

**Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção**

Denilson Sell

**UMA ARQUITETURA PARA BUSINESS INTELLIGENCE
BASEADA EM TECNOLOGIAS SEMÂNTICAS PARA
SUPORTE A APLICAÇÕES ANALÍTICAS**

Florianópolis

2006

Denilson Sell

**UMA ARQUITETURA PARA BUSINESS INTELLIGENCE
BASEADA EM TECNOLOGIAS SEMÂNTICAS PARA
SUPORTE A APLICAÇÕES ANALÍTICAS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina para obtenção do título de Doutor em Engenharia de Produção

Orientador: Prof. Roberto Carlos dos Santos Pacheco, Dr.

Florianópolis

2006

FICHA CATALOGRÁFICA

S467u

Sell, Denilson

Uma arquitetura para business intelligence baseada em tecnologias semânticas para suporte a aplicações analíticas. / Denilson Sell.

210 f. : il

Tese (doutorado) - Universidade Federal de Santa Catarina. Programa de Pós-graduação em Engenharia de Produção. Florianópolis, 2006.

AGRADECIMENTOS

Desejo manifestar minha enorme gratidão às pessoas que contribuíram direta e indiretamente para a realização deste trabalho.

Primeiramente a Deus, que sempre me deu forças e iluminou o meu caminho. Por ter me acompanhado, por ter me dado a vida, saúde, bons amigos, uma ótima família e uma esposa maravilhosa.

Ao meu mentor, professor Roberto Pacheco, que contribuiu muito para o desenvolvimento do trabalho e para a minha formação como pessoa e pesquisador.

Aos grandes amigos, do Instituto Stela, que acompanharam e contribuíram para o meu trabalho, em especial ao pessoal das células de documentação e de business intelligence.

À minha família, que mesmo acompanhando a minha luta de longe, foi fundamental para o meu triunfo.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento parcial do presente estudo.

Ao Knowledge Media Institute (KMi) e The Open University pela acolhida, em especial, ao professor Enrico Motta, pelo grande apoio e incentivo.

E, em especial, a quem dedico este trabalho, minha esposa Graciele Sell, pela compreensão, carinho, apoio e incentivo.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Apresentação	1
1.2	Caracterização do problema	5
1.2.1	Suporte para representação e utilização da semântica do negócio no apoio ao processamento analítico	6
1.2.2	Extensão de Funcionalidades Exploratórias e Recomendação de Funcionalidades durante o Processo Decisório	9
1.3	Pressuposto da Pesquisa	13
1.4	Objetivos	13
1.4.1	Objetivo Geral	13
1.4.2	Objetivos Específicos	14
1.5	Justificativa.....	15
1.6	Delimitação do Escopo.....	17
1.7	Metodologia	18
1.8	Estrutura do Trabalho.....	20
2	BUSINESS INTELLIGENCE.....	22
2.1	Introdução.....	22
2.2	Sistemas de Informação.....	22
2.2.1	Classificação de Sistemas de Informação	23
2.3	Soluções de Business Intelligence	25
2.3.1	Definição	25
2.3.2	Componentes de uma Arquitetura Típica de BI	27
2.4	Data Warehouse	28
2.4.1	Definições, objetivos e características	28
2.4.2	Modelagem de Dados em um Data Warehouse	29
2.5	Área de Apresentação.....	31
2.6	OLAP	32
2.6.1	Definição	32
2.6.2	Características do processamento OLAP	34
2.6.3	Funcionalidades exploratórias	36
2.6.4	Metadados para o suporte de funcionalidades analíticas	38
2.7	A Metodologia BUS para Projetos de BI	40
2.8	Limitações das Soluções Correntes de BI	42
2.9	Considerações sobre o Capítulo.....	43

3	TECNOLOGIAS SEMÂNTICAS	46
3.1	Introdução	46
3.2	Web Semântica	46
3.2.1	Definição	46
3.2.2	Camadas da Web Semântica	47
3.3	Ontologia	49
3.3.1	Definição	49
3.3.2	Níveis de Representação	50
3.3.3	Tipos de Ontologias	52
3.3.4	Formalismos para Representação de Ontologias	53
3.3.5	OCML	53
3.4	Web services Semânticos	56
3.4.1	Definição e Motivações	56
3.4.2	Frameworks para Web services Semânticos	57
3.4.3	IRS-III	58
3.5	Aplicação de Tecnologias Semânticas no Domínio de BI	60
3.6	Aplicação de Tecnologias Semânticas para Suporte de Aplicações Analíticas	61
3.7	Considerações sobre o Capítulo	64
4	ARQUITETURA SEMANTIC BUSINESS INTELLIGENCE (SBI)	66
4.1	Introdução	66
4.2	Revisão dos Requisitos Funcionais para Arquiteturas de BI	66
4.3	Identificação de Requisitos não Funcionais para Arquiteturas de BI	71
4.4	Visão Geral da Arquitetura SBI	72
4.5	A Camada Fontes de Dados	73
4.6	A Camada Repositórios de Ontologias	74
4.7	A Camada Mecanismos de Inferência	75
4.8	A Camada Módulos Funcionais	76
4.9	A Camada Aplicações Clientes	77
4.10	O Processo de Desenvolvimento da Arquitetura SBI	77
4.11	Considerações sobre o Capítulo	80
5	A EPISTEMOLOGIA DA ARQUITETURA	82
5.1	Introdução	82
5.2	Ontologia do Domínio	82
5.3	Ontologia BI	86
5.3.1	Mapeando dados do Data Warehouse aos conceitos do negócio	87

5.3.2	Realizando cortes semânticos nas análises.....	94
5.3.3	Personalização da nomenclatura para a apresentação das informações	95
5.3.4	Descrevendo semanticamente as análises criadas pelos usuários.....	97
5.3.5	Descrevendo temas de análise	101
5.4	Ontologia de Serviços.....	104
5.4.1	Descrevendo <i>Web services</i> na Ontologia de Serviços	106
5.4.2	Suporte para Composição de Serviços.....	109
5.5	Considerações sobre o Capítulo.....	118
6	OS MÓDULOS FUNCIONAIS DA ARQUITETURA SBI	120
6.1	Introdução.....	120
6.2	Gerenciador de Ontologias.....	120
6.3	Gerenciador de Serviços.....	125
6.3.1	Classes para localização e execução de WSS.....	127
6.3.2	Classes para montagem e execução de composições.....	130
6.4	Gerenciador de Análises.....	135
6.4.1	Suporte para a definição de análises	137
6.4.2	Realizando cortes semânticos e reescrevendo análises.....	140
6.4.3	Recomendação de análises.....	146
6.5	Considerações sobre o Capítulo.....	147
7	DEMONSTRAÇÃO DE VIABILIDADE E ANÁLISE COMPARATIVA....	150
7.1	Apresentação	150
7.2	A ferramenta OntoDSS	150
7.2.1	Cenário de aplicação	150
7.2.2	Os módulos da ferramenta OntoDSS.....	153
7.2.3	Módulo de definição de análises	154
7.2.4	Módulo de assistência à análise	157
7.2.5	Módulo de composição de <i>Web services</i> semânticos	162
7.3	Análise sobre a flexibilidade da Arquitetura SBI	166
7.4	Análise sobre o desempenho das inferências no protótipo utilizando o mecanismo de inferência OCML.....	172
7.5	Comparação entre a arquitetura SBI, iniciativas acadêmicas relacionadas e soluções comerciais.....	177
7.6	Considerações sobre o Capítulo.....	186
8	CONSIDERAÇÕES FINAIS.....	189
8.1	Trabalhos futuros.....	192
	REFERÊNCIAS BIBLIOGRÁFICAS.....	195

Lista de Ilustrações

Figura 1 - Representação da metodologia utilizada no trabalho	20
Figura 2 - Níveis do Conhecimento	23
Figura 3 – Arquitetura típica de uma solução de Business Intelligence	28
Figura 4 - Exemplo de modelo-estrela	30
Figura 5 - Exemplo de estrutura pai-filho como metadado para a definição de hierarquias em uma dimensão de produtos	40
Figura 6 - A Metodologia BUS.....	41
Figura 7 - Estruturação das camadas da <i>Web</i> semântica.....	47
Figura 8 - Arquitetura do IRS-III	59
Figura 9 - Arquitetura SEWASIE	62
Figura 10 - Arquitetura Proposta por Priebe e Pernul (2003)	63
Figura 11 - Ilustração dos Módulos da Arquitetura Proposta	73
Figura 12 – Adição das etapas de desenvolvimento específicas da Arquitetura SBI a Metodologia BUS.....	80
Figura 13 - Ilustração de conceitos associados à gestão de C&T representados na Ontologia do Domínio.....	83
Figura 14 - Visão geral dos principais elementos reunidos na Ontologia BI	87
Figura 15 – Identificação dos elementos de uma análise.....	97
Figura 16 - Etapas da Metodologia do Instituto Stela para Projeto da Área de Apresentação de Soluções de BI.	102
Figura 17 - Apresentação das Unidades de Análise, Filtros e de Conteúdo na Ferramenta Plano Tabular.....	103
Figura 18 - Hierarquia de componentes no modelo de composição de WSS da Arquitetura SBI.....	110
Figura 19 - Exemplo de composição de serviços para uma análise de indicadores de instituições	113
Figura 20 - Ilustração das classes que compõem o módulo Gerenciador de Ontologias	124
Figura 21 - Ilustração do processo de comparação de conceitos	125
Figura 22 - Ilustração da interface principal do Gerenciador de Serviços, sua implementação para o framework IRS-III e classes de suporte	127
Figura 23 - Ilustração das classes utilizadas para a definição de uma composição	131
Figura 24 - Ilustração das classes utilizadas para a definição de <i>Control Components</i> , <i>Service Components</i> e <i>Mediators</i> em uma composição	134
Figura 25 - Fluxo de processamento em um <i>Service Component</i>	135

Figura 26 - Ilustração das principais classes que compõem o Módulo Gerenciador de Análises	137
Figura 27 - Classes de apoio para a definição de análises e filtros.....	139
Figura 28 - Processo de geração de uma consulta a partir das definições de uma análise	140
Figura 29 - Descrição do processo de redefinição de consultas	144
Figura 30 - Ilustração de conceitos associados à gestão de C&T e representados na Ontologia do Domínio	153
Figura 31 - Interface para definição de análises no OntoDSS	155
Figura 32 - Recuperação dos temas de análises definidos na Ontologia BI ..	156
Figura 33 - Ilustração do resultado de uma análise no OntoDSS e do menu de recomendações	158
Figura 34 - Listagem dos filtros semânticos relacionados ao conceito <i>institution</i> e apresentação parcial do resultado de uma inferência	161
Figura 35 - Listagem dos serviços relacionados ao conceito <i>institution</i> e apresentação do resultado da execução do serviço <i>list-groups-institution</i>	162
Figura 36 - Ilustração da funcionalidade de busca de WSS	162
Figura 37 - Ilustração de uma composição definida com o módulo de composição de <i>Web Services</i> semânticos	164
Figura 38 - Ilustração das tabelas utilizadas para a identificação das relações entre instituições antes e após a modificação nos requisitos	169
Figura 39 - Ilustração da alteração no modelo de dados por conta da nova regra de classificação de docentes	171
Figura 40 - Tempo para carga das instâncias	174
Figura 41 - Tempo para processamento da relação <i>institution-competitor-city</i>	175
Figura 42 - Tempo para processamento da relação <i>institution-competitor</i>	176
Figura 43 - Tempo para processamento da relação <i>institution-enrolled-teacher</i>	176

Lista de Quadros

Quadro 1 - Características dos Sistemas de Informação	24
Quadro 2 - Sumário das características de utilização de soluções de BI	26
Quadro 3 - Características de fato, dimensão e medida de um DW	31
Quadro 4 - Características Básicas	35
Quadro 5 - Características de Relato	35
Quadro 6 - Características Dimensionais	36
Quadro 7 - Funcionalidades OLAP.....	36
Quadro 8 - Sumário das Etapas e Papéis Associados na Metodologia BUS ...	41
Quadro 9 - Ontologias que definem a semântica operacional para a linguagem OCML.....	54
Quadro 10 - Sumário das Novas Etapas de Desenvolvimento Introduzidas pela Arquitetura SBI e dos Papéis Associados às Etapas	78
Quadro 11 - Descrição das classes utilizadas para suportar o mapeamento de fontes de dados e conceitos.....	90
Quadro 12 - Descrição das classes utilizadas para suportar o controle de acesso dos usuários e a personalização da nomenclatura	95
Quadro 13 - Descrição das classes utilizadas para representar as análises definidas pelos usuários	98
Quadro 14 - Descrição da classe <i>Analysis_Unit</i>	104
Quadro 15 - Descrição das classes utilizadas para representar temas de análise	108
Quadro 16 - Visão geral das características das ontologias da Arquitetura vis-à-vis requisitos funcionais e não funcionais	119
Quadro 17 - Formatação de filtros para a busca de WSS na Ontologia de Serviços através do método <i>FindGoalbyFilter</i>	128
Quadro 18 - Formatação de filtros para a localização de Análises na Ontologia BI através do método <i>findGoalbyFilter</i>	147
Quadro 19 - Visão geral das características dos módulos funcionais da arquitetura vis-à-vis requisitos funcionais e não funcionais.....	148
Quadro 20 - Descrição do hardware e software utilizado no estudo de caso.	153
Quadro 21 - Sumário das funcionalidades recomendadas por OntoDSS de acordo com o contexto de uma análise	158
Quadro 22 - Métricas do Instituto Stela para o desenvolvimento de rotinas ETL para carga de tabelas de dimensões e fatos.....	168
Quadro 23 - Dimensionamento das bases de conhecimento utilizadas nos testes de performance sobre o framework OCML.....	173

Quadro 24 - Comparação entre as funcionalidades da arquitetura SBI e
iniciativas correlatas 178

Listagens

Listagem 1 - Definição das relações <i>institution-competitor</i> e <i>institution-alumnus</i> em OCML	84
Listagem 2 – Definição das instâncias utilizadas para mapear a dimensão <i>dim_grupo</i> ao conceito <i>research_group</i>	93
Listagem 3 – Fragmento da ontologia BI ilustrando uma hierarquia e a ligação entre a dimensão <i>dim_grupo</i> e tabela de fatos <i>fato_grupo</i>	94
Listagem 4 - Definição OCML das relações <i>label-role-idiom</i> e <i>presentation-attribute</i>	96
Listagem 5 - Definição OCML da análise do número de estudantes por instituição e grupos de pesquisa	100
Listagem 6 - Definição dos construtores semânticos para a definição do início de uma composição	113
Listagem 7 - Definição do construtor para <i>service component</i> e de uma instância correspondente ao primeiro serviço da composição exemplo.....	115
Listagem 8 - Definição da ligação de saída do primeiro <i>Service Component</i> e respectivo <i>Mediator</i> para o <i>Control Component</i> da composição	117
Listagem 9 – Construtor semântico utilizado para a definição de Control Components e sua instância no contexto da composição ilustrada na Figura 19	118
Listagem 10 - Demonstração de uma consulta antes e após o processo de redefinição	144
Listagem 11 - Ilustração da modificação na relação <i>institution-competitor</i>	170
Listagem 12 - Definição em OCML das regras para classificar docentes vinculados e não vinculados a uma instituição.....	172
Listagem 13 - Definição em OCML da relação <i>institution-competitor-city</i>	174

Lista de Siglas

API - Application Program Interface
BI - Business Intelligence
CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
C&T - Ciência e Tecnologia
CWM - Common Warehouse Metamodel
DL - Description Logics
DSS - Decision Support System
DW - Data Warehouse
ebXML - e-Business eXtended Markup Language
ESS - Executive Support System
ETL - Extraction, Transformation and Loading
HTTP - Hypertext Transfer Protocol
IRS - Internet Reasoning Services
KWS - Knowledge Work Systems
MIS - Management Information System
MOLAP - Multidimensional On-line Analytical Processing
OAS - Office Automation System
OCML - Operational Conceptual Modeling Language
OLAP - On-line Analytical Processing
OLTP - On-line Transaction Processing
OMG - Object Management Group
OWL - Ontology Web Language
OWL-S - OWL-based Web Service Ontology
RDF - Resource Description Framework
ROLAP - Relational On-line Analytical Processing
SAD - Sistema de Apoio a Decisão
SGBD - Sistema Gerenciador de Banco de Dados
SI - Sistema de informação
SQL - Structured Query Language
TI - Tecnologia da Informação
TPS - Transaction Processing Systems
UF - Unidade da Federação

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

WS - Web service

WSDL - Web Service Definition Language

WSMF - Web Service Modeling Framework

WSMO - Web Service Modeling Ontology

WSS - *Web services* semânticos

XML - eXtended Markup Language

XMLA - XML for Analysis

RESUMO

Soluções de Business Intelligence visam prover informação estratégica a partir das fontes de dados da organização. Entretanto, verificam-se desafios que permanecem não resolvidos nestas soluções no contexto da sociedade do conhecimento. Entre as principais deficiências das soluções atuais verificam-se a falta de perspectiva de utilização da semântica do negócio (i.e. sua terminologia e regras) no apoio ao processamento analítico e a falta de flexibilidade para a extensão das funcionalidades exploratórias de acordo com as especificidades de cada organização. Propõe-se uma arquitetura para BI - a arquitetura SBI – baseada em um novo paradigma que prevê a utilização intensiva do conhecimento do negócio para guiar o processamento analítico. A semântica do negócio na abordagem utilizada é representada em uma ontologia e mapeada aos dados e serviços em outras duas ontologias. O suporte à inferência sobre essa semântica é agregado aos módulos de gerenciamento de serviços e de análises da arquitetura, permitindo: a) extensão e composição de funcionalidades exploratórias através de *Web services* semânticos; b) recomendação de informação, serviços e inferências de acordo com o contexto da análise em um processo analítico-iterativo assistido pela arquitetura; e c) manipulação de dados e serviços utilizando a terminologia do negócio. A viabilidade da proposta é demonstrada através do protótipo de ferramenta analítica OntoDSS, o qual é integrado a arquitetura SBI e aplicado em um estudo de caso no domínio da gestão de Ciência & Tecnologia. O ineditismo da contribuição está na busca de insumos na engenharia de conhecimento para prover alternativas de solução para as limitações nas soluções de BI atuais.

Palavras-chave: *Business Intelligence*; OLAP; Ontologias; *Web services* semânticos.

ABSTRACT

Business Intelligence (BI) solutions aim at providing strategical information from organizations' data sources. However, critical issues in the context of knowledge society remain untackled. Among those issues is the lack of perspective on using business semantics (i.e. business terminology and rules) to support the analytical processing and the lack of flexibility for the extension of exploratory functionalities. A BI architecture is proposed – the architecture SBI - based on a new paradigm in which business knowledge is used intensively to support analytical processing. In this approach, business semantics is represented by an ontology and mapped to data and services onto two other ontologies. Support for inference over this semantic aspect is added to the other modules of the architecture, enabling: a) extension and composition of exploratory functionalities through Semantic Web Services; b) recommendation of information, services and inferences according to the context of analyses, in an interactive-analytical process supported by the architecture; c) manipulation of data and services by using business terminology. The viability of the proposal is demonstrated by the prototype of an analytical tool called OntoDSS which is integrated to the architecture SBI and applied to the domain of Science & Technology. The innovation of this research lies in its focus on knowledge engineering methodologies and techniques to provide alternatives to the aforementioned issues in the current BI solutions.

Keywords: Business Intelligence; Analytical Tools; Ontologies; Semantic Web Services.

1 INTRODUÇÃO

1.1 Apresentação

A constante transformação dos valores e das prioridades da sociedade prescreve uma nova agenda de pesquisa. Essa agenda muda o enfoque até então na automação industrial e no desenvolvimento de tecnologia, para a identificação das necessidades da sociedade e na criação e no compartilhamento do conhecimento para o desenvolvimento humano e das organizações. Tal mudança de enfoque caracteriza o ponto de inflexão entre a sociedade industrial e a sociedade do conhecimento. No contexto das organizações, segundo Hesselbein et al. (1997), as transformações são tão profundas que é possível afirmar que está havendo uma terceira revolução industrial, ou revolução do conhecimento. Essa era pós-industrial passa de uma sociedade baseada na manufatura (*i.e.*, o “fazer”) e na aplicação de tecnologia para outra com interesse crescente no valor da informação, na sua difusão e interpretação (*i.e.*, o “saber”).

A sociedade do conhecimento preconiza o desenvolvimento de meios para identificação, produção, processamento, transformação, disseminação e uso de informação para criar e aplicar conhecimento no desenvolvimento da sociedade (UNESCO, 2005). No contexto da era do conhecimento, as organizações vencedoras serão aquelas que conseguirem acúmulo de saber, contando com o empenho coletivo, a capacidade das pessoas envolvidas de se relacionarem umas com as outras, dentro de uma linguagem comum, de esforço conjunto (HOWARD, 2000). Se até então o conhecimento era guardado para benefício do seu proprietário, hoje ele deve ser compartilhado, pois o conhecimento compartilhado cresce, enquanto o conhecimento não utilizado torna-se obsoleto e perde o seu valor.

No contexto do desenvolvimento da tecnologia de informação, soluções de *Business Intelligence* (BI) visam oferecer os meios necessários para a transformação de dados em informação e para a sua divulgação através de ferramentas analíticas, a fim de suportar o processo decisório (BROHMAN et al., 2000; DEVANEY, 2001; GANGADHARAN; SWAMI, 2004; SIMMERS, 2004). Tais soluções são compostas tipicamente de ferramentas ETL (*i.e.*,

Extraction, Transformation and Loading ou Extração, Transformação e Carga), de um ou vários *data marts* e de ferramentas de processamento analítico e de mineração de dados (CÔRTEZ, 2002; KIMBALL et al., 1998; MOSS; ATRE, 2003). As ferramentas de processamento analítico para apoio à decisão permitem a criação de análises (*i.e.*, relatórios e cubos de dados) obedecendo a diferentes perspectivas sobre os dados disponíveis (KIMBALL et al., 1998; THOMSEN, 2002). As funcionalidades exploratórias oferecidas por essas aplicações analíticas limitam-se normalmente a um conjunto de funcionalidades genéricas de navegação e filtragem de dados sobre os *data marts*.

De acordo com o Instituto IDC, o mercado para soluções de BI vale mais de 7 bilhões de dólares mundialmente e deve dobrar até o final de 2006 (COMPUTERWORLD, 2005). Outro número que comprova a inserção e a importância dessas soluções nas organizações é apresentado pela Forrester Research, a qual realizou uma pesquisa com empresas europeias e americanas e apurou que mais de 44% dessas companhias iniciaram a adoção de soluções de BI em 2005 (BUSINESSINTELLIGENCE.COM, 2005). Entretanto, de acordo com o Gartner Group (2004) e com a ComputerWorld (2004), até 2007 cerca de 50% dos projetos de BI vão ter uma aceitação limitada por parte dos usuários ou vão fracassar.

De fato, muitas organizações já constataram a complexidade em se traduzirem os dados providos por essas soluções em conhecimento e resultados positivos para o negócio (HOFFMAN, 2001). Entre as causas enumeradas pelos institutos de pesquisa encontra-se a falta de aderência das soluções implantadas com os requisitos analíticos das organizações (COMPUTERWORLD, 2004; GARTNER GROUP, 2004). Isso ocorre porque as soluções de BI foram concebidas ainda sob influência dos valores da sociedade industrial, priorizando o desenvolvimento tecnológico para facilitar a integração de dados e o processamento rápido de consultas, mas subjugando os valores e as necessidades de conhecimento específicos de cada organização (LIEBOWITZ, 2005b; WHITE, 2005).

As soluções de BI de fato não provêm a “inteligência” referida no termo que cunha essas aplicações aos tomadores de decisão, pois se limitam a oferecer funcionalidades genéricas para exploração das montanhas de dados

reunidas nos repositórios das organizações (LIEBOWITZ, 2005a). A esse grande volume de dados é submetido o tomador de decisão, que dispõe unicamente do seu conhecimento pessoal para apoiá-lo durante o processamento analítico, conhecimento este que muitas vezes se mostra insuficiente para fazer os julgamentos corretos em um processamento tão complexo. Esse fenômeno que ocorre cotidianamente nas organizações usuárias de soluções de BI é qualificado pela UNESCO (2005) como “desconhecimento”. A seleção, transformação e análise da informação para formar conhecimento ainda dependem exclusivamente das habilidades cognitivas, críticas e teóricas dos tomadores de decisão.

As organizações modernas, ante os desafios e as oportunidades da sociedade do conhecimento, almejam soluções que permitam a utilização da terminologia e das regras do seu negócio de forma efetiva no processamento analítico (WHITE, 2005). Nesse contexto, uma das maiores deficiências das soluções atuais de BI é a impossibilidade de utilização da *semântica do negócio*¹ para apoiar a localização, a seleção e a transformação de informação para formar e divulgar conhecimento aos seus colaboradores e parceiros estratégicos (LIEBOWITZ, 2005b). Atualmente, várias empresas já possuem um repositório de conhecimento constituído e agora buscam maneiras de extraírem todo o potencial desses repositórios em busca de diferencial competitivo.

O advento da *Web* semântica despertou o interesse da comunidade científica para áreas tradicionais de pesquisa, tais como a de Engenharia de Conhecimento, em busca de novas abordagens para tratamento de problemas tradicionais como a integração e a recuperação de informação (ANTONIOU; VAN HARMELEN, 2004; BERNERS-LEE et al., 2001; DACONTA et al., 2003; DAVIES et al., 2003). O principal objetivo da *Web* semântica é o de tornar possível a descrição dos recursos disponibilizados na *Web* ou reunidos em

¹ Semântica do negócio nesta tese refere-se os conceitos utilizados para descrever a terminologia relacionada aos processos do negócio de uma organização, além dos conceitos utilizados no mapeamento das suas unidades de informação e das suas regras de negócio.

repositórios pessoais de maneira a permitir que agentes localizem conteúdo e realizem inferências sobre esse conteúdo (ALESSO; SMITH, 2004; BERNERS-LEE et al., 2001). Para tanto, ontologias vêm sendo utilizadas para agregar descrição semântica e lógica ao conteúdo da *Web*, de forma a impulsionar a *Web* corrente para a *Web* semântica (FENSEL, 2001).

Uma ontologia identifica as entidades e os relacionamentos em um universo de discurso, definindo um vocabulário conceitual para suportar referências e raciocínio sobre esse domínio. O conhecimento representado através de ontologias pode ser explorado por máquinas de inferência visando à geração de conhecimento adicional (FENSEL, 2001; GRUBER, 1993; GUARINO, 1998; MOTTA, 1999).

Web service (WS) é outra tecnologia-chave para impulsionar a *Web* e o desenvolvimento dos sistemas de informação (W3C, 2002). Essa tecnologia permite a reutilização de código e tem sido aplicada largamente em projetos de integração de sistemas. No contexto da *Web* semântica, as ontologias são aplicadas para descrever as funcionalidades providas por um WS de forma a tornar possível a descoberta automática e a composição de WS de acordo com as necessidades de um usuário na *Web* ou em uma Intranet (ALESSO; SMITH, 2004; CABRAL et al., 2004; MARTIN et al., 2004; MCILRAITH et al., 2001; MOTTA et al., 2003; SYCARA et al., 2003). *Web services* semânticos (WSS) buscam prover formas de processamento e raciocínio sobre a representação estática provida pelas ontologias na *Web* semântica (GÓMEZ-PEREZ et al., 2004).

Examinando-se a literatura e os produtos relacionados a BI, verifica-se que existe um nicho a ser explorado por uma nova linha de pesquisa, a do desenvolvimento de soluções de BI baseadas em conhecimento. Nesta tese, são propostas novas possibilidades de integração dos conceitos e das tecnologias da área de engenharia de conhecimento com uma área tradicionalmente dominada pela indústria de software, a área de BI. As iniciativas comerciais e acadêmicas na aplicação de processamento semântico nessa categoria de aplicações estão em sua maioria restritas ao processo de integração de dados, e não ao apoio ao processamento analítico.

Esta pesquisa almeja oferecer aos desenvolvedores de software uma alternativa de arquitetura para a concepção de soluções de BI. Na abordagem proposta, a qual é detalhada nas seções seguintes, utilizam-se as tecnologias que vêm sendo desenvolvidas no contexto da *Web* semântica para permitir a criação de soluções de BI diferenciadas, considerando-se as limitações das soluções atuais. Através da representação semântica das informações e dos serviços de interesse da organização, além da descrição de suas regras de negócio, torna-se possível suportar inferências semânticas não exploradas até então nas soluções analíticas atuais e tampouco discutidas na literatura correlata. Procura-se despertar a atenção da comunidade de desenvolvimento e da academia para as possibilidades que a representação e a inferência semântica podem oferecer para apoiar o processamento analítico de tomada de decisão, utilizando de forma ativa a semântica do negócio para apoiar o processo decisório.

Na Seção seguinte, descreve-se de forma mais detalhada a problemática que a pesquisa se propõe a investigar.

1.2 Caracterização do problema

As demandas das organizações modernas exigem formas de empregar o conhecimento sobre o seu negócio como insumo ao processo decisório (WHITE, 2005; LIEBOWITZ, 2005b). A estratégia das soluções correntes de BI é focada na integração dos dados e na disponibilização de funcionalidades para permitir a exploração dos repositórios de dados. Essas funcionalidades exploratórias, na maioria das soluções de BI, são inspiradas na lista de pré-requisitos instituída por Codd et al. (1995), as quais definem características mínimas que uma solução analítica deve contemplar. No entanto, o alto percentual de fracasso descrito pelos institutos de pesquisa demonstra que não basta dar acesso às montanhas de dados acumuladas nos repositórios das organizações.

Deve-se buscar uma estratégia em que a semântica do negócio possa ser utilizada para guiar os colaboradores das organizações a obterem informação para auxiliar na transformação e na interpretação do dado localizado e para apoiar a formação de conhecimento e a sua divulgação. O

dado por si só, sem o contexto do negócio e isolado em tabelas e documentos, constitui artefato de pouca importância no processo decisório (LIEBOWITZ, 2005b). Dessa forma, deverão ser considerados aspectos até então negligenciados, tais como o mapeamento da semântica do negócio e sua utilização intensiva durante o processo de recuperação de informação e transformação desta em conhecimento. Para tanto, nesta pesquisa, são abordadas duas importantes limitações das soluções de BI, como se segue.

- A. Falta de suporte para o acoplamento da semântica do negócio às arquiteturas de BI e para a utilização dessa semântica no apoio ao processo de seleção, transformação e análise de informações.
- B. Dificuldade para a extensão ou personalização das funcionalidades exploratórias nas ferramentas analíticas, levando-se em consideração as especificidades do negócio.

1.2.1 Suporte para representação e utilização da semântica do negócio no apoio ao processamento analítico

Em relação à falta de suporte para a representação e utilização da semântica do negócio nas soluções correntes de BI, primeiramente deve-se analisar a forma com que são estruturadas as arquiteturas de BI. Essas arquiteturas não prevêem formas de integração do conhecimento do negócio aos seus módulos para o suporte do processamento analítico. O conhecimento sobre o negócio é obtido pelos arquitetos de BI durante a etapa de concepção da solução os quais tentam projetar esse conhecimento no modelo de dados do *data warehouse* e na codificação das rotinas de ETL (INMON, 2002; KIMBALL et al., 1998). Essa visão do negócio é então apresentada aos usuários da solução através de cubos de dados. Os usuários muitas vezes não conseguem compreender qual a relação desses cubos com os conceitos do negócio e acabam abandonando a solução por não conseguirem satisfazer as suas necessidades analíticas, engrossando as estatísticas de fracasso desse tipo de projeto (GARTNER GROUP, 2004).

Além do distanciamento das soluções dos conceitos do negócio, a velocidade com que as regras do negócio mudam faz com que muitas soluções de BI tornem-se obsoletas antes mesmo de serem disponibilizadas aos

tomadores de decisão (BUSINESSINTELLIGENCE, 2006). Nesse contexto, as ferramentas analíticas das soluções de BI pouco contribuem para facilitar o processo decisório. Essas ferramentas oferecem apenas funcionalidades genéricas para a exploração dos dados das organizações através de funcionalidades como *drill-down* (i.e., navegação de um nível macro para o detalhe da informação) e de *slice* (i.e., aplicação de filtros sobre a informação). De acordo com a ComputerWorld (2005), 48% dos usuários de ferramentas analíticas precisam de suporte contínuo da equipe de TI nas organizações para a formatação de análises. Esse suporte compreende a identificação da informação nos cubos de dados para o usuário, o suporte no processamento dos dados e a criação de cubos de dados específicos levando-se em consideração novas regras de negócio.

A criação de cubos específicos e de adaptações no DW ocorre com frequência e pode representar um custo adicional de até 80% por ano ao valor inicial do projeto (BUSINESSINTELLIGENCE, 2006). Esse custo adicional advém de fatores como a falta de suporte para a explicitação de regras de negócio nas soluções de BI para que os usuários possam utilizá-las na execução de operações como *slice* e *drill-down*. Por exemplo, para um pró-reitor de uma instituição de ensino seria importante empregar regras de negócio como *egressos* e *concorrentes* para a manipulação de suas análises. Essas regras poderiam ser utilizadas, por exemplo, para efetuar um *slice* sobre um cubo que reúne dados acerca da produtividade dos pesquisadores por região de endereço, com a finalidade de identificar quantos dos pesquisadores com maior número de patentes são seus *egressos* e quantos são *egressos* de seus *concorrentes*. Com o passar do tempo, a noção de *concorrente*, por exemplo, pode ser alterada conforme as mudanças no planejamento estratégico da instituição deste pró-reitor. Nas soluções atuais de BI, essas regras precisam ser implementadas diretamente no sistema ETL, e novas tabelas deveriam ser agregadas ao DW. A cada modificação nas regras, o sistema ETL precisa ser modificado, bem como o DW e os cubos de dados relacionados.

Verifica-se nesse contexto um hiato de pesquisas acadêmicas para a abordagem desses desafios. A maior parte dos trabalhos relacionados a BI

restringe-se a aplicar ontologias apenas para suportar a integração semântica² de dados entre *data marts* e outros repositórios de dados. Nesses trabalhos, as ontologias são aplicadas no mapeamento entre as estruturas de repositórios de dados distribuídos a fim de permitir intercâmbio de dados entre as aplicações e o processamento de consultas sobre repositórios de dados heterogêneos e distribuídos.

Verificam-se apenas três iniciativas que descrevem a utilização de ontologias para apoiar aplicações analíticas em BI: SEWASIE (BERGAMASCHI et al., 2004, 2005), BIKM (CODY et al., 2002) e a proposta de Priebe e Pernul (2003). No entanto, essas pesquisas estão focadas basicamente na utilização de ontologias para relacionar documentos aos dados dos *data marts*. Esses documentos são recuperados à medida que o usuário utiliza ferramentas analíticas específicas. Nenhuma dessas iniciativas aplica as ontologias para o apoio na navegação sobre os repositórios de dados e para suportar as funcionalidades exploratórias das ferramentas OLAP (como no suporte para operações de *drill* ou *slice* utilizando regras ou relações dos conceitos do negócio). Mais detalhes sobre as diferenças entre essas arquiteturas e a proposta desta pesquisa são apresentados nas seções 3.6 e 7.5.

De fato, pouca ênfase é dada na academia e na indústria de *software* sobre a aplicação de ontologias para utilização do conhecimento do negócio no suporte ao processamento analítico. Deve-se investigar como a semântica do negócio pode ser capturada, integrada a arquiteturas de BI e processada pelas aplicações analíticas de forma a oferecer novas possibilidades de navegação sobre os dados e as funcionalidades exploratórias que sejam guiadas por essa representação semântica.

Dessa forma, surge a primeira problemática da pesquisa:

² Vide Motik et al. (2002), Hofreiter (2002), Maedche et al. (2003), Sure et al. (2002) e Fileto et al. (2003) para obter mais informações sobre a aplicação de ontologias no processo de integração de dados.

Problemática 1: como integrar a representação da semântica do negócio às soluções de BI e utilizar esse conhecimento no apoio ao processamento analítico?

Os desenvolvimentos nas áreas da engenharia do conhecimento e de tecnologias correlatas como ontologias e *Web services* semânticos oferecem o potencial para a criação de novas alternativas de exploração das fontes de dados para a formação de conhecimento útil ao processo decisório nas organizações. Nesse contexto, ontologias de domínio que descrevam o negócio da organização podem ser utilizadas para a criação de operações semântico-analíticas inéditas que combinem as funcionalidades exploratórias atuais com inferência semântica sobre os conceitos e as regras de negócio.

1.2.2 Extensão de Funcionalidades Exploratórias e Recomendação de Funcionalidades durante o Processo Decisório

O segundo desafio está relacionado à maneira através das quais as aplicações analíticas das soluções de BI são concebidas. As soluções atuais foram criadas partindo-se do princípio de que se deve atender ao maior número de organizações possível a partir de um núcleo básico e genérico de funcionalidades analíticas. Essas soluções contam com um conjunto limitado de funcionalidades exploratórias e não apresentam maneiras escaláveis para extensão ou adaptação dessas funcionalidades de acordo com as necessidades específicas de cada organização.

De forma geral, as ferramentas analíticas oferecem ao desenvolvedor a possibilidade de agregação de novas funcionalidades através de uma linguagem de programação específica e do uso de uma API disponibilizada pelo fabricante (CA MAGAZINE, 2004-2005). Entretanto, essas soluções de customização não são escaláveis, pois exigem um conhecimento sobre uma linguagem específica e não permitem o reuso de aplicações já existentes na organização ou na Web.

Uma solução escalável deve ser oferecida através de padrões abertos, que sejam de conhecimento comum e que permitam que os desenvolvedores das organizações realizem customizações ou adicionem novas funcionalidades

para atender às especificidades de suas organizações. Além disso, devem-se investigar formas de se oferecer, automaticamente, alternativas de funcionalidades exploratórias ao usuário de acordo com o contexto de suas análises, visando guiá-lo e facilitar o processamento da informação e a extração de conhecimento útil ao processo decisório. Dessa forma, surge a segunda problemática a ser abordada pela pesquisa:

Problemática 2: como prover aos desenvolvedores a possibilidade de customização das funcionalidades exploratórias nas aplicações analíticas e aos usuários a possibilidade de receber recomendações dessas funcionalidades de acordo com o contexto de suas análises?

Tendências na área de integração de aplicações incluem XML e *Web services* como instrumentos para a integração de dados, aplicações e processos de negócio (DACONTA et al., 2003; SHETH; RAMAKRISHNAN, 2003). Alguns provedores de soluções de BI (como *Business Objects*, *Hyperion* e *MicroStrategy*) expuseram parte de suas APIs através de *Web services*, visando facilitar a chamada de suas operações analíticas por aplicações de terceiros (HURWITZ GROUP, 2002) a partir da gramática XMLA (*XML for Analysis*) (XMLA, 2004). Uma solução completa deveria também permitir que as aplicações analíticas invocassem serviços externos.

Nas soluções atuais, para que uma informação seja disponibilizada pelas aplicações analíticas, deve-se primeiro proceder a uma modificação no modelo de dados do DW e em seguida deve-se desenvolver uma rotina ETL específica para capturar o dado e integrá-lo ao modelo do DW. Esse processo normalmente representa um processo custoso e demorado (BUSINESSINTELLIGENCE, 2006). Um modelo mais escalável e ágil deveria permitir que serviços pudessem ser adicionados à aplicação analítica para que dados adicionais (internos ou externos à organização) pudessem ser trazidos diretamente das fontes de origem dos dados para o cubo ou relatório do usuário. Por exemplo, um *Web service* do IBGE poderia ser chamado por uma aplicação analítica no momento em que o usuário solicitasse detalhes demográficos de uma determinada região. Esse recurso permitiria a incorporação de dados à análise corrente sem exigir que fosse realizada uma

alteração no modelo de dados do DW local e desenvolvidas rotinas ETL para integração dos dados externos ao DW local. Essa solução pode ser interessante principalmente quando for necessário acessar dados externos para a busca de detalhes adicionais relacionados ao escopo de uma análise.

Serviços também poderiam ser invocados pelas aplicações analíticas para proceder a outros tipos de transações usando os dados de uma análise como entrada. Por exemplo, uma aplicação de ranking de pesquisadores por produtividade poderia ser acionada por um diretor de departamento no momento da análise da distribuição de vagas de orientação em uma aplicação analítica. Essa aplicação de ranking poderia ser desenvolvida em qualquer linguagem e exposta através de um *Web service* para ser posteriormente invocada pela aplicação analítica. Dessa forma, poder-se-ia oferecer mais flexibilidade aos desenvolvedores para adaptar as funcionalidades exploratórias de acordo com as necessidades analíticas dos tomadores de decisão.

Para melhor atender aos requisitos analíticos dos usuários, deve-se ainda lidar com o desafio de criação de mecanismos de identificação automática da necessidade de informação do usuário e da identificação das capacidades dos *Web services* disponíveis para tornar possível a recomendação de serviços que possam ser úteis ao usuário. Com a proliferação de *Web services* nas organizações e na *Web*, um tomador de decisão não conseguiria identificar qual serviço poderia satisfazer sua necessidade de processamento analítico. Isso ocorre porque a especificação XML dos *Web services* limita-se à descrição sintática das entradas e saídas dos serviços (WSDL, 2001; W3C, 2002). Dessa forma, torna-se inviável a localização (automática ou através de buscas guiadas pelos usuários) de um serviço de acordo com as suas funcionalidades (DOMINGUE et al., 2004a; MARTIN et al., 2004; SYCARA et al., 2003).

XML e *Web services* aplicados em conjunto com tecnologias semânticas e com a semântica do negócio podem alavancar o processamento analítico, permitindo a livre customização das funcionalidades exploratórias e a recomendação automática de serviços ao usuário durante o processo de investigação. Com a explicitação da semântica da informação manipulada em

uma análise através de ontologias, é possível inferir qual é o *contexto* de uma análise. *Contexto de uma análise* refere-se nesta tese à descrição semântica dos elementos de dados introduzidos pelo usuário em um cubo de dados ou relatório e a disposição desses elementos nas dimensões e nos cabeçalhos da sua análise. Essa descrição semântica envolve ainda os filtros e parâmetros selecionados, além das operações de sumarização utilizadas para o processamento das medidas da análise.

A descrição semântica do contexto da análise é utilizada na presente tese para apoiar o processo de inferência da necessidade de informação do usuário. O conjunto de informações reunido nessa descrição semântica permite identificar qual é o assunto da análise (*i.e.*, quais são as medidas reunidas na análise), qual é a granularidade da análise (*i.e.*, quais dimensões foram utilizadas para o agrupamento das medidas) e quais são os focos de interesse sobre o tema de análise (*i.e.*, quais os filtros foram relacionados na análise).

Ontologias são aplicadas ainda nesta tese para a descrição semântica dos *Web services* de forma a apoiar a localização de serviços pela descrição semântica de suas funcionalidades, a partir da identificação do contexto de uma análise. Dessa forma, torna-se possível, por exemplo, oferecer automaticamente ao tomador de decisão o *Web service* do IBGE citado anteriormente no momento em que o usuário está analisando o volume de publicações por Unidades Federativas. Essa inferência torna-se possível porque o *Web service* referido manipula uma unidade de informação relacionada na análise do usuário.

A descrição semântica de *Web services* nesta tese é utilizada ainda para suportar a composição de serviços para a realização de processamentos mais complexos que possam compreender vários serviços. Por exemplo, no cenário apresentado anteriormente sobre a análise da distribuição de vagas de orientação, um novo serviço poderia ser acoplado à saída do serviço que processa o ranking. Esse segundo serviço poderia, a partir do ranking gerado pelo primeiro serviço, apresentar uma proposição de número de vagas de orientação para cada professor, de acordo com o limite de vagas por orientador definido pelo departamento. Composições como esta podem oferecer aos usuários recursos ilimitados de processamento de informação e geração de

conhecimento. Esse recurso pode ainda oferecer aos desenvolvedores a liberdade para a concepção de aplicações modulares a partir da reutilização de código existente na organização ou na *Web*.

Não foram localizadas referências acadêmicas ou iniciativas da indústria de *software* na aplicação de tecnologias semânticas para a criação de um modelo de BI adaptável às necessidades analíticas das organizações.

1.3 Pressuposto da Pesquisa

Conforme apresentado nas seções anteriores, verificam-se várias limitações nas soluções atuais de BI que permanecem não resolvidas com as abordagens tradicionais de desenvolvimento de BI. Verificam-se, no entanto, características nas tecnologias desenvolvidas no contexto da engenharia de conhecimento que podem ser exploradas na busca de soluções para os problemas da dissociação da semântica do negócio das soluções de BI e da falta de flexibilidade para extensão das soluções atuais.

Assim, parte-se do pressuposto de que tecnologias semânticas, combinadas com os componentes tradicionais de uma arquitetura de *Business Intelligence*, proverão meios para a utilização da semântica do negócio para a criação de alternativas de processamento analítico e de extensão para as soluções de BI.

1.4 Objetivos

Com a finalidade de contemplar o pressuposto definido anteriormente, os objetivos desta tese são relacionados a seguir.

1.4.1 Objetivo Geral

O objetivo geral deste trabalho é propor uma arquitetura para BI que permita a incorporação da semântica do negócio, além da descrição semântica dos serviços e dados de interesse aos tomadores de decisão, visando apoiar a construção de aplicações analíticas *flexíveis, orientadas às necessidades dos tomadores de decisão* e que possam *guiar o usuário no processo decisório*.

Por *flexíveis* entende-se a criação de meios para que desenvolvedores agreguem novas funcionalidades analíticas a uma solução de BI a partir da

reutilização de implementações existentes na organização ou na *Web*. A agregação deverá ser feita através da anotação semântica do código existente, sem necessidade de adaptação no código. Por *flexíveis* entende-se ainda a possibilidade do desenvolvedor customizar as regras do negócio diretamente sobre a ontologia que descreve a semântica do negócio. Essas regras deverão ser utilizadas para suportar, entre outras, as operações de *drill* e *slice*, permitindo dessa forma a adaptação do comportamento da ferramenta analítica sem necessidade de alteração do código-fonte das suas funcionalidades.

Por *orientadas às necessidades dos tomadores de decisão* entende-se a possibilidade de, através do mapeamento dos dados à semântica do negócio, oferecer aos usuários a oportunidade de elaboração de análises através da navegação sobre os conceitos do negócio (independentemente da forma como o dado está organizado). A representação da semântica do negócio permitirá ainda a utilização das regras do negócio para suportar inferências sobre os resultados de uma análise. Essas inferências compreenderão, além do apoio às operações de *drill*, a sugestão de serviços e de dados relacionados ao contexto da análise, com a finalidade de *guiar o tomador de decisão* na análise do seu negócio.

1.4.2 Objetivos Específicos

- Propor uma arquitetura (*i.e.*, a arquitetura SBI) que permita a incorporação da semântica do negócio nos seus alicerces de forma a possibilitar a criação de funcionalidades analíticas guiadas pelo conhecimento do negócio.
- Permitir na arquitetura a representação semântica dos dados da organização com o intuito de oferecer acesso transparente sobre esses dados aos tomadores de decisão e possibilitar inferências sobre essa representação semântica.
- Agregar à arquitetura a possibilidade de incorporação de novas funcionalidades exploratórias às ferramentas analíticas a partir da reutilização do código existente na organização ou disponibilização remota desse código por terceiros.

- Permitir a descrição semântica dos serviços agregados à arquitetura para suportar a localização e composição desses serviços de acordo com as necessidades dos tomadores de decisão.
- Agregar à arquitetura a capacidade de inferência sobre a descrição semântica dos dados e dos serviços, e sobre o conhecimento e a lógica do negócio, de forma a suportar a recomendação desses recursos de acordo com o contexto das explorações feitas nas ferramentas analíticas.
- Implementar um protótipo de ferramenta analítica sobre a arquitetura para demonstrar a viabilidade da solução proposta.

1.5 Justificativa

Conforme apresentado anteriormente, apesar da importância das soluções de BI no contexto do processo decisório, verificam-se limitações relacionadas à falta de flexibilidade para adaptação dessas soluções às necessidades dos tomadores de decisão e à impossibilidade de se utilizar a semântica do negócio para apoiar o processamento analítico. Essas deficiências estão entre as causas identificadas por institutos de pesquisa pelo alto custo de manutenção, o que leva à preocupante previsão de que até 2007 cerca de 50% dos projetos de implantação de BI vão falhar (BUSINESSINTELLIGENCE, 2006; COMPUTERWORLD, 2004; GARTNER GROUP, 2004).

Além das deficiências identificadas, verificam-se novas demandas que surgem com a quebra de paradigma proposto pela sociedade do conhecimento. Essas demandas não estão contempladas na lista de pré-requisitos descritos por Codd et al. (1995) para aplicações analíticas. Esses pré-requisitos, constituídos na forma de 12 regras, apesar de não serem consenso entre fabricantes de soluções de BI, guiaram a estruturação da maioria das aplicações analíticas e, por conseqüência, as arquiteturas de BI atuais. No entanto, o atendimento dessas regras significa apenas que a solução respeita um conjunto de características técnicas compatíveis com a tecnologia dominada pelos fabricantes de BI.

Os desafios apresentados na caracterização do problema (Seção 1.2) permanecem insolúveis pela forma com que as soluções de BI são concebidas atualmente. É necessário, portanto, pesquisar como a semântica do negócio pode ser utilizada pelos desenvolvedores de maneira efetiva na constituição de soluções de BI para apoiar o processo decisório. A necessidade de se abordarem as deficiências correntes das soluções de BI e as novas demandas não se justificam apenas pelas deficiências identificadas pelos institutos de pesquisa e pela importância dessas ferramentas no processo decisório das organizações, mas também pela virtual inexistência de pesquisas que tratem especificamente desses problemas.

Propõe-se uma abordagem inédita para o desenvolvimento de arquiteturas de BI, a qual envolve técnicas que vêm sendo desenvolvidas pela área de Engenharia do Conhecimento (*i.e.*, representação e raciocínio sobre conhecimento do domínio). No entanto, apesar de buscar insumos para a realização da pesquisa na Engenharia do Conhecimento, a contribuição desta pesquisa não está inserida apenas nessa área.

Esta pesquisa está voltada à otimização de métodos e produtos em virtude de necessidades reais do mercado, foco da área de Engenharia de Produção, área de desenvolvimento desta tese. De acordo com a Associação Brasileira de Engenharia de Produção (ABEPRO, 2005), a Engenharia de Produção é uma área voltada à pesquisa, ao desenvolvimento ou ao aperfeiçoamento de sistemas que envolvem pessoas, materiais, equipamentos e ambiente com o intuito de suprir as necessidades do mercado e da sociedade. Trata-se de uma área abrangente, que engloba um conjunto de teorias e técnicas de outras áreas de pesquisa, buscando uma combinação destas para a resolução de um problema identificado em produtos ou processos nas organizações.

Através do estudo das tecnologias semânticas popularizadas pela Web semântica e em desenvolvimento no contexto da engenharia do conhecimento, busca-se nesta tese apresentar alternativas para contribuir na resolução das deficiências identificadas nas soluções atuais de BI. Elabora-se uma arquitetura que permitirá aos desenvolvedores oferecer aos usuários acesso às informações e aos serviços necessários para suas análises, levando-se em

conta a terminologia própria do negócio. A arquitetura provê ainda os meios para que ferramentas analíticas possam guiar seus usuários durante o processo decisório através da recomendação de informações, serviços e outras análises relevantes ao contexto do processo decisório.

Procura-se com este estudo munir os desenvolvedores de software de meios para a criação de ferramentas analíticas adaptáveis às especificidades das organizações. Para tanto, busca-se utilizar a representação da semântica do negócio destas organizações e de serviços disponíveis para apoiar o processamento analítico. Deseja-se prover os meios para a criação de soluções capazes de utilizar esta semântica visando oferecer novas formas de exploração dos repositórios de informação para a obtenção de conhecimento sobre o negócio. Além disso, procura-se utilizar dos recursos oferecidos pelas tecnologias semânticas para inferir o contexto do processo decisório e oferecer informações proativas e alternativas exploratórias ao tomador de decisões.

1.6 Delimitação do Escopo

Sendo o objetivo principal desta pesquisa a proposição de uma nova arquitetura para BI que utilize a semântica do negócio para apoiar o processamento analítico, não são abordados os componentes de arquiteturas de BI relacionados ao processamento ETL. Além disso, apesar de outros tipos de ferramentas de apoio à decisão como ferramentas de *data mining* e outras ferramentas de apresentação de dados como *dashboards* e de *score carding* serem prováveis beneficiárias da abordagem proposta neste trabalho, o estudo de caso está voltado para o escopo das ferramentas analíticas. Essa decisão se deve às várias especificidades observáveis em cada tipo de ferramenta de apoio à decisão, tornando-se inviável ao autor explorar cada uma dessas especificidades no escopo de uma tese de doutorado.

A arquitetura proposta prevê a integração de uma ontologia em que são representadas a terminologia, as regras de negócio e outras informações sobre os processos de uma organização. A ontologia (Ontologia do Domínio) deverá, nesta versão da arquitetura, ser atualizada a partir do uso de uma ferramenta ETL de terceiros. Não está prevista nesta versão da arquitetura a

implementação de um módulo específico para a criação e atualização da Ontologia do Domínio.

O principal aspecto observado na demonstração de viabilidade da arquitetura é a possibilidade de se utilizarem tecnologias semânticas e conhecimento sobre o negócio para contribuir para a resolução das limitações identificadas na Seção 1.2. Para a definição da arquitetura SBI, foi observado um conjunto de requisitos funcionais definidos a partir de uma análise das principais deficiências das soluções atuais no que tange ao processamento analítico e que permanecem insolúveis. Além disso, foram levadas em consideração características de sistemas baseados em conhecimento não previstas na lista de regras de Codd et al. (1995) e que permanecem inéditas no contexto de soluções de BI. Esse conjunto de requisitos é apresentado na Seção 4 desta tese.

Assim sendo, o estudo de viabilidade não está focado em requisitos não funcionais, como a ergonomia da ferramenta analítica desenvolvida sobre a arquitetura, e tampouco na mensuração de desempenho e do nível de satisfação dos usuários sobre as funcionalidades exploratórias providas pela arquitetura proposta. A demonstração de viabilidade se dá por meio de um estudo de caso que ilustra a maneira com que os requisitos funcionais identificados foram atendidos. Descrevem-se ainda os recursos inéditos resultantes da combinação de técnicas desenvolvidas no contexto da engenharia de conhecimento e de BI.

1.7 Metodologia

Visando à concepção e ao desenvolvimento deste trabalho, foi realizada, inicialmente, uma pesquisa bibliográfica sobre a forma como são concebidas as soluções atuais de *Business Intelligence* e suas aplicações analíticas de apoio à decisão. Em seguida, realizou-se um estudo a respeito das tecnologias semânticas que poderiam ser aplicadas para o alcance dos objetivos deste trabalho.

Assim, para que os objetivos sejam alcançados, o trabalho foi dividido nas etapas a seguir.

1. **Estudo sobre sistemas de informação e soluções de BI:** nesta etapa é realizado um estudo sobre a evolução dos sistemas de informação e sobre as principais características das soluções atuais de *Business Intelligence*, dando-se ênfase ao levantamento das principais limitações a serem abordadas no contexto da sociedade do conhecimento.
2. **Estudo sobre tecnologias semânticas:** é realizado um estudo sobre os conceitos e as formas de aplicação de ontologias para a representação de conhecimento de um dado domínio e para o suporte de inferências sobre esse domínio. Além disso, investigam-se as formas como as ontologias são aplicadas na descrição das funcionalidades de *Web services* e como essas tecnologias vêm sendo aplicadas no domínio de soluções de BI.
3. **Desenvolvimento da arquitetura:** nesta etapa é identificado um conjunto de requisitos a partir das novas demandas das organizações e das deficiências das soluções de BI atuais. São especificados os módulos que comporão a arquitetura SBI *vis-à-vis* os requisitos levantados. São identificados os novos papéis que a equipe de desenvolvimento deverá assumir em face da nova forma de estruturação de soluções de BI através da arquitetura SBI.
4. **Estudo de viabilidade:** nesta etapa é desenvolvido um protótipo de ferramenta de apoio à decisão sobre a arquitetura, e sua aplicação é demonstrada em um estudo de caso para confirmar a viabilidade da arquitetura SBI. Avaliam-se como os requisitos funcionais definidos são atendidos pela arquitetura no estudo de caso.

A Figura 1 a seguir ilustra a metodologia descrita.

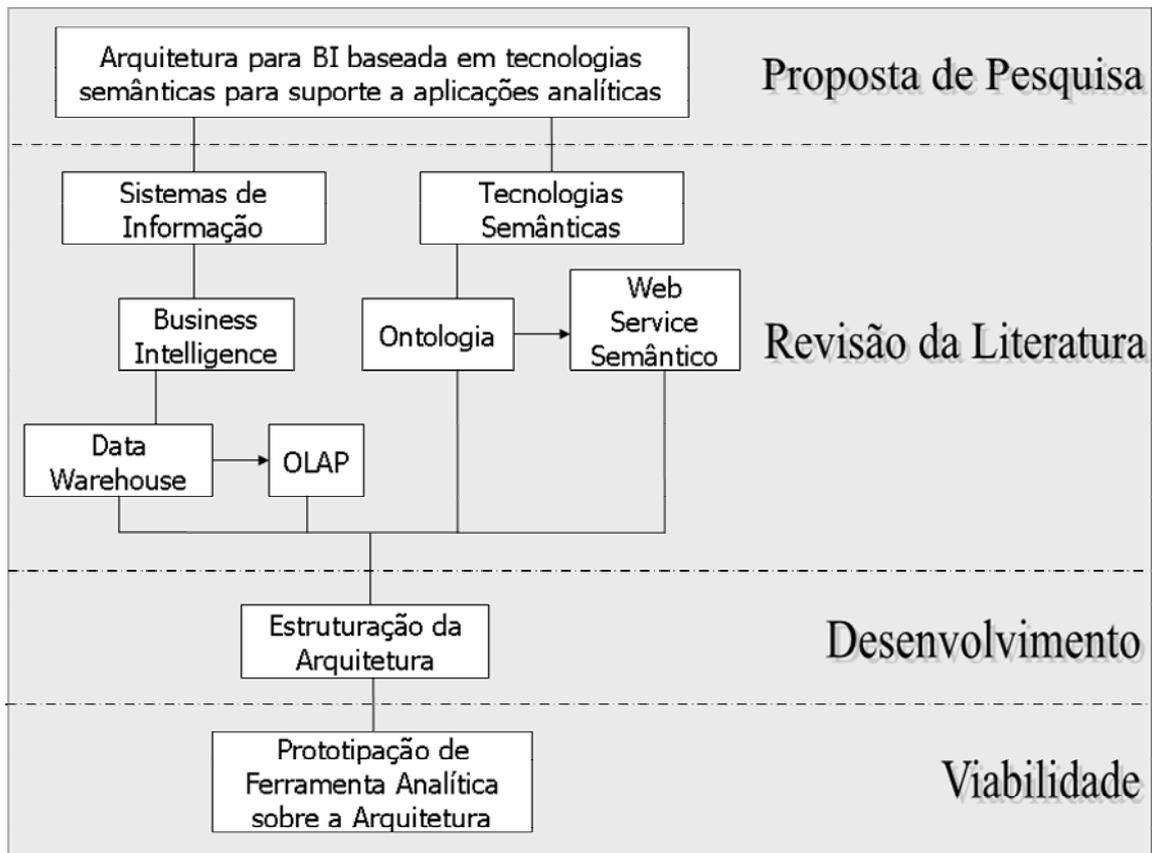


Figura 1 - Representação da metodologia utilizada no trabalho

1.8 Estrutura do Trabalho

O presente capítulo traz uma visão geral deste trabalho, sua motivação, a justificativa, as hipóteses da pesquisa, os objetivos, a metodologia e a estrutura.

No segundo capítulo, é apresentada uma breve descrição sobre a evolução dos sistemas de informação e os principais conceitos referentes à BI, com ênfase no papel das aplicações analíticas no processo de tomada de decisão das organizações. Além disso, é apresentada uma visão geral sobre as regras que norteiam o desenvolvimento de aplicações analíticas e as principais limitações das soluções correntes.

No terceiro capítulo, são explicitados aspectos teóricos a respeito de tecnologias semânticas, sendo apresentada uma introdução sobre *Web semântica* e sobre as principais tecnologias relacionadas ao contexto deste

trabalho: ontologias e *Web services* semânticos. São exibidas ainda as pesquisas relacionadas na presente tese.

No capítulo quatro, são identificados os requisitos para arquiteturas de BI *vis-à-vis* as demandas das organizações na sociedade do conhecimento e as deficiências das soluções atuais. Apresenta-se a arquitetura proposta neste trabalho, descrevendo-se a forma de organização de seus módulos e relacionando-a com os conceitos de BI e com tecnologias semânticas abordados nos capítulos anteriores. É exibido um quadro comparativo entre a solução proposta, as pesquisas acadêmicas relacionadas e as soluções comerciais de BI.

No capítulo cinco, é apresentada uma ferramenta analítica implementada sobre a arquitetura SBI. Demonstra-se através de um estudo de caso a viabilidade da solução proposta, identificando-se como a arquitetura atende aos requisitos funcionais mostrados no capítulo quatro.

No capítulo seis, são efetuadas as conclusões finais sobre a pesquisa e apresentados os trabalhos futuros. Por fim, relacionam-se as referências bibliográficas e os apêndices.

2 BUSINESS INTELLIGENCE

2.1 Introdução

A arquitetura proposta neste trabalho é uma alternativa para o desenvolvimento de soluções de BI alinhadas à semântica e às especificidades do negócio. Assim, neste capítulo é apresentada uma introdução sobre os principais conceitos relacionados a BI, resgatando inicialmente a evolução de sistemas de informação e suas modalidades. Em seguida são abordados os principais componentes de soluções de BI, com ênfase sobre as ferramentas analíticas de apoio à decisão.

O objetivo é introduzir os elementos teóricos relacionados às soluções de BI, com destaque para a forma como essas soluções são projetadas, incluindo uma descrição dos principais problemas encontrados nas soluções atuais.

2.2 Sistemas de Informação

Segundo Laudon e Laudon (1998, p. 21), “um sistema de informação pode ser tecnicamente definido como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena, e distribui informação para a tomada de decisão em uma organização”. Stair (1998, p. 6) define *sistema de informação* como “um conjunto de elementos ou componentes que interagem para se atingir objetivos, apresentando entradas, mecanismos de processamento, saídas e *feedback*”.

O processamento básico realizado por um sistema de informação é baseado em três etapas: coleta de dados, processamento e disponibilização da informação (DAVENPORT; PRUSAK, 1999; FELICIANO NETO; SHIMIZU, 1996). A informação obtida no processamento é utilizada então para a geração de conhecimento.

Segundo McGee e Prusak (1994) e Gil (1995), dado pode ser entendido como a matéria-prima com que o sistema de informações irá trabalhar, e informação é o produto final do sistema de informação e deve ser apresentada em forma, prazo e conteúdo adequados ao usuário. Os dados são transformados em informação através de um processo de análise, no qual

informações transformam-se em conhecimento, e esse conhecimento, através de um processo de síntese, leva ao julgamento (FELICIANO NETO; SHIMIZU, 1996; O'BRIEN, 2003; WALTZ, 2003). Enquanto a produção do conhecimento surge de um processo de transmutação da informação, o conhecimento por si próprio é explicitado, transformado em informação, para que possa ser processado e produzir novo conhecimento. Esta inovação propicia a formação de um ciclo virtuoso de formação de conhecimento (UNESCO, 2005).

2.2.1 Classificação de Sistemas de Informação

Os principais tipos de sistemas de informação são apresentados na Figura 2. Cada nível contempla as necessidades de conhecimento de um determinado público-alvo (LAUDON; LAUDON, 1998).

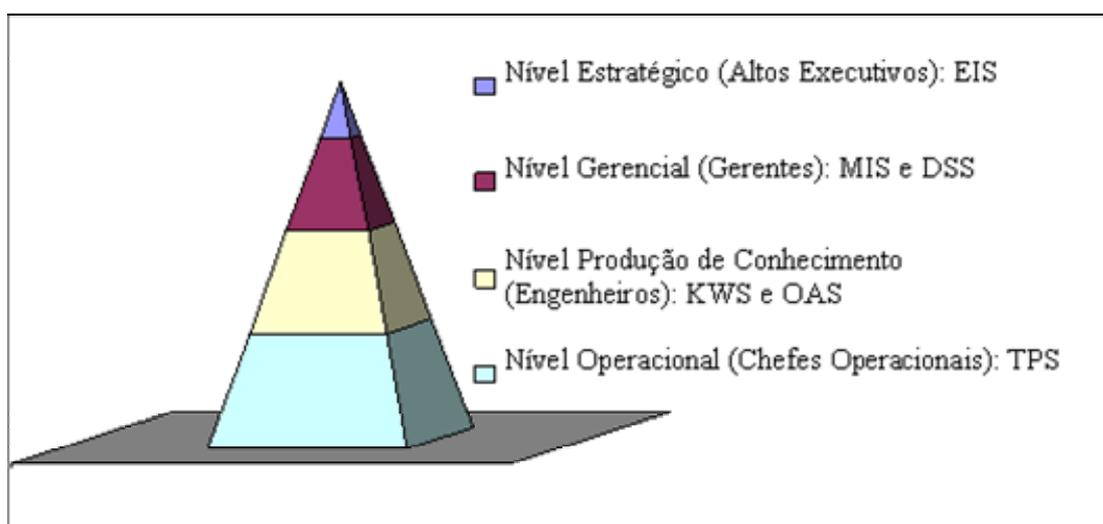


Figura 2 - Níveis do Conhecimento

Os sistemas de informação do nível operacional são utilizados para o controle do fluxo das atividades básicas da organização, enquanto os do nível de produção de conhecimento apóiam o processo de coleta e armazenamento de novo conhecimento associado ao negócio, para a administração da continuidade das tarefas cotidianas. Os sistemas de informação do nível gerencial são utilizados periodicamente no monitoramento, no controle e na tomada de decisões pelos gerentes de nível médio da organização. Finalmente, os sistemas do nível estratégico são utilizados pelos executivos para a realização do planejamento estratégico, pois proporcionam a visão

necessária da empresa para o planejamento das próximas ações em face do ambiente externo (LAUDON; LAUDON, 1998; TURBAN; ARAONSON, 2000).

De acordo com os níveis organizacionais, pode-se classificar os seis principais tipos de sistemas de informação como mostra o Quadro 1 (CRUZ, 1996; HOLSAPPLE; WHINSTON, 1996; LAUDON; LAUDON, 1998; O'BRIEN, 2003).

Quadro 1 - Características dos Sistemas de Informação

SISTEMA	ENTRADAS	PROCESSAMENTO	SAÍDA	USUÁRIOS
ESS	Dados agregados, externos e internos.	Gráficos, simulações, interações.	Projeções, tabelas.	Executivos.
DSS	Baixo volume de dados ou grande volume otimizado para análises.	Interações, simulações, análises.	Relatórios especiais, análises de decisão, tabelas.	Gerentes, equipe da gerência.
MIS	Sumários de dados, grande volume de dados.	Relatórios rotineiros, análises de baixo nível.	Sumários e relatórios de exceções.	Gerentes médios.
KWS	Base de conhecimento.	Modelagens e simulações.	Modelos, gráficos.	Equipe de administração de conhecimento.
OAS	Documentos, cronogramas.	Gerência de documentos e cronogramas.	Documentos, cronogramas e cartas.	Equipe de administração de conhecimento.
TPS	Transações, eventos.	Sorteio, listagem, agregação e atualização.	Relatórios detalhados, listagens e sumários.	Equipe operacional e supervisores.

Fonte: Adaptado de: Laudon e Laudon (1998).

Com a evolução da tecnologia utilizada pelos sistemas de informação, tornou-se possível oferecer mais velocidade para o processamento das informações e apoio ao processo decisório (INMON, 1997). O surgimento da técnica de *data warehousing* e das soluções de BI permitiu a criação de sistemas de informação que possibilitam à organização um maior controle dos processos internos e a definição e o acompanhamento de estratégias de gestão dos seus negócios.

2.3 Soluções de Business Intelligence

2.3.1 Definição

Segundo Côrtes (2002), *Business Intelligence* é um conjunto de conceitos e metodologias que visa ao apoio à tomada de decisões nos negócios a partir da transformação do dado em informação e da informação em conhecimento. Para Almeida et al. (1999), BI objetiva usar os dados da organização para apoiar decisões bem informadas, facilitando o acesso e a análise de dados, e possibilitando a descoberta de novas oportunidades.

De acordo com Sharma e Gupta (2004) e Simmers (2004), implantações bem-sucedidas de soluções de BI provêm uma visão integrada do negócio, estendem as capacidades analíticas dos usuários e impulsionam a formação de expertise nas organizações. O foco de soluções de BI é facilitar o entendimento do negócio das organizações, fornecendo a todos os níveis das organizações informações relevantes sobre suas operações internas e o ambiente externo, incluindo clientes e competidores, parceiros e fornecedores (MOSS; ATRE, 2003; WALTZ, 2003). O ambiente externo inclui ainda variáveis independentes que possam impactar no negócio, como tecnologia, leis e economia mundial, entre outros (BROHMAN et al., 2000). Um sumário dos objetos de análise, objetivos e das formas de utilização de acordo com o ambiente é apresentado no Quadro 2.

A evolução das soluções de BI está relacionada com a evolução do papel dos sistemas de informação nas organizações (GANGADHARAN; SWAMI, 2004; SIMMERS, 2004). Inicialmente, nos anos 70 e até meados dos anos 80, soluções de processamento e impressão de relatórios em lote dominavam a cena do processo de apoio à decisão. Os usuários tinham então que trabalhar sobre extensos relatórios para extrair elementos básicos de informação. Com a proliferação dos terminais de acesso aos *mainframes*, o acesso aos relatórios digitais foi disseminado, mas o acesso à informação era dificultado devido à complexidade dos sistemas da época.

A segunda fase dos sistemas de apoio à decisão é marcada pelo surgimento do *Data warehouse* (DW), repositórios de dados integrados e preparados para o apoio à decisão, que em conjunto com a evolução das

ferramentas analíticas ofereceu performance e poder analítico para o nível tático e executivo nas organizações (INMON, 2002; KIMBALL et al., 1998).

Quadro 2 - Sumário das características de utilização de soluções de BI

FOCO	AMBIENTE INTERNO	AMBIENTE EXTERNO
Objetos de Análise	<ul style="list-style-type: none"> - Operações do negócio. - Cadeia de suprimentos. - Gestão de relacionamento com os clientes. - Clientes e fornecedores. 	<ul style="list-style-type: none"> - Segmentação, preferências e comportamento dos clientes. - Economia. - Aspectos regulatórios. - Concorrência: <ul style="list-style-type: none"> • segmentação; • líderes. - Perfil de compra.
Objetivos	Eficiência.	Posicionamento no mercado.
Utilização	Análise, refinamento e reengenharia do desempenho do mercado.	Modelagem e previsão do comportamento do mercado. Posicionamento no mercado. Aprendizagem das tendências de consumo. Identificação de riscos, tecnologias e regulação.

Fonte: Adaptado de: Waltz (2003).

A terceira fase corresponde ao surgimento do BI. Segundo Almeida et al. (1999), o foco do DW estava muito orientado à tecnologia de consolidação dos dados. Ainda segundo os autores, as vantagens de projetos de BI em relação aos de DW são:

- soluções de BI não são orientadas unicamente à aplicação de tecnologia de informação de última geração, mas também ao fornecimento de soluções que integram pacotes verticais de aplicativos e metodologias para diversos segmentos de negócio;
- o foco das soluções de BI está no acesso e na distribuição de informação para o apoio à decisão; e
- soluções de BI suportam o acesso a todos os dados da organização, estruturados e não estruturados, e não somente aos armazenados no DW.

A seguir são introduzidos os principais componentes de soluções de BI.

2.3.2 Componentes de uma Arquitetura Típica de BI

A arquitetura típica de soluções de BI possui três componentes, conforme ilustrado na Figura 3. São os componentes (CÔRTEZ, 2002; RASMUSSEN et al., 2002; WG SYSTEMS, 2003):

- **ETL** (Extração, Transformação e Carga de Dados): conjunto de aplicativos e ferramentas que fazem a coleta de dados nos repositórios da organização, procedem com a limpeza e transformação para enfim carregar o DW (BRACKET, 1996; INMON, 1997; KIMBALL et al., 1998);
- **data warehouse**: repositório de dados integrado e não-volátil onde são armazenados os dados transformados pelo módulo ETL. Esse repositório deverá suportar as demandas analíticas das ferramentas de apoio à decisão e os aplicativos de extração de conhecimento (INMON, 1997; KIMBALL et al., 1998; SELL; PACHECO, 2001);
- **área de apresentação**: diz respeito ao conjunto de instrumentos que serão utilizados pelos usuários na organização para navegar no DW. Esses instrumentos correspondem a relatórios previamente configurados, aplicativos para confecção de relatórios, ferramentas OLAP (*On-line Analytical Processing*), ferramentas de *Data Mining* (mineração de dados), entre outras (BERRY; LINOFF, 1997; BERSON, 1997; GONZAGA; 2005; THOMSEN, 2002).

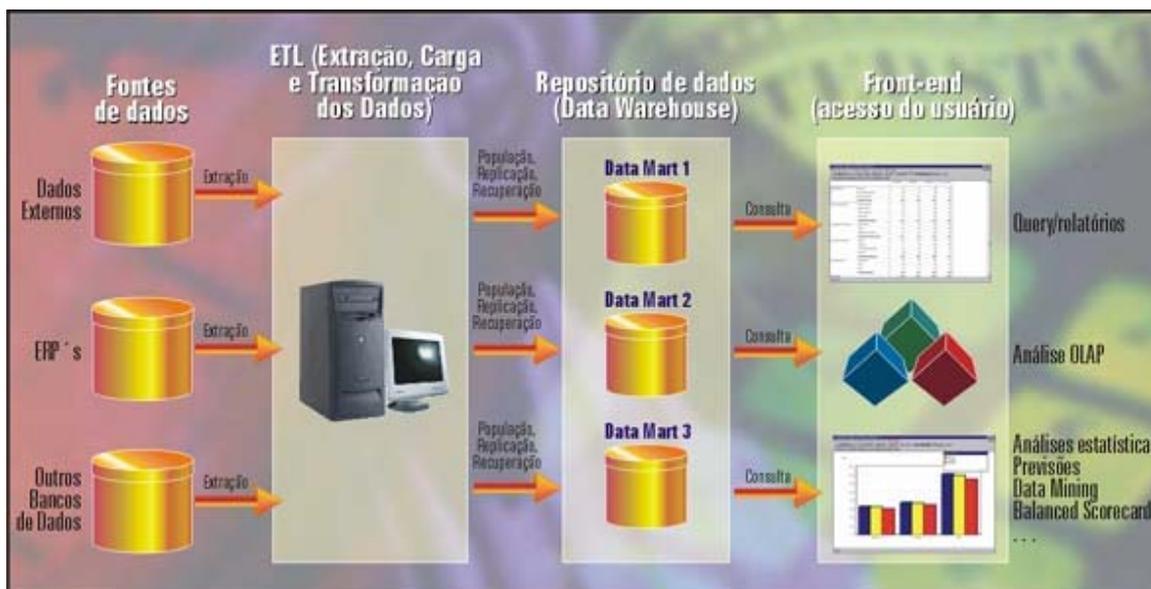


Figura 3 – Arquitetura típica de uma solução de Business Intelligence
 Fonte: WG SYSTEMS, 2003.

A seguir, são introduzidos os módulos de arquiteturas de BI que estão relacionados com a arquitetura proposta neste trabalho. Desse modo, são introduzidos apenas os conceitos afetos ao projeto do DW e sobre as aplicações OLAP.

2.4 Data Warehouse

2.4.1 Definições, objetivos e características

Segundo Inmon (1997, p. 33), DW é “um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais”. Para uma análise detalhada dessa definição, vide Sell (2001).

Para responder aos questionamentos dos usuários em um DW, parte-se do pressuposto de que o acesso aos dados seja efetuado sobre uma (ou várias) base de dados consolidada (CHUCK et al., 1998; COREY et al., 1998). Tais dados são consolidados pelas ferramentas ETL no repositório de dados integrado (BRACKETT, 1996; DEBEVOISE, 1999). Esse repositório deve ser modelado para permitir que consultas diversas sejam processadas com alto desempenho pelas ferramentas analíticas. A seguir, são apresentados brevemente alguns preceitos que guiam a modelagem de dados em um DW.

2.4.2 Modelagem de Dados em um Data Warehouse

A primeira fase no projeto do repositório é a definição do modelo de dados. Nesta fase, são analisados os dados necessários e que podem ser obtidos junto aos sistemas transacionais. É então criado um modelo de dados corporativo, que difere do modelo corporativo operacional à medida que são retirados os dados que são utilizados apenas no ambiente operacional e incorporados elementos de tempo nas chaves, além de haver o agrupamento dos dados conforme sua utilização (CHUCK et al., 1998; KIMBALL; ROSS, 2002; SELL, 2001).

Além do imprescindível aspecto do desempenho, o período em que os dados ficam armazenados no banco de dados deve ser levado em conta pelo modelo de dados do DW. Em um sistema operacional, o dado é atualizado e normalmente não mantém um registro histórico das mudanças. Já em um DW, o dado normalmente não é atualizado. As modificações de atributos são armazenadas em registros separados (DEBEVOISE, 1999; DONALD; 1997).

Deve-se considerar também a legibilidade do modelo de dados. Não existem ferramentas gráficas que expõem um modelo entidade-relacionamento ao usuário para que este possa montar suas pesquisas customizadas, devido à enorme confusão que um emaranhado de tabelas propicia. Por esses motivos, deve-se considerar a adoção de um modelo mais adequado à realidade de um DW (DONALD, 1997; KIMBALL et al., 1998; KIMBALL; ROSS, 2002).

Uma alternativa ao modelo relacional é o modelo dimensional (GIOVINAZZO, 2000; KELLY, 2000; KIMBALL et al., 1998; KIMBALL; ROSS, 2002). Esse modelo lembra a idéia do cubo contendo três ou mais dimensões, cada uma representando um atributo diferente conforme apresenta a Figura 4.

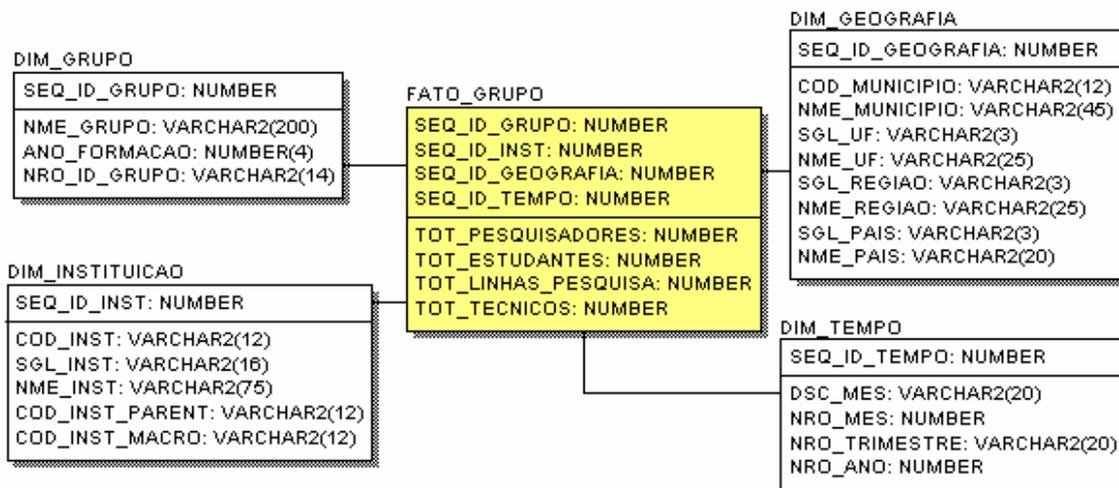


Figura 4 - Exemplo de modelo-estrela

No exemplo ilustrado na Figura 4, tomam-se as dimensões descritivas de instituições de ensino, de grupos de pesquisa, do local onde a instituição está situada e da data. Essas dimensões são ligadas a uma tabela de fatos. A tabela de fato *fato_grupo* representa a quantidade de pesquisadores, estudantes e técnicos que atuam em cada grupo de pesquisa, além da quantidade de linhas de pesquisa desses grupos em um dado mês. O usuário pode “girar” o cubo para obter diferentes perspectivas sobre os fatos a partir das quatro dimensões ligadas à tabela de fatos.

Conforme descrito no exemplo anterior, um modelo dimensional é constituído de uma ou várias *tabelas de fato*, que conterão as *medidas* do negócio a serem analisadas, e, em torno, várias tabelas descritivas, chamadas *tabelas de dimensão*. Cada tabela de dimensão possui uma única ligação com a tabela de fatos, a qual é feita através de chaves externas (CHUCK et al., 1998; KELLY, 2000; KIMBALL; ROSS, 2002). Uma descrição comparativa entre fatos, dimensões e medidas é apresentada no Quadro 3.

O modelo dimensional, além de agilizar o processamento das consultas, permite uma melhor visualização dos dados devido à forma simples de organizá-los e propicia mais flexibilidade para eventuais ajustes no modelo (KIMBALL; ROSS, 2002).

Na segunda fase da modelagem é construído o modelo físico, onde são incluídas características físicas e chaves. Tendo-se definido os atributos que figurarão nas tabelas de fatos e nas dimensões, resta ainda decidir sobre

importantes aspectos de organização dos dados e performance, representados inicialmente pela definição da granularidade, do particionamento e da definição de agregados (DONALD, 1997; KIMBALL et al., 1998; KIMBALL; ROSS, 2002).

Quadro 3 - Características de fato, dimensão e medida de um DW

	FATOS	DIMENSÕES	MEDIDAS
Escopo	Representam um item, uma transação ou um evento de negócio.	Determinam o contexto de um assunto de negócios, como, por exemplo, uma análise da produtividade dos grupos de pesquisa.	São os atributos numéricos que representam um fato e são determinados pela combinação das dimensões que participam desse fato.
Objetivo	Refletem a evolução dos negócios.	São os balizadores de análise de dados.	Representam o desempenho de um indicador de negócios relativo às dimensões que participam de um fato.
Tipo de Dado	São representados por conjuntos de valores numéricos (medidas) que variam ao longo do tempo.	Normalmente não possuem atributos numéricos, pois são somente descritivas e classificatórias dos elementos que participam de um fato.	Podem possuir uma hierarquia de composição de seu valor.

Fonte: Adaptado de: Machado (2000).

2.5 Área de Apresentação

O valor de uma solução de BI pode ser medido sob o ponto de vista das informações que possam ser extraídas (ALMEIDA et al., 1999). Entre os tipos de ferramentas que podem ser utilizadas para interagir com o DW, destacam-se as ferramentas OLAP (HARRISON, 1997; KIMBALL et al., 1998; MACHADO, 2000; THOMSEN, 2002) e as ferramentas de *Data Mining* (BERRY; LINOFF, 1997; BERSON, 1997).

Segundo Thomsen (2002), ferramentas OLAP são ferramentas de caráter descritivo, ou seja, permitem a apresentação dos dados do negócio sob diferentes perspectivas e implementam funcionalidades de agregação, razão e produto, entre outras. Ferramentas de *Data Mining*, por outro lado, são ferramentas de caráter explicativo e implementam funcionalidades de

regressão, extração de associações e agrupamentos, entre outras (BERRY; LINOFF, 1997; BERSON, 1997).

Ambos os tipos de ferramentas normalmente contam com o apoio de algum tipo de metadado. Esse metadado, entretanto, normalmente restringe-se à descrição sintática do modelo de dados do DW e a configurações específicas dessas ferramentas, como a definição de hierarquias e dimensões dos cubos em ferramentas OLAP e a definição dos parâmetros de entrada para os algoritmos de *Data Mining* (ALENQUER, 2002; THOMSEN, 2002).

A arquitetura proposta neste trabalho pretende agregar às arquiteturas de BI descrições semânticas que transcendem as definições sintáticas tradicionais. Essa proposta pretende incorporar ontologias para a definição da lógica e dos conceitos do negócio a ser suportado e da associação dessa semântica à representação também semântica das fontes de dados e dos serviços de interesse ao tomador de decisão.

Como a arquitetura proposta pelo autor visa suportar principalmente aplicações analíticas, é apresentada a seguir uma introdução sobre os conceitos de aplicações OLAP.

2.6 OLAP

2.6.1 Definição

Segundo Harrison (1997), o rótulo OLAP (*On-Line Analytical Processing* ou *Processamento Analítico On-Line*) foi reservado para análises multidimensionais. As consultas simples e os relatórios representam as facilidades mais simples proporcionadas pelo OLAP. Tipos mais sofisticados de consultas permitem que o usuário “navegue” sobre enormes quantidades de dados por níveis diferentes de resumo dos dados até chegar ao nível de detalhe encontrado nos sistemas transacionais, ou ainda possibilitam combinar dados na mesma consulta para a detecção de relações entre dados de forma multidimensional.

Como o objetivo das ferramentas OLAP é permitir análises para analistas de negócios, elas têm requisitos bem particulares se comparadas a outros sistemas de informação (BERSON, 1997; THOMSEN, 2002):

- **flexibilidade.** Os analistas de negócio devem ter liberdade e facilidade para escolher os dados a serem analisados bem como o formato no qual eles devem ser visualizados;
- **simplicidade.** Ferramentas OLAP devem prover maneiras simples e intuitivas de confecção de análises para os analistas de negócio;
- **expressividade.** A linguagem de consulta utilizada pela ferramenta OLAP deve ser poderosa o suficiente para que o analista consiga extrair informações realmente úteis para o negócio;
- **poder de análise.** Ferramentas OLAP devem prever um conjunto de operações de agregação e exploração de dados que permitam análises de tendências e comparações complexas utilizando os dados que descrevem o negócio;
- **velocidade.** O processamento das consultas sobre o DW deve ser realizado em um tempo relativamente baixo. O tempo de processamento, dependendo do volume de dados e da complexidade do código da consulta pode em casos específicos chegar a horas.

Um dos desafios no projeto de aplicações OLAP se refere a como aliar a simplicidade para o usuário à expressividade nas consultas e análises. Parte da resposta é obtida através da maneira como os dados são organizados no modelo dimensional (DONALD, 1997; KIMBALL; ROSS, 2002). Como descrito anteriormente, uma das características desse modelo em relação aos modelos normalizados é a legibilidade. Porém, nas soluções atuais, usuários ainda assim precisam lidar com descrições técnicas do modelo de dados. Para definir suas análises, os usuários deverão entender como os dados estão organizados no modelo dimensional.

Através do modelo dimensional, o usuário tem liberdade para visualizar diversos aspectos do negócio sob a perspectiva que entenda ser a mais interessante para sua análise (MACHADO, 2000). A amplitude de análise desse modelo reside no fato de que um grande número de aspectos do

negócio pode ser expresso em termos desses dois conceitos (BERSON, 1997; KIMBALL et al., 1998).

Entretanto, alguns desafios no contexto de ferramentas OLAP permanecem sem solução, entre eles a falta de possibilidade de se representar a lógica do negócio nas soluções atuais e a dificuldade de extensão das funcionalidades exploratórias. Esses desafios e a forma como o autor propõe abordar esses problemas são apresentados em detalhes no Capítulo 4.

Em relação às arquiteturas utilizadas para OLAP, destacam-se o ROLAP (*Relational On-Line Analytical Processing*) baseado em banco relacionais e o MOLAP (*Multidimensional On-Line Analytical Processing*) baseado em bancos multidimensionais. Variações dessas arquiteturas incluem DOLAP (*Desktop On-Line Analytical Processing*) e HOLAP (*Hybrid On-Line Analytical Processing*) (HARRISON, 1997; MACHADO, 2000; THOMSEN, 2002). Vantagens e desvantagens de cada modelo são discutidas em Thomsen (2002).

2.6.2 Características do processamento OLAP

O termo OLAP foi criado pelo Codd (1995), fundador do modelo de dados relacional. Logo após a publicação desse artigo, OLAP se tornou um dos termos da moda no domínio dos sistemas de informação. Além de cunhar o termo, Codd definiu 12 princípios básicos a serem atendidos por aplicações OLAP. Esses 12 princípios ajudaram a definir as aplicações OLAP existentes na atualidade (DONALD, 1997; THOMSEN, 2002).

Thomsen (2002, p 668) dividiu as 12 regras em 3 categorias: básica, relato e dimensional. Essas categorias foram criadas para facilitar o entendimento das regras de Codd, que, segundo Thomsen, confundiam por misturar aspectos lógicos com aspectos físicos. As categorias e as respectivas regras de Codd são apresentadas nos Quadros 4, 5 e 6, incluindo o número de característica original identificada por Codd.

A definição das 12 regras de Codd, apesar de despertar divergências na comunidade de desenvolvimento sobre a obrigatoriedade dessas regras e sobre a necessidade de inclusão de novas regras, gerou a publicidade necessária para a disseminação do termo.

No entanto, na visão deste autor, o atendimento dessas regras significa apenas que a ferramenta respeita um conjunto de características compatíveis com a tecnologia dominada pelos fabricantes OLAP. Busca-se nesta pesquisa lançar luz sobre tecnologias pouco exploradas pela indústria e pela academia nesse contexto, as tecnologias semânticas. Espera-se que este estudo subsidie no futuro uma mudança na maneira que as 12 regras básicas serão interpretadas.

Quadro 4 - Características Básicas

CARACTERÍSTICA	DESCRIÇÃO
10: manipulação de dados intuitiva	Permitir a ação direta sobre as células e os recursos de interação gráficos.
3: acessibilidade	Apresentar uma visão lógica única dos dados da empresa.
5: arquitetura cliente/servidor	Permitir a execução de operações OLAP sobre um servidor a partir de uma ferramenta-cliente.
2: transparência	Possibilidade de embutir as funcionalidades OLAP dentro do contexto de uma aplicação conhecida pelo usuário final.
8: suporte para multiusuário	Suportar operações de leitura e escrita simultâneas, observando-se critérios de integridade e segurança.

Fonte: Adaptado de: Thomsen (2002).

Quadro 5 - Características de Relato

CARACTERÍSTICA	DESCRIÇÃO
11: relato flexível	Apresentar dimensões de um cubo através de diferentes combinações de linhas, colunas e páginas.
4: desempenho coerente do relato	Preservar o desempenho da apresentação à medida que o volume de dados e dimensões aumente.
7: tratamento dinâmico de matriz esparsa	A organização física dos dados deve ser sensível à mudança das características dos dados ao longo do tempo.

Fonte: Adaptado de: Thomsen (2002).

Quadro 6 - Características Dimensionais

CARACTERÍSTICA	DESCRIÇÃO
1: visão conceitual multidimensional	A visão conceitual do usuário analista deve possibilitar a análise do negócio sobre diferentes perspectivas.
6: dimensionalidade genérica	Cada dimensão deve ser equivalente tanto em sua estrutura quanto nas capacidades exploratórias aplicáveis.
12: dimensões e níveis de agregação ilimitados	A estrutura criada no servidor deverá possibilitar a criação de tantas dimensões e cubos quanto necessário para se analisar o negócio.
9: operações irrestritas de dimensão cruzada	Permitir o cálculo e a manipulação de dados através de qualquer quantidade de dimensões e relacionamento de dados.

Fonte: Adaptado de: Thomsen (2002).

A seguir são introduzidas as funcionalidades exploratórias tradicionalmente implementadas pelas ferramentas OLAP.

2.6.3 Funcionalidades exploratórias

O desenvolvimento de ferramentas OLAP foi motivado, em princípio, pela necessidade de se propiciar ao usuário final uma visão multidimensional do seu negócio. Através dessas ferramentas, o usuário define as informações que deseja mensurar (medidas de um fato) e as perspectivas pelas quais quer analisar essas mensurações (dimensões), formando um cubo multidimensional. A partir da formatação do cubo, o usuário passa então a manipulá-lo através de uma série de funcionalidades analíticas que são disponibilizadas pela aplicação OLAP (BERSON, 1997; HARRISON, 1997).

Gonzaga (2005) apresenta um sumário das principais funcionalidades disponibilizadas pelas ferramentas OLAP. Esse sumário é mostrado a seguir no Quadro 7.

Quadro 7 - Funcionalidades OLAP

FUNCIONALIDADE	DESCRIÇÃO	EXEMPLO
Drill Down	Aumentar o nível de detalhe da informação e, conseqüentemente, diminuir o grau de granularidade. Normalmente caracterizado pela adição de colunas ao cabeçalho de	Uma análise de vendas por Estado é alterada para uma incluir a faixa etária dos consumidores dentro desses Estados.

FUNCIONALIDADE	DESCRIÇÃO	EXEMPLO
	uma consulta (KIMBALL et al., 1998).	
Drill Up	Diminuir o nível de detalhe e aumentar o grau de granularidade. Normalmente caracterizado pela retirada de colunas do cabeçalho de uma consulta (KIMBALL et al., 1998).	Uma análise de vendas por faixa etária e Estado é alterada para apresentar apenas os totais gerais por faixa etária.
Slice	Cortar o cubo, mas manter a mesma perspectiva de visualização dos dados. Funciona como um filtro, restringindo uma dimensão a somente algum ou alguns de seus valores.	Em uma dimensão tempo de um modelo, é selecionado somente o ano de 2000.
Dice	Mudar a perspectiva da visão multidimensional como se o cubo fosse girado. Permite descobrir comportamentos e tendências entre os valores das medidas analisadas, em diversas perspectivas.	A análise é alterada de região (linha) por ano (coluna) para ano (linha) por região (coluna).
Drill Across	Alterar o nível de análise dentro de uma mesma dimensão, pulando um nível intermediário dentro de uma mesma dimensão.	O nível da análise é alterado direto de ano para mês dentro de uma dimensão tempo, quando esta é composta de ano, semestre e mês.
Drill Through	Ocorre quando o usuário passa de uma informação contida em uma dimensão para uma outra.	O usuário está realizando uma análise na dimensão de tempo e no próximo passo analisa a informação por região.
Drill Out	É o detalhamento para informações externas, como fotos, som, arquivos-texto, tabelas.	O usuário seleciona um cliente ou fornecedor para visualizar um resumo sobre seus dados.
Sort	Tem a função de ordenar a informação, podendo ser aplicada a qualquer tipo de informação, não somente a valores numéricos.	Ordenar as instituições em ordem alfabética.
Ranking	Permite agrupar resultados numéricos por ordem de tamanho, refletindo somente na apresentação do resultado e não no resultado em si.	Ordenar a relação de filiais de acordo com os maiores volumes de vendas.
Paging	Apresentação dos resultados de uma consulta em várias páginas, permitindo a navegação do usuário.	Permitir um máximo de 10 resultados por página.
Filtering	Apresentação de consultas com restrições sobre atributos ou fatos, ou seja, restringe um fato a somente	Em um fato Vendas, apenas os valores das vendas a prazo são apresentados.

FUNCIONALIDADE	DESCRIÇÃO	EXEMPLO
	algum ou alguns de seus valores.	
Alerts	Utilizados para indicar situações de destaque em elementos dos relatórios, baseados em condições envolvendo objetos e variáveis.	Definir que os valores das vendas mensais inferiores a R\$ 50.000,00 devem aparecer com destaque em vermelho.
Break	Permite separar o resultado de uma análise em grupos de informações, possibilitando também a subtotalização de valores para cada grupo.	O usuário tem a necessidade de visualizar a informação por cidades, então ele solicita um <i>break</i> . O relatório será automaticamente agrupado por cidades, somando os valores mensuráveis por cidades.

Fonte: Adaptado de: Gonzaga (2005).

Além das funcionalidades apresentadas no Quadro 7, são disponibilizadas em algumas ferramentas funcionalidades estatísticas e outras de caráter descritivo. Thomsen (2002, p. 19) cita como exemplos de funções descritivas as funções de agregação, de alocação, razões, produto, etc.

As funcionalidades descritas anteriormente não são em sua plenitude implementadas por todas as ferramentas OLAP. Na verdade, o portfólio de funcionalidades varia de acordo com o fabricante. Alguns fabricantes de soluções OLAP permitem a adição de novas funcionalidades, mas através da utilização de APIs implementadas em linguagens específicas (CA MAGAZINE, 2004, 2005). De maneira geral, as funcionalidades implementadas pelos fabricantes incluem *slice* e *dice*, além de algumas variações de *drill*, principalmente *drill down*, *drill up* e *drill across*. Essas operações são baseadas na estruturação dos dados e dos relacionamentos do DW ou em metadados proprietários, como descrito na Seção a seguir.

2.6.4 Metadados para o suporte de funcionalidades analíticas

A maneira como as funcionalidades OLAP são implementadas depende da arquitetura selecionada, mas, de maneira geral, essas funcionalidades baseiam-se nas hierarquias de campos nas dimensões e dos relacionamentos entre dimensões e fatos. Assim, para permitir um *drill through*, por exemplo, uma ferramenta baseada em ROLAP verifica quais dimensões compartilham um fato através das referências de chaves estrangeiras das dimensões na

tabela de fatos (HARRISON, 1997; KIMBALL et al., 1998; KIMBALL; ROSS, 2002).

Diferentes formas de descrever as hierarquias em dimensões são implementadas pelos fabricantes OLAP para o suporte das funcionalidades *drill down* e *drill up*. Segundo Thomsen (2002, p. 228), as formas mais comuns para a definição de vínculos hierárquicos são através de definição de tabelas pai-filho e a tabela de nível.

Uma tabela pai-filho possui uma coluna para identificar o atributo-pai e outra para identificar o filho em uma hierarquia, conforme ilustrado na Figura 5. A definição da estrutura hierárquica da dimensão é realizada em termos de valores de linhas na tabela pai-filho. Pode haver variações no vínculo pai-filho, conforme exigido pelas ferramentas específicas. Entretanto, segundo Thomsen (2002), relacionamentos pai-filho isolados não definem níveis semânticos. Assim, não seria possível representar que janeiro, fevereiro e março são filhos do trimestre um e que possuem, além disso, relacionamento com dias.

Uma tabela de nível mantém a definição hierárquica em colunas separadas, cada uma definindo um nível na hierarquia (THOMSEN, 2002). Assim, é possível percorrer a hierarquia na quantidade de níveis desejada, mas torna-se difícil incluir ou remover níveis da hierarquia. Vale ressaltar que esse tipo de estrutura também não identifica relações semânticas entre os níveis.

Além dos metadados proprietários mantidos pelo fabricante, destaca-se uma iniciativa para definição de um metadado padrão para soluções de BI, o CWM (*Common Warehouse Metamodel*). CWM é um padrão proposto pelo *Object Management Group* (OMG) para intercâmbio de metadados no ambiente BI (OMG, 2000). Através da especificação sintática proposta pelo CWM, metadados específicos dos componentes de soluções de BI poderão ser representados e integrados ao ambiente, facilitando a integração entre tecnologias desenvolvidas por diferentes fabricantes (CHANG, 2000).

Apesar de se tratar de um avanço no que tange à integração de soluções de diferentes fabricantes em um ambiente de BI, o CWM não vem sendo aplicado para apoiar o funcionamento dos componentes de arquiteturas

de BI. No momento, cada fabricante tem desenvolvido o seu próprio conjunto de metadados sintáticos.

A solução que o autor propõe não se assemelha ao CWM, pois o autor está mais preocupado em como integrar definições semânticas do negócio, dos dados e dos serviços relacionados ao negócio, visando apoiar as funcionalidades analíticas, e não apoiar o intercâmbio entre aplicações, propósito do CWM. Soluções de metadados semânticos que abrangem a representação dos conceitos e da lógica do negócio, além dos dados e serviços relacionados, conforme descrito no Capítulo 4, são virtualmente inexistentes no contexto de soluções de BI.

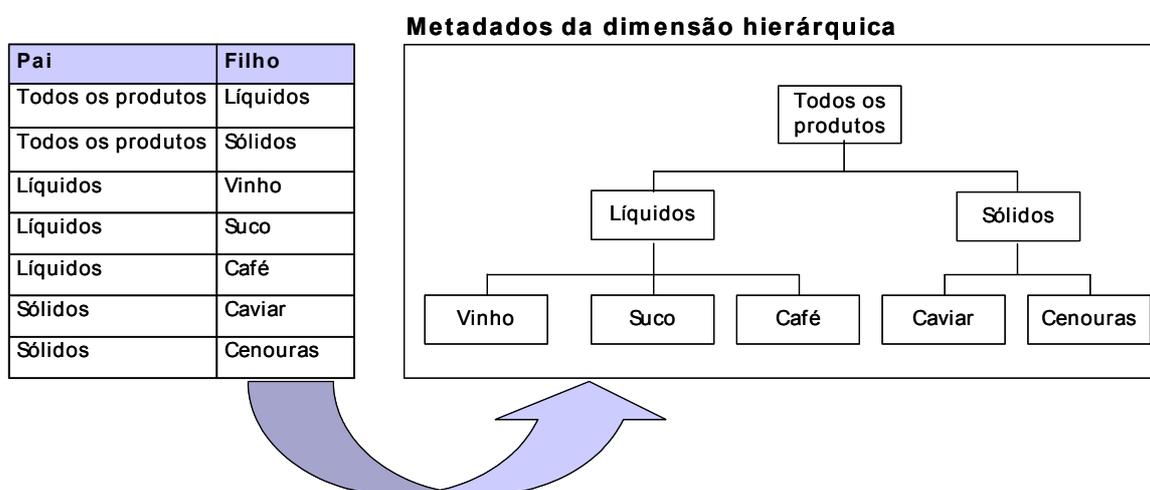


Figura 5 - Exemplo de estrutura pai-filho como metadado para a definição de hierarquias em uma dimensão de produtos
Fonte: Adaptado de: Thomsen (2002).

2.7 A Metodologia BUS para Projetos de BI

Entre as metodologias existentes para a consecução de projetos de BI, destaca-se a metodologia BUS, proposta por Kimball et al. (1998). Esta metodologia prevê um conjunto de etapas executadas em cascata e em espiral, conforme ilustra a Figura 6, visando possibilitar um desenvolvimento interativo e incremental da solução de BI. Segundo (SELL, 2001), esta perspectiva incremental de desenvolvimento tornou esta metodologia popular entre projetistas de BI, pois torna possível oferecer aos usuários resultados em um espaço de tempo inferior do que as metodologias que pregam um processo monolítico de desenvolvimento, como a proposta de Inmon (2002). Uma visão geral das etapas e dos principais papéis a serem desempenhados pelos

membros da equipe de desenvolvimento em cada etapa é apresentada no Quadro 8 (KIMBALL et al., 1998).

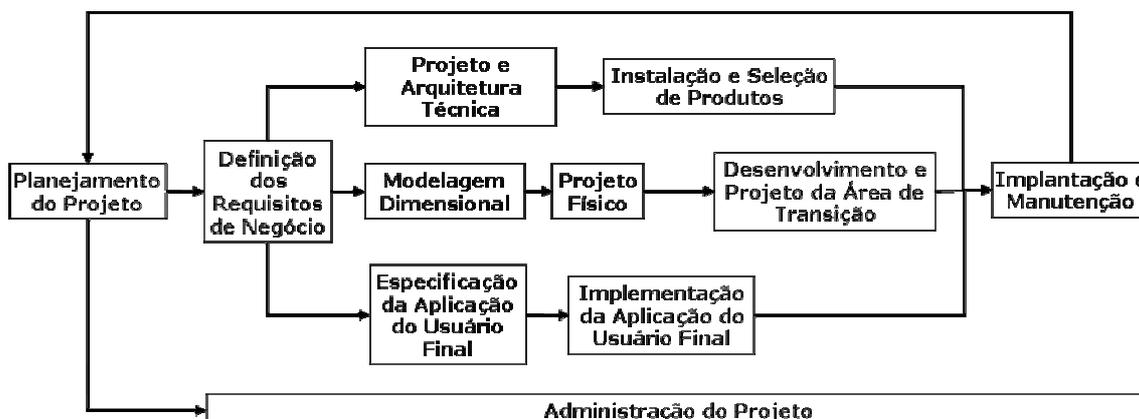


Figura 6 - A Metodologia BUS

Quadro 8 - Sumário das Etapas e Papéis Associados na Metodologia BUS

Fonte: adaptado de Kimball et al. (1998)

Etapa	Descrição	Papéis Associados
Planejamento do Projeto	Definição do Escopo	Gerente de Projeto
Definição dos Requisitos de Negócio	Análise dos processos da organização e levantamento de requisitos analíticos	Gerente de Projeto, Analista de Negócios e Modelador de Dados
Administração do Projeto	Controle da Execução	Gerente de Projeto
Projeto e Arquitetura Técnica	Definição da estratégia de desenvolvimento do <i>back room</i>	Arquiteto de Segurança e Analista de Suporte
Instalação e Seleção de Produtos	Seleção de ferramentas para apoiar o controle e o desenvolvimento do projeto	Gerente de Projeto, Arquiteto de Segurança, Analista de Suporte, Desenvolvedor de Front End, Projetista da Área de Transição, DBA e Modelador de Dados
Modelagem Dimensional	Confecção do modelo lógico do DW	Analista de Negócios, Modelador de Dados, DBA e Projetista da Área de Transição
Projeto Físico	Confecção do modelo físico do DW	Modelador de Dados e DBA
Desenvolvimento da Área de Transição	Projeto e desenvolvimento da área de estágio	Projetista da Área de Transição, Desenvolvedor da Área de Transição, DBA e Analista de Qualidade
Implantação e Manutenção	Implantação, treinamento dos usuários e ajustes	Educador, Gerente de Projeto, Arquiteto de Segurança, Analista de Suporte, Desenvolvedor de Front End, Projetista da Área de Transição,

		DBA e Modelador de Dados
Especificação da Aplicação do Usuário Final	Planejamento da área de apresentação de dados	Desenvolvedor de Front End
Implementação da Aplicação do Usuário Final	Implementação da área de apresentação de dados	Desenvolvedor de Front End e Analista de Qualidade

2.8 Limitações das Soluções Correntes de BI

No tocante às soluções comerciais de BI, verificam-se limitações relacionadas à falta de suporte para a representação dos conceitos e da lógica do negócio para o suporte de inferências mais sofisticadas do que as operações genéricas de *drill* oferecidas atualmente aos usuários dessas soluções. Alguns fabricantes tradicionais de soluções de BI introduzem um nível de flexibilidade adicional através de metadados proprietários (CA MAGAZINE, 2004, 2005). Conforme descrito na Seção anterior, normalmente são adotados esquemas para a representação de relacionamentos entre atributos para o suporte de *drill*, como através de tabelas pai-filho e de tabelas de nível (THOMSEN, 2002).

No entanto, os metadados introduzidos pelos fabricantes de soluções de BI restringem-se a uma representação sintática dos relacionamentos existentes entre estruturas de dados. Essa representação sintática não suporta inferências como a recomendação de recursos (*i.e.*, regras, serviços e dados) relacionados ao objeto da análise. Além disso, a lógica das funcionalidades exploratórias é embutida nas suas implementações, desconsiderando-se especificidades do domínio do negócio no comportamento dessas funcionalidades.

As ferramentas analíticas atuais não oferecem maneiras escaláveis para a agregação de novas funcionalidades exploratórias ou modificação das existentes. Frequentemente, usuários devem desenvolver a extensão requerida sem possibilidade de reutilização de código já existente que poderia ser utilizado. De maneira geral, as ferramentas analíticas oferecem a possibilidade de se agregarem novas funcionalidades, mas através de uma linguagem de programação específica e do uso de uma API disponibilizada pelo fabricante (CA MAGAZINE, 2004, 2005).

Alguns fabricantes de soluções de BI, como Business Objects, Hyperion Solutions, Informatica, SAS Institute e MicroStrategy, expuseram parte de suas APIs através de *Web services*, visando facilitar a chamada de suas operações analíticas por aplicações de terceiros. Entretanto, não se verifica suporte para o caminho oposto, ou seja, a chamada de aplicações de terceiros pelas soluções de BI (HURWITZ GROUP, 2002).

Recentemente, fabricantes de ferramentas OLAP lideradas pela Microsoft começaram a aderir à gramática XMLA (*XML for Analysis*) (XMLA, 2004) para auxiliar no intercâmbio de dados entre aplicações analíticas e para a exposição de funcionalidades OLAP através de *Web services*. Entretanto, apesar de essa iniciativa representar um avanço para a integração de dados e serviços no contexto de aplicações analíticas, é necessário lidar com o problema da interpretação do contexto de uma análise e da semântica dos dados manipulados nessa análise. XML limita-se à descrição sintática de dados, não suportando nenhum tipo de inferência semântica. Da mesma forma, como a descrição das funcionalidades de um *Web service* é restrita à representação sintática de suas entradas e saídas, torna-se inviável a localização de um serviço de acordo com suas funcionalidades.

2.9 Considerações sobre o Capítulo

As soluções de BI surgiram como uma evolução dos sistemas de informação. Neste capítulo foram abordados os conceitos básicos acerca de sistemas de informações, sua classificação e adequação aos níveis organizacionais. Sobre cada nível organizacional poderá ser aplicado pelo menos um tipo de sistema de informação, preparado para auxiliar na realização das tarefas pertinentes ao nível. Soluções de BI suportam principalmente os níveis tático e estratégico nas organizações, apoiando a transformação de dados em informação útil para a tomada de decisão.

A vitrine de uma solução de BI é a área de apresentação. Através das ferramentas disponibilizadas na área de apresentação, usuários da organização poderão explorar os dados integrados no DW. Ferramentas OLAP permitem que usuários possam ter uma perspectiva multidimensional dos dados corporativos. As funcionalidades analíticas providas pelas ferramentas

OLAP permitem a agregação dos dados, a navegação em diferentes níveis de detalhes e a realização de análises de tendências ou outras comparações complexas utilizando os dados que descrevem o negócio.

Entretanto, conforme descrito no capítulo introdutório deste trabalho, soluções de BI carecem de suporte para a representação dos conceitos e da lógica do negócio e de poder de inferência sobre essa descrição semântica para apoiar operações analíticas. Virtualmente, nenhuma solução de BI provê ferramentas analíticas que permitam, por exemplo, a representação de regras de negócio e a utilização dessas regras em operações de *drill*.

As funcionalidades exploratórias são suportadas por metadados proprietários, que se limitam em descrever sintaticamente as fontes de informação e possuem participação limitada na execução das ferramentas. Em outras palavras, a lógica das funcionalidades exploratórias é embutida na sua implementação, desconsiderando-se especificidades do domínio do negócio. Essa abordagem limita as possibilidades de assistência ao usuário no momento da exploração. Recursos de inferência poderiam oferecer, por exemplo, informações de maneira pró-ativa ou alternativas de análise para o usuário de acordo com o contexto de suas explorações. Dessa forma, ferramentas analíticas podem ser caracterizadas como meras coadjuvantes no processo decisório; o processo de análise é totalmente dependente dos *insights* do tomador de decisão, do seu conhecimento sobre o modelo de dados e do seu conhecimento sobre as funcionalidades da ferramenta analítica.

Outro fator que limita a inserção das ferramentas analíticas no processo decisório é a dificuldade de incorporação de novas funcionalidades ao conjunto predefinido pelos fabricantes. As soluções de customização, quando oferecidas, não são escaláveis, pois exigem um conhecimento sobre uma linguagem específica que em muitos casos não é dominada pelas equipes de tecnologia de informação das organizações e por vezes não permite o reuso de aplicações já existentes.

A arquitetura SBI, introduzida no Capítulo 4, se propõe a contribuir para a resolução desses problemas através de uma abordagem baseada em

tecnologias semânticas. No próximo capítulo, são introduzidas as tecnologias semânticas aplicadas no desenvolvimento da arquitetura proposta.

3 TECNOLOGIAS SEMÂNTICAS

3.1 Introdução

Na abordagem proposta nesta pesquisa, utilizam-se tecnologias que vêm sendo desenvolvidas no contexto da *Web* semântica para tornar soluções de BI mais flexíveis e alinhadas às necessidades dos tomadores de decisão.

Dessa forma, neste capítulo é apresentada uma introdução sobre tecnologias da *Web* semântica aplicadas no desenvolvimento da arquitetura proposta. São exibidos os principais conceitos relacionados a ontologias, sua classificação, os formalismos de representação e uma breve introdução sobre OCML, o formalismo adotado para a representação e o suporte a inferências sobre as ontologias da arquitetura.

Neste capítulo são ainda resgatados os conceitos relacionados a *Web services*, utilizados nesta tese como uma das estratégias para oferecer flexibilidade na incorporação de novas funcionalidades nas aplicações analíticas. Além disso, é apresentada uma introdução sobre *Web services* semânticos, abordagem utilizada nesta pesquisa para possibilitar a localização e recomendação automática de serviços de acordo com o contexto de uma análise em ferramentas analíticas.

3.2 Web Semântica

3.2.1 Definição

Na arquitetura apresentada no Capítulo 4, são aplicadas tecnologias relacionadas à *Web* semântica. *Web* semântica foi um termo cunhado por Tim Berners-Lee, o pai da *Web*, para denominar uma nova versão da *Web*, em que a semântica é adicionada para a descrição do conteúdo não estruturado da *Web* atual. Essa semântica tornará possível que o conteúdo na *Web* seja processável pela máquina. Essa visão foi tornada pública em um artigo escrito pelo próprio Berners-Lee em conjunto com James Hendler e Ora Lassila na *Scientific American* (BERNERS-LEE et al., 2001).

A *Web* semântica é motivada pelos problemas encontrados na *Web* atual. Antoniou e Van Harmelen (2004) destacam, entre esses problemas, a baixa precisão nas buscas, a qual pode resultar em um número muito alto de

documentos não relacionados ao objetivo da busca ou à não-recuperação de qualquer documento. Isso ocorre porque não há suporte nos mecanismos de busca mais populares para a análise do significado dos documentos e das palavras-chave entradas na busca.

Daconta et al. (2003), Davies et al., (2003) e Studer (2000) apontam ainda outras necessidades das organizações que não estão apenas relacionadas às buscas na *Web*, entre elas, o grande volume de dados existente nos sistemas de informação dessas organizações, grande quantidade de componentes de software isolados e dificuldade em integrar conteúdo de diferentes fontes de informação. Sobre os cenários de aplicação dentro das organizações, Fensel et al. (2002b), Daconta et al. (2003) e Gómez-Pérez et al. (2004) incluem, entre outros, suporte a vendas, marketing, suporte à decisão, administração, gestão de conhecimento, gestão de competências.

3.2.2 Camadas da Web Semântica

Berners-Lee et al. (2001) propuseram uma estrutura em camadas para a *Web* semântica, conforme ilustrada na Figura 7. Essa divisão visa permitir o desenvolvimento em etapas, em que cada camada é apoiada pelas camadas inferiores.

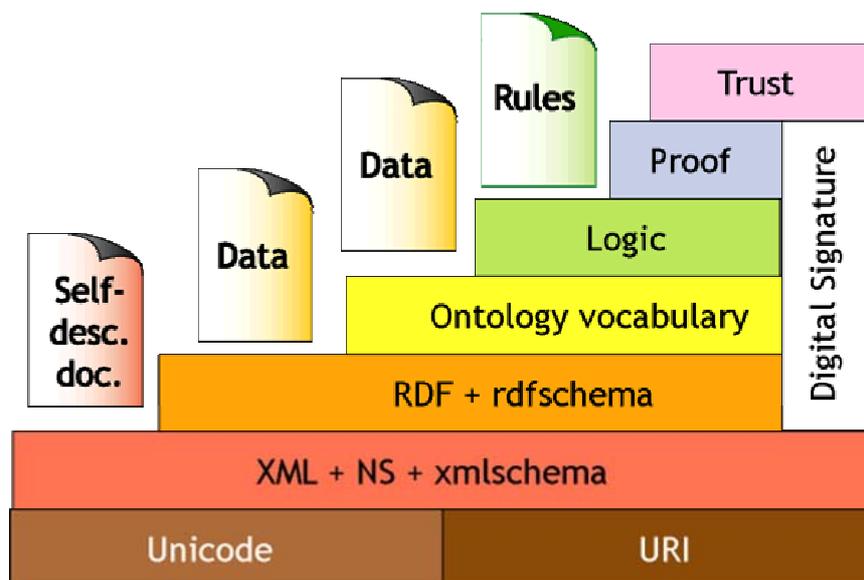


Figura 7 - Estruturação das camadas da *Web* semântica
Fonte: Berners-Lee et al. (2001).

No nível inferior encontram-se as referências para troca de conteúdo (Unicode) (UNICODE, 2004) e para a localização de conteúdo (URI) (W3,

2004). Logo acima se encontra o XML (*eXtended Markup Language*) (W3C, 1996), uma linguagem que permite a definição da estrutura do conteúdo. O XML *Schema* suporta a definição de regras de validação sobre documentos XML (W3C, 2001), formando a base para a criação de vocabulários.

RDF (*Resource Description Framework*) (W3C, 1999) é um modelo de dados baseado em triplas (objeto, predicado e valor). O modelo de triplas permite que sejam feitas descrições sobre conteúdo independentemente da estrutura desse conteúdo. O RDF *Schema* (W3C, 2004) compreende primitivas para a organização de hierarquias e para a definição de restrições sobre RDF. O RDF *Schema* é na verdade um formalismo para a estruturação de ontologias mais simples. Entretanto, para o suporte de inferências mais poderosas, são necessários formalismos mais expressivos para a representação de ontologias, os quais devem incluir suporte para axiomas e outras formas de relacionamentos entre os conceitos (FENSEL, 2001; GÓMEZ-PÉREZ et al., 2004).

A camada *Logic* (lógica) é utilizada para estender a representação ontológica, permitindo a declaração de conhecimento através de regras de produção ou lógica de predicado para suportar inferências (ANTONIOU; VAN HARMELEN, 2004; BRACHMAN; LEVESQUE, 2004). A camada *Proof* (prova) compreende a representação de evidências e o suporte dedutivo para a validação das assertivas feitas nas camadas inferiores (BERNERS-LEE, 2001). Finalmente, a camada *Trust* surgirá através do uso de assinaturas digitais e outras formas de representação de recomendações de confiança por agentes ou organizações de certificação (DACONTA et al., 2003; FENSEL et al., 2002a).

Na presente pesquisa, aplicam-se ontologias para a representação do domínio suportado pela arquitetura (conceitos e regras de negócio) e para a descrição dos dados da organização. Ontologias são aplicadas ainda na descrição dos serviços de interesse à organização, de forma a possibilitar a criação de *Web services* semânticos. A próxima Seção apresenta uma introdução sobre ontologias.

3.3 Ontologia

3.3.1 Definição

Ontologia, segundo um ponto de vista filosófico, pode ser conceituada como o que pode “existir” ou “ser” e que pode ser entendido pelos seres humanos (GAMPER et al., 1999). No contexto da inteligência artificial, o termo tem sido utilizado para se referir a modelos de conhecimento, em que objetos, seus atributos e relacionamentos são especificados para a resolução de um problema específico em um dado domínio (GAMPER et al., 1999; SHUM et al., 2000).

Uma das conceituações mais citadas no domínio da inteligência artificial é a de Gruber (1993), que afirma que ontologia é uma especificação de uma conceituação. Guarino (1998) acrescenta que uma ontologia é uma especificação parcial e explícita de um determinado domínio, isentando-se de representar conceitos que estejam fora desse domínio. Outras definições podem ser encontradas em Motta (1999) e em Van Heijst et al. (1997).

Do ponto de vista da aplicação de ontologias, Mena et al. (2000) descrevem que o principal objetivo de uma ontologia é tornar um conteúdo explícito, independentemente da maneira como o dado foi estruturado, de qual plataforma de armazenamento foi utilizada e do tipo de conteúdo (literal, gráfico, numérico). Ontologia constitui tratados entre provedores de dados em que essas representações semânticas descrevem e mapeiam o dado armazenado por esses provedores (MENA et al., 2000; MISSIKOFF, 2002).

Uma ontologia caracteriza um domínio através da representação dos seguintes conceitos (FENSEL, 2001; GÓMEZ-PÉREZ et al., 2004; GRUBER, 1993):

- **classes** no domínio de interesse;
- **instâncias** (indivíduos) das classes representadas;
- **relações** para representar os tipos de interações entre classes e entre indivíduos;

- **funções** para a representação de casos especiais de relações, em que o enésimo elemento de um relacionamento é único para o elemento predecessor (e.g., é-mãe-de);
- **axiomas** para a modelagem de sentenças que são sempre verdadeiras. Um axioma é uma afirmação lógica que não pode ser comprovada a partir de outras afirmações, mas que pode servir para a construção de sistemas e para a formulação de teorias (como a teoria de conjuntos) (BRACHMAN; LEVESQUE, 2004). São aplicados na definição de restrições sobre relações e classes.

3.3.2 Níveis de Representação

Guarino e Welty (2000) propõem uma classificação para ontologias baseada no nível de detalhe da representação de um dado domínio, conforme apresentado a seguir.

- **Vocabulário:** consiste de uma lista de termos e respectivas definições. Corresponde a uma definição em XML *Schema*, por exemplo.
- **Taxonomia:** definição de hierarquias sobre os termos do vocabulário. Cada termo em uma taxonomia pode estar relacionado a uma ou mais relações do tipo “pai-filho”.
- **Sistema relacional:** prevê relacionamentos associativos além dos hierárquicos entre os termos do vocabulário.
- **Teoria axiomática:** além dos relacionamentos, suporta a definição de regras de inferência e de restrições através de axiomas.

Os vocabulários são de uso limitado e podem auxiliar na padronização da troca de dados entre as partes, mas não permitem que as máquinas raciocinem a respeito do significado de um item de informação (CASTOLDI, 2003; FENSEL, 2001).

As taxonomias introduzem um nível adicional de formalização ao vocabulário, definindo um elo hierárquico entre os termos, mas não

expressando qual é o significado desse elo. Esse relacionamento hierárquico pode expressar relações do tipo “é um”, “parte de”, “tópico mais abrangente que”, dependendo ainda da direção tomada para analisar essa hierarquia (DACONTA et al., 2003; GÓMEZ-PÉREZ et al., 2004).

O nível Sistema Relacional prevê a adição de relacionamentos sem detalhamento de restrições, possibilitando a navegação sobre os termos representados na ontologia através de relacionamentos diversos. Esses relacionamentos estruturam os termos em grafos, determinando como os nós (termos) conectam-se entre eles através de arestas (relacionamentos) e indicando se existe ou não direção na relação entre os nós. Esse nível, apesar de introduzir um nível de descrição superior ao das taxonomias, ainda possui descrição semântica limitada (FENSEL et al., 2002a; GÓMEZ-PÉREZ et al., 2004).

O último nível introduz axiomas à representação do conteúdo. A representação e restrição dos termos e seu relacionamento permite que a máquina possa, de maneira restrita, fazer inferências sobre o conhecimento representado de maneira similar ao homem (em circunstâncias comparáveis). Nesse caso, os axiomas tentam limitar os modelos formais de interpretação (semântica) da representação do conhecimento para um conjunto de significados que o engenheiro da ontologia pretendeu (BRACHMAN; LEVESQUE, 2004; DACONTA, 2003).

À medida que introduz mais recursos para formalização do conhecimento, cada nível introduz também maior complexidade para a definição inicial e para a manutenção da ontologia (BRACHMAN; LEVESQUE, 1985).

O suporte a inferências introduzido pelo nível da Teoria Axiomática é desejável na implementação da arquitetura proposta, para, por exemplo, recomendar serviços que estejam associados ao contexto de uma análise. Serviços (*i.e.*, *Web services* semânticos) são recomendados automaticamente ao usuário se todos os valores para as entradas desses serviços puderem ser inferidos a partir dos conceitos utilizados na análise e se todas as precondições desses serviços puderem ser satisfeitas. Uma introdução sobre *Web services* semânticos é apresentada nas seções seguintes, e o processo de

representação e inferência aqui descrito é discutido no Capítulo 4. A seguir são introduzidos os tipos de ontologias.

3.3.3 Tipos de Ontologias

De acordo com Gómez-Pérez e Benjamins (1999), as ontologias podem ser classificadas em cinco categorias: ontologia genérica, ontologia de tarefa, ontologia de método, ontologia do domínio e ontologia da aplicação. Cada tipo de ontologia compreende um conjunto de classes, relações, funções e axiomas. A seguir, descreve-se de forma sucinta essas cinco categorias.

- **Ontologia Genérica:** define conceitos básicos que não estão relacionados a um domínio específico e que servirão de suporte para todas as demais ontologias (BORGIO et al., 2002).
- **Ontologia de Domínio:** representa o conhecimento, as atividades, as teorias e os princípios básicos que governam um dado domínio (GUARINO, 1998).
- **Ontologia de Tarefa:** descreve um conjunto de tarefas que não necessariamente estão relacionadas ao mesmo domínio (MIZOGUCHI et al., 1995; MOTTA, 1999).
- **Ontologia de Método:** descreve o processo de raciocínio necessário para se executar uma tarefa (CHANDRASEKARAN, 1999).
- **Ontologia de Aplicação:** contém o conhecimento necessário para modelar uma aplicação (MOTTA, 1999; VAN HEIJST et al., 1997).

No desenvolvimento da arquitetura proposta, definiu-se um conjunto de ontologias de domínio que suportam a descrição do negócio, a ligação dos conceitos do negócio com os respectivos repositórios e a descrição dos serviços úteis para o suporte do processo decisório. A arquitetura SBI prevê ainda a integração de um *framework* para WSS (*i.e.*, IRS-III) para suportar a descoberta, a composição e a execução de WSS de acordo com o contexto de análise. Esse *framework* prevê a utilização de ontologias de domínio, tarefa, método e aplicação, conforme descrito na Seção 3.4.3.

3.3.4 Formalismos para Representação de Ontologias

Atualmente existem várias linguagens para representação de ontologias. Diversas linguagens são baseadas em *frames* ou em lógica, ou mesmo em ambos. De forma geral, a linguagem utilizada para representar uma ontologia determinará que tipo de inferência é permitida (FENSEL, 2001; GÓMEZ-PÉREZ; BENJAMINS, 1999; GÓMEZ-PÉREZ et al., 2004).

Uma linguagem baseada em *frames* utilizará estruturas de classes e *slots* para a representação de um domínio (BRACHMAN; LEVESQUE, 1985). *Frames* suportam ainda a definição de taxonomias sobre as classes. Entre as linguagens baseadas em *frames* vale citar a Ontolingua (FARQUHAR et al., 1997), um sistema desenvolvido para a representação de ontologias que permite a tradução dessas para diversas linguagens. Um dos maiores problemas das linguagens baseadas em *frames* é a falta de uma semântica bem definida (BECHHOFFER et al., 2001). Outras linguagens baseadas em *frames* como FLogic (KIFER et al., 1995) e OCML (MOTTA, 1999) introduzem uma semântica definida em lógica de primeira ordem.

Linguagens baseadas em lógica, tais como *Description Logics* (DL) (GROSOF et al., 2003; GRUBER, 1993), são propostas como alternativas às baseadas em *frames*. DLs descrevem conhecimento em termos de conceitos e relações que são usadas para a derivação automática de taxonomias. Em DL, conceitos podem ser definidos tanto quanto através de assertivas como da união ou intersecção de outros conceitos. Essas assertivas são então utilizadas por mecanismos de raciocínio para automaticamente gerar hierarquias de conceitos. Exemplos de linguagens baseadas em DL incluem LOOM (MACGREGOR, 1992) e OWL (OWL, 2001).

A seguir é apresentada uma breve introdução sobre OCML, focando a apresentação das ontologias que definem a sua semântica operacional. OCML é a ferramenta de representação utilizada para a definição das ontologias e para o suporte às inferências na arquitetura proposta nesta pesquisa.

3.3.5 OCML

OCML (*Operational Conceptual Modelling Language*) foi originalmente desenvolvida no contexto do projeto VITAL (DOMINGUE et al., 1993). Seu

projeto originalmente visava prover à linguagem Ontolingua a capacidade de provar teoremas e avaliação de funções. Trata-se de uma linguagem baseada em *frames*, com semântica definida em lógica de primeira ordem com uma sintaxe padrão Lisp (GÓMEZ-PÉREZ et al. 2004; MOTTA, 1999).

OCML provê primitivas para a definição de classes, relações, funções, axiomas, instâncias e regras (com suporte para encadeamento para frente e para trás), além de suportar a definição de código procedural anexável às demais definições (MOTTA, 1999). OCML permite a representação de bases de conhecimento (FENSEL; VAN HARMELEN, 1994; NEWELL, 1982), além de suportar a inclusão de assertivas e de exploração do conhecimento através das funcionalidades *Tell* e *Ask*, respectivamente (GÓMEZ-PÉREZ et al. 2004).

OCML possui uma biblioteca composta de 12 ontologias, as quais são sumarizadas por Rajpathak (2004), como descrito no Quadro 9. Para uma descrição completa da linguagem OCML e respectivas ontologias, vide Motta (1999).

Quadro 9 - Ontologias que definem a semântica operacional para a linguagem OCML

ONTOLOGIA	DESCRIÇÃO
Meta	Ontologia que define os conceitos necessários para a linguagem, como expressões, funções, regras, relações, entre outros.
Functions	Define os conceitos associados com a especificação de funções, como domínio, faixa e relações unárias e binárias.
Relations	Compreende os conceitos associados com a especificação de relações, como o universo e a extensão de uma relação.
Sets	Define os construtores necessários para a definição de conjuntos, como união, intersecção e cardinalidade.
Números	Compreende os conceitos utilizados para descrever operações matemáticas.
Lists	Define conceitos usados para representar e manipular listas.
Strings	Define os conceitos e as operações associados com <i>strings</i> .
Mapping	Descreve os conceitos necessários para a especificação de mecanismos de mapeamento entre conceitos.
Frames	Define os conceitos associados com construtores para a representação de <i>frames</i> , incluindo a definição de classe e instância, além da definição de funções como <i>direct-instances</i> e <i>all-slot-values</i> e da definição de relações como <i>has-one</i> e <i>has-at-</i>

ONTOLOGIA	DESCRIÇÃO
	<i>most.</i>
Inferences	Suporta todos os mecanismos de inferências usados para definir funções e relações.
Environment	Provê operadores utilizados para suportar a execução de procedimentos e regras.
Task-method	Provê uma ontologia para a especificação de <i>Tasks</i> e <i>Problem-solving methods</i> (MOTTA, 1999).

Fonte: Adaptado de: Rajpathak (2004).

OCML tratava-se de um dos formalismos mais completos na época do início do desenvolvimento da arquitetura SBI (início de 2004). Os motivos que foram levados em consideração para a seleção dessa linguagem para a representação e suporte a inferências na implementação corrente da arquitetura incluem:

- **expressividade:** OCML permite a definição de classes, relações, funções, procedimentos, regras (com encadeamento para frente ou para trás) e axiomas (formalizados em primeira e segunda ordem), contando com uma semântica operacional bem-definida para todos os elementos. Axiomas podem ser aninhados às outras definições ou através de elementos independentes;
- **poder de inferência:** possui suporte para interpretação de funções, interpretação de controle e um mecanismo de prova. O mecanismo de prova suporta checagem de restrições, inferência de valores e processamento de heranças;
- **possibilidade de rápida prototipação e execução de modelos de conhecimento;**
- **possibilidade de integração com outras aplicações** através de APIs disponibilizadas pelos interpretadores Lisp.
- **existência de conversores para outros formalismos** que suportam a transformação de ontologias representadas em OCML para OWL e Ontolingua (ou vice-versa) (KMI, 2004).

3.4 *Web services* Semânticos

3.4.1 Definição e Motivações

Nesta pesquisa, são aplicados *Web services* semânticos para a extensão das funcionalidades analíticas providas por soluções de BI. Assim, nesta Seção são introduzidos os conceitos relacionados à *Web services* e *Web services* semânticos, incluindo uma descrição do *framework* utilizado para a criação e execução de *Web services* semânticos - IRS-III.

Web service é uma aplicação identificada por uma URI, cujas interfaces e ligações são descritas, localizadas e acessadas através de protocolos baseados em XML (W3C, 2002). Entre os objetivos dessa tecnologia encontram-se a reutilização de código e suporte à integração de aplicações heterogêneas (ALESSO; SMITH, 2005; W3C, 2002). Aplicações diversas se comunicam normalmente com *Web services* através de mensagens SOAP (SOAP, 2003) sobre o protocolo HTTP (W3C, 2000). Tecnologias como registros UDDI (UDDI, 2003) e ebXML (OASIS, 2004) permitem que aplicações recuperem informações sobre os *Web services* registrados. A sintaxe da assinatura dos métodos implementados pelos *Web services* é descrita por WSDL (WSDL, 2001).

À medida que os *Web services* proliferaram, eles se tornaram similares às páginas *Web* no sentido da dificuldade em se descobrir um serviço de acordo com as necessidades do usuário (MCILRAITH et al., 2001; MOTTA et al., 2003; SYCARA et al., 2003). Ocorre que a descrição feita em XML através dos arquivos WSDL não é expressiva o suficiente para permitir a inferência do que o *Web service* faz efetivamente. Dessa forma, tornam-se inviáveis a localização e a composição automática de *Web services* (DOMINGUE et al., 2004a; MARTIN et al., 2004; SYCARA et al., 2003).

Recentemente, pesquisadores começaram a investigar maneiras de permitir a descoberta, a composição e a execução de *Web services* através de descrições semânticas. Essa nova área – *Web services* semânticos (WSS) – visa proporcionar à *Web* semântica a possibilidade de se oferecer conteúdo dinâmico e identificável pela máquina (ALESSO; SMITH, 2005; MOTTA et al., 2003; SWSI, 2002; THE DAML SERVICES COALITION, 2002). *Web services*

semânticos buscam ainda prover formas de processamento e raciocínio sobre a representação estática provida pelas ontologias na *Web* semântica (GÓMEZ-PEREZ et al., 2004).

3.4.2 Frameworks para Web services Semânticos

Existem no momento duas iniciativas predominantes para a definição de ontologias para WSS: OWL-S (OWL-S, 2002) e WSMO (WSMO, 2004).

OWL-S utiliza a linguagem OWL para a representação de três tipos de conhecimento a respeito de um serviço: o que o serviço faz (*ServiceProfile*), como ele faz (*ServiceModel*) e como ele é acessado (*ServiceGrounding*) (MARTIN et al., 2004; OWL-S, 2002). OWL-S, por basear-se em tecnologias derivadas do popular padrão XML, possui como principal deficiência a falta de suporte para representação de regras, recurso necessário para a definição das precondições e pós-condições de um serviço. Essa deficiência é superada no futuro com a incorporação de regras através da iniciativa RuleML (RULEML, 2004).

WSMO (*Web Service Modeling Ontology*) descreve os vários aspectos relacionados à *Web services* semânticos baseando-se no *framework* WSMF (FENSEL; BUSSLER, 2002). Seus principais componentes são: *Goals*, *Web services*, *Ontologies* e *Mediators*. Esses são sumarizados a seguir (WSMO, 2004).

- *Goals* representam os tipos de objetivos que os usuários almejam atingir através de um *Web service*. A descrição de um *goal* compreende o estado desejado da memória de trabalho e o estado desejado no mundo real após a execução de um dado *Web service*. Um *goal* pode importar conceitos e relações definidos em uma ontologia de domínio qualquer.
- *Web services* corresponde a um conjunto de conceitos utilizados para descrever o comportamento de um *Web service* que pode atender aos requisitos de um *Goal*. Essa descrição compreende a definição da coreografia do *Web service* (*i.e.*, maneira que se pode comunicar com o *Web service*) e a definição da sua orquestração

(maneira como *Web services* podem ser executados quando compostos).

- *Ontologies* provêm a descrição dos conceitos utilizados para o apoio da representação dos demais componentes.
- *Mediators* descrevem os mecanismos pelos quais os componentes interoperam entre si e com o mundo exterior.

3.4.3 IRS-III

IRS-III (*Internet Reasoning Services*) é um *framework* e infra-estrutura para WSS que implementa as descrições WSMO, visando permitir a descrição, descoberta, composição e execução de WSS (DOMINGUE et al., 2004a). IRS-III possui quatro características que o distingue de outros *frameworks* para WSS (CABRAL et al, 2004), as quais foram consideradas no momento da seleção do *framework* WSS nesta pesquisa. Essas características são sumarizadas a seguir.

- Possui suporte para a transformação de código Java e Lisp em um *Web service*. Em seguida, o *Web service* criado pode ser anotado semanticamente gerando um WSS.
- Possibilidade de invocação de *Web services* através da seleção de *Goals*. Essa característica torna possível que os usuários simplesmente selecionem um objetivo a ser atingido (*Goal*), e a localização do *Web service* mais adequado ao atendimento daquele objetivo é feita automaticamente pelo IRS-III.
- Possibilidade de se reutilizar parte dos componentes do IRS-III para o atendimento de outras tarefas.
- IRS-III permite a descrição semântica de *Web services* já existentes, e WSS criados no IRS-III podem ser reconhecidos como *Web services* padrões.

A arquitetura do IRS-III (ilustrada na Figura 8) é formada principalmente pelos seguintes componentes: *Server*, *Publisher* e *Client*. O *Server* é baseado em uma implementação Lisp que dá suporte aos protocolos HTTP e SOAP (RIVA; RAMONI, 1996). O *Server* suporta ainda as inferências necessárias

para a invocação dos *Web services* e para execução dos *mediators* WSMO (DOMINGUE et al., 2004).

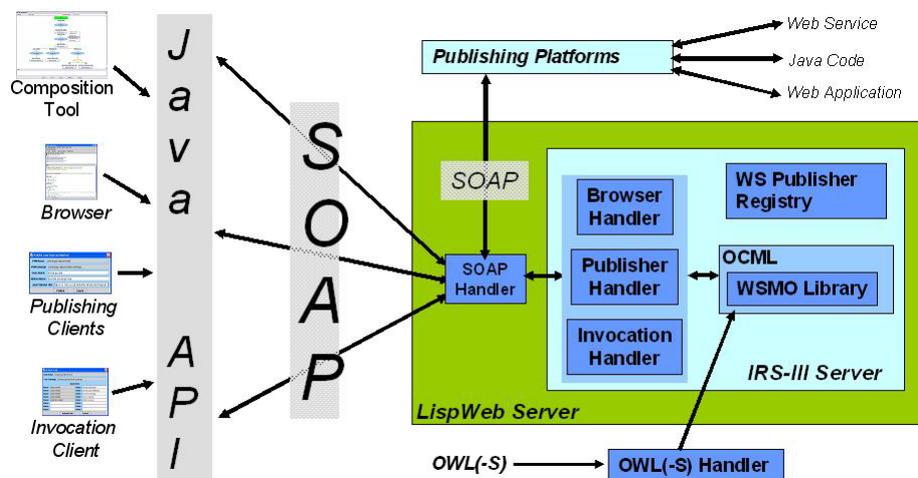


Figura 8 - Arquitetura do IRS-III
Fonte: Domingue et al. (2004a)

O *Publisher* liga um serviço implementado em Java, Lisp ou outro descrito através de um arquivo WSDL a uma descrição *Web service* WSMO, a qual compreende a descrição das interfaces, sua coreografia (GALIZIA e DOMINGUE, 2004) e orquestração (CONFALONIERi et al. 2004). Finalmente, o cliente permite a invocação de *Goals* e *Web services* através de uma interface gráfica ou de uma API Java (DOMINGUE et al., 2004b e 2004c).

Usuários podem empregar uma ferramenta que suporta composições semi-automáticas de *Web services* (SELL et al., 2004b). Essas composições são suportadas por um conjunto de descritores semânticos, discutidos em Hakimpour et al. (2005) e em Sell et al., (2004b).

Nesta pesquisa, IRS-III é utilizado para apoiar a descoberta de WSS que possam prover processamento de inferências sobre os dados da análise. Essa descoberta leva em consideração o contexto da análise definida pelo usuário na ferramenta analítica. IRS-III suporta ainda nesta pesquisa a composição de serviços para a realização de análises mais elaboradas que podem compreender mais de um WSS e operadores de controle de fluxo. Essa integração é descrita no capítulo seguinte.

3.5 Aplicação de Tecnologias Semânticas no Domínio de BI

As tecnologias relacionadas à *Web* semântica vêm sendo aplicadas em diferentes formas para suportar desafios tradicionais relacionados a sistemas de informação. Guarino (1998) enumera várias referências na literatura sobre a utilização de ontologias em áreas como engenharia do conhecimento, modelagem de banco de dados, modelagem e integração de informações, análise orientada a objetos, extração e recuperação de informações e modelagem de sistemas baseados em agentes. Além desses domínios, são verificados trabalhos na área da gestão da memória organizacional (VAN ELST; ABECKER, 2001) e de integração de serviços multilingües (AKAHANI et al., 2002; LEGER, 2000).

Em relação ao domínio das soluções de BI, verificam-se algumas referências relacionadas à aplicação de ontologias na integração semântica de dados, como descrito em alguns autores (MOTIK et al., 2002; HOFREITER, 2002; MAEDCHE et al., 2003; SURE et al., 2002; e FILETO et al., 2003). Basicamente, essas pesquisas propõem o mapeamento das fontes de dados através de um conjunto de ontologias de domínio para auxiliar o processo de integração de dados. Entretanto, verificam-se poucas referências relacionadas ao escopo desta pesquisa, isto é, o suporte de soluções de BI no que tange às suas aplicações analíticas, através de tecnologias da *Web* semântica.

No contexto da área de apresentação em soluções de BI, a maioria das referências e dos produtos comerciais disponíveis que são suportados por tecnologias semânticas visa prover buscas sobre conteúdo através de uma descrição semântica desse conteúdo. Nesse contexto, aplicações como Observer (MENA et al., 2000) e Tambis (PATON et al., 1999) suportam a busca sobre conteúdo distribuído em vários repositórios por meio da tradução das consultas em subconsultas que são então processadas sobre cada repositório. Sheth e Ramakrishnan (2003), Necib e Freitag (2003) e Bruckner et al. (2001) descrevem ainda como ontologias são aplicadas na redefinição de consultas, utilizando sinônimos e hipônimos dos termos informados na consulta para estender os resultados das consultas originais.

No entanto, aplicações de busca como as descritas no parágrafo anterior não possuem características analíticas, pois visam simplesmente apresentar

uma lista de referências para os dados ou documentos localizados de acordo com os termos de busca informados pelo usuário. Não são incorporados nessas ferramentas recursos analíticos como sumarização de dados ou operações de *drill*.

Na próxima Seção são apresentadas pesquisas que possuem escopo semelhante aos da presente pesquisa.

3.6 Aplicação de Tecnologias Semânticas para Suporte de Aplicações Analíticas

No tocante à integração de regras de negócio ao ambiente OLAP, verifica-se uma dissertação de mestrado (ALENQUER, 2002) em que o autor procura estender o modelo de metadados de uma ferramenta OLAP para inserir regras de negócio. No entanto, essa extensão é apenas utilizada com caráter informativo, para descrever as regras de negócio que foram levadas em consideração pelo desenvolvedor no momento da criação de um determinado cubo. As regras, contudo, não são utilizadas para apoiar as operações analíticas na ferramenta desenvolvida naquela pesquisa.

Em relação a arquiteturas de BI em desenvolvimento pela academia, destacam-se três iniciativas: SEWASIE (BERGAMASCHI et al., 2005), BIKM (CODY et al., 2002) e a proposta de Priebe e Pernul (2003).

SEWASIE é uma arquitetura que se encontra em desenvolvimento para suportar buscas de informações em fontes de dados diversas e para suporte a negociações (BERGAMASCHI et al.; 2004, 2005; CATARCI et al., 2004). Conforme ilustra a Figura 9, as consultas são formuladas pelos usuários e distribuídas pelo *Query Manager* para serem processadas nos nodos de informação. Essas consultas são executadas por agentes que cuidam de cada nodo (*i.e.*, repositórios de dados) através de um mecanismo que examina o mapeamento semântico entre as fontes de dados e uma ontologia de domínio. A definição dos cubos OLAP também é mapeada à ontologia de domínio, visando apoiar a recomendação de documentos relacionados à definição dos cubos e a personalização da apresentação dos dados. No entanto, não são utilizadas as definições da ontologia do domínio para suportar as funcionalidades exploratórias na ferramenta OLAP (*i.e.*, suporte para

operações de *drill* utilizando regras ou relações). SEWASIE também não suporta a utilização de WSS para a extensão das capacidades exploratórias, mas introduz suporte para negociações integrado à ferramenta de análise.

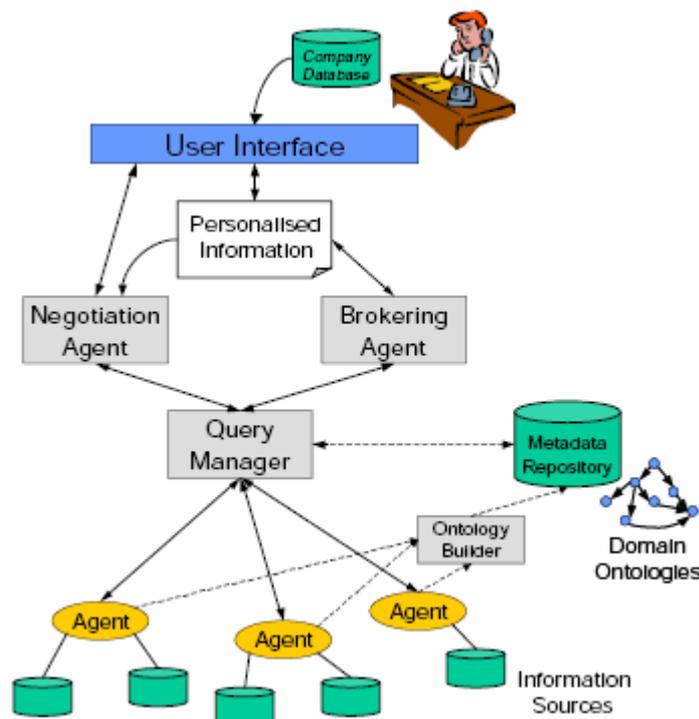


Figura 9 - Arquitetura SEWASIE
Fonte: BERGAMASCHI et al. (2005)

Da mesma forma que a arquitetura SEWASIE, o BIKM (CODY et al., 2002) busca suportar a localização e a recomendação de documentos ao tomador de decisão à medida que o usuário explora cubos de dados em ferramentas OLAP. O foco dessas iniciativas está no mapeamento e no suporte à integração de dados distribuídos, incluindo os dados não estruturados de interesse ao tomador de decisão. BIKM inclui algoritmos de mineração de textos para extrair a relação entre um *corpus* e as dimensões de um DW suportado pela arquitetura e para a geração de dimensões e fatos a partir de documentos. Esses algoritmos encontram-se em fase de desenvolvimento. Dessa forma, BIKM possibilitará a confecção de análises exploratórias sobre um cubo de dados e a localização de documentos relacionados a um cubo de dados ou utilização de características dos próprios documentos como fontes de colunas, linhas e medidas em cubos de dados.

Priebe e Pernul (2003) propõem uma abordagem para mapear cubos e documentos para possibilitar buscas integradas de recursos. Uma ontologia é

aplicada para identificar documentos mantidos por um gerenciador de documentos e para descrever os cubos gerados pelo usuário em uma ferramenta OLAP comercial. Usuários podem então realizar buscas integradas, utilizando a ontologia para localizar os documentos e os cubos anotados. A anotação dos cubos e dos documentos é feita através do mapeamento dos metadados das ferramentas comerciais à ontologia da arquitetura. O sistema utiliza informações como o título e os elementos relacionados nas dimensões de um cubo, além da descrição entrada manualmente no gerenciador de documentos. O sistema permite que usuários façam buscas utilizando algoritmos de análise de similaridade entre os termos da busca e a ontologia da arquitetura. Esse sistema, no entanto, não utiliza a ontologia para apoiar operações de *drill* e tampouco aborda a questão da flexibilidade de extensão das funcionalidades exploratórias utilizando WSS.

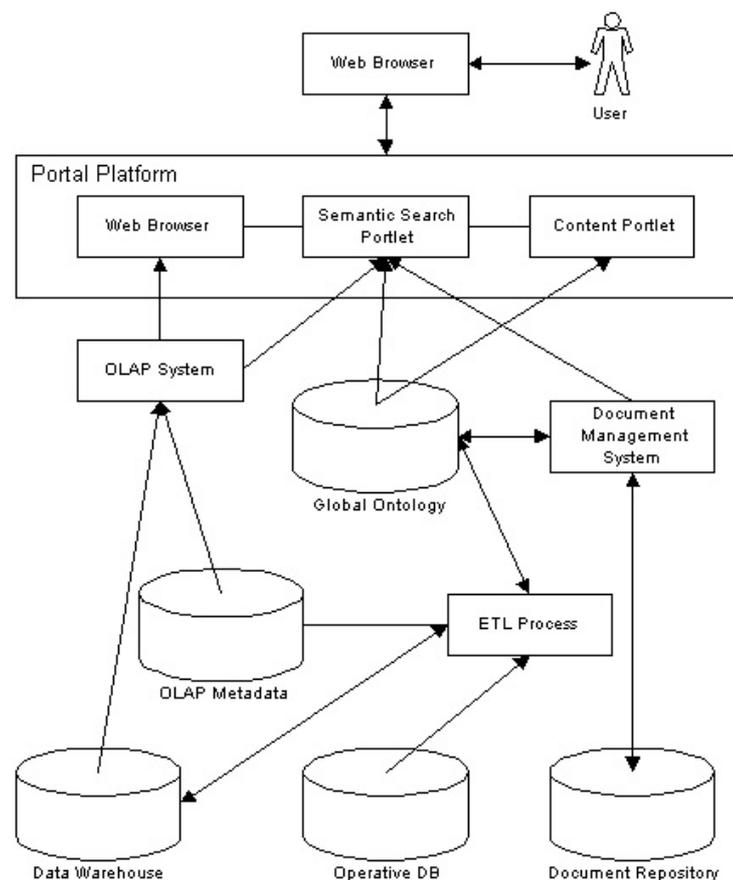


Figura 10 - Arquitetura Proposta por Priebe e Pernul (2003)

3.7 Considerações sobre o Capítulo

Neste capítulo, foram introduzidas as tecnologias que suportam a inserção da semântica do negócio na base da arquitetura proposta no presente trabalho. A pesquisa em ontologias no começo dos anos 90 era limitada a um pequeno grupo da comunidade acadêmica. Essa situação começou a mudar com o foco na disseminação do conhecimento, culminando com um grande salto no interesse da comunidade científica em geral com o surgimento da *Web semântica*.

O principal objetivo da *Web semântica* é o de tornar possível a descrição dos recursos disponibilizados na *Web*, de maneira a permitir que agentes localizem conteúdo e realizem inferências sobre esse conteúdo. Para tanto, ontologias vêm sendo utilizadas para agregar descrição semântica e lógica ao conteúdo da *Web*.

Busca-se nesta pesquisa contribuir para a resolução dos problemas das soluções correntes de BI, nominalmente a falta de flexibilidade para adaptação e personalização das funcionalidades analíticas, de acordo com as necessidades das organizações, e o distanciamento das aplicações analíticas dos conceitos e regras de negócio dessas organizações. Pesquisas relacionadas à aplicação de tecnologias semânticas no contexto de BI abordam apenas de forma periférica esses desafios, buscando basicamente possibilitar a combinação de dados estruturados e não estruturados para o apoio ao processo decisório.

Nesta pesquisa, as ontologias são aplicadas para a descrição semântica dos conceitos e regras do negócio, além dos dados e dos serviços de interesse dos tomadores de decisão. Essa representação semântica pode ser explorada por máquinas de inferência para apoiar o processo de tomada de decisão, visando guiar o usuário através da recomendação de dados, serviços e inferências, de acordo com o contexto de uma análise. Dessa forma, busca-se oferecer uma alternativa para a resolução das principais deficiências das arquiteturas atuais.

No próximo capítulo são apresentados os requisitos levados em consideração no projeto da arquitetura, além de uma visão geral da forma que

a arquitetura SBI foi organizada e das etapas para concepção de soluções de BI a partir da arquitetura.

4 ARQUITETURA SEMANTIC BUSINESS INTELLIGENCE (SBI)

4.1 Introdução

Neste capítulo, é apresentado como a arquitetura proposta foi organizada visando à resolução das problemáticas desta pesquisa. Apresenta-se uma visão geral das camadas que formam a arquitetura bem como uma visão dos módulos e ontologias reunidos em cada camada.

O projeto da arquitetura SBI teve como base um conjunto de requisitos funcionais e não funcionais. Esses requisitos foram identificados a partir de uma análise das principais limitações das soluções de BI atuais e de novos desafios no contexto da sociedade do conhecimento. Os requisitos identificados foram classificados de acordo com os grupos de regras de Codd et al. (1995), visando complementar essas regras e fornecer diretrizes para a criação de uma arquitetura que ofereça os recursos necessários para a resolução das limitações das soluções de BI correntes.

Por fim, são apresentados as etapas necessárias para a consecução de uma solução de BI a partir da arquitetura SBI e os novos papéis a serem desempenhados pela equipe de desenvolvimento diante dos novos recursos e requisitos introduzidos pela arquitetura.

4.2 Revisão dos Requisitos Funcionais para Arquiteturas de BI

Os requisitos funcionais da arquitetura SBI foram identificados a partir de um levantamento em revistas especializadas sobre BI e em artigos divulgados por institutos de pesquisa sobre as principais limitações das abordagens atuais para projetos de BI. Para a definição dos requisitos, foram consideradas ainda as limitações nas arquiteturas atuais encontradas pelo autor nos vários projetos de pesquisa e desenvolvimento relacionados a BI em que o autor atuou³. Os requisitos identificados foram comparados com a lista de regras propostas por

³ Para uma relação das atividades de pesquisa e desenvolvimento no contexto de BI, vide o currículo Lattes do autor em <<http://buscatextual.cnpq.br/buscatextual/visualizacv.jsp?id=K4762632A6>>.

Codd et al. (1995) a fim de verificar os hiatos entre as regras tradicionais e as necessidades identificadas.

Poucas referências acadêmicas sobre os desafios na área de BI foram encontradas para embasar o levantamento de requisitos. Como descrito no capítulo anterior, basicamente as pesquisas na área focam a questão da integração de fontes de dados heterogêneas no contexto de BI. No entanto, apesar da importância do tema “integração de dados” para as soluções de BI, entre as principais causas citadas na literatura para os altos percentuais de fracasso dos projetos de BI, encontra-se a falta de aderência das soluções implantadas com os requisitos analíticos das organizações (COMPUTERWORLD, 2004; GARTNER GROUP, 2004).

As soluções de BI se limitam a oferecer funcionalidades genéricas para exploração das montanhas de dados reunidas nos repositórios das organizações, funcionalidades estas normalmente dissociadas da semântica do negócio. O tomador de decisão dispõe unicamente do seu conhecimento pessoal para apoiá-lo durante o processamento analítico sobre as fontes de dados, conhecimento este que muitas vezes se mostra insuficiente para fazer os julgamentos corretos em um processamento tão complexo. A seleção, transformação e análise da informação para a formação do conhecimento ainda dependem exclusivamente das habilidades cognitivas, críticas e teóricas dos tomadores de decisão (LIEBOWITZ, 2005b; WHITE, 2005).

À medida que o investimento em soluções de BI aumenta e que soluções de gestão de conhecimento tornam-se cada vez mais populares nas organizações, a análise das possibilidades de integração entre as duas iniciativas torna-se mais importante. O conhecimento adquirido sobre o negócio através das soluções de gestão de conhecimento, se mapeado às fontes de dados da organização, pode prover o contexto do negócio necessário para apoiar o processamento analítico.

Da necessidade de se explorarem as formas de utilização do conhecimento sobre o negócio para suporte ao processamento analítico, formulam-se três requisitos funcionais a serem inseridos no grupo de características básicas propostas por Codd et al. (vide Quadro 4, página 24):

Requisito Funcional 1 (RF1): Possibilitar a navegação sobre as fontes de dados a partir dos conceitos do negócio e seus relacionamentos

Descrição	Abstrair a forma que os dados estão organizados nos repositórios. Permitir que os usuários selecionem os dados que formarão suas análises e naveguem sobre o resultado dessas análises através da descrição semântica dos conceitos do negócio.
------------------	---

Requisito Funcional 2 (RF2): Permitir a utilização das regras de negócio para apoiar o processamento analítico

Descrição	Permitir que as regras de negócio definidas na semântica do negócio sejam utilizadas para filtrar o conteúdo de uma análise ou ainda ser utilizada para apoiar operações de <i>drill</i> sobre a informação reunida em uma análise.
------------------	---

Requisito Funcional 3 (RF3): Propiciar flexibilidade para modificação dos conceitos e regras do negócio

Descrição	Permitir a mudança das regras e a visão conceitual do negócio sem necessariamente mudar o modelo de dados e a implementação ETL da solução de BI.
------------------	---

No levantamento dos desafios associados às soluções de BI, identificou-se ainda a questão da falta de flexibilidade para a extensão das funcionalidades analíticas das soluções de BI de acordo com as especificidades de cada organização. As soluções atuais foram concebidas a partir de um núcleo básico e genérico de funcionalidades analíticas e não apresentam maneiras escaláveis para extensão ou adaptação dessas funcionalidades (CA MAGAZINE, 2004, 2005).

Uma solução escalável deve ser oferecida através de padrões abertos, que sejam de conhecimento comum e que permitam que os desenvolvedores das organizações realizem customizações ou adicionem novas funcionalidades para atender às especificidades de suas organizações. Além disso, devem-se investigar formas de se oferecerem automaticamente alternativas de funcionalidades exploratórias ao usuário de acordo com o contexto de suas análises, com vistas a guiá-lo e facilitar o processamento da informação e a extração de conhecimento útil ao processo decisório. Dessa forma, formulam-

se dois novos requisitos funcionais a serem inseridos no grupo de características básicas propostos por Codd et al. (vide Quadro 4, página 24):

Requisito Funcional 4 (RF4): Permitir a extensão das funcionalidades exploratórias a partir de aplicações existentes na Web ou na organização	
Descrição	Permitir a adição de novas funcionalidades analíticas às ferramentas analíticas a partir da descrição semântica dos serviços existentes na Web ou na Intranet. Esses serviços poderão ser utilizados para o atendimento de necessidades específicas nas organizações como na busca de dados complementares para uma análise ou para a realização de uma transação a partir dos dados reunidos em uma análise.

Requisito Funcional 5 (RF5): Permitir a composição de serviços para a extensão de funcionalidades exploratórias	
Descrição	Oferecer ao usuário a possibilidade de montar composições de serviços visando atender às necessidades analíticas que não possam ser resolvidas por apenas um serviço. Permitir ainda que a descrição semântica dos serviços disponíveis seja utilizada para facilitar o processo de composição.

O RF4 refere-se à possibilidade de se utilizarem serviços para que dados adicionais (internos ou externos à organização) possam ser trazidos diretamente das fontes de origem dos dados para o cubo ou relatório do usuário, como no exemplo da busca de dados do IBGE apresentado na introdução desta tese. Serviços também poderiam ser invocados pelas aplicações analíticas para a execução de outros tipos de transações, usando-se os dados de uma análise como entrada, tal como no exemplo da classificação de pesquisadores que foi apresentado na introdução desta tese.

Em relação ao RF5, devem-se buscar maneiras de compor serviços para a realização de transformações sobre os dados ou outros tipos de transação a partir dos dados reunidos na análise. Essas transações poderiam envolver não somente um serviço, mas vários, organizados de maneira que o resultado da execução do primeiro seja usado como entrada pelo segundo e assim por diante. Um exemplo de aplicação desse requisito foi apresentado na introdução desta tese, onde foi descrita uma composição de serviços para classificar pesquisadores (primeiro serviço) e realizar a distribuição de vagas de orientação de acordo com a classificação obtida (segundo serviço).

Composições como essa podem oferecer aos usuários recursos ilimitados de processamento de informação e geração de conhecimento. Esse recurso pode ainda oferecer aos desenvolvedores a liberdade para a concepção de aplicações modulares a partir da reutilização de código existente na organização ou na *Web*.

Entretanto, não é suficiente criar a possibilidade de utilizar-se *Web services* para estender as capacidades exploratórias das ferramentas analíticas. É necessário lidar ainda com o desafio de criação de mecanismos de identificação automática da necessidade de informação do usuário e da identificação das capacidades dos *Web services* disponíveis para tornar possível a recomendação de serviços que possam ser úteis ao usuário. Com a proliferação de *Web services* nas organizações e na *Web*, um tomador de decisão não conseguiria identificar qual serviço poderia satisfazer sua necessidade de processamento analítico. Da mesma forma, devem-se oferecer ao usuário, de forma proativa, possibilidades de uso das regras de negócio e dos relacionamentos entre os conceitos do negócio para a seleção e transformação dos dados necessários para a tomada de decisão. Dessa forma, define-se um novo requisito funcional a ser inserido no grupo de características de relato proposto por Codd et al. (vide Quadro 5, página 35):

Requisito Funcional 6 (RF6): Suportar a recomendação proativa de recursos aos usuários para apoiar o processamento analítico	
Descrição	Inferir a necessidade de informação do usuário através do contexto da sua análise e recomendar, de forma proativa, serviços, dados e formas de navegação sobre as fontes de dados, de maneira a guiar o usuário durante o processamento analítico.

Para o atendimento do requisito RF6, deve-se buscar uma maneira de identificar a necessidade de informação do usuário, através da determinação do tema da análise, do seu escopo e dos focos de interesse do usuário nessa análise. Deve-se ainda buscar uma forma de descrever os *Web services* para apoiar a localização de serviços pela descrição de suas funcionalidades, a partir da identificação da necessidade de informação do usuário. Dessa maneira, tornar-se-ia possível, por exemplo, oferecer automaticamente ao tomador de decisão o *Web service* do IBGE citado na introdução desta tese.

Devem-se também buscar maneiras de se oferecer de forma proativa ao usuário alternativas de navegação sobre os dados utilizando as regras de negócio e os relacionamentos associados aos conceitos manipulados na análise. No exemplo de análise de patentes dos egressos pela região de endereço desses egressos, conforme apresentado na introdução desta tese, deveriam ser oferecidas ao usuário possibilidades de *drill* sobre o total de patentes. Para tanto, poderiam ser inferidos os relacionamentos do conceito *região*, por exemplo, de forma a possibilitar ao usuário a análise do número de patentes por *estado*, por *município* e pelas *instituições* localizadas na região de interesse do usuário.

4.3 Identificação de Requisitos não Funcionais para Arquiteturas de BI

Além dos requisitos funcionais apresentados na Seção anterior, o projeto da arquitetura SBI teve como base um conjunto de requisitos não funcionais. Os requisitos não funcionais foram definidos visando garantir flexibilidade para implantação e adaptação da arquitetura, complementando dessa maneira os requisitos funcionais descritos na Seção anterior. Assim, foram identificados três requisitos não funcionais, apresentados a seguir.

Nas próximas seções é apresentada uma introdução sobre as camadas que formam a arquitetura SBI e sua relação com os requisitos identificados nesta Seção e na Seção anterior.

Requisito Não Funcional 1 (RNF1): Possibilitar a aplicação da arquitetura no desenvolvimento de soluções de BI para qualquer domínio de negócio	
Descrição	Os módulos da arquitetura deverão ser projetados para ter independência de domínio de aplicação. Para tanto, deverá ser previsto um baixo nível de acoplamento entre a representação da semântica do negócio, as fontes de dados da organização e os módulos da arquitetura.

Requisito Não Funcional 2 (RNF2): Permitir que a representação semântica do negócio, das fontes de dados e dos serviços sejam definidas através de diferentes formalismos

Descrição	A arquitetura deverá oferecer ao desenvolvedor a liberdade para a seleção da linguagem de sua preferência para a representação das ontologias. Deverá ser permitida ainda a reutilização de ontologias disponíveis na <i>Web</i> ou nas bases de conhecimento já existentes na organização.
------------------	---

Requisito Não Funcional 3 (RNF3): Permitir a integração dos módulos componentes da solução de BI a soluções de terceiros	
Descrição	A arquitetura deverá permitir a integração de parte ou de todos os seus componentes a ferramentas de terceiros para oferecer maior flexibilidade de integração aos sistemas existentes na organização. Para tanto, seus módulos deverão ser projetados com baixo nível de acoplamento entre si e expostos através de APIs ou serviços, garantindo sua utilização para outros fins.

4.4 Visão Geral da Arquitetura SBI

Visando atender aos requisitos funcionais e não funcionais identificados, a Arquitetura SBI foi estruturada através de um conjunto de módulos que reúnem ontologias, repositórios de dados e outros módulos funcionais (SELL et al., 2004a; 2005a; 2005b). Os módulos foram projetados com baixo nível de acoplamento entre si e distribuídos em cinco camadas, conforme ilustra a Figura 11. A distribuição em camadas e a preocupação com o baixo nível de acoplamento entre os módulos visam prover maior flexibilidade para o desenvolvedor manter a arquitetura e para facilitar a integração dos módulos a outras soluções na organização, em consonância com o requisito não funcional RNF3.

A utilização de ontologias e de *Web services* semânticos para o suporte ao processamento analítico bem como os módulos funcionais que integram essas tecnologias às soluções analíticas são propostas introduzidas pela arquitetura SBI. A combinação de tecnologias semânticas e componentes tradicionais de arquiteturas de BI visa ao cumprimento dos novos requisitos identificados nas seções anteriores. Na Seção 3.6 foram apresentadas outras iniciativas acadêmicas que buscam utilizar ontologias, mas com objetivos pouco semelhantes aos desta tese. Na Seção 7.5 são apresentadas as

diferenças entre a abordagem desta pesquisa, as iniciativas acadêmicas e as soluções comerciais, no contexto de arquiteturas de BI.

A seguir são introduzidas as camadas da arquitetura, seus objetivos e suas relações com os requisitos apresentados nas seções anteriores. Em seguida, são apresentados os módulos que compõem cada camada.

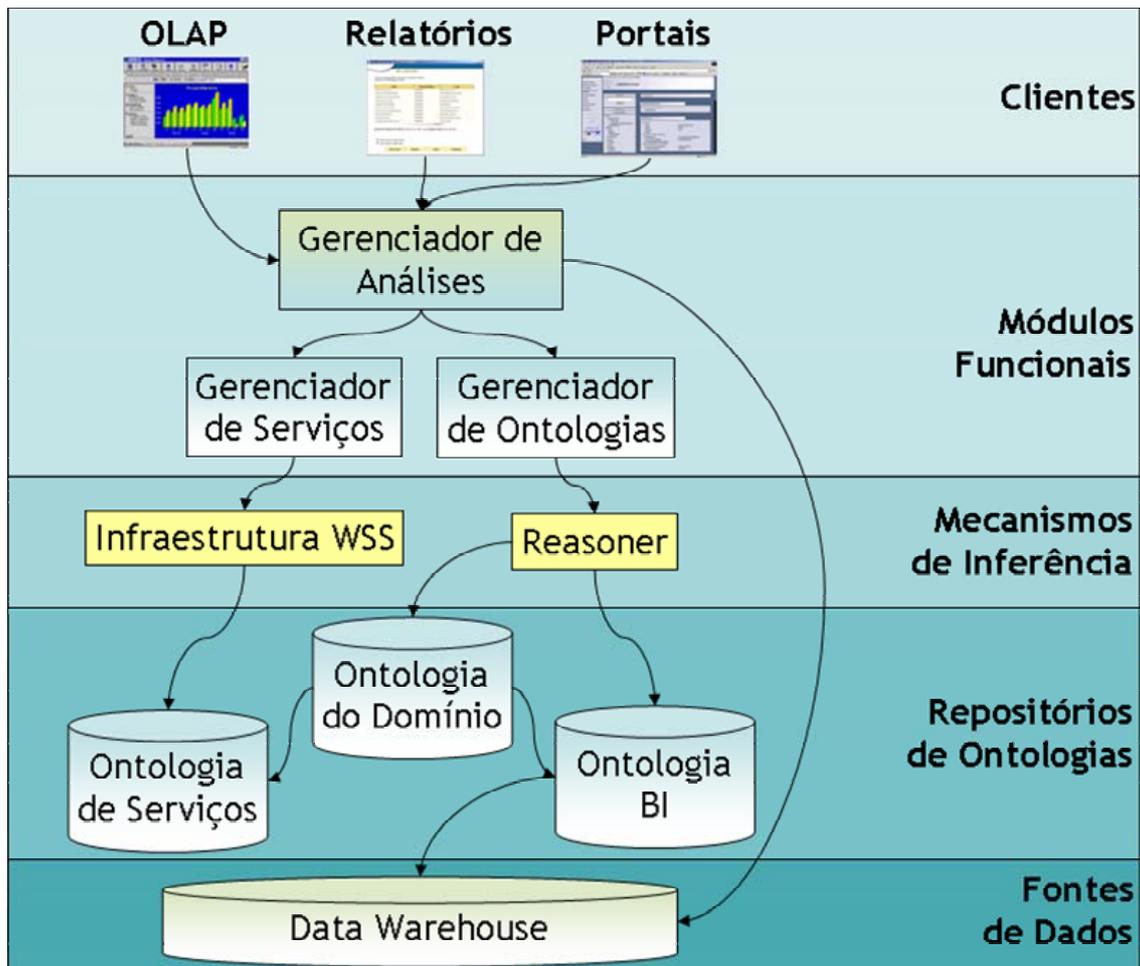


Figura 11 - Ilustração dos Módulos da Arquitetura Proposta

4.5 A Camada Fontes de Dados

A camada Fontes de Dados corresponde aos *data marts* da organização. Os *data marts* conterão os dados estratificados dos vários repositórios de dados da organização ou de fontes externas. A arquitetura deverá prover recursos de acesso a esses *data marts*, de maneira a tornar transparente a forma como os dados estão organizados nestes *data marts*. Para tanto, a arquitetura proverá nas demais camadas o suporte para a representação semântica das fontes de dados e um conjunto de métodos

padrão para acesso aos repositórios. Maiores detalhes sobre os módulos que proverão o mapeamento e o acesso aos dados da camada Fontes de Dados são apresentados nas próximas seções.

4.6 A Camada Repositórios de Ontologias

A camada Repositórios de Ontologias corresponde aos modelos de conhecimento necessários para o atendimento dos requisitos funcionais apresentados anteriormente. Para tanto, foram constituídas três ontologias que permitem o mapeamento da semântica do negócio, dos dados da organização e dos serviços necessários para o apoio ao processo decisório. São as ontologias:

- **Ontologia do Domínio:** modela a semântica do negócio através da representação da terminologia e das regras associadas aos processos do negócio. Os conceitos modelados nesta ontologia são utilizados para apoiar o cumprimento de todos os requisitos funcionais. Mais precisamente, essa descrição semântica é utilizada para: a) a identificação do contexto do negócio para suportar a navegação sobre as fontes de dados (RF1); b) a modelagem dos conceitos e das regras de negócio para apoiar o processamento analítico (RF2 e RF3) e; c) o fornecimento dos conceitos do negócio para apoiar a descrição semântica dos serviços e das análises dos usuários visando ao atendimento dos requisitos RF4, RF5 e RF6;
- **Ontologia BI:** provê os construtores semânticos utilizados para mapear os conceitos do negócio definidos na Ontologia do Domínio às fontes de dados da organização. Através desse mapeamento, torna-se possível a navegação sobre as fontes de dados usando a semântica do negócio (RF1) e a utilização das regras de negócio para apoiar o processamento analítico (RF2). Na Ontologia BI são mantidos ainda os conceitos utilizados para modelar os temas de análise e as análises confeccionadas pelos usuários por meio das ferramentas analíticas. Através da descrição semântica das análises do usuário, é possível inferir a

sua necessidade de informação e oferecer alternativas de navegação sobre a informação (RF6);

- **Ontologia de Serviços:** mantém os construtores utilizados para descrever semanticamente *Web services*. Essa descrição apóia a utilização de serviços para estender as funcionalidades exploratórias (RF4) e suporta a composição de serviços (RF5). Essa descrição semântica é empregada ainda para suportar a localização de serviços de acordo com o contexto de uma análise, de forma a oferecer alternativas exploratórias ao tomador de decisão (RF6).

A separação dos módulos da arquitetura da representação do conhecimento do negócio, da descrição semântica das fontes de dados e da representação dos serviços visa atender ao requisito não funcional RNF1. Com a separação do conhecimento representado nas ontologias dos demais módulos da arquitetura, torna-se possível aplicar a arquitetura em qualquer área de negócio. Essa separação permite ainda a realização de modificações nas regras e na terminologia do negócio a qualquer momento, sem prejuízo ao funcionamento dos módulos e das aplicações clientes da arquitetura. Assim, novas regras de negócio poderão ser inseridas bem como modificações nos relacionamentos dos conceitos do negócio poderão ser efetuadas na Ontologia do Domínio sem que os repositórios de dados precisem ser alterados (RF3).

4.7 A Camada Mecanismos de Inferência

A camada “Mecanismos de Inferência” corresponde ao framework de *Web services* semânticos e ao mecanismo para inferência semântica acoplados à arquitetura. Essas ferramentas de terceiros suportarão, respectivamente, a integração de *Web services* à arquitetura e as inferências sobre as três ontologias da arquitetura.

Na implementação corrente da arquitetura SBI são utilizados o framework para *Web services* semânticos IRS-III (DOMINGUE et al. 2004a) e a linguagem OCML (MOTTA, 1999), bem como o seu respectivo mecanismo de inferência.

No entanto, visando atender ao requisito não funcional RNF2, a arquitetura prevê um baixo nível de acoplamento com as ferramentas de terceiros. A integração com esses mecanismos dá-se através de implementações das interfaces padrões dos módulos Gerenciador de Serviços e Gerenciador de Ontologias. Assim, uma implementação específica dos métodos previstos pelos dois módulos foi feita para o framework WSS IRS-III e para OCML. Para utilizar diferentes *frameworks* WSS ou outras linguagens para representação de ontologia, bastarão serem desenvolvidas novas implementações das interfaces desses dois módulos. Maiores detalhes sobre as interfaces desses módulos são apresentados no Capítulo 6.

4.8 A Camada Módulos Funcionais

A camada “Módulos Funcionais” compreende os módulos que são utilizados para articular a integração de serviços e o suporte a inferências às aplicações clientes da arquitetura SBI. Os módulos que compõem esta camada implementam funcionalidades que, com o apoio do conhecimento representado nas ontologias da arquitetura, visam atender aos requisitos funcionais previstos. Para tanto, foram constituídos três módulos funcionais que permitem que as ferramentas clientes tenham um acesso integrado à semântica do negócio, aos dados da organização e aos serviços necessários para atender às demandas analíticas dos tomadores de decisão. São os módulos funcionais:

- **Gerenciador de Ontologias:** módulo que especifica e implementa todas as funcionalidades necessárias para a manipulação da semântica do negócio. Compreende métodos para a obtenção dos conceitos do negócio modelados na Ontologia do Domínio visando suportar a navegação sobre as fontes de dados (RF1), além de métodos para localização e processamento de regras de negócio para apoiar o processamento analítico (RF2). Para tanto, este módulo especifica um conjunto de métodos, os quais são implementados sobre os *frameworks* da camada Mecanismos de Inferência;
- **Gerenciador de Serviços:** compreende as funcionalidades necessárias para localização, execução, monitoramento e

composição de serviços, visando atender aos requisitos funcionais RF4, RF5 e RF6. Para tanto, baseia-se na descrição semântica dos serviços mantida pela Ontologia de Serviços e apóia-se no framework para WSS da camada de Mecanismos de Inferência;

- **Gerenciador de Análises:** provê às ferramentas clientes acesso a todos os módulos da arquitetura SBI. Apoiado pelo Gerenciador de Ontologias, provê os métodos para a navegação sobre as fontes de dados usando a semântica do negócio (RF1) e para a utilização das regras de negócio para apoiar o processamento analítico (RF2). Implementa ainda os métodos necessários para a confecção e a execução de análises sobre as fontes de dados. Para tanto, conta com o apoio do Gerenciador de Ontologias e do Gerenciador de Serviços para inferir a necessidade de informação do usuário e oferecer alternativas de navegação sobre a informação (RF6).

4.9 A Camada Aplicações Clientes

Nesta camada figurarão as ferramentas analíticas que proverão aos tomadores de decisão acesso às fontes de dados para a obtenção das informações desejadas e as operações semântico-analíticas providas pela arquitetura. Essas ferramentas acessarão todos os recursos da arquitetura SBI através do Gerenciador de Análises.

4.10 O Processo de Desenvolvimento da Arquitetura SBI

Para a operacionalização da arquitetura SBI e para o acoplamento de ferramentas de apoio à decisão aos seus módulos, é necessária a incorporação de novas etapas nas metodologias de desenvolvimento de BI. Essas novas etapas, por tratarem de aspectos não abordados até então pelas metodologias atuais (*e.g.*, representação do conhecimento do negócio e o seu mapeamento aos dados e serviços), exigem da equipe de desenvolvimento o desempenho de dois novos papéis, o de Engenheiro do Conhecimento e o de Arquiteto de Serviços. As novas etapas e os papéis associados são descritos no Quadro 10.

Quadro 10 - Sumário das Novas Etapas de Desenvolvimento Introduzidas pela Arquitetura SBI e dos Papéis Associados às Etapas

Etapa	Descrição	Papéis Associados
Representação da Semântica do Negócio	Identificação da terminologia e das regras do negócio e representação dessas definições na Ontologia do Domínio	Engenheiro do Conhecimento e Analista de Negócios
Mapeamento Semântico do Modelo de Dados	Mapeamento da semântica do negócio aos dados da organização na Ontologia BI	Engenheiro do Conhecimento, Analista de Negócios e Modelador de Dados
Mapeamento Semântico de Serviços	Descrição semântica dos serviços internos ou externos de interesse aos tomadores de decisão na Ontologia de Serviços e projeto dos mediadores para integração desses serviços à arquitetura	Engenheiro do Conhecimento, Arquiteto de Serviços, Analista de Negócios, Arquiteto de Segurança e Desenvolvedor de Front End

Os papéis introduzidos exigem dos membros da equipe novas competências. Até então, nos projetos de soluções de BI, o conhecimento era obtido pelo analista de negócios e repassado para a equipe de desenvolvimento, que por sua vez concebia o modelo de dados e as ferramentas ETL e de apresentação de dados. Esse processo muda com a Arquitetura SBI.

O conhecimento sobre o negócio, cerne da nova proposta, deixa de ser somente interpretado pelo analista de negócio, passando a ser efetivamente representado e integrado aos módulos da solução de BI. Essa quebra de paradigma de desenvolvimento exige que novas competências sejam buscadas pelos membros da equipe. Assim são criados dois novos papéis a serem desempenhados pela equipe de desenvolvimento, a do Engenheiro do Conhecimento e do Arquiteto de Serviços.

As competências associadas ao papel Engenheiro do Conhecimento envolvem o conhecimento sobre técnicas e ferramentas para aquisição e representação da semântica do negócio. Os papéis de Engenheiro de Conhecimento e de Analista de Negócio possivelmente serão desempenhados pelas mesmas pessoas, dada a complementaridade dos dois papéis. Cabe ao

Analista de Negócio preparar e conduzir as entrevistas junto aos tomadores de decisão e cabe ao Engenheiro do Conhecimento o acompanhamento destas entrevistas e o levantamento de informações complementares sobre o negócio, além da representação dos conceitos identificados na Ontologia do Domínio. O Engenheiro de Conhecimento atuará ainda em conjunto com o Modelador de Dados no mapeamento dos dados à semântica do negócio.

Em relação à possibilidade de integração de serviços para extensão da arquitetura SBI, é importante que sejam desenvolvidas pela equipe competências associadas ao projeto e à implementação de *Web services* e de representação semântica de serviços. Essas competências são aqui associadas ao papel Arquiteto de Serviços. A integração de serviços à arquitetura SBI demanda do Arquiteto de Serviços uma colaboração com outros papéis tradicionalmente desempenhados pelas equipes de projetos de BI. Entre esses papéis, destaca-se o Analista de Negócios, que informará ao Arquiteto de Serviços sobre as necessidades dos tomadores de decisão, e o Arquiteto de Segurança, que definirá as políticas de segurança durante o projeto de integração de serviços. O Arquiteto de Serviços atuará ainda em conjunto com o Engenheiro do Conhecimento, informando os serviços que serão integrados à arquitetura para que o Engenheiro do Conhecimento possa apoiar a descrição semântica do novo serviço.

As novas etapas de desenvolvimento introduzidas pela arquitetura SBI poderão ser incorporadas a uma metodologia de desenvolvimento tradicional de BI, como a metodologia BUS descrita na Seção 2.7, conforme ilustra a Figura 12. Assim, a representação da semântica do negócio poderia ser iniciada logo após a definição dos requisitos do negócio e em paralelo às demais fases de desenvolvimento. A semântica do negócio é mapeada ao modelo de dados tão logo seja finalizada a fase de projeto do *data warehouse*. Os serviços são descritos semanticamente a partir da definição dos conceitos do negócio. Finalmente, as ferramentas de apoio à decisão são integradas ao Gerenciador de Análises logo após a conclusão do seu desenvolvimento.

As novas etapas podem também ser executadas de forma independente do fluxo de desenvolvimento da solução de BI. A qualquer momento, novas regras de negócio podem ser incorporadas à representação da semântica do

negócio, assim como novos serviços podem ser incorporados à arquitetura à medida que sejam identificadas novas necessidades analíticas junto aos tomadores de decisão.

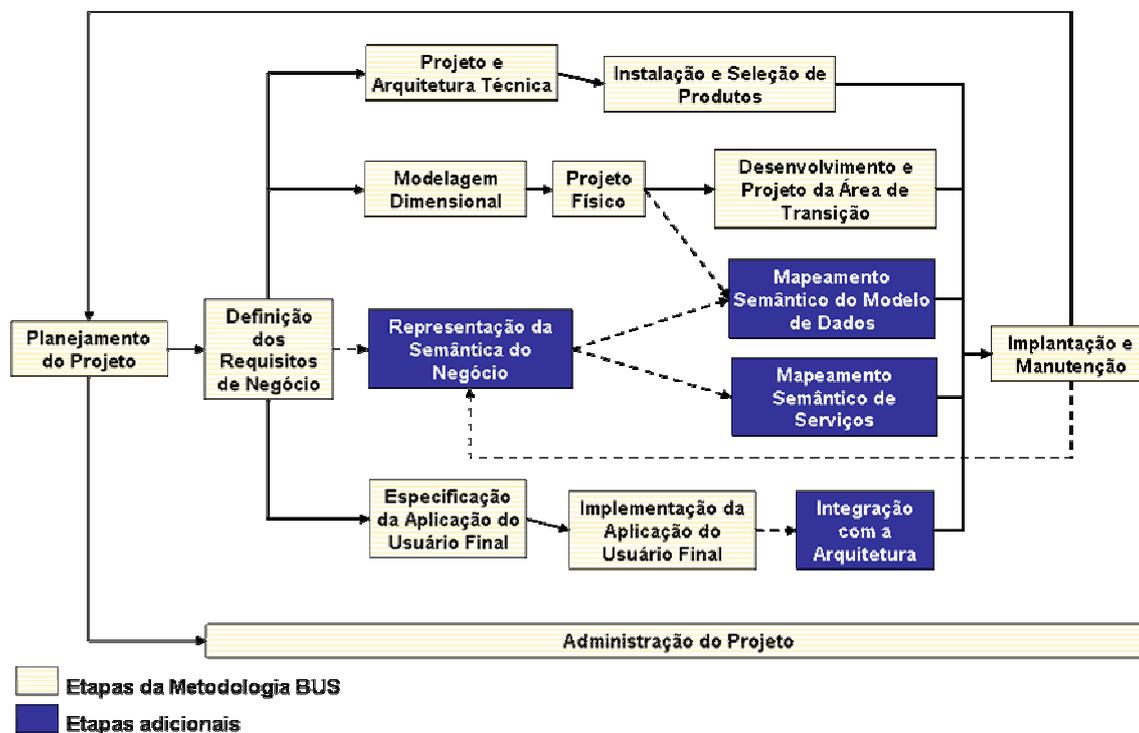


Figura 12 – Adição das etapas de desenvolvimento específicas da Arquitetura SBI a Metodologia BUS

4.11 Considerações sobre o Capítulo

Neste capítulo foram apresentados os requisitos funcionais e não funcionais que nortearam a concepção da arquitetura SBI. Esses requisitos foram identificados a partir de uma análise das principais deficiências apontadas pela literatura nas soluções de BI atuais e pelos novos desafios dessas soluções no contexto da sociedade do conhecimento.

Visando atender a esse conjunto de requisitos, foi especificado um conjunto de camadas que envolvem fontes de dados, repositórios de ontologias, mecanismos de inferência e módulos funcionais, que reunidos oferecem às aplicações analíticas um acesso integrado à semântica do negócio, além dos dados e serviços de interesse ao tomador de decisão.

Para a concepção de soluções de BI a partir da arquitetura SBI, deverão ser consideradas etapas adicionais às etapas das metodologias tradicionais. Essas novas etapas compreendem a representação da semântica do negócio e

a descrição semântica das fontes de dados e dos serviços de interesse aos tomadores de decisão. Para tanto, são identificados dois novos papéis a serem desempenhados pela equipe de desenvolvimento, o de Engenheiro do Conhecimento e de Arquiteto de Serviços.

No próximo capítulo são apresentadas em detalhes as ontologias que compõem a camada de repositórios de conhecimento da arquitetura. No capítulo subsequente são apresentados os módulos funcionais que formam a terceira camada da arquitetura.

5 A EPISTEMOLOGIA DA ARQUITETURA

5.1 Introdução

Neste capítulo são apresentadas as ontologias que formam a segunda camada da arquitetura SBI, a camada Repositórios de Ontologia. O conhecimento representado nessas ontologias corresponde ao mapeamento da terminologia e das regras do negócio, além da descrição semântica dos dados e dos serviços que subsidiarão a tomada de decisão na organização.

São apresentados os principais conceitos reunidos em cada ontologia e a sua relação com os módulos funcionais apresentados no próximo capítulo e com os requisitos funcionais e não funcionais mostrados no capítulo anterior.

5.2 Ontologia do Domínio

A Ontologia do Domínio provê a terminologia formalmente especificada do domínio do negócio que está sendo suportado pela arquitetura. Nessa ontologia são descritos os conceitos, os axiomas, as relações e as regras afetos ao negócio que é apoiado pela arquitetura SBI, visando suportar todos os requisitos funcionais dessa arquitetura.

Esta ontologia provê os conceitos do negócio utilizados para a descrição das entradas, saídas e propriedades não funcionais dos *Web services* semânticos na Ontologia de Serviços e para a contextualização das fontes de dados na Ontologia BI. Assim, é possível oferecer aos usuários a possibilidade de navegar sobre recursos disponibilizados pela arquitetura (*i.e.*, regras do negócio, dados e serviços) usando conceitos de negócio em vez de descrições técnicas. Além disso, a Ontologia do Domínio visa fornecer o contexto do negócio para apoiar o processamento de inferências sobre as fontes de dados e sobre os serviços de interesse aos tomadores de decisão.

As relações, regras e expressões lógicas do domínio do negócio suportarão a definição de análises, a redefinição automática de consultas geradas pelas análises e a extração de detalhes adicionais sobre os resultados de uma análise. Essas características são descritas com mais detalhes nas próximas seções, sendo ilustradas no Capítulo 7, onde é apresentado um protótipo de ferramenta analítica integrado à arquitetura SBI.

No desenvolvimento do estudo de caso e do protótipo de ferramenta analítica descritos no Capítulo 7, foram criados diversos conceitos e regras de negócio na Ontologia do Domínio para descrever o domínio da gestão de Ciência e Tecnologia (C&T), visando apoiar o processo decisório de diferentes atores ligados à C&T. Dessa forma, para o suporte do protótipo, foram representados nesta ontologia conceitos como *institution* (instituições de ensino e P&D), *person* (identificação de um estudante ou pesquisador) e *research_group* (grupos de P&D), como ilustra a Figura 13.

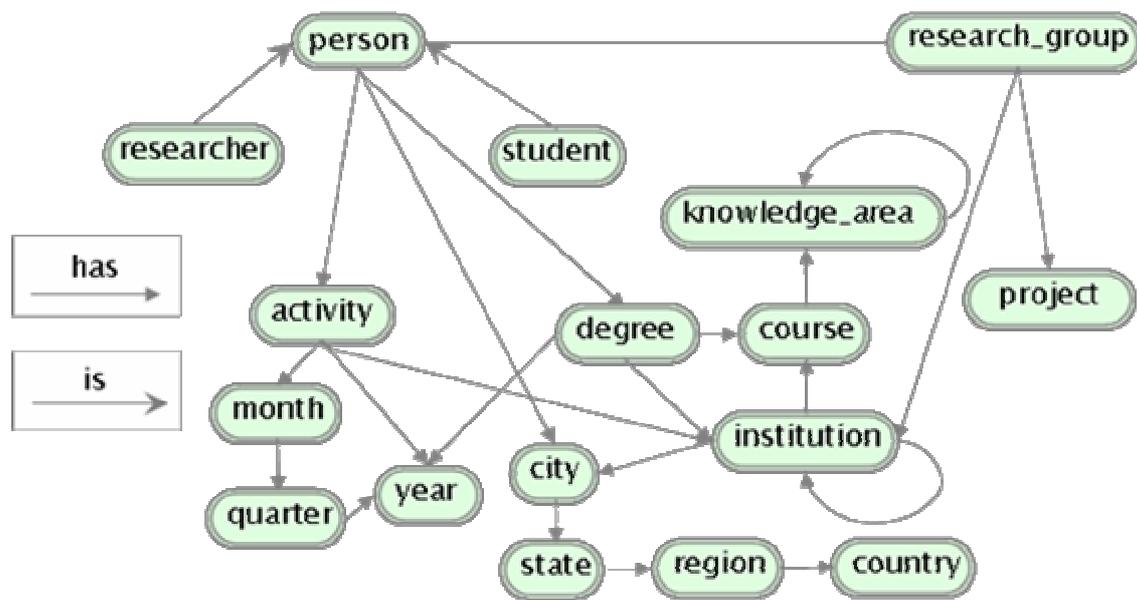


Figura 13 - Ilustração de conceitos associados à gestão de C&T representados na Ontologia do Domínio

Além da representação de conceitos, nesta ontologia figuram também relações e expressões lógicas que são utilizadas para representar as regras de negócio associadas à área de atuação da organização. Assim, no contexto da gestão da C&T, são descritas regras como *institution-alumnus* (i.e., *Egressos*), a qual descreve que “uma *pessoa* é considerada um *egresso* de uma dada *instituição* se essa *pessoa* possuir pelo menos uma *formação concluída* naquela *instituição* ”. Outro exemplo de regra de negócio é a regra *institution-competitor* (i.e., instituições de ensino ou pesquisa competidoras), sendo essa regra uma expressão lógica descrita na Ontologia do Domínio como “ *instituições* que *oferecem cursos* na mesma *área de conhecimento* e na mesma *cidade* ”. Essas duas relações, expressas na linguagem OCML, são apresentadas na Listagem 1.

Listagem 1 - Definição das relações *institution-competitor* e *institution-alumnus* em OCML

```
(def-relation institution-competitor (?i1 ?i2)
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (city ?c)
    (knowledge_area ?k)
    (is_institution_located_at_city ?i1 ?c)
    (has_course_institution ?co1 ?i1)
    (has_course_knowledge_area ?co1 ?k)
    (is_institution_located_at_city ?i2 ?c)
    (has_course_institution ?co2 ?i2)
    (has_course_knowledge_area ?co2 ?k)
    (not (= ?i1 ?i2))))

(def-relation institution-alumnus (?i ?p)
  :constraint (and (person ?p)(university ?i))
  :sufficient (and
    (has_degree_person ?d ?p)
    (is_degree_completed ?d "S")
    (has_degree_institution ?d ?i)))
```

As relações definidas entre os conceitos da Ontologia do Domínio são utilizadas para suportar *drill* e *slice* semânticos em análises. Operações de *drill-down* e *drill-up* podem utilizar as taxonomias definidas sobre os conceitos da Ontologia do Domínio para permitir a navegação de um nível mais macro da informação (e.g., análise do número de pesquisadores por *região do País*) para um nível mais detalhado (e.g., descendo ao nível dos *estados brasileiros*) e vice-versa. A navegação descrita no exemplo seria suportada pela relação transitiva utilizada para formar uma hierarquia geográfica conceitual. Esse tipo de operação torna-se possível através da ligação entre os conceitos do negócio representados na Ontologia do Domínio com os dados da camada Fontes de Dados. Essa ligação é definida na Ontologia BI, apresentada na próxima Seção.

Além disso, regras de negócio como as descritas na Listagem 1 podem ser utilizadas para suportar *slice* em análises. Por exemplo, o resultado da relação *institution-competitor* pode ser utilizado pelo usuário de uma ferramenta analítica para a realização de um *slice* em análises envolvendo dados de instituições, como por exemplo, em um relatório sobre o número de pesquisadores das instituições de ensino e pesquisa brasileiras. O processamento dessa inferência seria realizado após o usuário selecionar uma das instituições do cubo e solicitar o processamento da relação *institution-competitor*. O cubo seria então redefinido apresentando um comparativo do

número de pesquisadores entre a instituição selecionada e as suas concorrentes retornadas pelo processamento da relação.

As regras e as relações definidas na Ontologia do Domínio podem ser recomendadas ao tomador de decisão como alternativas para explorar suas análises de acordo com os conceitos selecionados pelos usuários para compor as dimensões e medidas de suas análises. No exemplo da análise do número de pesquisadores das instituições, podem ser oferecidas aos usuários, automaticamente, todas as regras e as relações definidas para os conceitos referentes a instituições e pesquisadores. Esse processo de recomendação leva em consideração todas as regras (*i.e.*, filtros semânticos) que possuam uma premissa ou parâmetro envolvendo os dois conceitos ou ainda suas relações transitivas e associativas com outros conceitos. Dessa forma, torna-se possível navegar sobre as fontes de dados utilizando definições específicas do negócio que está sendo apoiado pela arquitetura SBI. Maiores detalhes sobre o processo de recomendação são apresentados nas seções que descrevem os módulos Gerenciador de Ontologia (Seção 6.2) e Gerenciador de Análises (Seção 6.4).

Os conceitos do negócio e suas relações, além das regras de negócio, podem ser modificados a qualquer momento. Além disso, novas regras e relações podem ser adicionadas na ontologia sem a necessidade de modificação no modelo de dados do *Data warehouse* e respectivos sistemas de carga, o que propicia maior agilidade na realização de adaptações nas formas de navegação sobre os dados.

A Ontologia do Domínio pode ser formada através de um levantamento de conceitos e regras junto à organização. Dessa forma, parte-se do pressuposto de que um novo papel deve ser desempenhado pela equipe de desenvolvimento do projeto de BI. Esse papel, o do Engenheiro de Conhecimento, conforme descrito na Seção 4.10, pode ser acumulado pela equipe de analista de negócios responsável pelo levantamento de requisitos e pelo projeto da arquitetura. Assim, durante o ciclo de entrevistas para identificação das necessidades gerenciais, além do mapeamento dos requisitos de informação de cada entrevistado, deve-se em paralelo identificar e representar os conceitos e as regras de negócio na Ontologia do Domínio.

O processo de captura das definições semânticas do negócio pode ser agilizado através do reaproveitamento de definições constantes em outras ontologias correlatas ao domínio do negócio. Essas representações semânticas podem ser importadas de bases de conhecimento existentes nas organizações ou de uma das diversas bibliotecas de ontologias existentes na *Web* (e.g., DAML; PRÓTEGÉ OWL; SCHEMAWEB; 2006). Os conceitos importados de outras ontologias podem ser utilizados para estender as definições adquiridas durante as entrevistas ou para servir de benchmark, possibilitando assim a validação das representações feitas durante as entrevistas.

A integração dos conceitos descritos na Ontologia do Domínio às demais ontologias da arquitetura é apresentada nas seções 5.3 (Ontologia BI) e 5.4 (Ontologia de Serviços).

5.3 Ontologia BI

A Ontologia BI modela os conceitos usados para descrever como os dados estão organizados nos repositórios da organização e para mapear esses dados aos conceitos descritos na Ontologia do Domínio. Essas definições, além dos demais construtores semânticos reunidos nesta ontologia, são usadas para apoiar as seguintes tarefas:

- A. permitir a navegação sobre as fontes de dados a partir dos conceitos do negócio descritos na Ontologia do Domínio, de acordo com o requisito RF1;
- B. prover meios para a customização da apresentação dos resultados de uma consulta de acordo com o perfil do usuário visando melhor atender ao requisito RF1;
- C. suportar inferências usando os conceitos da Ontologia do Domínio para estender os resultados de consultas (*i.e.*, redefinição de uma consulta) visando atender ao requisito RF2;
- D. descrever semanticamente as análises dos usuários para apoiar a recomendação de informações e de serviços relacionados ao escopo dessas análises, visando atender ao requisito RF6; e;

máquinas de inferência já existentes para a exploração da base de conhecimento. Nessa estratégia, os dados dos bancos de dados são transformados em instâncias de classes, permitindo que sejam processadas inferências a partir de axiomas ou regras diretamente sobre as instâncias geradas na base de conhecimento. Entretanto, essa alternativa não é viável no domínio de soluções de BI, em que são agregados tipicamente centenas ou milhares de *gigabytes* de dados em um DW (KIMBALL et al. 1998), tornando-se complexa e cara a manutenção dessa replicação. Além disso, consultas analíticas típicas envolvem várias restrições e operadores de agregação (BERSON, 1997; THOMSEN, 2002), que não são trivialmente reproduzidos em linguagens de consultas para ontologias.

Na abordagem adotada na arquitetura proposta, as consultas são processadas diretamente sobre o DW. Entretanto, os conceitos do negócio descritos na Ontologia do Domínio são utilizados para suportar a redefinição das consultas de maneira a expandir ou filtrar os resultados de uma consulta. O processo de redefinição de consultas, descrito nas próximas seções, é apoiado pela replicação *parcial* dos dados do DW (*i.e.*, somente os dados descritivos mantidos nas dimensões do DW), na forma de instâncias dos conceitos do negócio dentro da Ontologia de Domínio. Em consultas OLAP, o usuário tipicamente relaciona atributos das dimensões para definir os cabeçalhos de linha e os marcadores de página. Do mesmo modo, os atributos das dimensões são utilizados como fonte de restrição e para dar suporte aos diversos tipos de *drill* implementados pelas ferramentas OLAP (HARRISON, 1997; THOMSEN, 2002). Dimensões são tipicamente tabelas pequenas e com pouca frequência de alteração (KIMBALL et al., 1998; KIMBALL; ROSS, 2002).

No exemplo de modelo dimensional ilustrado na Figura 4 (apresentada no Capítulo 2, página 30), figuram as dimensões descritivas de instituições de ensino, de grupos de pesquisa, do local onde a instituição está situada e da data. Essas dimensões, ligadas a uma tabela de fatos, a qual naquele exemplo representa a quantidade de pesquisadores, estudantes e linhas de pesquisa dos grupos de pesquisa em um dado mês, permitem análises como a do acompanhamento da evolução do número de pesquisadores por unidade federativa. As dimensões como as deste exemplo mantêm todos os dados que

possuem alguma relação com os conceitos do negócio, visto que os fatos em um DW normalmente mantêm apenas valores numéricos que mensuram o negócio da organização (CHUCK et al., 1998; KIMBALL et al., 1998; TANLER, 1998; SELL, 2001).

Assim, na abordagem utilizada nesta pesquisa, são geradas instâncias para os conceitos da Ontologia do Domínio de acordo com os dados das dimensões do DW, visando apoiar o processo de redefinição de consultas e outras inferências. No caso do exemplo do modelo dimensional de grupos de pesquisa, torna-se necessário modelar conceitos para representar instituições de ensino, de grupos de pesquisa, localidades e tempo. Em seguida, são criadas as instâncias para cada conceito a partir dos registros das tabelas de dimensão correspondentes. Esse processo de criação de instâncias não é suportado por um componente da arquitetura na presente versão. Este processo deverá ser feito através de uma ferramenta ETL ou através de APIs para criação de instâncias de ontologia a partir de dados mantidos em banco de dados, como a API SOFA (2006) ou a API JENA (2006), entre outras.

Os mapeamentos entre as dimensões mantidas nas fontes de dados da organização e os conceitos na Ontologia do Domínio são feitos através das classes *DB_Attribute* e *DB_Collection*. Essas e outras classes utilizadas para apoiar esse mapeamento são apresentadas no Quadro 11.

Na definição dos conceitos da Ontologia BI, a classe *DB_Collection* mapeia uma tabela criada em um repositório de dados a um conceito descrito na Ontologia do Domínio, identificando ainda seus atributos e sua chave, além de outras informações. Os mapeamentos entre os atributos das tabelas e os slots dos conceitos da Ontologia do Domínio são feitos através da classe *DB_Attribute*. A propriedade *has_attribute_name* na classe *DB_Attribute* é usada para identificar um campo de uma determinada tabela, enquanto *has_concept* e *has_slot* identificam o conceito e o slot correspondentes na Ontologia do Domínio. A propriedade *has_pk_attribute* na classe *DB_Collection* é usada para retornar valores da chave primária durante o processo de redefinição de condições das consultas implementado pelo módulo Gerenciador de Análise (vide Seção 6.4.2).

As classes *Dimension* e *Fact* estendem a classe *DB_Collection* para descrever as dimensões e as tabelas de fato de um DW. Uma instância da classe *Dimension* pode ter várias hierarquias de dados, como, por exemplo, uma hierarquia geográfica descrevendo que uma região possui vários Estados e que cada Estado possui vários municípios. As hierarquias (representadas pela classe *Hierarchy*) são utilizadas para descrever como podem ser realizadas operações de *drill-down* e *drill-up* utilizando os atributos das dimensões de um DW. Cada nível de uma hierarquia é descrito por uma instância da classe *Level*. As hierarquias explicitadas pela classe *Hierarchy* correspondem às hierarquias mapeadas no momento do projeto do DW. Além dessas, o usuário conta com as hierarquias dos conceitos definidos na Ontologia do Domínio para efetuar operações de *drill* sobre os dados.

A classe *Fact* representa uma tabela de fatos e possui atributos que identificam as dimensões associadas. Essas ligações são representadas através de instâncias da classe *Dimension_Usage*. A classe *Dimension_Usage* identifica como uma determinada dimensão pode ser unida a uma tabela de fatos através de uma coleção de instâncias da classe *DB_Collection_Join*, a qual identifica os atributos necessários para se efetuar a junção entre duas tabelas.

Quadro 11 - Descrição das classes utilizadas para suportar o mapeamento de fontes de dados e conceitos

Classe	Atributo	Descrição
DB_Collection	has_collection_name	Descreve o nome tabela
	has_pk_attribute	Descreve a chave primária da tabela através de uma lista de instâncias de <i>DB_Attribute</i>
	has_attribute	Mantém a lista de <i>DB_Attribute</i> que expressa os campos da tabela
	has_schema	Identifica o esquema do banco de dados no qual a tabela é mantida
DB_Attribute	has_attribute_name	Descreve o nome do campo da tabela correspondente
	has_concept	Aponta um conceito descrito na Ontologia do Domínio (e.g., <i>research_group</i>)
	has_slot	Identifica o slot do conceito (e.g., <i>group_name</i>)
	has_allowed_summarization_operation	Descreve quais operações da classe <i>Summarization</i> são permitidas sobre o atributo

Classe	Atributo	Descrição
		(e.g., <i>AVG</i> , <i>SUM</i> , <i>MAX</i> , <i>MIN</i> , etc.)
	is_additive	Descreve se o atributo pode ser sumarizado (e.g., somado, computada a média, etc.)
	has_presentation_format	Expressa a máscara de formatação para apresentação em relatórios (e.g., #.##9,99)
DB_Collection_Join	has_coljoin_first_collection	Identifica a primeira <i>DB_Collection</i> .
	has_coljoin_first_collection_key	Identifica um atributo da primeira <i>DB_Collection</i> que será utilizado para unir as <i>DB_Collections</i>
	has_coljoin_second_collection	Identifica a segunda <i>DB_Collection</i> .
	has_coljoin_second_collection_key	Identifica um atributo da segunda <i>DB_Collection</i> .
Dimension	has_dimension_name	Descreve o nome da dimensão
	has_presentation_slots	Determina a lista de <i>slots</i> que são utilizados para apresentar uma instância ao usuário
	has_dimension_top_agg	Determina se podem ser apresentados agrupadores para os níveis existentes na dimensão (e.g., todas as regiões, todos os sexos, etc)
	has_dimension_hierarchy	Determina as hierarquias existentes na dimensão (e.g., geografia, área de conhecimento)
Hierarchy	has_hierarchy_name	Descreve o nome da hierarquia
	has_hierarchy_level	Relaciona os níveis existentes em uma hierarquia (e.g., região, estado e município)
Level	has_level_name	Descreve o nome do nível de uma hierarquia.
	is_unique_member	Determina se o nível pode ter múltiplos valores.
	has_level_attribute	Identifica a instâncias de <i>DB_Attribute</i> relacionada ao nível corrente.
Dimension_Usage	has_usage_dimension	Identifica uma instância da classe <i>Dimension</i>
	has_usage_collection_join	Identifica uma coleção de instâncias da classe <i>DB_Collection_Join</i> visando mostrar como a dimensão identificada em <i>has_usage_dimension</i> será ligada a uma tabela de fatos ou à unidade de análise.
Fact	has_fact_dimension_usage	Identifica as instâncias da classe <i>Dimension_Usage</i> utilizadas para associar as <i>Dimensions</i> da tabela de fatos.
	has_fact_measure	Identifica as instâncias de <i>DB_Attribute</i> que correspondem às medidas da tabela de fatos.

Um fragmento do código OCML utilizado na descrição do mapeamento entre a tabela de dimensão *dim_grupo* e o conceito *research_group* da

Ontologia do Domínio para gestão de C&T é apresentado na Listagem 2. Este mapeamento é realizado por quatro instâncias da classe *DB_Attribute* e uma instância da classe *Dimension*. A partir dessas definições, o usuário pode navegar sobre os dados desta tabela e das tabelas de fato correspondentes através do conceito *research_group* da Ontologia do Domínio. Este processo de navegação é ilustrado no Capítulo 7.

Um exemplo de definição de hierarquia, de níveis de hierarquia e do mapeamento do relacionamento entre a dimensão *dim_geografia* e a tabela de fatos *fato_grupo* é apresentado de forma parcial na Listagem 3 a seguir.

Listagem 2 – Definição das instâncias utilizadas para mapear a dimensão `dim_grupo` ao conceito `research_group`

```
(def-instance dimension dim_grupo dimension
  ((has_collection_name "dim_GRUPO")
   (has_pk_attribute att_dim_grupo_seq_id_grupo)
   (has_presentation_slots att_dim_grupo_nme_grupo)
   (has_attribute att_dim_grupo_nme_grupo
                  att_dim_grupo_ano_formacao
                  att_dim_grupo_nro_id_grupo)))

(def-instance att_dim_grupo_nme_grupo db_attribute
  ((has_attribute_name "NME_GRUPO")
   (has_concept research_group)
   (has_slot has_group_name)
   (is_additive "false")))

(def-instance att_dim_grupo_seq_id_grupo db_attribute
  ((has_attribute_name "SEQ_ID_GRUPO")
   (has_concept research_group)
   (has_slot has_group_id)
   (is_additive "true")
   (has_allowed_additive_summarization Summarization_Count
                                       Summarization_Count_Distinct)))

(def-instance att_dim_grupo_nro_id_grupo db_attribute
  ((has_attribute_name "NRO_ID_GRUPO")
   (has_concept research_group)
   (has_slot has_group_code)
   (is_additive "true")
   (has_allowed_additive_summarization Summarization_Count
                                       Summarization_Count_Distinct)))

(def-instance att_dim_grupo_year_formation db_attribute
  ((has_attribute_name "ANO_FORMACAO")
   (has_concept research_group)
   (has_slot has_group_creation_year)
   (is_additive "true")
   (has_allowed_additive_summarization Summarization_Avg
                                       Summarization_Min Summarization_Max)))
```

Listagem 3 – Fragmento da ontologia BI ilustrando uma hierarquia e a ligação entre a dimensão dim_grupo e tabela de fatos fato_grupo

```
(def-instance hierarchy_geography hierarchy
  ((has_hierarchy_name "Geografia")
   (has_hierarchy_level level_region level_state level_city)))

(def-instance level_region level
  ((has_level_name "Região")
   (is_unique_member "false")
   (has_level_attribute att_di_geografia_nme_regiao)))

(def-instance dimension_usage_fato_grupo_dim_grupo dimension_usage
  ((has_usage_dimension dimension_dim_grupo)
   (has_usage_collection_join
    db_collection_join_fato_grupo_dim_grupo)))

(def-instance db_collection_join_fato_grupo_dim_grupo
  db_collection_join
  ((has_coljoin_first_collection fact_fato_grupo)
   (has_coljoin_first_collection_key att_fato_grupo_seq_id_grupo)
   (has_coljoin_second_collection dimension_dim_grupo)
   (has_coljoin_second_collection_key att_dim_grupo_seq_id_grupo)))
```

5.3.2 Realizando cortes semânticos nas análises

Nesta pesquisa, utiliza-se o suporte para processamento de inferências proporcionado pelas ontologias da arquitetura para permitir ao usuário diferentes possibilidades de navegação e filtragem de dados em suas análises. A abordagem usada para apoiar a redefinição de consultas permite que o usuário navegue sobre as fontes de dados utilizando para tanto os conceitos definidos na Ontologia do Domínio. Os conceitos representados na Ontologia do Domínio são mapeados aos dados através da Ontologia BI, conforme apresentado na Seção anterior. Esse mapeamento permite que a semântica do negócio seja aplicada na busca de diferentes perspectivas sobre os dados durante o processo decisório.

Os usuários selecionam, entre as relações, expressões lógicas e regras definidas na Ontologia do Domínio e solicitam a sua execução. O resultado da inferência pode ser utilizado para filtrar os dados reunidos em uma análise ou para expandir o escopo dos dados. Este processo é implementado pelo módulo Gerenciador de Análises. O Gerenciador de Análises é apresentado na Seção 6.4.2, na qual são descritos os métodos deste módulo e a demonstração do uso da Ontologia BI e da Ontologia do Domínio no suporte à execução desses métodos.

5.3.3 Personalização da nomenclatura para a apresentação das informações

A Ontologia BI compreende ainda conceitos usados para apoiar a apresentação da estrutura das fontes de dados e dos resultados de uma consulta de acordo com o perfil do usuário. Cada usuário é identificado por uma instância da classe *User* e é associado a um conjunto de instâncias da classe *Role* que identificam os perfis de utilização e os níveis de permissão que os usuários terão nas ferramentas analíticas. As permissões englobam as tabelas que o usuário pode acessar, além dos níveis das dimensões, filtros semânticos e serviços que serão disponibilizados aos usuários através das ferramentas analíticas. Essas classes são descritas no Quadro 12 apresentado a seguir.

Quadro 12 - Descrição das classes utilizadas para suportar o controle de acesso dos usuários e a personalização da nomenclatura

Classe	Atributo	Descrição
User	has_user_login	Identificador de acesso do usuário
	has_user_role	Mantém a lista de <i>Roles</i> que identificam os perfis de utilização e níveis de acesso do usuário
	has_user_idiom	Identifica o idioma (<i>Idiom</i>) preferencial do usuário
Role	has_role_name	Identificador do perfil
	has_role_granted_db_element	Descreve as tabelas (<i>DB_Collection</i>) e atributos (<i>DB_Attribute</i>) que o perfil possui acesso
	has_role_granted_level	Identifica quais os níveis (<i>Level</i>) das hierarquias de dimensões que o perfil possui acesso
	has_role_granted_service	Identifica quais <i>Goals</i> e <i>Web services</i> poderão ser acessados pelo perfil
	has_role_granted_filter	Identifica as relações e filtros semânticos que poderão ser utilizados pelo perfil para navegação sobre as fontes de dados
Label	has_label_db_element	Identifica uma tabela (<i>DB_Collection</i>) ou um atributo (<i>DB_Attribute</i>)
	has_label_idiom	Identifica o idioma (<i>Idiom</i>) do rótulo
	has_label_user_role	Identifica à qual perfil de usuário (<i>Role</i>) o rótulo da tabela ou do atributo se destina
	has_label_description	Descreve o rótulo a ser apresentado

A relação *label-role-idiom* descrita na Listagem 4 é usada para permitir a inferência de um rótulo (*i.e.*, uma instância da classe *Label*) para um atributo da base de dados (*i.e.*, uma instância da classe *DB_Attribute*), conforme o perfil do usuário (*i.e.*, de acordo com as *Roles* associadas ao usuário). Assim, podem ser definidas formas de apresentação distintas para diferentes classes de usuário. Por exemplo, uma instância da classe *Label* poderia descrever o atributo “tot_pesquisadores” como “quantidade de especialistas” para a *Role* “empresário”. Outra instância da classe *Label* poderia descrever o mesmo atributo como “number of researchers” para a *Role* “researchers”. A descrição do atributo é obtida a partir da execução da relação *label-role-idiom* informando-se o nome da coluna desejada, além da *role* e do idioma do usuário. O resultado, conforme apresentado na Listagem 4, é retornado através da variável *label*.

A relação *presentation-attribute*, incluída na Listagem 4, é utilizada para apresentar a descrição do resultado de uma inferência. Essa relação é utilizada para apresentar ao usuário somente as propriedades identificadas pela propriedade *has_presentation_slots* da classe *Dimension* quando uma ou várias instâncias são retornadas através de uma inferência. Por exemplo, em vez de apresentar todas as propriedades da classe *university*, a relação *institution-competitor* mostrada na Listagem 1 poderia apresentar apenas a sigla e o nome das instâncias recuperadas, facilitando ao usuário a interpretação do resultado.

Listagem 4 - Definição OCML das relações *label-role-idiom* e *presentation-attribute*

```
(def-relation label-role_idiom (?column ?role ?idiom
?label)
:sufficient
  (and (has_attribute_name      ?att ?column)
        (has_label_db_element  ?db ?att)
        (has_label_user_role   ?db ?role)
        (has_label_idiom       ?db ?idiom)
        (has_label_description ?db ?label)))

(def-relation presentation-attribute (?concept ?slot)
:sufficient (and
  (has_concept ?att ?concept)
  (has_attribute ?dim ?att)
  (has_presentation_slots ?dim ?pres)
  (has_slot ?pres ?slot)
))
```

5.3.4 Descrevendo semanticamente as análises criadas pelos usuários

A Ontologia BI prevê um conjunto de conceitos que são utilizados para representar as análises definidas pelos usuários. Conforme ilustrado na Figura 15, a definição de uma análise compreende quais dimensões, medidas, filtros, privilégios e parâmetros foram relacionados pelo usuário.

O diagrama mostra uma tabela com as seguintes colunas e linhas:

Universidade	Grupo de Pesquisa	Total de Estudantes
UFSC	LED	200
	LIA	210
	LRV	190
UFSC Total		600
Total geral		600

As setas indicam:

- Filtros (vermelha):** Aponta para o campo 'Ano' com o valor '2002'.
- Medidas (amarela):** Aponta para a coluna 'Total de Estudantes'.
- Dimensões (verde):** Aponta para as colunas 'Universidade' e 'Grupo de Pesquisa'.

Figura 15 – Identificação dos elementos de uma análise

Dimensões correspondem aos cabeçalhos de linha definidos pelo usuário, os quais serão utilizados ainda para agrupar as medidas numéricas definidas na análise. No exemplo ilustrado na Figura 15, a medida corresponde ao número total de estudantes. As dimensões definidas são instituição e grupo de pesquisa. Dessa forma, serão apresentados os totais de estudantes alocados nos grupos de pesquisa e instituições desses grupos. Como filtro, foi definido que serão computados apenas os estudantes alocados nos grupos de pesquisa no ano de 2002. O filtro de ano, neste exemplo, foi definido como um parâmetro. Os parâmetros são usados para preencher valores de ligação na definição de filtros. Assim, cada vez que essa análise for executada, será solicitado ao usuário qual o ano desejado. Dessa maneira, uma análise pode ser utilizada para apresentar o número de estudantes em qualquer ano desejado, possibilitando que uma mesma análise possa atender a diferentes necessidades de usuários diversos na organização.

As análises definidas pelo usuário nas ferramentas analíticas ligadas à arquitetura SBI podem ser persistidas como instâncias da classe *Analysis*. As

classes previstas na Ontologia BI para representar as análises definidas pelos usuários são descritas no Quadro 13. A Listagem 5 apresenta um fragmento da Ontologia BI para representar como a análise da evolução do número de estudantes por grupo de pesquisa e instituição é explicitada.

As análises definidas pelo usuário nas ferramentas analíticas ligadas à arquitetura SBI podem ser persistidas como instâncias da classe *Analysis*. As classes previstas na Ontologia BI para representar as análises definidas pelos usuários são descritas no Quadro 13. A Listagem 5 apresenta um fragmento da Ontologia BI para representar como a análise da evolução do número de estudantes por grupo de pesquisa e instituição é explicitada.

Quadro 13 - Descrição das classes utilizadas para representar as análises definidas pelos usuários

Classe	Atributo	Descrição
Analysis	has_analysis_description	Descrição livre informada pelo usuário
	has_analysis_content	Mantém a lista de instâncias da classe <i>Analysis_Content</i> , as quais descreverão quais medidas e operações de sumarização foram relacionadas pelo usuário
	has_analysis_dimension	Descreve quais instâncias da classe <i>Dimension</i> formam as dimensões (cabeçalhos de linha) da análise
	has_analysis_filter	Identifica as instâncias da classe <i>Analysis_Filter</i> utilizadas para definir os filtros da análise
	has_analysis_parameter	Identifica os parâmetros que devem ser informados pelo usuário assim que a análise for executada. Corresponde a uma lista de instância da classe <i>Analysis_Parameter</i> .
	has_analysis_creator	Identificação do usuário que criou a análise. Corresponde a uma instância da classe <i>User</i> .
	has_analysis_allowed_user	Lista de usuários (instâncias de <i>User</i>) que podem executar a análise.
	has_analysis_allowed_role	Lista de grupos de usuários (instâncias de <i>Role</i>) que podem executar a análise.
Analysis_Content	has_content_measure	Define qual instância da classe <i>DB_Attribute</i> foi relacionada como medida (e.g., <i>att_fato_grupo_tot_estudante</i>)
	has_content_summarization	Define como a medida é sumarizada, através da identificação de uma instância classe <i>Summarization</i> (e.g., SUM, AVG, etc.)
Analysis_Dimension	has_dimension_attribute	Define qual instância da classe <i>DB_Attribute</i> foi relacionada como dimensão (e.g.,

Classe	Atributo	Descrição
		<i>att_dim_instituicao_sgl_instituicao</i>)
	has_dimension_position	Identifica a posição do atributo na lista de todos os atributos que formam as dimensões da análise
Analysis_Filter	has_filter_attribute	Identifica qual instância da classe <i>DB_Attribute</i> foi utilizada na definição do filtro (e.g., <i>att_dim_grupo_ano_formacao</i>)
	has_filter_operator	Identifica o operador utilizado no filtro. O operador de filtro corresponde a uma instância da classe <i>DB_Operation</i> (e.g., <i>equal</i> , <i>greater_than</i> , etc.)
	has_filter_value	Descreve o valor ao qual o atributo será confrontado pelo operador definido. O valor pode ser uma literal, valor numérico, um identificador de uma instância da Ontologia do Domínio (e.g., "2004", <i>person_15</i> , etc) ou uma instância de <i>Analysis_Parameter</i>
Analysis_Parameter	has_parameter_value	Identifica o(s) valor(es) entrado(s) pelo usuário que serão atribuídos a uma instância de <i>Analysis_Filter</i>
	is_parameter_single_value	Identifica se o parâmetro poderá receber vários valores ou não.

Listagem 5 - Definição OCML da análise do número de estudantes por instituição e grupos de pesquisa

```
(def-instance groups_productivity Analysis
((has_analysis_description "Research Groups Publication Evolution")
 (has_analysis_content groups_productivity_content)
 (has_analysis_dimension groups_productivity_dim_1
 groups_productivity_dim_2)
 (has_analysis_filter groups_productivity_filter_1
 groups_productivity_filter_2
 groups_productivity_filter_3
 groups_productivity_filter_4)
 (has_analysis_parameter groups_productivity_parameter_1)
 (has_analysis_creator user_denilson)
 (has_analysis_allowed_user user_denilson)
 (has_analysis_allowed_role role_researcher)))

(def-instance groups_productivity_content Analysis_Content
((has_content_measure att_ft_producao_qtd_artigo_anais)
 (has_content_summarization summarization_sum)))

(def-instance groups_productivity_dim_1 Analysis_Dimension
((has_dimension_attribute att_di_instituicao_sigla)
 (has_dimension_position "1")))

(def-instance groups_productivity_dim_2 Analysis_Dimension
((has_dimension_attribute att_dim_grupo_nme_grupo)
 (has_dimension_position "2")))

(def-instance groups_productivity_filter_1 Analysis_Filter
((has_filter_attribute att_dim_grupo_seq_id_grupo)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_grupo)))

(def-instance groups_productivity_filter_2 Analysis_Filter
((has_filter_attribute att_di_instituicao_seq_id_inst)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_inst)))

(def-instance groups_productivity_filter_3 Analysis_Filter
((has_filter_attribute att_di_tempo_seq_id_tempo)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_tempo)))

(def-instance groups_productivity_filter_4 Analysis_Filter
((has_filter_attribute att_di_tempo_ano)
 (has_filter_operator db_operation_equal)
 (has_filter_value groups_productivity_parameter_1)))

(def-instance groups_productivity_parameter_1 Analysis_Parameter
((has_parameter_value "2002")
 (is_parameter_single_value "false")))
```

Os construtores descritos no Quadro 13 permitem que análises previamente elaboradas possam ser automaticamente localizadas e recomendadas de acordo com o contexto das explorações do usuário em

ferramentas analíticas. Essa recomendação automática de análises permite que sejam oferecidas automaticamente ao usuário diferentes perspectivas para a tomada de decisão. Trata-se de uma característica até o momento não suportada pelas ferramentas analíticas atuais. A recomendação automática de análises leva em consideração os sinônimos e hipônimos dos conceitos utilizados na definição das dimensões, dos filtros e dos parâmetros da análise atual e das análises previamente definidas. Os métodos para comparação entre análises e para sugestão de alternativas de análises previamente confeccionadas para o tomador de decisão são apresentados na Seção 6.4, quando da discussão sobre o módulo Gerenciador de Análises.

A descrição semântica da análise é utilizada ainda para, em conjunto com as informações sobre o perfil do usuário, suportar a recomendação de serviços, filtros semânticos e relações de acordo com a análise definida. Este processo de recomendação é também apresentado na Seção 6.4.

5.3.5 Descrevendo temas de análise

Verificam-se na literatura várias metodologias para o desenvolvimento de soluções de BI (INMON, 2002; KIMBALL et al., 1998, 2002; SELL, 2001). Essas metodologias prevêem etapas para o levantamento dos requisitos da solução, modelagem do *Data warehouse*, construção das ferramentas ETL e construção das ferramentas analíticas. As metodologias tradicionais, no entanto, não abordam métodos para facilitar a apresentação dos dados reunidos no DW para a comunidade de usuários. Usuários constantemente têm de recorrer à equipe técnica para extrair informações do DW por que não conseguem compreender quais dados existem no DW e como podem extraí-los.

A equipe de desenvolvimento de projetos de BI do Instituto Stela (2006), em que o autor atua na pesquisa e desenvolvimento de soluções de BI desde 1998, percebendo a dificuldade dos usuários em compreender a abrangência dos dados reunidos em seus repositórios de dados, desenvolveu uma metodologia para mapear os dados reunidos nesses repositórios em assuntos. Essa metodologia é descrita em detalhes em Gonzaga (2005) e em Tissot (2004).

Seguindo as etapas dessa metodologia, conforme ilustra a Figura 16, primeiramente os elementos de dados do DW são divididos em assuntos. Um assunto corresponde a uma coleção de tabelas de fatos e suas respectivas dimensões, criados para reunir os dados sobre um determinado processo do negócio. Cada assunto pode compreender um ou vários tópicos, denominados unidades de análise. Uma unidade de análise corresponde a uma unidade de informação, mantida por uma (comumente) ou por várias tabelas de fatos. Cada unidade de análise reunirá várias unidades de conteúdo, que são as medidas pelas quais são mensuradas as unidades de análise. As unidades de conteúdo correspondem às medidas das tabelas de fatos e respectivas operações de sumarização ou ainda a outros atributos utilizados para apresentação das unidades de conteúdo. Cada unidade de análise pode conter ainda uma ou várias unidades de filtros, que são atributos descritivos mantidos pelas dimensões do DW, utilizados para permitir, entre outras, operações de *slice* e *drill* sobre uma unidade de análise (GONZAGA, 2005; TISSOT, 2004).

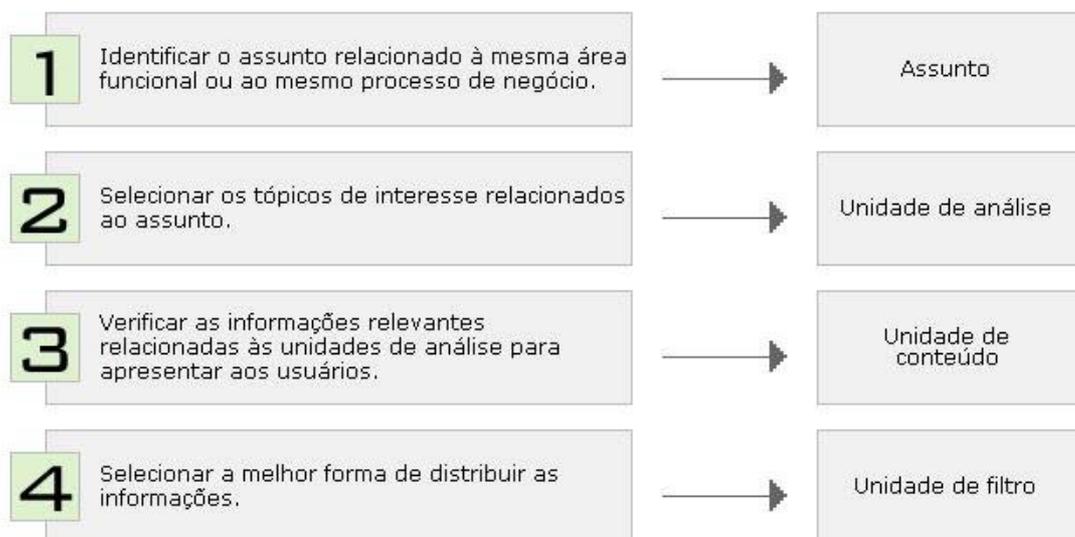


Figura 16 - Etapas da Metodologia do Instituto Stela para Projeto da Área de Apresentação de Soluções de BI.

Fonte: Gonzaga (2005)

A Figura 17 apresenta uma tela da ferramenta analítica Plano Tabular do Diretório de Grupos de Pesquisa do Brasil (DGP). O Plano Tabular é um instrumento concebido para permitir que a comunidade científica possa facilmente obter dados sobre as atividades de pesquisa e desenvolvimento conduzidas pelos grupos de pesquisa brasileiros. No Plano Tabular, o assunto

corresponde ao *data mart* do Diretório de Grupos de Pesquisa. As unidades de análise correspondem a “Grupos de Pesquisa”, “Linhas de Pesquisa”, “Pesquisadores”, “Estudantes”, “Técnicos”, “Produção C,T&A (Científica, Tecnológica e Artística)” e “Empresas/Grupos”. No caso da unidade de análise Produção C,T&A, são relacionadas a unidade de conteúdo total de autores e várias outras que representam totalizações por tipo de produção. Essa unidade de análise possui diversas unidades de filtro, tais como ano da publicação, instituição de lotação do autor da produção, entre outras. As unidades de filtro permitem agrupar ou filtrar as informações sobre a unidade de análise selecionada.

The screenshot shows the 'Plano Tabular' web application interface. The main content is a table titled 'Produção bibliográfica segundo instituição para pesquisadores, 2000-2003. Censo 2004.' The table has columns for 'Instituição', 'Total de autores', 'Artigos completos publicados em periódicos especializados' (with sub-columns for national and international circulation), 'Trabalhos completos publicados em anais de eventos', 'Livros ou capítulos de livro publicados' (with sub-columns for books and chapters), 'Outras publicações bibliográficas', and 'Resumos de trabalhos publicados em' (with sub-columns for specialized periodicals and event analyses). The rows list various institutions like ANHEMBI, MORUMBI, ANIS, APS, CBMDF, CBPF, etc.

Annotations on the screenshot include:

- A red box labeled 'Unidades de Análise' pointing to the navigation menu at the top.
- A blue box labeled 'Unidades de Conteúdo' pointing to the sub-columns of the table headers.
- A green box labeled 'Unidades de Filtro' pointing to the filter options on the left side of the interface.

Instituição	Total de autores	Artigos completos publicados em periódicos especializados		Trabalhos completos publicados em anais de eventos	Livros ou capítulos de livro publicados		Outras publicações bibliográficas (3)	Resumos de trabalhos publicados em	
		Circulação nacional (1)	Circulação internacional (2)		Livros	Capítulos de livros		Periódicos especializados	Anais de eventos
ANHEMBI MORUMBI	28	57	9	33	7	31	23	0	72
ANIS	2	21	3	1	4	10	32	0	3
APS	54	28	61	39	1	8	40	10	121
CBMDF	2	1	15	1	0	0	0	0	18
CBPF	162	315	1.946	607	30	66	0	0	933
CEBPF	69	192	40	88	31	168	218	1	54
CEBPF	27	153	8	46	7	65	170	18	84
CEBPF	7	31	54	44	0	1	9	6	120
CEFET/AM	7	2	1	8	2	0	8	0	4
CEFET/BA	38	20	23	63	1	14	44	2	95
CEFET/CAMPOS	24	24	7	50	1	3	138	1	20
CEFET/CF	39	28	8	160	4	17	27	4	77

Figura 17 - Apresentação das Unidades de Análise, Filtros e de Conteúdo na Ferramenta Plano Tabular

Visando atender projetos de ferramentas analíticas estruturadas segundo a metodologia do Instituto Stela, foi introduzida na Ontologia BI uma classe para descrever as unidades de análise. As unidades de conteúdo e de filtro são definidas pelos conceitos *DB_Attribute*, *Dimension*, *Dimension_Usage* e por outros descritos nas seções anteriores. Até o momento, essas definições eram embutidas no próprio código das ferramentas produzidas pelo Instituto Stela, dificultando a modificação dessas definições em face às mudanças de

requisitos e inviabilizando a reutilização das ferramentas em domínios de aplicação diferentes.

Busca-se com a Arquitetura SBI apoiar a construção de ferramentas analíticas guiadas pelas definições da Ontologia BI, permitindo a reutilização do mesmo código para atender a demandas de diferentes tomadores de decisão. Dessa forma, as definições dos temas de análise podem ser alteradas sem a necessidade de modificar a sua codificação, além de permitir diferentes modalidades de navegação sobre os dados através dos relacionamentos semânticos entre os conceitos do domínio, conforme descrito nas seções seguintes. A classe *Analysis_Unit*, utilizada para descrever os temas de análise, é apresentada no Quadro 14 a seguir. Esta classe está associada às demais classes utilizadas para descrever tabelas de fato, dimensões e atributos, as quais foram descritas nas seções anteriores.

Quadro 14 - Descrição da classe *Analysis_Unit*

Atributo	Descrição
has_analysis_unit_code	Código identificador da unidade de análise
has_analysis_unit_name	Descreve o nome da unidade de análise (e.g., Produção Científica e Tecnológica)
has_analysis_unit_dimension_usage	Descreve quais instâncias da classe <i>Dimension</i> formam as dimensões (cabecinhos de linha) da análise através de uma lista de instâncias da classe <i>Dimension_Usage</i> .
has_analysis_unit_measure	Mantém a lista de <i>DB_Attributes</i> que correspondem às medidas e operações de sumarização associadas a unidade de análise
has_analysis_unit_allowed_user	Lista de usuários (instâncias de <i>User</i>) que terão acesso à unidade de análise.
has_analysis_unit_allowed_role	Lista de grupos de usuários (instâncias de <i>Role</i>) que terão acesso à unidade de análise.

5.4 Ontologia de Serviços

Na arquitetura SBI, construtores semânticos são também aplicados na descrição de *Web services* (WS) utilizados para estender as funcionalidades exploratórias (RF4) e para suportar a composição de serviços (RF5).

As ferramentas analíticas correntes não possuem maneiras flexíveis para a extensão ou modificação das funcionalidades exploratórias. Normalmente, os usuários contam com um número limitado de funcionalidades

(e.g., *drill*, *slice*, etc), as quais normalmente não podem ser modificadas para suportar algum tipo de processamento específico demandado pelo tomador de decisão. Em algumas ferramentas, é oferecido o recurso de extensão, mas, para tanto, o desenvolvedor deve utilizar uma linguagem específica suportada pelo fabricante, o que impede a reutilização de código e torna a customização de difícil execução.

Da mesma forma, o acesso a dados remotos por parte das arquiteturas de BI atuais é feito apenas em tempo de construção do *Data warehouse*. Os desenvolvedores da área de estágio devem desenvolver rotinas específicas a serem incluídas no processo de ETL para realizar a carga dos dados mantidos em repositórios externos. Essas rotinas normalmente demandam um grande esforço de desenvolvimento por se tratar de um desenvolvimento de baixo nível e específico para cada necessidade. No entanto, a conexão com repositórios de dados remotos é de extrema importância em vários domínios de processos decisórios, em que, por exemplo, seja demandado o acesso a bases de órgãos reguladores ou de parceiros comerciais, de bases de indicadores macroeconômicos, ou a qualquer outra base de interesse ao tomador de decisão.

Conforme apresentado na Seção 3.4, WS permitem a reutilização de código e o acesso a conteúdos e aplicações remotas, independentemente da maneira que o conteúdo remoto está organizado e como a aplicação remota foi desenvolvida. WS são aplicados na arquitetura SBI para a busca de dados adicionais (internos ou externos à organização) para complementar os dados manipulados em uma análise. WS podem ainda executar qualquer tipo de transação usando os dados de uma análise como entrada, garantindo flexibilidade para o processamento analítico.

A Ontologia de Serviços é aplicada na descrição dos WS existentes na organização ou disponíveis na Web visando apoiar a localização de serviços que possam estar associados ao contexto de uma análise. Esse processo de recomendação automática de serviços leva em consideração os conceitos da ontologia do domínio utilizados pelo usuário na definição das dimensões, medidas e filtros de suas análises. Os conceitos usados pelo usuário na definição das suas análises são comparados com os conceitos empregados

para descrever os WS. Em caso de *match* entre os conceitos da análise e dos WS, é oferecida uma lista de serviços para o usuário. O processo de localização de serviços é apresentado em detalhes na Seção 6.3.

Serviços como os descritos anteriormente podem executar tarefas tais como extração de dados de repositórios externos, visando complementar os dados reunidos em uma análise (e.g., apresentar o conceito dos cursos de uma dada instituição a partir de um serviço disponibilizado pelo órgão que avalia as instituições de ensino no Brasil). Demandas por acesso a bases remotas rotineiramente aparecem após a concepção das ferramentas ETL e do *Data warehouse*. Com WS, pode-se projetar a qualquer momento o acoplamento de uma fonte de dados remota a uma solução de BI, sem a necessidade de ter-se mapeado essa demanda no momento do projeto do *Data warehouse*.

A Ontologia de Serviços visa ainda apoiar a composição de WS quando for necessário utilizar não apenas um, mas vários WS organizados de forma que o resultado do processamento de um WS seja usado como valor de entrada para outro(s) WS, formando fluxos decisórios com estágios de processamento definidos pelo próprio usuário. Esse recurso é apresentado em detalhes nas próximas seções.

5.4.1 Descrevendo *Web services* na Ontologia de Serviços

A Ontologia de Serviços modela os conceitos usados para integrar *Web services* semânticos (WSS) à arquitetura. Para adicionar um serviço à arquitetura, torna-se necessário criar uma anotação semântica na Ontologia de Serviços para o código que se deseja incorporar à arquitetura. Na versão corrente, essa ontologia corresponde ao conjunto de ontologias utilizadas pelo *framework* IRS-III (DOMINGUE et al., 2004a) e a um conjunto de conceitos adicionais criados para suportar a composição de WSS. A anotação semântica de serviços pode ser feita através da ferramenta *IRS Browser* (DOMINGUE et al., 2004a), que disponibiliza um conjunto de interfaces gráficas que facilitam a anotação de um código Java, Lisp ou de um Web service. Essa anotação resulta em um Web service semântico que pode ser utilizado durante os processos de análise em ferramentas analíticas. Os motivos que levaram à

escolha do framework IRS-III para representação da versão inicial da Ontologia de Serviços foram apresentados na Seção 3.4.3.

No entanto, como a arquitetura foi projetada para possuir um baixo nível de acoplamento entre os seus módulos, seria possível subscrever a outra ontologia para descrição de WS, como OWL-S (OWL-S, 2002). Anotações OWL-S podem ainda ser traduzidas e importadas para o IRS-III *Server* através do tradutor OWL-S–WSMO (HAKIMPOUR et al., 2004).

As ontologias do framework IRS-III correspondem a uma extensão do padrão WSMO D2v02 (WSMO, 2004). As adições do framework IRS-III ao padrão WSMO são necessárias para o suporte da seleção e invocação automática de WS que possam atender a uma meta definida pelo usuário (*i.e.*, invocação automática de WS que implementam um dado *Goal* descrito na ontologia). Para tanto, a descrição de *Goals* e WS nas ontologias do framework IRS-III prevêem a descrição de entradas e saídas (*i.e.*, *inputs* e *output roles*), tanto para *Goals* quanto para WS. Assim, um *Goal* no IRS-III possui os seguintes slots extras: *has-input-role*, *has-output-role*, *has-input-role-soap-binding* e *has-output-role-soap-binding* (DOMINGUE et al., 2004a). As entradas e saídas dos *Goals* e WS correspondem a conceitos do domínio representados na Ontologia do Domínio da arquitetura SBI (SELL et al., 2004b).

Goals são ligados a WS através de *Mediators*, mais especificamente os *Mediators* definidos no slot *used-mediator* da classe *Capability* de um dado WS. Nesse caso, um tipo específico de *Mediator* é utilizado, o *wg-mediator*, que liga o *Goal* identificado pelo slot *source* a um WS. Os *Mediators* são utilizados ainda para transformar a saída de um WS e para gerar a entrada necessária para um segundo WS (*i.e.*, através de um *ww-mediator*), para transformar os valores de entrada de um *Goal* para a entrada do WS que executará este *Goal* (*i.e.*, através de um *gw-mediator*), para executar transformações entre conceitos definidos na Ontologia do Domínio (*i.e.*, através de um *oo-mediator*), ou para transformar a saída de um WS para a saída do *Goal* correspondente (*i.e.*, através de um *wg-mediator*) (DOMINGUE et al., 2004a; WSMO, 2004).

No exemplo apresentado no Quadro 15, são descritos dois serviços criados para realizar a conversão de moedas, um para moedas europeias e

outro para moedas não europeias. De acordo com a moeda especificada como entrada para o *Goal exchange-rate-goal*, é automaticamente selecionado um WS para a executar a conversão.

Quadro 15 - Descrição das classes utilizadas para representar temas de análise

Classe	Descrição OCML
<p>Goal: o goal deste exemplo possui dois input roles, <i>has_source_currency</i> e <i>has_target_currency</i> ambos do tipo <i>currency</i> e o output role <i>has_exchange_rate</i> do tipo <i>positive-number</i>. <i>Currency</i> e <i>Positive-number</i> são conceitos descritos na Ontologia do Domínio.</p>	<pre>exchange-rate-goal (goal) has-input-role :value has_source_currency :value has_target_currency has-output-role :value has_exchange_rate has_source_currency :type currency has_target_currency :type currency has_exchange_rate :type positive-number</pre>
<p>Mediator: o <i>gw-mediator exchange-rate-mediator</i> especifica que o <i>source-component</i> é o <i>exchange-rate-goal</i>. Não é especificado um <i>target-component</i> porque este <i>Mediator</i> é usado por dois WS (especificado na classe <i>Capability</i> daqueles serviços). O valor do slot <i>has-mediation-service</i> é o <i>Goal exchange-rate-mediation-goal</i>. Quando este <i>Goal</i> é invocado, os valores de entrada são transformados por um WS associado com <i>exchange-rate-mediation-goal</i> em um formato aceitável pelo WS que será ligado ao <i>exchange-rate-goal</i>. Neste exemplo, é utilizado um WS que transforma os símbolos das moedas (e.g., '\$' e '£') para o nome da moeda (e.g., <i>dollar</i>, <i>pound</i>).</p>	<pre>exchange-rate-mediator (gw-mediator) has-source-component :value exchange-rate-goal has-mediation-service :value exchange-rate-mediation-goal</pre>
<p>Web Services: duas descrições semânticas foram criadas para o <i>european</i> e o <i>non-European exchange-rate-web-services</i>. A definição dos WS identifica a <i>Capability</i> e a <i>Interface</i> de cada WS.</p>	<pre>european-exchange-rate-web-service (web-service) has-capability :value european-exchange-rate-capability has-interface :value european-exchange-rate-web-service-interface</pre>
<p>Capability: a definição dos WS <i>european</i> e <i>non-European exchange-rate-web-services</i> são herdadas de <i>exchange-rate-capability</i>. Esta especifica um <i>Mediator</i>, <i>exchange-rate-mediator</i>, que liga o WS ao <i>Goal exchange-rate</i>. <i>European-exchange-rate-capability</i> contém uma função de validação (em OCML) que especifica que o WS associado deve ser selecionado somente se os valores para <i>has_source_currency</i> e para <i>has_target_currency</i> correspondem a instâncias de <i>european-currency</i>.</p>	<pre>exchange-rate-capability (capability) used-mediator :value exchange-rate-mediator european-exchange-rate-capability (exchange-rate-capability) has-assumption :value (kappa (?goal) (and (european-currency (wsmo-role-value ?goal 'has_source_currency)) (european-currency (wsmo-role-value ?goal 'has_target_currency))))</pre>
<p>Interface: a interface é especificada quando um WS ou um código Java ou</p>	<pre>european-exchange-rate-web-service-interface (interface) ((has-choreography</pre>

Classe	Descrição OCML
Lisp são publicados sobre uma descrição de WS. A publicação é realizada através do IRS-III Server, o qual cria a descrição da coreografia do serviço, apontando para o local onde o código a ser executado está instalado.	<pre> :value exchange-rate-choreography) (has-orchestration :value exchange-rate-orchestration)) european-exchange-rate-web-service-choreography (choreography) ((has-web-service-host :value "150.162.50.1") (has-web-service-port :value 3001) (has-web-service-location :value "/soap")) </pre>

Na arquitetura SBI, a Ontologia do Domínio descreve os conceitos do negócio utilizados na Ontologia de Serviços para a descrição dos *input* e *output roles*, além de fornecer os conceitos empregados para a descrição das propriedades não-funcionais de WSS. As descrições semânticas dos WSS são mantidas na Ontologia de Serviços da arquitetura SBI para serem integradas aos demais componentes da arquitetura. Dessa forma, através dos métodos implementados pelo Gerenciador de Serviços da arquitetura SBI, torna-se possível localizar WSS através de suas descrições semânticas, bem como compor serviços para a realização de tarefas complexas e executar WSS. O módulo Gerenciador de Serviços é apresentado na Seção 6.3.

5.4.2 Suporte para Composição de Serviços

A Ontologia de Serviços provê ainda construtores semânticos para representar composições de serviços. O recurso de composição de serviços no contexto do processo de tomada de decisão é introduzido de forma inédita pela Arquitetura SBI. Esse recurso visa apoiar a composição de WSS quando da necessidade de se utilizar não apenas um, mas vários serviços para estender uma análise. Assim, WSS são organizados de forma que o resultado do processamento de um WSS seja utilizado como valor de entrada para outro(s) WSS ou para operadores condicionais, esses últimos utilizados para definir o caminho a ser seguido pelo fluxo de execução de acordo com condições definidas pelos usuários.

Desse modo, a arquitetura SBI oferece ao tomador de decisão a possibilidade de montar fluxos de processamento de informação que iniciam com os dados resultantes de uma análise e passam por estágios de processamento definidos pelo próprio usuário, onde cada etapa corresponde à execução de um WS. As descrições das composições de WSS feitas pelos usuários são armazenadas na Ontologia de Serviços, visando suportar a orquestração (*i.e.*, execução) de composições.

A integração do recurso de composição de WSS em ferramentas analíticas permite a definição de análises flexíveis e poderosas, em que WS implementados dentro ou fora da organização são utilizados para, por exemplo, trazer dados adicionais ao contexto da análise, transformar os dados da análise em informação ou ainda realizar qualquer outro tipo de transação utilizando os dados da análise como entrada. Um exemplo de transação poderia ser a diminuição da margem de lucro de um produto após verificar a queda no volume de vendas desse produto.

Além disso, essa descrição é empregada para manter as abordagens utilizadas pelos tomadores de decisão na análise de seus negócios e para a conversão dos *insights* obtidos em ações sobre o negócio. Dessa forma, cria-se uma sistemática para aquisição de conhecimento tácito dos tomadores de decisão. Cada composição definida por usuários na Arquitetura SBI pode ser mantida na Ontologia de Serviços. Esse conhecimento poderá ser utilizado como referência nos processos de tomada de decisão subseqüentes. Assim, os passos definidos nas composições estabelecidas por um dado usuário para a interpretação e transformação de dados podem ser utilizados parcial ou integralmente por outros usuários.

Nesta Seção são apresentados os construtores semânticos utilizados para a definição de composições através de um modelo guiado por fluxo de dados. O modelo prescreve o uso de *Mediators* para descrever o fluxo de dados entre os elementos de uma composição. O modelo de composição é definido por dois tipos de componentes, nominalmente componentes de controle (*i.e.*, *Control Components*) e componentes de serviço (*Service Components*) (SELL et al., 2004b; HAKIMPOUR et al., 2005). A hierarquia de componentes do modelo é ilustrada na Figura 18.

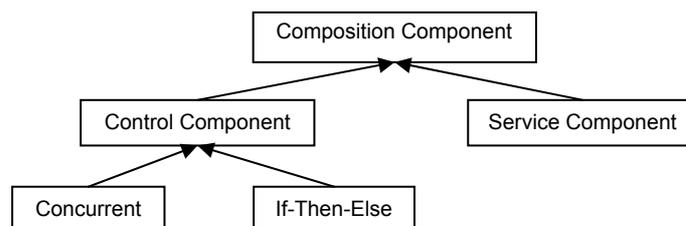


Figura 18 - Hierarquia de componentes no modelo de composição de WSS da Arquitetura SBI

O componente *Service Component* corresponde aos serviços que farão parte da composição, os quais correspondem a *Goals* ou WS descritos na Ontologia de Serviços. Os componentes de controle permitem definir como os *Service Components* serão executados. Até o momento, dois componentes dessa categoria foram definidos no modelo, o operador *Concurrent* e o operador *If-Then-Else*. O operador *Concurrent* define que *Service Components* adicionados à composição serão executados tão logo esses tiverem suas entradas preenchidas pelo fluxo de execução. O operador *If-Then-Else* descreve uma condição a ser avaliada e aponta um *Service Component* a ser executado caso a condição seja verdadeira e, opcionalmente, um *Service Component* a ser executado caso a condição seja falsa.

A seguir é apresentada a formalização do modelo utilizado para representar composições e a semântica correspondente. Um modelo adequado para composição de serviços deve conter três conjuntos de informação, como se segue:

- informação sobre *Component Services*;
- informação sobre os operadores de controle de execução ou *Control Components*; e
- informação sobre o intercâmbio de dados entre os serviços formando o fluxo de dados da composição.

Os serviços que fazem parte de uma composição podem ser estaticamente ligados à composição como WS ou descritos como *Goals* e, por conseguinte, ligados à composição dinamicamente em tempo de execução da composição, através de um processo de descoberta suportado pelo IRS-III Server (DOMINGUE et al., 2004a; SELL et al., 2004b). As definições a seguir descrevem como os componentes são relacionados na composição e como os dados são repassados entre cada componente de uma composição.

Definição: Uma composição O é uma tupla bidimensional $\langle T, \Delta T \rangle$ onde (SELL et al., 2004b; HAKIMPOUR et al., 2005):

T é uma árvore com raiz (WIKIPEDIA, 2004a) chamada árvore de composição com nós XT , arestas ET e componente raiz $\chi_0 \in XT$. $ST \subseteq XT$ é o

conjunto de todas as folhas em T e $CT \subset XT$ é o conjunto de todos os vértices internos ($\neg \exists \chi \in XT : \chi \in S \wedge \chi \in C$). Os membros de XT , ST e CT são chamados *Composition Components*, *Service Components* e *Control Components*, respectivamente. Para cada *Control Component* $c \in CT$, um e apenas um do seguinte é verdadeiro: *Concurrent*(c) ou *IfThenElse*(c). Os filhos de um *Composition Component* $c \in CT$ são denominados *Children*(c).

ΔT é um grafo direcional (WIKIPEDIA, 2004a) chamado fluxo de dados definido em T , com vértices $i\Delta$, $o\Delta$ e membros de XT (i.e., *Composition Components* de T) e as arestas $B\Delta$, onde $B\Delta \subseteq \{ \langle si, tj \rangle : (si \in ST \vee si = i\Delta), (tj \in XT \vee tj = o\Delta) \}$. Um membro $B\Delta$ é uma tupla binária chamada ligação entre si e tj , onde si é a fonte e tj é o destino da ligação.

Service Components (ST) em uma composição representam um *Goal* ou um *WS*. *Control Components* (CT) provêm a capacidade de definir o controle do fluxo de ordem de execução. *Control Components* podem ser do tipo *Concurrent* ou *If-Then-Else*. A semântica funcional de um *Control Component* (i.e., o comportamento do mecanismo de orquestração associado com cada *Control Component*) depende do seu tipo. $B\Delta$ representa um conjunto de ligações que descreve o fluxo de dados entre componentes. O mecanismo de orquestração irá executar um *Mediator* para cada ligação.

A Figura 19 ilustra uma composição formada com os construtores da Ontologia de Serviços. Esta composição descreve uma seqüência de serviços que são executados para avaliar indicadores gerenciais de uma dada instituição, visando identificar os problemas correntes e as possíveis causas. Os WSS incluídos neste exemplo são adicionados ao controle de execução *Concurrent*. O primeiro serviço lista os indicadores que estão abaixo das metas definidas para uma dada instituição fornecida como parâmetro de entrada para a composição. Uma condição é estabelecida para guiar o fluxo de acordo com os indicadores retornados do primeiro WSS, de forma a identificar, através de outros WSS, mais informações sobre as possíveis causas do mau desempenho. Assim, se o indicador crítico identificado pelo primeiro WSS estiver relacionado ao índice de docentes com baixo nível de titulação, será executado um serviço que apresenta a lista de docentes com nível de

graduação abaixo de doutorado. Caso contrário, será executado um serviço que lista os departamentos da instituição que estejam com mal desempenho de acordo com o indicador crítico. Assumindo neste exemplo que o indicador poderia ser relacionado ao nível de produtividade, em seguida seria executado outro serviço que apresentaria os departamentos com maior queda em volume de publicações nos últimos 24 meses.

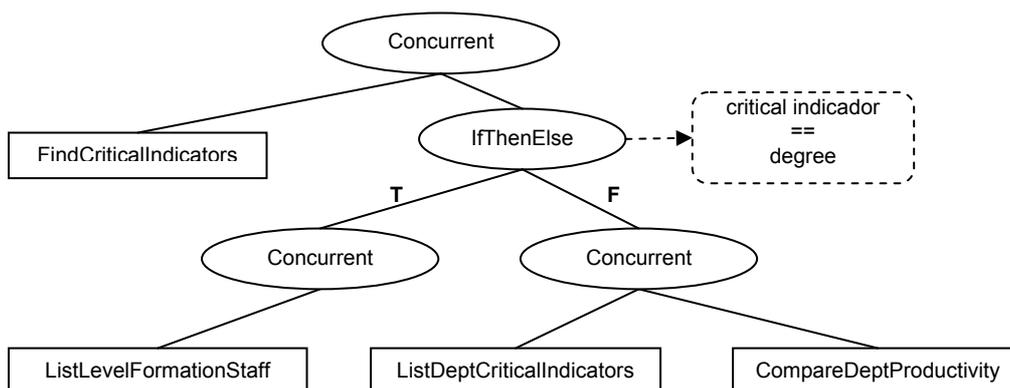


Figura 19 - Exemplo de composição de serviços para uma análise de indicadores de instituições

A definição do construtor usado para estabelecer a raiz de uma composição bem como a instância referente à composição do exemplo da Figura 19 são apresentados na Listagem 6. Todas as listagens referentes à semântica do modelo de composição introduzido na arquitetura SBI e sua instanciação para a consecução do exemplo da Figura 19 são apresentadas de forma simplificada nesta Seção. As listagens completas dos construtores semânticos e do exemplo estão disponíveis respectivamente no Apêndice B e no Apêndice C deste documento. Para fins didáticos, a seguir é apresentada apenas uma parte dos construtores e da sua instanciação para a consecução do exemplo.

Listagem 6 - Definição dos construtores semânticos para a definição do início de uma composição

```

(def-class data-flow-orchestration (orchestration)
  ((has-components :type composition-component)))

(def-instance checking-university-indicators data-flow-orchestration
  ((has-components
    (class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ControlComponent)
  )))
  
```

Um *Service Component* é na verdade um invólucro que mantém a informação necessária sobre os *Mediators* de ligação para um serviço executado diretamente (*i.e.*, quando o *Service Component* tratar-se de um WS) ou após um processo de descoberta (*i.e.*, quando o *Service Component* tratar-se de um *Goal*, que, ao ser executado, demandará a localização de um WS que possa executar a tarefa definida pelo *Goal*). O exemplo na Listagem 7 demonstra como um *Service Component* é definido. Um *Service Component* envolve exatamente a descrição de um *Goal* ou WS. Este *Goal* (ou WS) é invocado pelo mecanismo de orquestração durante a execução de uma composição. *Goals* podem estar localizados no mesmo servidor IRS-III onde a composição está armazenada ou em outro servidor IRS-III. Os serviços que serão chamados para atender às necessidades dos *Goals* podem estar localizados em qualquer servidor Web (DOMINGUES et al., 2004a; MOTTA et al., 2003).

No exemplo apresentado na Listagem 7, é apresentada a definição do primeiro *Service Component* da composição (*i.e.*, *service-component-find-critical-institution-indicators-goal*), incluindo a identificação do *Goal* que será invocado quando da sua execução na orquestração (*i.e.*, *find-critical-institution-indicators-goal-invocable*). São identificados também os *Mediators* que efetuarão suas ligações com os demais componentes da composição (*i.e.*, através da propriedade *has-internal-bindings*), sua ligação com as entradas fornecidas para a composição (*i.e.*, propriedade *has-binding-to-composition-input*) e a identificação de uma ligação condicional de sua saída através da propriedade *has-conditional-bindings*.

Listagem 7 - Definição do construtor para *service component* e de uma instância correspondente ao primeiro serviço da composição exemplo

```
(def-class service-component (composition-component)
  ((has-invocable-description :type invocable-description :cardinality 1)
   (has-internal-bindings :type output-binding)
   (has-binding-to-composition-input :type GInvGInvMediator)
   (has-binding-to-composition-output :type GInvGInvMediator)
   (has-conditional-bindings :type if-then-else)
   (has-failure-handling-service :type invocable-description :max-cardinality 1)))

(def-class invocable-description ()
  ((has-invocable-name :type string :cardinality 1)
   (has-ontology-name :type string :cardinality 1)
   (has-irs-server-url :type string :cardinality 1)))

(def-instance service-component-find-critical-institution-indicators-goal
  service-component
  ((has-invocable-description find-critical-institution-indicators-goal-invocable)
   (has-internal-bindings (
     mapping-mediation-goal5-service-component-list-staff-level-formation-goal
     mapping-mediation-goal6-service-component-list-depto-critical-indicator-goal
     GInvGInvMediator ))
   (has-binding-to-composition-input mapping-mediation-goal7)
   (has-conditional-bindings mapping-mediation-goal8-ifThenElse-1)
  ))

(def-instance find-critical-institution-indicators-goal-invocable
  invocable-description
  ((has-invocable-name find-critical-institution-indicators-goal)
   (has-ontology-name ontodss-goal)
   (has-irs-server-url "http://localhost:3000/soap")))
```

Todas as ligações de dados são definidas através de *Mediators*. No *Service Component* da Listagem 7, podem ser observadas três ligações de saída descritas através da propriedade *has-internal-bindings*, duas para outros *Service Components* e uma para um *Control Component*. Além dessas, descreve-se a ligação do valor recebido como entrada para a composição (*i.e.*, uma instância da classe *university*) com o *Service Component* através da propriedade *has-binding-to-composition-input*. De forma geral, quatro conjuntos de *Mediators* podem ser ligados para cada *Service Component*:

- entre a saída de um *Service Component* e a entrada de outros *Service Components*, através da propriedade *has-internal-bindings*;
- entre um *Service Component* e a saída da composição, através da propriedade *has-binding-to-composition-output*;
- entre valores de entrada para a composição e o *Service Component* que irá receber estes valores, através da propriedade *has-binding-to-composition-input*; e

- entre um *Service Component* e a entrada de outros *Control Components*, através da propriedade *has-conditional-bindings*.

O uso de *Mediators* é uma importante característica no modelo de composição utilizado pela Arquitetura SBI. O framework IRS-III suporta diversos tipos de *Mediators* previstos na ontologia WSMO (DOMINGUE et al., 2004a). Entretanto, o único tipo de *Mediator* que é requerido no modelo de composição desenvolvido é o *Goal Invocation Mediator* (i.e., *Ginv Mediator*), um *Mediator* introduzido especificamente para suportar o modelo de composição descrito nesta tese. Os demais *Mediators* prescritos no framework IRS-III são empregados de forma indireta, cabendo ao *IRS-III Server* utilizá-los quando necessário. O *Goal Invocation Mediator* (*Ginv*) contém seis propriedades, três das quais herdadas da superclasse *Mediator* da ontologia WSMO e três adicionais, as quais são apresentadas a seguir.

- *Has-source*: especifica o *Service Component* cujas saídas deverão ser tratadas pelo *Mediator*.
- *Has-target*: especifica o *Service Component*, que será alimentado após a execução do *Mediator*.
- *has-mediation-service*: representa um *Goal* ou um WS que executará a mediação (i.e., a transformação dos dados recebidos para o formato de saída, ou, simplesmente, o encaminhamento do dado recebido como entrada pelo *Mediator* para o *Service Component* destino). Todos os *Mediators* são também WS que são invocados através do *IRS-III Server*.

Uma das ligações de saída do *Service Component* apresentado na Listagem 7, e a descrição do respectivo *Ginv Mediator* é apresentada na Listagem 8. A ligação de saída especifica que o *Mediator mapping-mediation-goal8* deve ser aplicado para a saída do *Service Component* apresentado na Listagem 7 e para produzir a entrada para o *Control Component ifThenElse-1*. O *Control Component* referido receberá como saída do primeiro serviço o pior indicador em termos de desempenho da instituição entrada como parâmetro, conforme apresentado na Listagem 9. As ligações entre todos os componentes de uma composição funcionam de forma semelhante. Mais detalhes sobre tipos

de *Mediators*, sua execução e sua classificação podem ser obtidos na especificação WSMO (WSMO, 2004).

Listagem 8 - Definição da ligação de saída do primeiro *Service Component* e respectivo *Mediator* para o *Control Component* da composição

```
(def-class output-binding ()
  ((has-output-mediator :type GInvGInvMediator :cardinality 1))
  (to-component-service :type component-service :cardinality 1)
))

(def-class GInvGInvMediator ()
  ((has-source :type string)
   (has-target :type string :cardinality 1)
   (has-mediation-service :type invocable-description :cardinality 1)
   ;; mediation output specifies which output of the mediation service should be
   ;; taken and feed which input of which invocable.
   (has-mediation-output-role-name :type string)
   ;; mediation input specifies which output from which source invocable to be
   ;; taken by the orchestration engine and feed the input of the mediation
   ;; service.
   (has-mediation-input-role-name :type string)
   ;; parameters are used to specialize a generic mediator
   (has-mediation-parameters :type string)))

(def-instance mapping-mediation-goal8-ifThenElse-1 output-binding
  ((has-output-mediator mapping-mediation-goal8)
   (to-component-service ifThenElse-1)))

(def-instance mapping-mediation-goal8 GInvGInvMediator
  ((has-source find-critical-institution-indicators-goal)
   (has-target ifThenElse-1)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-output-crit-ind-institution)
   (has-mediation-output-role-name ?value)
   (has-mediation-parameters (has-output-crit-ind-institution ?value null))))
```

A especificação do *Control Component if-Then-Else1* compreende uma condição a ser verificada e a indicação de até dois componentes, como ilustrado na Figura 19. O primeiro corresponde a um *Service Component* que será invocado caso a condição seja verdadeira (*i.e.*, o componente *Then*), e o segundo, para o caso da condição não ser verdadeira (*i.e.*, o componente *Else*). O componente *Then* é obrigatório, assim como a condição. Tanto o componente *Then* quanto o *Else* deverão prever *Mediators* para a ligação dos seus valores de entrada para a saída do operador *If-Then-Else*.

Finalmente, o *Control Component Concurrent* do modelo permite a execução paralela de *Service Components*. A execução é feita pelo Gerenciador de Serviços através de um conjunto de classes Java a serem apresentadas na próxima Seção.

Listagem 9 – Construtor semântico utilizado para a definição de Control Components e sua instância no contexto da composição ilustrada na Figura 19

```
(def-class If-then-else (control-component)
  ((has-condition :type string :cardinality 1)
   (has-then-binding :type output-binding :cardinality 1)
   (has-else-binding :type output-binding :max-cardinality 1)
  ))

(def-instance ifThenElse-1 if-then-else
  ((has-condition "ifThenElse-1-func")
   (has-then-binding
    mapping-mediation-goall2-service-component-list-staff-level-formation-goal)
   (has-else-binding
    mapping-mediation-goall3-service-component-list-depto-critical-indicator-goal)
  ))

(def-function ifThenElse-1-func (?value )
  :body (the true (= ?value 'degree)))
```

O modelo de composição apresentado não prevê construtores para a definição da seqüência que cada serviço será executado. A ordem de execução depende dos dados providos aos *Service Components* durante a orquestração da composição, e não no tempo de definição da composição. Assim, um serviço é executado quando os dados necessários para preencher todos os parâmetros de entrada estão disponíveis. Esse modelo de composição torna a definição de composições mais simples para os usuários, dado que eles não precisam indicar explicitamente a ordem que serão executados os serviços. Um modelo semelhante é adotado também pela linguagem BPEL4WS (IBM CORPORATION, 2004). Além da simplicidade para definição de composições introduzida pelo modelo, verificava-se na literatura, na época da implementação deste modelo, uma virtual inexistência de mecanismos para orquestração de *Web services* semânticos. Este modelo serviu de referência para algumas publicações, entre elas Sell et al. (2004b) e Hakimpour et al. (2005).

5.5 Considerações sobre o Capítulo

Neste capítulo foi apresentada uma visão geral sobre as ontologias que formam a segunda camada da arquitetura SBI. A camada de ontologias foi constituída para possibilitar: a navegação sobre as fontes de dados a partir dos conceitos do negócio (RF1); a utilização das regras de negócio para apoiar o processamento analítico (RF2); a modificação dos conceitos e regras do negócio (RF3); a extensão das funcionalidades exploratórias a partir de aplicações existentes (RF4); a composição de serviços para a extensão de

funcionalidades exploratórias (RF5); a recomendação proativa de recursos aos usuários (RF6); a aplicação da arquitetura em qualquer domínio de negócio (RNF1); a representação semântica do negócio, das fontes de dados e dos serviços através de diferentes formalismos (RNF2); e a integração dos módulos componentes da solução de BI a soluções de terceiros (RNF3).

Uma visão geral das principais características das ontologias e dos requisitos associados a cada ontologia é apresentada no Quadro 16.

Quadro 16 - Visão geral das características das ontologias da Arquitetura vis-à-vis requisitos funcionais e não funcionais

Características Apoiadas pelas Ontologias	Requisitos Funcionais						Requisitos Não Funcionais		
	1	2	3	4	5	6	1	2	3
Ontologia do Domínio									
Descrição dos conceitos e regras do negócio	X	X	X	X	X	X	X	X	X
Ontologia BI									
Mapeamento semântico das fontes de dados	X								X
Personalização da apresentação dos resultados das análises	X								
Redefinição de consultas a partir de inferências sobre os conceitos e regras do negócio		X							
Descrição semântica de análises						X			
Navegação sobre os dados através de temas de análise	X								
Ontologia de Serviços									
Mapeamento semântico de serviços				X		X	X		X
Suporte para composição de serviços					X	X	X		X

No próximo capítulo são apresentados os módulos funcionais que compõem a terceira camada da arquitetura SBI e a sua relação com as ontologias apresentadas no presente capítulo.

6 OS MÓDULOS FUNCIONAIS DA ARQUITETURA SBI

6.1 Introdução

A arquitetura SBI é organizada em camadas que compreendem ontologias, fontes de informação e um conjunto de módulos utilizados para prover às ferramentas analíticas o acesso aos dados e serviços de interesse aos tomadores de decisão e o suporte a inferências a partir das ontologias e dos mecanismos de inferência acoplados à arquitetura.

Nesta Seção são apresentados os módulos funcionais que compõem a terceira camada da arquitetura SBI. Os módulos funcionais compreendem vários métodos utilizados para integrar os recursos das tecnologias semânticas às aplicações analíticas, visando garantir o cumprimento dos requisitos definidos para a arquitetura SBI. Apresenta-se como cada módulo se relaciona com os demais componentes da arquitetura para atender aos requisitos funcionais e não funcionais apresentados no Capítulo 4.

6.2 Gerenciador de Ontologias

Para o cumprimento dos requisitos funcionais da arquitetura no que tange à navegação sobre as fontes de dados, à utilização de regras de negócio e à recomendação de recursos durante as análises, torna-se necessário definir um conjunto de métodos que possibilitem:

1. a manipulação dos conceitos definidos nas ontologias da arquitetura, através de métodos que permitam a criação, a alteração e a recuperação de instâncias das ontologias; e
2. a recuperação e a execução de regras definidas na Ontologia do Domínio, visando suportar o processamento das inferências preconizadas pelos requisitos da arquitetura.

O Gerenciador de Ontologias é o módulo da arquitetura SBI que provê o conjunto de métodos necessários para a execução de consultas e inferências sobre os repositórios de ontologias da arquitetura SBI. De forma geral, este módulo visa possibilitar que as ontologias da arquitetura sejam utilizadas no cumprimento de todos os requisitos funcionais definidos no Capítulo 4. Este módulo foi concebido para tornar transparente a manipulação das ontologias

aos demais módulos da arquitetura e oferecer liberdade para seleção do mecanismo de inferência e do formalismo de representação das ontologias, conforme previsto no requisito RNF2.

A partir da análise dos requisitos funcionais e não funcionais da arquitetura, foi definida uma interface (FOLDOC, 2005a) que reúne um conjunto de métodos para o acesso aos repositórios de ontologias. Essa interface pode ser implementada sobre diferentes mecanismos de inferência e formalismos de representação e armazenamento de ontologias. A Figura 20 ilustra as classes que constituem o Gerenciador de Ontologias. Uma ilustração das classes que implementam a versão atual do Gerenciador de Ontologias é apresentada na Figura 20. No momento, o módulo conta com uma implementação para a linguagem OCML, através da classe *OntologyManagerOCML*. Essa classe implementa os métodos definidos na interface *OntologyManager* apoiando-se no *plugin* Java para a linguagem OCML (KMi, 2005).

O Gerenciador de Ontologias na arquitetura SBI permite um baixo nível de acoplamento (FOLDOC, 2005b) entre aplicações analíticas, mecanismo de consulta e repositório de ontologias, resultando em maior flexibilidade para mudanças nos repositórios e formalismos de representação. Essa característica é possibilitada porque os demais módulos da arquitetura SBI acessam as ontologias através dos métodos padrão definidos na interface *OntologyManager*. Dessa forma, a Arquitetura SBI torna possível a adoção de diferentes formalismos de representação de ontologias, desde que uma classe específica para o formalismo desejado seja desenvolvida de acordo com a lista de métodos da interface padrão.

Conforme ilustra a Figura 20, a interface *OntologyManager* prevê métodos para:

1. criação de instâncias (*i.e.*, método *createInstance*). Através deste método, o Gerenciador de Análises persiste na Ontologia BI as definições das análises criadas pelos usuários. Da mesma forma, este método é utilizado pelo Gerenciador de Instâncias para criar os conceitos na ontologia do Domínio a partir de dados selecionados dos repositórios de dados da organização. Para a

linguagem OCML, a execução deste método corresponde à chamada do método homônimo no *plugin* OCML;

2. recuperação de informações gerais sobre as instâncias e classes definidas nas ontologias da arquitetura (*i.e.*, através dos métodos *get* e *retrieve*, como o método *getInstanceData* e *getSlots*). Esses métodos utilizam duas classes de suporte para auxiliar na manipulação das informações das instâncias e respectivos slots, as classes *Instance* e *Slot*. Através desses métodos torna-se possível recuperar as definições das ontologias da arquitetura, como, por exemplo, para apresentar as instâncias de instituições recuperadas pela relação *institution-competitor* apresentada na Listagem 1;
3. recuperação de filtros semânticos e relações para um dado conceito (*i.e.*, *getRelation* e *getSemanticFilters*). Esses métodos são utilizados para recuperar alternativas de inferências para os conceitos relacionados em uma análise. Tais alternativas consistem de relações e regras definidas na Ontologia do Domínio que envolvam um determinado conceito nas premissas das regras ou como parâmetro de uma relação. Na linguagem OCML, a execução desses métodos demanda a execução de uma consulta na Ontologia do Domínio, buscando relações e regras que possuam em seus cabeçalhos, na Seção *constraint* (MOTTA, 1999), referências ao conceito a ser localizado; e
4. extração dos rótulos para as tabelas e respectivos campos de acordo com o perfil do usuário (*i.e.*, *retrieveLabel* e *retrieveTableLabel*). Esses dois últimos métodos correspondem à execução da relação *label-role-idiom* definida na Ontologia BI e apresentada na Listagem 4.

Além dos métodos descritos acima, um método específico para comparação entre conceitos é previsto na interface *OntologyManager*. Trata-se do método *compareConcepts*, que verifica se dois conceitos são compatíveis, conforme algoritmo ilustrado na Figura 21. Esse método foi criado a partir da definição de graus de semelhança entre descrições de *Web services*

semânticos apresentada em Li e Horrocks (2003), aqui adaptado para a comparação de qualquer tipo de conceito representado na Ontologia do Domínio. O método *compareConcepts* é utilizado no processo de localização e recomendação de análises e serviços, de acordo com os conceitos manipulados em uma análise definida pelo tomador de decisão em uma ferramenta analítica, conforme o requisito RF6. Nesse processo de localização, são considerados os conceitos utilizados na definição da análise corrente do usuário e os conceitos que descrevem os serviços, as análises, as relações e as regras nas ontologias da arquitetura. Esses métodos de localização e recomendação são apresentados nas seções correspondentes ao Gerenciador de Serviços (Seção 6.3) e Gerenciador de Análises (Seção 6.4).

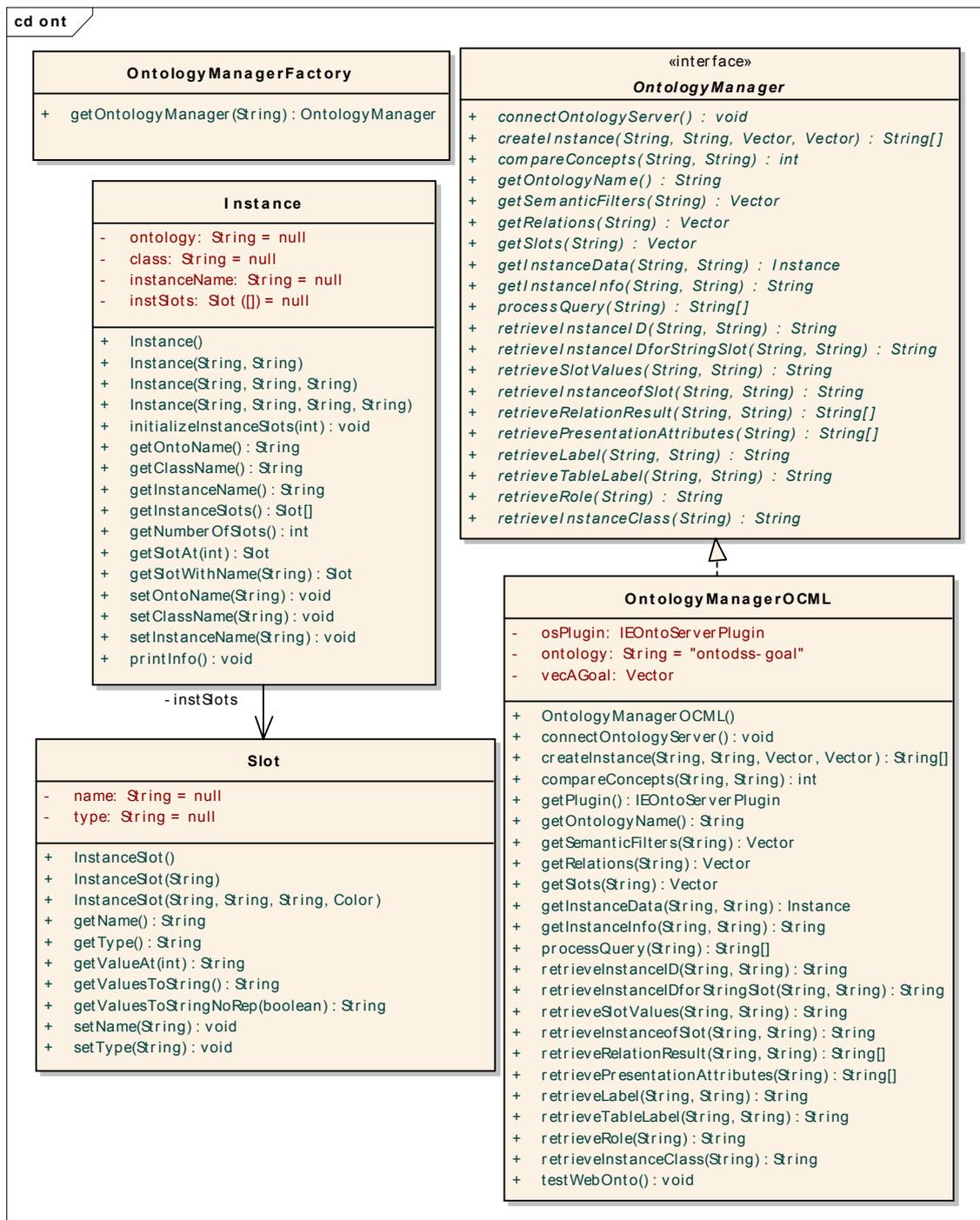


Figura 20 - Ilustração das classes que compõem o módulo Gerenciador de Ontologias

Conforme ilustrado na Figura 21, caso os dois conceitos informados como parâmetro para o método *compareConcepts* sejam idênticos, diz-se que existe um *match exato* (*exact*) entre eles. Se o conceito 2 for pai do conceito 1, diz-se que há uma relação do tipo *plugin* entre o conceito 1 e o conceito 2 e, caso contrário, diz-se que o conceito 1 inclui (*subsume*) o conceito 2. Caso um filho do conceito 1 seja idêntico a um filho do conceito 2, diz-se que há uma

relação de intersecção (*intersection*) entre os dois conceitos. Em caso de total incompatibilidade entre os dois conceitos, o método retorna que os dois conceitos são disjuntos (*disjoint*).

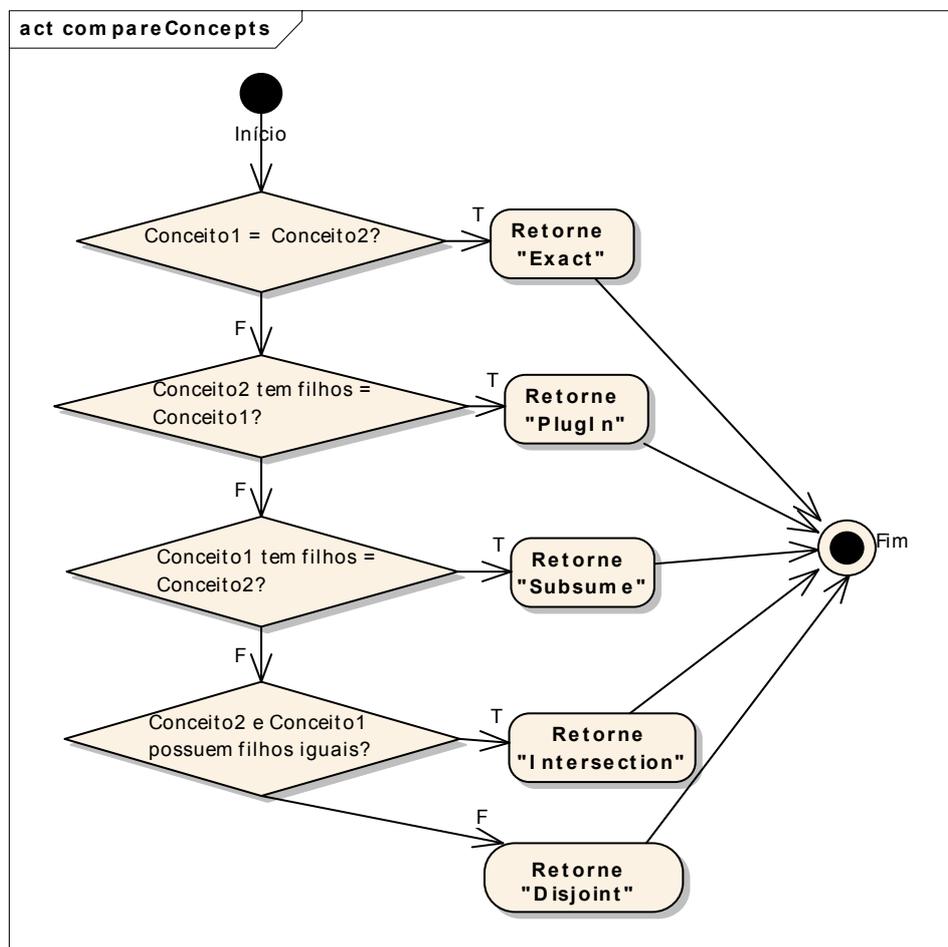


Figura 21 - Ilustração do processo de comparação de conceitos

6.3 Gerenciador de Serviços

Conforme apresentado no capítulo anterior, a Ontologia de Serviços visa descrever a semântica de serviços que possam ser utilizados para a extração de informação e para o processamento de transações. A Ontologia de Serviços é utilizada pela infra-estrutura WSS para a localização e execução de *Web services*. Para tornar esses recursos acessíveis às aplicações analíticas, torna-se necessário criar um módulo que implemente um conjunto de métodos visando suportar as seguintes operações:

1. a localização de serviços relacionados ao contexto de uma análise, visando atender o requisito RF6;

2. a execução de um dado serviço e a recuperação do resultado do processamento realizado pelo serviço, visando oferecer novos recursos analíticos, conforme o requisito RF4; e
3. a montagem e a execução de composições de serviços, conforme prescrito no requisito RF5.

O Gerenciador de Serviços permite o reuso de código existente na organização ou na Web para a agregação de novas funcionalidades analíticas ou para a substituição das funcionalidades existentes em uma solução de BI, conforme prescrito no requisito RF4. Dessa forma, a arquitetura SBI oferece mais uma solução para garantir escalabilidade às soluções de BI.

O suporte à recomendação automática de WSS é feito através do cruzamento entre os conceitos utilizados para descrever uma análise e os conceitos utilizados para descrever as capacidades dos WSS na Ontologia de Serviços, visando atender o requisito RF6. A descoberta é feita através do cruzamento entre os conceitos utilizados para definir uma análise e os conceitos utilizados para a descrição dos WSS. Nesse cruzamento são considerados sinônimos, hipônimos e hiperônimos.

O Gerenciador de Serviços suporta ainda a composição e a orquestração de WSS, conforme prescrito no requisito RF5. Uma composição pode ser alimentada por dados analisados por um usuário na ferramenta analítica. O resultado de uma composição pode ser apresentado ao usuário na ferramenta analítica ou encaminhado como parâmetro de entrada para outras análises.

Em linhas gerais, este módulo funciona como uma interface para uma infra-estrutura WSS e é implementado através de uma API Java integrada à Ontologia de Serviços no intuito de suportar a descoberta, a composição, a invocação e o monitoramento de WSS. Na sua versão atual, este módulo se integra ao framework IRS-III através da API Java IRS-III Client (IRS-III API, 2004) e da API Java para a orquestração de composições desenvolvidas pelo autor em cooperação com pesquisadores do Knowledge Media Institute (vide SELL et al., 2004b e HAKIMPOUR et al., 2005). A seguir, é apresentada uma visão geral das principais classes que formam o Gerenciador de Serviços.

6.3.1 Classes para localização e execução de WSS

O Gerenciador de Serviços possui uma interface que especifica uma série de métodos para a manipulação de *Web services* semânticos. Essa interface permite que possam existir implementações do Gerenciador de Serviços para diferentes framework de *Web services* semânticos, tornando a arquitetura SBI independente de uma solução específica. Até o momento, a arquitetura SBI possui uma implementação dessa interface voltada para o framework *IRS-III*. A interface, a classe que implementa a interface para o framework *IRS-III* e a classe *factory* (WIKIPEDIA, 2004b) utilizada para instanciar essa classe, são ilustradas na Figura 22. Esse conjunto de classes é mais uma contribuição desta tese para agregar os recursos introduzidos pelas tecnologias semânticas de última geração a soluções de BI.

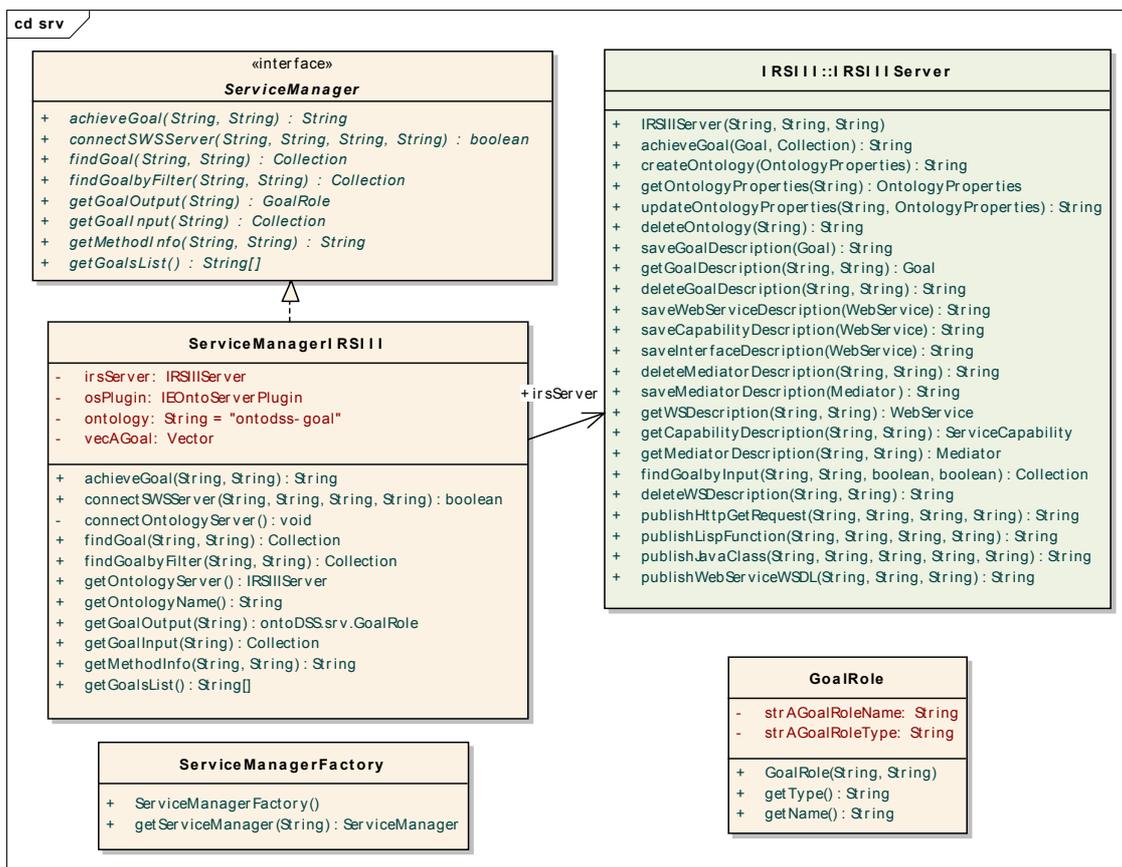


Figura 22 - Ilustração da interface principal do Gerenciador de Serviços, sua implementação para o framework *IRS-III* e classes de suporte

A interface *ServiceManager* do Gerenciador de Serviços prevê um método (*connectSWSServer()*) para conexão a um framework WSS e métodos para a localização de WSS pelo nome do serviço desejado (*i.e.*, através do

método *findGoal()*) ou por um filtro definido em tempo de execução. (i.e., através do método *findGoalbyFilter()*). Nesse último, o primeiro parâmetro corresponde ao nome da ontologia onde se deseja procurar o serviço, e o segundo parâmetro corresponde a um filtro formatado como ilustrado no Quadro 17.

Quadro 17 - Formatação de filtros para a busca de WSS na Ontologia de Serviços através do método *FindGoalbyFilter*

Sintaxe para a formatação de filtros:

```
(<logical operator> (property1 value1 ?service) ... (propertyn valuen ?service))
```

Onde:

- `logical operator` corresponde ao operador lógico OR ou AND;
- `property` corresponde à propriedade do WSS que deseja-se filtrar (i.e. `input-role`, `output-role`, `publisher`, `name` e outras propriedades da descrição WSMO (WSMO, 2004));
- `value` corresponde ao valor utilizado para filtrar a propriedade (i.e. um conceito, uma determinada instância, uma string, etc).

Por exemplo, para localizar um WSS que possui como entrada *university* e como saída *indicator*, seria constituído o seguinte filtro:

- `(and (input-role university) (output-role indicator))`

A recomendação de serviços para os tomadores de decisão é realizada através do método *findGoalbyFilter()*. Na ferramenta OLAP OntoDSS apresentada na Seção 7.2, a recomendação de serviços é feita da seguinte forma:

1. o usuário clica com o botão direito sobre qualquer elemento de um cubo de dados e seleciona a opção “Related Services”;
2. o conceito referente ao elemento clicado é adicionado como filtro para os parâmetros de entrada e saída, sendo utilizado o conector *or*, como ilustrado no Quadro 17;
3. O filtro é processado através do mecanismo de inferência sobre as descrições dos serviços mantidas na Ontologia de Serviços. Dessa forma, serão recuperados os *Goals* e *Web services* que possuírem em seus *Input* e *Output Roles* conceitos compatíveis com o identificado pelo usuário. A análise de compatibilidade entre os conceitos é efetuada através do método *compareConcept*, descrito na Seção 6.2;

4. os serviços listados pelo método *findGoalbyFilter()* são apresentados ao usuário;
5. o serviço selecionado pelo usuário é acionado através do método *achieveGoal()*, e o parâmetro de entrada correspondente ao elemento clicado no cubo de dados é preenchido com o valor clicado. Os demais parâmetros de entrada são então solicitados ao usuário; e
6. o resultado do serviço é apresentado ao usuário.

A interface *ServiceManager* descreve ainda o método *achieveGoal()* para invocar um *Goal* ou *Web service*. Além desses, a interface *ServiceManager* prevê métodos para recuperar a descrição de um ou vários *Goals* ou *Web services*, através dos seguintes métodos:

- *getGoalInput()*: retorna uma coleção de instâncias da classe *GoalRole*, uma para cada parâmetro de entrada do *Goal* desejado. A classe *GoalRole* descreve o nome de cada parâmetro e seu tipo;
- *getGoalOutput()*: retorna um *GoalRole* para a saída do *Goal* desejado;
- *getGoalList()*: retorna todos os *Goals* definidos na Ontologia de Serviços; e
- *getMethodInfo()*: retorna uma descrição textual do *Web service* desejado.

A implementação da interface *ServiceManager* para o framework de WSS IRS-III é feita pela classe *ServiceManagerIRSIII*, que, por sua vez, trata todos os métodos previstos na interface apoiando-se na API Java *IRSIIIServer*. A API *IRSIIIServer* prevê uma série de métodos para a manutenção de descrições de WSS além de outros para localização, execução e monitoramento de WSS. O autor desta tese atuou no desenvolvimento de métodos nessa API para a descoberta de WSS. Mais informações sobre a API do IRS-III e os métodos para descoberta e execução de WSS podem ser obtidas em IRS-III API (2004).

6.3.2 Classes para montagem e execução de composições

Para o atendimento do requisito RF5 (suporte a composições de serviços), o Gerenciador de Serviços prevê ainda um conjunto de classes para a criação e orquestração de composições de WSS baseadas em fluxos de dados. Essas classes permitem que tomadores de decisão possam definir composições de serviços para realizar processamentos diversos, tomando como ponto de partida os dados reunidos em suas análises. Até o momento, esse conjunto de classes suporta, através da Arquitetura SBI, a ferramenta para definição de composições, descrita em Sell et al. (2004b), e a ferramenta OntoDSS, descrita na Seção 7.2.

Uma composição é definida através de um objeto da classe *DataFlowComposition*, uma extensão da classe *Composition*, conforme ilustrado na Figura 23. A classe *DataFlowComposition* reúne várias informações sobre a composição criada pelo usuário, incluindo: seu nome (*i.e.*, atributo *name*); a ontologia (*i.e.*, atributo *ontology*), onde serão depositados os construtores semânticos correspondentes, conforme descrito na Seção 5.4 (Ontologia de Serviços); os repositórios de dados de entrada e de saída dos serviços reunidos na composição (*i.e.*, *compositionInput* e *compositionOutput* respectivamente); informação sobre quais serviços estão sendo executados e qual o status de cada serviço (*i.e.*, *runningThreadNames* e *threadStatus* respectivamente); além de outras informações sobre o estado da composição e sobre a coleção de serviços adicionados à composição (*i.e.*, *orchestrationRunning*, *orchestrationWaiting* e *componentServices*). Essas informações são manipuladas por um conjunto de métodos utilizados para a adição de serviços à composição (*i.e.*, *addComponent*) ou para orquestração e encerramento da execução da composição (*i.e.*, métodos *perform* e *terminate* respectivamente).

Uma composição definida segundo este modelo reúne uma coleção de serviços que aguardam valores para suas entradas e valores produzidos como saída dos seus processamentos. Esses valores são reunidos em uma classe chamada *Repository*, a qual funciona como uma lista de objetos da classe *RoleValuePair*. A classe *RoleValuePair* identifica o nome de um parâmetro de entrada ou de saída e seu respectivo valor.

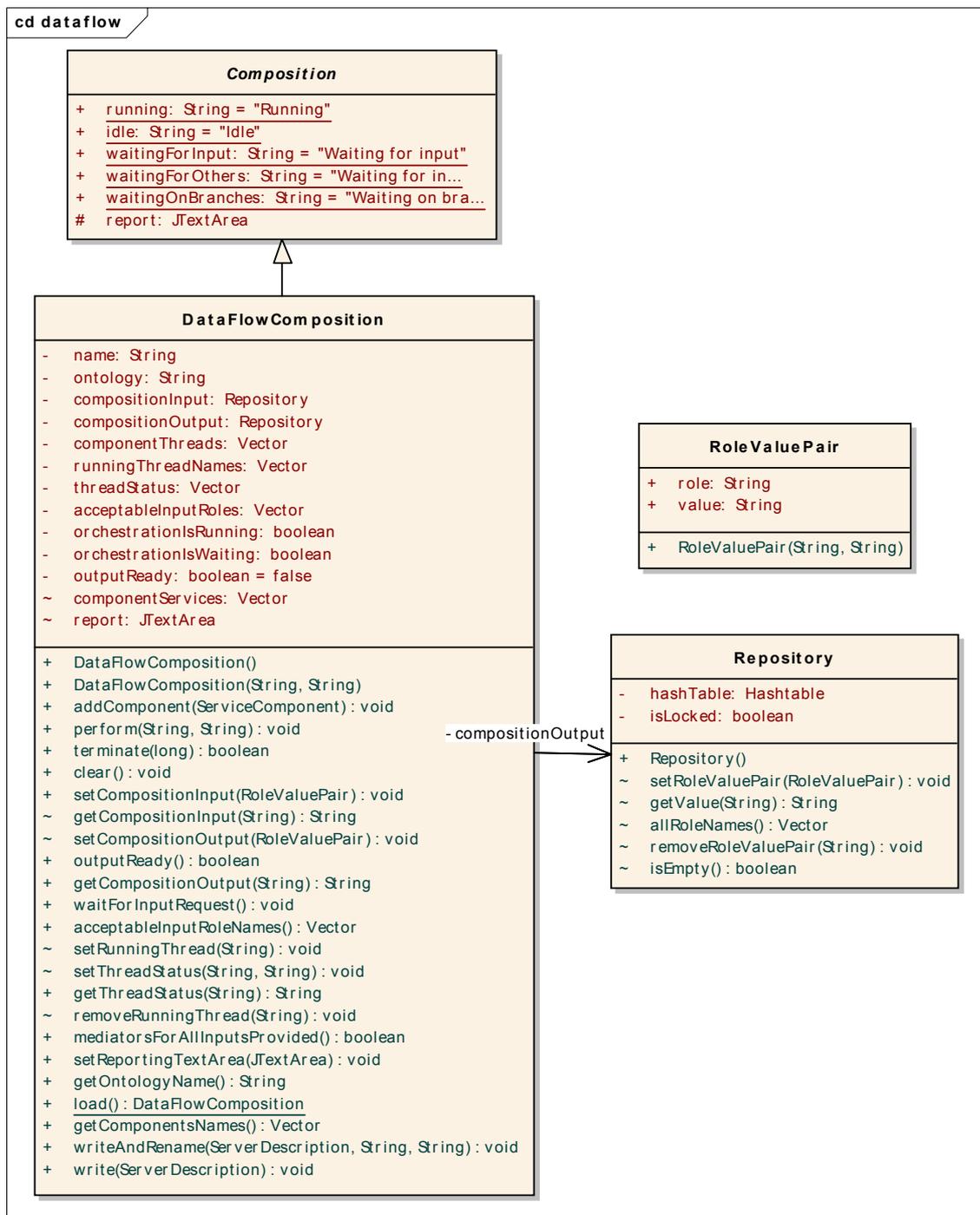


Figura 23 - Ilustração das classes utilizadas para a definição de uma composição

Como descrito na Seção 5.4, o modelo de composição desta tese prevê componentes de controle e componentes de serviço. Os componentes de controle prescrevem como será encaminhado o fluxo de dados na composição, e os componentes de serviço descrevem os *Goals* e *Web services* relacionados na composição. Ambos os componentes são filhos da classe *CompositionComponent* e, portanto, herdam a propriedade *belongsTo*, que

descreve à qual composição um componente está atrelado, além do método *perform* que executa o componente.

A classe *ServiceComponent* mantém em suas propriedades informações sobre um *Goal* ou *Web service* definido na Ontologia de Serviços, como a sua identificação (*i.e.*, através da propriedade *invocable*), além das ligações definidas na composição entre o componente corrente e os outros componentes ou das suas ligações com as entradas e saída da composição. Essas ligações são definidas por meio dos métodos *setInternalBinding*, *setConditionalBinding*, *setBindingToCompositionInput* e *setBindingToCompositionOutput*. A classe *IfThenElse* descreve uma condição (*i.e.*, uma expressão condicional definida pela propriedade *condition*) e suas ligações condicionais definidas através dos métodos *addThenBindingMediator* e *addElseBindingMediator*.

As descrições de ligações da classe *ServiceComponent* e da classe *IfThenElse* são feitas através de objetos da classe *GInVMediator*. A classe *GInVMediator* descreve quais as saídas (*mediationOutputRoleNames*) do componente origem (*i.e.*, o nome de um *ServiceComponent* ou *ControlComponent* identificado pela propriedade *source*) serão fornecidas para as entradas identificadas pela propriedade *mediationOutputRoleNames* do componente destino (identificado pela propriedade *target*). Esta classe descreve ainda, opcionalmente, um *Goal* ou *Web service* que realizará uma transformação entre a saída do primeiro componente e a entrada do segundo.

Todos os objetos da classe *ServiceComponent* são transformados em *threads* (WIKIPEDIA, 2004c) através de objetos da classe *ComponentThread*. Tais objetos prevêm um controle para verificar a cada segundo se o seu repositório de dados de entrada (*i.e.*, objeto da classe *Repository* identificado através de sua propriedade *repository*) já possui todos os dados necessários para preencher seus parâmetros de entrada. Quando um *ServiceComponent* possui todos os dados necessários, ele é executado acionando-se o *Goal* ou o *Web service* descrito pela propriedade *invocable*. O resultado do *ServiceComponent* é então processado pelos *Mediators* definidos nas propriedades *internalBindings*, *conditionalBindings* e *bindingToCompositionOutput* do *ServiceComponent*. Os *Mediators* acionados

preencherão os repositórios dos componentes associados ao componente que acabou de ser executado, e esse processo continua até que todos os componentes da composição tenham sido executados, produzindo a saída da composição. Esse processo é ilustrado na Figura 25.

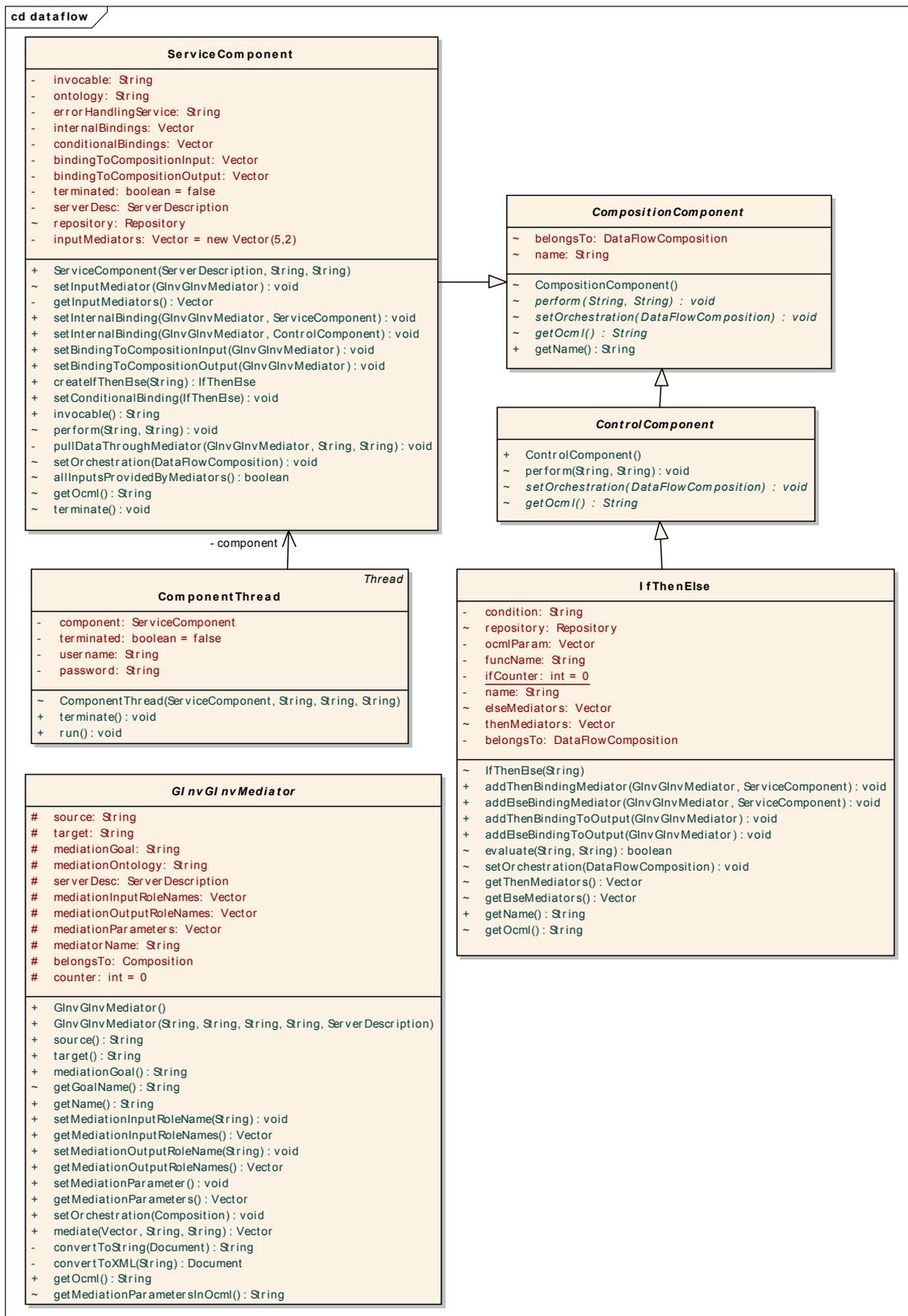


Figura 24 - Ilustração das classes utilizadas para a definição de *Control Components*, *Service Components* e *Mediators* em uma composição

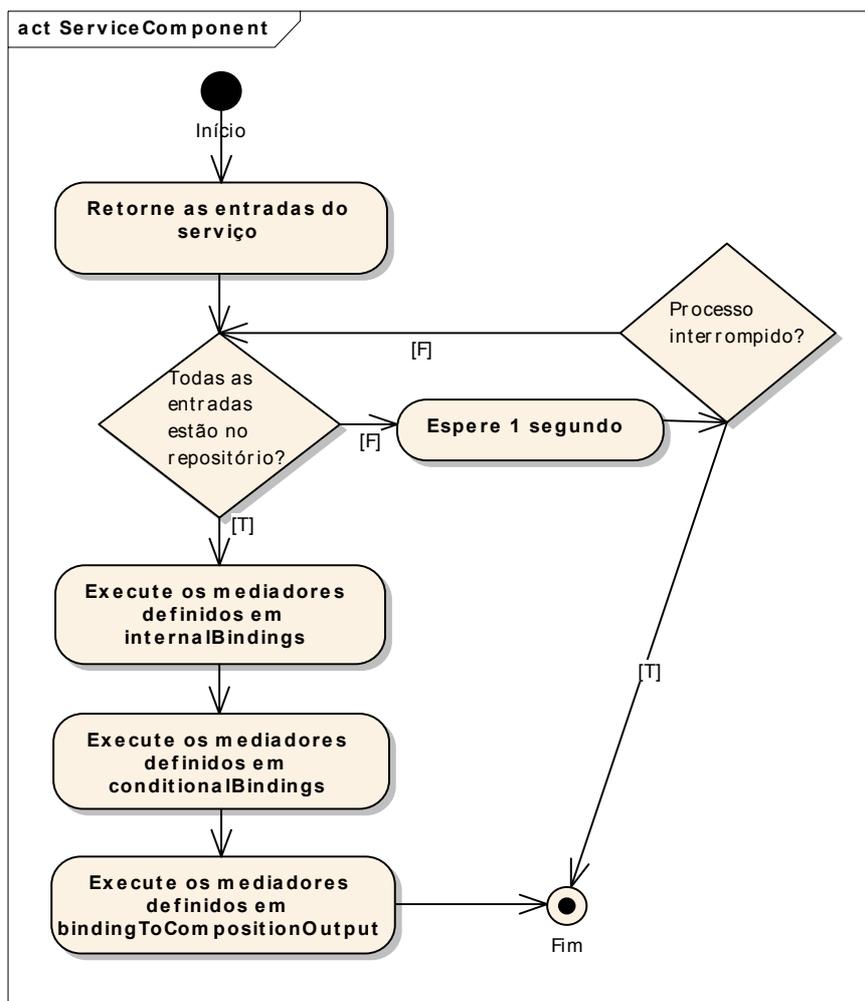


Figura 25 - Fluxo de processamento em um *Service Component*

6.4 Gerenciador de Análises

Para o atendimento dos requisitos funcionais relativos à aplicação da semântica do negócio para a extensão das funcionalidades analíticas e no suporte a um acesso transparente aos dados reunidos nos repositórios de interesse do tomador de decisão, torna-se necessário implementar na arquitetura SBI métodos que possibilitem:

1. a definição de análises a partir da associação dos conceitos do negócio a unidades de análise, de conteúdo e de filtro;
2. o processamento das definições semânticas das análises diretamente sobre bancos de dados relacionais;
3. a aplicação de regras e relações definidas sobre a Ontologia do Domínio para filtrar ou expandir os resultados de uma análise; e

4. a recomendação de recursos (como opções de navegação e de serviços) ao tomador de decisão, de acordo com o contexto de suas análises;

O Gerenciador de Análises foi concebido para implementar os quatro métodos acima identificados e trata-se também do módulo pelo qual as ferramentas de suporte à decisão terão acesso a todos os componentes da arquitetura. Este módulo intermedeia o acesso ao Gerenciador de Ontologia e ao Gerenciador de Serviços. Provê ainda um conjunto de métodos para o acesso às fontes de dados, às unidades de análise e às análises previamente definidas na Ontologia BI, visando apoiar o cumprimento dos requisitos RF1 (navegação sobre dados usando conceitos do negócio), RF2 (utilização de regras no processamento analítico) e RF6 (recomendação de recursos). As classes que compõem o módulo Gerenciador de Análises são ilustradas na Figura 26.

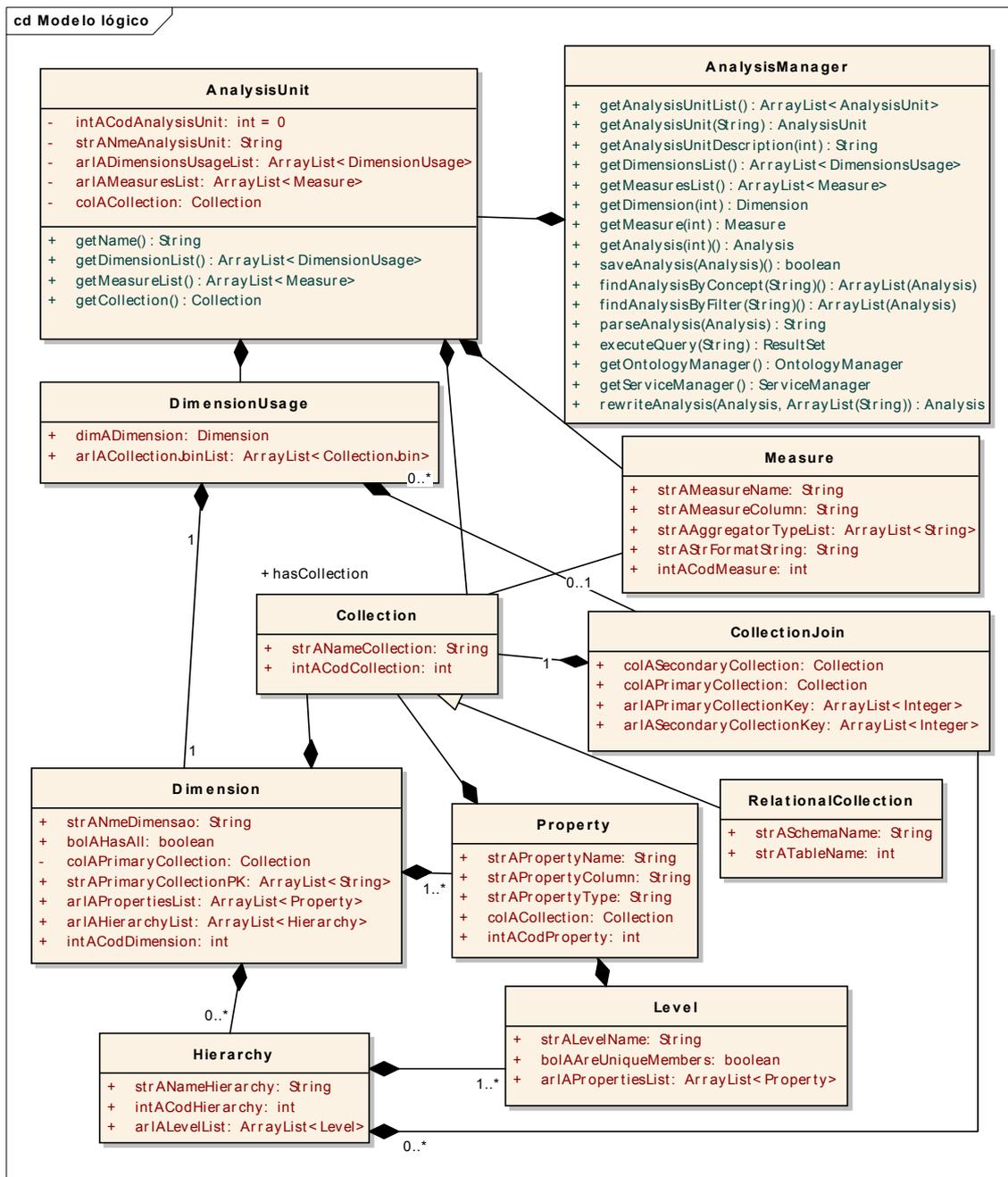


Figura 26 - Ilustração das principais classes que compõem o Módulo Gerenciador de Análises

6.4.1 Suporte para a definição de análises

O módulo Gerenciador de Análises é implementado pela classe *AnalysisManager* e pelas classes de suporte ilustradas na Figura 26. A classe *AnalysisManager* implementa uma coleção de métodos *get* que são utilizados por aplicações analíticas para recuperar os temas de análise e as análises definidas previamente. Essas definições são utilizadas pelas ferramentas para apresentar aos usuários que Unidades de Análise, Unidades de Filtro e de

Conteúdo estão disponíveis, além de todas as análises feitas anteriormente e que estão disponíveis para o grupo de usuários ao qual o usuário integra.

As definições de unidades de análise, de filtro e de medidas permitem a concepção de ferramentas de apoio a decisões genéricas, guiadas pelas definições mantidas na Ontologia BI. Assim, essas ferramentas podem ser utilizadas em qualquer domínio de negócio, facilitando a sua adaptação e aumentando a reutilização por parte do grupo de desenvolvimento. No Capítulo 7 é apresentada uma ferramenta concebida dessa forma. Outras ferramentas de apoio a decisões genéricas foram desenvolvidas pelo Instituto Stela utilizando essas definições.

O processo de recuperação das definições semânticas das fontes de dados é suportado por um conjunto de métodos implementado pelo Gerenciador de Análises. Inicialmente são recuperadas as unidades de análise definidas na Ontologia BI através do método *getAnalysisUnit()* da classe *AnalysisManager*. Na seqüência, para cada Unidade de Análise (*i.e.*, instâncias da classe *AnalysisUnit* recuperadas), são extraídas suas medidas (*i.e.*, instâncias da classe *Measure* através do método *getMeasureList()*) e suas definições de filtros e agrupamentos através do método *getDimensionUsage()*. A classe *DimensionUsage* confere o acesso às dimensões (e, por conseqüência, às suas propriedades e hierarquias) e à descrição das tabelas onde as dimensões são mantidas e sobre os joins necessários para recuperar os dados das dimensões (*i.e.*, informações da classe *CollectionJoin*).

Análises definidas pelos usuários podem ser persistidas na Ontologia BI, gerando instâncias do conceito *Analysis*, conforme descrito na Seção 5.3.4. Na classe *AnalysisManager*, um método foi reservado para a persistência de análises: o método *saveAnalysis()*. Este método recebe como parâmetro uma instância da classe Java *Analysis* ilustrada na Figura 27. Esta classe Java implementa todos os atributos definidos no conceito homônimo e é utilizada como classe de apoio pelas ferramentas analíticas ligadas à arquitetura SBI para suportar o processo de definição de consultas, cubos e relatórios. A classe *Filter*, também ilustrada na Figura 27, é utilizada para a definição de restrições sobre uma análise.

As análises podem ser definidas através da seleção de conceitos estabelecidos na Ontologia do Domínio. O usuário pode definir, por exemplo, que deseja adicionar a instância *Brazil* do conceito *Country* na lista de filtros. Conforme mostrado na Seção 5.3.1, na Ontologia BI, as tabelas são descritas semanticamente através de conceitos como *DB_Collection* e *DB_Attribute*. A tabela e o campo correspondente ao conceito *Country* são recuperados por meio de uma consulta sobre a Ontologia BI. Da mesma forma, todos os conceitos selecionados para a identificação das unidades de análise, de conteúdo e de filtro são traduzidos pelas informações sobre as tabelas e os campos.

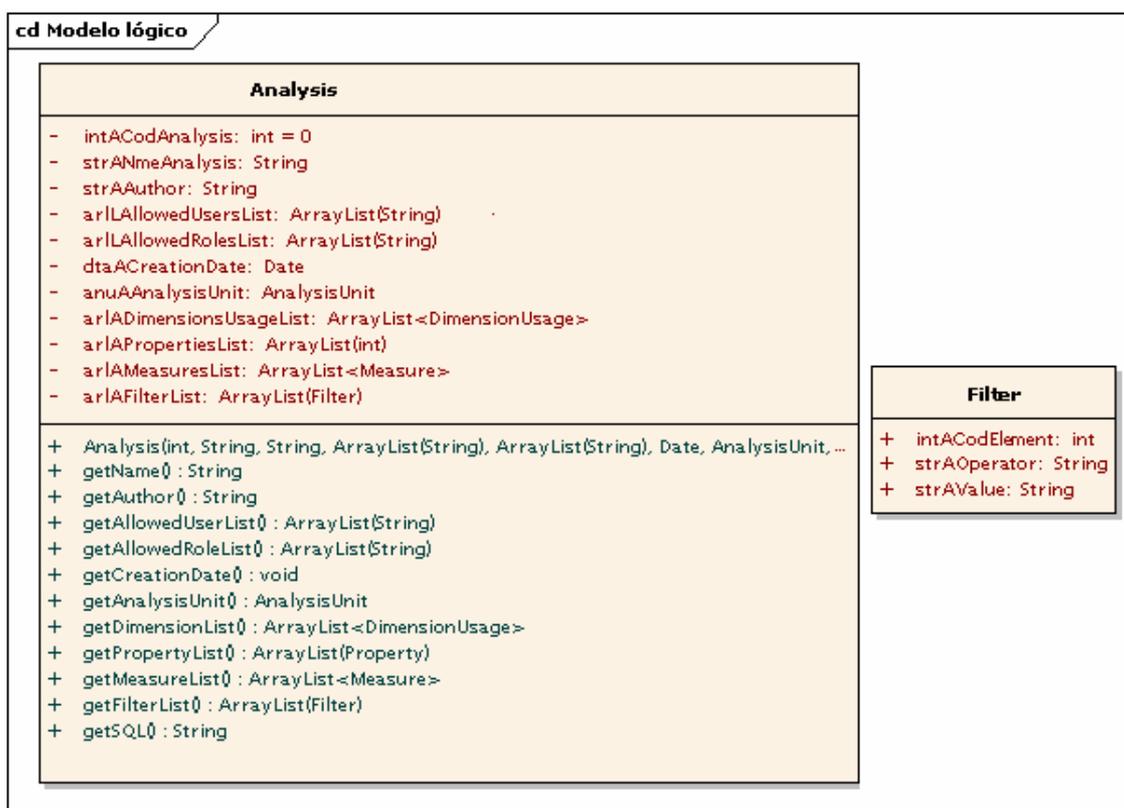


Figura 27 - Classes de apoio para a definição de análises e filtros

O método *getSQL* da classe *Analysis* traduz as definições da análise em um SQL Ansi, observando a seqüência de passos ilustrada na Figura 28. Na montagem do SQL Ansi, no primeiro passo, recupera-se a tabela de fatos da unidade de análise através do método *getCollection* da classe *AnalysisUnit*. Em seguida, são recuperadas as medidas da tabela de fatos através das instâncias da classe *Measure*, associadas à unidade de análise. As informações de como a tabela de fatos se une às dimensões são extraídas através das instâncias da

classe *DimensionUsage*. Finalmente, são adicionados os filtros definidos na análise.

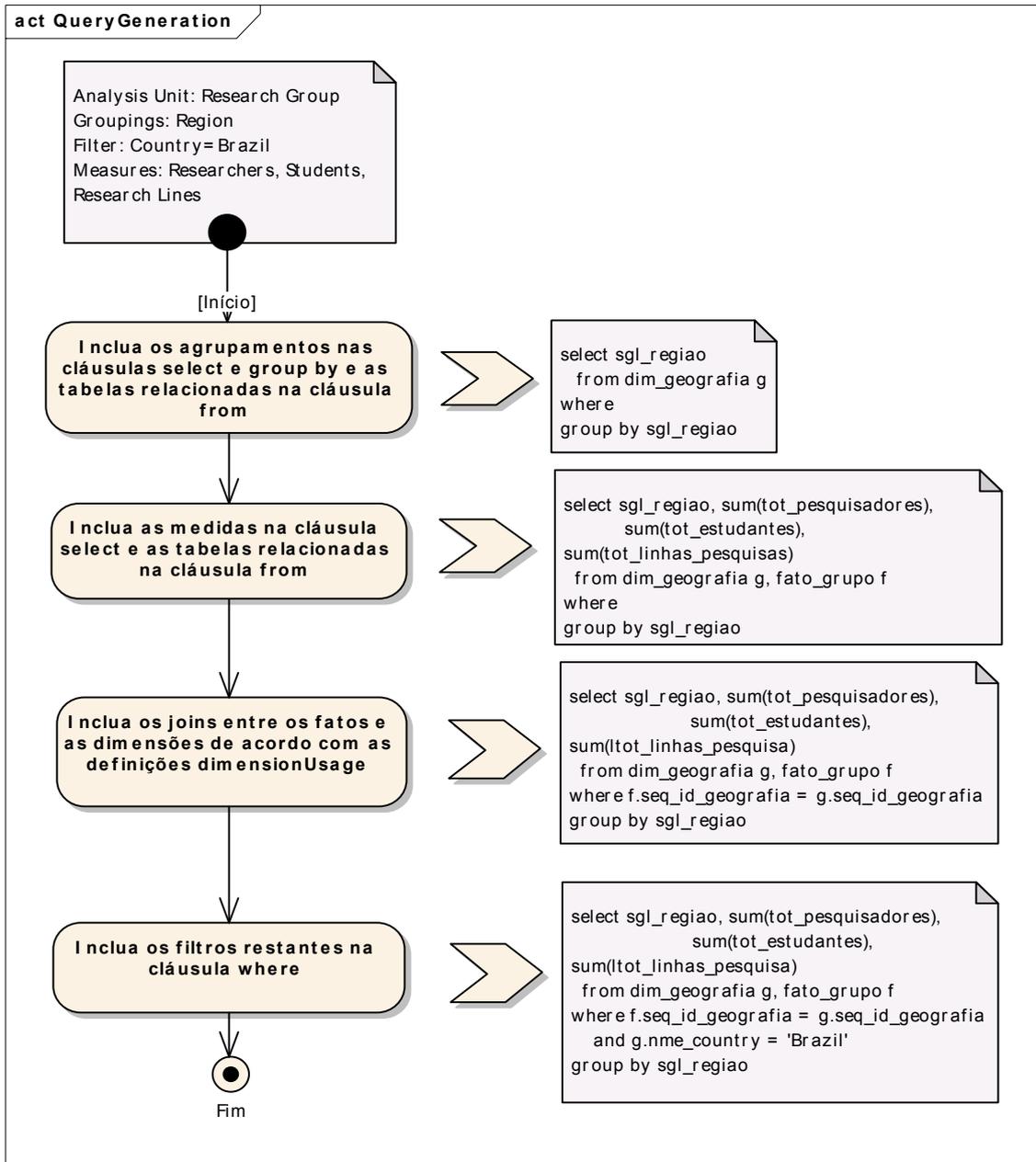


Figura 28 - Processo de geração de uma consulta a partir das definições de uma análise

6.4.2 Realizando cortes semânticos e reescrevendo análises

O Gerenciador de Análises implementa um método que permite que uma análise seja redefinida a partir do resultado de uma inferência, quando um usuário desejar fazer um corte ou ampliar o escopo de dados reunidos em uma análise com base em um filtro semântico. Esse método é similar ao descrito em

Paton et al. (1999), que reescrevem uma consulta utilizando sinônimos dos termos informados para expandir os resultados da consulta através de conectores lógicos do tipo *OR*. Entretanto, o método utilizado nesta pesquisa é mais abrangente, pois oferece diversas opções ao usuário para a expansão ou filtragem dos resultados de uma consulta. Os usuários selecionam, entre as relações, expressões lógicas e regras associadas a um conceito incluído na lista de dimensões ou nos parâmetros de uma análise e solicita a sua execução. O resultado do processamento da inferência é utilizado para redefinir a consulta original da análise.

A abordagem utilizada para suportar a redefinição de consultas permite que a semântica do negócio seja aplicada na busca de diferentes perspectivas sobre uma análise. Por exemplo, toma-se um pró-reitor de pesquisa que define uma análise da evolução da produtividade dos pesquisadores de sua instituição. A partir da identificação nessa análise dos pesquisadores que mais produziram, o usuário poderia filtrar os pesquisadores relacionados à mesma área de pesquisa dos campeões de publicação no período selecionado. Essa análise serviria para verificar se a elevação da produtividade se manteve nessas áreas de pesquisa como resultado de alguma ação institucional. Nesse exemplo, assume-se que a categorização de áreas de pesquisa não está descrita nas dimensões do DW, mas expressa como uma taxonomia na Ontologia do Domínio. Seguindo esse exemplo, a redefinição da consulta seria feita da seguinte maneira:

1. após visualizar a evolução da produtividade, o usuário seleciona os pesquisadores que mais publicaram no período e pede sugestões de navegação;
2. através da execução do método *getRelations* do Gerenciador de Ontologias, o usuário visualiza todas as relações definidas sobre o conceito *person* na Ontologia do Domínio e seleciona a relação *has_person_research_area*;
3. a relação *has_person_research_area* é processada pelo Gerenciador de Ontologia identificando que dentre as linhas de pesquisa dos pesquisadores selecionados figura a linha “Knowledge Representation”, que, por sua vez, está associada

ao conceito *knowledge_area* através de uma relação transitiva. Como resultado da inferência é apresentada a área “Knowledge Engineering”;

4. o usuário continua sua análise, selecionando dentre as relações e regras definidas para o conceito *knowledge_area* a relação *has_researchers*. O resultado dessa inferência é a lista dos pesquisadores que estão ligados à área de conhecimento “Knowledge Engineering”;
5. as chaves primárias dos pesquisadores inferidos são incluídas como restrição na consulta original e outras restrições relacionadas à tabela correspondente ao conceito *person* – a dimensão “dim_pessoa” – são removidas da consulta original; e
6. a nova consulta é submetida ao SGBD, e o resultado é apresentado ao usuário.

Basicamente, o processo de redefinição de consultas nesta arquitetura funciona como no exemplo acima, descrito para qualquer tipo de inferência. Os usuários selecionam, entre as relações, expressões lógicas e regras definidas para um conceito relacionado a um cubo de dados e solicitam a sua execução. A lista de opções para cada conceito é obtida pelo Gerenciador de Ontologia através dos métodos *getRelations()* e *getSemanticFilters()*, descritos na Seção 6.2.

O processamento das inferências é realizado pelo Gerenciador de Ontologias em conjunto com o Mecanismo de Inferência integrado à arquitetura SBI⁴. O resultado da inferência é apresentado ao usuário, que, por sua vez, pode novamente selecionar uma relação, expressão lógica ou regra associada aos conceitos retornados pela última inferência. Quando o usuário estiver satisfeito com seus *drills* e *slices* semânticos, pode ainda solicitar a redefinição do cubo original a partir do resultado das suas inferências.

⁴ Na versão corrente da arquitetura SBI, o mecanismo de inferência utilizado corresponde ao interpretador OCML.

A redefinição da consulta do cubo original corresponde aos itens 5 e 6 da seqüência de passos apresentada anteriormente e é descrita de forma mais detalhada na Figura 29. Primeiramente verifica-se qual é a classe associada às instâncias resultantes da última inferência solicitada pelo usuário (no exemplo apresentado, uma instância da classe *person*). Em seguida, é identificada a tabela associada ao conceito retornado, verificando as instâncias da classe *DB_Attribute* que possuem a propriedade *has_concept* igual a *person*, além da instância da classe *Dimension* correspondente (neste exemplo, a instância *dimension_dim_pessoa*) e da identificação da chave primária desta dimensão (o atributo *seq_id_pessoa*). Os valores correspondentes ao atributo *seq_id_pessoa* das instâncias de *person* retornadas são incluídos na cláusula *Where* da consulta original através de uma instrução *IN* (e.g. *seq_id_pessoa IN (1001, 1050, 3000, 4020)*). Caso houvesse outra restrição relacionada à dimensão *pessoa* na consulta original, esta seria eliminada. A consulta redefinida é então submetida ao gerenciador de banco de dados para ser processada, e o resultado é apresentado ao usuário. O código SQL da consulta original após a redefinição é apresentado na Listagem 10.

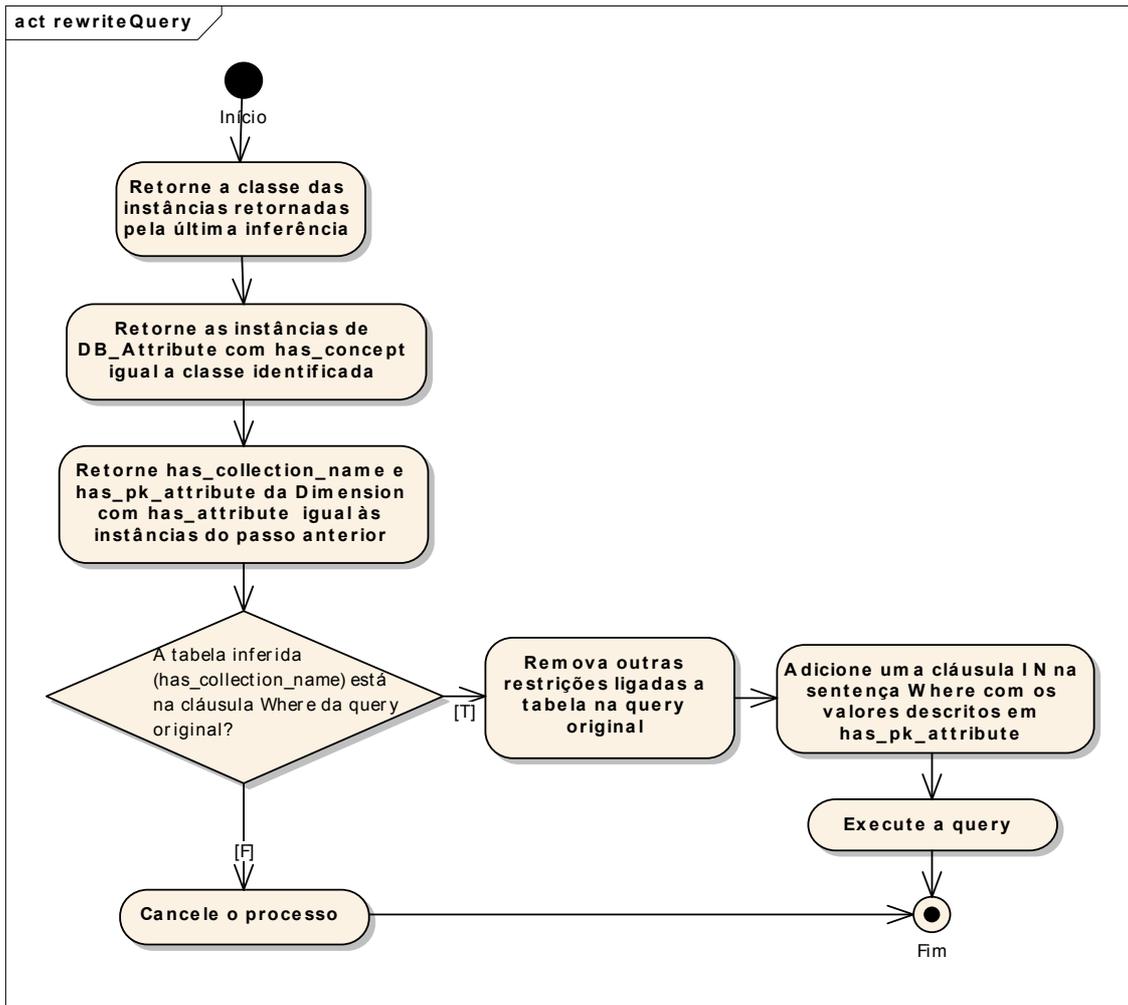


Figura 29 - Descrição do processo de redefinição de consultas

Listagem 10 - Demonstração de uma consulta antes e após o processo de redefinição

```

select nme_pessoa, sum(tot_producao)
  from fato_producao f, dim_pessoa p, di_tempo t
 where f.seq_id_pessoa = p.seq_id_pessoa
       and f.seq_id_tempo = t.seq_id_tempo
       and t.ano = 2005
       and t.nro_trimestre = 3
 order by sum(tot_producao)

select nme_pessoa, sum(tot_producao)
  from fato_producao f, dim_pessoa p, di_tempo t
 where f.seq_id_pessoa = p.seq_id_pessoa
       and f.seq_id_tempo = t.seq_id_tempo
       and p.seq_id_pessoa IN (1001, 1050, 3000, 4020)
       and t.ano = 2005
       and t.nro_trimestre = 3
 order by sum(tot_producao)
  
```

A maior vantagem da abordagem adotada nesta pesquisa em relação à abordagem utilizada por Mena et al. (2000), Paton et al. (1999) e Necib e

Freitag (2003) é que a redefinição não se limita ao uso dos sinônimos e hipônimos, podendo ser utilizado qualquer tipo de relação ou expressão lógica descrita na Ontologia do Domínio, o que resulta em maior flexibilidade para o tomador de decisão analisar seus repositórios de dados. O exemplo descrito anteriormente demonstra como uma relação transitiva suportou *drill up* e *drill down* semânticos. Em aplicações analíticas tradicionais, os relacionamentos entre fatos e dimensões guiam o funcionamento de *drill through* e *drill out*. Relações simétricas e associativas podem suportar essas operações sem necessariamente haver ligação entre fatos e dimensões. Por exemplo, uma relação associativa do tipo “*trabalha em*” entre o conceito *pesquisador* e o conceito *instituição* poderia apoiar a recuperação das produções científicas de um dado pesquisador na biblioteca digital daquela instituição.

Outras relações mais sofisticadas baseadas em expressões lógicas podem suportar inferências ainda mais poderosas sobre os repositórios de dados. Essas expressões podem ser usadas para suportar questões como “qual foi a produtividade dos meus *competidores* no último trimestre?”, sendo *competidor* uma expressão lógica especificada na Ontologia do Domínio, e descrita conforma apresentado na Listagem 1. Dessa forma, definições semânticas conhecidas pelos tomadores de decisão podem ser descritas na Ontologia do Domínio e utilizadas para apoiar o processo de navegação das fontes de informação e, conseqüentemente, apoiar o processo decisório.

Na abordagem tradicional de desenvolvimento de projetos de BI, para cada redefinição de regra de negócio, ou para a incorporação de uma nova definição de regra de negócio como a expressão *competidores* citada no parágrafo anterior, seria necessário criar uma nova tabela de fatos e desenvolver uma rotina de carga específica para essa tabela, como descrito em Kimball et al. (1998; 2002). Esse desenvolvimento, por depender diretamente da equipe técnica e demandar um tempo considerável, é em muitos casos custoso e moroso (Kimball et al., 1998). Procura-se na arquitetura SBI oferecer uma alternativa mais flexível a mudanças nas regras, buscando-se maior velocidade e menor custo de desenvolvimento a partir da interligação dos módulos da arquitetura a uma ontologia criada especificamente para representar as regras e conceitos do negócio do usuário. Essa ontologia (*i.e.*,

Ontologia do Domínio) pode ser mantida com facilidade por um membro da equipe de desenvolvimento ou até mesmo pelo usuário, desde que este receba um treinamento sobre ferramentas de manutenção de ontologias. Esse cenário se torna ainda mais promissor com a evolução das ferramentas para manutenção de ontologias, como WebOnto (DOMINGUE, 1998), Prótegé (PROTEGE, 2005) e OntoEdit (SURE et al., 2002). Essas ferramentas facilitam as tarefas de manutenção de ontologias por permitir que seus usuários realizem modificações nas definições dos conceitos e das regras de negócio através de cliques de mouse, sem a necessidade de conhecer detalhes técnicos e a linguagem utilizada na criação da ontologia.

6.4.3 Recomendação de análises

A interface *AnalysisManager* prevê dois métodos para a localização de análises semelhantes ao escopo da análise corrente do usuário, os métodos *findAnalysysByConcept* e *findAnalysysByFilter*. Esses dois métodos permitem que sejam reempregadas as sistemáticas adotadas por outros tomadores de decisão em situações de análise semelhantes.

O método *findAnalysysByConcept* recupera análises que possuam na sua definição (*i.e.*, em suas dimensões, filtros ou medidas) o conceito informado como parâmetro de entrada para o método. Para tanto, o método utiliza o método *compareConcepts* do Gerenciador de Ontologias para realizar a comparação entre o conceito informado e os utilizados na definição das dimensões, filtros e medidas das análises mantidas na Ontologia BI. Esse método é útil quando da necessidade de se recomendar ao usuário outras análises que envolvam uma determinada instituição, pessoa ou outro conceito de interesse ao tomador de decisão. Por exemplo, poderia ser do interesse do usuário de uma ferramenta analítica identificar quais foram as análises definidas por outros tomadores de decisão para avaliar os seus competidores.

O método *findAnalysisByFilter* permite a localização de análise a partir da definição de um filtro especificado segundo a sintaxe descrita no Quadro 18. A formatação do filtro e o processamento de localização de análises deste método são semelhantes ao método *findGoalbyFilter*, apresentado na Seção 6.3.1.

Quadro 18 - Formatação de filtros para a localização de Análises na Ontologia BI através do método *findGoalbyFilter*

Sintaxe para a formatação de filtros para a localização de análises:

```
(<logical operator> (property1 value1 ?service) ... (propertyn valuen ?service))
```

Onde:

- `logical operator` corresponde ao operador lógico OR ou AND;
- `property` corresponde a um elemento da análise que deseja-se filtrar (i.e. `dimension`, `filter`, `measure`, `author`, entre outras);
- `value` corresponde ao valor utilizado para filtrar a propriedade (i.e. um conceito, uma determinada instância, uma `string`, etc).

Por exemplo, para localizar uma análise que possui como dimensão o conceito `country` e como medida `researcher`:

- `(and (dimension country) (measure researcher))`

6.5 Considerações sobre o Capítulo

Neste capítulo foram apresentados os módulos funcionais que permitem o acoplamento de tecnologias semânticas às aplicações analíticas. Cada módulo implementa uma série de métodos visando atender aos requisitos funcionais e não funcionais definidos no Capítulo 4 desta tese.

De forma geral, o Gerenciador de Ontologias foi concebido para prover aos demais módulos da arquitetura um acesso transparente às ontologias da arquitetura, conforme o requisito RNF2, e para prover suporte ao atendimento a todos os requisitos funcionais especificados. Os métodos definidos neste módulo permitem total independência dos conceitos representados na Ontologia do Domínio, possibilitando a sua aplicação sobre qualquer domínio de negócio, conforme prescrito no requisito RNF1. Por fim, a flexibilidade para a seleção de mecanismos de inferência de preferência da organização facilita o atendimento do requisito RNF3 (integração a soluções de terceiros).

O Gerenciador de Serviços provê métodos para o suporte à localização e recomendação de serviços visando atender ao requisito RF6 para o acoplamento de serviços que ofereçam novos recursos analíticos, conforme o requisito RF4, e para a montagem e a execução de composições de serviços, conforme o requisito RF5. A flexibilidade de incorporação de novas funcionalidades facilita o atendimento de necessidades específicas de uma organização, apoiando o atendimento do requisito RNF1 e a integração a soluções de terceiros, conforme o requisito RNF3.

Características Apoiadas pelos Módulos Funcionais	Requisitos Funcionais						Requisitos Não Funcionais		
	1	2	3	4	5	6	1	2	3
Redefinição de consultas a partir de inferências sobre os conceitos e as regras do negócio	X	X				X			
Navegação sobre os dados através de temas de análise e conceitos do negócio	X								
Suporte para a localização e recomendação de análises						X			

7 DEMONSTRAÇÃO DE VIABILIDADE E ANÁLISE COMPARATIVA

7.1 Apresentação

Neste capítulo é apresentado OntoDSS, um protótipo desenvolvido em Java com o objetivo de demonstrar todos os novos recursos que a arquitetura SBI introduz no contexto do processamento analítico. OntoDSS introduz uma série de recursos de análise exploratória baseados em tecnologias semânticas que são inéditos no contexto de ferramentas analíticas e que possibilitam maior poder de análise e mais flexibilidade de customização sobre operações de *drill* e *slice*. A apresentação das funcionalidades do OntoDSS é feita à luz de um cenário de aplicação real, no contexto da gestão da Ciência e Tecnologia. Por fim, apresenta-se um quadro comparativo que sumariza as diferenças entre a arquitetura SBI e outras iniciativas acadêmicas e da indústria de software.

7.2 A ferramenta OntoDSS

A ferramenta OntoDSS é uma ferramenta OLAP desenvolvida para ilustrar como a representação semântica das informações e dos serviços de interesse à organização, além da representação das suas regras de negócio, torna possível inferências semânticas não exploradas até então nas soluções analíticas atuais. Ilustra-se ainda como serviços de recuperação e tratamento de informação podem ser acoplados à ferramenta de forma flexível e automaticamente sugeridos aos usuários de acordo com o contexto de suas análises.

A seguir, descreve-se um cenário real de aplicação escolhido para demonstrar as funcionalidades da ferramenta OntoDSS e do potencial ofertado pela arquitetura SBI. Nas seções seguintes, apresentam-se os módulos do OntoDSS.

7.2.1 Cenário de aplicação

As funcionalidades do OntoDSS são ilustradas no contexto do apoio à decisão para gestores de Ciência e Tecnologia (C&T). Para tanto, foi criado um *Data mart* a partir de uma amostra de 11.129 mil currículos de pesquisadores da Plataforma Lattes (PL), pertencentes a pesquisadores e estudantes e

obtidos junto a uma instituição de ensino superior do estado de Santa Catarina. A PL é uma plataforma de governo eletrônico constituída de um conjunto de sistemas e repositórios de dados que facilitam e integram as atividades de fomento, gestão e planejamento em C&T (CNPq, 2003). O objetivo da PL é racionalizar o processo de gestão de C&T, tanto do ponto de vista dos estudantes e pesquisadores quanto das agências de fomento e das instituições de ensino e pesquisa do País (REVISTA DA PLATAFORMA LATTES, 2002).

O estudo de caso visa ilustrar como a arquitetura e o protótipo podem ser utilizados para apoiar as necessidades de pró-reitores de pesquisa em instituições de ensino na avaliação da produtividade dos pesquisadores de suas instituições e no *benchmarking* com outras instituições de ensino. As informações curriculares reunidas no *Data mart* poderiam subsidiar o processo decisório de outros atores ligados à C&T, como agências de fomento, pesquisadores e empresas, dado o interesse comum desses atores no conjunto de dados reunidos nos currículos de pesquisadores e estudantes. No entanto, para fins de ilustrar as possibilidades analíticas inéditas introduzidas pela arquitetura SBI, foi escolhido apenas o ator “pró-reitor” para descrever o processo de formatação de análises e de recomendações nas seções a seguir.

Para a execução do protótipo, foi representado um conjunto de conceitos e regras de negócio relacionadas à C&T na Ontologia do Domínio, dentre os quais se destacam os ilustrados na Figura 30. Sucintamente, uma pessoa pode atuar como líder, pesquisador ou estudante de um grupo de pesquisa (*research_group*). Um grupo de pesquisa está associado a uma instituição de ensino ou pesquisa (*institution*). Uma instituição pode ter vários níveis departamentais. Uma pessoa reside em uma localidade (*i.e.* em uma cidade – *city*, estado – *state* e país – *country*), atua em instituições de ensino e pesquisa (*activity*) e possui titulações acadêmicas (*degree*). Os cursos (*course*) nos quais foram obtidas as titulações acadêmicas podem estar associados a várias áreas do conhecimento (*knowledge_area*).

Além da representação de conceitos, foram modeladas na Ontologia do Domínio relações e expressões lógicas utilizadas para representar as regras de negócio de interesse de pró-reitores no contexto da gestão da C&T, como as apresentadas na Listagem 1. A lista completa das relações e dos construtores

utilizados na Ontologia do Domínio para o estudo de caso é apresentada no Apêndice E.

Os conceitos reunidos na Ontologia do Domínio foram mapeados manualmente às tabelas de fatos e dimensões do *Data mart* através dos construtores da Ontologia BI. Esse mapeamento foi utilizado para suportar a geração de instâncias dos conceitos da Ontologia do Domínio a partir dos dados das dimensões do *Data mart*. Esse processo foi realizado através de scripts desenvolvidos pelo autor especificamente para criar uma amostra de conceitos para análise de viabilidade da arquitetura. O mapeamento feito na Ontologia BI é apresentado no Apêndice A. O modelo de dados do *data mart* do estudo de caso é apresentado no Apêndice F.

Foram ainda criadas as descrições semânticas de dezenove serviços na Ontologia de Serviços para demonstração da integração de WSS ao protótipo. Entre os serviços prototipados para ilustrar a integração de dados no contexto analítico, encontram-se serviços para listar os grupos de pesquisa de uma dada instituição, para identificar indicadores de desempenho críticos, para comparar indicadores de produtividade com outras instituições, além de outros prototipados em Lisp e em Java. A Ontologia de Serviços criada para o estudo de caso é apresentada no Apêndice D.

Para a execução dos testes, foi utilizado um computador para manter a instalação dos módulos da arquitetura SBI e o framework OCML e outro computador para manter o sistema gerenciador de banco de dados. A descrição do hardware de cada computador e dos softwares instalados é apresentada no Quadro 20.

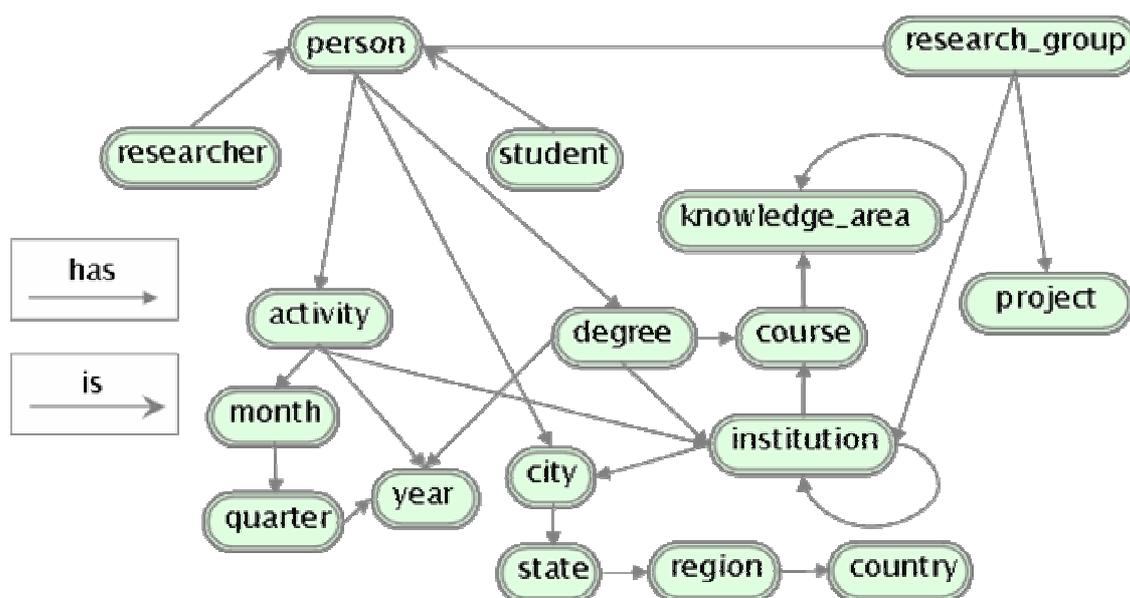


Figura 30 - Ilustração de conceitos associados à gestão de C&T e representados na Ontologia do Domínio

Quadro 20 - Descrição do hardware e software utilizado no estudo de caso

Servidores	Especificação	
(1) Arquitetura SBI, OntoDSS e Framework OCML	Hardware	1 Intel Pentium 4 3.2 Hyper-threading
		1 Gb RAM, 1 HD 80 Gb
	Software	Windows XP Professional
		OCML v5.0 e Xanalys LispWorks 4.3
		Java 2 SDK 1.4.2
(2) SGBD Oracle	Hardware	Dual processor Xeon 3.4
		4 Gb RAM, 6 HDs 147Gb SCSI, RAID 5
	Software	Linux Suse Enterprise 9
		Oracle Enterprise Edition 10.2.0.1.0g
		Java 2 SDK 1.4.2

7.2.2 Os módulos da ferramenta OntoDSS

OntoDSS é dividido em três módulos principais, como descrito a seguir.

- **Módulo de definição de análises:** permite que os usuários explorem as definições semânticas de suas fontes de dados para escolher o conteúdo e o formato das suas análises.
- **Módulo de assistência à análise:** apresenta alternativas de explorações (filtros semânticos, inferências e serviços) para o usuário, de acordo com o contexto de suas análises.

- **Módulo de composição de serviços:** permite a composição de WSS para a definição de explorações mais sofisticadas nas quais é necessário utilizar mais de um WSS para o atendimento dos objetivos na análise.

7.2.3 Módulo de definição de análises

O módulo de definição de análises provê os meios para os usuários optarem pelo conteúdo e pelo formato de suas análises. Para tanto, os usuários navegam sobre as definições semânticas de suas fontes de dados usando a Ontologia BI, com vistas a selecionar os elementos de dados que serão incluídos em suas análises.

A opção de oferecer a representação semântica das fontes de dados ao usuário em detrimento da representação técnica dessas fontes (*i.e.*, maneira pela qual estão organizadas as tabelas do DW) visa facilitar o entendimento dos dados disponíveis para utilização e facilitar a incorporação de novas fontes de dados à arquitetura BI com total transparência para os usuários finais. A qualquer momento novos *data marts* podem ser disponibilizados pelas ferramentas analíticas, bastando criar as anotações semânticas na Ontologia BI. Além disso, a representação semântica dos dados incluídos na análise subsidiará o processo de recomendação automática discutido na próxima Seção.

A definição de uma análise no OntoDSS compreende a seleção dos itens de dados que irão compor as dimensões (*i.e.*, cabeçalhos de linha e de coluna), medidas (*i.e.*, pontos de dados no cubo) e os respectivos operadores de sumarização (*e.g.*, média, soma, contagem etc.), além dos filtros (*i.e.*, restrições aplicadas sobre as dimensões e medidas) e dos parâmetros (*i.e.*, variáveis de ligação utilizadas para filtros dinâmicos).

A Figura 31 ilustra a interface para definição de análises. A escolha das medidas é feita na aba *Measures*, na qual são apresentadas todas as unidades de conteúdo descritas na Ontologia BI. As dimensões são definidas na aba *Dimensions*, na qual são apresentadas as variáveis de agrupamento definidas na Ontologia BI. Nas abas *Filters* e *Parameters*, o usuário poderá definir quais variáveis de agrupamento serão utilizadas como filtro e como parâmetros para

a análise. Os conceitos apresentados em cada aba para o tomador de decisão utilizam rótulos inferidos de acordo com o perfil do usuário, no exemplo da Figura 31 é apresentado um usuário do grupo *researcher-manager*.

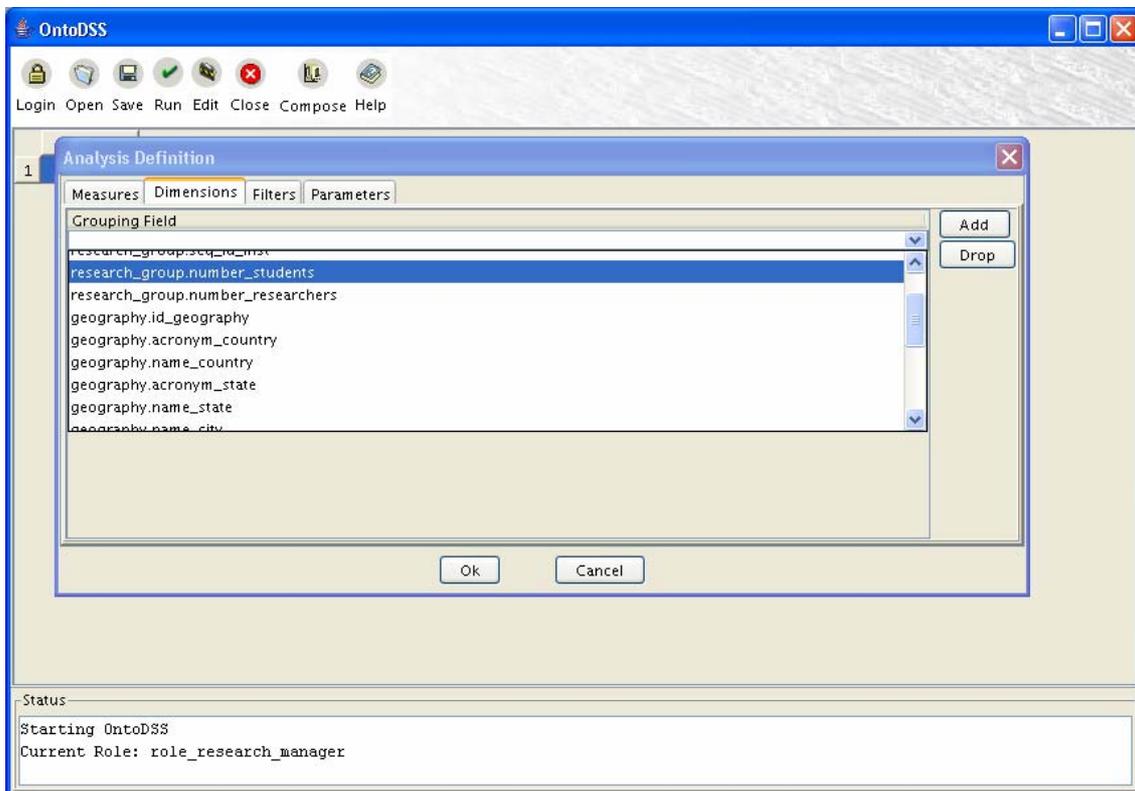


Figura 31 - Interface para definição de análises no OntoDSS

O processo de montagem da interface de definição de análises do OntoDSS é apoiado pelos métodos do Gerenciador de Análises da arquitetura SBI. Conforme ilustrado na Figura 32, inicialmente são recuperadas as unidades de análise definidas na Ontologia BI através do método *getAnalysisUnit()* da classe *AnalysisManager*. Na seqüência, para cada Unidade de Análise (*i.e.*, instâncias da classe *AnalysisUnit* recuperadas), são extraídas suas medidas (*i.e.*, instâncias da classe *Measure* através do método *getMeasureList()*) e suas definições de filtros e agrupamentos através do método *getDimensionUsage()*. A classe *DimensionUsage* confere o acesso às dimensões (e, por conseqüência, às suas propriedades e hierarquias) e à descrição das tabelas nas quais as dimensões são mantidas, além de informar sobre os joins necessários para recuperar os dados das dimensões (*i.e.*, informações da classe *CollectionJoin*).

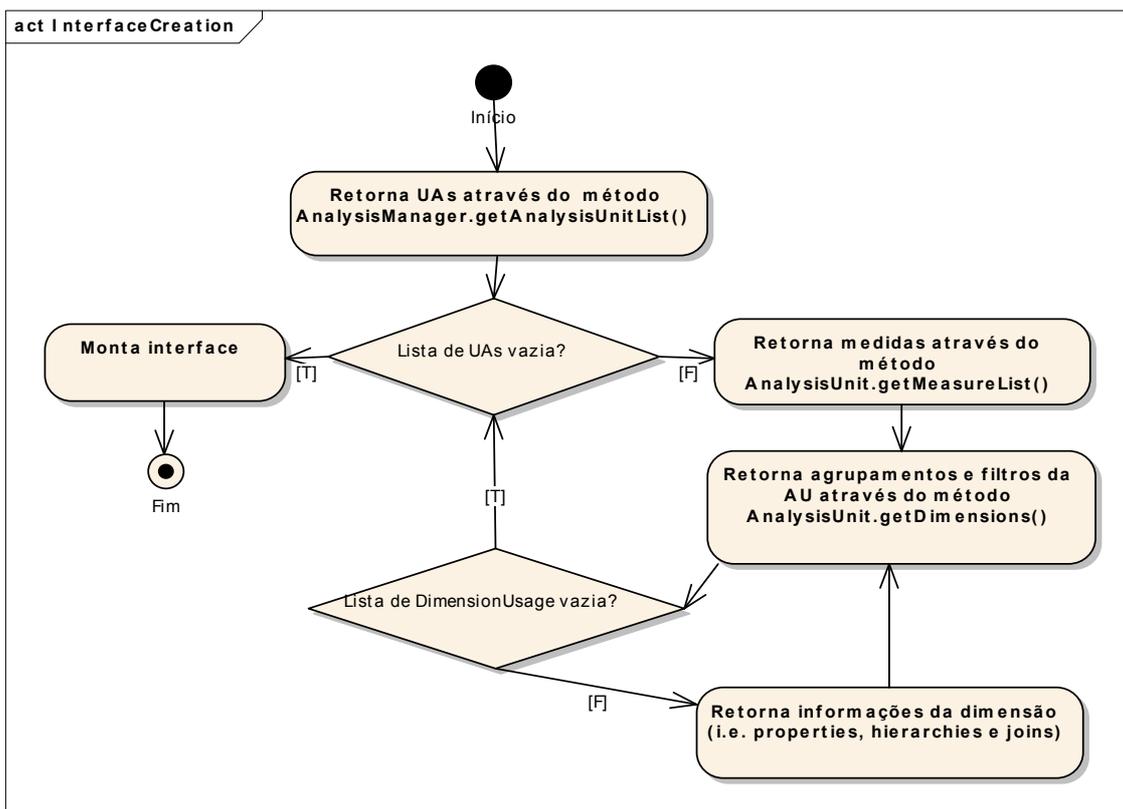


Figura 32 - Recuperação dos temas de análises definidos na Ontologia BI

A nomenclatura utilizada para apresentar as fontes de dados aos usuários é inferida das relações existentes na Ontologia BI entre as tabelas e o perfil de cada usuário. Essa nomenclatura personalizada visa facilitar a compreensão do significado do dado disponível nas fontes de dados e o entendimento do conteúdo representado em uma análise. Por exemplo, um pró-reitor de uma instituição irá visualizar os tipos de pessoas no DW como “pesquisador” e “estudante”. Já um gerente de uma companhia irá visualizar os mesmos elementos de dados como “especialista” e “*trainee*”, respectivamente.

O processo de recuperação dos rótulos dos conceitos, de acordo com o perfil do usuário, é implementado pelo Gerenciador de Ontologias através do método *retrieveLabel*. Esse método executa a relação *label-role-idiom* definida na Ontologia BI e apresentada na Listagem 4.

A definição semântica da análise é traduzida pelo Gerenciador de Análises em uma sentença SQL, conforme processo descrito na Seção 6.4.1. A sentença é então processada junto ao DW, e os resultados da consulta são apresentados ao usuário, conforme ilustrado na Figura 33.

7.2.4 Módulo de assistência à análise

Este módulo oferece ao usuário um conjunto de alternativas para a exploração dos resultados de uma análise. Para tanto, o módulo baseia-se na representação semântica dos dados, dos serviços e da lógica do negócio descrita nas três ontologias da arquitetura.

Para ilustrar as funcionalidades exploratórias e as recomendações efetuadas pela ferramenta com o apoio da arquitetura SBI, toma-se uma análise criada no módulo de definição de análises por um pró-reitor de pesquisas com o perfil *research_manager*. A análise apresenta o total de pesquisadores e estudantes por instituição. Na definição da análise, a sigla da instituição foi empregada como uma dimensão e a contagem do número de pesquisadores e de estudantes, como medidas. Como filtro, foi definido que somente as instituições da Unidade Federativa “Santa Catarina” deveriam ser incluídas no relatório.

As funcionalidades oferecidas são apresentadas de acordo com o contexto da análise definida pelo usuário. A Figura 33 ilustra os resultados da análise em questão. O OntoDSS automaticamente identifica quais conceitos foram utilizados na definição da análise (neste caso *institution*, *researcher* e *student*) através da relação entre as unidades de análise, o filtro e as medidas e os conceitos definidos na Ontologia do Domínio.

Assim, se o usuário clicar sobre qualquer célula do resultado da análise, o OntoDSS irá apresentar uma lista de recomendações, levando em consideração o conceito sobre o qual o usuário clicou. Na Figura 33 é apresentado o menu principal da lista de funcionalidades recomendadas por OntoDSS ao usuário. Essas funcionalidades estão relacionadas à célula “Univali”. E “Univali” corresponde a uma instância do conceito *institution* na Ontologia do Domínio.

A lista de recomendações sumarizada no Quadro 21 leva em conta não só o conceito *institution*, mas também seus sinônimos e desdobramentos em taxonomias (*i.e.*, conceitos que são filhos e pais nas taxonomias criadas na Ontologia do Domínio).

The screenshot shows the OntoDSS application window. At the top, there is a menu bar with options: Login, Open, Save, Run, Edit, Close, Compose, Help. Below the menu bar is a toolbar with icons for file operations. The main area contains a table with three columns: 'university_acronym', 'number_students', and 'number_researchers'. The table lists 20 universities, with 'UNIVALI' selected and a dropdown menu open showing options: Semantic Filters, Semantic Browsing, Related Analysis, Related Services, All Services, Search Services, and Compose Services. Below the table is a status bar showing 'Starting OntoDSS' and 'Current Role: role_research_manager'.

	university_acronym	number_students	number_researchers
2	UNOCHAPECO	930	14
3	UNIVILLE	1,300	66
4	UNIVALI	5,021	71
5	UNISUL	7,032	39
6	UNIPLAC	3,562	30
7	UNIFACRI	1,770	22
8	UNIDAVI	3,010	40
9	UNESC	1,233	13
10	UNERJ	2,013	49
11	UNC	1,461	19
12	UFSC	10,500	320
13	UDESC	5,010	43
14	IEE	76	4
15	IBES	220	9
16	FURB	6,567	92
17	ETT	209	11
18	ES	120	12
19	CES	54	4
20	CEFET	67	8

Figura 33 - Ilustração do resultado de uma análise no OntoDSS e do menu de recomendações

Quadro 21 - Sumário das funcionalidades recomendadas por OntoDSS de acordo com o contexto de uma análise

OPÇÃO	DESCRIÇÃO
Semantic Filters (Filtros Semânticos)	<p>Nesta opção, será acionado o método <i>getSemanticFilters</i> (Seção 6.2) do Gerenciador de Ontologias. A execução deste método retorna a lista de regras de negócio que o usuário pode utilizar para realizar explorações adicionais sobre suas análises. Essas regras são representadas na Ontologia do Domínio como regras de produção ou expressões lógicas e são processadas pelo mecanismo de inferência acoplado à arquitetura SBI, na versão atual, o da linguagem OCML.</p> <p>Conforme ilustrado na Figura 34, a primeira opção de inferência apresentada foi <i>institution-competitor</i>, uma expressão lógica apresentada na Listagem 1. A lista de instituições retornadas pela inferência, levando-se em consideração a instituição “Univali”, é apresentada em outro diálogo.</p> <p>O usuário pode ainda clicar no botão “<i>Rewrite Analysis</i>” (Redefinir Análise) para reformular a análise original de acordo com o resultado da inferência. O processo de redefinição de análises é executado através do método <i>rewriteAnalysis</i> do Gerenciador de Análises (Seção 6.4.2). Neste caso, a reformulação apresentaria a quantidade de pesquisadores e estudantes das três instituições retornadas pelo filtro semântico, além dos números relativos à instituição “Univali”.</p>

OPÇÃO	DESCRIÇÃO
	<p>A possibilidade de utilizar as regras do negócio do usuário para fazer inferências do tipo <i>drill</i> ou <i>slice</i> semânticos permite uma maior aproximação entre as soluções analíticas e as reais necessidades dos tomadores de decisão. Esta funcionalidade não é encontrada nas ferramentas analíticas atuais e virtualmente não foi abordada por nenhuma pesquisa acadêmica no contexto de ferramentas analíticas para BI.</p> <p>Este recurso permite ainda uma maior flexibilidade para a adaptação da solução de BI às necessidades do tomador de decisão. Nas soluções atuais, modificações de regra de negócio exigem a reprogramação das rotinas ETL e o ajuste no modelo de dados do <i>data warehouse</i>. Na arquitetura SBI, o desenvolvedor pode fazer o ajuste rapidamente através da modelagem de uma nova regra de negócio na ontologia do domínio ou ajustando as regras atuais.</p>
<p>Semantic Browsing (Navegação Semântica)</p>	<p>Apresenta todas as relações listadas pelo método <i>getRelations</i> (Seção 6.2) do Gerenciador de Ontologias, que, neste caso, o conceito <i>institution</i> possui. Tais relações podem guiar o usuário em uma navegação sobre os seus repositórios de dados. Para tal, o usuário utiliza as definições semânticas do seu negócio providas pela Ontologia do Domínio. Essa navegação pode também ficar restrita à representação dos conceitos na Ontologia do Domínio.</p> <p>Esta funcionalidade suporta explorações mais flexíveis que as opções de <i>drill</i> oferecidas pelas ferramentas analíticas atuais, porque não dependem das relações e hierarquias previamente definidas nas tabelas do DW. A qualquer momento, novas hierarquias e relacionamentos conceituais podem ser agregados à Ontologia do Domínio e mapeados às fontes de dados na Ontologia BI, o que resulta grande flexibilidade para a modificação da lógica dos <i>drills</i> oferecidos aos usuários. Do ponto de vista do usuário, isso significa ter possibilidades de analisar o seu negócio sob inúmeras perspectivas. Para o desenvolvedor, maior flexibilidade para ajustar as aplicações analíticas às modificações cotidianas nas regras de negócio e na visão conceitual atual dos processos das organizações.</p>
<p>Related Analyses (Análises Relacionadas)</p>	<p>Lista todas as análises previamente definidas pelos tomadores de decisão através das ferramentas analíticas e recuperadas pelo método <i>findAnalysisByConcept</i> do Gerenciador de Análises (Seção 6.4.3). Esse método retorna análises que possuam pelo menos um conceito que corresponda, neste exemplo, ao conceito <i>institution</i>. Esta funcionalidade permite que o usuário continue suas explorações abrindo análises previamente definidas e que possuam relação com a análise atual. Assim, é oferecida ao usuário a possibilidade de reutilizar práticas de análise passadas e, com isso, explorar seu negócio sobre as diferentes perspectivas utilizadas por</p>

OPÇÃO	DESCRIÇÃO
	<p>outros tomadores de decisão em situações semelhantes. Isso permite reter o conhecimento dos tomadores de decisão na organização através da captura dos cenários retratados nos resultados das suas análises e das abordagens utilizadas por eles para obter mais informações a partir desses cenários.</p>
<p>Related Services (Serviços Relacionados)</p>	<p>Nesta opção, são apresentados todos os WSS retornados pelo método <i>findGoal</i> (Seção 6.3.1) do Gerenciador de Serviços que possuam pelo menos uma entrada compatível, no exemplo, com o conceito <i>institution</i>. Conforme ilustrado na Figura 35, o tomador de decisão pode selecionar esses serviços para realizar qualquer tipo de transação, como acessar um repositório de dados externo para a recuperação de dados sobre a instituição selecionada. O resultado do WSS selecionado é apresentado ao usuário e pode ser utilizado como parâmetro de entrada para outra análise, automaticamente recomendada de acordo com a comparação da saída do WSS e dos parâmetros das análises armazenadas na Ontologia BI.</p> <p>Desse modo, torna-se possível oferecer aos tomadores de decisão possibilidades ilimitadas de funcionalidades analíticas a partir da anotação semântica de código existente dentro ou fora da organização. Essa característica torna a solução de BI mais escalável. A qualquer momento, as funcionalidades podem ser alteradas ou novos serviços podem ser adicionados. Esses serviços são, então, descobertos e oferecidos aos tomadores de decisão de acordo com o contexto das suas análises.</p>
<p>All Services (Todos os Serviços)</p>	<p>Lista todos os serviços definidos na Ontologia de Serviços e recuperados pelo Gerenciador de Serviços através do método <i>getGoalsList</i> (Seção 6.3.1) que não necessariamente possuam relação com o conceito <i>institution</i>.</p>
<p>Search Services (Busca de Serviços)</p>	<p>Permite a localização dos WSS que podem suportar explorações adicionais sobre as análises ou realizar qualquer outra ação usando como entrada os dados da análise. Usuários definem os filtros de busca utilizando os atributos e as relações da Ontologia de Serviços. As seleções feitas pelo usuário são informadas ao método <i>findGoalbyFilter</i> (Seção 6.3.1) do Gerenciador de Serviços, o qual traduz a seleção de filtros de busca de acordo com o método para descoberta de serviços da infra-estrutura de WSS acoplada à arquitetura (na versão atual IRS-III). A lista dos WSS localizados é apresentada em um novo diálogo, conforme ilustra a Figura 36.</p> <p>A tendência é a de que a cada dia sejam publicados milhares de novos <i>Web services</i> na <i>Web</i>. Assim, esta opção permite que o usuário localize com mais facilidade os serviços de seu interesse.</p>

É importante destacar que o módulo de assistência a análises pode ser implementado pelo desenvolvedor de outras maneiras. Pode-se optar, por exemplo, por utilizar-se um avatar para a apresentação das recomendações, ou por uma barra de recomendações (e.g. www.amazon.com). A escolha da maneira de apresentação das recomendações ao usuário pode tornar a assistência mais ou menos efetiva na tomada de decisão e deverá levar em consideração questões ergonômicas e de análise do perfil do usuário, questões essas que não fazem parte do escopo desta tese.

A última opção oferecida no menu de funcionalidades é a “*Compose Services*” (i.e., Composição de Serviços). Essa funcionalidade é descrita na próxima Seção.

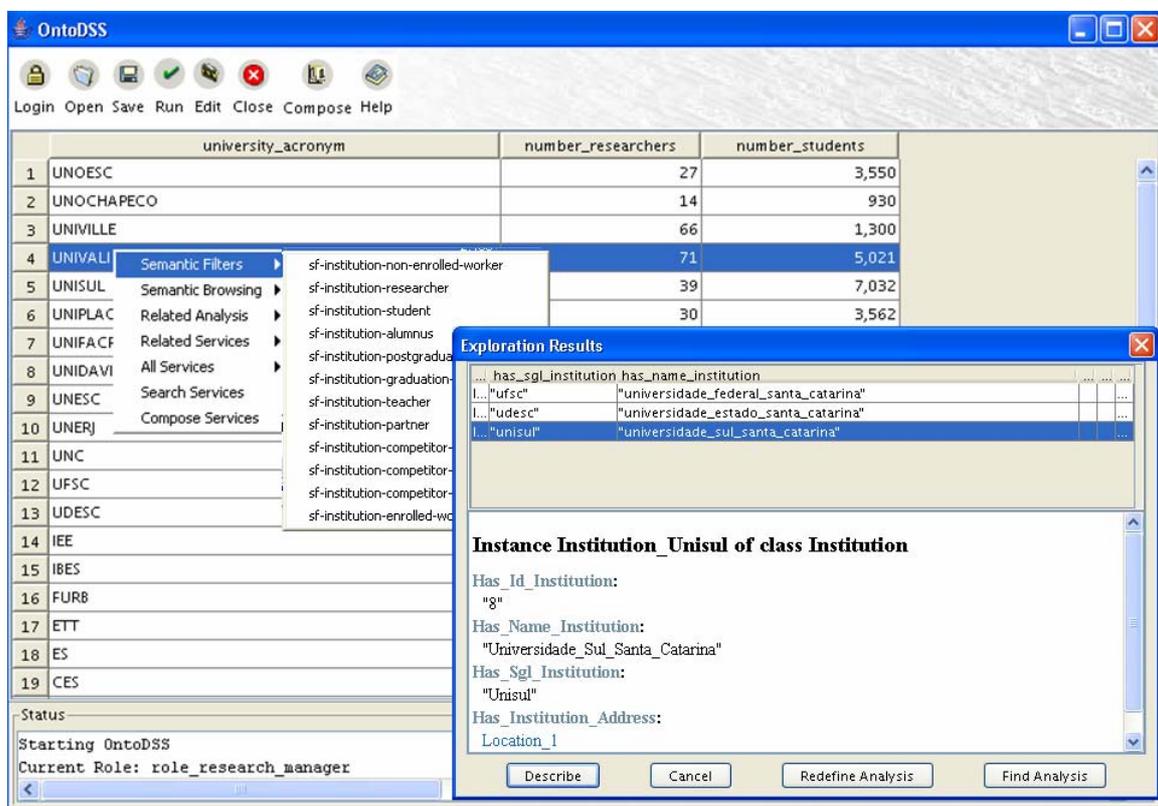


Figura 34 - Listagem dos filtros semânticos relacionados ao conceito *institution* e apresentação parcial do resultado de uma inferência

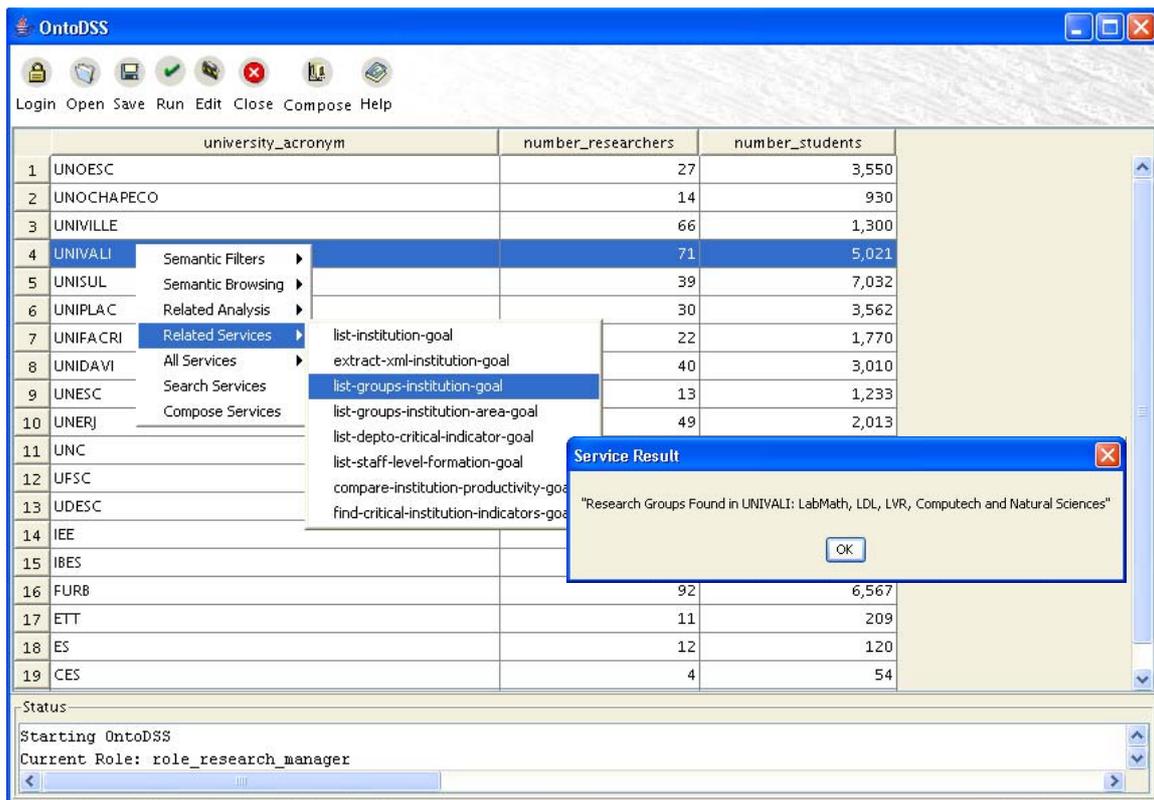


Figura 35 - Listagem dos serviços relacionados ao conceito *institution* e apresentação do resultado da execução do serviço *list-groups-institution*

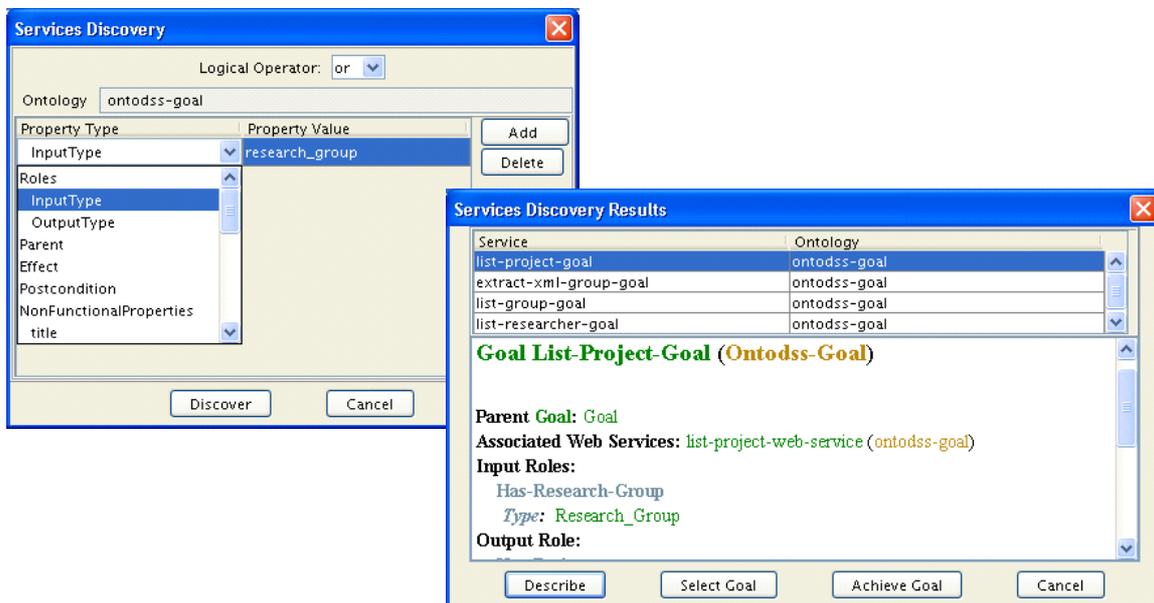


Figura 36 - Ilustração da funcionalidade de busca de WSS

7.2.5 Módulo de composição de *Web services* semânticos

O módulo de composição corresponde a uma ferramenta para composição de WSS. Composições podem ser definidas para suportar o processamento de uma transformação de dados ou de uma consulta mais

elaborada, utilizando como ponto de partida os dados resultantes de uma análise.

Esse módulo suporta um processo interativo e incremental para a definição de composições nas quais WSS e análises são recomendados de acordo com o contexto de uma composição (SELL et al., 2004b).

A Figura 37 ilustra a ferramenta de composição e algumas das suas funcionalidades. A ferramenta guia os usuários em uma composição interativa na qual WSS e operadores de controle de fluxo (condições) são inseridos incrementalmente à composição (item *a* da Figura 37) pelo usuário da ferramenta. Esse processo interativo é suportado pela ferramenta. Em cada passo são recomendados WSS que possuem entradas ou saídas compatíveis com os WSS previamente inseridos na composição (item *b* da Figura 37). Esse processo de recomendação é apoiado pelo método *findGoalByFilter* (Seção 6.3.1) do Gerenciador de Serviços. Mediadores podem ser definidos para realizar qualquer tipo de transformação entre a saída do primeiro WSS e as entradas do segundo WSS (item *c* da Figura 37). Usuários contam ainda com uma funcionalidade para busca de WSS (item *d* da Figura 37).

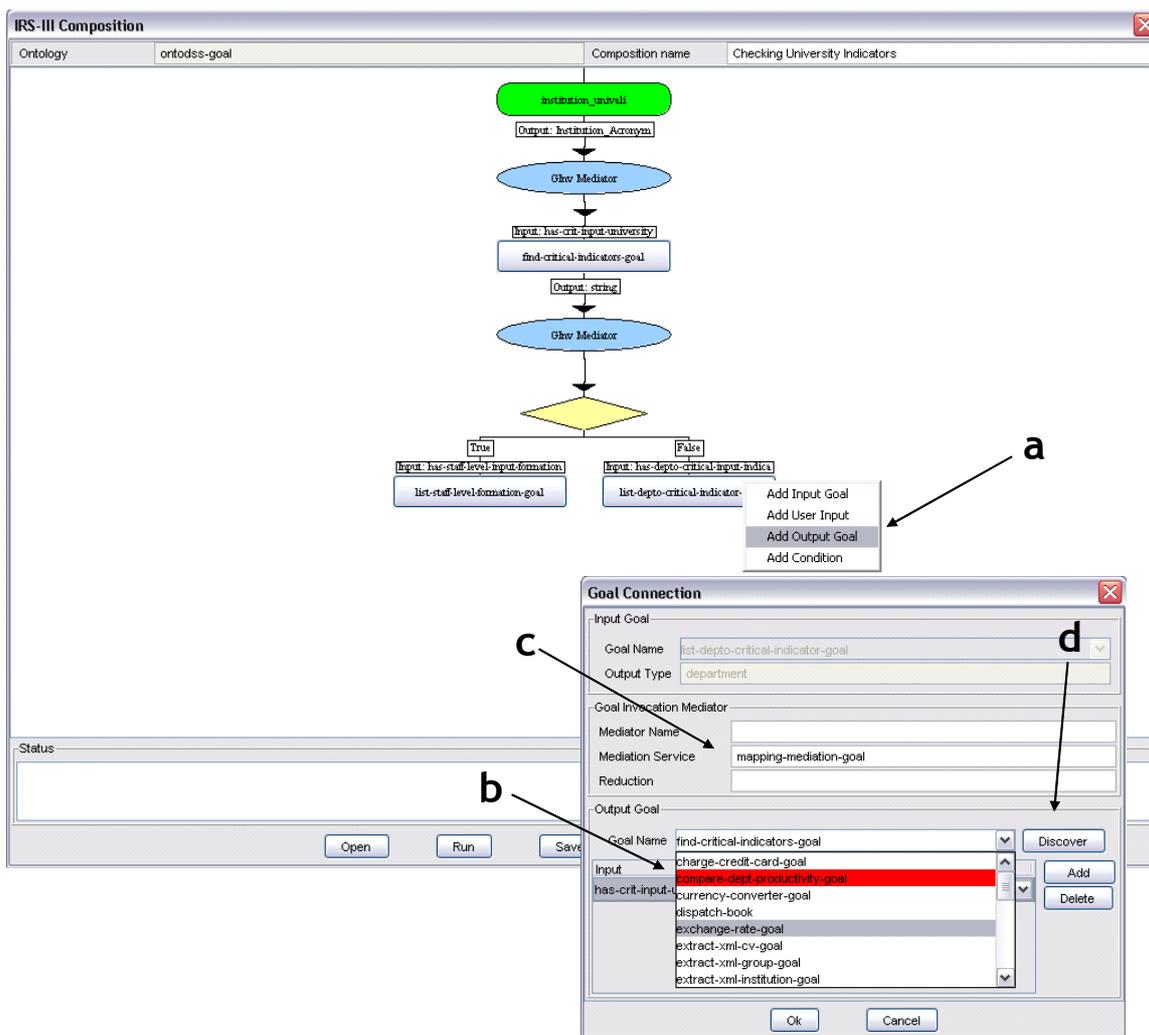


Figura 37 - Ilustração de uma composição definida com o módulo de composição de *Web Services* semânticos

Em cada passo da composição, o usuário pode adicionar WSS que receberão o resultado ou que alimentarão WSS previamente inseridos na composição. Cada WSS pode ter mais de um fornecedor para alimentar seus parâmetros de entrada. Por exemplo, um WSS que possui três parâmetros de entrada pode receber os valores para os dois primeiros parâmetros de dois WSS diferentes e o valor para o terceiro parâmetro a partir de uma entrada de dados feita pelo usuário no momento da composição. Finalmente, os usuários podem definir operadores de controle do tipo *If-Then-Else* à composição para guiar o fluxo de orquestração da composição.

Uma característica importante dessa ferramenta é permitir a utilização de mediadores para mapear e processar transformações entre a saída de um WSS e as entradas de outro WSS. Esses mediadores – *WSMO Mediators* (WSMO, 2005) – podem resolver incompatibilidades relacionadas a dados,

protocolo e processo entre diferentes partes. Além dos mediadores WSMO, foi introduzido nessa ferramenta um mediador de invocação, categoria não existente nas definições WSMO. Esse mediador (*GInv Mediator*) suporta a ligação de dois WSS e foi criado especificamente para suportar a composição de WSS (HAKIMPOUR et. al., 2005; SELL et. al., 2004b).

A incorporação de mediadores na composição resulta em mais flexibilidade para o usuário, dado que é inevitável selecionar serviços definidos e implementados por diferentes partes. Na opinião do autor, o suporte à mediação é um dos principais requisitos para viabilização de composição de serviços e para o suporte de interoperabilidade entre diferentes organizações.

Ao término da definição, a ferramenta instancia a composição através das classes de suporte do Gerenciador de Serviços descritas na Seção 5.3.5. Assim, uma instância da classe *DataFlowComposition* é criada, à qual são agregadas instâncias de *CompositionComponents*, como *ServiceComponents* e *ControlComponents*, além de informações sobre as ligações entre as entradas e a saída da composição e entre os componentes adicionados, através de instâncias de *GInvMediator*. Os resultados da composição são apresentados ao usuário e podem ser usados como parâmetro de entrada para outras análises ou para a redefinição da análise original.

A ordem de execução dos serviços não é definida no momento da composição, mas sim pela API de orquestração, que executa um WSS quando este possuir todos os dados necessários para preencher seus parâmetros de entrada. Durante a orquestração, será solicitado ao usuário entrar com os valores necessários para preencher os dados de entrada que não foram fornecidos no momento da composição e que não podem ser supridos por outros WSS. Cada WSS é então executado pela infra-estrutura WSS acoplada à arquitetura através do Gerenciador de Serviços (IRS-III na versão corrente). Todo o processamento da orquestração pode ser acompanhado pelo usuário através de uma barra de mensagens fornecida na ferramenta de composição.

A Figura 37 ilustra uma composição criada para suportar uma análise complexa envolvendo vários WSS e um operador de controle de fluxo. O primeiro WSS adicionado neste exemplo lista os indicadores que estão abaixo das metas definidas pela instituição previamente selecionada (*i.e.*, instituição

“Univali”). Uma condição é estabelecida para guiar o fluxo de acordo com os indicadores retornados do primeiro WSS visando identificar, através de outros WSS, mais informações sobre as possíveis causas do mau desempenho.

Composições como a ilustrada na Figura 37 podem ser armazenadas na Ontologia de Serviços para posterior utilização, formando desse modo um rico repositório de conhecimento tácito que descreve as abordagens utilizadas por diferentes usuários no processo decisório. Esse conhecimento pode servir para apoiar futuras decisões.

7.3 Análise sobre a flexibilidade da Arquitetura SBI

Nesta Seção é apresentado como situações de modificações demandadas pelos tomadores de decisão são tratadas por arquiteturas tradicionais de BI e como podem ser assimiladas na arquitetura SBI. Nas soluções atuais de BI, normalmente qualquer modificação de regra de negócio exige a reprogramação das rotinas ETL e o ajuste no modelo de dados do *data warehouse*. Esse processo de ajuste da solução de BI às novas necessidades dos usuários representa um custo adicional de até 80% por ano ao valor inicial do projeto (BUSINESSINTELLIGENCE, 2006). A cada modificação nas regras, normalmente o sistema ETL precisa ser modificado, bem como o DW e os cubos de dados relacionados. Na arquitetura SBI, o desenvolvedor pode fazer o ajuste rapidamente através da incorporação de uma nova regra de negócio na Ontologia do Domínio ou através da modificação das regras já existentes nessa ontologia. Além disso, o desenvolvedor pode acoplar serviços desenvolvidos na própria organização ou por terceiros para atender à incorporação de novas fontes de dados à solução de BI.

Visando ilustrar a flexibilidade introduzida pelos módulos da arquitetura SBI no cenário identificado, apresentam-se a seguir duas situações rotineiras de modificação de requisitos no contexto de solução de BI. Apresenta-se, de maneira breve, o impacto dessas mudanças em uma arquitetura tradicional em termos de pontos de caso de uso (JALOTE, 2002) e demonstrado como as modificações podem ser feitas na arquitetura SBI com simples ajustes nas ontologias da arquitetura.

A análise de pontos de caso de uso é aplicada neste estudo por retratar de maneira mais precisa o esforço e o custo de desenvolvimento de aplicações projetadas através de caso de uso e desenvolvidas pelo paradigma da orientação a objeto (WIKIPEDIA, 2006). Essa técnica vem sendo aplicada em substituição à tradicional análise de pontos de função em decorrência da popularização da análise e do projeto de sistemas orientados a objeto (JALOTE, 2002).

As métricas utilizadas para a análise de pontos de caso de uso de BI foram obtidas junto ao Instituto Stela (INSTITUTO STELA, 2006), instituto de pesquisa e desenvolvimento com tradição no desenvolvimento de soluções de BI para organizações de diferentes portes e segmentos de atuação. Essas métricas, conforme apresentado no Quadro 22, classificam os tipos de desenvolvimento por níveis de complexidade e associam um montante de horas necessárias para um desenvolvedor proceder com o desenvolvimento e com os testes necessários. As métricas definidas pelo Instituto Stela são caracterizadas pelo desenvolvimento das rotinas ETL através de classes Java a partir de um framework Java proprietário. Não foram encontradas na literatura outras métricas sobre projetos de soluções de BI que pudessem ser utilizadas para embasar o presente estudo.

O primeiro cenário de modificação de requisitos analisado refere-se à modificação de uma regra de negócio. Assume-se que por determinação da pró-reitoria administrativa torna-se necessário redefinir a regra de negócio utilizada para identificar instituições competidoras. No projeto inicial do *data mart*, havia sido codificada uma rotina para a carga da tabela de fatos *fato_relac_institucional* que mantém a identificação dos tipos de relacionamento entre instituições. Nessa tabela (vide Figura 38), instituições são ditas competidoras se mantêm cursos na mesma área e na mesma cidade. A solicitação de modificação no projeto desta tabela é decorrente da necessidade do pró-reitor em acompanhar com maior precisão os competidores, de acordo com as áreas de conhecimento dos cursos oferecidos pela sua instituição. Em uma arquitetura de BI tradicional, essa modificação demandaria a inclusão de uma nova tabela de dimensão (*dim_area_conhec*) que identificaria a árvore do conhecimento. Além da nova dimensão, torna-se

necessário o redesenvolvimento da rotina de carga da tabela *fato_relac_institucional* para contemplar a mudança de grão da tabela de fatos.

Quadro 22 - Métricas do Instituto Stela para o desenvolvimento de rotinas ETL para carga de tabelas de dimensões e fatos.

Tabela	Complexidade	Descrição	Horas
Dimensão	Baixa	Dimensão com uma fonte de dados	8h
Dimensão	Média	Dimensão com uma fonte de dados, modificação lenta e/ou hierárquica	20h
Dimensão	Alta	Dimensão com fontes de dados heterogêneas, modificação lenta e/ou hierárquica	32h
Fato	Baixa	Fato com uma fonte de dados e dimensões sem modificação lenta	8h
Fato	Média	Fato com uma fonte de dados até 8 dimensões, e com modificação lenta e/ou hierárquica	24h
Fato	Alta	Fato com uma fonte de dados e mais de 8 dimensões ou com fontes de dados heterogêneas	40h

Fonte: (INSTITUTO STELA, 2005)

De acordo com o Quadro 22, a rotina de carga da tabela *dim_area_conhec* é considerada um caso de uso de média complexidade por tratar-se de uma dimensão que possui estrutura hierárquica para identificação dos níveis da árvore do conhecimento. A modificação na rotina de carga da tabela de fatos é considerada de baixa complexidade por tratar-se de uma tabela ligada a dimensões sem modificação lenta (sem tratamento para mudanças no nome das áreas de conhecimento e das instituições). As duas tarefas corresponderiam, neste exemplo, a cerca de 28 horas de um desenvolvedor. Além do tempo de desenvolvimento, deve-se considerar também o tempo de recarga da tabela de fatos e de carga da dimensão *dim_area_conhec*, tempos os quais podem variar de acordo com o volume de registros e da infra-estrutura do servidor de banco de dados.

No contexto da arquitetura SBI, essa modificação poderia ser tratada unicamente com uma modificação na regra de negócio *institution-competitor* descrita na Ontologia do Domínio, conforme ilustram a Figura 38 e a Listagem 11. Essa modificação demanda poucos minutos do desenvolvedor e poderia

ser prontamente refletida nas ferramentas OLAP, sem necessidade de proceder-se a uma recarga da tabela de fatos.

Modelagem Original



Modelagem Ajustada

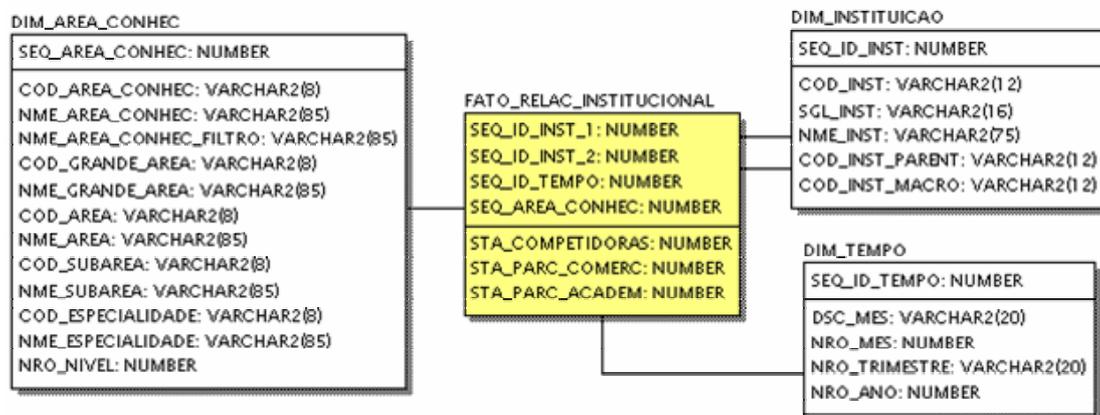


Figura 38 - Ilustração das tabelas utilizadas para a identificação das relações entre instituições antes e após a modificação nos requisitos

Listagem 11 - Ilustração da modificação na relação *institution-competitor***Regra Original:**

```
(def-relation institution-competitor (?i1 ?i2)
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (city ?c)
    (knowledge_area ?k)
    (is_institution_located_at_city ?i1 ?c)
    (has_course_institution ?co1 ?i1)
    (has_course_knowledge_area ?co1 ?k)
    (is_institution_located_at_city ?i2 ?c)
    (has_course_institution ?co2 ?i2)
    (has_course_knowledge_area ?co2 ?k)
    (not (= ?i1 ?i2))))
```

Regra Ajustada:

```
(def-relation institution-competitor (?i1 ?i2 ?k)
  :constraint (and (institution ?i1)(institution ?i2)
    (knowledge_area ?k))
  :sufficient (and
    (city ?c)
    (knowledge_area ?k)
    (is_institution_located_at_city ?i1 ?c)
    (has_course_institution ?co1 ?i1)
    (has_course_knowledge_area ?co1 ?k)
    (is_institution_located_at_city ?i2 ?c)
    (has_course_institution ?co2 ?i2)
    (has_course_knowledge_area ?co2 ?k)
    (not (= ?i1 ?i2))))
```

Uma segunda situação típica de mudança no projeto de uma solução de BI é decorrente da criação de novas regras de negócio. Assume-se que o pró-reitor deseja distinguir os docentes que atuam na instituição entre docentes que não possuem vínculo empregatício com a sua instituição (professores visitantes e horistas) e docentes com vínculo empregatício. Na abordagem das soluções de BI tradicionais, o primeiro passo para contemplar a modificação é proceder com a retirada da coluna *tot_fim_docentes* da tabela de fatos *fato_pessoa* e incluir as colunas *tot_fim_docentes_vinc* (para docentes vinculados) e *tot_fim_docentes_nvinc* (para docentes não vinculados), conforme ilustra a Figura 39. De acordo com o Quadro 22, a modificação na rotina de carga da tabela de fatos é considerada de média complexidade. Por tratar-se de uma tabela ligada a 3 dimensões, dentre as quais uma possui tratamento de modificação lenta, a dimensão *dim_pessoa* pode sofrer alteração em seus dados cadastrais, e deve-se proceder com a atualização dos fatos mensais quando da saída ou entrada de docentes, estudantes e funcionários. Essa

alteração na rotina ETL corresponderia, neste exemplo, a cerca de 24 horas de um desenvolvedor. Além do tempo de desenvolvimento, deve-se considerar também o tempo de recarga da tabela de fatos.

Essa modificação no contexto da arquitetura SBI poderia ser tratada com a incorporação de duas regras de negócio na Ontologia do Domínio, conforme ilustra a Listagem 12. A modificação pode ser implementada em poucos minutos por um desenvolvedor e refletida rapidamente nas ferramentas OLAP, sem a necessidade de proceder a uma recarga da tabela de fatos.

Modelagem Original



Modelagem Ajustada

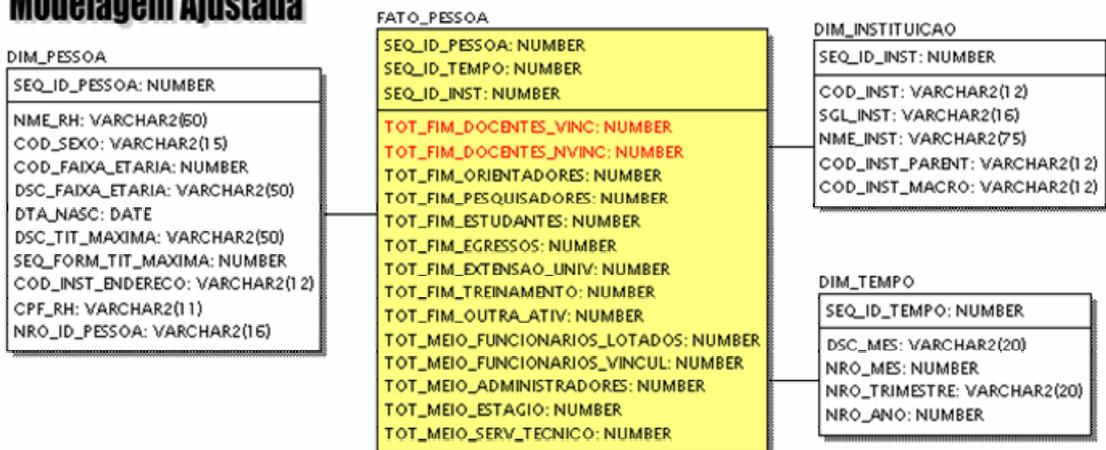


Figura 39 - Ilustração da alteração no modelo de dados por conta da nova regra de classificação de docentes

Listagem 12 - Definição em OCML das regras para classificar docentes vinculados e não vinculados a uma instituição

```
(def-relation institution-enrolled-teacher (?i ?p)
 :constraint (and (institution ?i)(person ?p))
 :sufficient
 (and
 (has_person_institutional_address ?p ?i)
 (has_activity_person ?act ?p)
 (has_activity_type ?act "ENSINO")
 (has_activity_institution ?act ?i)))

(def-relation institution-non-enrolled-teacher (?i ?p)
 :constraint (and (institution ?i)(person ?p))
 :sufficient
 (and
 (has_activity_person ?act ?p)
 (has_activity_institution ?act ?i)
 (has_activity_type ?act "ENSINO")
 (not (has_person_institutional_address ?p ?i))))
```

7.4 Análise sobre o desempenho das inferências no protótipo utilizando o mecanismo de inferência OCML

O mecanismo de inferência do framework OCML (MOTTA, 1999) foi o primeiro a ser acoplado na arquitetura SBI para fins de demonstrar a viabilidade da arquitetura. Esse mecanismo de inferência possui suporte para interpretação de funções, interpretação de controle e um mecanismo de prova. O mecanismo de prova suporta checagem de restrições, inferência de valores e processamento de heranças. Entretanto, por tratar-se de um formalismo expressivo e com um mecanismo de inferência poderoso, OCML apresenta um desempenho inferior quando comparado a outros formalismos. Quanto mais expressivo for o formalismo, pior será a velocidade de processamento das inferências (GÓMEZ-PÉREZ et al., 2004).

É importante salientar que não existem dependências entre os módulos da Arquitetura SBI e o framework OCML, qualquer mecanismo de inferência pode ser acoplado a arquitetura, bastando para tanto que a classe *OntologyManager* (Seção 6.2) seja estendida para integrar o mecanismo desejado à arquitetura SBI.

Para ilustrar o desempenho no contexto do estudo de caso, foram criadas quatro bases de conhecimento com tamanhos diferentes e quatro *data marts* com volume de dados correspondentes ao número de instâncias de cada

base de conhecimento. Para cada base de conhecimento, foram criadas instâncias para a identificação de pesquisadores e para suas atividades profissionais, além da descrição de cursos e instituições de ensino e pesquisa ligadas a essas atividades. A primeira base de conhecimento foi gerada a partir de dados reais, enquanto as demais foram criadas a partir de dados fictícios. O tamanho das bases de conhecimento é apresentado no Quadro 23. Para as estimativas de tempo apresentadas a seguir, foram utilizados os servidores descritos no Quadro 20.

Quadro 23 - Dimensionamento das bases de conhecimento utilizadas nos testes de performance sobre o framework OCML

Base de Conhecimento	Composição	Total de Instâncias
1	11.129 currículos de 55 instituições	50.319
2	59.989 currículos de 2.203 instituições	157.300
3	100.991 currículos de 15.424 instituições	288.291
4	267.716 currículos de 23.100 instituições	605.016

Para cada base de conhecimento, foram mensurados os seguintes tempos:

1. carga da base de conhecimento: tempo necessário para o mecanismo de inferência ler todas as definições dos construtores, carregar todas as instâncias e apurar todas as relações;
2. tempo para processamento da relação *institution-competitor-city*: para o processamento dessa relação, são consideradas todas as instâncias das classes *institution* e *city*. A relação *institution-competitor-city* é apresentada na Listagem 13;
3. tempo para processamento da relação *institution-competitor*: para o processamento dessa relação, são consideradas todas as instâncias das classes *institution*, *knowledge-area* e *course*. A relação *institution-competitor* é descrita na Listagem 1;
4. tempo para processamento da relação *institution-enrolled-teacher*: para o processamento dessa relação, são consideradas

todas as instâncias das classes *person*, *activity* e *institution*. A relação *institution-enrolled-teacher* é descrita na Listagem 12.

Listagem 13 - Definição em OCML da relação *institution-competitor-city*

```
(def-relation institution-competitor-city (?i1 ?i2)
  "It is sufficient that both institutions are in the same city"
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (city ?c)
    (is_institution_located_at_city ?i1 ?c)
    (is_institution_located_at_city ?i2 ?c)
    (not (= ?i1 ?i2))))
```

O tempo de carga das quatro bases de conhecimento é ilustrado na Figura 40. Analisando-se o tempo de processamento da relação *institution-competitor-city*, conforme ilustra a Figura 41, verifica-se que o processamento de inferências sobre classes com menor número de instâncias foi muito semelhante. O número máximo de instâncias manipulado nessa inferência foi a de 23.100 instâncias da classe *institution* na base de conhecimento 4.

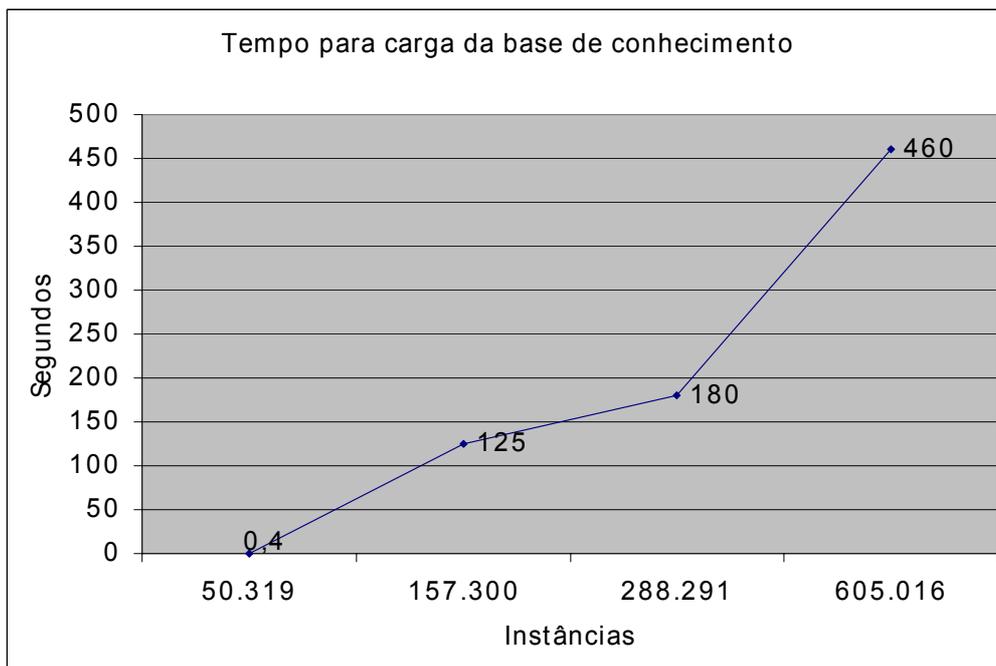


Figura 40 - Tempo para carga das instâncias

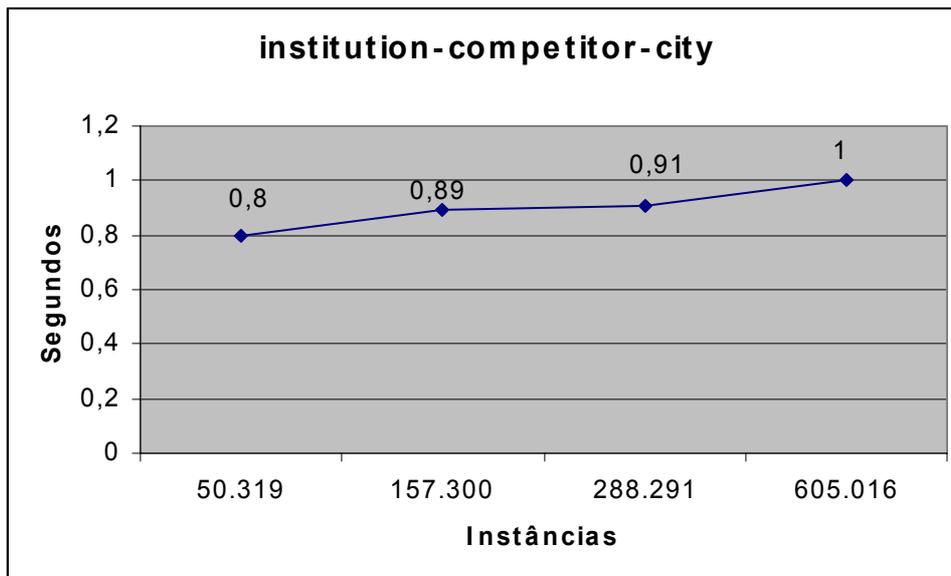


Figura 41 - Tempo para processamento da relação *institution-competitor-city*

O tempo de processamento da relação *institution-competitor* ilustrado na Figura 42 manteve-se num patamar aceitável em termos de consultas a *data marts*. O maior tempo de processamento dos testes foi verificado para a relação *institution-enrolled-teacher* sobre as 267.716 instâncias de pessoas carregadas na base de conhecimento 4. Acredita-se que o grande salto no tempo de processamento deu-se por falta de memória RAM na máquina onde foi instalado framework OCML. A quarta base de conhecimento excedeu o limite de 1 gigabyte de memória RAM dessa máquina e começou a utilizar área de swap para processar a relação, degradando o desempenho da inferência.

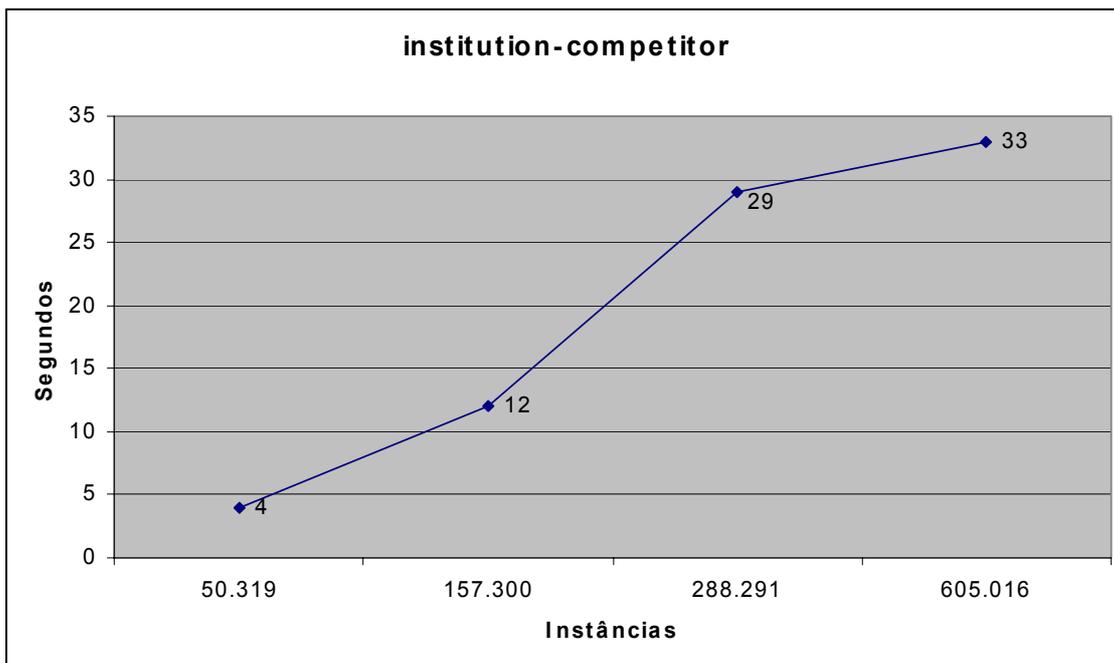


Figura 42 - Tempo para processamento da relação *institution-competitor*

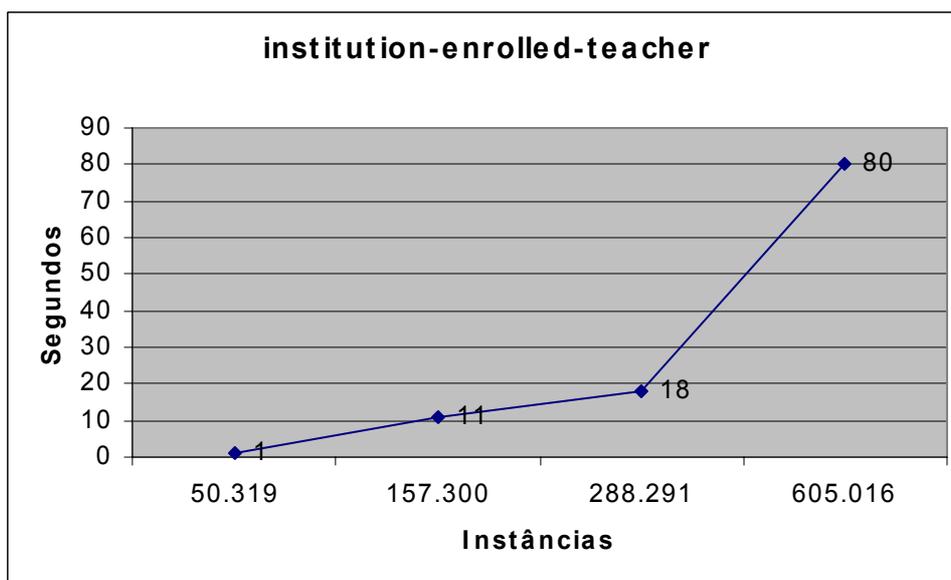


Figura 43 - Tempo para processamento da relação *institution-enrolled-teacher*

Todos os tempos de processamentos obtidos, incluindo-se o da relação *institution-enrolled-teacher* para a base de conhecimento 4, são aceitáveis no contexto de um projeto de BI. Nesses projetos, o tempo de processamento de consultas pode demorar desde milésimos de segundos a até várias horas (KIMBALL et al., 1998).

Em relação à performance e robustez de mecanismos de inferência vis-à-vis tamanho de ontologias, testes descritos em GUS et al. (2005), KIRIAKOV et al. (2005) e PAN et al. (2006) com novos frameworks para armazenamento e manipulação de ontologias reportam a utilização bem-sucedida de massas de testes com até 45 milhões de instâncias, com tempos de processamento de inferências bem inferiores aos apresentados pelo framework OCML para inferências semelhantes às utilizadas nos testes de performance dessa Seção. Além disso, fabricantes tradicionais de sistemas gerenciadores de banco de dados (SGBD) como a Oracle e a IBM já anunciaram suporte nas suas soluções para armazenamento e utilização de instâncias de ontologias (ORACLE; IBM, 2006). A tendência é que com a integração de recursos para armazenamento de ontologias e suporte a inferências nas soluções de SGBDs tradicionais, não haverá limitações para a utilização de ontologias no suporte de soluções de BI.

7.5 Comparação entre a arquitetura SBI, iniciativas acadêmicas relacionadas e soluções comerciais

Nesta Seção, é apresentado um quadro comparativo que resume as diferenças entre a arquitetura SBI e outras iniciativas relacionadas. No Quadro 24 é descrito como diversas características relacionadas ao processamento analítico são tratadas pela academia (*i.e.*, pesquisas descritas na Seção 3.6) e pela indústria (*i.e.*, produtos discutidos na Seção 2.8).

Os itens incluídos no quadro a seguir descrevem a maneira como a arquitetura proposta e as abordagens relacionadas tratam os aspectos relacionados à confecção de análises, à estratégia de execução e extensão de funcionalidades analíticas, além do desempenho e da complexidade para administração do ambiente. A descrição das características da arquitetura SBI considera os resultados alcançados através do protótipo OntoDSS.

Quadro 24 - Comparação entre as funcionalidades da arquitetura SBI e iniciativas correlatas

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
Forma de definição de cubos	Definição de consultas através da navegação sobre uma ontologia de domínio, simplificando a compreensão dos dados disponíveis. Possibilidade de navegação sobre as relações entre os conceitos para a seleção dos elementos da consulta. A definição do conteúdo é feita em uma seqüência de etapas, em que cada etapa corresponde a uma seção do cubo.	Definição de consultas através da navegação sobre uma ontologia de domínio. A consulta é formulada em duas etapas: a) seleção dos conceitos/restrições da consulta; e b) definição do formato da consulta. À medida que o usuário seleciona os conceitos e filtros, a consulta é representada em linguagem natural e em um grafo, facilitando a compreensão do escopo.	Cubos são definidos por ferramentas de terceiros integradas à arquitetura. Não há suporte para navegação sobre os dados através de ontologia do domínio.	Cubos são definidos na maneira tradicional, sem suporte para navegação sobre os dados através de ontologia do domínio.	Navegação sobre os metadados do fabricante ou sobre a definição técnica das tabelas no DW. A representação dos metadados reflete a forma com que o dado está organizado e não a maneira como está inserido no domínio do usuário.
Suporte à redefinição de consultas	Usuário guia o processo selecionando filtros semânticos, relações ou serviços, os quais são recomendados pela ferramenta.	Usuários podem navegar sobre as relações existentes entre os conceitos utilizados na consulta. Não há suporte para uso de regras ou expressões lógicas.	Não suportam.	Não é mencionado suporte para esta característica.	O suporte à redefinição de consultas é restrito à utilização de agregados para melhoria de performance e não para expansão ou filtragem da consulta.
Recomendações durante as análises	Inferência automática de serviços, relações, filtros semânticos e análises que possuem conceitos em comum	Possui suporte para a recomendação de textos associados à definição da análise.	Não suportam.	Possui suporte para a recomendação de textos associados à definição da análise.	Não suportam.

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
	com a definição da análise.				
Suporte a <i>drill down</i> e <i>up</i>	Suporta <i>drill</i> utilizando relações transitivas definidas sobre os conceitos da Ontologia do Domínio.	Suporta <i>drill</i> utilizando relações transitivas definidas sobre os conceitos da ontologia que suporta a arquitetura.	Feita de maneira tradicional, apoiada sobre a definição sintática de dimensões e fatos no banco de dados.	Feita de maneira tradicional, apoiada sobre a definição sintática de dimensões e fatos no banco de dados.	Utilização de metadados proprietários para a definição de hierarquias, sem, no entanto, a definição da semântica dessa hierarquia.
Suporte a <i>drill across</i> e <i>through</i>	Suporta <i>drill</i> utilizando relações simétricas e associativas definidas sobre os conceitos.	Utiliza relações associativas definidas entre os conceitos.	Restrito às relações de chave estrangeiras entre as tabelas do DW.	Restrito às relações de chave estrangeiras entre as tabelas do DW.	Restrito às relações de chave estrangeiras entre as tabelas do DW.
Personalização da apresentação	Cada perfil de usuário pode ter uma nomenclatura diferenciada para o mesmo conceito.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Nenhum recurso adicional é agregado à ferramenta OLAP.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Suportado em algumas ferramentas através da tradução do modelo físico em um modelo lógico. No entanto, a mesma definição lógica é apresentada para todos os usuários.
Personalização das funcionalidades exploratórias	Permite a definição das funcionalidades exploratórias que serão disponibilizadas para cada perfil de usuário.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.
Suporte à modificação da lógica dos <i>drills</i>	Regras de negócio podem ser incorporadas ou alteradas na ontologia do domínio a qualquer	Esta característica não é descrita nas referências consultadas sobre este projeto.	Nenhum recurso adicional é agregado à ferramenta OLAP de terceiros.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Suporte através de APIs proprietárias.

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
	momento e serem utilizadas pelo usuário através de filtros semânticos em uma análise.				
Integração de serviços no contexto analítico	Permite a agregação de qualquer serviço através de WSS. Serviços são descobertos e oferecidos ao usuário automaticamente.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Suporte através de APIs proprietárias.
Composição de serviços	Permite a composição de WSS, incluindo condições e mediadores.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.
Utilização de regras de negócio	Suporta a representação de regras e expressões lógicas envolvendo conceitos da Ontologia do Domínio. Estas definições suportam filtros semânticos no momento da análise.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.	Esta característica não é descrita nas referências consultadas sobre este projeto.	Não suportam.
Utilização de dados não estruturados	Não possui nenhum suporte específico para esta característica no momento. No entanto, com a importação do	Prevê suporte para classificação de textos de acordo com os conceitos da ontologia que suporta a arquitetura	Prevê a integração com um sistema gerenciador de documentos para suportar a localização de documentos.	Menciona que estão em desenvolvimento recursos para a extração de dimensões e fatos a partir de um <i>corpus</i> .	Prevêem suporte para a importação de textos através de ferramentas ETL e localização desses textos através de

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
	metadado ou do próprio texto para o DW através de ferramentas ETL, o usuário poderia localizar os textos durante a manipulação de cubos.	para a localização e a recomendação desses textos.		Possui recurso para relacionar uma dimensão a um conjunto de documentos através da extração semi-automática de taxonomias e outras características de documentos ou de seus metadados.	ponteiros dos textos durante a manipulação de cubos.
Integração de dados externos	Dados externos podem ser trazidos através de WSS. Não há suporte para a geração de consultas em bases distribuídas.	Suporte para a integração de bases distribuídas através de nós de informações controlados por agentes e por estruturas de mediação baseadas em ontologias.	Metadados de cubos e de documentos são mapeados através da <i>ontologia global</i> da arquitetura. Desse modo, usuários podem localizar cubos e documentos relacionados a um dado conceito.	Não é mencionado o suporte para esta característica.	Integração de dados feita através de ferramentas ETL.
Flexibilidade para modificações	Permite a qualquer momento modificação nas regras de negócio diretamente sobre as ontologias, sem necessidades de proceder a alterações no modelo de dados e nas rotinas ETL. Permite ainda a incorporação de novas fontes de dados através de WSS	Permite a incorporação de novas fontes de dados através da descrição de novos nós de informações.	Nenhuma contribuição em relação às soluções tradicionais é mencionada.	Nenhuma contribuição em relação às soluções tradicionais é mencionada.	Necessariamente, deve-se proceder com ajustes nas rotinas ETL e no modelo de dados quando da modificação de uma regra de negócio ou da incorporação de novas fontes de dados.

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
	desenvolvidos <i>in-house</i> ou externamente.				
Performance das consultas	As consultas são processadas diretamente sobre o SGBD, sem penalização alguma para o desempenho das consultas. Inferências são processadas sobre o resultado de uma consulta, sobre quantidades limitadas de dados.	O processamento das consultas é distribuído nos nós de informação. Não foi possível identificar informações a respeito do desempenho nas referências deste projeto.	O desempenho da ferramenta OLAP integrada à arquitetura é mantida, dado que nenhuma inferência adicional é suportada pela arquitetura.	Nenhuma inferência adicional sobre o processamento de consultas é descrita. Portanto, o desempenho não é afetada.	Nenhuma inferência adicional sobre o processamento de consultas é suportada. Possuem esquemas de otimização de consultas.
Complexidade de atualização de metadados		A atualização dos mapeamentos é feita de maneira distribuída. Cada nó conta com uma estrutura para a indexação dos dados relacionados à ontologia do sistema.	A manutenção do metadados não é tratada por nenhum componente da arquitetura. Os autores sugerem que isso seja feito através de uma ferramenta ETL.	Um conjunto de aplicações integradas à arquitetura suporta a criação e manutenção semi-automática dos metadados que relacionam documentos e cubos.	Nenhuma exigência adicional é imposta ao administrador, somente a definição das tabelas e a criação da nomenclatura para o modelo lógico.
Complexidade de Administração	Necessidade de configuração inicial e manutenção das ontologias de domínio, BI e de serviços, além da necessidade de manter a infra-estrutura de WSS.	O projeto consiste em várias ferramentas (OLAP, ontologias, estrutura de agentes, algoritmos de <i>data mining</i> etc.). Essas ferramentas foram desenvolvidas por	Necessidade de manter a ontologia global. Os autores sugerem que essa manutenção seja feita por uma ferramenta ETL.	O projeto reúne ferramentas OLAP, para extração de características e para classificação de documentos. Mencionada o uso de uma ferramenta ETL.	Pouca complexidade. Todas contam com ferramentas integradas para a gestão dos metadados proprietários. A integração entre os módulos dessas soluções fica

Característica	SBI e OntoDSS	SEWASIE	Priebe e Pernul	BIKM	Soluções de BI Tradicionais
		<p>grupos participantes do consórcio e por terceiros. Apesar da aparente complexidade, não foram verificadas informações sobre o ônus dessa estrutura para os administradores.</p>		<p>Não é descrito o grau de complexidade para manter a integração das ferramentas.</p>	<p>transparente para o usuário final e para o administrador do ambiente.</p>

Verifica-se que as abordagens propostas por Bergamaschi et al. (2004, 2005), Cody et al. (2002) e por Priebe e Pernul (2003) introduzem características complementares às desenvolvidas no contexto desta pesquisa, conforme descrito no Quadro 24 e na Seção 3.6. O objetivo da aplicação de tecnologias semânticas nessas arquiteturas visa, principalmente, possibilitar a inserção e a recomendação de textos no processamento analítico.

Tecnologias da *Web semântica* vêm sendo aplicadas de diferentes maneiras para resolver desafios tradicionais relacionados a sistemas de informação, mas pouca ênfase é dada às ferramentas analíticas. Ontologias são utilizadas por sistemas como Observer (MENA et al., 2000) e Tambis (PATON et al., 1999) para mapear fontes de dados e suportar a tradução de consultas em subconsultas, que são então processadas nas fontes de dados distribuídas. Ontologias vêm sendo aplicadas ainda para suportar a redefinição de consultas utilizando sinônimos e hipônimos para estender os resultados das consultas (GUHA; McCOOL, 2003; NECIB; FREITAG, 2003).

Na abordagem adotada na arquitetura SBI, consultas são processadas sobre fontes de dados porque consultas analíticas não são suportadas pelas linguagens atuais de consultas de ontologias. Além disso, as estratégias correntes para persistência de instâncias em ontologias estão longe de oferecer o mesmo desempenho proporcionado pelos sistemas gerenciadores de banco de dados comerciais. A abordagem utilizada prevê a replicação dos dados contidos nas dimensões de um DW na Ontologia de Domínio para suportar inferências sobre os resultados das consultas. A replicação completa dos dados do DW é inviável. As dimensões do DW contêm a maior parte dos dados que possam suportar inferências.

Uma das características não encontradas em nenhuma iniciativa acadêmica e industrial foi a da integração de WSS em aplicações analíticas. A arquitetura SBI introduz um módulo específico (*i.e.*, Gerenciador de Serviços) e uma ontologia (*i.e.*, Ontologia de Serviços) para suportar a descoberta, composição e execução de WSS de acordo com o contexto da análise. Esse módulo deverá ser suportado por um *framework* de WSS na versão atual pelo IRS-III (DOMINGUE et al., 2004b). A integração do *framework* IRS-III à arquitetura através do Gerenciador de Serviços visa contribuir para a resolução

da falta de flexibilidade nas soluções de BI atuais no que tange à extensão das funcionalidades exploratórias. Assim, o código existente na organização ou na *Web* pode ser descrito semanticamente e facilmente integrado a ferramentas analíticas, oferecendo meios para integrar novas fontes de dados à solução de BI e possibilitando a criação de operações analíticas customizadas de acordo com as necessidades dos tomadores de decisão.

Para a utilização do *framework* IRS-III, o autor desta tese teve que, em colaboração com os pesquisadores do The Knowledge Media Institute, desenvolver um módulo para a descoberta de WSS, além de um mecanismo de orquestração (HAKIMPOUR et al., 2005; SELL et al., 2004b) e de uma ferramenta de composição (SELL et al., 2004b). Esses desenvolvimentos fizeram-se necessários para suportar a descoberta e a composição de serviços, visando atender a requisitos de análise mais sofisticados no contexto do processamento analítico.

A ferramenta de composição integrada à arquitetura SBI e ao OntoDSS é um passo no caminho da composição automática de *Web services*. Essa ferramenta sugere serviços de acordo com o contexto de uma composição e suporta a definição de mediadores e operadores de controle em um processo interativo de composição. Outras ferramentas de composição de *Web services*, como CAT (KIM et al., 2004) e Mindswap Composer (SIRIN et al., 2004), também suportam composições semi-automáticas, mas não suportam mediadores nem componentes de controle de fluxo. A adoção de mediadores visa oferecer mais flexibilidade ao usuário, uma vez que é imprescindível a seleção de serviços implementados e disponibilizados por diferentes partes durante o processo de composição. Mediadores permitem ainda a integração da ferramenta de composição ao ambiente analítico na arquitetura SBI.

Não foram localizadas referências na literatura sobre a integração de ferramentas de composição de serviços a ferramentas analíticas. O resultado de uma composição pode ser usado como entrada para uma análise. Do mesmo modo, o resultado de uma análise pode ser utilizado como entrada para uma composição de serviços. Além do suporte para a definição de explorações complexas, essa integração pode permitir que usuários ajam sobre seus negócios de acordo com *insights* obtidos através da ferramenta analítica. Por

exemplo, um gerente pode definir uma análise através da composição de WSS, sendo para cada passo a seguir executado um WSS.

1. Identificar produtos cujas vendas diminuíram.
2. Checar o preço dos competidores para esses produtos.
3. Verificar a margem de lucro desses produtos.
4. Checar possibilidades de redução da margem de lucro observando os custos relacionados.
5. Opcionalmente, o gerente pode executar outro WSS para modificar a margem de lucro dos produtos, intercedendo diretamente no negócio a partir do ambiente analítico.

Composições como a descrita no exemplo anterior representam as abordagens assumidas pelos tomadores de decisão para analisar os seus negócios e agir sobre eles; e elas podem ser utilizadas para subsidiar futuras decisões e ações. Esse processo iterativo permite a captura de conhecimento tácito dos tomadores de decisão, constituindo um rico repositório de conhecimento para as organizações.

7.6 Considerações sobre o Capítulo

Neste capítulo foi ilustrado como a arquitetura SBI apoiou a criação de alternativas exploratórias inéditas no contexto do processamento analítico. Apresentou-se a ferramenta analítica OntoDSS, ferramenta desenvolvida para demonstrar o potencial exploratório que a combinação de tecnologias semânticas combinadas com tecnologia de BI introduzem no contexto analítico. Um estudo de caso foi apresentado, no qual se descreveu a configuração da arquitetura para apoiar a tomada de decisão no contexto da gestão de C&T, mais precisamente no contexto das atividades de gestão de pró-reitores de instituições de ensino e pesquisa.

O estudo de caso ilustrou que a arquitetura SBI é uma alternativa viável para a construção de soluções de BI flexíveis e diferenciadas. Ilustrou-se como a arquitetura SBI tornou possível, sem codificação, apenas com a configuração das regras de negócio nas ontologias da arquitetura, a personalização da solução de acordo com as necessidades analíticas do tomador de decisão

dentro do cenário do estudo de caso. A aplicação OntoDSS demonstrou ainda os recursos analíticos inéditos introduzidos pela arquitetura SBI no contexto de BI. Resumidamente, as seguintes características foram possibilitadas pela arquitetura e ilustradas pela aplicação OntoDSS:

- a informação é apresentada aos usuários utilizando seus próprios vocabulários, o que pode facilitar a compreensão do significado da informação;
- funcionalidades exploratórias inéditas combinando processamento analítico e inferências são oferecidas aos usuários;
- WSS são aplicados para possibilitar extensões flexíveis das funcionalidades exploratórias e a incorporação de novas fontes de dados da organização ou externas sem necessidade de desenvolvimento de rotinas ETL específicas;
- as definições do conhecimento e das regras de negócio podem ser alteradas a qualquer momento, sem necessidade de mudar o modelo de dados e de alteração nas rotinas ETL, proporcionando maior flexibilidade para mudanças de requisitos analíticos;
- as descrições semânticas dos serviços, das análises e dos dados são utilizadas para orientar a recomendação automática de recursos relevantes, visando guiar os usuários em um processo interativo de tomada de decisão;
- uma ferramenta de composição é integrada ao ambiente analítico. Essa ferramenta visa possibilitar a definição de análises complexas e suportar ações sobre o negócio motivadas pelos *insights* obtidos nas análises;
- composições feitas pelos usuários constituem um repositório de conhecimento tácito que descreve a abordagem dos tomadores de decisão para analisar seus negócios e para agir de acordo com os *insights* obtidos. Esse repositório de conhecimento pode apoiar futuras decisões ou prover os meios para automatizar ações sobre o negócio através de agentes.

Um quadro comparativo foi apresentado, resumindo como aspectos relacionados à flexibilidade, inferência e personalização são abordados pela arquitetura SBI e por outras iniciativas acadêmicas e industriais. De maneira geral, várias características introduzidas pela arquitetura SBI não são suportadas pelas iniciativas relacionadas, como o uso de regras de negócio para o suporte de *drill* e *slice*, além da utilização (e composição) de WSS no contexto de aplicações analíticas. As características inéditas introduzidas pela arquitetura SBI lançam luz ao poder que técnicas pesquisadas na área da engenharia do conhecimento podem agregar às soluções de BI e resolver sérias limitações como as apresentadas na problemática da pesquisa.

8 CONSIDERAÇÕES FINAIS

A arquitetura SBI incorpora várias características que a distinguem das soluções de BI existentes e das pesquisas relacionadas. Esta pesquisa buscou a criação de uma solução abrangente para integrar a semântica do negócio e as fontes de dados e serviços, visando suportar o processamento analítico de apoio à decisão e contribuir para a resolução de deficiências nas soluções atuais de BI.

Esta pesquisa aponta novas maneiras para a resolução de problemas clássicos no contexto de soluções de BI. Tecnologias semânticas são aplicadas em domínios variados, mas a aplicação destas no contexto de soluções de BI está restrita quase que exclusivamente ao suporte de integração de dados. Conforme descrito anteriormente, são verificadas três referências de aplicação de ontologias em aplicações analíticas. Entretanto, essas pesquisas visam basicamente suportar a recomendação de documentos relacionados a um cubo ou a anotação de cubos para a recuperação desses em sistemas de buscas integrados. Não existem referências na literatura nos moldes da solução proposta.

Para a resolução da problemática desta pesquisa criou-se uma arquitetura baseada em um conjunto de módulos funcionais implementados em Java e suportados por três ontologias. A concepção da arquitetura foi orientada por um conjunto de requisitos funcionais e não funcionais identificados no Capítulo 4. No Capítulo 5 foi apresentado como as três ontologias da arquitetura foram concebidas para apoiar o cumprimento dos requisitos, e no Capítulo 6 descreveu-se como os módulos da arquitetura foram projetados visando também ao atendimento dos requisitos da arquitetura. Recomenda-se a integração dos componentes da arquitetura a repositórios de dados organizados e otimizados para o processamento analítico.

A primeira problemática da pesquisa refere-se a como integrar a representação da semântica do negócio às soluções de BI e utilizar esse conhecimento no apoio ao processamento analítico. Foi apresentado como o conhecimento sobre o negócio é representado nas ontologias da arquitetura e utilizado para guiar o processamento analítico através de operações de *drill* e

slice e para oferecer ao usuário um acesso transparente a recursos como informações e serviços através da terminologia do negócio. O conhecimento do negócio na arquitetura SBI passa a ser explicitado em bases de conhecimento e não apenas interpretado e transformado em rotinas ETL pelos desenvolvedores. Assim, como ilustrado no capítulo anterior, torna-se mais simples a modificação nas regras do negócio, bastando ajustar a ontologia para mudar o comportamento do processamento analítico e da forma de apresentação dos recursos necessários para apoiar o processamento analítico.

Essa mudança de paradigma de construção de soluções de BI visa apoiar a construção de soluções de BI mais flexíveis e que propiciem aos tomadores de decisão novos recursos analíticos centrados na terminologia e nas regras do negócio dos usuários. O foco até então na tecnologia passa a ser no próprio negócio, nos seus conceitos, nas regras e nas necessidades analíticas dos tomadores de decisão. Essa quebra de paradigma introduz novas etapas nas metodologias de desenvolvimento de BI. Essas novas etapas, por tratarem de aspectos não abordados até então pelas metodologias atuais (e.g., representação do conhecimento do negócio e o seu mapeamento aos dados e serviços), exigem da equipe de desenvolvimento o desempenho de dois novos papéis, o de Engenheiro do Conhecimento e o de Arquiteto de Serviços. Conforme descrito no Capítulo 4, os papéis introduzidos exigem dos membros da equipe novas competências. Até então, nos projetos de soluções de BI, o conhecimento era obtido pelo analista de negócios e repassado para a equipe de desenvolvimento, que por sua vez concebia o modelo de dados e as ferramentas ETL e de apresentação de dados. Esse processo muda com a arquitetura SBI. O conhecimento sobre o negócio, cerne da nova proposta, deixa de ser somente interpretado pelo analista de negócio, passando a ser efetivamente representado e integrado aos módulos da solução de BI.

A segunda problemática refere-se a como prover aos desenvolvedores a possibilidade de customização das funcionalidades exploratórias nas aplicações analíticas e, aos usuários, a possibilidade de receber recomendações dessas funcionalidades de acordo com o contexto de suas análises. Além do recurso de customização das operações de *drill* e *slice* através das regras de negócio definidas na Ontologia do Domínio, a arquitetura

SBI oferece a possibilidade de acoplamento dos serviços disponíveis na organização ou na Web. Essa característica permite o reaproveitamento de código e a liberdade necessária para o desenvolvimento de rotinas customizadas para atender às demandas analíticas dos tomadores de decisão. As ontologias da arquitetura são aplicadas na descrição semântica dos serviços e das informações disponíveis para facilitar a sua localização e recomendação aos tomadores de decisão de acordo com o contexto das suas análises. Através desse recurso de recomendação, torna-se possível criar ferramentas analíticas proativas, que antecipam as demandas analíticas dos usuários e oferecem os recursos disponíveis para guiar o processo analítico.

Técnicas de recomendação são aplicadas em larga escala e com grande sucesso principalmente em sites de comércio eletrônico e em sistemas de busca (vide, por exemplo, www.amazon.com, www.google.com e www.buscape.com.br) (BARBOSA et. al., 2003; KIMBALL; MERZ, 2000). A efetividade desse recurso no contexto do processo decisório não foi mensurada nessa pesquisa por entender-se que a análise de efetividade estaria muito mais focada em questões ergonômicas, em como apresentar as recomendações de acordo com o perfil do tomador de decisão. As recomendações poderiam ser realizadas por avatares, por painel de recomendações ou como implementado na ferramenta OntoDSS (i.e. através de cliques de mouse sobre o assunto desejado). A decisão sobre como apresentar esse recurso ao tomador de decisão não é o foco dessa tese; o foco está em como possibilitar o processamento da inferência de recursos relacionados ao contexto da análise.

Os mecanismos de inferência e repositórios de ontologia podem limitar a aplicação da arquitetura em projetos de *data warehouse* envolvendo dimensões grandes, que correspondam a milhares de instâncias na ontologia do domínio. No mecanismo de inferência OCML utilizado nos estudos de viabilidade, chegou-se a utilizar até 605.106 instâncias. Entretanto, o mecanismo utilizado limita o armazenamento de instâncias à memória RAM disponível no computador. Testes com outros mecanismos de inferências reportam a utilização de massas de testes com até 45 milhões de instâncias (vide GUS et al. (2005), KIRIAKOV et al. (2005) e PAN et al. (2006) para maiores detalhes). Entretanto, fabricantes tradicionais de sistemas

gerenciadores de banco de dados (SGBD) como a Oracle e a IBM já anunciaram suporte nas suas soluções para armazenamento e utilização de ontologias. A tendência é que com a integração de recursos para armazenamento de ontologias e suporte a inferências nas soluções de SGBDs tradicionais, não haverá limitações para a utilização de ontologias no suporte de soluções de BI de grande porte.

A aplicação da arquitetura a repositórios de sistemas transacionais pode apresentar tempos de resposta inaceitáveis. Para maiores detalhes sobre os benefícios da aplicação dos preceitos de projetos de business intelligence para a organização de repositório de informações de aplicações analíticas, vide Kimball et. al. (1998).

Examinando-se a literatura correlata e os produtos comerciais, verifica-se que existe um nicho a ser explorado por uma nova linha de pesquisa, a do desenvolvimento de soluções de BI baseadas em conhecimento do negócio para o suporte do processamento analítico. As iniciativas acadêmicas para representação e inferência semântica nessa categoria de aplicação estão, em sua maioria, restritas ao processo de integração de dados. Esta tese inaugura essa nova linha de pesquisa lançando luz a técnicas pesquisadas no contexto da Engenharia do Conhecimento para a criação de soluções de BI flexíveis e diferenciadas. O cerne dessas novas soluções de BI passa a ser o conhecimento do negócio e o foco passa a ser na sua aplicação em prol de maior poder de customização e flexibilidade para acompanhamento das mudanças no negócio das organizações e no atendimento das necessidades dos tomadores de decisão.

8.1 Trabalhos futuros

O foco da análise de viabilidade desta pesquisa foi na demonstração das novas possibilidades analíticas e na flexibilidade que a arquitetura introduz a soluções analíticas. Este estudo será estendido para analisar o impacto no processo decisório em decorrência das características inéditas introduzidas pela arquitetura. Este estudo deverá ser realizado em organizações de diferentes portes e setores econômicos, levando em consideração questões

ergonômicas para disponibilizar esses novos recursos de maneira efetiva ao tomador de decisão.

Um estudo mais abrangente deverá ser conduzido para analisar as dificuldades introduzidas pela arquitetura para os desenvolvedores. O estudo de caso ilustrou a agilidade introduzida para acompanhar mudanças de requisitos. Torna-se necessário, entretanto, verificar como os desenvolvedores reagirão à necessidade de desenvolvimento de novas competências, identificando possíveis resistências e maneiras de capacitá-los para exercer esses novos papéis.

O estudo de viabilidade da arquitetura deverá ser estendido também para analisar como oferecer maior velocidade no processamento de inferências quando da utilização de regras de negócio em operações de *drill* e *slice*. Em relação à performance do processamento de inferências sobre *data marts* com grandes volumes de dados, iniciou-se um estudo sobre a utilização das regras de negócio da Ontologia do Domínio para orientar o processamento ETL das regras mais freqüentemente utilizadas pelos tomadores de decisão em *drill* e *slice*. Assim, as inferências mais utilizadas poderão ser processadas em *batch* e o resultado poderá ser mantido em novas dimensões e fatos no *data warehouse*, oferecendo maior desempenho aos usuários das ferramentas analíticas, mas mantendo a flexibilidade de mudanças das regras diretamente sobre a ontologia. Serão efetuados ainda testes com outros mecanismos de inferência e repositórios de ontologia mais robustos.

No contexto da recomendação de recursos, deverão ser investigadas maneiras de se identificar, além do contexto das análises, o perfil do tomador de decisão. Essas informações poderão tornar o processo de recomendação mais eficaz, levando os tomadores de decisão a encontrar, de maneira mais rápida, o que eles procuram e receber recomendações de informações e outros recursos tão ou mais importantes do que eles procuravam. Outras informações sobre legislação, sobre os processos da organização e sobre o mercado deverão ser analisadas como informação de contexto a ser incorporada no contexto das ontologias da arquitetura para subsidiar um processo de recomendação de recursos mais eficaz.

O autor, em conjunto com outros pesquisadores do Instituto Stela, já iniciou um estudo sobre a integração de dados não estruturados na arquitetura SBI. Essa pesquisa vem orientando o desenvolvimento de novas ferramentas de localização de informação e de apoio ao processo decisório em áreas como memória organizacional, comunidades de prática e aprendizagem organizacional. Os resultados mais recentes desses desenvolvimentos são apresentados em NAPOLI et al. (2006).

REFERÊNCIAS BIBLIOGRÁFICAS

AKAHANI J.; HIRAMATSU, K.; KOGURE, K. Coordinating Heterogeneous Information Services based On Approximate Ontology Translation. In: AAMAS-2002 WORKSHOP ON AGENTCITIES: CHALLENGES IN OPEN AGENT SYSTEMS.

ALENQUER, P. L. Regras de negócio para análise em ambientes OLAP. 2002. Dissertação (Mestrado em Informática) – Instituto de Matemática - IM, Núcleo de Computação - NCE, Universidade Federal do Rio de Janeiro, 2002.

ALESSO, H. P.; SMITH, C. F. **Developing Semantic Web services**. Massachusetts: A. K. Peters, 2004.

ALMEIDA, S. et al. Getting Started with DataWarehouse and Business Intelligence. **IBM Corporation**. 1999. Disponível em: <www.redbooks.ibm.com>. Acesso em: 29 dez. 1999.

ANTONIOU, G.; HARMELEN, F. V. **A Semantic Web Primer**. USA: MIT Press, 2004.

ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA DE PRODUÇÃO (ABEPRO). Engenharia de Produção. Disponível em: <http://www.abepro.org.br/saiba_mais_1.htm>. Acesso em: 1 fev. 2005.

BARBOSA, D. M. et al. Uma metodologia para desenvolvimento de Data Warehouse voltado ao desenvolvimento de Data Web House para o monitoramento de portais corporativos. In: KNOWLEDGE MANAGEMENT BRASIL 2002, São Paulo.

BARBOSA, D. M. et al. Aplicação de Data Webhousing para monitoramento de acessos a sites Web de grupos de pesquisa e desenvolvimento: um estudo de caso. In: KNOWLEDGE MANAGEMENT BRASIL, 2003, São Paulo.

BECHHOFFER, S.; GOBLE, C.; HORROCKS, I. DAML+OIL is not enough. In: THE FIRST SEMANTIC WEB WORKING SYMPOSIUM (SWWS'01), 2001. Disponível em: <<http://ceur-ws.org/>>. Acesso em: 15 abr. 2004.

BERGAMASCHI, S.; QUIX, C.; JARKE, M. The SEWASIE EU IST Project. **SIG SEMIS Bulletin**, v. 2, n. 1, fev. 2005.

BERGAMASCHI, S.; FILLOTTRANI, P.; GELATI, G. The SEWASIE multi-agent system. In: THIRD INTERNATIONAL WORKSHOP ON AGENTS AND PEER-TO-PEER COMPUTING (AP2PC 2004), July 2004, New York City, USA.

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

BERRY, M. J. A.; LINOFF, G. **Data Mining techniques**. New York: John Wiley & Sons Inc., 1997.

BERSON, A. **Data Warehousing, Data Mining & OLAP**. USA: McGraw-Hill, 1997.

BORGO, S. et al. **Ontology RoadMap**. WonderWeb Deliverable D15, Dec 2002. Disponível em: <<http://wonderweb.semanticweb.org>>. Acesso em: 20 fev. 2004.

BRACHMAN, R. J.; LEVESQUE, H. J. **Readings in knowledge representation**. Los Altos, Calif: Morgan Kaufmann, 1985.

_____. **Knowledge Representation and Reasoning**. San Francisco, USA: Morgan Kaufmann, 2004.

BRACKETT, M. H. **The Data Warehouse challenge**. Taming Data Chaos. New York: John Wiley & Sons Inc., 1996.

BROHMAN, M, K. et al. The Business Intelligence value chain: Data-Driven decision support in a Data Warehouse environment: an exploratory study. In: THE 37th HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, IEEE, 2000, Hawaii. ISNN0-7695-0493-0/00.

BRUCKNER, R. M. et al. A framework for a multidimensional OLAP model using topic maps. **WISE**, (2), p. 109-118, 2001.

BUSINESSINTELLIGENCE.COM. **Forrester Research Finds Enterprise IT Spending Intentions Show Modest Growth For 2005**. Disponível em: <<http://www.businessintelligence.com/ex/asp/id.850/xe/binewsdetail.htm>>. Acesso em: 23 jul. 2005.

_____. **Not so Real: Real-Time Business Intelligence**. Disponível em: <<http://www.businessintelligence.com/ex/asp/code.101/xe/article.htm>>. Acesso em: 4 jan. 2006.

CA MAGAZINE. **Survey of Business Intelligence systems**. Disponível em: <www.camagazine.com/index.cfm/ci_id/21747/la_id/1.htm>. Acesso em: 10 nov. 2004.

_____. **Second survey of Business Intelligence systems**. Disponível em: <http://www.camagazine.com/index.cfm/ci_id/26573/la_id/1.htm>. Acesso em: 13 maio 2005.

CABRAL, L. et al. Approaches to Semantic Web Services: An Overview and Comparisons. In: THE FIRST EUROPEAN SEMANTIC WEB SYMPOSIUM (ESWS2004), May 2004, Heraklion, Crete, Greece.

CASTOLDI, A. V. **Uma ontologia para enlaces de unidades de informação em plataformas de governo eletrônico**. 2003. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2003.

CATARCI, T. et al. An ontology based visual tool for query formulation support. In: THE 16th EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI-04), August 2004, Valencia, Spain.

CHANDRASEKARAN, B.; JOSEPHSON, J. R.; BENJAMINS, V. R. Ontologies: what are they? why do we need them? **IEEE Intelligent Systems and Their Applications**, 14(1), p. 20-26, 1999. (Special Issue on Ontologies).

CHANG, D .T. **CWM enablement showcase**: Warehouse Metadata interchange made easy using CWM. Disponível em: <<http://www-4.ibm.com/software/data/pubs/papers/cwm/cwm.pdf>>. Acesso em: 22 nov. 2001.

CHUCK, B. et al. **Data modeling techniques for Data Warehousing**. Fev. 1998. Disponível em: <<http://www.redbooks.ibm.com/redbooks.nsf/0/d5f8e9f4b913822d8525659d002a58f4?OpenDocument>>. Acesso em: 12 dez. 1998.

CNPq. **Plataforma Lattes**: Diretório de Grupos de Pesquisa. Disponível em: <<http://www.cnpq.br/plataformalattes/dgp/index.htm>>. Acesso em: 5 jan. 2005.

CODD, E. F.; CODD, S. B.; SALLEY, C. T. **Providing OLAP (on-line analytical processing) to user-analysts**: An IT mandate. 1995. Disponível em: <http://dev.hyperion.com/resource_library/white_papers/providing_olap_to_user_analysts.pdf>. Acesso em: 14 nov. 2005.

CODY, W. F. et al. The integration of business intelligence and knowledge management. **IBM Systems Journal**. v. 41, issue 4, p. 697-713, 2002. Disponível em: <<http://www.research.ibm.com/journal/sj/414/cody.pdf>>. Acesso em: 10 fev. 2004.

COMPUTERWORLD. **AlphaBlox deal anchors IBM's BI strategy**. Disponível em: <<http://www.computerworld.com.au/index.php/id;807272035;fp;16;fpid;0>>. Acesso em: 11 jan. 2005.

_____. **The Top 10 Critical Challenges for Business Intelligence Success**. Disponível em: <<http://www.computerworld.com/services/whitepapers/story/0,4793,82630,00.html>>. Acesso em: 10 out. 2004.

CONFALONIERI, R.; DOMINGUE, J.; MOTTA, E. Orchestration of Semantic Web services in IRS-III. In: THE FIRST AKT WORKSHOP ON SEMANTIC WEB SERVICES (AKT-SWS04), December 2004, KMi, The Open University, Milton Keynes, UK.

CONSELHO NACIONAL DE PESQUISA CIENTÍFICA. **Histórico da Plataforma Lattes**. Disponível em: <<http://lattes.cnpq.br/historico.jsp>>. Acesso em: 24 jul. 2003.

COREY, M. J.; ABBEY, M.; IAN, A. **Oracle 8 Data Warehousing**. Berkeley: Osborne/McGraw-Hill, 1998.

CÔRTEZ, S. **BI, Data Warehouse e Data Mining** - Como a Tecnologia aumenta a Inteligência do Negócio. PUC-Rio, 2002. Disponível em: <<http://www.tec.com.br/BI.asp>>. Acesso em: 15 dez. 2002.

CRUZ, T. **Sistemas de informações gerenciais**. São Paulo: Atlas, 1996.

DACONTA, D. C.; OBRST, L. J.; SMITH K. T. **The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management**. USA: John Wiley & Sons, 2003.

DAML. **DAML Ontology Library**. Disponível em: <<http://www.daml.org/ontologies/>>. Acesso em: 3 jan. 2006.

_____. Services Coalition. Web Service Description for the Semantic Web. In: THE FIRST INT'L. SEMANTIC WEB CONF. (ISWC), 2002, Sardinia, Italy. Disponível em: <<http://www.daml.org/services/owl-s/ISWC2002-DAMLS.pdf>>. Acesso em: 23 jan. 2004.

DAVENPORT, T.; PRUSAK, L. **Conhecimento empresarial**. Rio de Janeiro: Campus, 1999.

DAVIES, J.; FENSEL, D.; VAN HARMELEN F. **Towards the Semantic Web: Ontology-driven knowledge management**. USA: John Wiley & Sons, 2003.

DEBEVOISE, N. T. **The Data Warehouse method**. New Jersey: Prentice Hall, 1999.

DEVANEY, J. **Business Intelligence strategies**. USA: Microsoft Press International, 2001.

DOMINGUE, J.; MOTTA, E.; WATT, S. The emerging VITAL workbench. In: THE 7th EUROPEAN WORKSHOP ON KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS. AUSSENAC, N. et al. (Ed.). Sep. 1993. p. 320-339.

DOMINGUE, J. et al. IRS-III: a platform and infrastructure for creating WSMO-based Semantic Web Services. In: WORKSHOP ON WSMO IMPLEMENTATIONS (WIW 2004), 2004, Frankfurt, Germany. Frankfurt, Germany, 2004a. ISSN 1613-0073.

_____. Demo of IRS-III: a platform and infrastructure for creating WSMO-based Semantic Web services. In: THE THIRD INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC 2004), 2004b, Hiroshima, Japan. (Demo Notes).

_____. Demo of IRS-III: a WSMO compliant platform and infrastructure for creating Semantic Web services. In: 14th INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT (EKAW 2004), 2004c, Whittlebury Hall, UK. (Demo Notes).

DOMINGUE, J. Tadzebao and WebOnto: discussing, browsing, and editing Ontologies on the Web. In: 11th KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP, 1998.

DONALD, B. **High performance Oracle Data Warehousing**. USA: The Coriolis Group, 1997.

DURKIN, J. **Expert systems: design and development**. New Jersey: Prentice Hall, 1994.

FARQUHAR, A.; FIKES, R.; RICE, J. P. The ontolingua server: A tool for collaborative ontology construction. **Journal of Human-Computer Studies**, 46, p. 707-728, 1997.

FELICIANO NETO, A.; SHIMIZU, T. **Sistemas flexíveis de informações**. São Paulo: Makron Books, 1996.

FENSEL, D. **Ontologies: silver bullet for knowledge management and electronic commerce**. Berlin: Springer-Verlag, 2001.

FENSEL, D.; BUSSLER, C. The Web Service Modeling Framework WSMF. **Electronic Commerce Research and Applications**, 1(2), p. 113-137, 2002.

FENSEL, D.; VAN HARMELEN, F. A comparison of languages which operationalize and formalize KADS models of expertise. **The Knowledge Engineering Review**, 9 (2), p. 105-146, 1994.

FENSEL, D. et al. **Spinning the Semantic Web: bringing the World Wide Web to Its full potential**. The MIT Press, 2002a.

FENSEL, D. et al. Semantic Web application areas. In: THE 7th INTERNATIONAL WORKSHOP ON APPLICATIONS OF NATURAL LANGUAGES TO INFORMATION SYSTEMS, Jun. 2002b, Stockholm, Sweden.

FILETO, R. et al. Using domain ontologies to help track data provenance. In: XVIII SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 2003, Manaus.

FOLDOC. **Coupling meaning**. Disponível em: <<http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=coupling&action=Search>>. Acesso em: 10 jan. 2005b.

_____. **Interface definition**. Disponível em: <<http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=interface>>. Acesso em: 10 jan. 2005a.

GALIZIA, S.; DOMINGUE, J. Towards a choreography for IRS-III. In: WORKSHOP ON WSMO IMPLEMENTATIONS (WIW 2004), Sep. 2004, Frankfurt, Germany. Frankfurt, Germany, 2004. ISSN 1613-0073.

GAMPER J.; NEJDL, W.; WOLPERS, M. Combining Ontologies and Terminologies in Information Systems. In: 5th INTERNATIONAL CONGRESS ON TERMINOLOGY AND KNOWLEDGE ENGINEERING, 1999, Innsbruck, Austria. Innsbruck, Austria, 1999.

GANGADHARAN, G. R.; SWAMI, S. N. Business Intelligence systems: design and implementation strategies. In: THE 26th INTERNATIONAL CONFERENCE

ON INFORMATION TECHNOLOGY INTERFACES - ITI 2004, 2004, Dubrovnik, Croatia. v.1. p. 139- 144.

GARTNER GROUP. **Gartner says more than 50 Percent of data warehouse projects will have limited acceptance or will be failures through 2007.**

Disponível em:
<http://www.gartner.com/press_releases/asset_121817_11.html>. Acesso em: 14 ago. 2004.

GIL, A. L. **Sistemas de informações:** contábil, financeira. 2. ed. São Paulo: Atlas, 1995.

GIOVINAZZO, W. A. **Object-Oriented Data Warehouse Design - Building a Star Schema.** New Jersey: Prentice Hall, 2000.

GÓMEZ-PÉREZ, A.; BENJAMINS, V. R. Overview of knowledge sharing and reuse: Ontologies and problem solving methods. In: WORKSHOP ON ONTOLOGIES AND PROBLEM-SOLVING METHODS: LESSONS LEARNED AND FUTURE TRENDS, INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI'99), 1999, Estocolmo.

GOMEZ-PEREZ, A.; CORCHO, O.; FERNANDEZ-LOPEZ, M. **Ontological Engineering:** with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. London: Springer-Verlag, 2004.

GONÇALVES, A. L. et al. Uma proposta baseada em agentes para a integração de serviços WEB. In: I2T2 - INTERNATIONAL INFORMATION TECHNOLOGY SYMPOSIUM, 2002, Florianópolis.

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

GROSOF, B. N. et al. Description logic programs: combining logic programs with description logic. In: INTL. CONF. ON THE WORLD WIDE WEB (WWW-2003), 2003, Budapest, Hungary.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, p.199-220. 1993.

GUARINO, N.; WELTY, C. **Conceptual modeling and ontological analysis.** Slides presented at the AAAI-2000 Tutorial on Conceptual Modeling and Ontological Analysis (MP-2). Jul. 31 2000. p. 5. Disponível em: <<http://www.cs.vassar.edu/faculty/welty/presentations/aaai-2000/>>. Acesso em: 20 nov. 2003.

GUARINO, N. Formal ontology and information systems. In: FOIS'98, Jun. 1998, Trento, Italy. Amsterdam: IOS Press, 1998. p. 3-15.

GUHA, R.; McCOOL, R. Tap: A semantic web platform. **Computer Networks: The International Journal of Computer and Telecommunications Networking**, v. 42, p. 557-577, 2003.

GUO, Y; PAN, Z; HEFLIN, J. An Evaluation of Knowledge Base Systems for Large OWL Datasets. In: International Semantic Web Conference, 2004. Volume 3, Issues 2-3 , October 2005, Pages 158-182.

HAKIMPOUR, F. et al. Semantic Web Service composition in IRS-III: the structured approach. In: 7th IEEE CONFERENCE ON E-COMMERCE (CEC2005), 2005, Munique, Alemanha.

HAKIMPOUR, F. et al. **IRS-III supporting the World of Semantic Web Services**. Disponível em: <http://www.hakimpour.com/farshad/download/WSMOWLS_IRS3.pdf>. Acesso em: 22 dez. 2004.

HARRISON, T. H. **Intranet Data Warehouse**. USA: McGraw-Hill, 1997.

HESSELBEIN, F. et al. **A Organização do Futuro**: como preparar hoje as empresas de amanhã. São Paulo: Futura, 1997.

HOFFMAN, T. Conference Attendees: CRM Initiatives May Miss Their Marks. **ComputerWorld**, 35(9), 7, 2001.

HOFREITER, B.; HUEMER, C.; WINIWARTER, W. Towards Syntax-Independent B2B. **ERCIM News No. 51, Special Theme: Semantic Web**, Oct. 2002.

HOLSAPPLE, C. W.; WHINSTON, A. B. **Decision support systems**: a knowledge-based approach. Minneapolis, MN: West Publishing Co., 1996.

HOWARD, Robert et al. **Aprendizado Organizacional**: gestão de pessoas para inovação contínua. Rio de Janeiro: Campus, 2000.

HURWITZ GROUP. **BI Web Services**. Disponível em: <<http://www.businessobjects.com/solutions/extranet/hurwitz.asp>>. Acesso em: 3 maio 2002.

IBM CORPORATION. **Business Process Execution Language for Web Services**. Disponível em: <<http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>>. Acesso em: 29 jun. 2004.

_____. **IBM Integrated Ontology Development Toolkit**. Disponível em: <<http://www.alphaworks.ibm.com/tech/semanticstk>>. Acesso em: 25 jul. 2006.

INMON, W. H. **Building the Data Warehouse**. 3. ed. New York: John Wiley & Sons, 2002.

_____. **Como construir o Data Warehouse**. Rio de Janeiro: Campus, 1997.

IRS-III API. Disponível em: <<http://kmi.open.ac.uk/projects/irs/irs3docs/api/index.html>>. Acesso em: 20 nov. 2004.

INSTITUTO STELA. **Métricas para Projetos ETL**. Relatório Técnico. 2005.

_____. Disponível em: <<http://www.stela.org.br>>. Acesso em: 20 jan. 2006.

Providing OLAP (On-line Analytical Processing) to user-analysts: an IT mandate. Technical Report. EUA, 1993.

JALOTE, P. *Software Project Management in Practice*. USA: Addison Wesley, 2002.

JENA. **Jena** – A Semantic Web Framework for Java. Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: 9 jan. 2006.

KELLY, T. J. **Dimensional Data Modeling**. Disponível em: <<http://www.sybase.com>>. Acesso em: 10 jun. 2000.

KIFER, M.; LAUSEN, G.; WU, J. Logical foundations of object-oriented and frame based languages. **Journal of the ACM**, 42(4), p. 741-843, 1995.

KIM, J.; SPRARAGEN, M.; GIL, Y. An intelligent assistant for interactive workflow composition, In: INT'L CONF. ON INTELLIGENT USER INTERFACES, 2004, Portugal.

KIMBALL, R. et al. **The Data Warehouse lifecycle toolkit: expert methods for designing, developing and deploying Data Warehouses**. New York: John Wiley & Sons, 1998.

KIMBALL, R.; MERZ, R. **The Data Webhouse toolkit: building the Web-enabled Data Warehouse**. John Wiley & Sons, 2000.

KIMBALL, R.; ROSS, M. **The Data Warehouse toolkit: the complete guide to dimensional modeling**. New York: John Wiley & Sons, 2002.

KIRYAKOV, A.; OGNJANOV, D.; MANOV, D. OWLIM - a Pragmatic Semantic Repository for OWL. In: International Workshop on Scalable Semantic Web Knowledge Base Systems, New York City, USA, 2005.

KMi. **Semantic Web – OCML to OWL Translator**. Disponível em: <<http://plainmoor.open.ac.uk:8080/ksw/pages/owl.html>>. Acesso em: 21 dez. 2004.

_____. **Plugin Mechanism**. Disponível em: <<http://kmi.open.ac.uk/projects/akt/aqualog/pluing.html>>. Acesso em: 20 jan. 2005.

LAUDON, K. C.; LAUDON, J. P. **Management information systems**. New approaches to organization & technology. New Jersey: Prentice Hall, 1998.

LEGER, A. et al. Ontology domain modeling support for multi-lingual services in E-Commerce. In: MKBEEM. ECAI'00 WORKSHOP ON APPLICATIONS OF ONTOLOGIES AND PSMS, Aug. 2000,

LI, L.; HORROCKS, I. A Software Framework for Matchmaking Based on Semantic Web Technology. In: 12th INT'L WORLD WIDE WEB CONF., WORLD WIDE WEB CONSORTIUM, 2003. p. 48.

LIEBOWITZ, J. Where is the "Intelligence" behind "Business Intelligence"?. **BusinessIntelligence.com**. Disponível em: <<http://www.businessintelligence.com/ex/asp/code.128/xe/article.htm>>. Acesso em: 28 mar. 2005a.

_____. Business Intelligence Cannot Exist Without Knowledge Management. **BusinessIntelligence.com**. Disponível em: <<http://www.businessintelligence.com/ex/asp/code.123/xe/article.htm>>. Acesso em: 30 mar. 2005b.

MACGREGOR, R. **LOOM Users Manual**. USC/Information Sciences Inst. Technical Reportn. ISI/WP-22, 1992.

MACHADO, F. N. R. **Projeto de Data Warehouse: Uma Visão Multidimensional**. São Paulo: Érica, 2000.

MAEDHE, A. et al. Ontologies for Enterprise Knowledge Management. **IEEE Intelligent Systems**, v.18, n. 2, p. 26-33, Mar./April 2003.

MARTIN, D. et al. Bringing Semantics to Web services: the OWL-S approach. In: THE FIRST INTERNATIONAL WORKSHOP ON SEMANTIC WEB SERVICES AND WEB PROCESS COMPOSITION (SWSWPC 2004), Jul. 2004, San Diego, California, USA.

McGEE, J.; PRUSAK, L. **Gerenciamento estratégico da informação**. Rio de Janeiro: Campus, 1994.

MCILRAITH, S.; SON, T. C.; ZENG, H. Semantic Web Services. **IEEE Intelligent Systems**, v. 16, n. 2, p. 46-53, 2001.

MECHELN, P. J. V. **SAIIP1-GI** - Sistema de apoio ao planejamento no processo de tomada de decisão do jogo de empresas GI-EPS. 1997. Dissertação (Mestrado em Engenharia de Produção) – Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, 1997.

MENA, E. et al. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. **International journal on Distributed and Parallel Databases (DAPD)**, 8(2), p. 223-272, 2000.

MISSIKOF, M. H. An Ontology-Based Approach for Semantic Interoperability. **ERCIM News**, n. 51, p. 33-34, Oct. 2002.

MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. **Task Ontology for reuse of problem solving knowledge**. Amsterdam: IOS Press, 1995. p. 46-57.

MOSS, L. T.; ATRE, S. **Business Intelligence roadmap: the complete project lifecycle for decision-support applications**. USA: Addison Wesley, 2003.

MOTIK, B.; MAEDCHE, A.; VOLZ, R. A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications. In: THE FIRST INTERNATIONAL CONFERENCE ON ONTOLOGIES, DATABASES AND APPLICATION OF SEMANTICS (ODBASE), 2002.

MOTTA, E. **Reusable Components for Knowledge Modelling**. Amsterdam, The Netherlands: IOS Press, 1999.

MOTTA, E. et al. IRS-II: A framework and infrastructure for Semantic Web services. In: THE 2nd INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC2003), Oct. 2003, Florida, USA.

NAPOLI M.; SELL D; LIDUARIO L.; BORGES R.; TODESCO J.; PACHECO R. Um framework para concepção de ferramentas de apoio à decisão baseadas em ontologias. In: XXI Simpósio Brasileiro de Banco de Dados (SBBD06), Oct. 2006, Florianópolis, Brasil.

NECIB, C. B.; FREITAG, J. Ontology based query processing in Database management systems. In: INTERNATIONAL CONFERENCE ON ONTOLOGIES, DATABASES AND APPLICATIONS OF SEMANTICS, Nov. 2003, Catania, Italy.

NEIL, B.; SCHRADER, M.; DAKIN, J. **Oracle Data Warehousing unleashed**. Publishing, Inc., 1997.

NEWELL, A. **The Knowledge Level. Artificial Intelligence**, 18 (1), p. 87-127, 1982.

OASIS. **ebXML** - Enabling a global electronic market. 2004. Disponível em: <<http://www.ebxml.org/>>. Acesso em: 11 jan. 2004.

O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet**. São Paulo: Saraiva, 2003.

OMG. **Common Warehouse metamodel specification**. 2000. v. 1 e 2. Disponível em: <<http://www.omg.org/>>. Acesso em: 13 nov. 2003.

_____. **Unified Modeling Language Resource Page**. Disponível em: <<http://www.uml.org/>>. Acesso em: 3 fev. 2005.

ORACLE. **Semantic Technologies Center**. Disponível em: <http://www.oracle.com/technology/tech/semantic_technologies/index.html>. Acesso em: 29 jul. 2006.

W3C. **OWL: The Web Ontology Language.** Disponível em: <<http://www.w3.org/2001/sw/WebOnt/>>. Acesso em: 5 jan. 2004.

PACHECO, R. C. S. **Uma metodologia de desenvolvimento de plataformas de governo para geração e divulgação de informações e de conhecimento.** Artigo apresentado em cumprimento a requisito parcial de concurso para professor no INE/UFSC. Florianópolis, 14 jan. 2003. 35 p.

PAN Z., QASEM A., HEFLIN J. An Investigation into the Feasibility of the Semantic Web. In Proceedings of the Twenty First National Conference on Artificial Intelligence (AAAI 2006), Boston, USA, 2006.

PATON, N. W. et al. Query processing in the Tambis Bioinformatics source integration system. **Statistical and Scientific Database Management**, p. 138-152, 1999.

PRIEBE, T.; PERNUL, G. Ontology-based integration of OLAP and information retrieval. In: THE DEXA 2003 WORKSHOP ON WEB SEMANTICS (WebS 2003), Sept. 2003, Prague, Czech Republic.

PROTÉGÉ. **The Protége Ontology Editor and Knowledge Acquisition System.** Disponível em: <<http://protege.stanford.edu/>>. Acesso em: 2 jan. 2005.

PROTÉGÉ **OWL. Ontologies.** Disponível em: <<http://protege.stanford.edu/plugins/owl/owl-library/>>. Acesso em: 3 jan. 2006.

RAJPATHAK, D. **A generic library of problem solving methods for scheduling applications.** Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Artificial Intelligence. 2004. The Open University, Milton Keynes, 2004.

RASMUSSEN, N.; GOLDY, P. S.; SOLLI, P. O. **Financial Business Intelligence.** Trends, technology, software selection, and implementation. New York: John Wiley and Sons, 2002.

REVISTA PLATAFORMA LATTES. Revista de informações em C&T do Brasil. Florianópolis: Grupo Stela, 2002.

RIVA, A.; RAMONI, M. LispWeb: A Specialised HTTP Server for Distributed AI Applications. **Computer Networks and ISDN Systems**, 28(7-11), p.953-961, 1996.

RULEML. **The Rule Markup Initiative.** Disponível em: <<http://www.ruleml.org/>>. Acesso em: 10 out. 2004.

SCHEMAWEB. **RDF Schemas Directory.** Disponível em: <<http://www.schemaweb.info/>>. Acesso em: 3 jan. 2006.

SELL, D. et al. A Framework to Improve Semantic Web services Discovery and Integration in an E-Gov Knowledge Network. In: 14th INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE

MANAGEMENT (EKAW2004), 2004a, Whittlebury Hall, UK. Lecture Notes in Computer Science. Short Paper.

SELL, D. et al. Interactive Composition of WSMO-based Semantic Web services in IRS-III. In: AKT WORKSHOP ON SEMANTIC WEB SERVICES, 2004, Milton Keynes, UK. Milton Keynes, UK, 2004b. v. 122.

SELL, D. et al. A Semantic Web based architecture for analytical tools. In: 7th IEEE CONFERENCE ON E-COMMERCE (CEC2005), 2005, Munique, Alemanha: IEEE Compute Society Press, 2005a.

SELL, D. et al. Adding Semantics to Business Intelligence. In: 4th INTERNATIONAL WORKSHOP ON WEB SEMANTICS, 16th INTERNATIONAL WORKSHOP ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA2005), 2005, Copenhagen, Denmark. IEEE Compute Society Press, 2005b.

SELL, D. **Uma arquitetura para distribuição de componentes tecnológicos de sistemas de informação baseados em Data Warehouse.** 2001. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001.

SELL, D. et al. Integração de agentes móveis e Web services: um estudo de caso. In: I2T2 - INTERNATIONAL INFORMATION TECHNOLOGY SYMPOSIUM, 2002, Florianópolis.

SELL, D.; PACHECO, R. C. S. Uma arquitetura para distribuição de componentes tecnológicos de sistemas de informações baseados em Data Warehouse. In: XXI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO - ENEGEP, 2001, Salvador. Rio de Janeiro: ABEPRO, 2001.

SHARMA, S. K.; GRUPTA, J. N. D. Knowledge Economy and Intelligent Enterprises. In: INTELLIGENT ENTERPRISES OF THE 21st Century, 2004, USA: Idea Group Publishing.

SHETH, A.; RAMAKRISHNAN, C. Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real. DAYAL, U.; KUNO, H.; WILKINSON, K. (Ed). December 2003.

SHUM S. B.; MOTTA E.; DOMINGUE J. ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. **International Journal on Digital Libraries**, Springer-Verlag, 3 (3), 2000.

SIMMERS, C. A. A stakeholder model of Business Intelligence. In: THE 37th HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES IEEE, 2004, Hawaii. ISSN 0-7695-2056-1.

SIRIN, E.; PARSIA, B.; HENDLER, J. Filtering semantic Web services with interactive composition techniques. **IEEE Intelligent Systems**, v.19, Issue 4, p. 42-49, 2004.

SOAP. **SOAP Version 1.2 Part 0**: Primer. 2003. Disponível em: <<http://www.w3.org/TR/soap12-part0/>>. Acesso em: 11 jan. 2004.

SOFA. **Simple Ontology Framework API**. Disponível em: <<http://sofa.projects.semwebcentral.org/>>. Acesso em: 10 jan. 2006.

STAIR, R. M. **Princípios de sistemas de informação**. Rio de Janeiro: S.A., 1998.

STUDER, R. et al. Situation and prospective of knowledge engineering. In: CUENA, J. et al. (Ed.). **Knowledge Engineering and Agent Technology**. IOS Series on Frontiers in Artificial Intelligence and Applications. IOS Press, 2000. Disponível em: <http://wwwdb.stanford.edu/%7Estefan/paper/2000/ios_2000.pdf>. Acesso em: 2 jan. 2003.

SUN. **Java Technology**. Disponível em: <<http://java.sun.com/>>. Acesso em: 10 out. 2004a.

_____. **JDBC Technology**. Disponível em: <<http://java.sun.com/products/jdbc/>>. Acesso em: 10 out. 2004b.

SURE Y. et al. OntoEdit: Collaborative Ontology Development for the SemanticWeb. In: THE FIRST INTERNATIONAL SEMANTIC WEB CONFERENCE 2002 (ISWC 2002), 2002, Sardinia, Italia. Sardinia, Italia, 2002.

SWSI. **Semantic Web Service Initiative**. 2002. Disponível em: <<http://www.swsi.org/>>. Acesso em: 2 fev. 2004.

SYCARA, K. et al. Automated Discovery, Interaction and Composition of Semantic Web services. **Journal of Web Semantics**, v. 1, issue 1, p. 27-46, Sep. 2003.

TAIT, T. F. C. **Um modelo de arquitetura de sistemas de informação para setor público: estudo em empresas estatais prestadoras de serviços de informática**. 2000. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2000.

TANLER, R. **Intranet Data Warehouse**. Rio de Janeiro: Infobook, 1998.

THOMSEN, E. **OLAP**: construindo sistemas de informação multidimensionais. Rio de Janeiro: Campus, 2002.

TISSOT, H. **Proposta para documentação de requisitos em projetos de Data Warehouse**. 2004. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004.

TODESCO, J.; SELL, D.; PACHECO, R. C. S. Arquitetura de Data Warehouse da Plataforma Lattes. In: CONFERÊNCIA SUL-AMERICANA EM CIÊNCIA E TECNOLOGIA APLICADA AO GOVERNO ELETRÔNICO, 2004, Florianópolis.

_____. Uma plataforma de Gestão para redes de distribuição de baixa tensão. In: XXIV ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO (ENEGEP, 2004), 2004, Florianópolis. **Anais...** Florianópolis, 2004.

TURBAN, E.; ARAONSON, J. E. **Decision support systems and intelligent systems**. 6. ed. Upper Saddle River, NJ: Prentice Hall, 2000.

UDDI. **UDDI Spec Technical Committee Specification v. 3.0**. 2003. Disponível em: <<http://uddi.org/pubs/uddi-v3.0.1-20031014.htm>>. Acesso em: 11 jan. 2004.

UNESCO. **Towards Knowledge Societies**. 2005. Disponível em: <<http://www.unesco.org/publications>>. Acesso em: 20 nov. 2005.

UNICODE. Disponível em: <<http://www.unicode.org>>. Acesso em: 15 jan. 2004.

VAN ELST, L.; ABECKER, A. Domain Ontology Agents in Distributed Organizational Memories. In: IJCAI'2001 WORKING NOTES OF THE WORKSHOP ON KNOWLEDGE MANAGEMENT AND ORGANIZATIONAL MEMORIES, 2001, Seattle, Washington, USA. p. 39-48. 2001.

VAN HEIJST, G. et al. Using explicit ontologies in KBS development. **International Journal of Human-Computer Studies**, 46 (2), p. 183-292, 1997.

VERSTRAET, A. A. **Systems definition**. Disponível em: <<http://www.smeal.psu.edu/misweb/systems/sycodef.html>>. Acesso em: 10 ago. 2000.

W3C. **Naming and addressing**. Disponível em: <<http://www.w3.org/Addressing>>. Acesso em: 15 jan. 2004.

_____. **Extensible Markup Language (XML)**. 1996. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 20 out. 2003.

_____. **HTTP** - Hypertext Transfer Protocol. 2000. Disponível em: <<http://www.w3.org/Protocols/>>. Acesso em: 20 out. 2003.

_____. **RDF Vocabulary Description Language 1.0: RDF Schema**. 2004. Disponível em: <<http://www.w3.org/TR/rdf-schema/>>. Acesso em: 11 jan. 2004.

_____. **Resource Description Framework (RDF)**. 1999. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em: 11 jan. 2004.

_____. **Web Services Activity**. World Wide Web Consortium. Disponível em: <<http://www.w3.org/2002/ws/>>. Acesso em: 12 jan. 2004.

_____. **Web Services Architecture**. 2004. Disponível em: <<http://www.w3.org/TR/ws-arch/>>. Acesso em: 12 jan. 2004.

_____. **XML Schema**. 2001. Disponível em: <<http://www.w3.org/XML/Schema>>. Acesso em: 20 out. 2003.

WALTZ, E. **Knowledge management in the Intelligence Enterprise**. Artech House, 2003.

WG SYSTEMS. **Arquitetura de um Sistema de BI**. Disponível em: <http://www.wgsystems.com.br/bi/bi_arquitetura.htm>. Acesso em: 12 fev. 2003.

WHITE, C. **The Role of Business Intelligence in Knowledge Management**. B Eye. Mar. 2005. Disponível em: <<http://www.b-eye-network.com/view/720>>. Acesso em: 30 mar. 2005.

WIKIPEDIA. **Graph Theory**. Disponível em: <http://en.wikipedia.org/wiki/Category:Graph_theory>. Acesso em: 15 abr. 2004a.

_____. **Factory method pattern**. Disponível em: <http://en.wikipedia.org/wiki/Factory_method_pattern>. Acesso em: 15 abr. 2004b.

_____. **Thread (computer science)**. Disponível em: <http://en.wikipedia.org/wiki/Thread_%28computer_science%29>. Acesso em: 20 abr. 2004c.

_____. **Object-oriented programming**. Disponível em: <http://en.wikipedia.org/wiki/Object-oriented_programming>. Acesso em: 20 abr. 2004d.

_____. **Object-oriented design**. http://en.wikipedia.org/wiki/Object-oriented_programming. Acesso em: 20 abr. 2004e.

WSDL. **Web Services Description Language (WSDL) 1.1**. 2001. Disponível em: <<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>>. Acesso em: 12 jan. 2004.

WSMO. **Web Service Modeling Ontology-Standard**. Disponível em: <<http://www.wsmo.org/2004/d2/>>. Acesso em: 1 nov. 2004.

_____. **WSMO Mediators**. Disponível em: <<http://www.wsmo.org/TR/d29/v0.1/>>. Acesso em: 2 abr. 2005.

XMLA. **XML for Analysis**. Disponível em: <<http://www.xmla.org/>>. Acesso em: 23 ago. 2004

Apêndice A

(Construtores da Ontologia BI)

```

(def-class db_element ()
)

(def-class db_collection (db_element)
  ((has_pk_attribute :type db_attribute)
   (has_attribute :type db_attribute)
   (has_schema :type string)
   (has_collection_name :type string)))

(def-class db_attribute (db_element)
  ((has_concept :type Class)
   (has_slot :type Slot)
   (has_allowed_additive_summarization :type Summarization)
   (is_additive :type boolean)
   (has_presentation_format :type string)
   (has_attribute_name :type string)))

(def-class db_collection_join ()
  ((has_coljoin_first_collection :type db_collection)
   (has_coljoin_first_collection_key :type db_attribute)
   (has_coljoin_second_collection :type db_collection)
   (has_coljoin_second_collection_key :type db_attribute)))

(def-class fact (db_collection)
  ((has_fact_dimension_usage :type dimension_usage)
   (has_fact_measure :type db_attribute)))

(def-class dimension (DB_Collection)
  ((has_dimension_name :type string)
   (has_presentation_slots :type db_attribute)
   (has_dimension_top_agg :type Boolean)
   (has_dimension_hierarchy :type hierarchy)))

(def-class dimension_usage ()
  ((has_usage_dimension :type dimension)
   (has_usage_collection_join :type db_collection_join)))

(def-class hierarchy ()
  ((has_hierarchy_name :type string)
   (has_hierarchy_level :type level)))

(def-class level ()
  ((has_level_name :type string)
   (is_unique_member :type boolean)
   (has_level_attribute :type db_attribute)))

(def-class db_operation ()
  ((has_sgbd_vendor :type DB_SGBD_Vendor)
   (has_initial_version_supported :type string)
   (has_operation_name :type string)
   (has_operation_syntax :type string)))

(def-class db_sgbd_vendor ()
  ((has_sgbd_name :type string)))

(def-class Summarization (DB_Operation)
)

(def-class Join (DB_Operation)
)

```

```

(def-class Aggregation (DB_Operation)
)

;;; Personalization of nomenclature used to present domain concepts and data sources

(def-class user ()
  (has_user_login :type string)
  (has_user_role :type role)
  (has_user_idiom :type idiom))

(def-class role ()
  (has_role_name :type string)
  (has_role_granted_db_element :type db_element)
  (has_role_granted_level :type level)
  (has_role_granted_service :type invocable-entity)
  (has_role_granted_filter :type relation))

(def-class idiom ()
  (has_idiom_name :type string))

(def-class label ()
  (has_label_db_element :type db_element)
  (has_label_idiom :type idiom)
  (has_label_user_role :type role)
  (has_label_description :type string))

(def-relation label-role-idiom (?column ?role ?idiom ?label)
:sufficient
  (and (has_attribute_name ?att ?column)
        (has_label_db_element ?db ?att)
        (has_label_user_role ?db ?role)
        (has_label_idiom ?db ?idiom)
        (has_label_description ?db ?label)))

(def-relation label-role (?column ?role ?label)
:sufficient (and
  (has_attribute_name ?att ?column)
  (has_label_db_element ?db ?att)
  (has_label_user_role ?db ?role)
  (has_label_description ?db ?label)
))

(def-relation presentation-attribute (?concept ?slot)
:sufficient (and
  (has_concept ?att ?concept)
  (has_attribute ?dim ?att)
  (has_presentation_slots ?dim ?pres)
  (has_slot ?pres ?slot)
))

;;; Semantic definition of Analyses

(def-class Analysis ()
  (has_analysis_description :type string)
  (has_analysis_content :type Analysis_Content)
  (has_analysis_dimension :type Analysis_Dimension)
  (has_analysis_filter :type Analysis_Filter)
  (has_analysis_parameter :type Analysis_Parameter)
  (has_analysis_creator :type User)
  (has_analysis_allowed_user :type User)
  (has_analysis_allowed_role :type Role)
))

(def-class Analysis_Content ()
  (has_content_measure :type DB_Attribute)
  (has_content_summarization :type Summarization)
))

```

```

(def-class Analysis_Dimension ()
  ((has_dimension_attribute :type DB_Attribute)
   (has_dimension_position :type string)
  ))

(def-class Analysis_Filter ()
  ((has_filter_attribute :type DB_Attribute)
   (has_filter_operator :type DB_Operation)
   (has_filter_value :type Class)
  ))

(def-class Analysis_Parameter ()
  ((has_parameter_value :type Class)
   (is_parameter_single_value :type boolean)
  ))

(def-class Analysis_Unit ()
  ((has_analysis_unit_code :type string)
   (has_analysis_unit_name :type string)
   (has_analysis_unit_dimension_usage :type Dimension_Usage)
   (has_analysis_unit_measure :type DB_Attribute)
   (has_analysis_unit_allowed_user :type User)
   (has_analysis_unit_allowed_role :type Role)
  ))

;;; Sample of Instances

(def-instance lbl_att_fato_grupo_tot_pesquisador_role_company_manager label
  ((has_label_db_element att_fato_grupo_tot_pesquisador)
   (has_label_idiom idiom_english)
   (has_label_user_role role_company_manager)
   (has_label_description "NUMBER_EXPERTS")))

(def-instance lbl_att_fato_grupo_tot_pesquisador_role_research_manager label
  ((has_label_db_element att_fato_grupo_tot_pesquisador)
   (has_label_idiom idiom_english)
   (has_label_user_role role_research_manager)
   (has_label_description "NUMBER_RESEARCHERS")))

(def-instance lbl_att_fato_grupo_tot_estudante_role_company_manager label
  ((has_label_db_element att_fato_grupo_tot_estudante)
   (has_label_idiom idiom_english)
   (has_label_user_role role_company_manager)
   (has_label_description "NUMBER_TRAINEES")))

(def-instance lbl_att_fato_grupo_tot_estudante_role_research_manager label
  ((has_label_db_element att_fato_grupo_tot_estudante)
   (has_label_idiom idiom_english)
   (has_label_user_role role_research_manager)
   (has_label_description "NUMBER_STUDENTS")))

(def-instance lbl_att_dim_instituicao_nme_inst_role_company_manager label
  ((has_label_db_element att_dim_instituicao_nme_inst)
   (has_label_idiom idiom_english)
   (has_label_user_role role_company_manager)
   (has_label_description "INSTITUTION_NAME")))

(def-instance lbl_att_dim_instituicao_nme_inst_role_research_manager label
  ((has_label_db_element att_dim_instituicao_nme_inst)
   (has_label_idiom idiom_english)
   (has_label_user_role role_research_manager)
   (has_label_description "UNIVERSITY_NAME")))

(def-instance lbl_att_dim_instituicao_sgl_inst_role_company_manager label
  ((has_label_db_element att_dim_instituicao_sgl_inst)
   (has_label_idiom idiom_english)
   (has_label_user_role role_company_manager)
   (has_label_description "INSTITUTION_ACRONYM")))

(def-instance lbl_att_dim_instituicao_sgl_inst_role_research_manager label
  ((has_label_db_element att_dim_instituicao_sgl_inst)
   (has_label_idiom idiom_english)
   (has_label_user_role role_research_manager)
   (has_label_description "UNIVERSITY_ACRONYM")))

(def-instance lbl_att_dim_geografia_nme_uf label

```

```

((has_label_db_element att_dim_geografia_nme_uf)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "NAME_STATE"))

(def-instance lbl_att_dim_geografia_sgl_pais label
 ((has_label_db_element att_dim_geografia_sgl_pais)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "ACRONYM_COUNTRY"))

(def-instance lbl_att_dim_geografia_nme_municipio label
 ((has_label_db_element att_dim_geografia_nme_municipio)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "NAME_CITY"))

(def-instance lbl_att_dim_geografia_sgl_uf label
 ((has_label_db_element att_dim_geografia_sgl_uf)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "ACRONYM_STATE"))

(def-instance lbl_att_dim_geografia_seq_id_geografia label
 ((has_label_db_element att_dim_geografia_seq_id_geografia)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "ID_GEOGRAPHY"))

(def-instance lbl_att_dim_geografia_nme_pais label
 ((has_label_db_element att_dim_geografia_nme_pais)
 (has_label_idiom idiom_english)
 (has_label_user_role role_research_manager role_company_manager)
 (has_label_description "NAME_COUNTRY"))

;;; Dim_Instituicao

(def-instance dimension_dim_instituicao dimension
 ((has_pk_attribute att_dim_instituicao_seq_id_inst)
 (has_attribute att_dim_instituicao_nme_inst
                 att_dim_instituicao_cod_inst
                 att_dim_instituicao_sgl_inst
                 att_dim_instituicao_cod_inst_parent)
 (has_presentation_slots att_dim_instituicao_nme_inst)
 (has_collection_name "dim_INSTITUICAO")))

(def-instance att_dim_instituicao_seq_id_inst db_attribute
 ((has_allowed_additive_summarization Summarization_Count_Distinct Summarization_Count)
 (has_concept institution)
 (has_slot has_institution_id)
 (is_additive "true")
 (has_attribute_name "SEQ_ID_INST")))

(def-instance att_dim_instituicao_nme_inst db_attribute
 ((has_attribute_name "NME_INST")
 (has_concept institution)
 (has_slot has_institution_name)
 (is_additive "false")))

(def-instance att_dim_instituicao_sgl_inst db_attribute
 ((has_attribute_name "SGL_INST")
 (has_concept institution)
 (has_slot has_institution_acronym)
 (is_additive "false")))

(def-instance att_dim_instituicao_cod_inst db_attribute
 ((has_attribute_name "COD_INST")
 (has_concept institution)
 (has_slot has_institution_code)
 (has_allowed_additive_summarization
           Summarization_Count_Distinct
           Summarization_Count)
 (is_additive "true")))

(def-instance att_dim_instituicao_cod_inst_parent db_attribute
 ((has_attribute_name "COD_INST_PARENT"))

```

```
(has_concept institution)
(has_slot has_institution_parent)
(has_allowed_additive_summarization
  Summarization_Count_Distinct
  Summarization_Count)
(is_additive "true"))
```

```
;;; Dim_Grupo
```

```
(def-instance dimension_dim_grupo dimension
  (has_collection_name "dim_GRUPO")
  (has_pk_attribute att_dim_grupo_seq_id_grupo)
  (has_presentation_slots att_dim_grupo_nme_grupo)
  (has_attribute att_dim_grupo_nme_grupo
    att_dim_grupo_ano_formacao
    att_dim_grupo_nro_id_grupo)))

(def-instance att_dim_grupo_nme_grupo db_attribute
  (has_attribute_name "NME_GRUPO")
  (has_concept research_group)
  (has_slot has_group_name)
  (is_additive "false"))

(def-instance att_dim_grupo_seq_id_grupo db_attribute
  (has_attribute_name "SEQ_ID_GRUPO")
  (has_concept research_group)
  (has_slot has_group_id)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct)))

(def-instance att_dim_grupo_nro_id_grupo db_attribute
  (has_attribute_name "NRO_ID_GRUPO")
  (has_concept research_group)
  (has_slot has_group_code)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct)))

(def-instance att_dim_grupo_year_formation db_attribute
  (has_attribute_name "ANO_FORMACAO")
  (has_concept research_group)
  (has_slot has_group_creation_year)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Avg
    Summarization_Min Summarization_Max)))
```

```
;;; Dim_Geografia
```

```
(def-instance dimension_dim_geografia dimension
  (has_attribute att_dim_geografia_nme_uf
    att_dim_geografia_sgl_uf
    att_dim_geografia_nme_municipio
    att_dim_geografia_sgl_pais
    att_dim_geografia_nme_pais)
  (has_pk_attribute att_dim_geografia_seq_id_geografia)
  (has_presentation_slots
    att_dim_geografia_sgl_pais
    att_dim_geografia_sgl_uf
    att_dim_geografia_nme_municipio)
  (has_collection_name "dim_GEOGRAFIA"))

(def-instance att_dim_geografia_seq_id_geografia db_attribute
  (has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
  (has_attribute_name "SEQ_ID_GEOGRAFIA")
  (has_concept city)
  (has_slot has_city_id)
  (is_additive "true"))

(def-instance att_dim_geografia_nme_municipio db_attribute
  (has_attribute_name "NME_MUNICIPIO"))
```

```

(has_concept city)
(has_slot has_city_name)
(is_additive "false")
(has_concept has_city_name)))

(def-instance att_dim_geografia_sgl_uf db_attribute
((is_additive "false")
 (has_concept state)
 (has_slot has_state_acronym)
 (has_attribute_name "SGL_UF"))))

(def-instance att_dim_geografia_nme_uf db_attribute
((has_attribute_name "NME_UF")
 (has_concept state)
 (has_slot has_state_name)
 (is_additive "false"))))

(def-instance att_dim_geografia_sgl_pais db_attribute
((is_additive "false")
 (has_concept country)
 (has_slot has_country_acronym)
 (has_attribute_name "SGL_PAIS"))))

(def-instance att_dim_geografia_nme_pais db_attribute
((has_attribute_name "NME_PAIS")
 (has_concept country)
 (has_slot has_country_name)
 (is_additive "false")
 (has_concept has_country_name)))

;;; Dim_Tempo

(def-instance dimension_dim_tempo dimension
((has_collection_name "dim_TEMPO")
 (has_pk_attribute att_dim_tempo_seq_id_tempo)
 (has_presentation_slots att_dim_tempo_dsc_mes)
 (has_attribute att_dim_tempo_nro_mes
                 att_dim_tempo_dsc_mes
                 att_dim_tempo_nro_trimestre
                 att_dim_tempo_nro_ano)))

(def-instance att_dim_tempo_seq_id_tempo db_attribute
((has_attribute_name "SEQ_ID_TEMPO")
 (is_additive "true")
 (has_allowed_additive_summarization
  Summarization_Count
  Summarization_Count_Distinct)))

(def-instance att_dim_tempo_dsc_mes db_attribute
((has_attribute_name "DSC_MES")
 (has_concept month_name)
 (has_slot month_name)
 (is_additive "false")))

(def-instance att_dim_tempo_nro_mes db_attribute
((has_attribute_name "NRO_MES")
 (has_concept month)
 (has_slot month_number)
 (is_additive "false")))

(def-instance att_dim_tempo_nro_trimestre db_attribute
((has_attribute_name "NRO_TRIMESTRE")
 (has_concept quarter)
 (has_slot quarter_number)
 (is_additive "true")
 (has_allowed_additive_summarization Summarization_Count
  Summarization_Count_Distinct)))

(def-instance att_dim_tempo_nro_ano db_attribute
((has_attribute_name "NRO_ANO")
 (has_concept year)
 (has_slot year_number)
 (is_additive "true")
 (has_allowed_additive_summarization Summarization_Count
  Summarization_Count_Distinct)))

```

```

;;; Dim_Pessoa

(def-instance dimension_dim_pessoa dimension
  (has_collection_name "dim_PESSOA")
  (has_pk_attribute att_dim_pessoa_seq_id_pessoa)
  (has_presentation_slots att_dim_pessoa_nme_rh)
  (has_attribute att_dim_pessoa_nme_rh
    att_dim_pessoa_cod_sexo
    att_dim_pessoa_dsc_faixa_etaria
    att_dim_pessoa_cod_inst_endereco
    att_dim_pessoa_cod_municipio_endereco
    att_dim_pessoa_sgl_uf_endereco
    att_dim_pessoa_sgl_regiao_endereco
    att_dim_pessoa_dsc_pais_endereco))

(def-instance att_dim_pessoa_seq_id_pessoa db_attribute
  (has_attribute_name "SEQ_ID_PESSOA")
  (has_concept person)
  (has_slot has_person_id)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct))

(def-instance att_dim_pessoa_nme_rh db_attribute
  (has_attribute_name "NME_RH")
  (has_concept person)
  (has_slot has_person_name)
  (is_additive "false"))

(def-instance att_dim_pessoa_cod_sexo db_attribute
  (has_attribute_name "COD_SEXO")
  (has_concept person)
  (has_slot has_person_gender)
  (is_additive "false"))

(def-instance att_dim_pessoa_dsc_faixa_etaria db_attribute
  (has_attribute_name "DSC_FAIXA_ETARIA")
  (has_concept person)
  (has_slot has_person_age)
  (is_additive "false"))

(def-instance att_dim_pessoa_cod_inst_endereco db_attribute
  (has_attribute_name "COD_INST_ENDERECO")
  (has_concept person)
  (has_slot has_person_institutional_address)
  (is_additive "false"))

(def-instance att_dim_pessoa_cod_municipio_endereco db_attribute
  (has_attribute_name "COD_MUNICIPIO_ENDERECO")
  (has_concept city)
  (has_slot has_city_code)
  (is_additive "false"))

(def-instance att_dim_pessoa_sgl_uf_endereco db_attribute
  (has_attribute_name "SGL_UF_ENDERECO")
  (has_concept state)
  (has_slot has_state_acronym)
  (is_additive "false"))

(def-instance att_dim_pessoa_sgl_regiao_endereco db_attribute
  (has_attribute_name "SGL_REGIAO_ENDERECO")
  (has_concept region)
  (has_slot has_region_acronym)
  (is_additive "false"))

(def-instance att_dim_pessoa_dsc_pais_endereco db_attribute
  (has_attribute_name "DSC_PAIS_ENDERECO")
  (has_concept country)
  (has_slot has_country_name)
  (is_additive "false"))

;;; Dim_Formacao

(def-instance dimension_dim_formacao dimension

```

```

((has_collection_name "dim_FORMACAO")
 (has_pk_attribute att_dim_formacao_seq_id_form att_dim_formacao_seq_id_pessoa)
 (has_presentation_slots att_dim_formacao_dsc_nivel_form att_dim_formacao_sgl_inst)
 (has_attribute att_dim_formacao_dsc_nivel_form
  att_dim_formacao_cod_curso
  att_dim_formacao_dsc_curso
  att_dim_formacao_tpo_curso
  att_dim_formacao_cod_nivel_curso
  att_dim_formacao_cod_grande_area_curso
  att_dim_formacao_nme_grande_area_curso
  att_dim_formacao_cod_area_curso
  att_dim_formacao_nme_area_curso
  att_dim_formacao_ano_ini_form
  att_dim_formacao_ano_ter_form
  att_dim_formacao_cod_inst
  att_dim_formacao_sgl_inst)))

(def-instance att_dim_formacao_seq_id_pessoa db_attribute
 (has_attribute_name "SEQ_ID_PESSOA")
 (has_concept person)
 (has_slot has_person_id)
 (is_additive "true")
 (has_allowed_additive_summarization Summarization_Count
  Summarization_Count_Distinct)))

(def-instance att_dim_formacao_seq_id_form db_attribute
 (has_attribute_name "SEQ_ID_FORM")
 (has_concept degree)
 (has_slot has_degree_id)
 (is_additive "true")
 (has_allowed_additive_summarization Summarization_Count
  Summarization_Count_Distinct)))

(def-instance att_dim_formacao_dsc_nivel_form db_attribute
 (has_attribute_name "DSC_NIVEL_FORM")
 (has_concept degree)
 (has_slot has_degree_level)
 (is_additive "false")))

(def-instance att_dim_formacao_cod_curso db_attribute
 (has_attribute_name "COD_CURSO")
 (has_concept course)
 (has_slot has_course_code)
 (is_additive "false")))

(def-instance att_dim_formacao_dsc_curso db_attribute
 (has_attribute_name "DSC_CURSO")
 (has_concept course)
 (has_slot has_course_name)
 (is_additive "false")))

(def-instance att_dim_formacao_tpo_curso db_attribute
 (has_attribute_name "TPO_CURSO")
 (has_concept course)
 (has_slot has_course_type)
 (is_additive "false")))

(def-instance att_dim_formacao_cod_nivel_curso db_attribute
 (has_attribute_name "COD_NIVEL_CURSO")
 (has_concept course)
 (has_slot has_course_level)
 (is_additive "false")))

(def-instance att_dim_formacao_cod_grande_area_curso db_attribute
 (has_attribute_name "COD_GRANDE_AREA_CURSO")
 (has_concept knowledge_area)
 (has_slot has_knowledge_area_code)
 (is_additive "false")))

(def-instance att_dim_formacao_nme_grande_area_curso db_attribute
 (has_attribute_name "NME_GRANDE_AREA_CURSO")
 (has_concept knowledge_area)
 (has_slot has_knowledge_area_name)
 (is_additive "false")))

(def-instance att_dim_formacao_cod_area_curso db_attribute
 (has_attribute_name "COD_AREA_CURSO")

```

```

(has_concept knowledge_area)
(has_slot has_knowledge_area_code)
(is_additive "false"))

(def-instance att_dim_formacao_nme_area_curso db_attribute
  (has_attribute_name "NME_AREA_CURSO")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false"))

(def-instance att_dim_formacao_ano_ini_form db_attribute
  (has_attribute_name "ANO_INI_FORM")
  (has_concept degree)
  (has_slot has_start_year)
  (is_additive "false"))

(def-instance att_dim_formacao_ano_ter_form db_attribute
  (has_attribute_name "ANO_TER_FORM")
  (has_concept degree)
  (has_slot has_end_year)
  (is_additive "false"))

(def-instance att_dim_formacao_cod_inst db_attribute
  (has_attribute_name "COD_INST")
  (has_concept institution)
  (has_slot has_institution_code)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct))

(def-instance att_dim_formacao_sgl_inst db_attribute
  (has_attribute_name "SGL_INST")
  (has_concept institution)
  (has_slot has_institution_acronym)
  (is_additive "false"))

;;; Dim_Atividade

(def-instance dimension_dim_atividade dimension
  (has_collection_name "dim_ATIVIDADE")
  (has_pk_attribute att_dim_atividade_seq_id_atividade att_dim_atividade_seq_id_pessoa)
  (has_presentation_slots att_dim_atividade_dsc_natur_ativ att_dim_atividade_sgl_inst)
  (has_attribute att_dim_atividade_dsc_natur_ativ
    att_dim_atividade_cod_curso
    att_dim_atividade_dsc_curso
    att_dim_atividade_tpo_curso
    att_dim_atividade_cod_nivel_curso
    att_dim_atividade_cod_grande_area_curso
    att_dim_atividade_nme_grande_area_curso
    att_dim_atividade_cod_area_curso
    att_dim_atividade_nme_area_curso
    att_dim_atividade_mes_inic
    att_dim_atividade_ano_inic
    att_dim_atividade_mes_fim
    att_dim_atividade_ano_fim
    att_dim_atividade_cod_inst
    att_dim_atividade_sgl_inst))

(def-instance att_dim_atividade_seq_id_pessoa db_attribute
  (has_attribute_name "SEQ_ID_PESSOA")
  (has_concept person)
  (has_slot has_person_id)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct))

(def-instance att_dim_atividade_seq_atividade db_attribute
  (has_attribute_name "SEQ_ATIVIDADE")
  (has_concept activity)
  (has_slot has_activity_id)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct))

(def-instance att_dim_atividade_dsc_natur_ativ db_attribute
  (has_attribute_name "DSC_NATUR_ATIV")

```

```

(has_concept activity)
(has_slot has_activity_type)
(is_additive "false"))

(def-instance att_dim_atividade_cod_curso db_attribute
  (has_attribute_name "COD_CURSO")
  (has_concept course)
  (has_slot has_course_code)
  (is_additive "false"))

(def-instance att_dim_atividade_dsc_curso db_attribute
  (has_attribute_name "DSC_CURSO")
  (has_concept course)
  (has_slot has_course_name)
  (is_additive "false"))

(def-instance att_dim_atividade_tpo_curso db_attribute
  (has_attribute_name "TPO_CURSO")
  (has_concept course)
  (has_slot has_course_type)
  (is_additive "false"))

(def-instance att_dim_atividade_cod_nivel_curso db_attribute
  (has_attribute_name "COD_NIVEL_CURSO")
  (has_concept course)
  (has_slot has_course_level)
  (is_additive "false"))

(def-instance att_dim_atividade_cod_grande_area_curso db_attribute
  (has_attribute_name "COD_GRANDE_AREA_CURSO")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false"))

(def-instance att_dim_atividade_nme_grande_area_curso db_attribute
  (has_attribute_name "NME_GRANDE_AREA_CURSO")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false"))

(def-instance att_dim_atividade_cod_area_curso db_attribute
  (has_attribute_name "COD_AREA_CURSO")
  (has_concept area)
  (has_slot has_area_code)
  (is_additive "false"))

(def-instance att_dim_atividade_nme_area_curso db_attribute
  (has_attribute_name "NME_AREA_CURSO")
  (has_concept area)
  (has_slot has_area_name)
  (is_additive "false"))

(def-instance att_dim_atividade_ano_inic db_attribute
  (has_attribute_name "ANO_INIC")
  (has_concept activity)
  (has_slot has_start_year)
  (is_additive "false"))

(def-instance att_dim_atividade_mes_inic db_attribute
  (has_attribute_name "MES_INIC")
  (has_concept activity)
  (has_slot has_start_month)
  (is_additive "false"))

(def-instance att_dim_atividade_ano_fim db_attribute
  (has_attribute_name "ANO_FIM")
  (has_concept activity)
  (has_slot has_end_year)
  (is_additive "false"))

(def-instance att_dim_atividade_mes_fim db_attribute
  (has_attribute_name "MES_FIM")
  (has_concept activity)
  (has_slot has_end_month)
  (is_additive "false"))

(def-instance att_dim_atividade_cod_inst db_attribute

```

```

((has_attribute_name "COD_INST")
 (has_concept institution)
 (has_slot has_institution_code)
 (is_additive "true")
 (has_allowed_additive_summarization Summarization_Count
   Summarization_Count_Distinct)))

(def-instance att_dim_atividade_sgl_inst db_attribute
 ((has_attribute_name "SGL_INST")
  (has_concept institution)
  (has_slot has_institution_acronym)
  (is_additive "false")))

;;; Dim_Area_Conhec

(def-instance dimension_dim_area_conhec dimension
 ((has_collection_name "dim_AREA_CONHEC")
  (has_pk_attribute att_dim_area_conhec_seq_id_area)
  (has_presentation_slots att_dim_area_conhec_nme_grande_area
    att_dim_area_conhec_nme_area_conhec
    att_dim_area_conhec_nme_subarea_conhec
    att_dim_area_conhec_nme_especialidade)
  (has_attribute att_dim_area_conhec_cod_grande_area
    att_dim_area_conhec_cod_area_conhec
    att_dim_area_conhec_cod_subarea_conhec
    att_dim_area_conhec_cod_especialidade
    att_dim_area_conhec_nme_grande_area
    att_dim_area_conhec_nme_area_conhec
    att_dim_area_conhec_nme_subarea_conhec
    att_dim_area_conhec_nme_especialidade
    att_dim_area_conhec_cod_area_parent
    att_dim_area_conhec_nme_nivel_area)))

(def-instance att_dim_area_conhec_seq_id_area db_attribute
 ((has_attribute_name "SEQ_ID_AREA")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_id)
  (is_additive "true")
  (has_allowed_additive_summarization Summarization_Count
    Summarization_Count_Distinct)))

(def-instance att_dim_area_conhec_cod_grande_area db_attribute
 ((has_attribute_name "COD_GRANDE_AREA")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false")))

(def-instance att_dim_area_conhec_nme_grande_area db_attribute
 ((has_attribute_name "NME_GRANDE_AREA")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false")))

(def-instance att_dim_area_conhec_cod_area_conhec db_attribute
 ((has_attribute_name "COD_AREA_CONHEC")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false")))

(def-instance att_dim_area_conhec_nme_area_conhec db_attribute
 ((has_attribute_name "NME_AREA_CONHEC")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false")))

(def-instance att_dim_area_conhec_cod_subarea_conhec db_attribute
 ((has_attribute_name "COD_SUBAREA_CONHEC")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false")))

(def-instance att_dim_area_conhec_nme_subarea_conhec db_attribute
 ((has_attribute_name "NME_SUBAREA_CONHEC")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false")))

```

```

(is_additive "false"))

(def-instance att_dim_area_conhec_cod_especialidade db_attribute
  (has_attribute_name "COD_ESPECIALIDADE")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false"))

(def-instance att_dim_area_conhec_nme_especialidade db_attribute
  (has_attribute_name "NME_ESPECIALIDADE")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_name)
  (is_additive "false"))

(def-instance att_dim_area_conhec_cod_area_parent db_attribute
  (has_attribute_name "COD_AREA_PARENT")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_code)
  (is_additive "false"))

(def-instance att_dim_area_conhec_cod_nivel_area db_attribute
  (has_attribute_name "COD_NIVEL_AREA")
  (has_concept knowledge_area)
  (has_slot has_knowledge_area_level)
  (is_additive "false"))

;;; Fato_Grupo

(def-instance fact_fato_grupo fact
  (has_pk_attribute att_fato_grupo_seq_id_grupo
    att_fato_grupo_seq_id_inst
    att_fato_grupo_seq_id_tempo
    att_fato_grupo_seq_id_geografia)
  (has_fact_measure att_fato_grupo_tot_pesquisador
    att_fato_grupo_tot_linha_pesquisa
    att_fato_grupo_tot_tecnico
    att_fato_grupo_tot_estudante)
  (has_fact_dimension_usage
    dimension_usage_fato_grupo_dim_grupo
    dimension_usage_fato_grupo_dim_geografia)
  (has_collection_name "fato_GRUPO"))

(def-instance att_fato_grupo_tot_pesquisador db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_PESQUISADORES")
  (is_additive "true"))

(def-instance att_fato_grupo_tot_estudante db_attribute
  ((is_additive "true")
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_ESTUDANTES"))

(def-instance att_fato_grupo_tot_tecnico db_attribute
  (is_additive "true")
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_TECNICOS"))

(def-instance att_fato_grupo_tot_linha_pesquisa db_attribute
  ((is_additive "true")
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)

```

```

(has_attribute_name "TOT_LINHAS_PESQUISA"))

(def-instance att_fato_grupo_seq_id_geografia db_attribute
  ((has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
   (has_attribute_name "SEQ_ID_GEOGRAFIA")
   (is_additive "true")))

(def-instance att_fato_grupo_seq_id_inst db_attribute
  ((has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
   (has_attribute_name "SEQ_ID_INST")
   (is_additive "true")))

(def-instance att_fato_grupo_seq_id_tempo db_attribute
  ((has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
   (has_attribute_name "SEQ_ID_TEMPO")
   (is_additive "true")))

(def-instance att_fato_grupo_seq_id_grupo db_attribute
  ((has_attribute_name "SEQ_ID_GRUPO")
   (is_additive "true")
   (has_allowed_additive_summarization
    Summarization_Count
    Summarization_Count_Distinct)))

;;; Fato_GRUPO Relationships definition

(def-instance dimension_usage_fato_grupo_dim_grupo dimension_usage
  ((has_usage_dimension dimension_dim_grupo)
   (has_usage_collection_join db_collection_join_fato_grupo_dim_grupo)))

(def-instance db_collection_join_fato_grupo_dim_grupo db_collection_join
  ((has_coljoin_first_collection fact_fato_grupo)
   (has_coljoin_first_collection_key att_fato_grupo_seq_id_grupo)
   (has_coljoin_second_collection dimension_dim_grupo)
   (has_coljoin_second_collection_key att_dim_grupo_seq_id_grupo)))

(def-instance dimension_usage_fato_grupo_dim_geografia dimension_usage
  ((has_usage_dimension dimension_dim_geografia)
   (has_usage_collection_join db_collection_join_fato_grupo_dim_geografia)))

(def-instance db_collection_join_fato_grupo_dim_geografia db_collection_join
  ((has_coljoin_first_collection fact_fato_grupo)
   (has_coljoin_first_collection_key att_fato_grupo_seq_id_geografia)
   (has_coljoin_second_collection dimension_dim_geografia)
   (has_coljoin_second_collection_key att_dim_grupo_seq_id_geografia)))

(def-instance dimension_usage_fato_grupo_dim_tempo dimension_usage
  ((has_usage_dimension dimension_dim_tempo)
   (has_usage_collection_join db_collection_join_fato_grupo_dim_tempo)))

(def-instance db_collection_join_fato_grupo_dim_tempo db_collection_join
  ((has_coljoin_first_collection fact_fato_grupo)
   (has_coljoin_first_collection_key att_fato_grupo_seq_id_grupo)
   (has_coljoin_second_collection dimension_dim_tempo)
   (has_coljoin_second_collection_key att_dim_tempo_seq_id_tempo)))

(def-instance dimension_usage_fato_grupo_dim_instituicao dimension_usage
  ((has_usage_dimension dimension_dim_grupo)
   (has_usage_collection_join db_collection_join_fato_grupo_dim_instituicao)))

(def-instance db_collection_join_fato_grupo_dim_instituicao db_collection_join
  ((has_coljoin_first_collection fact_fato_grupo)
   (has_coljoin_first_collection_key att_fato_grupo_seq_id_grupo)
   (has_coljoin_second_collection dimension_dim_instituicao)
   (has_coljoin_second_collection_key att_dim_instituicao_seq_id_inst)))

;;; Fato_Pessoa

(def-instance fact_fato_pessoa fact

```

```

((has_pk_attribute att_fato_pessoa_seq_id_pessoa
  att_fato_pessoa_seq_id_inst
  att_fato_pessoa_cod_periodo
  att_fato_grupo_seq_id_geografia)
 (has_fact_measure att_fato_pessoa_tot_fim_docentes
  att_fato_pessoa_tot_fim_orientadores
  att_fato_pessoa_tot_fim_pesquisadores
  att_fato_pessoa_tot_fim_estudantes
  att_fato_pessoa_tot_fim_egressos
  att_fato_pessoa_tot_fim_extensao_univ
  att_fato_pessoa_tot_fim_treinamento
  att_fato_pessoa_tot_fim_outra_ativ
  att_fato_pessoa_tot_meio_funcionarios_lotados
  att_fato_pessoa_tot_meio_funcionarios_vincul
  att_fato_pessoa_tot_meio_administradores
  att_fato_pessoa_tot_meio_estagio
  att_fato_pessoa_tot_meio_serv_tecnico)
 (has_fact_dimension_usage
  dimension_usage_fato_pessoa_dim_pessoa
  dimension_usage_fato_pessoa_dim_geografia
  dimension_usage_fato_pessoa_dim_instituicao)
 (has_collection_name "fato_PESSOA_PERIODO_INST")))

(def-instance att_fato_pessoa_seq_id_geografia db_attribute
 ((has_allowed_additive_summarization Summarization_Count_Distinct Summarization_Count)
 (has_attribute_name "SEQ_ID_GEOGRAFIA")
 (is_additive "true")))

(def-instance att_fato_pessoa_seq_id_pessoa db_attribute
 ((has_allowed_additive_summarization Summarization_Count_Distinct Summarization_Count)
 (has_attribute_name "SEQ_ID_PESSOA")
 (is_additive "true")))

(def-instance att_fato_pessoa_seq_id_inst db_attribute
 ((has_allowed_additive_summarization Summarization_Count_Distinct Summarization_Count)
 (has_attribute_name "SEQ_ID_INST")
 (is_additive "true")))

(def-instance att_fato_pessoa_cod_periodo db_attribute
 ((has_attribute_name "COD_PERIODO")
 (is_additive "false")))

(def-instance att_fato_pessoa_tot_fim_docentes db_attribute
 ((has_allowed_additive_summarization
  Summarization_Sum
  Summarization_Max
  Summarization_Min
  Summarization_Avg)
 (has_attribute_name "TOT_FIM_DOCENTES")
 (is_additive "true")))

(def-instance att_fato_pessoa_tot_fim_orientadores db_attribute
 ((has_allowed_additive_summarization
  Summarization_Sum
  Summarization_Max
  Summarization_Min
  Summarization_Avg)
 (has_attribute_name "TOT_FIM_ORIENTADORES")
 (is_additive "true")))

(def-instance att_fato_pessoa_tot_fim_pesquisadores db_attribute
 ((has_allowed_additive_summarization
  Summarization_Sum
  Summarization_Max
  Summarization_Min
  Summarization_Avg)
 (has_attribute_name "TOT_FIM_PESQUISADORES")
 (is_additive "true")))

(def-instance att_fato_pessoa_tot_fim_estudantes db_attribute
 ((has_allowed_additive_summarization
  Summarization_Sum
  Summarization_Max
  Summarization_Min
  Summarization_Avg)
 (has_attribute_name "TOT_FIM_ESTUDANTES")

```

```

(is_additive "true"))

(def-instance att_fato_pessoa_tot_fim_egressos db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_EGRESSOS")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_fim_extensao_univ db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_EXTENSAO_UNIV")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_fim_treinamento db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_TREINAMENTO")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_fim_outra_ativ db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_OUTRA_ATIV")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_funcionarios_lotados db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_FUNCIONARIOS_LOTADOS")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_funcionarios_vinculados db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_FIM_FUNCIONARIOS_VINCULADOS")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_meio_administradores db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_MEIO_ADMINISTRADORES")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_meio_estagio db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Max
    Summarization_Min
    Summarization_Avg)
  (has_attribute_name "TOT_MEIO_ESTAGIO")
  (is_additive "true"))

(def-instance att_fato_pessoa_tot_meio_serv_tecnico db_attribute
  (has_allowed_additive_summarization
    Summarization_Sum

```

```

        Summarization_Max
        Summarization_Min
        Summarization_Avg)
    (has_attribute_name "TOT_MEIO_SERV_TECNICO")
    (is_additive "true"))

```

```
;;; Fato_PESSOA Relationships definition
```

```

(def-instance dimension_usage_fato_pessoa_dim_geografia dimension_usage
  ((has_usage_dimension dimension_dim_geografia)
   (has_usage_collection_join db_collection_join_fato_pessoa_dim_geografia)))

(def-instance db_collection_join_fato_pessoa_dim_geografia db_collection_join
  ((has_coljoin_first_collection fact_fato_pessoa)
   (has_coljoin_first_collection_key att_fato_pessoa_seq_id_geografia)
   (has_coljoin_second_collection dimension_dim_geografia)
   (has_coljoin_second_collection_key att_dim_pessoa_seq_id_geografia)))

(def-instance dimension_usage_fato_pessoa_dim_pessoa dimension_usage
  ((has_usage_dimension dimension_dim_pessoa)
   (has_usage_collection_join db_collection_join_fato_pessoa_dim_pessoa)))

(def-instance db_collection_join_fato_pessoa_dim_pessoa db_collection_join
  ((has_coljoin_first_collection fact_fato_pessoa)
   (has_coljoin_first_collection_key att_fato_pessoa_seq_id_geografia)
   (has_coljoin_second_collection dimension_dim_pessoa)
   (has_coljoin_second_collection_key att_dim_pessoa_seq_id_pessoa)))

(def-instance dimension_usage_fato_pessoa_dim_instituicao dimension_usage
  ((has_usage_dimension dimension_dim_instituicao)
   (has_usage_collection_join db_collection_join_fato_pessoa_dim_instituicao)))

(def-instance db_collection_join_fato_pessoa_dim_instituicao db_collection_join
  ((has_coljoin_first_collection fact_fato_pessoa)
   (has_coljoin_first_collection_key att_fato_pessoa_seq_id_geografia)
   (has_coljoin_second_collection dimension_dim_instituicao)
   (has_coljoin_second_collection_key att_dim_pessoa_seq_id_instituicao)))

```

```
;;; Fato_RELAC_INSTITUCIONAL
```

```

(def-instance fact_fato_relac_institucional fact
  ((has_pk_attribute att_fato_fato_relac_institucional_seq_id_inst_1
                    att_fato_fato_relac_institucional_seq_id_inst_2
                    att_fato_fato_relac_institucional_seq_id_tempo)
   (has_fact_measure att_fato_fato_relac_institucional_sta_competidoras
                    att_fato_fato_relac_institucional_sta_parc_comerc
                    att_fato_fato_relac_institucional_sta_parc_academ)
   (has_fact_dimension_usage
    dimension_usage_fato_relac_institucional_dim_instituticao_1
    dimension_usage_fato_relac_institucional_dim_instituticao_2
    dimension_usage_fato_relac_institucional_dim_tempo)
   (has_collection_name "fato_RELAC_INSTTUCIONAL")))

(def-instance att_fato_relac_institucional_seq_id_inst_1 db_attribute
  ((has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
   (has_attribute_name "SEQ_ID_INST_1")
   (is_additive "true")))

(def-instance att_fato_relac_institucional_seq_id_inst_2 db_attribute
  ((has_allowed_additive_summarization
    Summarization_Count_Distinct
    Summarization_Count)
   (has_attribute_name "SEQ_ID_INST_2")
   (is_additive "true")))

(def-instance att_fato_relac_institucional_seq_id_tempo db_attribute
  ((has_attribute_name "SEQ_ID_TEMPO")
   (is_additive "false")))

(def-instance att_fato_relac_institucional_sta_competidoras db_attribute
  ((has_allowed_additive_summarization
    Summarization_Sum

```

```

        Summarization_Avg)
    (has_attribute_name "STA_COMPETIDORAS")
    (is_additive "true"))

(def-instance att_fato_relac_institucional_sta_parc_comerc db_attribute
  ((has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Avg)
  (has_attribute_name "STA_PARC_COMERC")
  (is_additive "true")))

(def-instance att_fato_relac_institucional_sta_parc_academ db_attribute
  ((has_allowed_additive_summarization
    Summarization_Sum
    Summarization_Avg)
  (has_attribute_name "STA_PARC_ACADEM")
  (is_additive "true")))

;;; Fato_RELAC_INSTITUCIONAL Relationships definition

(def-instance dimension_usage_fato_relac_institucional_dim_instituicao_1 dimension_usage
  ((has_usage_dimension dimension_dim_instituicao)
  (has_usage_collection_join db_collection_join_fato_instituicao_dim_instituicao_1)))

(def-instance db_collection_join_fato_relac_institucional_dim_instituicao_1
  db_collection_join
  ((has_coljoin_first_collection fact_fato_relac_institucional)
  (has_coljoin_first_collection_key att_fato_instituicao_seq_id_inst_1)
  (has_coljoin_second_collection dimension_dim_instituicao)
  (has_coljoin_second_collection_key att_dim_instituicao_seq_id_inst)))

(def-instance dimension_usage_fato_relac_institucional_dim_instituicao_2 dimension_usage
  ((has_usage_dimension dimension_dim_instituicao)
  (has_usage_collection_join db_collection_join_fato_instituicao_dim_instituicao_2)))

(def-instance db_collection_join_fato_relac_institucional_dim_instituicao_2
  db_collection_join
  ((has_coljoin_first_collection fact_fato_relac_institucional)
  (has_coljoin_first_collection_key att_fato_instituicao_seq_id_inst_2)
  (has_coljoin_second_collection dimension_dim_instituicao)
  (has_coljoin_second_collection_key att_dim_instituicao_seq_id_inst)))

(def-instance dimension_usage_fato_relac_institucional_dim_tempo dimension_usage
  ((has_usage_dimension dimension_dim_tempo)
  (has_usage_collection_join db_collection_join_fato_instituicao_dim_tempo)))

(def-instance db_collection_join_fato_relac_institucional_dim_tempo db_collection_join
  ((has_coljoin_first_collection fact_fato_relac_institucional)
  (has_coljoin_first_collection_key att_fato_instituicao_seq_id_tempo)
  (has_coljoin_second_collection dimension_dim_tempo)
  (has_coljoin_second_collection_key att_dim_tempo_seq_id_tempo)))

;;; Sample of Analysis

(def-instance groups_productivity Analysis
  ((has_analysis_description "Research Groups Publication Evolution")
  (has_analysis_content groups_productivity_content)
  (has_analysis_dimension groups_productivity_dim_1
    groups_productivity_dim_2)
  (has_analysis_filter groups_productivity_filter_1
    groups_productivity_filter_2
    groups_productivity_filter_3
    groups_productivity_filter_4)
  (has_analysis_parameter groups_productivity_parameter_1)
  (has_analysis_creator user_denilson)
  (has_analysis_allowed_user user_denilson)
  (has_analysis_allowed_role role_researcher)))

(def-instance groups_productivity_content Analysis_Content
  ((has_content_measure att_ft_producao_qtd_artigo_anais)
  (has_content_summarization summarization_sum)))

(def-instance groups_productivity_dim_1 Analysis_Dimension

```

```
((has_dimension_attribute att_di_instituicao_sigla)
 (has_dimension_position "1"))

(def-instance groups_productivity_dim_2 Analysis_Dimension
 ((has_dimension_attribute att_dim_grupo_nme_grupo)
 (has_dimension_position "2")))

(def-instance groups_productivity_filter_1 Analysis_Filter
 ((has_filter_attribute att_dim_grupo_seq_id_grupo)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_grupo)))

(def-instance groups_productivity_filter_2 Analysis_Filter
 ((has_filter_attribute att_di_instituicao_seq_id_inst)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_inst)))

(def-instance groups_productivity_filter_3 Analysis_Filter
 ((has_filter_attribute att_di_tempo_seq_id_tempo)
 (has_filter_operator db_operation_equal)
 (has_filter_value att_ft_producao_seq_id_tempo)))

(def-instance groups_productivity_filter_4 Analysis_Filter
 ((has_filter_attribute att_di_tempo_ano)
 (has_filter_operator db_operation_equal)
 (has_filter_value groups_productivity_parameter_1)))

(def-instance groups_productivity_parameter_1 Analysis_Parameter
 ((has_parameter_value "2002")
 (is_parameter_single_value "false")))
```

Apêndice B

(Definição dos construtores semânticos do modelo para composições baseado em fluxo de dados)

```
(in-ontology data-flow-composition)

(def-class composition-component ())

(def-class data-flow-orchestration (orchestration)
  ((has-components :type composition-component)))

(def-class control-component (composition-component)
  ;; The range of has-components is either 'service-component' or 'control-component'.
  ((has-components :type composition-component)))

(def-class if-then-else (control-component)
  ((has-condition :type string :cardinality 1)
   (has-then-binding :type output-binding :cardinality 1)
   (has-else-binding :type output-binding :max-cardinality 1)
  ))

(def-class service-component (composition-component)
  ((has-invocable-description :type invocable-description) ;; :cardinality 1
   (has-internal-bindings :type output-binding)
   (has-binding-to-composition-input :type GInvGInvMediator)
   (has-binding-to-composition-output :type GInvGInvMediator)
   (has-conditional-bindings :type if-then-else)
   (has-failure-handling-service :type invocable-description :max-cardinality 1)))

(def-class output-binding ()
  ((has-output-mediator :type GInvGInvMediator :cardinality 1)
   (to-component-service :type component-service :cardinality 1)
  ))

(def-class GInvGInvMediator ()
  ((has-source :type string)
   (has-target :type string :cardinality 1)
   (has-mediation-service :type invocable-description :cardinality 1)
   ;; mediation output specifies which output of the mediation service should be taken
   ;; and feed which input of which invocable.
   (has-mediation-output-role-name :type string)
   ;; mediation input specifies which output from which source invocable to be taken by
   ;; the orchestration engine and feed the input of the mediation service.
   (has-mediation-input-role-name :type string)
   ;; parameters are used to specialize a generic mediator
   (has-mediation-parameters :type string)))

(def-class invocable-description ()
  ((has-invocable-name :type string :cardinality 1)
   (has-ontology-name :type string :cardinality 1)
   (has-irs-server-url :type string :cardinality 1)))
```

Apêndice C

(Definição de uma composição para análise de indicadores de uma instituição)

```

#|  Orchestration definition  |#
(def-instance checking-university-indicators data-flow-orchestration
  ((has-components
    (class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ServiceComponent
      class kmi.irsIII.composition.dataFlowComposition.ControlComponent)
    )))

#|
  Definition of the first Service Component (find-critical-institution-indicators),
  including it's binding to the control component ifTheElse-1 and
  it's binding to the input values for the composition
|#
(def-instance service-component-find-critical-institution-indicators-goal
  service-component
  ((has-invocable-description find-critical-institution-indicators-goal-invocable)
    (has-internal-bindings (
      mapping-mediation-goal5-service-component-list-staff-level-formation-goal
      mapping-mediation-goal6-service-component-list-depto-critical-indicator-goal
      GInvGInvMediator ))
    (has-binding-to-composition-input mapping-mediation-goal7)
    (has-conditional-bindings mapping-mediation-goal8-ifThenElse-1)
  ))

#|
  Description of the goal defined in IRS-III that is going to be achieved
  by the first service component in order to retrieve the critical indicators
  from the university
|#
(def-instance find-critical-institution-indicators-goal-invocable invocable-description
  ((has-invocable-name find-critical-institution-indicators-goal)
    (has-ontology-name ontodss-goal)
    (has-irs-server-url "http://localhost:3000/soap")))

#|
  Definition of the mediator that will bind the value given as input to the
  composition (i.e. a instance of the concept university) to the service
  component find-critical-institution-indicators
|#
(def-instance mapping-mediation-goal7 GInvGInvMediator
  ((has-target find-critical-institution-indicators-goal)
    (has-mediation-service mapping-mediation-goal)
    (has-mediation-input-role-name input-value-1)
    (has-mediation-output-role-name has-input-crit-ind-institution)
    (has-mediation-parameters (input-value-1 has-input-crit-ind-institution null))))

#|
  Binding between the service component and the mediator that will connect
  the service component to the control component
|#
(def-instance mapping-mediation-goal8-ifThenElse-1 output-binding
  ((has-output-mediator mapping-mediation-goal8)
    (to-component-service ifThenElse-1)))

#|
  Definition of the mediator that will connect the output of the service component
  find-critical-institution-indicators (i.e. a instance of the concept indicator) to
  the input of the control component ifThenElse-1 (i.e. the variable "value")
  through a mediation service (i.e. the goal mapping-mediation)
|#
(def-instance mapping-mediation-goal8 GInvGInvMediator
  ((has-source find-critical-institution-indicators-goal)
    (has-target ifThenElse-1)
    (has-mediation-service mapping-mediation-goal)
    (has-mediation-input-role-name has-output-crit-ind-institution)
  ))

```

```

(has-mediation-output-role-name ?value)
(has-mediation-parameters (has-output-crit-ind-institution ?value null)))

#|
Definition of the mediators that will feed either list-staff-level-formation or
list-depto-critical-indicator with the university given as input to the composition
according to the Then and Else clauses of the control component
|#
(def-instance mapping-mediation-goal5-service-component-list-staff-level-formation-goal
output-binding
  ((has-output-mediator mapping-mediation-goal5)
   (to-component-service service-component-list-staff-level-formation-goal)))

(def-instance mapping-mediation-goal5 GInvGInvMediator
  ((has-source find-critical-institution-indicators-goal)
   (has-target list-staff-level-formation-goal)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-output-crit-ind-institution)
   (has-mediation-output-role-name has-staff-level-input-formation)
   (has-mediation-parameters (has-output-crit-ind-institution has-staff-level-input-
formation null))))

(def-instance mapping-mediation-goal6-service-component-list-depto-critical-indicator-
goal output-binding
  ((has-output-mediator mapping-mediation-goal6)
   (to-component-service service-component-list-depto-critical-indicator-goal)))

(def-instance mapping-mediation-goal6 GInvGInvMediator
  ((has-source find-critical-institution-indicators-goal)
   (has-target list-depto-critical-indicator-goal)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-output-crit-ind-institution)
   (has-mediation-output-role-name has-depto-critical-input-university)
   (has-mediation-parameters (has-output-crit-ind-institution has-depto-critical-input-
university null))))

#|
Definition of the control component ifThenElse-1 and it's condition
|#
(def-instance ifThenElse-1 if-then-else
  ((has-condition "ifThenElse-1-func")))

(def-function ifThenElse-1-func (?value )
  :body (the true (= ?value 'degree)))

#|
Definition of the second service component (list-staff-level-formation),
that will be achieved when the condition of the control component could
be proved. The output of this service will be given as an output of the
composition through the mediator mapping-mediation-goal10
|#
(def-instance service-component-list-staff-level-formation-goal service-component
  ((has-invocable-description list-staff-level-formation-goal-invocable)
   (has-binding-to-composition-output mapping-mediation-goal10)))

(def-instance list-staff-level-formation-goal-invocable invocable-description
  ((has-invocable-name list-staff-level-formation-goal)
   (has-ontology-name ontodss-goal)
   (has-irs-server-url "http://localhost:3000/soap")))

(def-instance mapping-mediation-goal10 GInvGInvMediator
  ((has-source list-staff-level-formation-goal)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-staff-level-output)
   (has-mediation-output-role-name has-staff-level-output)
   (has-mediation-parameters (has-staff-level-output has-staff-level-output null))))

#|
Definition of the third Service Component (list-depto-critical-indicator),
that will be achieved when the condition of the control component could not
be proved. The output of this service will be connect to the input of the
fourth service, the compare-institution-productivity service component
|#

```

```

(def-instance service-component-list-depto-critical-indicator-goal service-component
  ((has-invocable-description list-depto-critical-indicator-goal-invocable)
   (has-internal-bindings
    mapping-mediation-goall1-service-component-compare-institution-productivity-goal )
   (has-binding-to-composition-output mapping-mediation-goall0)))

(def-instance list-depto-critical-indicator-goal-invocable invocable-description
  ((has-invocable-name list-depto-critical-indicator-goal)
   (has-ontology-name ontodss-goal)
   (has-irs-server-url "http://localhost:3000/soap")))

#|
  Definition of the mediator that will connect the output of the third service
  to the input of the fourth service
|#
(def-instance mapping-mediation-goall1-service-component-compare-institution-
productivity-goal output-binding
  ((has-output-mediator mapping-mediation-goall1)
   (to-component-service service-component-compare-institution-productivity-goal)))

(def-instance mapping-mediation-goall1 GInvGInvMediator
  ((has-source list-depto-critical-indicator-goal)
   (has-target compare-institution-productivity-goal)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-output-depto)
   (has-mediation-output-role-name has-input-comp-institution)
   (has-mediation-parameters (has-output-depto has-input-comp-institution null))))

(def-instance mapping-mediation-goall1-service-component-compare-institution-
productivity-goal output-binding
  ((has-output-mediator mapping-mediation-goall1)
   (to-component-service service-component-compare-institution-productivity-goal)))

#|
  Definition of the fourth Service Component (compare-institution-productivity) and
  it's mediator to the output of the composition
|#
(def-instance service-component-compare-institution-productivity-goal service-component
  ((has-invocable-description compare-institution-productivity-goal-invocable)
   (has-binding-to-composition-output mapping-mediation-goall2)))

(def-instance compare-institution-productivity-goal-invocable invocable-description
  ((has-invocable-name compare-institution-productivity-goal)
   (has-ontology-name ontodss-goal)
   (has-irs-server-url "http://localhost:3000/soap")))

(def-instance mapping-mediation-goall2 GInvGInvMediator
  ((has-source compare-institution-productivity-goal)
   (has-mediation-service mapping-mediation-goal)
   (has-mediation-input-role-name has-output-comp-university)
   (has-mediation-output-role-name has-output-comp-university)
   (has-mediation-parameters (has-output-comp-university has-output-comp-university
null))))

```

Apêndice D

(Construtores criados na Ontologia de Serviços para os serviços utilizados no estudo de caso)

```

(def-class extract-xml-cv-goal
  (goal)
  ?goal
  ((has-input-role :value has-cv)
   (has-input-soap-binding :value (has-cv "sexpr"))
   (has-output-role :value has-xml-cv)
   (has-output-soap-binding :value (has-xml-cv "xml"))
   (has-cv :type cv)
   (has-xml-cv :type string)))

(def-class extract-xml-cv-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    extract-xml-cv-web-service-capability)
   (has-interface :value extract-xml-cv-web-service-interface)
   (has-non-functional-properties
    :value
    extract-xml-cv-web-service-non-func-props)))

(def-class extract-xml-cv-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value extract-xml-cv-mediator)
   (has-non-functional-properties
    :value
    extract-xml-cv-web-service-non-func-props)))

(def-class extract-xml-cv-web-service-interface (interface)
  ((has-orchestration :value extract-xml-cv-web-service-orchestration)
   (has-choreography :value extract-xml-cv-web-service-choreography)))

(def-class extract-xml-cv-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value extract-xml-cv-goal)
   (has-target-component :value extract-xml-cv-web-service)))

(def-class list-productions-cv-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-productions-cv-mediator)
   (has-non-functional-properties
    :value
    list-productions-cv-web-service-non-func-props)))

(def-class list-productions-cv-web-service-interface (interface)
  ((has-orchestration :value list-productions-cv-web-service-orchestration)
   (has-choreography :value list-productions-cv-web-service-choreography)))

(def-class list-project-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-project-mediator)
   (has-non-functional-properties
    :value
    list-project-web-service-non-func-props)))

(def-class list-project-web-service-interface
  (interface)
  ?interface
  ((has-choreography
    :value
    list-project-web-service-choreography)
   (has-orchestration
    :value
    list-project-web-service-orchestration)))

```

```

        :value
        list-project-web-service-orchestration)))

(def-class list-project-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-projects)))

(def-class list-project-goal
  (goal)
  ?goal
  ((has-input-role :value has-research-group)
   (has-input-soap-binding
    :value
    (has-research-group "sexpr")))
  (has-output-role :value has-project)
  (has-output-soap-binding :value (has-project "sexpr"))
  (has-research-group :type research_group)
  (has-project :type project)))

(def-class list-project-web-service
  (web-service)
  ?web-service
  ((has-capability :value list-project-web-service-capability)
   (has-interface :value list-project-web-service-interface)
   (has-non-functional-properties
    :value
    list-project-web-service-non-func-props)))

(def-class list-project-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value list-project-goal)
   (has-target-component :value list-project-web-service)))

(def-class list-institution-goal
  (goal)
  ?goal
  ((has-input-role :value has-institution)
   (has-input-soap-binding :value (has-institution "sexpr"))
   (has-output-role :value has-list-institution)
   (has-output-soap-binding
    :value
    (has-list-institution "xml")))
  (has-institution :type institution)
  (has-list-institution :type string)))

(def-class list-institution-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    list-institution-web-service-capability)
   (has-interface
    :value
    list-institution-web-service-interface)
   (has-non-functional-properties
    :value
    list-institution-web-service-non-func-props)))

(def-class list-institution-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-institution-mediator)
   (has-non-functional-properties
    :value
    list-institution-web-service-non-func-props)))

(def-class list-institution-web-service-interface (interface)
  ((has-orchestration :value list-institution-web-service-orchestration)
   (has-choreography :value list-institution-web-service-choreography)))

(def-class list-institution-mediator
  (wg-mediator)

```

```

?mediator
  ((has-source-component :value list-institution-goal)
   (has-target-component :value list-institution-web-service)))

(def-class list-institution-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-institution)))

(def-class extract-xml-institution-goal
  (goal)
  ?goal
  ((has-input-role :value has-institution)
   (has-input-soap-binding :value (has-institution "sexpr"))
   (has-output-role :value has-xml-institution)
   (has-output-soap-binding
    :value
    (has-xml-institution "xml")))
  (has-institution :type institution)
  (has-xml-institution :type string)))

(def-class extract-xml-institution-web-service
  (web-service)
  ?web-service
  ((has-capability
   :value
   extract-xml-institution-web-service-capability)
   (has-interface
   :value
   extract-xml-institution-web-service-interface)
   (has-non-functional-properties
   :value
   extract-xml-institution-web-service-non-func-props)))

(def-class extract-xml-institution-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value extract-xml-institution-mediator)
   (has-non-functional-properties
   :value
   extract-xml-institution-web-service-non-func-props)))

(def-class extract-xml-institution-web-service-interface (interface)
  ((has-orchestration :value extract-xml-institution-web-service-orchestration)
   (has-choreography :value extract-xml-institution-web-service-choreography)))

(def-class extract-xml-institution-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value extract-xml-institution-goal)
   (has-target-component
   :value
   extract-xml-institution-web-service)))

(def-class extract-xml-institution-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value extract-xml-institution)))

(def-class extract-xml-cv-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value extract-xml-researcher)))

(def-class extract-xml-group-goal
  (goal)
  ?goal
  ((has-input-role :value has-group)
   (has-input-soap-binding :value (has-group "sexpr"))
   (has-output-role :value has-xml-group)
   (has-output-soap-binding :value (has-xml-group "xml")))

```

```

      (has-group :type research_group)
      (has-xml-group :type string)))

(def-class extract-xml-group-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    extract-xml-group-web-service-capability)
   (has-interface
    :value
    extract-xml-group-web-service-interface)
   (has-non-functional-properties
    :value
    extract-xml-group-web-service-non-func-props)))

(def-class extract-xml-group-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value extract-xml-group-mediator)
   (has-non-functional-properties
    :value
    extract-xml-group-web-service-non-func-props)))

(def-class extract-xml-group-web-service-interface (interface)
  ((has-orchestration :value extract-xml-group-web-service-orchestration)
   (has-choreography :value extract-xml-group-web-service-choreography)))

(def-class extract-xml-group-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value extract-xml-group-goal)
   (has-target-component
    :value
    extract-xml-group-web-service)))

(def-class extract-xml-group-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value extract-xml-group)))

(def-class list-productions-cv-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-production)))

(def-class list-researcher-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-researcher-mediator)
   (has-non-functional-properties
    :value
    list-researcher-web-service-non-func-props)))

(def-class list-researcher-web-service-interface
  (interface)
  ?interface
  ((has-choreography
    :value
    list-researcher-web-service-choreography)
   (has-orchestration
    :value
    list-researcher-web-service-orchestration)))

(def-class list-researcher-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-researcher)))

```

```

(def-class list-group-goal
  (goal)
  ?goal
  ((has-input-role :value has-group)
   (has-input-soap-binding :value (has-group "sexpr"))
   (has-output-role :value has-list-group)
   (has-output-soap-binding :value (has-list-group "sexpr"))
   (has-group :type research_group)
   (has-list-group :type string)))

(def-class list-group-web-service
  (web-service)
  ?web-service
  ((has-capability :value list-group-web-service-capability)
   (has-interface :value list-group-web-service-interface)
   (has-non-functional-properties
    :value
    list-group-web-service-non-func-props)))

(def-class list-group-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-group-mediator)
   (has-non-functional-properties
    :value
    list-group-web-service-non-func-props)))

(def-class list-group-web-service-interface (interface)
  ((has-orchestration :value list-group-web-service-orchestration)
   (has-choreography :value list-group-web-service-choreography)))

(def-class list-group-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value list-group-goal)
   (has-target-component :value list-group-web-service)))

(def-class list-group-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-group)))

(def-class list-production-goal
  (goal)
  ?goal
  ((has-input-role :value has-cv)
   (has-input-soap-binding :value (has-cv "sexpr"))
   (has-output-role :value has-list-production)
   (has-output-soap-binding
    :value
    (has-list-production "sexpr"))
   (has-cv :type cv)
   (has-list-production :type string)))

(def-class list-production-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    list-production-web-service-capability)
   (has-interface
    :value
    list-production-web-service-interface)
   (has-non-functional-properties
    :value
    list-production-web-service-non-func-props)))

(def-class list-production-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-production-mediator)
   (has-non-functional-properties
    :value
    list-production-web-service-non-func-props)))

```

```

(def-class list-production-web-service-interface (interface)
  ((has-orchestration :value list-production-web-service-orchestration)
   (has-choreography :value list-production-web-service-choreography)))

(def-class list-production-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value list-production-goal)
   (has-target-component :value list-production-web-service)))

(def-class list-production-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-production)))

(def-class list-researcher-goal
  (goal)
  ?goal
  ((has-input-role :value has-research-group)
   (has-input-soap-binding
    :value
    (has-research-group "sexpr")))
  (has-output-role :value has-list-cv)
  (has-output-soap-binding :value (has-list-cv "sexpr"))
  (has-research-group :type research_group)
  (has-list-cv :type cv)))

(def-class list-researcher-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    list-researcher-web-service-capability)
   (has-interface
    :value
    list-researcher-web-service-interface)
   (has-non-functional-properties
    :value
    list-researcher-web-service-non-func-props)))

(def-class list-researcher-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value list-researcher-goal)
   (has-target-component :value list-researcher-web-service)))

(def-class list-group-institution-web-service-capability
  (capability)
  ?capability
  ((has-non-functional-properties
    :value
    list-group-institution-web-service-non-func-props)))

(def-class list-group-institution-web-service-interface
  (interface)
  ?interface
  ((has-choreography
    :value
    list-group-institution-web-service-choreography)
   (has-orchestration
    :value
    list-group-institution-web-service-orchestration)))

(def-class list-groups-institution-goal
  (goal)
  ?goal
  ((has-input-role :value has-institution)
   (has-input-soap-binding :value (has-institution "sexpr"))
   (has-output-role :value has-research-group)
   (has-output-soap-binding
    :value
    (has-research-group "sexpr")))
  (has-institution :type institution))

```

```

        (has-research-group :type research_group)))

(def-class list-groups-institution-web-service
  (web-service)
  ?web-service
  ((has-capability
    :value
    list-groups-institution-web-service-capability)
   (has-interface
    :value
    list-groups-institution-web-service-interface)
   (has-non-functional-properties
    :value
    list-groups-institution-web-service-non-func-props)))

(def-class list-groups-institution-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-groups-institution-mediator)
   (has-non-functional-properties
    :value
    list-groups-institution-web-service-non-func-props)))

(def-class list-groups-institution-web-service-interface (interface)
  ((has-choreography :value list-groups-institution-web-service-choreography)))

(def-class list-groups-institution-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value list-groups-institution-goal)
   (has-target-component
    :value
    list-groups-institution-web-service)))

(def-class list-groups-institution-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-group-institution)))

(def-class search-institution-goal
  (goal)
  ?goal
  ((has-input-role :value has-search-arguments)
   (has-input-soap-binding
    :value
    (has-search-arguments "sexpr")))
  (has-output-role :value has-institution)
  (has-output-soap-binding :value (has-institution "sexpr"))
  (has-search-arguments :type string)
  (has-institution :type institution)))

(def-class search-knowledge-area-goal
  (goal)
  ?goal
  ((has-input-role :value has-ka-search-arguments)
   (has-input-soap-binding
    :value
    (has-ka-search-arguments "sexpr")))
  (has-output-role :value has-knowledge-area)
  (has-output-soap-binding
    :value
    (has-knowledge-area "sexpr"))
  (has-ka-search-arguments :type string)
  (has-knowledge-area :type knowledge_area)))

(def-class list-groups-institution-area-goal
  (goal)
  ?goal
  ((has-input-role
    :value
    has-institution
    :value
    has-knowledge-area)
   (has-input-soap-binding
    :value

```

```

      (has-institution "sexpr")
      :value
      (has-knowledge-area "sexpr"))
    (has-output-role :value has-research-group)
    (has-output-soap-binding
     :value
     (has-research-group "sexpr"))
    (has-institution :type institution)
    (has-knowledge-area :type knowledge-area)
    (has-research-group :type research_group)))
(def-class find-critical-indicators-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value find-critical-indicators-mediator)
   (has-non-functional-properties
    :value
    find-critical-indicators-web-service-non-func-props)))

(def-class find-critical-indicators-web-service-interface (interface)
  ((has-choreography :value find-critical-indicators-web-service-choreography63324)))

(def-class find-critical-indicators-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-critical-indicator)))

(def-class list-depto-critical-indicator-goal
  (goal)
  ?goal
  ((has-input-role
   :value
   has-depto-critical-input-university
   :value
   has-depto-critical-input-indicator)
   (has-input-soap-binding
    :value
    (has-depto-critical-input-university "sexpr")
    :value
    (has-depto-critical-input-indicator "sexpr")))
   (has-output-role :value has-output-depto)
   (has-output-soap-binding :value (has-output-depto "sexpr"))
   (has-depto-critical-input-university :type institution)
   (has-depto-critical-input-indicator :type indicator)
   (has-output-depto :type department)))

(def-class list-depto-critical-indicator-web-service-capability
  (capability)
  ?capability
  ((used-mediator
   :value
   list-depto-critical-indicator-mediator)
   (has-non-functional-properties
    :value
    list-depto-critical-indicator-web-service-non-func-props)))

(def-class list-depto-critical-indicator-web-service-interface (interface)
  ((has-orchestration :value list-depto-critical-indicator-web-service-orchestration)
   (has-choreography :value list-depto-critical-indicator-web-service-choreography)))

(def-class list-depto-critical-indicator-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-depto-critical-indicator)))

(def-class find-critical-indicators-web-service-choreography63324
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-critical-indicator)))

(def-class list-depto-critical-indicator-mediator2

```

```

(wg-mediator)
?mediator
((has-source-component
 :value
 list-depto-critical-indicator-goal)
 (has-target-component
 :value
 list-depto-critical-indicator-web-service2)))

(def-class list-depto-critical-indicator-web-service2
 (web-service)
 ?web-service
 ((has-capability
 :value
 list-depto-critical-indicator-web-service2-capability)
 (has-interface
 :value
 list-depto-critical-indicator-web-service2-interface)
 (has-non-functional-properties
 :value
 list-depto-critical-indicator-web-service2-non-func-props)))

(def-class list-depto-critical-indicator-web-service2-capability
 (capability)
 ?capability
 ((used-mediator
 :value
 list-depto-critical-indicator-mediator2)
 (has-non-functional-properties
 :value
 list-depto-critical-indicator-web-service2-non-func-props)))

(def-class list-depto-critical-indicator-web-service2-interface (interface)
 ((has-orchestration :value list-depto-critical-indicator-web-service2-orchestration)
 (has-choreography :value list-depto-critical-indicator-web-service2-choreography)))

(def-class list-depto-critical-indicator-web-service2-choreography
 (choreography)
 ((has-web-service-host :value "127.0.0.1")
 (has-web-service-port :value 3001)
 (has-web-service-location :value "/soap")
 (has-lisp-function :value list-depto-critical-indicator)))

(def-class list-staff-level-formation-goal
 (goal)
 ?goal
 ((has-input-role
 :value
 has-staff-level-input-university
 :value
 has-staff-level-input-formation)
 (has-input-soap-binding
 :value
 (has-staff-level-input-university "sexpr")
 :value
 (has-staff-level-input-formation "sexpr"))
 (has-output-role :value has-staff-level-output)
 (has-output-soap-binding
 :value
 (has-staff-level-output "sexpr")))
 (has-staff-level-input-university :type institution)
 (has-staff-level-input-formation :type string)
 (has-staff-level-output :type cv)))

(def-class list-staff-level-formation-mediator
 (wg-mediator)
 ?mediator
 ((has-source-component
 :value
 list-staff-level-formation-goal)
 (has-target-component
 :value
 list-staff-level-formation-web-service)))

(def-class list-staff-level-formation-web-service
 (web-service)
 ?web-service

```

```

    (has-capability
     :value
     list-staff-level-formation-web-service-capability)
    (has-interface
     :value
     list-staff-level-formation-web-service-interface)
    (has-non-functional-properties
     :value
     list-staff-level-formation-web-service-non-func-props)))

(def-class list-staff-level-formation-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value list-staff-level-formation-mediator)
   (has-non-functional-properties
    :value
    list-staff-level-formation-web-service-non-func-props)))

(def-class list-staff-level-formation-web-service-interface (interface)
  ((has-orchestration :value list-staff-level-formation-web-service-orchestration)
   (has-choreography :value list-staff-level-formation-web-service-choreography)))

(def-class list-staff-level-formation-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-staff-by-formation)))

(def-class compare-dept-productivity-web-service-capability
  (capability)
  ?capability
  ((used-mediator :value compare-dept-productivity-mediator)
   (has-non-functional-properties
    :value
    compare-dept-productivity-web-service-non-func-props)))

(def-class compare-dept-productivity-web-service-interface (interface)
  ((has-orchestration :value compare-dept-productivity-web-service-orchestration)
   (has-choreography :value compare-dept-productivity-web-service-choreography)))

(def-class compare-dept-productivity-web-service-choreography
  (choreography)
  ((has-web-service-host :value "137.108.24.239")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value compare-depto-productivity)))

(def-class compare-institution-productivity-goal
  (goal)
  ?goal
  ((has-input-role :value has-input-comp-institution)
   (has-input-soap-binding
    :value
    (has-input-comp-institution "sexpr")))
  (has-output-role :value has-output-comp-university)
  (has-output-soap-binding
   :value
   (has-output-comp-university "sexpr")))
  (has-input-comp-institution :type institution)
  (has-output-comp-university :type string)))

(def-class compare-institution-productivity-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component
   :value
   compare-institution-productivity-goal)
   (has-target-component
   :value
   compare-institution-productivity-web-service)))

(def-class compare-institution-productivity-web-service
  (web-service)
  ?web-service
  ((has-capability
   :value

```

```

        compare-institution-productivity-web-service-capability)
      (has-interface
       :value
       compare-institution-productivity-web-service-interface)
      (has-non-functional-properties
       :value
       compare-institution-productivity-web-service-non-func-props)))

(def-class compare-institution-productivity-web-service-capability
  (capability)
  ?capability
  ((used-mediator
   :value
   compare-institution-productivity-mediator)
   (has-non-functional-properties
    :value
    compare-institution-productivity-web-service-non-func-props)))

(def-class compare-institution-productivity-web-service-interface (interface)
  ((has-orchestration :value compare-institution-productivity-web-service-orchestration)
   (has-choreography :value compare-institution-productivity-web-service-choreography)))

(def-class compare-institution-productivity-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function
    :value
    compare-institution-productivity)))

(def-class find-critical-institution-indicators-goal
  (goal)
  ?goal
  ((has-input-role :value has-input-crit-ind-institution)
   (has-input-soap-binding
    :value
    (has-input-crit-ind-institution "sexpr"))
   (has-output-role :value has-output-crit-ind-institution)
   (has-output-soap-binding
    :value
    (has-output-crit-ind-institution "sexpr"))
   (has-input-crit-ind-institution :type institution)
   (has-output-crit-ind-institution :type indicator)))

(def-class find-critical-institution-indicators-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component
   :value
   find-critical-institution-indicators-goal)
   (has-target-component
    :value
    find-critical-institution-indicators-web-service)))

(def-class find-critical-institution-indicators-web-service
  (web-service)
  ?web-service
  ((has-capability
   :value
   find-critical-institution-indicators-web-service-capability)
   (has-interface
    :value
    find-critical-institution-indicators-web-service-interface)
   (has-non-functional-properties
    :value
    find-critical-institution-indicators-web-service-non-func-props)))

(def-class find-critical-institution-indicators-web-service-capability
  (capability)
  ?capability
  ((used-mediator
   :value
   find-critical-institution-indicators-mediator)
   (has-non-functional-properties
    :value
    find-critical-institution-indicators-web-service-non-func-props)))

```

```

(def-class find-critical-institution-indicators-web-service-interface (interface)
  ((has-orchestration :value find-critical-institution-indicators-web-service-orchestration)
   (has-choreography :value find-critical-institution-indicators-web-service-choreography)))

(def-class find-critical-institution-indicators-web-service-choreography
  (choreography)
  ((has-web-service-host :value "127.0.0.1")
   (has-web-service-port :value 3001)
   (has-web-service-location :value "/soap")
   (has-lisp-function :value list-critical-indicator)))

(def-class ranking-goal
  (goal)
  ?goal
  ((has-input-role :value input-content)
   (has-input-soap-binding :value (input-content "sexpr"))
   (has-output-role :value output-content)
   (has-output-soap-binding :value (output-content "sexpr"))
   (input-content :type string)
   (output-content :type string)))

(def-class ranking-ws
  (web-service)
  ?web-service
  ((has-capability :value ranking-ws-capability)
   (has-interface :value ranking-ws-interface)
   (has-non-functional-properties
    :value
    ranking-ws-non-func-props)))

(def-class ranking-ws-capability
  (capability)
  ?capability
  ((used-mediator :value ranking-mediator)
   (has-non-functional-properties
    :value
    ranking-ws-non-func-props)))

(def-class ranking-ws-interface (interface)
  ((has-choreography :value ranking-ws-choreography)))

(def-class ranking-mediator
  (wg-mediator)
  ?mediator
  ((has-source-component :value ranking-goal)
   (has-target-component :value ranking-ws)))

(def-class ranking-ws-choreography
  (choreography)
  ((has-web-service-host :value "192.168.1.72")
   (has-web-service-port :value 8080)
   (has-web-service-location
    :value
    "/irspublisher/wsdlinvoker")
   (has-message-exchange-pattern :value "nil")
   (has-wsdl-file-name
    :value
    "http://localhost:8080/axis/ranking.jws?wsdl")
   (has-wsdl-operation-name :value "getrankedlist")
   (has-wsdl-port-type :value "ranking")
   (has-wsdl-port-type-name-space
    :value
    "http://localhost:8080/axis/ranking.jws")
   (has-wsdl-web-service-provider-type :value "axis")))

```

Apêndice E

(Construtores criados na Ontologia do Domínio para o estudo de caso)

```

(def-class institution ()
  ((is_institution_located_at_city :type city)
   (is_institution_located_at_state :type state)
   (has_institution_name :type string)
   (has_institution_acronym :type string)
   (has_institution_code :type string)
   (has_institution_parent :type institution)
   (has_institution_id :type string :cardinality 1)))

(def-class country ()
  ((has_country_name :type string)
   (has_country_acronym :type string)))

(def-class month ()
  ((has_month_name :type string)
   (has_month_number :type string)))

(def-class quarter ()
  ((has_quarter_number :type string)))

(def-class year ()
  ((has_year_number :type string)))
  (def-class person ()
    ((has_person_id :type string :cardinality 1)
     (has_person_name :type string)
     (has_person_gender :type string)
     (has_person_age :type string)
     (has_person_research_area :type knowledge_area)
     (has_person_institutional_address :type institution)))

(def-class degree ()
  ((has_degree_id :type string :cardinality 1)
   (has_degree_person :type person)
   (has_degree_level :type string)
   (has_degree_course :type course)
   (has_degree_start_year :type year)
   (has_degree_end_year :type year)
   (is_degree_completed :type string)
   (has_degree_institution :type institution)))

(def-class activity ()
  ((has_activity_id :type string :cardinality 1)
   (has_activity_type :type string)
   (has_activity_subtype :type string)
   (has_activity_person :type person)
   (has_activity_course :type course)
   (has_activity_start_year :type year)
   (has_activity_start_month :type month)
   (has_activity_end_year :type year)
   (has_activity_end_month :type month)
   (has_activity_institution :type institution)))

(def-class course ()
  ((has_course_name :type string)
   (has_course_level :type string)
   (has_course_institution :type institution)
   (has_course_knowledge_area :type knowledge_area)))

(def-class knowledge_area ()
  ((has_knowledge_area_id :type string)
   (has_knowledge_area_code :type string)
   (has_knowledge_area_name :type string)
   (has_knowledge_area_level :type string)
   (has_knowledge_area_parent :type knowledge_area)))

(def-class research_group ()
  ((is_group_located_at_institution :type institution)
   (has_group_leader :type person)
   (has_group_team :type person)
   (has_group_creation_year :type string)
   (has_group_name :type string))

```

```

(has_group_code :type string)
(has_group_id :type string))

(def-class project ()
  ((has_project_name :type string)
   (has_project_group_worker :type research_group)))

(def-class city ()
  ((is_city_located_at_state :type state)
   (has_city_id :type string)
   (has_city_name :type string)))

(def-class region ()
  ((has_region_acronym :type string :cardinality 1)
   (has_region_name :type string)))

(def-class state ()
  ((is_state_located_at_region :type region)
   (is_state_located_at_country :type country)
   (has_state_acronym :type string :cardinality 1)
   (has_state_name :type string)))

(def-relation sf-institution-competitor-city (?i1 ?i2)
  "It is sufficient that both institutions are in the same city"
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (city ?c)
    (is_institution_located_at_city ?i1 ?c)
    (is_institution_located_at_city ?i2 ?c)
    (not (= ?i1 ?i2))))

(def-relation sf-institution-competitor-state (?i1 ?i2)
  "It is sufficient that both institutions are in the same state"
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (state ?s)
    (is_institution_located_at_state ?i1 ?s)
    (is_institution_located_at_state ?i2 ?s)
    (not (= ?i1 ?i2))))

(def-relation sf-institution-competitor-area (?i1 ?i2)
  "It is sufficient that both institutions are in the same city and offer courses in the
  same area"
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (city ?c)
    (knowledge_area ?k)
    (is_institution_located_at_city ?i1 ?c)
    (has_course_institution ?co1 ?i1)
    (has_course_knowledge_area ?co1 ?k)
    (is_institution_located_at_city ?i2 ?c)
    (has_course_institution ?co2 ?i2)
    (has_course_knowledge_area ?co2 ?k)
    (not (= ?i1 ?i2))))

(def-relation sf-institution-partner (?i1 ?i2)
  "It is sufficient that both institutions have groups working in the same project"
  :constraint (and (institution ?i1)(institution ?i2))
  :sufficient (and
    (project ?p)
    (has_group_project ?g1 ?p)
    (is_group_located_at ?g1 ?i1)
    (has_group_project ?g2 ?p)
    (is_group_located_at ?g2 ?i2)
    (not (= ?i1 ?i2))))

(def-relation sf-institution-enrolled-worker (?i ?p)
  :constraint (and (institution ?i)(person ?p))
  :sufficient
  (and
    (has_person_institutional_address ?p ?i)
    (has_activity_person ?act ?p)
    (has_activity_institution ?act ?i)))

(def-relation sf-institution-non-enrolled-worker (?i ?p)
  :constraint (and (institution ?i)(person ?p))
  :sufficient

```

```

(and
  (has_activity_person ?act ?p)
  (has_activity_institution ?act ?i)
  (not (has_person_institutional_address ?p ?i))))

(def-relation sf-institution-enrolled-teacher (?i ?p)
:constraint (and (institution ?i) (person ?p))
:sufficient
  (and
    (has_person_institutional_address ?p ?i)
    (has_activity_person ?act ?p)
    (has_activity_type ?act "ENSINO")
    (has_activity_institution ?act ?i)))

(def-relation sf-institution-non-enrolled-teacher (?i ?p)
:constraint (and (institution ?i) (person ?p))
:sufficient
  (and
    (has_activity_person ?act ?p)
    (has_activity_institution ?act ?i)
    (has_activity_type ?act "ENSINO")
    (not (has_person_institutional_address ?p ?i))))

(def-relation sf-research_group-partner (?g1 ?g2)
:sufficient (and
  (project ?p)
  (has_project_group_worker ?p ?g1)
  (has_project_group_worker ?p ?g2)
  (not (= ?g1 ?g2))))

(def-relation sf-institution-researcher (?p)
:sufficient (and
  (has_degree_person ?d ?p)
  (is_degree_completed ?d "S")
  (has_degree_level ?d "DOCTORADO")))

(def-relation sf-institution-student (?p)
:sufficient (and
  (person ?p)
  (not (researcher ?p))))

(def-relation sf-institution-alumnus (?i ?p)
:constraint (and (person ?p) (university ?i))
:sufficient (and
  (has_degree_person ?d ?p)
  (is_degree_completed ?d "S")
  (has_degree_institution ?d ?i)))

(def-relation sf-institution-postgraduation-teacher (?i ?p)
:constraint (and (institution ?i) (person ?p))
:sufficient (and
  (has_activity_person ?act ?p)
  (has_activity_institution ?act ?i)
  (has_activity_type ?act "ENSINO")
  (has_activity_subtype ?act "POS-GRADUACAO")))

(def-relation sf-institution-graduation-teacher (?i ?p)
:constraint (and (institution ?i) (person ?p))
:sufficient (and
  (has_activity_person ?act ?p)
  (has_activity_institution ?act ?i)
  (has_activity_type ?act "ENSINO")
  (has_activity_subtype ?act "GRADUACAO")))

(def-relation sf-institution-teacher (?i ?p)
:constraint (and (institution ?i) (person ?p))
:sufficient (and
  (has_activity_person ?act ?p)
  (has_activity_institution ?act ?i)
  (has_activity_type ?act "ENSINO")))

(def-class indicator ())

```

Apêndice F

(Modelo de dados do *data mart* criado no estudo de caso)

