estruturas de dados da própria linguagem Java. Ela é gerada automaticamente a partir da ontologia de domínio em OWL, em que todos os caminhos são obtidos juntamente com o uso do dicionário de sinônimos dos conceitos. A fim de realizar essa construção automática, o framework Jena é novamente utilizado para a leitura da ontologia de domínio e organização dos caminhos na matriz. Para ontologias com grandes quantidades de caminhos recomenda-se o uso de índices textuais.

A partir da pergunta reformulada, o Tradutor OLAP foi desenvolvido para aplicar os padrões e heurísticas baseadas nas stop-words explicitadas na Tabela 5. Aqui, as funções e os cálculos associados aos termos do domínio são usados para a criação de filtros nas requisições OLAP. Sendo assim, um conjunto de funções padrão foi implementado exclusivamente para o cenário de aplicação da arquitetura. Os termos e conceitos envolvidos nessas funções podem ser visualizados no Apêndice B.

Com base na requisição OLAP gerada pelo Tradutor OLAP, O Gerenciador de Consultas foi projetado para aplicar consultas SQL sobre o DW. Como a requisição OLAP possui apenas os conceitos do domínio, o Gerenciador de Consultas interage com o Gerenciador de Ontologias para obter as tabelas, as dimensões e suas junções a partir do mapeamento da Ontologia BI. Na prática, o Gerenciador de Consultas é desenvolvido para que as cláusulas SQL da consulta possuam: a) todas as medidas com suas funções de quantificação (como exemplo, *select sum(tot_atividade) ...*); b) todos os agrupamentos projetados com *group by*; c) todos os filtros com os operadores relacionais e lógicos (com os sinais *>*; *>=*; *<=*; *<*; *IN*; *AND*; *OR*, etc.) e; d) todos os relacionamento entre tabelas (como exemplo *Inner join*, *Left join*, etc.). Em particular, para os critérios de igualdade (na linguagem SQL simbolizado pelo *IN* ou *=*) os valores são normalizados em três formas: caixa-alta, caixa-baixa e na forma como foram informados na pergunta. Por exemplo na pergunta “*Quantos alunos estudam na Udesc?*”, O token “*Udesc*”, na qual é um valor de filtro, seria normalizado para *UDESC*, *udesc*, e *Udesc* na cláusula *IN* do SQL. Isto permite que valores escritos de forma diferente possam ser localizados no DW. A seção a seguir traz alguns exemplos de consultas produzidos pelo Gerenciador de Consultas a partir de algumas perguntas interpretadas.

Por fim, os dois últimos módulos funcionais da arquitetura, Gerenciador de Ontologias e Mecanismo de Inferência são implementados também com o auxílio do framework Jena. Este framework além de possibilitar a gerência de bases de conhecimento (criação, atualização e remoção de modelos e instâncias), possui um mecanismo de inferência próprio. Esse mecanismo de inferência é adotado neste trabalho para que o Gerenciador de Ontologias possa realizar o processo de raciocínio conforme as abordagens *in-batch* e *on-the-fly*.

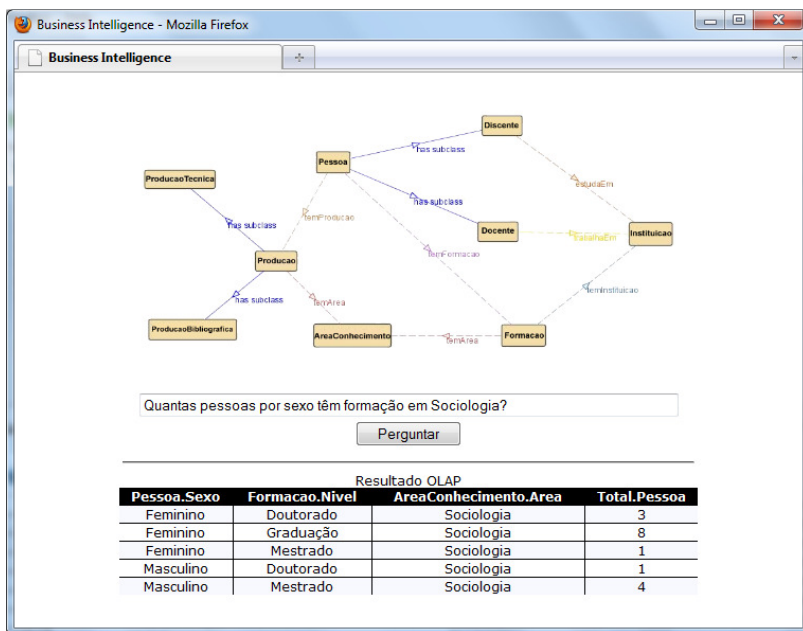
4.5 INTERPRETAÇÃO DE PERGUNTAS E RESULTADOS OBTIDOS

Esta seção exemplifica como os módulos da arquitetura realizam a interpretação semântica de perguntas até a obtenção de respostas dentro do domínio de C&T. Os exemplos de perguntas cobrem os cenários e problemas de interpretação de linguagem natural, como a classificação de termos, as questões de ambigüidade de entidades e de caminhos, reformulação de perguntas, aplicação de padrões e heurísticas para a construção das consultas.

Para demonstrar a viabilidade da arquitetura, um protótipo foi projetado de modo que qualquer navegador Web possa ser utilizado como ferramenta analítica. Esse protótipo coloca o tomador de decisão em contato com os módulos da arquitetura por meio de uma interface simples e muito similar aos sistemas de busca da Web. A interface analítica permite basicamente: a) uma única entrada em linguagem natural; b) a interação com o usuário para os casos de ambigüidades de conceitos e de caminhos; c) a exibição das informações estratégicas sumarizadas ao tomador de decisão em atendimento à pergunta. Por motivos de simplificação, a interface foi construída para que o cubo OLAP mostre as descrições das dimensões nos cabeçalhos de coluna e o restante do conteúdo nas linhas. As subseções adiante estão organizadas para compreensão dos módulos funcionais da arquitetura conforme a complexidade das perguntas e respostas na prática.

4.5.1 Pergunta com conceitos do domínio de C&T

A Figura 15 a seguir exibe a interface analítica em um navegador Web, onde a ontologia do domínio de C&T é graficamente ilustrada juntamente com as regiões de entrada da pergunta e de visualização dos resultados. Essa figura traz um exemplo de pergunta no contexto de C&T já com a sua respectiva resposta. Os passos até a obtenção do cubo OLAP a partir dessa pergunta são detalhados adiante. Esses passos são praticamente os mesmos para todas as subseções a seguir. Por conta disso, as seções posteriores dão ênfase nas particularidades da resolução de ambigüidades, uso de funções e aplicação de inferências.



Business Intelligence - Mozilla Firefox

Business Intelligence

Ontology Diagram:

- Pessoa** (Superclass)
 - ProducaoTecnica** (Subclass)
 - Producao** (Subclass)
 - ProducaoBiografica** (Subclass)
 - Discente** (Subclass)
 - Docente** (Subclass)
- Formacao** (Superclass)
 - FormacaoArea** (Subclass)
 - FormacaoInstituicao** (Subclass)
- AreaConhecimento** (Superclass)
 - Sociologia** (Subclass)
- Instituicao** (Superclass)
 - Sociologia** (Subclass)

Quantas pessoas por sexo têm formação em Sociologia?

Perguntar

Resultado OLAP

Pessoa.Sexo	Formacao.Nivel	AreaConhecimento.Area	Total.Pessoa
Feminino	Doutorado	Sociologia	3
Feminino	Graduação	Sociologia	8
Feminino	Mestrado	Sociologia	1
Masculino	Doutorado	Sociologia	1
Masculino	Mestrado	Sociologia	4

Figura 15 - Ilustração do protótipo da interface analítica

A pergunta “*Quantas pessoas por sexo têm formação em Sociologia?*”, exibida na Figura 15, é um exemplo relativamente simples que não apresenta ambigüidades, conclusões de regras de inferência e nem funções. Isto é, apenas as propriedades de classes, as classes e seus respectivos relacionamentos da ontologia de domínio de C&T são envolvidos na pergunta.

Inicialmente, essa pergunta, após ser informada na interface analítica, deve passar pelo processo de análise léxica, sintática e semântica do Analisador Lingüístico. Por consulta aos dicionários e conceitos da ontologia do domínio de C&T, o Analisador Lingüístico determina a classificação de cada termo da pergunta, que neste caso seria: “*Quantas*” (stop-word de quantificação); “*pessoas*” (classe *Pessoa*); “*por*” (stop-word de projeção); “*sexo*” (propriedade da classe *Pessoa*); “*têm*” (*token* não reconhecido); “*formação*” (classe *Formacao*); “*em*” (*token* não reconhecido) e; “*Sociologia*” (instância da classe *AreaConhecimento*). Os *tokens* não reconhecidos (nesse exemplo, *têm* e *em*) não estão entre os termos do dicionário e hierarquias de classes, e dessa forma não possuem classificação definida.

Antes de localizar o melhor caminho com base no modelo da ontologia de domínio, o Reformulador neste exemplo deve substituir a instância “*Sociologia*” por sua classe direta “*AreaConhecimento*”. Já que a pergunta não possui termos classificados como conclusões de regras, não há reformulação baseada em regras de inferência. No entanto, para este exemplo ocorre a reformulação por sinônimos e hierarquias de classes. Como citado na seção 4.3.1, essa reformulação é efetuada pelo Motor de Busca por Similaridade que acumula o papel do módulo Reformulador.

Com isso, a pergunta reformulada que deve ser usada como entrada para a busca pelo Motor de Busca por Similaridade apresenta somente os termos: “*Pessoa sexo tem Formacao em AreaConhecimento*”. Observe que os termos classificados como stop-words foram ignorados no vetor de entrada para a busca. Os termos *têm* e *em*, mesmo sem classificação, são usados na busca, sendo que os caracteres especiais (acentos, hífen) são removidos.

Somente os caminhos que possuem o maior número de conceitos identificados a partir do vetor de entrada são retornados pelo Motor de Busca por Similaridade. Logo, o melhor caminho da ontologia do domínio de C&T é aquele representado neste exemplo na notação N3 por: (*Pessoa temFormacao Formacao*) \wedge (*Formacao temArea AreaConhecimento*). Note que os outros caminhos menores contidos nesse caminho, como aqueles formados por apenas um único vértice (*Pessoa*; *Formacao* ou *AreaConhecimento*) e aqueles formados pelas triplas; (*Pessoa temFormacao Formacao*) ou (*Formacao temArea AreaConhecimento*) não devem ser retornados na busca. Com isso, nesse caso um único caminho é obtido sem a necessidade de participação do tomador de decisão para a desambiguação de entidades e caminhos.

Com o melhor caminho definido, o conjunto de padrões e heurísticas comentados na Tabela 5 é aplicado pelo Tradutor OLAP. Assim, a partir dos elementos classificados e dos tipos de stop-words, a requisição OLAP gerada possui como medida: a classe *Pessoa*; como agrupamento: a propriedade *sexo*, e as classes *Formacao* e *AreaConhecimento*; e como filtro: o termo *Sociologia*, que é uma instância da classe *AreaConhecimento*. Os relacionamentos (*temFormacao* e *temArea*) são também traduzidos na requisição como os relacionamentos (junções) que interligam os conceitos do domínio.

A partir da requisição OLAP, o Gerenciador de Consultas atua com o módulo Gerenciador de Ontologias para montar e aplicar a consulta com as dimensões ou tabelas de fato associadas aos conceitos identificados na pergunta (no caso, *Pessoa*, *sexo*, *temFormacao*, *Formacao*, *temArea*, *AreaConhecimento*). Para tal, as instâncias da Ontologia BI que mapeiam esses conceitos à estrutura do DW são recuperadas pelo Gerenciador de Ontologias. Após localizar essas instâncias, o Gerenciador de Ontologias informa ao Gerenciador de Consultas as dimensões, os atributos de dimensão, as tabelas de fato e como elas são interligadas para a criação da consulta SQL.

Como visto, as propriedades de classes são na maioria das vezes associadas aos atributos de dimensão na Ontologia BI. No entanto, somente uma propriedade (propriedade *sexo* da classe *Pessoa*) foi informada e reconhecida na pergunta. Como mencionado na seção 4.3.3, de acordo com a tradução do Tradutor OLAP, um atributo padrão deve ser definido como medida, agrupamento ou filtro para a classe. Assim, quando uma classe não possui qualquer propriedade explicitamente informada, o atributo configurado como padrão na Ontologia BI é aquele que deve ser utilizado na consulta.

Destarte, considerando a configuração da Ontologia BI e a pergunta deste exemplo, o atributo *SEQ_ID_PESSOA* da dimensão *DI_PESSOA* é utilizado como medida padrão e correspondente à classe *Pessoa*. Já a classe *Formacao*, que corresponde na Ontologia BI à dimensão *DI_FORMACAO*, possui como agrupamento o atributo *DSC_NIVEL_FORMACAO*. Por fim, a classe *AreaConhecimento* tem como agrupamento e também filtro o atributo *NME_AREA_CONHEC* da dimensão *DI_AREA_CONHECIMENTO*.

Para saber quais as junções que relacionam as dimensões *DI_PESSOA*, *DI_FORMACAO* e *DI_AREA_CONHEC* na consulta, o Gerenciador de Consultas utiliza os relacionamentos *temFormacao* e *temArea* obtidos do caminho identificado. Igualmente, o Gerenciador de Consultas obtém as informações da Ontologia BI por meio do

Gerenciador de Ontologias para produzir as junções entre as tabelas. Essas informações configuradas na Ontologia BI indicam quais os atributos das dimensões usados para realizar a junção e o tipo de junção (*inner join*, *left join*, etc.) Por fim, a consulta SQL resultante do Gerenciador de Consultas que responde a pergunta é apresentada no Quadro 5 a seguir.

```

SELECT t0.DSC_SEXO AS "c1",
t1.DSC_NIVEL_FORMACAO AS "c2",
    t2.NME_AREA_CONHEC AS "c3",
    COUNT(DISTINCT t0.SEQ_ID_PESSOA) AS "m1"
FROM LATTES.DI_PESSOA t0
    INNER JOIN LATTES.DI_FORMACAO t1
        ON (t0.SEQ_ID_FORMACAO = t1.SEQ_ID_FORMACAO)
    INNER JOIN LATTES.DI_AREA_CONHECIMENTO t2
        ON (t1.SEQ_ID_AREA = t2.SEQ_ID_AREA_CONHEC)
WHERE t2.NME_AREA_CONHEC IN
    ('SOCIOLOGIA', 'sociologia', 'Sociologia')
GROUP BY t0.DSC_SEXO, t1.DSC_NIVEL_FORMACAO, t2.NME_AREA_CONHEC
ORDER BY t0.DSC_SEXO, t1.DSC_NIVEL_FORMACAO, t2.NME_AREA_CONHEC

```

Quadro 5 - Consulta SQL gerada a partir da pergunta do domínio de C&T.

O Gerenciador de Consultas, após executar a consulta SQL do Quadro 5 sobre o DW, disponibiliza o cubo OLAP com os cabeçalhos de coluna contendo as informações de classes e propriedades do domínio (veja a Figura 15). Conforme comentado na seção 4.4, observe que o valor *Sociologia*, usado como filtro na cláusula *WHERE* da consulta SQL, é colocado em caixa-alta, em caixa-baixa e na forma como é informado na pergunta. Por padrão, todos os agrupamentos são ordenados (cláusula *ORDER BY*) na resposta.

4.5.2 Pergunta com ambigüidades de conceitos e caminhos

O protótipo construído também permite a interação com o usuário para os casos de ambigüidade. Para demonstrar como os módulos da arquitetura interagem na presença de ambigüidades, considere agora a seguinte pergunta: “*Qual a quantidade de pessoas da*

UFSC por estado?”. Esta pergunta possui duas ambigüidades, uma já identificada na primeira etapa feita pelo Analisador Lingüístico e outra somente verificada pelo Motor de Busca por Similaridade. Todas as ambigüidades devem ser resolvidas pelo usuário em um processo interativo com o módulo Motor de Busca por Similaridade.

A ambigüidade reconhecida pelo Analisador Lingüístico refere-se ao termo “*estado*” redigido na pergunta. No modelo da ontologia do domínio de C&T deste trabalho, esse termo representa uma propriedade que é compartilhada entre as classes *Pessoa* e *Instituicao*. Como há dois termos na pergunta que remetem respectivamente às classes *Pessoa* e *Instituicao* (termos “*pessoas*” – classe *Pessoa* e; “*UFSC*” – instância de *Instituicao*) a ambigüidade é encontrada. No caso, para a classe *Pessoa*, essa propriedade modela os estados da federação onde a pessoa nasceu. Já para a classe *Instituicao*, a propriedade “*estado*” situa o estado da federação de endereço da instituição.

A segunda ambigüidade da pergunta está relacionada aos possíveis caminhos ou relacionamentos entre a classe *Pessoa* e *Instituicao*. Como ilustrado na ontologia de domínio na Figura 11, a classe *Pessoa* pode se relacionar com a classe *Instituicao* por meio de três relacionamentos: 1) (*Discente estudaEm Instituicao*); 2) (*Docente trabalhaEm Instituicao*) ou 3) (*Pessoa temFormacao Formacao*) \wedge (*Formacao temInstituicao Instituicao*). No primeiro e segundo caminho, ambas as classes *Discente* e *Docente* são subclasses de *Pessoa*. No terceiro caminho, a relação de *Pessoa* com *Instituicao* é realizada por um relacionamento intermediário com a classe *Formacao*.

Como as ambigüidades são resolvidas somente pelo Motor de Busca por Similaridade, após a pergunta passar pelo Analisador Lingüístico e pelo Reformulador, a desambiguação de caminhos deve ser inicialmente tratada. Dá-se preferência para a resolução de ambigüidades de caminhos, pois é possível que as demais ambigüidades sejam resolvidas ou talvez reduzidas. A Figura 16 exhibe como o tomador de decisão participa do processo de resolução de ambigüidades de caminhos e também de conceitos específicos (no caso, a propriedade *estado*).

por conseguinte, na consulta sobre o DW, é o estado de nascimento da Pessoa.

Subseqüente a resolução de ambigüidades e a execução das mesmas tarefas descritas anteriormente na seção anterior, o cubo OLAP com as informações sumarizadas pode ser obtido do DW. A Figura 17 a seguir mostra o resultado na interface analítica juntamente com a consulta SQL produzida pelo Gerenciador de Consultas.

```

SELECT t1.DSC_NIVEL_FORMACAO AS "c1",
       t2.SGL_INSTITUICAO AS "c2",
       t0.DSC_ESTADO_NASCIMENTO AS "c3",
       COUNT(DISTINCT t0.SEQ_ID_PESSOA) AS "m1"
FROM LATTES.DI_PESSOA t0
     INNER JOIN LATTES.DI_FORMACAO t1
       ON t0.SEQ_ID_FORMACAO = t1.SEQ_ID_FORMACAO
     INNER JOIN LATTES.DI_INSTITUICAO t2
       ON t1.SEQ_ID_INSTITUICAO = t2.SEQ_ID_INSTITUICAO
WHERE t2.SGL_INSTITUICAO IN ('UFSC', 'ufsc', 'UFSC')
GROUP BY t1.DSC_NIVEL_FORMACAO, t2.SGL_INSTITUICAO, t0.DSC_ESTADO_NASCIMENTO
ORDER BY t1.DSC_NIVEL_FORMACAO, t2.SGL_INSTITUICAO, t0.DSC_ESTADO_NASCIMENTO

```

Qual a quantidade de pessoas da UFSC por estado?

Perguntar

Resultado OLAP

Formacao.Nivel	Instituicao.Sigla	Pessoa.Estado	Total.Pessoa
Doutorado	UFSC	Minas Gerais	3
Doutorado	UFSC	Pernambuco	1
Doutorado	UFSC	Rio de Janeiro	7
Doutorado	UFSC	Rio Grande do Sul	1
Doutorado	UFSC	Santa Catarina	1
Doutorado	UFSC	São Paulo	1
Doutorado	UFSC	Não Informado	1
Especialização	UFSC	Rio de Janeiro	5
Especialização	UFSC	Não Informado	1
Graduação	UFSC	Amazonas	1
Graduação	UFSC	Bahia	1
Graduação	UFSC	Ceará	1
Graduação	UFSC	Goiás	3
Graduação	UFSC	Mato Grosso do Sul	1
Graduação	UFSC	Minas Gerais	5
Graduação	UFSC	Pará	1
Graduação	UFSC	Paraíba	2
Graduação	UFSC	Pernambuco	2
Graduação	UFSC	Rio de Janeiro	31
Graduação	UFSC	Rio Grande do Sul	1
Graduação	UFSC	Santa Catarina	6

Figura 17 - Consulta SQL e Cubo OLAP projetado na interface analítica.

Na Figura 17, embora a pergunta inicial tenha apenas três conceitos do domínio envolvidos, *Pessoa* (classe), *Instituicao* (classe) e *estado* (propriedade), a consulta leva em conta todos os conceitos presentes no caminho. Como o caminho escolhido no processo de desambiguação possui também a classe *Formacao*, a resposta é projetada com o atributo padrão mapeado na Ontologia BI para essa classe (no caso, o atributo *DSC_NIVEL_FORMACAO*).

4.5.3 Pergunta com inferência

Com o intuito de descobrir novos conhecimentos a partir das fontes de dados da organização, as análises das informações estratégicas podem ser conjugadas com o processo de inferência previsto na arquitetura. Para demonstrar como esse processo pode apoiar às ações estratégicas das organizações, suponha que um tomador de decisão queira divulgar as normas do estatuto da universidade para os alunos que recém ingressaram. Assim, no âmbito de C&T tratado, esse gestor deseja saber o número de alunos brasileiros e estrangeiros que ingressaram na UFSC para solicitar a confecção de materiais de divulgação do estatuto universitário conforme o idioma. Para poder saber para quantos alunos esse material deve ser entregue, uma regra de inferência pode ser criada. No caso, essa regra teria as premissas para a identificação das pessoas que estão ingressando na universidade, ou seja, os calouros da universidade. A regra que explicita o relacionamento denominado *calouro* entre pessoa e a instituição é exibida no Quadro 6.

<i>(?aluno</i>	<i>rdf:type</i>	<i>Pessoa)</i>
<i>(?aluno</i>	<i>temFormacao</i>	<i>?formacao)</i>
<i>(?formacao</i>	<i>rdf:type</i>	<i>Formacao)</i>
<i>(?formacao</i>	<i>anoInicio</i>	<i>currentYear(?ano))</i>
<i>(?aluno</i>	<i>estudaEm</i>	<i>(?instituicao)</i>
<i>(?instituicao</i>	<i>rdf:type</i>	<i>Instituicao)</i>
→		
<i>(?aluno</i>	<i>calouro</i>	<i>?instituicao)</i>

Quadro 6 - Regra de inferência para explicitar o conceito Calouro.

No Quadro 6, a especificação formal da regra de inferência denota que caso o ano de início de formação de um aluno na instituição seja igual ao ano atual, esse aluno é considerado um calouro dessa instituição. A partir dessa regra, a pergunta que atende à necessidade de informação do tomador de decisão poderia ser: “*Qual a quantidade de*

calouros da UFSC por nacionalidade?”. Por aplicação da regra de inferência, essa pergunta somente deve retornar as informações dos alunos que possuem o relacionamento *calouro* com a instituição UFSC.

Para o correto processamento da regra pelo Mecanismo de Inferência, a ontologia de domínio deve ter o relacionamento *calouro* entre *Pessoa e Instituicao* em seu modelo. Conseqüentemente, esse relacionamento deve estar também contido na matriz de caminhos para a localização pelo Motor de Busca por Similaridade. Na abordagem de raciocínio *in-batch*, o relacionamento *calouro* deve estar diretamente mapeado ao modelo tripla na Ontologia BI, já que ele é uma conclusão de regra de inferência. Na abordagem *on-the-fly*, da mesma forma esse relacionamento deve ser mapeado aos relacionamentos dos conceitos que determinam o termo calouro. No entanto, o Gerenciador de Consultas ainda deve esperar o resultado da inferência sobre a base de conhecimento para criar a consulta OLAP. Independente da abordagem adotada para o raciocínio (*in-batch* ou *on-the-fly*), a resposta da pergunta obtida a partir do DW deve ser exatamente igual. A seguir, explica-se sucintamente como a pergunta acima é formalizada até a obtenção do cubo OLAP para as duas abordagens de inferências.

Resumidamente, na etapa de interpretação da pergunta o Analisador Lingüístico classificaria os termos como: “*Qual a quantidade de*” (*stop-word* de quantificação); “*calouros*” (conclusão de regra de inferência); “*da*” (*token* não reconhecido); “*UFSC*” (instância da classe *Instituicao*); “*por*” (*stop-word* de projeção) e; “*nacionalidade*” (propriedade da classe *Pessoa*). Conforme explana a seção 3.4.2, a pergunta oriunda do processo de reformulação resultaria em: “*Qual a quantidade de Pessoa calouro Instituicao da Instituicao por nacionalidade?*”. Ressalta-se que o termo *Instituicao* é duplicado na pergunta, sendo o primeiro termo explicado pela reformulação baseado nas regras de inferência (relacionamento *Pessoa calouro Instituicao do* Quadro 6) e o segundo termo refere-se à substituição da instância *UFSC* por sua classe direta *Instituicao*. Essa forma de reformular a pergunta seria útil para localizar caminhos com auto-relacionamentos entre conceitos, no entanto, neste caso isto não se aplica. Essa pergunta, que é insumo para o Motor de Busca por Similaridade, seria desmembrada no vetor com os seguintes termos: “*Pessoa calouro Instituicao da Instituicao nacionalidade*”. Com base na matriz de caminhos, o caminho obtido nesse exemplo seria composto basicamente por: (*Pessoa calouro Instituicao*). Deste modo, as heurísticas e padrões sintáticos determinados na Tabela 5 conduziram o Tradutor OLAP a definir: a classe *Pessoa* como medida; a classe *Instituicao* e a propriedade

nacionalidade como agrupamentos; a instância da classe *Instituicao UFSC* como valor do filtro *e*; e o relacionamento *calouro* como ligação entre os conceitos *Pessoa* e *Instituicao*.

Na abordagem de raciocínio *in-batch*, todos os resultados da aplicação de inferências devem estar atualizados no modelo tripla. Nessa abordagem, o mapeamento na Ontologia BI deve associar o relacionamento *calouro* entre a classe *Pessoa* e a classe *Instituicao* ao modelo tripla. Nesse mapeamento, as dimensões *DI_PESSOA* e *DI_INSTITUICAO* são ligadas à tabela *TRIPLE_FACT* do modelo tripla (vide Figura 10) para que a dimensão *INFERRED_PREDICATE* seja utilizada para filtrar o conceito desejado (no caso o relacionamento *calouro*). Como discutido na seção 3.10, o atributo *PREDICATE_NAME* da dimensão *INFERRED_PREDICATE* é usado como filtro na consulta SQL. Com isso, o Gerenciador de Consulta cria a consulta *SQL* que combina as informações já inferidas e armazenadas no modelo tripla com as informações das dimensões do DW. A Figura 18 dispõe o a consulta *SQL* produzida pelo Gerenciador de Consultas e o resultado (cubo OLAP) com as informações extraídas do DW.

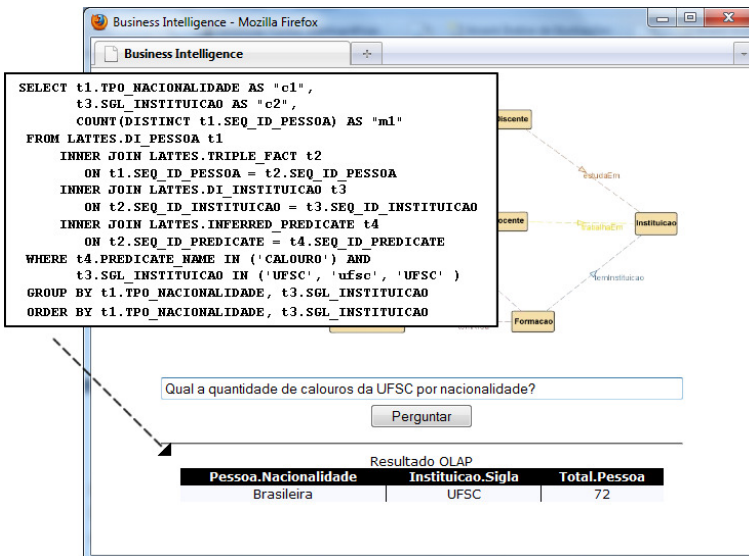


Figura 18 - Visualização do resultado da pergunta com aplicação de inferência

Com o resultado das análises, o tomador de decisão neste exemplo não teria preocupação com a produção de materiais de divulgação do estatuto da UFSC para os calouros estrangeiros. Isto

porque a resposta recuperada a partir das fontes de dados informa que todos os 72 alunos ingressantes no ano atual são brasileiros. (Obs.: A amostra de dados derivada e armazenada no modelo tripla não apresenta dados de alunos estrangeiros para o conceito *calouro*).

Caso a abordagem *on-the-fly* seja adotada, o Gerenciador de Consultas deve interagir com o Gerenciador de Ontologias para que os resultados das conclusões de regras do Mecanismo de Inferência sejam considerados na consulta. Todas as instâncias do modelo da ontologia devem estar presentes na Base de Conhecimento para que o Mecanismo de Inferência execute as regras de inferência. Para o caso da regra de inferência que determina os calouros da universidade, o resultado do processo de inferência é um conjunto de triplas da relação *Pessoa calouro Instituicao*. Conforme tratado por Sell (2006), neste caso os valores identificadores (ou chaves primárias) desses conceitos devem ser usados na consulta como filtro das respectivas dimensões a que estão relacionados na Ontologia BI. Dessa forma, após a relação *calouro* entre uma instância da classe *Pessoa* e *Instituicao* for determinada, o valor chave resultante das dimensões associadas a esses conceitos são usados como valores no critério de filtro. Considerando que as junções das dimensões para a relação *calouro* mapeada na Ontologia BI indica que os relacionamentos a serem usados são aqueles associados ao (*Pessoa temFormacao Formacao*) e (*Formacao temInstituicao Instituicao*), a consulta resultante do processo de inferência *on-the-fly* é mostrada a seguir no Quadro 7.

```

SELECT t0.TPO_NACIONALIDADE AS "c1",
       t2.SGL_INSTITUICAO AS "c2",
       COUNT(DISTINCT t0.SEQ_ID_PESSOA) AS "m1"
FROM LATTES.DI_PESSOA t0
      INNER JOIN LATTES.DI_FORMACAO t1
            ON (t0.SEQ_ID_FORMACAO = t1.SEQ_ID_FORMACAO)
      INNER JOIN LATTES.DI_INSTITUICAO t2
            ON (t1.SEQ_ID_INSTITUICAO = t2.SEQ_ID_INSTITUICAO)
WHERE t2.SGL_INSTITUICAO IN ('UFSC', 'ufsc', 'UFSC') AND
      t0.SEQ_ID_PESSOA IN (45,94,217,283,357,443,...) AND
      t2.SEQ_ID_INSTITUICAO IN (21,7621,288,36,...)
GROUP BY t0.TPO_NACIONALIDADE, t2.SGL_INSTITUICAO
ORDER BY t0.TPO_NACIONALIDADE, t2.SGL_INSTITUICAO

```

Quadro 7 - Consulta gerada no processo de inferência *on-the-fly*.

Obs.: As reticências (...) presentes na cláusula *IN* simbolizam que mais valores deveriam constar como filtro.

A consulta do Quadro 7 destaca em negrito os valores resultantes do processo de inferência usados para escrever o filtro de dados na abordagem *on-the-fly*. Nesta consulta, esse filtro selecionaria somente aqueles alunos e instituições que satisfazem a regra de inferência para o conceito *calouro*. Os 72 estudantes devem estar contemplados no filtro e na totalização do cubo OLAP. Observe que além dos valores chave das pessoas, há também os valores de filtro para as instituições e órgãos resultantes das conclusões da regra de inferência. Embora esses valores sejam usados na criação da consulta, somente a instituição UFSC é retornada, visto que outro filtro para essa instituição é considerado na pergunta. (Obs.: Por motivos de espaço, apenas um pequeno conjunto de valores aparecem no filtro nas cláusulas *IN* da consulta SQL do Quadro 7).

4.5.4 Pergunta com função

Para ilustrar o cenário em que as perguntas envolvam termos classificados como *Função* pelo Analisador Lingüístico, suponha a seguinte pergunta: “*Quantas produções bibliográficas foram escritas a partir do ano passado?*”. Nessa pergunta, os termos da pergunta seriam categorizados como: “*Quantas*” (stop-word de quantificação); “*produções bibliográficas*” (classe *ProducaoBibliografica*); “*foram*” (token não reconhecido); “*escritas*” (token não reconhecido); “*a partir do*” (stop-word de seleção); “*ano passado*” (Função).

O Reformulador ao analisar os termos classificados não necessita modificar ou expandir a pergunta. Isto porque os termos encontram-se adequados conforme o contexto da ontologia de domínio, nenhuma instância ou valor de propriedade é identificado na pergunta e não há conclusões de regras de inferência. Por conta disso, o Motor de Busca por Similaridade efetua a busca com os seguintes termos de entrada: “*producoes bibliográficas foram escritas ano passado*”. Do mesmo modo, todas as stop-words são removidas para a busca do melhor caminho. Por comparação na matriz de caminhos, o caminho retornado pelo Motor de Busca por Similaridade é formado apenas por um único vértice – classe *ProducaoBibliografica*. Isto ocorre pois os tokens “*foram*”, “*escritas*” e “*ano passado*” não contribuem para a descoberta

das demais relações da classe *ProducaoBibliografica* com os outros conceitos. Assim, apenas uma classe da ontologia forma o caminho neste exemplo.

O próximo passo é então traduzir os conceitos em medidas, agrupamentos, filtros e junções. Consoante às heurísticas e padrões dispostos na Tabela 5, a classe *ProducaoBibliografica* é traduzida como medida por sua proximidade à direita da stop-word “*Quantas*”. Os demais termos, com exceção da função “*ano passado*” e da stop-word “*a partir do*”, são *tokens* não reconhecidos e; portanto, não são considerados pelo Tradutor OLAP. Já que o termo “*ano passado*” foi classificado como *Função* pelo Analisador Lingüístico, a função correspondente a esse termo deve ser calculada pelo Tradutor OLAP para a criação de filtros. A função relacionada a esse conceito pode ser vista no Apêndice B.

Então, o Tradutor OLAP deve consultar os repositórios da arquitetura para reconhecer qual a função a ser aplicada para a expressão “*ano passado*”. Neste exemplo a função está associado à propriedade *ano* da classe *ProducaoBibliografica*. Com base no ano atual, o cálculo resultante deverá retornar o valor numérico que representa o ano anterior. O critério de filtro, em vez de comparação por igualdade, leva em conta também a stop-word de seleção usada na pergunta e composta pelos tokens “*a partir do*”. Como mostrado na Tabela 6, essa stop-word simboliza o operador \geq (*maior que ou igual a*) que deve ser usado pelo Tradutor OLAP para criar a requisição OLAP. Destarte, após obter o mapeamento da classe *ProducaoBibliografica* e da sua propriedade *ano* na Ontologia BI, o Gerenciador de Consultas pode criar a consulta sobre o DW. Considerando que a função retornaria o valor 2009 a consulta juntamente com a resposta gerada pode ser visualizada na Figura 19.

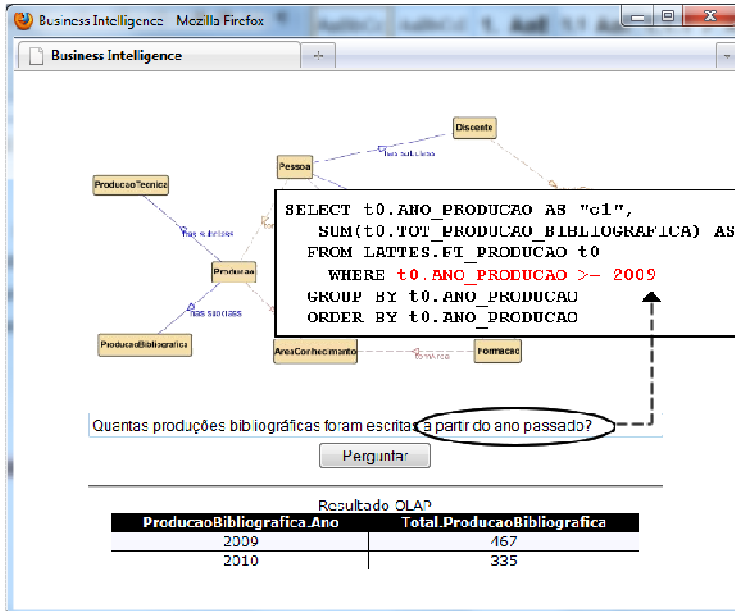


Figura 19 - Consulta e resultado gerados para uma pergunta com função.

A interface analítica, neste caso, mostraria ao gestor o total de produções bibliográficas a partir do ano de 2009. O resultado da análise ilustrado na Figura 19 indica que para o ano de 2009 foi registrado um total de 467 produções bibliográficas e em 2010, um total de 335.

A Figura 19 traz o filtro criado pela aplicação da função em destaque (na cor vermelha) na consulta executada pelo Gerenciador de Consultas sobre o DW. Esse filtro é montado sempre com a propriedade à esquerda do valor, de modo que alguns operadores como (`<`, `>`, `<=` ou `>=`) não tenham a lógica invertida para a formação do critério de seleção. Como nos outros exemplos anteriores, toda a propriedade usada para filtrar o conteúdo é também projetada no retorno da resposta. Por isso, a propriedade *ano* da classe *ProducaoBibliografica* aparece tanto como filtro quanto agrupamento na resposta.

4.6 AVALIAÇÃO SOBRE OS RESULTADOS OBTIDOS NO DOMÍNIO DE C&T

As análises apresentadas anteriormente abrangem alguns exemplos de perguntas conforme os tipos e complexidades dos elementos envolvidos. Esses elementos, como funções, conclusões de regras de inferências, dentre outros podem ser combinados em uma única análise. Assim, conforme a necessidade de informação, as etapas que envolvem funções, conclusões de regras de inferência e inclusive resolução de ambigüidades podem ser realizadas a partir de uma mesma pergunta.

As perguntas não necessariamente precisam obedecer fielmente à norma culta da língua em questão. Desconstruções gramaticais inclusive sem concordâncias como na pergunta “*Por ano, agrupado por estado quanto pessoas estudar UFSC?*” podem ser perfeitamente informadas para obter o cubo OLAP. Perguntas desse tipo foram testadas na arquitetura e os resultados foram os mesmos para as perguntas sintaticamente corretas. Para que isso ocorra, salienta-se que as perguntas devem possuir pelo menos os conceitos do domínio (com sinônimos e hierarquia de classes) para a localização do melhor caminho na ontologia e seguir os padrões e heurísticas para a identificação dos elementos da consulta.

Para verificar a confiabilidade dos dados e também o desempenho das análises na arquitetura, um conjunto de perguntas foi submetido como amostra de teste, combinando os quatro cenários dispostos nas subseções anteriores. Inicialmente algumas perguntas foram erroneamente interpretadas pelos módulos da arquitetura. No entanto, verificou-se que as falhas eram causadas por: ausência de alguns sinônimos para os conceitos da ontologia; falta de definições de termos para as hierarquias de classes; mapeamento incompleto de algumas stop-words; ou ainda, padrões e heurísticas indevidamente especificados no Modelo e Base de Conhecimento.

Por incorporar algumas técnicas de estudos relacionados, a arquitetura pode apresentar tipos de falhas semelhantes para a interpretação da pergunta. Lopez (et. al., 2007) discute sobre alguns dos possíveis tipos de falhas, que nesta proposta podem ser traduzidos em: falhas na análise lingüística; falhas na reformulação, falhas na localização de caminhos da ontologia e falhas da identificação dos elementos da consulta. As falhas de análise lingüísticas referem-se aos erros nas tarefas de reconhecimento de entidades e resolução de co-

referências ou anáforas na pergunta. Este caso ocorre quando o Analisador Lingüístico classifica equivocadamente um termo ou ainda não encontra a classe para o referido termo no Modelo de Base de Conhecimento. Do mesmo modo, os sinônimos e hierarquias de classes da tarefa de reformulação podem ser associados incorretamente. Quanto à localização de caminhos, o Motor de Busca por Similaridade pode falhar ao não abranger todo o contexto de busca informado na pergunta, não recuperar alguns caminhos candidatos ou também recuperar caminhos não relacionados. A última falha é relacionada à identificação de medidas, agrupamentos, filtros e ligações entre os conceitos. Esse tipo de falha ocorre quando os padrões e heurísticas não estão em conformidade com o modo de escrita e sintaxe da pergunta.

Após as manutenções iterativas no Modelo e Base de Conhecimento com o intuito de corrigir os tipos de falhas supracitados, as perguntas que no início obtiveram respostas incorretas, retornaram o hipercubo conforme o esperado. Evidentemente, todas as perguntas foram elaboradas em conformidade ao escopo definido pela ontologia de domínio. A correta construção do Modelo e Base de Conhecimento é de extrema importância para a obtenção de informações confiáveis e completas conforme a cultura e modo de escrita da organização. Portanto, o papel do engenheiro do conhecimento é fundamental para o sucesso das análises na arquitetura.

Quanto ao desempenho, a arquitetura apresenta um tempo de resposta variante conforme múltiplos critérios, tais como, computacionais (velocidade e quantidade de *CPUs*, *memória*, *etc.*); complexidade das análises (perguntas com muitos relacionamentos, uso de funções ou conclusões de regras de inferência e combinação destes); ambigüidades (quantidade de interações com o usuário); aplicação de inferências e volume da base de conhecimento (afeta o tempo de processamento de inferência conforme a abordagem adotada); e ainda, o tempo de processamento da consulta no data warehouse. Como todas as perguntas foram aplicadas de forma dedicada na mesma máquina, os critérios computacionais são omitidos. Observou-se que o tempo somado das tarefas desempenhadas pelos módulos Analisador Lingüístico, Reformulador, Motor de Busca por Similaridade e Tradutor OLAP resultou abaixo de um décimo de segundo (0,1s) em média. Como era de se esperar, essas tarefas tiveram maior tempo de resposta quando perguntas com muitos termos, relacionamentos, função e reformulação por regras de inferência foram usadas em conjunto.

O tempo de resposta para as perguntas ambíguas é praticamente desprezível com relação as não ambíguas. Como explicado durante o

trabalho, o Motor de Busca por Similaridade em alguns casos específicos pode submeter duas ou mais buscas para recuperar o caminho da ontologia. Em média o tempo de uma busca na matriz de caminhos construída neste trabalho implicou em menos um centésimo de segundo (0,01s). O maior tempo da etapa de interpretação da pergunta está na classificação feita pelo Analisador Lingüístico, em que neste trabalho é realizado por meio de buscas em um amplo dicionário de sinônimos para cada termo da pergunta. O tempo médio do Analisador Lingüístico para classificar um único termo da pergunta é também inferior a um centésimo de segundo (0,01s).

Neste trabalho, definiram-se três regras de inferência que podem ser analisadas no Apêndice A. O tempo gasto para processar qualquer uma dessas três regras individualmente sobre toda a base de conhecimento é praticamente igual e aproximou-se de 2,8 segundos. Dependendo da abordagem utilizada para a realização de inferências, o desempenho das análises é afetado. Sell (et. al., 2008) comenta em seu trabalho que a abordagem *on-the-fly* necessita do processamento de inferências e reescritas de consultas. Dependendo do tamanho do data warehouse, da complexidades das consultas e da concorrência de acesso, essa abordagem pode apresentar perda de desempenho. Além disso, tecnicamente alguns bancos de dados possuem limitações quanto à quantidade de valores possíveis na cláusula *IN*, o que pode também prejudicar a criação de filtros e o desempenho das consultas.

Já na abordagem *in-batch*, há uma melhora de desempenho uma vez que as conclusões do processamento de inferência estão prontas para serem consultadas no DW. Deste modo, o tempo extra de 2,8 segundos obtido no estudo de caso com a abordagem *on-the-fly* seria desconsiderado na análise. Dado um aumento do número de usuários da arquitetura, esse tempo de resposta pode aumentar e inclusive inviabilizar as análises. Com isso, a abordagem *in-batch* provê uma maior escalabilidade, pois as bases de dados e as plataformas de DW são desenvolvidas para garantir o acesso uniforme mesmo com múltiplas consultas e o aumento do número de usuários. Por outro lado, o volume de inferências mantido no modelo tripla pode aumentar ao longo do tempo e afetar o desempenho das análises. Para esses casos, técnicas de *tuning* e indexação de bases de dados devem ser aplicadas para diminuir o tempo de resposta das consultas.

4.7 DISCUSSÃO SOBRE OS TRABALHOS RELACIONADOS

As tarefas e os módulos funcionais da arquitetura são motivados em estudos das áreas de processamento de linguagem natural e BI. Boa parte dos pontos fortes e fracos desses estudos são herdados na arquitetura. Por conta disso, as características pontuais oriundas dos trabalhos relacionados podem ser discutidas individualmente e, no ponto de vista sistêmico, esta proposta pode ser comparada a outras arquiteturas de BI baseadas em conhecimento e em linguagem natural.

A etapa relacionada à interpretação de perguntas baseia-se principalmente nos frameworks propostos por Lopez (et. al., 2007) e Wang (et. al., 2007). As contribuições desses frameworks a esta pesquisa estão associadas basicamente ao uso de ontologias para auxiliar as tarefas de reformulação e a representação formal da pergunta a partir dos caminhos. Esses autores chegam a avaliar os benefícios do uso de ontologia quando aplicados também em fontes de dados estruturadas ou banco de dados, tal com neste trabalho. Eles concluem que a ontologia é uma alternativa portátil para a representação de conhecimento em qualquer domínio e ainda oferecem expressividade para a interpretação semântica das perguntas e obtenção de respostas nesses tipos de fontes. Kaufmann e Bernstein (2007) complementam a idéia acima afirmando que a aplicação de ontologias e métodos de QA em bases de dados estruturadas provêem formas mais intuitivas onde os usuários finais dão preferência em relação a buscas por palavras-chave, interfaces gráficas ou baseadas em menus.

Em distinção, os frameworks de Lopez (et. al., 2007) e Wang (et. al., 2007) tentam criar uma representação intermediária da pergunta antes de obter informações contextuais da ontologia. Essa representação é formada por triplas, tal como no modelo RDF, que são geradas somente por regras sintáticas e gramaticais do idioma. Essas triplas são usadas como entrada para um módulo semelhante ao Motor de Busca por Similaridade, cuja intenção basicamente é localizar o melhor caminho da ontologia. Nesse processo, o usuário também participa dos processos de desambiguação de conceitos e relações até obter a formalização final da pergunta. Diferentemente deste trabalho, a versão atual desses frameworks não suporta perguntas com quantificações e mensurações de informações (exemplo, perguntas iniciadas por *Quantos*, *Quantas*, etc.), por outro lado, viabiliza outros tipos de questões (exemplo, *Quais*, *Quem*, *Onde*, etc.). Embora o objetivo seja a obtenção de um cubo OLAP, na qual o uso das *stop-words* *Quantas*,

Quantos, etc. são mais adequados, a arquitetura proposta não limita quanto ao uso de outros tipos de questões factuais (por exemplo, Quem, Onde, Qual, Quando, etc.). As stop-words usadas para a identificação desse tipo de pergunta podem ser associadas aos padrões e heurísticas para o reconhecimento de medidas, agrupamentos ou filtros conforme desejado.

Os trabalhos destinados a transformar a pergunta em uma linguagem formal de consulta, como SPARQL (sobre a base de conhecimento e modelo da ontologia em OWL) ou SQL (sobre os bancos de dados relacionais), comumente utilizam padrões e heurísticas para identificação dos elementos da consulta (WANG, et. al., 2007; MENG; CHU, 1999). No entanto, a maioria desses trabalhos apresentam formas individualizadas de uso de stop-words para a avaliação desses elementos. Para facilitar a formalização do conjunto de padrões e heurísticas, este trabalho sugere uma classificação para o conjunto de stop-word. Essa classificação permite que os padrões idiomáticos ou heurísticas sejam determinados tanto para a categoria de stop-word quanto para uma stop-word específica.

A ambigüidade e os problemas de interpretação das stop-words de seleção usadas para a construção de filtros, especificamente as que determinam os operadores lógicos (*AND* e *OR*), são também comentados por Lopez (et. al., 2007) e Smart (2008). Esta proposta utiliza os tokens *E* e *OU* exatamente como seus respectivos operadores lógicos (*AND* e *OR*) quando associados aos critérios de filtros. Isto porque, mesmo que o *E* lingüístico seja confundido com os operadores lógicos (*AND* ou *OR*), permite que ambos os operadores sejam utilizados na pergunta. Dessa forma, usuário pode levar em conta a semântica que deseja atribuir para um determinado critério de filtro.

Com base nas pesquisas de Sell (2006; et. al., 2008) e Silva (2006), todos os elementos do data warehouse devem ser anotados e mapeados à ontologia de domínio por meio da Ontologia BI. Essa anotação semântica de fontes de dados é vista também nas abordagens de QA baseadas na Web Semântica, onde também sentenças textuais são extraídas a partir de repositórios de documentos anotados (LOPEZ, et. al., 2007; THAI, et. al., 2006). Esses tipos de anotações requerem que os esquemas ou estruturas das fontes de dados sejam configurados e relacionados de acordo com os conceitos do domínio. Dado um grande número de dimensões e tabelas de fato no data warehouse, esta atividade pode ser custosa, de modo que meios automáticos ou semi-automáticos podem ser desenvolvidos. Não é escopo deste trabalho tratar a engenharia de ontologias e a manutenção de bases de conhecimento, e

por isso, outras iniciativas, como as propostas de Ceci (et. al., 2010) e Ghisi (2008) podem ser aplicadas em colaboração à arquitetura.

Conceitualmente, alguns trabalhos de QA utilizam o termo *inferência* ou *raciocínio* com relação à tarefa de extrair a sentença textual exata a partir de documentos segundo a sintaxe da pergunta (KAUFMANN; BERNSTEIN, 2007; DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2010). Neste trabalho, trata-se como inferência a tarefa de derivar ou explicitar novas informações (relacionamentos ou conceitos do domínio) a partir da aplicação das regras de inferência sobre a base de conhecimento. Essas informações são úteis para que a geração de novas análises. Isto é, as conclusões semânticas procedentes do processo de inferência são compartilhadas em uma visão sumarizada juntamente com as informações estratégicas do data warehouse. Para tal, esta proposta busca inspiração na arquitetura *SBI* (SELL, 2006; et. al.; 2008) e nas abordagens de processamento de inferências *on-the-fly* e *in-batch*.

O uso de funções e cálculos para a criação de consultas, embora seja comum nas soluções de BI, é pouco explorado na literatura relacionada aos sistemas de QA e interfaces de linguagem natural para banco de dados. Funções e cálculos são vistos nas linguagens de consulta (SQL, SPARQL, dentre outras) e também na sintaxe das regras dos mecanismos de inferência (MCBRIDE, 2002). Neste trabalho, para facilitar a vinculação a determinados termos usados pelo tomador de decisão, todas as funções e cálculos são especificados conforme uma sintaxe XML própria. Assim, uma das contribuições deste trabalho é permitir o uso de funções nas análises a partir de terminologias específicas informadas na pergunta. Conforme mostrado, os resultados das funções ou cálculos são usados como valores de filtros para as operações de *slice and dice*.

Quanto à aplicação de linguagem natural em plataformas de apoio à decisão, verifica-se que as tendências da área de BI já podem ser percebidas na prática. Dentre as soluções de mercado disponíveis, destacam-se *Semantra* e *EasyAsk*. *Semantra* possui uma arquitetura dividida semelhante as três etapas utilizadas neste trabalho: 1) um repositório hierárquico de conceitos, ontologias e regras de negócio denominado *OntoloNet* que fornece o contexto da pergunta e o mapeamento para as fontes de dados; 2) um interpretador semântico para a análise lingüística da pergunta; 3) um gerador de consultas SQL responsável por retornar as informações do data warehouse. Já a solução *EasyAsk* se baseia em dicionários de sinônimos e thesaurus para analisar as perguntas, resolver erros de escrita e considerar o contexto de

aplicação. Com relação a esta proposta, as duas soluções permitem ainda a visualização de gráficos, e relatórios por meio de perguntas ou palavras-chave, além do retorno de cubos OLAP. *EasyAsk*, em particular, torna possível a integração de fontes heterogêneas e retorno de documentos em um mesmo ambiente analítico. Ambas as soluções possuem mecanismos de desambiguação e recomendação de perguntas baseadas no contexto conforme os termos de entrada. Esses projetos comerciais demonstram que a unificação de ambientes de busca ou linguagem natural com plataformas de BI é viável na prática e oferece um modo rápido e intuitivo de efetuar análises.

5 CONCLUSÃO

A arquitetura proposta neste trabalho integra os estudos baseados em linguagem natural e as pesquisas de *Business Intelligence* no estado da arte em uma única abordagem para obtenção de conhecimento a partir das fontes de dados estruturadas. Esta arquitetura pauta-se na interpretação semântica de perguntas informadas livremente pelo tomador de decisão para conduzir a realização de análises sobre o *data warehouse*. Com isso, essa proposta oferece um método para que as soluções de BI apropriem a linguagem e o modo de escrita habitual dos gestores da organização para guiar a estratificação de informações no processo decisório.

Verifica-se que algumas iniciativas de BI, inspiradas na Web Semântica, introduzem o uso de tecnologias semânticas e métodos baseados em conhecimento para a exploração dos repositórios da organização. Essas pesquisas provêm novas funcionalidades analíticas e ainda a capacidade de raciocínio para apoiar à tomada de decisão. Como contribuição também para as próximas gerações de soluções de BI, este trabalho baseia-se nessas iniciativas e tecnologias semânticas para aproximar a área de BI aos meios mais naturais e expressivos de consultas oriundos da disciplina de *Question Answering*. O uso de linguagem natural, similar ao modo familiarizado pelos milhares de usuários dos sistemas de busca na Web, já se revela como uma tendência para a área de BI. No entanto, em vez de palavras-chave, a necessidade de informação neste trabalho é expressa por meio de perguntas livres conforme o contexto do domínio da organização.

Os módulos funcionais, construídos com a linguagem Java, demonstram a viabilidade da arquitetura em um estudo de caso relacionado ao âmbito de C&T da gestão curricular da Plataforma Lattes. Dentro desse cenário, as etapas de interpretação de perguntas e obtenção de hipercubos são demonstradas em análises que envolvem: a resolução de ambigüidades; aplicação de inferências; reformulação da pergunta baseada em sinônimos e hierarquias de classes; uso de funções e cálculos com base nas terminologias específicas e; identificação da semântica da pergunta com base no modelo da ontologia. Essas etapas são auxiliadas por meio de tecnologias semânticas e recursos de representação de conhecimento, como ontologias, mecanismos e regras de inferência, funções, padrões idiomáticos e heurísticas motivados pelos trabalhos relacionados.

Um protótipo foi construído para ilustrar como alguns exemplos de perguntas em linguagem natural possibilitam auxiliar à tomada de decisão nesse contexto de C&T. Sem a necessidade de treinamento a priori, esse protótipo torna possível ao usuário da arquitetura obter as informações do data warehouse de maneira única. Deste modo, o custo e ainda a curva de aprendizado conforme os distintos métodos de consulta e manuseio das ferramentas analíticas podem ser reduzidos. Visto que a ferramenta analítica é um elemento externo à arquitetura, o protótipo em si apenas esboça uma maneira simples de aplicar as tarefas e os módulos da arquitetura. Portanto, outras formas de interação com o usuário, tais como na resolução de ambigüidades ou na visualização de informações, devem ser projetadas.

Com base nos trabalhos relacionados, a arquitetura prevê o uso de inferências para derivar novas informações a partir dos dados e conjugá-los nas análises. As inferências podem ser realizadas consoante duas abordagens delineadas ao longo do trabalho: *on-the-fly* ou *in-batch*. Em comparação no contexto de C&T aplicado, a abordagem *on-the-fly* apresenta um tempo de resposta maior. Ressalta-se que o desempenho oscila de acordo com inúmeros fatores, como a quantidade de instâncias mantidas na base de conhecimento, configuração da máquina para processamento das inferências, indexação e otimização das bases de dados, etc. Na abordagem *on-the-fly*, o tempo de processamento da inferência soma-se ao tempo de execução da consulta sobre o DW. No entanto, esse tipo de inferência não necessita adaptar o modelo dimensional e manter um processo de atualização e armazenamento das conclusões de inferências no DW. Já na abordagem *in-batch*, como todos os resultados das inferências são persistidos no modelo tripla anteriormente ao processo decisório, o desempenho envolve apenas o tempo de resposta da consulta sobre o DW. Essa última abordagem traz a vantagem também de poder usar as inferências com maior escalabilidade. Como desvantagem, a abordagem *in-batch* exige a presença de sistemas de atualização de inferências, similares aos sistemas ETL, e a gerência do volume de derivações semânticas armazenadas ao longo do tempo.

A arquitetura atua em um contexto limitado pelo modelo da ontologia de domínio. A interpretação das perguntas leva em conta somente os conceitos e terminologias do negócio da organização representados por esse modelo. Neste caso, a ontologia de domínio proporciona o universo possível de soluções para a interpretação das perguntas. Cada pergunta deve abranger o escopo delimitado pelas classes, propriedades e relacionamentos da ontologia. Assim, pressupõe-

se que todos os jargões, os principais atores do negócio e conceitos do domínio estejam modelados para a correta elaboração de perguntas. Cabe ao engenheiro de conhecimento juntamente com o analista de negócio da organização, propor formas de manutenção e evolução das bases de conhecimento, dado que isso não faz parte do escopo desta proposta.

Uma vez realizada a interpretação semântica da pergunta, a arquitetura viabiliza a construção de consultas multidimensionais com base em elementos construtores das operações OLAP, identificados na análise de pesquisas relacionadas. Tais elementos, como medidas, agrupamentos, filtros e junções entre tabelas e dimensões do *data warehouse* são descobertos pelos módulos da arquitetura a partir de heurísticas e padrões da linguagem armazenados no modelo e base de conhecimento.

No tocante à demonstração da viabilidade, este estudo assinala algumas perguntas que cobrem os principais cenários de interpretação previstos. Toda a interpretação da pergunta obedece à especificação formal do conjunto de ontologias e padrões idiomáticos configurados na arquitetura. Do mesmo modo que a base de conhecimento, os padrões idiomáticos, as heurísticas e os demais itens necessários para a formalização e análise semântica das perguntas devem sofrer manutenções ao longo do tempo. Esses elementos citados foram iterativamente adequados e evoluídos durante a fase de validação dos módulos da arquitetura até que todas as perguntas, além das ilustradas neste trabalho, obtivessem respostas exatas. Dado que o modelo dimensional do estudo de caso é conhecido e possui relativamente poucas tabelas, as perguntas puderam ser validadas uma a uma por análise da consulta SQL gerada e conferência das informações obtidas. Mesmo com melhoria e refinamento iterativo dos módulos funcionais, ainda sim, o tomador de decisão pode interagir com a ferramenta nos casos de ambigüidades de entidades e caminhos.

5.1 LIMITAÇÕES E TRABALHOS FUTUROS

Dada a complexidade do problema, algumas questões da Engenharia do Conhecimento sobretudo aquelas associadas à disciplina de processamento de linguagem natural e *QA* são tangenciadas nessa proposta. Essas questões não são aprofundadas pela arquitetura e por isso consistem em pontos a serem estudados em trabalhos futuros.

Outras constituem em linhas de pesquisa correlatas que podem ser usadas em conjunto para melhorar e estender a proposta deste trabalho.

Dentre as limitações deste trabalho, o estudo da língua expõe diversas problemáticas e variações que é evidente a dificuldade em interpretar e formalizar a escrita e a comunicação humana em sistemas de conhecimento. Somado a isso, frases sintaticamente ou semanticamente mal redigidas, formas coloquiais, desconstruções gramaticais e perguntas que podem ser feitas de inúmeras maneiras consistem também em óbices para quaisquer pesquisas dessa área. Com isso, este trabalho herda os problemas e imperfeições das disciplinas e pesquisas relacionadas. Não é foco deste trabalho propor inovações ou melhorias no campo de linguagem natural, e sim aproveitar os progressos dessa área para aplicá-los em conjunto com os instrumentos de apoio à tomada de decisão.

Em particular ao uso de expressões de negação, a linguagem natural possibilita que uma diversidade de formas contraditórias e tipos de negação sejam combinados em perguntas. A concepção de sistemas de conhecimento para a compreensão ainda que semi-automática dessas formas, está além do escopo dessa dissertação. O problema encontra-se nas variadas interpretações para as expressões de negação obtidas a partir de uma única pergunta. Neste trabalho, as possibilidades de interpretações normalmente trazem ambigüidades em decorrência da grande quantidade de caminhos candidatos oriundos da ontologia. A problemática pode ser melhor entendida no trabalho de Gavriel (2005), que trata unicamente do uso de expressões de negação em sistemas de QA e de extração de informação. Para não inviabilizar o uso das expressões de negação, uma alternativa simples é configurar as stop-words juntamente com os padrões sintáticos e heurísticas. Tal como os operadores relacionais (<, >, =, >=, <=, etc.) o uso do operador que simboliza a não igualdade (por exemplo, *NOT IN*, != ou <>) pode ser igualmente usado para a construção de filtros na consulta.

Este trabalho utiliza *data warehouses* construídos em bases de dados relacionais para criar consultas multidimensionais e obter o cubo OLAP. No entanto, os processos de gestão e inteligência competitiva requerem a obtenção de conhecimento também em fontes não estruturadas. As novas gerações de *data warehouse* e também muitas soluções de mercado já permitem a integração de fontes heterogêneas. Dessa forma, como trabalho futuro, a exploração de outros tipos de fontes de dados se faz necessária. A arquitetura parte do princípio que os dados das fontes heterogêneas já estejam integrados ao *data warehouse*. Contudo, o uso do cubo OLAP ou o formato tabular como resposta deve

ser revisto dada a complexidade de sumarizar e quantificar informações a partir de bases de dados textuais.

Como evolução desta pesquisa, outros formatos e alternativas ao tipo de resposta podem ser retornados em substituição ou complemento ao cubo OLAP. Observa-se na literatura que algumas soluções empregam técnicas de RI e EI para criar instâncias do modelo da ontologia de domínio a partir de documentos e baseiam-se nesse modelo para produzir sentenças textuais como resposta. Igualmente, trabalhos futuros podem usufruir dessas práticas de descrição ou verbalização de ontologias e aplicá-las sobre bases de dados estruturadas em vez de documentos. Assim, a arquitetura proposta pode ser estendida para que as respostas formem um enredo ou sumário textual com as informações do data warehouse. Outrossim, as formas visuais comuns nos ambientes de BI, como gráficos, dashboards, charts, dentre outros, podem também ser geradas.

Para garantir que as análises estejam corretas e válidas, mecanismos de prova baseados na ontologia, tais como os já previstos na arquitetura da Web Semântica devem ser desenvolvidos. Além de validar as análises, isto permitiria ao engenheiro do conhecimento avaliar e melhorar o modelo de representação de conhecimento (ontologias, padrões e heurísticas, funções, regras de inferência) da arquitetura.

Nos casos de ambigüidade de entidades ou de caminhos da ontologia, é necessário que o tomador de decisão interaja com os módulos da arquitetura até a completa desambiguação. Para facilitar a resolução de ambigüidades ou ainda considerar as interações anteriores, um mecanismo de aprendizado pode ser incorporado à arquitetura. Esse mecanismo, na qual não é contemplado no escopo atual desta pesquisa, seria um novo módulo funcional que levaria em conta as perguntas anteriormente interpretadas e resultados obtidos. Como extensão à arquitetura, esse mecanismo de aprendizado seria importante também para a manutenção da base de conhecimento e identificação de novos conceitos que ainda não estão contemplados no modelo.

REFERÊNCIAS

ANTONIOU, G.; HARMELEN, F. **A Semantic Web Primer**. 2. ed. Inglaterra: The MIT Press, 2008.

BAYEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. 2. ed. United Kingdom: ACM Press Books, 1999.

BENEVENTANO, D. BERGAMASCHI, S. GUERRA, F. VINCINI, M. **The SEWASIE MAS for semantic search**. University of Modena and Reggio Emilia, Itália: IEEE, Digital Information Management, 2007.

BEPPLER, F. **Um Modelo para Recuperação e Busca de Informação Baseado em Ontologia e no Círculo Hermenêutico**. 2008. Tese (Tese em Engenharia e Gestão do Conhecimento) – Universidade Federal de Santa Catarina, Florianópolis, 2008.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The Semantic Web**. Scientific American, 2001, p. 29-37.

BILOTTI, M. W. **Query Expansion Techniques for Question Answering**. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Estado Unidos, 2004.

BÖHRINGER, M.; GLUCHOWSKI, P.; KURZE, C.; SCHIEDER, C. **A Business Intelligence Perspective on the Future Internet**. Proceedings of the Sixteenth Americas Conference on Information Systems, Lima, Peru, 2010.

BORST, W.N., **Construction of Engineering Ontologies for Knowledge Sharing and Reuse**. Tese (Phd Dissertation) - University of Twente, Holanda, 1997.

BURGER, J.; CARDIE, C.; CHAUDHRI, V.; GAIZAUSKAS, R.; HARABAGIU, S.; ISRAEL, D.; JACQUEMIN, C.; LIN, C.; MAIORANO, S.; MILLER, G.; MOLDOVAN, D.; OGDEN, B.; PRAGER, J.; RILOFF, E.; SINGHAL, A.; SHRIHARI, R.;

STRZALKOWSKI, T.; VOORHEES, E.; WEISHEDEL, R. **Issues, Tasks and Program Structures to Roadmap Research in Question & Answering,** 2009. Disponível em: <www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc>. Acesso em Junho de 2010.

CECI, F.; SILVA, D.C.; GONÇALVEZ, A. L.; SELL, D. **Towards a Semi-Automatic Approach for Ontology Maintenance.** 7th CONTECSI International Conference on Information Systems and Technology Management. Universidade de São Paulo, São Paulo, 2010.

CHUNG, W.; CHEN, H.; NUNAMAKER J.F. **Business Intelligence Explorer: A Knowledge Map Framework for Discovering Business Intelligence on the Web.** Proceedings of the 36th Hawaii International Conference on System Sciences, 2002.

CIMIANO, P.; HAASE, P.; HEIZMANN, J.; MANTEL, M. **ORAKEL: A portable natural language interface to knowledge bases.** Technical report, Institute AIFB, University of Karlsruhe, Alemanha, 2007.

CNPq. **Plataforma Lattes.** Disponível em: <<http://lattes.cnpq.br>>. Acesso em 22 de janeiro de 2010.

CODY, W.F., KREULEN, J.T., KRISHNA, V., SPANGLER, W.S. **The integration of business intelligence and knowledge management.** IBM Systems Journal, 2002.

CONLON, S.J.; CONLON, J.R.; JAMES, T.L. **The economics of natural language interfaces: natural language processing technology as a scarce resource.** Holanda: Journal Decision Support Systems, vol. 38, 2004.

COMPUTER WEEKLY. **Business intelligence for the masses,** 2002. Disponível em <<http://www.computerweekly.com/Articles/2002/11/22/191182/microso-ft-business-intelligence-for-the-masses.htm>>. Acesso em 10 de Outubro de 2010.

DACONTA, M.; OBRST, L.; SMITH, K. **The Semantic Web-A Guide to the Future of XML, Web Services, and Knowledge Management.** Wiley, 2003. 312 p.

DAMLJANOVIC, D.; AGATONOVIC, M.; CUNNINGHAM, H. **Natural language interface to ontologies**: Combining syntactic analysis and ontology-based lookup through the user interaction. Extended Semantic Web Conference 2010. Grécia, 2010.

DAVIES, J.; FENSEL, D.; VAN HARMELEN, F. **Towards the semantic web**: ontology-driven knowledge management. 1. ed. Wiley. 2003.

ECKERSON, W. **A Marriage Made In Heaven**: Search and BI. Disponível em <<http://tdwi.org/Blogs/Wayne-Eckerson/2010/08/BI-Search.aspx>>. Acesso em 20 de Outubro de 2010.

ECKERSON, W. **Performance Dashboards**: measuring, monitoring and managing your business. John Wiley & Sons: 2006.

ECKERSON, W. **Smart companies in the 21st century**: The secrets of creating successful business intelligence solutions, 2003. Disponível em <http://download.101com.com/tdwi/research_report/2003BIReport_v7.pdf> Acesso em 30 de outubro de 2010.

EVELSON, B.; BROWN, M. **Search + BI**: Unified Information Access. Combining Unstructured And Structured Info Delivers Business Insight. Forrest Research, 2008.

FENSEL, D. **Ontologies: Silver Bullet for knowledge Management and Eletronic Commerce**. Springer-Verlag: Berlin, 2001.

GHISI, F. B. **Uma abordagem para manutenção de ontologias de uma plataforma de business intelligence semântico**. Trabalho de Conclusão de Curso (Graduação) – Universidade Federal de Santa Catarina, Florianópolis, 2008.

GAVRIEL, M. **Capturing Negation in Question Answering Systems**. Master of Science in Speech and Language Processing Theoretical and Applied Linguistics. University of Edinburgh, Escócia, 2005.

GRISHMAN, R. **Information extraction**: Techniques and challenges. International Summer School, New York University, 1997.

GRUBER, T. R. **A Translation Approach to Portable Ontology Specifications.** 1993. Disponível em <http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html>. Acesso em 19 de janeiro de 2011.

GUARINO, N. **Formal ontology and information systems.** Amsterdam, 1998.

HENSCHEN, D. **Natural Language Query:** Old Answer for 'New' BI Opportunity. Disponível em <<http://www.intelligententerprise.com/showArticle.jhtml?articleID=207000425>>. Acesso em 18 de outubro de 2008.

HIRSCHMAN, L.; GAIZAUSKAS, R.. **Natural language question answering:** the view from here. Natural Language Engineering, Cambridge University, United Kingdom, 2001.

HODGE, P. **Business intelligence Architecture.** Disponível em <<https://sites.google.com/a/paulhodge.com/www/architecture>>. Acesso em 22 de Janeiro de 2011.

HORROCKS, I.; PATEL-SCHNEIDER, P.F; BOLEY, H., TABEL, S.; GROSOFF, B.; DEAN, M. **SWRL:** A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. Disponível em: <<http://www.w3.org/Submission/SWRL>>. Acesso em 15 de novembro de 2010.

HOWSON, C. **Successful Business Intelligence:** Secrets to Making BI a Killer App, McGraw-Hill, New York, 2008. 244p.

INMON, W. H. **Building the Data Warehouse.** 4. ed. New York: John Wiley & Sons, 2005. 576p.

INMON, W.; STRAUSS, D.; NEUSHLOSS, G. **DW 2.0 The Architecture for the Next Generation of Data Warehousing.** 2007.

IMHOFF, C.; GALEMMO, N.; GEIGER, J. **Mastering Data Warehouse Design:** Relational and Dimensional Techniques. Wiley, 2003. 456p.

INTERNATIONAL DATA CORPORATION BRASIL. **IDC Brasil divulga panorama do mercado de Business Intelligence na América Latina.** Disponível em

<http://www.idclatin.com/news.asp?ctr=bra&year=2010&id_release=1738>. Acesso em 30 de outubro de 2010.

JENA. **Jena** – A Semantic Web Framework for Java. Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: 22 de janeiro. 2011.

KATZ, B.; LIN, J.; FELSHIN, S. **Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources.** MIT Artificial Intelligence Laboratory. In Proceeding of the ACL 2001 Workshop on Human Language Technology and Knowledge Managemet. França, 2001.

KAUFMANN, E.; BERNSTEIN, A. **How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?.** Proceeding in International Semantic Web Conference /Asia Semantic Web Conference, Korea, 2007.

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.** 2.ed. New York: John Wiley & Sons, Inc., 2002. 464 p.

KOWALSKI, G.J.; MAYBURY, M.T. **Information storage and retrieval systems: theory and implementation.** 2. ed. Estados Unidos: Springer, 2000. 336 p.

LAVBIC, D.; VASILECAS, O.; RUPNIK, R. **Ontology-based Multi-Agent System to support Business Users and Management.** Technological and economic development Of Economy. Baltic Journal on Sustainability. Vilnius: Technika, 2010.

LASSILA, O.; MCGUINNESS, D. **The Role of Frame-Based Representation on the Semantic Web.** Technical Report. Knowledge Systems Laboratory. Stanford University. Stanford, California, 2001.

LOPEZ, V.; UREN, V.; MOTTA, E.; PASIN, M. **AquaLog: An ontology-driven question answering system for organizational semantic intranets.** Web Semantics: Science, Services and Agents on the World Wide Web, 2007.

MANNING, C.D; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. United Kingdom: Cambridge University Press, 2008. 496 p.

MAYNARD, D.; BONTCHEVA, K.; CUNNINGHAM, H. **Towards a semantic extraction of Named Entities**. In Recent Advances in Natural Language Processing, Bulgaria, 2003.

MCBRIDE, B. **Jena: A Semantic Web Toolkit**. Internet Computing, IEEE, 2002.

MCGUINNESS, D. **Question Answering on the Semantic Web**. IEEE Intelligent Systems, Vol. 19, 2004.

MENG, F.; CHU, W. **Database query formation from natural language using semantic modeling and statistical keyword meaning disambiguation**. Technical Report CSD-TR 990003, University of California, 1999.

MOENS, M. **Information extraction: algorithms and prospects in a retrieval context**. 1a ed. Holanda: Springer, 2006. 246.p.

MOLDOVAN, D.; TATU, M.; CLARK, C. **Semantic Computing**. New York: John Wiley & Sons, 2009.

MOSS, L.; ATRE, S. **Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications**. Addison-Wesley Professional, 2003. 476 p.

NADEAU, D. **BALIE – Baseline Information Extraction**. Disponível em <<http://balie.sourceforge.net>>. Acesso em 22 de Janeiro de 2011.

NIRENBURG, S.; RASKIN, V. **Ontological Semantics**. Inglaterra: The MIT Press, 2004.

ORACLE. **Java Technology**. Disponível em <<http://www.oracle.com/technetwork/java/index.html>>. Acesso em 22 de janeiro de 2011.

PATEL-SCHNEIDER, P.F. **A Proposal for a SWRL Extension towards First-Order Logic**. W3C Member Submission 11 April 2005. Disponível em: <<http://www.w3.org/Submission/SWRL-FOL>>. Acesso em 20 de Novembro de 2010.

PÉREZ, G; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. **Ontological engineering**: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, 2004. 415 p.

PONNIAH, P. **Data Warehousing Fundamentals**: A Comprehensive Guide for IT Professionals. John Wiley & Sons Inc, 2001.

PRIEBE, T.; PERNUL, G. **Ontology-based Integration of OLAP and Information Retrieval**. Proceedings of the 14th International Workshop on Database and Expert Systems Applications. Prague, 2003.

QUARTERONI, S. **Advanced Techniques For Personalized, Interactive Question Answering**. The University of York, Department of Computer Science York. United Kingdom, 2007.

RAO, M. **Knowledge Management tools and techniques**: practitioners and experts evaluate KM solutions. 1a. ed. United Kingdom: Elsevier Butterworth-Heinemann, 2005. 456 p.

ROTHBERG, H. N.; ERICKSON, G. S. From **Knowledge to Intelligence**: creating competitive advantage in the next economy. United Kingdom: Elsevier Butterworth-Heinemann, 2005. 400 p.

SAGGION, H.; FUNK, A.; MAYNARD D.; BONTCHEVA, K. **Ontology-based Information Extraction for Business Intelligence**. Department of Computer Science, University of Sheffield, United Kingdom. 2007.

SCHREIBER, G.; AKKERMANS, H.; ANJEWIERDEN, A.; HOOG, R.; SHADBOLT, N.; DE VELDE, W. V.; AND WIELINGA, B. **Knowledge Engineering and Management**: the CommonKADS Methodology. MIT Press. Cambridge. Massachussets. 2002.

SELL, D. **Uma Arquitetura para Business Intelligence baseada em Tecnologias Semânticas para Suporte a Aplicações Analíticas**. Tese

(Tese em Engenharia de Produção e Sistemas) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

SELL, D., SILVA, D. C., BEPLER, F. D., NAPOLI, M., GHISI, F. B., PACHECO, R. C. S., TODESCO, J. L. **SBI: a semantic framework to support business intelligence**. In Proceedings of the first international workshop on Ontology-supported business intelligence. Karlsruhe, Alemanha, 2008.

SILVA, D. C. **Aplicação de Ontologias para o suporte ao processo ETL em soluções de Business Intelligence**. Monografia (Sistemas de Informação) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

SMALLTREE, H. **Business intelligence search: Five myths**. SearchBusinessAnalytics, 2006. Disponível em: <<http://searchbusinessanalytics.techtarget.com/news/1507286/Business-intelligence-search-Five-myths>>. Acesso em 17 agosto de 2010.

SMART, P. **Controlled Natural Languages and the Semantic Web**. School of Electronics and Computer Science University of Southampton. United Kingdom, 2008.

SMITH, K. **What is the ‘Knowledge Economy’? Knowledge Intensity and Distributed Knowledge Bases**. Discussion Paper Series, Institute for New Technologies, The United Nations University, 2002.

STANFORD. **The Protégé Ontology Editor and Knowledge Acquisition System**. Disponível em < <http://protege.stanford.edu/>>. Acesso em 22 de janeiro de 2011.

STUDER, R.; BENJAMINS, V.; FENSEL, D. **Knowledge Engineering: Principles and Method**. Data and Knowledge Engineering, 1998. Disponível em: <<http://citeseer.ist.psu.edu/article/studer98knowledge.html>>. Acesso em 14 de novembro de 2010.

SWOYER, S. **Pervasive Business Intelligence: Still a Vision, Not Reality**, 2010. Disponível em <<http://tdwi.org/Articles/2010/01/20/Pervasive-BI-Still-a-Vision-Not-Reality.aspx>> Acesso em 16 outubro de 2010.

THAI, V.; O'RIAIN, S.; DAVIS, B.; O'SULLIVAN, D. **Personalized Question Answering**: A Use Case for Business Analysis. Proceedings of the 1st International Workshop on Applications and Business Aspects of the Semantic Web, Georgia, 2006.

THOMSEN, E. **OLAP Solutions**: Building Multidimensional Information Systems. New York: John Wiley & Sons, Inc. 2nd, 2002.

WANG, C.; XIONG, M.; ZHOU, Q.; YU, Y. **PANTO**: A Portable Natural Language Interface to Ontologies, Proceedings of the 4th European conference on The Semantic Web: Research and Applications, Austria, 2007.

W3C. **OWL 2**: Web Ontology Language (OWL) Document Overview. W3C Recommendation 27 October 2009. Disponível em: <<http://www.w3.org/TR/owl2-overview>>. Acesso em 14 de novembro de 2010.

W3C. **Resource Description Framework (RDF)**: Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. Disponível em: <<http://www.w3.org/TR/rdf-concepts/>>. Acesso em 10 de outubro de 2009.

W3C. **Semantic Web**, 2010. Disponível em: <<http://www.w3.org/standards/semanticweb>>. Acesso em 13 de novembro de 2010.

W3C. **SPARQL Query Language for RDF**. W3C Recommendation 15 January 2008. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query>>. Acesso em 10 de novembro de 2010.

ZENG, L.; XU, L.; SHI, Z.; WANG, M.; WU, W. **Techniques, Process, and Enterprise Solutions of Business Intelligence**. IEEE International Conference on Systems, Man, and Cybernetics, Taiwan, 2006.

APÊNDICE A – Especificação das regras de inferência

[CALOURO:

```
(?aluno rdf:type Pessoa)
(?aluno temFormacao ?formacao)
(?formacao rdf:type Formacao)
(?formacao anoInicio currentYear(?ano) )
(?aluno estudaEm ?instituicao)
(?instituicao rdf:type Instituicao)
→
(?aluno calouro ?instituicao)
```

]

[FORMANDO:

```
(?pessoa rdf:type Pessoa)
(?pessoa temFormacao ?formacao)
(?formacao rdf:type Formacao)
(?formacao temInstituicao ?instituicao)
(?instituicao rdf:type Instituicao)
(?formacao anoTermino currentYear(?ano) )
→
(?pessoa formando ?instituicao)
```

]

[EGRESSO:

```
(?pessoa rdf:type Pessoa)
(?pessoa temFormacao ?formacao)
(?formacao rdf:type Formacao)
(?formacao temInstituicao ?instituicao)
(?instituicao rdf:type Instituicao)
(?fomacao anoTermino ?anoTermino)
lessThan(?anoTermino, currentYear(?ano))
→
(?pessoa egresso ?instituicao)
```

]

APÊNDICE B – Especificação das funções

```
<?xml version="1.0" encoding="UTF-8"?>
<functions>
  <function>
    <input>
      <term> hoje </term>
      <term> atualmente </term>
      <term> agora </term>
      <term> ano atual </term>
    </input>
    <output> ${current.year} </output>
    <concept> lattes:ano </concept>
  </function>
  <function>
    <input>
      <term> ano passado </term>
      <term> anteriormente </term>
      <term> ano anterior </term>
    </input>
    <output> ${current.year} - 1 </output>
    <concept> lattes:ano </concept>
  </function>
</functions>
```