



**ARQUITETURA DE UM AMBIENTE COLABORATIVO DE
BUSINESS INTELLIGENCE BASEADO EM UM
REPOSITÓRIO DE ONTOLOGIAS E SERVIÇOS DE DADOS**

VALÉRIO AYMORÉ MARTINS

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ARQUITETURA DE UM AMBIENTE COLABORATIVO DE
BUSINESS INTELLIGENCE BASEADO EM UM
REPOSITÓRIO DE ONTOLOGIAS E SERVIÇOS DE DADOS**

VALÉRIO AYMORÉ MARTINS

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGEE.DM-500/2012

BRASÍLIA / DF, 09 de julho de 2012.

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ARQUITETURA DE UM AMBIENTE COLABORATIVO
DE BUSINESS INTELLIGENCE BASEADO EM UM
REPOSITÓRIO DE ONTOLOGIAS E SERVIÇOS DE DADOS**

VALÉRIO AYMORÉ MARTINS

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

**RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB
(ORIENTADOR)**

**JOÃO PAULO CARVALHO LUSTOSA DA COSTA, Dr., ENE/UNB
(EXAMINADOR INTERNO)**

**ROBSON DE OLIVEIRA ALBUQUERQUE, Dr., ABIN
(EXAMINADOR EXTERNO)**

BRASÍLIA, 09 DE JULHO DE 2012.

FICHA CATALOGRÁFICA

MARTINS, VALÉRIO AYMORÉ

Arquitetura de um Ambiente Colaborativo de Business Intelligence Baseado em um Repositório de Ontologias e Serviços de Dados. [Distrito Federal] 2012, xiv, 166p, 297 mm (ENE/FT/UnB, MESTRE, Engenharia Elétrica, 2012).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Business Intelligence

2. Sistemas Distribuídos

3. Ontologia

4. Serviços de Dados

I. ENE/FT/UnB.

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

MARTINS, Valério Aymoré (2012). Arquitetura de um Ambiente Colaborativo de Business Intelligence Baseado em um Repositório de Ontologias e Serviços de Dados. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEE.DM-500/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 166p.

CESSÃO DE DIREITOS

AUTOR: Valério Aymoré Martins

TÍTULO: ARQUITETURA DE UM AMBIENTE COLABORATIVO DE BUSINESS INTELLIGENCE BASEADO EM UM REPOSITÓRIO DE ONTOLOGIAS E SERVIÇOS DE DADOS.

GRAU: Mestre

ANO: 2012

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

VALÉRIO AYMORÉ MARTINS
UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica
70910-900
Brasília, DF - BRASIL

AGRADECIMENTOS

Ao orientador e amigo professor Dr. Rafael Timóteo Sousa Júnior, que me orientou de forma profissional nas horas mais complicadas durante a criação deste trabalho e que suportou pacientemente tantas dúvidas e problemas relativos ao assunto e outros pequenos detalhes pertinentes a esta dissertação.

Aos professores e amigos Fábio Lúcio Lopes de Mendonça e João Paulo Carvalho Lustosa da Costa pelas grandes dicas e ajudas e pelo constante apoio, incentivo, dedicação e amizade, essenciais para o desenvolvimento deste trabalho.

Ao Centro de Apoio ao Desenvolvimento Tecnológico - CDT/UnB, do qual sou bolsista e tive grande auxílio e incentivo no decorrer desse trabalho.

Ao Ministério do Planejamento, Orçamento e Gestão, em destaque as Secretarias de Patrimônio da União - SPU, de Gestão de Pessoas - SEGEP e de Orçamento e Finanças - SOF, patrocinadores de projetos em que pude inserir minha pesquisa.

Aos colegas do Laboratório de Tecnologias da Tomada de Decisão - LATITUDE/UnB e ao corpo administrativo da Faculdade de Tecnologia que incentivaram para o desenvolvimento deste trabalho.

À minha mãe Elisa e minha irmã Ana Cláudia que não perderam a fé na minha dedicação aos meus estudos.

À minha esposa Ângela e à minha filha Paula pelo apoio e incentivo que foi dado durante todo o tempo em que estive envolvido neste trabalho.

Agradeço sobretudo a Deus.

DEDICATÓRIA

Para:

Antonio Martins (in memorian) e Elisa de Nazaré Aymoré Martins,

Queridos Pais,

Ângela Granato Aymoré Martins e Paula Granato Aymoré Martins,

Minhas queridas esposa e filha, e

Sérgio, Marcelo (in memorian), Marco, Júlio e Cláudia,

Meus irmãos.

RESUMO

ARQUITETURA DE UM AMBIENTE COLABORATIVO DE BUSINESS INTELLIGENCE BASEADO EM UM REPOSITÓRIO DE ONTOLOGIAS E SERVIÇOS DE DADOS.

Autor: Valério Aymoré Martins

Orientador: Rafael Timóteo de Sousa Júnior

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, julho de 2012.

O conceito de Business Intelligence (BI) refere-se a um conjunto de metodologias, métodos, ferramentas e software que são usados a fim de fornecer soluções sistêmicas no apoio à análise de informações cujas especificações e desenvolvimentos são limitados a domínios específicos de informações.

Em soluções de BI convencionais, é necessário promover cargas massivas de dados fornecidos por outras organizações em repositórios locais, o que pode tornar a informação não disponível ou causar erros devido à má interpretação dos dados recebidos.

Propõe-se neste trabalho, uma arquitetura sistêmica de BI que busca soluções para essas limitações. Esta arquitetura é baseada em um repositório ontológico centralizado e utiliza serviços de dados descentralizados para fornecer dados para consultas analíticas genéricas.

A proposta foi validada pelo desenvolvimento de uma prova de conceito que permite mostrar a arquitetura em ambiente funcional e ilustrar seu interesse em diversas aplicações de BI.

ABSTRACT

ARCHITECTURE OF A COLLABORATIVE BUSINESS INTELLIGENCE ENVIRONMENT BASED ON A ONTOLOGY REPOSITORY AND DATA SERVICES.

Author: Valério Aymoré Martins

Supervisor: Rafael Timóteo de Sousa Júnior

Post-Graduation Program on Electrical Engineering

Brasília, July 2012

Business Intelligence (BI) refers to a set of methodologies, methods, tools and software that are used in order to provide system solutions to support information analysis. The specification and development of these system solutions are still limited to specific information domains.

Furthermore, in conventional BI solutions, it is necessary to promote massive data loads provided by other organizations in local repositories. Such massive loads can make the information not available on-time or cause errors due to misinterpretation of received data.

In this dissertation, a systemic architecture that seeks to address these limitations is proposed. The architecture is based on a centralized ontology repository and uses distributed data services to provide data to generic analytical queries.

The proposal was validated by developing a proof of concept software that allows the architecture to be implemented in an operational environment so as to illustrate its interest for several BI applications.

ÍNDICE

1 - INTRODUÇÃO	1
1.1 MOTIVAÇÃO	2
1.2 BENEFÍCIOS ESPERADOS	4
1.3 OBJETIVOS E PREMISAS	5
1.3.1 Objetivo Geral	5
1.3.2 Objetivos Específicos	6
1.3.3 Premissas para o Atingimento do Objetivo	7
1.4 ORGANIZAÇÃO DO TRABALHO	8
2 - CONCEITOS E FUNDAMENTAÇÃO TEÓRICA	9
2.1 CONSIDERAÇÕES INICIAIS	9
2.2 CONCEITOS BÁSICOS DE AMBIENTES DE BI	11
2.2.1 <i>Business Intelligence</i>	11
2.2.2 Estrutura de ambientes tradicionais de BI	12
2.2.3 Modelos multidimensionais	14
2.3 CONCEITOS BÁSICOS DE REPRESENTAÇÃO DA INFORMAÇÃO	18
2.3.1 Representação da Informação	18
2.3.2 Ontologias e seus componentes.....	23
2.3.3 Alinhamento e Fusão de Ontologias.....	26
2.4 CONCEITOS BÁSICOS DE AMBIENTE DE DADOS	31
2.4.1 Serviços Web e SOA	31
2.4.2 Pacotes de Mensagens SOAP.....	32
3 - ESTADO DA ARTE	34
3.1 ESPAÇO DE INFORMAÇÕES E ESPAÇO DE DADOS	34
3.2 ESPAÇO DE INFORMAÇÕES: FERRAMENTAS DE ONTOLOGIAS	37
3.2.1 Linguagem para Representação de Ontologias	37
3.2.2 Ferramentas de Ontologia	40
3.3 ESPAÇO DE DADOS: SERVIÇOS DE DADOS	46
3.3.1 Aspectos de Desempenho em Serviço de Dados.....	50
3.3.2 Aspectos de Segurança de Serviços de Dados	51
3.4 NÚCLEO DA PLATAFORMA: AMBIENTE DE BI	52
3.4.1 Ambientes de Interface Analítica de BI	52
3.4.2 Servidores OLAP	53
4 - PROPOSTA DE ARQUITETURA	58

4.1	REPRESENTAÇÃO DA ARQUITETURA PROPOSTA	60
4.1.1	Ambientes.....	60
4.1.2	Componentes.....	61
4.1.3	Camadas.....	64
4.1.4	Ciclos	65
	Perfis de Usuários.....	66
4.2	PROCESSOS E COMPONENTES ENVOLVIDOS.....	67
4.2.1	Processos e Componentes e Etapas envolvidas no Ciclo de Construção..	67
4.2.2	Processos, Componentes e Etapas envolvidos no Ciclo de Utilização	70
4.3	DA PROVA DE CONCEITO	71
5 -	PROVA DE CONCEITO E RESULTADOS OBTIDOS.....	73
5.1	SOLUÇÕES PROPOSTAS JUNTO AO ESTADO DA ARTE	73
5.1.1	Plataforma de Integração: Solução de BI.....	73
5.1.2	Espaço de Informações: Plataformas para Ontologia.....	80
5.1.3	Espaço de Dados: Plataforma para Serviços de Dados	83
5.1.4	Sítio UBI: Repositório Ontológico Centralizado	84
5.2	MODELO FUNCIONAL DE INTEGRAÇÃO	87
5.2.1	Arquitetura Operacional.....	87
5.2.2	Ferramentas Utilizadas como Componentes e suas Implementações	87
5.3	ETAPAS DA PROVA DE CONCEITO	103
5.3.1	Ciclo de Construção e suas Implementações	106
5.3.2	Ciclo de Utilização e suas Implementações	108
5.4	AVALIAÇÕES PRELIMINARES	111
5.5	EXEMPLOS DE USO NO INTERESSE PÚBLICO	112
6 -	CONCLUSÕES E TRABALHOS FUTUROS	115
6.1	APLICAÇÕES DA ARQUITETURA PROPOSTA	115
6.2	ATINGIMENTO DE METAS DA MOTIVAÇÃO	116
6.3	ATINGIMENTO DOS OBJETIVOS	117
6.4	TRABALHOS FUTUROS	119
	TRABALHOS PUBLICADOS.....	121
	REFERÊNCIAS BIBLIOGRÁFICAS	122
	ANEXOS	139

ÍNDICE DE TABELAS

Tabela 2-1 - Perspectivas de Business Intelligence na Internet Futura.....	10
Tabela 2-2 - Camadas de um Ambiente de BI Convencional.	13
Tabela 3-1 – Ferramentas de Alinhamento e Fusão.....	42
Tabela 3-2 – Infraestrutura - Ferramentas de Edição e de Repositório.....	45
Tabela 3-3 – Parâmetros de Query do OData.....	48
Tabela 3-4 - Resumo de Arquiteturas destacadas de Soluções de BI.	55
Tabela 3-5 - Resumo das mais destacadas soluções de BI.....	56
Tabela 5-1 - Classificação Funcional da Despesa.	81
Tabela 5-2 - Tempo de Resposta da Carga das Quatro Fontes de Dados Exemplo	89

ÍNDICE DE FIGURAS

Figura 2-1 – Arquitetura Típica de um Ambiente Convencional de BI	13
Figura 2-2 - Modelo de um Cubo Multidimensional.....	14
Figura 2-3 - Elementos do Modelo Multidimensional.....	15
Figura 2-4 - Esquema estrela (<i>star-schema model</i>).....	17
Figura 2-5 - Esquema floco-de-neve (<i>snow-flake model</i>).....	17
Figura 2-6 - Arvore Enraizada.	19
Figura 2-7 - Evolução na Representação de Informações.....	22
Figura 2-8 - Tipos de Ontologia de acordo com o seu nível de dependência.	25
Figura 2-9 – Processo de alinhamento e fusão entre duas ontologias.	28
Figura 2-10 – SOAP do consumidor ao provedor de serviços	33
Figura 3-1 - Arquitetura de Berthold.....	34
Figura 3-2 - Evolução dos ecossistemas tecnológicos	36
Figura 3-3 - OWL e RDF na arquitetura da Web Semântica.....	38
Figura 3-4 - Evolução no Tempo das Linguagens de Representação.	39
Figura 3-5 - URI OData	48
Figura 4-1 – Caso crítico de troca de informações e dados entre quatro organizações.	58
Figura 4-2 – Componentes da Arquitetura.	60
Figura 4-3 - Ambientes, Ciclos, Camadas e Processos da Arquitetura Proposta.....	64
Figura 4-4 – Modelo em Camadas e Funcionamento.....	65
Figura 5-1 – Arquitetura do Servidor OLAP Mondrian	74
Figura 5-2 – Ontologia-Modelo "Educação" visualizada no Protégé.....	81
Figura 5-3 – Fragmento de resposta XML acessando o Pentaho Data Integration.	83
Figura 5-4 – Fragmento de resposta XML acessando o PHP OData Producer Library.	84
Figura 5-5 – Representação do Sitio UBI.	85
Figura 5-6 –Sitio UBI - Ontologias de Domínio.	86
Figura 5-7 – Modelo Operacional da Arquitetura Proposta.	87
Figura 5-8 – Transformação PDI como Serviços de Dados	88
Figura 5-9 – Construção de Classes - Hierarquias e Níveis	91
Figura 5-10 – Construção de Instâncias - Valores dos Níveis / Dimensão	92
Figura 5-11 – Construção das Medidas e dos acessos aos Serviço de Dados	92
Figura 5-12 – Arquivo ubi.properties.....	94
Figura 5-13 – Arquivo ubi.script	95
Figura 5-14 – Transformação PDI como Serviços de Dados	96
Figura 5-15 – Opção " <i>Show all loaded ontologies</i> " do Protégé OWL 4.0.....	97
Figura 5-16 – Opção " <i>Merge Ontologies ...</i> " do Protégé OWL 4.0.....	97

Figura 5-17 – Fluxo da Tarefa Sistêmica "Sitio UBI ... Fusão"	99
Figura 5-18 – Exemplo de uma Proposta de Serviço de Dados	102
Figura 5-19 – Exemplo de uma Proposta de Chaves Substitutas	102
Figura 5-20 – Ontologia "Template.owl"	104
Figura 5-21 – Ontologia "Demografia.owl"	104
Figura 5-22 – Ontologia "Educação.owl"	105
Figura 5-23 – Ontologia "Imoveis_Publicos.owl"	105
Figura 5-24 – Ontologia "Saude.owl"	106
Figura 5-25 – Ontologia "Template.owl" somente com as classes Tempo e Geo	107
Figura 5-26 – Ontologia "UBI.owl" já certos domínios incorporados	107
Figura 5-27 – Acesso com a medida "População" como padrão.	108
Figura 5-28 – Hierarquias / Dimensões presentes no modelo superior.	108
Figura 5-29 – Medidas no modelo UBI	109
Figura 5-30 – Todas as medidas no domínio de "Tempo" e "Geo".	109
Figura 5-31 – Dimensão e Medidas de "Saude" com uma medida de "Demografia".	110
Figura 5-32 – Outra medida adicional oriunda do domínio "Imoveis Públicos"	110
Figura 5-33 – Extrato do resultado do indicador e dos valores de IDEB e IDH em 2009	112
Figura 5-34 – Relação de Área de imóveis e de Regiões por Nível de Urbanização.	113
Figura 5-35 – População dentro do domínio da Saúde.	113
Figura 5-36 – Atendimento de Equipamentos em relação a População no Tempo.	114
Figura 6-1 – Motivação para Centralização de Conceitos	116
Figura 6-2 – Motivação para o uso de Dados como Serviços	116
Figura 6-3 – Motivação para o uso de processos de Alinhamento e Fusão	117
Figura 6-4 – Solução dos problemas de "não relacionamentos" de modelos	117

ÍNDICE DE ANEXOS

ANEXO I - ONTOLOGIA-MODELO "EDUCAÇÃO.OWL" UTILIZADA.....	139
ANEXO II - PROVEDORES DADOS XML	140
ANEXO III - CONSUMIDORES DE DADOS XML.....	144
ANEXO IV - CÓDIGO-FONTE DO PROCESSO "SITE UBI ... FUSÃO"	152

ACRÔNIMOS

API	Application Programming Interface – Interface de Programação de Aplicações
BI	Business Intelligence – Inteligência de Negócios
HTML	Hypertext Markup Language – Linguagem de Marcação de Hipertextos
IEEE	Institute of Electrical and Electronic Engineers – Instituto de Engenheiros Elétricos e Eletrônicos
OASIS	Organization for the Advancement of Structured Information Standards - Consórcio incluindo empresas tais como IBM, Sun, Microsoft, SAP, Oracle, BEA e Symantec, visando o estabelecimento de padrões abertos e interoperáveis
ODBC	Open Data Base Connectivity – Conectividade para Banco de Dados Aberta
OLAP	Online Analytical Processing - Processamento Analítico On-line
REST	Representational State Transfer - Transferência de Estado Representacional
SOAP	Simple Object Access Protocol - Protocolo Simples de Acesso a Objetos
W3C	World Wide Web Consortium - Comunidade internacional onde as organizações-membros, funcionários de tempo integral, e agentes de setores públicos trabalham juntos para desenvolver padrões para Web (<i>Web standards</i>).
WEB SERVICES	Modelo de integração de sistemas e utilizado na comunicação entre aplicações diferentes através de serviços no ambiente Web
XML	eXtensible Markup Language - Linguagem de marcação extensível.

1 - INTRODUÇÃO

O termo Inteligência de Negócios (*Business Intelligence - BI*) refere-se a metodologias, métodos, arquiteturas, tecnologias, ferramentas e softwares para transformar dados brutos em informações significativas e úteis para permitir a análise de informações gerenciais no suporte à decisão (Evelson, 2009).

BI envolve processos de coleta, organização, análise e compartilhamento de informações e permite, entre outras coisas, a consulta dinâmica de dados, através da composição de medidas (seja por soma, média, contagem) em hierarquias de conceitos desejadas, com a possível aplicação prévia de filtros sobre os operandos utilizados e produzidos no processo.

Porém, por mais complexas e abrangentes que sejam as especificações e desenvolvimentos dos ambientes de BI, esses apresentam carências e limitações oriundas de seus princípios de construção e de operação. Muitas vezes, por estarem restritos aos limites organizacionais e respectivos contextos, os ambientes de BI se mostram limitados para uso por gestores que necessitem de uma visão acima do foco de sua organização.

No modelo tradicional de BI, as arquiteturas baseadas em Universos de Informações isolados dificultam a combinação de medidas entre Universos diferentes. Com efeito, os modelos tradicionais baseados em "tabelas fato" (Kimball, 1998) limitam a construção de soluções que permitam a análise nas situações em que há um conjunto diverso e dispar de conceitos.

Na prática, se a elaboração, instalação e desenvolvimento de um ambiente de BI tradicional em uma organização são ações que se realizam através de uma forte introspecção do contexto organizacional, pouco se dá de análise quando essa necessidade ultrapassa os limites organizacionais. O que ocorre é que as soluções de BI implementadas que necessitam de dados oriundos de outras organizações envolvem, na melhor das hipóteses, processos de colaboração rudimentar, pela contínua carga de dados associados a dimensões minimamente comuns entre as organizações fornecedora e consumidora.

Segundo Kimball (1998), para que sejam disponibilizados cruzamentos de métricas que ultrapassem os limites organizacionais, faz-se necessário promover cargas de dados de organizações parceiras através de processos de Extração, Transformação e Carga (Extraction, Transform and Load - ETL), utilizando como fonte dos dados os repositórios

locais das organizações onde tais dados são utilizados ou produzidos. Em geral, por serem atemporais, essas cargas de dados causam atrasos pela disponibilização tardia das informações das fontes externas.

Outra questão é que essas cargas são demandadas da organização receptora para a organização que provê os dados. Nesse sentido, se a organização de origem muda seu entendimento das informações ou formato dos conceitos atribuídos aos dados a serem transmitidos, um novo tratamento tem que ser desenvolvido e realizado na organização receptora, para alterar o processo de ETL. Como, nem sempre esse realinhamento ocorre, há uma possível utilização errônea da informação oriunda das fontes externas.

Já no que se refere aos conceitos, em ambientes tradicionais de BI persiste ainda o foco na organização, diferenciadamente do ambiente Web, que atravessou os limites profissionais e pessoais. Enquanto trabalhos sobre ambientes Web, principalmente sob o assunto Web Semântica, buscam organizar um modelo semântico único para disponibilização totalmente pública e sem fronteiras do conhecimento, os ambientes de BI tradicionais ainda reforçam essas barreiras.

No que refere aos dados, com o incremento da capacidade dos meios de comunicação, os princípios gerais que preconizavam a centralização dos dados para BI em parte perdem sua importância, permitindo que os dados possam ser utilizados com uma menor latência, principalmente, entre organizações.

1.1 MOTIVAÇÃO

Como exposto, o modelo tradicional de BI pelo uso de processos de ETL potencializa a disponibilização tardia de dados. Além disso, com o forte acoplamento de informações e dados, os conceitos organizacionais, que são expressos como informações, são tratados dentro de limites organizacionais, setoriais ou por assunto, em estruturas isoladas, o que se contrapõe aos princípios modernos de interoperação, tais como aqueles como na Web Semântica.

Quando dados e informação são trocados entre organizações, este trabalho analisa quatro situações que podem ocorrer:

- Organizações que trocam conceitos para suportar o acesso de por outros de visões além do seu domínio de conhecimento local envolve processos de carga de conceitos em regime de parceria. Porém, se uma das organizações altera a estrutura de seus conceitos organizacionais, ou altera a semântica de conceitos organizacionais, ou altera a sintaxe ou instâncias de conceitos organizacionais, erros podem se propagar "em cascata" obrigando a interlocução entre todas as organizações de forma a recompor seus processos de carga. Assim, a centralização de conceitos deve impor a responsabilidade pela manutenção dos modelos de conceitos a cada organização.
- Organizações que trocam dados de forma a suportar as suas faturações com as de outras organizações nos mesmos processos de carga em regime de parceria com os conceitos e medidas fortemente relacionados. Porém, se uma organização altera o formato ou a aditividade de seus dados, e/ou altera a rota de acesso ou de disponibilização dos seus dados, os processos de carga entre elas podem ser interrompidos até a interlocução entre todas as organizações de forma a recompor seus processos de ETL. Além disso, se a temporalidade de dados ocorrer em momentos diferentes, os mesmos dados serão apresentados com valores diferentes no caso de acesso por agentes externos, o que provoca a desconfiança do agentes nos dados fornecidos por todas essas visões. Assim, se cada organização prover o acesso de seus dados, isso delega a responsabilidade na conceituação e na forma dos dados a cada organização, além de equalizar a latência dos mesmos.
- Organizações que têm conceitos similares, mas não os "reconhecem como tal" mesmo após longas interlocuções para integração dos seus conceitos e seus valores, e principalmente se envolve várias organizações. Assim, se não há "entendimento" entre as partes, o modelo fica desalinhado, formando redundâncias de domínios, além de que a incorporação dos mesmos conceitos dificulta a análise de dados. Assim, cada organização e uma estrutura centralizada de administração podem observar, durante um processo de construção, os problemas de agrupamento de conceito sem afetar os modelos dos outros participantes.
- Conceitos analisados entre organizações podem apresentar divisões diferentes ou os mesmos conceitos podem estar definidos em níveis de detalhes diferentes. Assim,

um processo de unificação de conceitos deve prever mecanismos de tratamento desses desalinhamentos.

Para resolução dessas situações, a motivação central da proposta deste trabalho é apresentar um modelo arquitetural que dê suporte a que diversas organizações alinhem somente seus conceitos em um repositório central comunitário, integrando-os sob uma única semântica. Em adição, essa arquitetura também provê o acesso a dados diretamente nas suas origens, mantendo a gestão e disponibilização de tais dados dentro dos seus limites organizacionais.

Neste sentido, novas organizações podem ser alinhadas pela adição da estrutura de seus conceitos, a declaração de seus dados sob a forma de medidas e as rotas de acesso aos seus dados. Isto permite um incremento aditivo de informações que podem ser avaliadas e combinadas dentro de um ambiente de BI de acesso integrado único.

1.2 BENEFÍCIOS ESPERADOS

Através dos mecanismos propostos é possível a formação de comunidade de informações públicas baseadas no resultado do alinhamento e integração das dimensões de informações em uma visão única, e permitindo a consulta dos valores agregados correspondentes as métricas de forma descentralizada (isto é, junto as suas origens), com menor latência na disponibilização de informações entre as partes do que as resultantes das massivas cargas de dados.

Em consequência, são os seguintes os benefícios esperados:

- O atendimento de pesquisas multidimensionais envolvendo medidas sob a gestão de diversos participantes, permitindo a criação de resultados diversos baseados nas relações entre medidas disponibilizadas e as hierarquias de conceitos ao qual elas estão alinhadas.
- A entrega de dados como serviços nos limites da organização gestora com a possibilidade de uso de recursos de Computação em Nuvem (*Cloud Computing*).
- A redução dos esforços de integração de dados, eliminando a transmissão e tratamento de dados entre as organizações.

- A redução da sobrecarga (*overhead*) associado com a evolução da semântica das métricas e dimensões do negócio de qualquer participante, transferindo esta sobrecarga a cada gestor, mitigando o impacto para as outras entidades usuárias.
- O aumento das possibilidades de associação entre organizações de forma pública e/ou comunitária.
- A resolução de problemas de qualificação de dados pelo tratamento centralizado de conceitos e dos seus valores associados (instâncias dos conceitos).
- A gestão centralizada do modelo de conceitos ontológicos que podem ser construídos de forma desacoplada e independente.
- O uso no interesse do serviço público, que tem oportunidades especiais no uso de semântica em conjunto com a inteligência de negócios em avaliações que inter-relacionam assuntos diversos, tais como nas áreas de política pública, segurança, economia, educação. (Ludwig, 2005).

Este trabalho utiliza então a hipótese de que a separação dos ambientes de dados e informações nas soluções de BI em componentes passíveis de integração possa permitir a resolução de diversos problemas afetos ao modelo tradicional de BI.

1.3 OBJETIVOS E PREMISSAS

Este trabalho objetiva especificar, detalhar e implementar uma solução sistêmica de informações gerenciais (BI) em arquitetura distribuída, baseada em um repositório de conceitos centralizado que se utiliza de serviços de dados descentralizados, permitindo assim a disponibilização comunitária de informações de negócio sem que haja dispersão de conceitos e a necessidade de uma centralização de dados.

1.3.1 Objetivo Geral

O objetivo principal desse trabalho é a reavaliação dos paradigmas originais dos modelos das aplicações de BI pelo uso de recursos e padrões tecnológicos que vem emergindo no mercado, tal como ontologias e arquiteturas orientadas a serviços. Neste sentido, essa proposta apresenta uma implementação que facilita essa disponibilização através do uso

dos princípios de baixo acoplamento e de abstração oriundos dos paradigmas de orientação a serviços.

Para comprovar a hipótese, este trabalho descreve a implementação de um modelo arquitetural de ambiente colaborativo de informações gerenciais baseado em repositórios ontológicos de conceitos e suas instâncias, com acesso a dados disponibilizados como serviços na sua origem, e aplicá-lo em uma prova de conceito.

Para isso, um conjunto de elementos / componentes será selecionado, adaptado e integrado formando a arquitetura proposta, de forma a permitir a formação de ambientes de consulta de informações entre diversas organizações em comunidade públicas ou privadas com um mínimo de esforço.

Assim destaca-se as principais características desses componentes a serem especificados e integrados:

- Um ambiente sistêmico baseado em técnicas e padrões que habilitem os sucessivos passos de alinhamento de modelos de conceitos pelas organizações em uma plataforma global e centralizada de conceitos.
- Uma plataforma, implantada nas organizações participantes, que permita a construção e disponibilização remota de serviços de dados.
- Um servidor analítico que permita uma adaptação no acesso a serviços de dados remotos e com leitura e transformação dos conteúdos oriundos do repositório de conceitos citado em esquemas multidimensionais para seu uso.

1.3.2 Objetivos Específicos

Para atingimento do objetivo geral apresentado, é relacionada as atividades necessárias para o cumprimento do proposto:

- Estudar e definir os conceitos e estado da arte nas soluções sistêmicas envolvidas;
- Projetar o modelo arquitetural proposto; .
- Identificar os elementos necessários para formação da arquitetura proposta;

- Identificar as atividades necessárias para todo ciclo de vida de incorporação de um requerente à arquitetura;
- Propor junto às soluções de software existentes, plataformas que possam ser utilizadas para composição da arquitetura;
- Especificar e detalhar mecanismos e mensagens necessários para integração desses elementos, prevendo a adaptação de soluções pré-existentes;
- Adequar o modelo e os protocolos que serão usados para integrar as soluções escolhidas para cada elemento da arquitetura;
- Colocar à prova a proposta em cenários, através de uma prova de conceito simulando organizações distintas com informações em hierarquias diferenciadas.

1.3.3 Premissas para o Atingimento do Objetivo

Como limitações de escopo, algumas pressuposições serão consideradas:

- Que os modelos de conceitos e dados disponibilizados por cada organização já estejam disponibilizados em conformidades com arquiteturas de BI, isto é, com qualidade, com suas métricas já agregadas, e com os conceitos fisicamente implementados nas hierarquias das tabelas de fatos dos modelos multidimensionais, de forma a minimizar o impacto do uso do BI colaborativo.
- Que as organizações participantes devem possuir na sua infraestrutura de plataformas locais para publicação de serviços de dados.

A realização desses objetivos específicos visa atender metas estratégicas de:

- Desacoplamento da gestão de conceitos da gestão de dados.
- Interoperabilidade ontológica e semântica na resolução de conceitos e medidas nas dimensões e métricas em modelos multidimensionais unificados.
- Integração de dados pela entrega de dados como serviço para um sistema de BI.

1.4 ORGANIZAÇÃO DO TRABALHO

Esta dissertação trata dos principais assuntos pertinentes ao tema proposto, explicando sua estrutura de funcionamento e utilização. Para um melhor entendimento e organização, este trabalho é dividido em Capítulos conforme relacionado a seguir.

No Capítulo 2 são explicados os conceitos e a fundamentação teórica de BI e ontologias.

No Capítulo 3 é apresentado o estado da arte nos domínios de BI, ontologias e serviços de dados.

A arquitetura proposta e seu funcionamento serão desenvolvidos no Capítulo 4.

No Capítulo 5 é apresentado os modelos utilizados na prova de conceitos, sua implementação e os resultados alcançados na sua realização.

No Capítulo 6 é apresentada a conclusão deste trabalho e a exposição de trabalhos futuros.

2 - CONCEITOS E FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo serão tratados os Conceitos e Fundamentação Teórica para suporte ao entendimento dos princípios científicos, tecnológicos e metodológicos utilizados na realização dos objetivos deste trabalho.

Assim, será apresentada às Considerações Iniciais sobre o domínio de Business Intelligence - BI e dos seus ambientes tradicionais, estendendo na conceituação dos seus componentes básicos - conceitos e medidas - e os diferentes modelos de representação para esses componentes básicos: os conceitos como representação da informação e medidas como dados valorados.

2.1 CONSIDERAÇÕES INICIAIS

Enquanto ambientes de *Business Intelligence* - BI, em geral, se concentram em modelos semânticos sobre domínios específicos de informação e/ou organizações, a Web Semântica visa organizar o conhecimento do público em um modelo semântico único que ultrapassa estes limites de domínio e organização.

Neste contexto, assim como surgiu o termo Web 2.0 com uma nova visão das informações no ambiente Web, aflorou o conceito de BI 2.0 (Nelson, 2010). *Business Intelligence 2.0* prevê a revisão dos princípios de disponibilização de dados e de exploração de informações, associando ao BI tradicional os recursos oferecidos pela Web 2.0, focando na colaboração em comunidades de informação, no uso de novas tecnologias da Web tal como XML, RSS, *mashups* e AJAX, e nos conceitos de integração pela Web Semântica através de ontologias compartilhadas. Ele agrega ao ambiente BI tradicional os recursos oferecidos pela Web 2.0, apoiado pelas novas tecnologias da Web (Galatis e Tsekeridou, 2011).

Nelson (2010) aponta que BI 2.0 implica em “um afastamento do armazém de dados padrão que as ferramentas de inteligência de negócios têm usado”, e que “dará lugar ao contexto e a necessidade de relacionar informações de forma rápida a partir de muitas fontes”. Além disto, o autor identifica que as “decisões, os fatos e o contexto serão desenvolvidos por meio de trabalho colaborativo – *crowdsourcing*”, modelo de produção que utiliza a inteligência e os conhecimentos coletivos e voluntários espalhados pela Internet para resolver problemas de informação, e criar conteúdo e soluções através do uso de novas tecnologias.

Um direcionamento mais claro dessa perspectiva pode ser observada no artigo de Böhringer et al. (2010) que resume essa associação entre o futuro da Internet e do BI, como apresentado na Tabela 2-1.

Tabela 2-1 - Perspectivas de Business Intelligence na Internet Futura.

Fonte: Böhringer et al., 2010.

Tendência	Perspectiva Organizacional	Perspectiva Funcional	Perspectiva Tecnológica
Futuro da Internet para as Pessoas	BI Colaborativo (<i>BI Crowdsourcing</i>)	BI Social (<i>Social BI</i>)	BI como Software integrado às redes sociais (<i>BI Social Software</i>)
Futuro da Internet quanto dos Conteúdos	Inteligência do conhecimento (<i>Knowledge Intelligence</i>)	BI associado as necessidades de informação (<i>Answer-based BI</i>)	BI baseado em ontologias (<i>Ontology-based BI</i>)
Futuro da Internet quanto do seu acesso	BI descentralizado (<i>BI decentralization</i>)	BI Ubíquo (<i>Ubiquitous BI</i>)	BI Ubíquo e Sensível ao Contexto (<i>Ubiquitous Sensing</i>)
Futuro da Internet como Serviços	BI como Serviço (<i>BI as a Service</i>)	BI embutido em outras mídias (<i>Embedded BI</i>)	BI na Nuvem (<i>Cloud BI</i>)

Observa-se que o futuro de BI prevê a disponibilização de informações gerenciais além dos limites de cada organização. As perspectivas de Böhringer (2010) apontam para um gerenciamento descentralizado, o uso colaborativo e uso de ontologias na gestão dos conteúdos.

Neste sentido, o gerenciamento descentralizado pode se dar pela construção e desenvolvimento de árvores de conceitos por cada gestor, porém esses conceitos devem ser alinhados e centralizados para atender o princípio da visão colaborativa.

O futuro de BI como serviço em ambientes de Computação em Nuvem (*Cloud Computing*), isto é "o BI em Nuvem", pode ser revisto o acesso local as fontes de dados criadas no processo de ETL, pois o incremento das taxas de transmissão das redes de computadores mundiais ora permitem uma perspectiva de acesso a grandes volumes de dados remotamente.

Assim, este trabalho trata os conceitos associados a BI observando uma extensão e revisão desses para um novo patamar de tratamento de inteligência de negócio, e detalha propostas para implementar arquiteturas modernas de BI para o futuro.

2.2 CONCEITOS BÁSICOS DE AMBIENTES DE BI

Este tópico destaca os conceitos envolvidos nos elementos de ambientes de BI que serão utilizados para alcançar os objetivos propostos.

2.2.1 *Business Intelligence*

Business Intelligence pode ser entendido como um "guarda-chuva" conceitual que envolve a Inteligência Competitiva (*Competitive Intelligence - CI*), a Gerência de Conhecimentos (*Knowledge Management System - KMS*) e Inteligência de Negócios na Internet (*Internet Business Intelligence - IBI*) (Barbieri, 2001).

Referência nessa conceituação, Kimball (1998), define BI, “como sendo um recurso de consulta e apresentação dos dados de negócio de uma organização, construído especificamente para este fim”. Outro autor que é referência nesse entendimento, Inmon (2002), define BI como uma solução que se utiliza “de um conjunto de dados que auxiliam o processo de tomada de decisão. Este conjunto é orientado por assunto, integrado, não volátil e, em geral, varia em função do tempo”.

As soluções de BI, em sua implementação, suportam a consolidação de dados internos e externos, que são integrados para gerar informações pertinentes para o processo de tomada de decisão. O papel do BI é disponibilizar um ambiente informacional no qual dados operacionais possam ser coletados, tanto dos sistemas transacionais como de fontes externas, e agrupados e analisados segundo as dimensões do negócio. (Petrini et al., 2006).

Porém, segundo Petrini et al. (2006) atualmente não existe uma definição rígida que limite o que é solução de BI, pois esse tipo de solução sistêmica pode “englobar um grande grupo de aplicações ou soluções diferenciadas, contanto que elas possibilitem a melhoria do processo de tomada de decisões”. Nesse sentido, várias iniciativas têm recebido o nome de BI, pois sua definição permite o atendimento por uma vasta categoria de software e soluções que permitam a coleta, consolidação, análise e disseminação de informações visando melhorar o processo decisório.

Nas implementações de soluções de BI que atendem o modelo tradicional de Kimball e de Inmon, a construção é representada por tabelas físicas onde um ou mais valores (métricas) estão associados a um grupo de conceitos. Neste sentido, as materializações físicas dos

ambientes de BI associam dois tipos diferenciados de informação, isto é, "as qualidades e quantidades dos dados, formando um relacionamento entre essas".

Kimball (1998) propõe que para essas construções "se faz necessário o conhecimento de técnicas de modelagem multidimensional", de modo a prover ao usuário uma arquitetura de alto desempenho e a mais intuitiva possível.

Assim, ao contrário dos sistemas transacionais, que tratam o gerenciamento de dados em bases de dados relacionais normalizadas, sistemas de BI são geralmente analisados através da modelagem dimensional, técnica de projeto que busca modelar os dados em uma estrutura com dados quantitativos e qualitativos que permita um alto desempenho de acesso pela consolidação prévia de informações e o uso do processamento analítico de modelos multidimensionais.

2.2.2 Estrutura de ambientes tradicionais de BI

Soluções de BI existentes utilizam de recursos sistêmicos para entregar resultados em diferentes estilos de apresentação e sumarizações de acordo com as necessidades do usuário. Segundo Turban et al.(2007 apud Amara, 2008), o modelo de análise de BI podem ser realizados através do uso de dados sobre o Processamento Analítico Online (*On-Line Analytical Processing - OLAP*) para análises avançadas, mineração de dados ou análises preditivas. Davenport (2007) inclui a esses os grupos de geração de relatórios (*reporting*), os mecanismos de alerta (*alerts tools*) e a análise de negócio (*business analytics*).

Destaca-se desses recursos o OLAP. De acordo com Chaudhuri e Dayal (1997), OLAP é um mecanismo genérico e conveniente que permite que a navegação nos dados através de uma estrutura multidimensional de forma a realizar a pesquisa de informação e controlar a maneira que a informação é apresentada.

Para compreender o uso de OLAP, é necessário analisar o papel dos vários elementos existentes dentro de arquiteturas cliente-servidor de ambientes convencionais de BI. De acordo com Davenport et al. (2007) e provedores de soluções de BI como a IBM (2004) e Microsoft (2006), existem cinco níveis básicos reconhecidos nessas arquiteturas, conforme listados na Tabela 2-2 e ilustrados na Figura 2-1.

Tabela 2-2 - Camadas de um Ambiente de BI Convencional.
 Fonte: Davenport et al., 2007; IBM, 2004; Microsoft, 2006.

#	Davenport	Microsoft/IBM	Função Primária
1	<i>Data Management</i>	<i>Data Sources</i>	Gerenciamento de fontes de dados.
2	<i>Transformation</i>	<i>Data Integration</i>	Extração, transformação e carga de dados (ETL).
3	<i>Repository</i>	<i>Data Storage</i>	Armazenamento de dados e metadados.
4	<i>Application</i>	<i>Data Analysis</i>	Tratamento dos dados analíticos.
5	<i>Presentation</i>	<i>Data Presentation</i>	Manipulação de dados e apresentação.

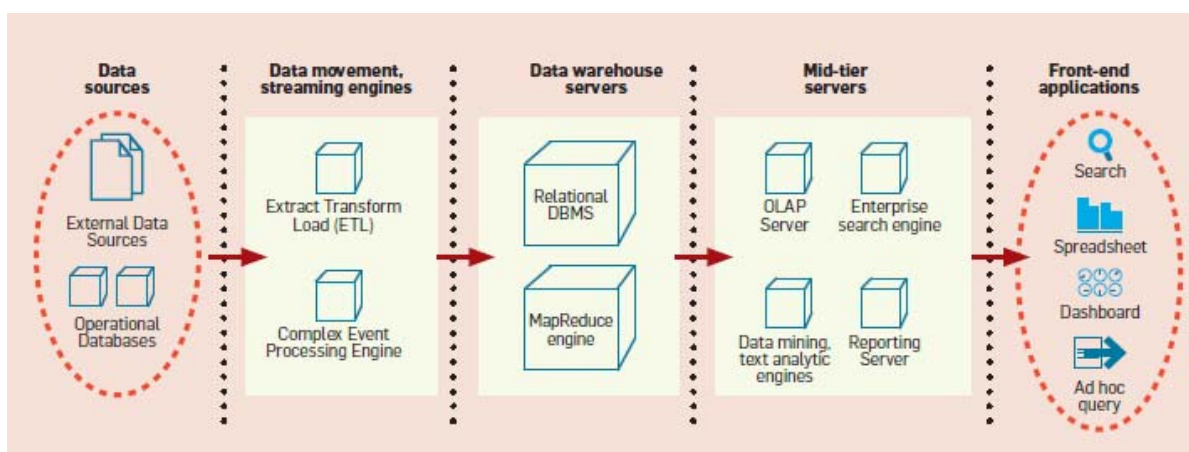


Figura 2-1 – Arquitetura Típica de um Ambiente Convencional de BI.
 Fonte: Chaudhuri e Dayal, 2011.

As primeiras três camadas na Tabela 2-2, relacionadas com as três primeiras colunas na Figura 2-1, *Data Management*, *Transformation* e *Repository*, são usadas no ciclo de construção convencional de arquiteturas de BI. Assim, os dados oriundos da camada *Data Management* são utilizados em um processo chamado de extração, transformação e carga (*Extraction, Transform and Load - ETL*) definido em *Transformation*. Os processos de ETL na camada *Transformation* atuam na camada *Data Management*, tratando e carregando dados e informações em modelos finais de dados dentro da camada *Repository*.

Estes modelos finais são estruturas específicas, conhecidas como modelos multidimensionais que são implementadas como tabelas físicas ou lógicas que sustentam a agregação de conceitos, porém fortemente acoplando os dados e as informações e essas entre si.

As camadas quarta e quinta apresentadas na Tabela 2-2, *Application* e *Presentation* respectivamente, representadas pelas quarta e quinta colunas da Figura 2-1, são executadas durante o ciclo de utilização de soluções convencionais de BI. Se essa utilização prevê a

realização de consultas genéricas e dinâmicas, é essencial o uso do servidor OLAP (OLAP Server) no papel de camada intermediária e de clientes de interfaces analíticas (*Ad Hoc Query*) como aplicação *front-end*. Estes componentes cooperam através de protocolos de solicitação e resposta. No ciclo de utilização, o servidor OLAP acessa diretamente modelos multidimensionais pré-definidos e também a camada *Repository*, de onde carrega os dados utilizados na formação das informações necessárias.

2.2.3 Modelos multidimensionais

A modelagem multidimensional representa a principal técnica para atender às necessidades exigidas em ambientes convencionais de BI. Os elementos básicos dessas estruturas são os "cubos multidimensionais" (ou cubo de dados), que são fisicamente *arrays* multidimensionais usados para facilitar o processamento das operações de matemática nas medidas que estão contidos dentro deles.

Tanto Kimball (1998, 2001) como Inmon (1992, 1998, 2005) observam um cubo multidimensional como uma forma materializada de dados que apresenta em suas arestas as “dimensões”, e onde as métricas unitárias estão representadas em cada elemento deste cubo, alocando os valores unitários de métricas em cada elemento desse cubo. A Figura 2-2 ilustra uma representação visual de um cubo multidimensional, formado por diversos cuboides, tendo como dimensões os conceitos de “Região”, “Produto” e “Mês”:

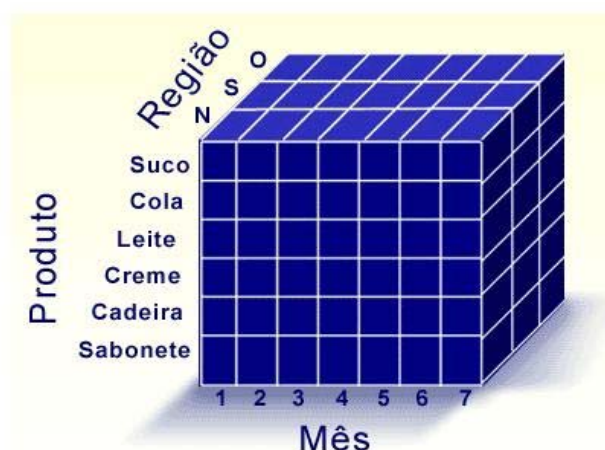


Figura 2-2 - Modelo de um Cubo Multidimensional.
Fonte: Siciliano, 2012.

No sentido de se compreender os princípios de modelagem multidimensional é necessária a definição de alguns dos conceitos envolvidos (Anzanello, 2005):

- Cubo é uma estrutura que armazena os dados em formato multidimensional.
- Dimensão é uma unidade de análise que agrupa dados de negócio relacionados. As dimensões se tornam cabeçalho de colunas e linhas.
- Hierarquia é composta por todos os níveis de uma dimensão, podendo ser balanceada (os número de níveis são equivalentes) ou não.
- Membro é um subconjunto de uma dimensão. Cada nível hierárquico tem membros apropriados aquele nível.
- Medidas (ou métricas) são os valores que são fatorados e apresentados.

A Figura 2-3 exemplifica alguns desses conceitos:

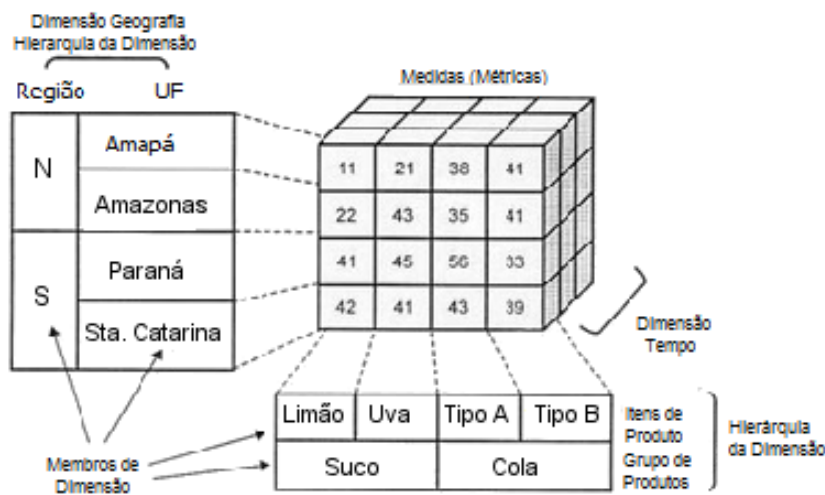


Figura 2-3 - Elementos do Modelo Multidimensional.

A cada uma das possíveis combinações de dimensões dá-se a designação de "cubóide". A computação dos cubóides pode ser total - com todos os cubóides ou parcial - apenas alguns, e se concretizar através de uma função de agregação nas medidas (Delis, 1999). Cabe ao tratamento das dimensões filtrar, agrupar e organizar as informações desejadas, segundo as questões gerenciais apresentadas.

As medidas do modelo multidimensional são agregadas conforme são realizadas funções sob as dimensões, funções essas denominadas como operações multidimensionais. Segundo Kimball (1998) e Inmon (1998), as aplicações de BI devem permitir que modelos multidimensionais realizem algumas operações multidimensionais específicas, tais como:

- *Slice* - Extração de informação sumariada (agregada) segundo um valor de dimensão a partir de um cubo de dados.
- *Dice* - Extração de um cubóide ou interseção de vários *slices*. Esta extração verifica as restrições de valor ao longo de várias dimensões.
- *Pivot* - Troca de linhas e colunas numa tabela (*crosstab*) para ajustar a forma como é apresentado o resultado.
- *Drill-up* - Apresentação de dados num nível de abstração superior.
- *Drill-down* - Apresentação de dados num nível de abstração mais específico.
- *Drill-across* - Detalha vários cubóides com dimensões compartilhadas, por desagregação ao longo de um nível específico.
- *Drill-through* - Detalha os valores, ao longo de uma dimensão dada, além do nível mais baixo do cubo, por consultas SQL diretamente na fonte relacional.
- *Ranking* (ou *Rank*) - Ordenação dos membros de uma dimensão de acordo com a ordem de uma das medidas.

O modelo multidimensional possui dois elementos básicos: dimensões e fato. “Fato” é uma coleção de dados implementados sobre tabelas que representam um assunto, sendo composto por dados de medida (quantificadores), e informações do contexto aos quais os dados estão associados (qualificadores) que são discriminados dentro das “dimensões”.

"Segundo Kimball (2001) e Inmon (2005), existem dois esquemas lógicos para a implementação dos esquemas lógicos do "fato" e das "dimensões" no suporte às representações de modelos multidimensionais, que são:

- Esquema Estrela (*Star Schema*), criado por Kimball (2001), que propõe uma visão cuja principal característica é a presença de dados altamente redundantes. É chamado de estrela porque a tabela de fatos fica ao centro com várias tabelas de dimensões que não tem outro relacionamento nas suas pontas. Neste modelo, as tabelas de dimensão não são normalizadas visando garantir melhores performance (Figura 2-4).

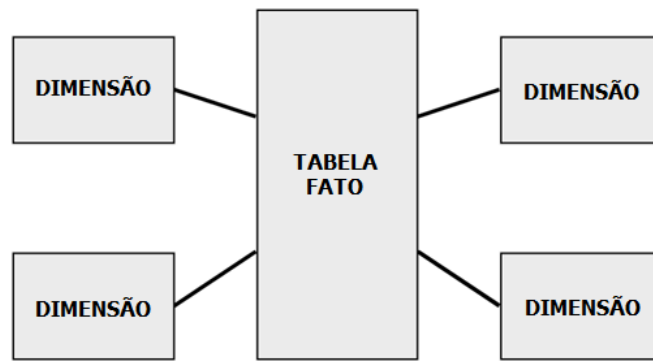


Figura 2-4 - Esquema estrela (*star-schema model*).

- Esquema Floco de Neve (*Snow Flake*), cujas tabelas dimensionais relacionam-se com a tabela de "fatos" e com outras tabelas dimensionais, que são representações da normalização das dimensões principais em diversos níveis de agrupamento. Este esquema tem como objetivo a normalização das tabelas dimensionais para diminuir assim o espaço ocupado por elas (Inmon, 2005) (Figura 2-5).

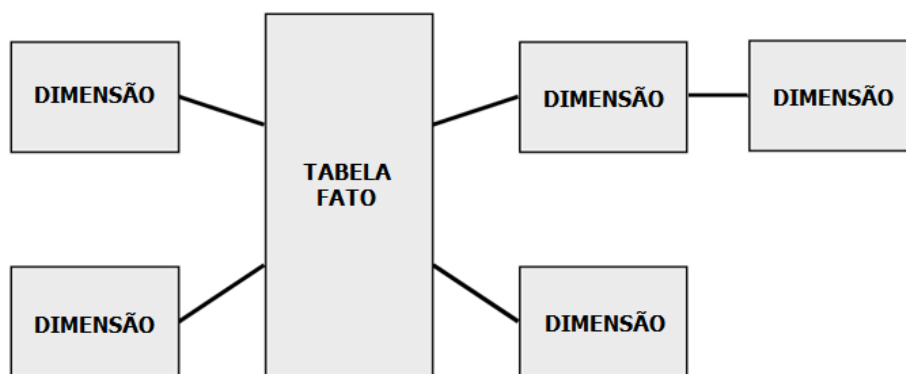


Figura 2-5 - Esquema floco-de-neve (*snow-flake model*).

Chaudhuri e Dayal (1997) reportam que os esquemas flocos de neve são um refinamento de esquemas estrela, onde a hierarquia dimensional é explicitamente representado através da normalização das tabelas de dimensão. Basicamente, no modelo estrela, todas as tabelas de dimensões são diretamente relacionadas com a "tabela de fato", enquanto no modelo floco de neve, as tabelas de dimensões formam hierarquias ligadas à "tabela de fato".

Machado (2007) define a modelagem multidimensional como uma técnica de concepção e visualização de modelos de dados de um conjunto de medidas que descrevem aspectos comuns de negócios. E esse modelo é formado por elementos básicos: as dimensões (qualificadores) e as medidas (quantificadores) agrupadas em contextos específicos (tuplas ou registros), chamados de "tabelas fato". Neste sentido:

- As dimensões determinam o contexto do assunto / fato do negócio. Possui uma ou mais hierarquias naturais além de atributos descritivos sem relacionamentos hierárquicos.
- As medidas são atributos numéricos que quantificam um fato e que são tratadas em conjunto ao contexto e as dimensões que participam do “fato”. Elas são fatoradas segundo suas categorias: medidas algébricas a partir de operações algébricas de agregação sobre dados atômicos ou por medidas distributivas / algébricas (média, desvio-padrão, etc.); e por medidas holísticas que armazenam agregados específicos (mediana, ranking).

Neste sentido, a visão lógica de BI, e suas representações físicas, tratam especialmente de dois aspectos informacionais: as dimensões, através das quais os conceitos qualificam a informação, e as medidas são representadas por resultados fatorados dos dados através de operações algébricas.

De modo geral, os conceitos informacionais dentro das dimensões são estruturados nos modelos multidimensionais usando árvores enraizadas, que organizam os conceitos através de relações “gênero-espécie”, uma representação de conhecimento chamado de taxonomia (Guarino, 1996).

Porém a taxonomia é uma representação simplificada, que apresenta limitações em relação a necessidade de organizar os diferentes níveis de granularidade de domínios de informação diferentes e de interrelacioná-las com as medidas.

De outra forma, a literatura moderna explora outro modelo de gestão do conhecimento e de representação de conceitos de informação, conhecido como ontologia (Guarino, 1996; Guarino e Giaretta, 1995; Noy e Musen, 2002; Noy, 2010).

2.3 CONCEITOS BÁSICOS DE REPRESENTAÇÃO DA INFORMAÇÃO

2.3.1 Representação da Informação

Desde os anos de ‘70, especialistas em computação reconhecem que a formação das árvores de conceitos é a chave para a construção de interligações de sistemas de informação. “Para proporcionar um modelo de metadados que descreva o contexto da

informação de forma não ambígua ou redundante, torna-se necessário um vocabulário específico para descrever cada realidade, adicionado de axiomas que dão significado pretendido às palavras deste vocabulário” (Silva Filho, 2006).

Vital (2011) aponta que é necessário "organizar o conhecimento na forma de teia de relações". Esta teia de relações, de maneira mais simples, vem sendo representada por casos específicos de árvores ordenadas e enraizadas. A Figura 2-6 ilustra uma árvore enraizada, terminologia comum em teoria dos grafos:

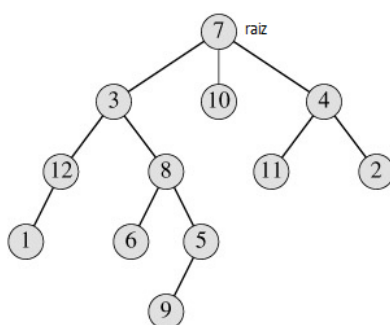


Figura 2-6 - Arvore Enraizada.
Fonte: Cormen et al., 2001.

Com os avanços da representação de conhecimento, temos o movimento colaborativo para a organização da Informação como estrutura de dados para Internet, denominada Web Semântica. A Web Semântica "não é uma Web separada, mas uma extensão da atual. Nela a informação é dada com um significado bem definido, permitindo melhor interação entre os computadores e as pessoas”. Berners-Lee (2001).

Trabalhos como Caldwell (2003) apontam que: “as estruturas de organização da informação para Web Semântica definem formalmente relacionamentos entre termos, sendo formados por taxonomia. Vale ressaltar que a taxonomia é um componente de suma importância para gestão de artefatos de informação” (Caldwell, 2003).

Porém, diversos outros trabalhos sobre o processo de representação do conhecimento em forma de árvores de conceitos exploram modelos diferenciados para Gestão de Conhecimento além da taxonomia, tais como *thesauri* e ontologias (Guarino 1995, 1996, 1998; Goldstone, 2002; Calvanese, 2001; Aumüller et al., 2005; Noy, 1999, 2000, 2010).

Vital (2011) observa tal situação, e cita que “as ontologias e taxonomias são modelos de representação do conhecimento com escopo conceitual próprio, e que se relacionam em

uma linha muito tênue, sendo por vezes confundidos e abordados de forma equivocada na literatura”.

De forma a buscar um alinhamento entre os três conceitos - de taxonomia, thesauri e ontologia - este trabalho utiliza o entendimento de Ferreira e Baptista (2005) e de Vital (2011) das semelhanças e diferenças entre esses:

- Taxonomia

“Taxonomia consiste em uma classificação por assunto que organiza os termos em um vocabulário controlado em uma hierarquia. Matematicamente, uma taxonomia é uma estrutura de árvore de classificações para um determinado conjunto de objetos. No topo desta estrutura temos o nó raiz, que se aplica a todos os objetos. Os nós abaixo desta raiz são classificações mais específicas que se aplicam a subconjuntos do conjunto total de objetos classificados.” (Garshol, 2004, apud Ferreira e Baptista, 2005).

Assim, uma taxonomia é um conjunto de termos usados para descrever as coisas que são agrupadas em uma estrutura de árvore. “Somos capazes de identificar relações pai-filho (gênero-espécie) entre os termos no vocabulário controlado.” (Ferreira e Baptista, 2005)

No âmbito da Gestão do Conhecimento, as taxonomias são definidas como "(...) elementos estruturantes, estratégicos e centrais para negócios baseados em informação e conhecimento (...) para classificar e facilitar o acesso à informação" (Terra, Gordon, 2002). Vital (2011) interpõe que “a estrutura mais citada na literatura para taxonomia é a hierárquica, sendo uma forma de caracterizá-la a visão de classes e subclasses, utilizadas nas construções de ontologias”.

A taxonomia organiza a informação da mais genérica para a mais específica, utilizando-se principalmente da relação hierárquica - gênero-espécie - entre os termos. Este tipo de relação é definido por Dahlberg (1978) como sendo aquela que aparece “entre dois conceitos que têm idênticas características, sendo, porém, que uma em relação à outra apresenta uma característica adicional, de modo que surge entre eles uma hierarquia”.

- *Thesaurus*

Thesaurus, basicamente, é ter taxonomias (como descrito acima) e estendê-las para torná-las mais capazes de descrever o mundo, não só permitindo que temas a serem organizados em uma hierarquia, mas também permitindo outras declarações a serem feitas sobre os assuntos.

Por definição, *thesaurus* é uma coleção controlada, em rede, de termos de vocabulário com as relações conceituais entre esses termos. Um *thesaurus* é uma extensão de uma taxonomia, permitindo que os termos sejam dispostos em uma hierarquia e também permitem outras declarações e relações possam ser feitas sobre os termos. (Gilchrist, 2003). Nos *thesauri*, algumas das outras declarações comuns são o significado dos termos e seus sinônimos.

- Ontologia

“Ontologias são semelhantes às taxonomias, porém usam uma semântica mais rica de relações entre termos e atributos, bem como apresenta regras estritas sobre como especificar termos e relacionamentos. Neste sentido, as ontologias podem se apresentar em vários formatos” (Jasper and Uschold, 1999), e como têm mais componentes, permite representações mais ricas do que a taxonomia tradicional.

Segundo Cardoso (2005), elas podem ser observadas como "uma conceitualização compartilhada do mundo, e consistem em aspectos como a definição de alto nível esquemas e aspectos de asserções, tais como entidades, atributos, inter-relações entre as entidades, o vocabulário do domínio e conhecimento é factual, isto é, tudo está ligado de uma forma semântica.” (Cardoso, 2005)

Ontologia então se refere a idéia de um conjunto de conceitos definidos que é mais geral do que taxonomia (Gruber, 2009). Em tal conjunto, os conceitos são representados por classes, seus atributos e suas instâncias, ligados por relações. Segundo Gruber (2009), elas devem ser desenvolvidas com o propósito de compartilhar o conhecimento entre as pessoas e / ou agentes computacionais.

Vital (2011) destaca os princípios das convergências e divergências entre os significados entre Ontologia e Taxonomia pelo seu uso. Segundo o autor as "taxonomias trabalham no

sentido de organizar a informação, já ontologias buscam estabelecer relações semânticas entre conceitos".

Se o relacionamento mais comum numa taxonomia é o de generalização, uma ontologia, permite outros tipos de relacionamentos mais complexos entre as classes (conceitos), tais como "parte-de", "causa-efeito", "localização", entre outros; o que permite atribuir características (propriedades) aos termos (atributos), isto é, ontologia agrega outras relações além da relação “generalização-espécie” comum na taxonomia.

Corroborando os princípios já expostos, que apontam que o processo de representação do conhecimento se assimila às taxonomias mas ultrapassa o limite tradicional do conceito, Ferreira e Baptista (2005) citam que as relações necessárias para a representação definidas em seus estudos com a ISO 2788:1986 de Thesauri e que a representação da informação pode ter uma “melhoria na sua sustentação por estruturas mais complexas, tais como Thesauri ou Ontologias”. Uschold et. al. (2004) buscam representar este modelo de forma semanticamente crescente segundo ilustrado na Figura 2-7:

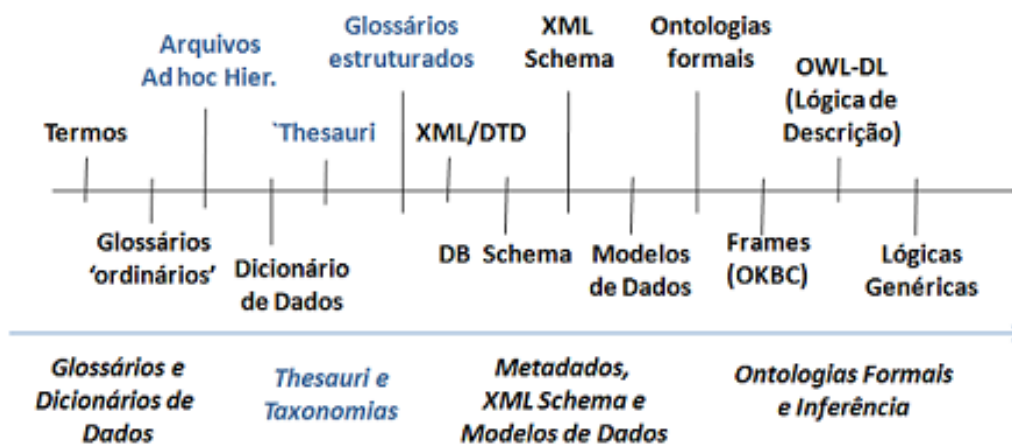


Figura 2-7 - Evolução na Representação de Informações.
Fonte: Uschold et. al., 2004.

Porém, se a representação da informação em modelos conceituais baseados em taxonomias pode dispor as classes e instâncias em árvores de conhecimentos necessárias, uma representação de conhecimento colaborativa necessita ainda da capacidade de interpretação de sinônimos ou termos correlatos e da adição de relações semânticas entre os conceitos, e assim extrapola a definição e uso da conceituação de taxonomia e de *thesaurus* sendo necessário o conhecimento de ontologia e sua construção.

Assim, na consecução desse trabalho, será tratado apenas o conceito de ontologia, porém observando suas características taxonômicas (árvores de especialização de conceitos) e de *thesauri* (sinônimos e termos correlatos).

2.3.2 Ontologias e seus componentes

Ontologias, para a ciência da computação, são em geral dispostas segundo a definição básica de Gruber (1993): "Uma ontologia é uma especificação explícita de uma conceitualização".

Para referenciar o entendimento comum, Borst (1997) acrescenta a essa definição o princípio de compartilhamento: "Uma ontologia é uma especificação formal de uma conceitualização compartilhada". Como "conceitualização compartilhada" é dado ênfase ao fato de que é necessário haver um acordo sobre a conceitualização que é especificada, pois a capacidade de reutilizar este entendimento será quase nula se essas são específicas e não são comumente aceitas.

Além da conceitualização clara e compartilhável, também deve se ter uma preocupação no uso de uma taxonomia apropriada dentro de uma ontologia, para que essa represente as reais relações de generalização correspondentes. O resultado será um melhor entendimento comum das pessoas envolvidas nas comunidades de informações assim como por sistemas informatizados.

Assim, quanto mais maduros os aspectos taxonômicos da ontologia "mais esses irão valorizar os ativos intelectuais das representações de informações propostas, criando contextos para sua oferta e uso" (Moodle/UFBA 2007). No uso compartilhado e público, o aspecto taxonômico pode servir para estimular novas descobertas, pelo desafio na geração de novas informações dentro de contextos global, e não apenas pelo uso limitado de um negócio ou de um assunto.

Dos seus componentes, ontologia refere-se a um conjunto de conceitos, classes ou termos (idéias abstratas) e instâncias individuais que esses conceitos podem assumir (valores), observada sob a forma de grafos que estendem as relações entre esses conceitos.

Gruber (1996) e Noy (1999, 2000, 2010) citam, com maiores ou menores detalhes, visões dos componentes básicos de uma ontologia, porém é utilizada nesse momento a visão simplificada de Vital (2011):

- classes: sets, coleções, conceitos, classes programáveis, tipos de objetos ou de coisas, organizadas em uma taxonomia;
- relações: representam o tipo de interação entre os conceitos ou descrevem adjetivos ou qualidades das classes;
- axiomas: usados para modelar sentenças sempre verdadeiras (restrições, etc.).
- instâncias ou indivíduos: utilizadas para representar elementos específicos das classes, ou seja, os próprios dados.

Apesar das ontologias possuírem estes elementos típicos, eles não são obrigatórios. Com isso, "uma ontologia pode assumir vários formatos, mas necessariamente deve incluir um vocabulário de termos e alguma especificação de seu significado. Este deve abranger definições e uma indicação de como os conceitos estão inter-relacionados, o que resulta na estruturação do domínio e nas restrições de possíveis interpretações de seus termos" (Breitman, 2005).

Como exemplos básicos dos aspectos taxonômicos de uma ontologia têm:

- O conceito "Tempo" que tem em suas divisões humanas naturais as classes de anos, meses de um ano, dias de um mês, entre outras divisões.
- O conceito de "Geografia" que no Brasil pode ser descrita pela hierarquia de regiões, unidade federativas das regiões, municípios das unidades federativas, etc.

Outro ponto importante é que no escopo de aplicação da ontologia se diferencia na visão de alguns autores:

- Na visão de Gruber (1993) ontologia é: "a especificação formal explícita de uma conceitualização compartilhada".
- Guarino (1994) e Uschold (1996) interpõem essa definição, acrescentando que o grau de especificação da conceitualização depende do propósito.

A definição desses autores para o “propósito desejado” para ontologia favorecem à uma classificação dessas. Neste sentido, segundo Guarino (1994), as ontologias podem ser classificadas de acordo com o nível de generalização. Dessa forma, o autor propõe as seguintes classificações:

- Ontologia genérica: inclui um vocabulário relacionado a coisas, eventos, tempo, espaço, casualidade, comportamento, funções, etc.;
- Ontologia de tarefa: fornece um vocabulário sistematizado de termos relacionados à execução de uma tarefa específica, independente do domínio em questão;
- Ontologia de domínio: especifica um vocabulário próprio de um dado domínio, como automóveis ou medicina;
- Ontologia de aplicação: contém as definições necessárias à aplicação de uma tarefa num dado domínio.

A Figura 2-8 apresenta a representação de dependência entre essas classificações identificada por Guarino (1998):

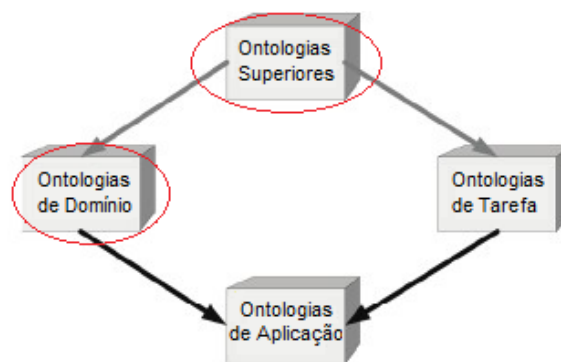


Figura 2-8 - Tipos de Ontologia de acordo com o seu nível de dependência.
Fonte: Guarino, 1998.

Destaca-se da Figura as categorias “ontologia de domínio” e “ontologia genérica”. Ontologias de domínio representam conceitos em assuntos de maneiras muito específicas, incompatível em âmbito com uma representação mais genérica. Os modelos com representação mais genéricos são denominados de ontologia de nível superior (*upper-level ontology*), ou apenas ontologia superior (*upper ontology*) (Noy, 1999).

Segundo Kiryakov (2001), ontologias superiores “capturam principalmente os conceitos que são fundamentais para a compreensão do mundo pelo ser humano. Ela é fundamentada no senso comum o que dificulta a formalização de uma definição específica para elas. Elas representam o conhecimento utilizando relações principalmente taxonômicas”.

Uma ontologia superior é um modelo de classes comuns que são geralmente aplicáveis sob as ontologias de domínio. Ele emprega um glossário núcleo que contém os termos com classes e as inter-relações com objetos associados dentro de conjuntos de domínios.

Como a construção de ontologia em um ambiente colaborativo envolve entidades participantes com prováveis assuntos diferenciados em domínios específicos é necessária a expansão do conceito e uso de ontologias de domínio à um nível superior de abstração, isto é, uma ontologia superior. Assim, para se organizar conceitos dispersos e diferentes se faz necessária a construção de uma ontologia superior, a partir de ontologias de domínios diversos (Noy, 1999, 2002). Porém essa agregação deve prever a unificação de conceitos similares ou iguais possam estar representados de forma diferente. Esse é um dos desafios da modelagem ontológica.

Essa possibilidade é real, tanto que Kiryakov et al. (2001, apud Kalfoglou, 2002) “desenvolveu uma estrutura para acesso e integração de ontologias de nível superior. Eles fornecem um serviço que permite que um usuário para importar ontologias linguísticas em um servidor, que depois são mapeadas em outras ontologias (*OntoMap Project*)”.

Será tratado então o processo da agregação de ontologias em uma nova ontologia que represente a união dessas em uma ontologia superior, mesmo que essas ontologias agreguem conceitos em domínios diferentes.

2.3.3 Alinhamento e Fusão de Ontologias

Segundo Noy (2002, 2010), quando duas ontologias são desenvolvidas com base em um conjunto de regras pré-definidas e que atendam uma linguagem de representação comum, o seu alinhamento e fusão de forma automática ou semiautomática são possíveis utilizando recursos de soluções sistêmicas.

Porém várias terminologias são usadas na literatura que correspondem em parte ou em todo ao processo que visa dar apoio a diferenciação e/ou unificação entre dois modelos

ontológicos em um novo modelo, entre eles: mapeamento (*mapping*), articulação (*articulation*), alinhamento (*alignment*), emparelhamento (*matching*), integração (*integration*), fusão (*merging*), mediação (*mediation*), entre outros.

De forma a limitar a “confusa” aplicação de termos distintos para processos similares, é apresentado na relação a seguir os conceitos de interesse a serem utilizados neste trabalho em relação aos diversos termos técnicos que são utilizados na literatura:

- Alinhamento: é o processo no qual duas ontologias são comparadas a partir de mecanismos de análise de correspondência (*matching*), onde seus elementos comuns são identificados. A esse processo cabe primordialmente a identificação da correspondência semântica entre as estruturas de metadados, modelos de esquemas de banco de dados, formatos de mensagens XML e ontologias. (Aumueller et al., 2005). Ehrig e Staab (2003) definem esse processo como: "Dada duas ontologias O1 e O2, um alinhamento de ontologias significa que para cada entidade na ontologia O1 tentamos encontrar uma entidade correspondente que tem o mesmo significado pretendido na ontologia O2".
- Fusão: é o processo no qual os resultados obtidos do alinhamento são utilizados para gerar uma ontologia superior que realiza a unificação dessas ontologias. O ponto final da fusão de ontologias é uma nova ontologia que representa a união das ontologias originais, captando todo o conhecimento a partir das ontologias originais. O desafio na fusão ontologia é garantir que todas as correspondências e as diferenças entre as ontologias sejam refletidas na ontologia resultante da fusão. (Bruijn et al., 2006). Em geral, a fusão de ontologias depreende que uma pode ser uma ontologia base comum e a outras seja um novo modelo que se deseja mesclar.

Assim, o processo completo é realizado através de dois passos: primeiro, o sistema deve identificar e alinhar estas ontologias analisando seus componentes básicos comuns (classes, instâncias, relações, axiomas). Em seguida, um processo de fusão é executado para construir uma nova ontologia superior (Gruber, 2009; Jasper e Uschold, 1999).

A Figura 2-9 (Abels et al., 2005) ilustra o processo de alinhamento e fusão de ontologias. A Figura 2-9(a) mostra a identificação e alinhamento de duas ontologias enquanto a Figura 2-9(b) ilustra o processo de fusão resultando em uma ontologia de nível superior.

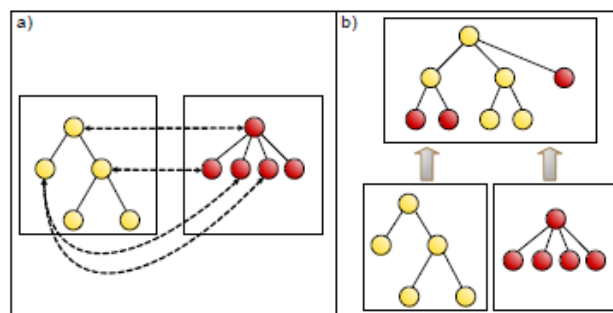


Figura 2-9 – Processo de alinhamento e fusão entre duas ontologias.
 Fonte: Abels, 2005.

O modelo resultante pode ser usado em processos de alinhamentos e fusão sucessivos, isto é, esses processos podem ser continuamente repetidos pela adição de outros modelos ontológicos e resultando em novas ontologias superiores. Observa-se que a correta resolução do problema de correspondência (Figura 2-9(a)) é de fundamental importância para a fusão de ontologias, principalmente entre múltiplos domínios.

Um processo similar permite também a subtração de uma ontologia previamente incorporada, rearranjando a ontologia final em um modelo que não incorpora conceitos referentes especificamente a ontologia retirada.

Para reduzir o esforço manual necessário para essa resolução, muitas técnicas e modelos foram desenvolvidas para resolver o problema de alinhamento. Diversos trabalhos apresentados (Noy e Musen, 2002; Ehrig e Staab, 2003; Aumueller et al., 2005; Andrejko et al., 2006), analisam e avaliam os diferentes modelos e soluções sistêmicas propostas para realizar esse esforço. As abordagens propostas tipicamente exploram as características do esquema das ontologias, tais como nomes, tipos e propriedades estruturais, e características dos seus valores (instâncias), bem como o conhecimento dos dicionários e dos thesauri.

Os mecanismos de avaliação de semelhança são algoritmos de correspondência (*matching*) que geralmente expressam seus resultados conhecidos como “similaridade” (*similarity*) (Andrejko et al., 2006; Abolhassani et al., 2006).

A computação da similaridade tem como abordagem o tratamento em dois grupos gerais: o as igualdades e diferenças léxicas e estruturais (Abolhassani et al., 2006). Existem várias estratégias utilizadas para comparar ontologias, onde características linguísticas, de contextos, heurísticas e probabilísticas podem ser usadas (Ehrig e Staab, 2004 - Tabela 3-1 - Estratégia Utilizada). No nível mais alto, essas estratégias tratam o problema de

alinhamento usando dois mecanismos: em fragmentos ou por reutilização de resultados esquemas existentes.

O alinhamento baseado em fragmentos apresenta a idéia de "dividir para conquistar". Esse mecanismo decompõe um grande problema de alinhamento em subproblemas menores, dividindo-os as ontologias ao nível de fragmentos (Aumueller et al., 2005). Com a redução do tamanho do problema, pretende-se uma melhor adequação em qualidade em cada nível, pois cada par de fragmentos semelhantes em cada ontologia passa a representar um problema de alinhamento individual.

A estratégia de reutilização de esquemas existentes (Madhavan, 2005) tem como foco a aprendizagem e uso de estatísticas em simultâneas operações de correspondência entre esquemas. Seu principal mecanismo é uma operação de junção de um modelo de mapeamento (alinhamento) que consiste em mais mapeamentos, tais como AB, BC, e CD, para, a seguir, obter um novo mapeamento entre A e D.

Além do princípio de estratégia, o alinhamento de ontologias pode ser realizado em diferentes elementos de uma ontologia. Assim, a semelhança das duas ontologias pode ser avaliada com base na semelhança das classes e/ou dos valores das suas instâncias (Ehrig e Staab, 2004 - Elementos Estruturais Utilizados – Tabela 3-1). Das classes em geral são avaliadas o número de classes comuns, a distância dos conceitos na árvore taxonômica e a semelhança que as classes possuem nos seus atributos. Já das instâncias são avaliados os quantitativos comuns e a correspondência dos atributos em valores de instância iguais (Andrejko et. al., 2006).

Assim, o esforço de alinhamento, e posterior fusão, entre ontologias pode se tornar tão correto quanto necessário se os elementos selecionados para o ambiente ontológico são capazes de usar uma rica combinação de diferentes algoritmos (*matchers*). (Kalfoglou, 2002).

Abels et al. (2005) identifica alguns desses métodos (algoritmos) com base no estudo da literatura:

- Semelhança de texto: é um método baseado em comparação de sequência de caracteres para a identificação de elementos semelhantes.

- Extração de palavras-chave: identifica e separa palavras-chaves em descrições textuais que são usadas para detectar o significado pela observação de seus nomes próprios ou através da análise da frequência de substantivos comuns.
- Métodos baseados em linguagem: trata da remoção de palavras desnecessárias (*stop-words*) e posterior análise do texto resultante, seja nos nomes de classe a ou em seus atributos. Por exemplo, eles podem ser usados a fim de detectar que a classe "casa" e "casas" são idênticas. Este método trata basicamente os prefixos e / ou sufixos típicos da linguagem.
- Identificação das relações entre palavra: inclui recursos linguísticos, tais como dicionários e enciclopédias, a fim de identificar sinônimos.
- Semelhança por tipo: analisa os elementos, olhando para seu tipo de dados pois, em teoria, atributos da classe idêntica tem que ser o mesmo tipo de dados.
- Análise de herança de classe (*is-a*): Este método observa a herança entre classes a fim de identificar os relacionamentos “é-um”. Por exemplo, na existência de duas ontologias A e B, onde A contem "carro", e B contém "veículo", a análise de herança de classe tenta encontrar relações entre os conceitos (carro “é-um” veículo).
- Análise estrutural e taxonômica: A análise estrutural identifica classes idênticas olhando para os seus atributos e relações. A ideia principal é que duas classes de duas ontologias são semelhantes ou idênticas se elas têm os mesmos atributos e as mesmas classes adjacentes.
- Interpretação dos dados e análise das propriedades-chaves: Inclui as instâncias e identifica classes semelhantes olhando para essas. Se duas classes têm as mesmas instâncias, então elas provavelmente são coincidentes.
- Mapeamento gráfico: usado para identificar estruturas semelhantes em duas ontologias, procurando por partes idênticas. Ao contrário da análise estrutural, o método de mapeamento gráfico tem como base interpretar apenas a representação gráfica da estrutura da ontológica.

Diversas definições, abordagens e modelos matemáticos já foram propostos, entre eles os de Zhong et al. (1995), Rahm e Bernstein (2001), Noy, N., Musen, M. (2001), Euzenat and Valtchev (1997, 1999, 2003, 2004), Ehrig and Sure (2003, 2004, 2005), Kalfoglou (2005), Abolhassani et al. (2006). Recentemente, Park et al. (2011) apresentou um trabalho matemático completo com referência aos fatores que afetam o cálculo dessa similaridade.

Porém em ontologias de um mesmo domínio, que utilizam o mesmo conjunto de elementos básicos para especificar os significados dos elementos, mesmo que seja entendido que essas possam ser mesclados automaticamente (Noy, 2000), em ontologias de colaboradores diferenciados e/ou de domínios diferentes deve-se impor a necessidade de troca de informações até que se atinja um patamar mínimo exigido e que não afete outros modelos existentes de outros colaboradores.

Neste sentido, durante a geração de ontologias superiores, inconsistências podem ocorrer se os processos são feitos automaticamente, mesmo quando as ontologias têm o mesmo conjunto de elementos padrões em suas especificações (Noy e Musen, 2000).

Apresentado os modelos de representação da informação e seus modelos de alinhamento e fusão, em sua consecução, este trabalho necessita estender o mesmo conhecimento para os dados e medidas. Neste sentido, serão tratados a seguir os conceitos utilizados para dados e seus ambientes de disponibilização.

2.4 CONCEITOS BÁSICOS DE AMBIENTE DE DADOS

A crescente utilização de ambientes distribuídos (Bennett et al., 2000) e o crescimento da largura de banda na transmissão de dados pela Internet (Kempf et al., 2010) favoreceram o uso de um modo de acesso de dados com base descentralizada e fontes de dados distribuídas. Uma vez que é necessária a integração entre parceiros diferentes com ambientes de infraestrutura *a priori* desconhecidos, o uso de padrões e modelos interoperáveis é necessário.

2.4.1 Serviços Web e SOA

Quando se trata de interoperabilidade, nos últimos anos intensificou-se o uso de Serviços Web (*Web Services*) e da Arquitetura Orientada a Serviços (*Service-Oriented*

Architecture - SOA), que oferece padrões abertos de interoperabilidade para sistemas distribuídos independente de plataforma aberta, através da disponibilização de processamento de dados de forma independente e orientada por contextos funcionais, isto é, orientada a serviços (Erl, 2005).

De acordo com Woods and Mattern (2006), a "Arquitetura Orientada a Serviços - SOA é um estilo de design que orienta todos os aspectos da criação e utilização de serviços de negócios em todo o seu ciclo de vida bem como a definição da infraestrutura de TI de forma a permitir que diferentes aplicativos que troquem dados e participem dentro de processos de negócios, independentemente dos sistemas operacionais ou linguagens de programação que são utilizados", garantindo assim a "interoperabilidade entre componentes de software utilizando acoplamentos fracos" (Silva, 2008 apud Nunes, 2005).

Utilizando-se dos mesmos mecanismos definidos pela W3C para *Web Services (Web Service Architecture, 2004)*, SOA apresenta como especificação básica: um protocolo de acesso (*Simple Object Access Protocol - SOAP*), uma linguagem de descrição de serviços (*Web Service Description Language - WSDL*) e um mecanismo de descoberta, descrição e integração (*Universal Description, Discovery and Integration - UDDI*).

2.4.2 Pacotes de Mensagens SOAP

O protocolo SOAP prevê a troca estruturada de informações entre o consumidor e o provedor de serviços e opera sobre protocolos de transporte comuns tais como HTTP, SMTP, JMS, RMI/IIOP - todos compatíveis com o RFC 1149 (Curbera, 2002). No coração dessa arquitetura estão os eventos e mensagens comunicados pelo SOAP usando esquemas baseados na Linguagem de Marcação Extensível (*Extensible Markup Language - XML*) cujo resultado físico (*payload*) se apresenta no formato de um texto plano (W3Schools.com).

Relacionado com as técnicas de SOA, e com o foco específico de fornecer dados sobre demanda de forma padrão, estão emergindo tecnologias voltadas disponibilizar Dados como Serviço (*Data as a Service - DaaS*) ou simplesmente Serviços de Dados (*Data Services*). Esses são serviços web específicos para entrega de dados como conteúdo, disponibilizando funcionalidade para as operações básicas de banco de dados, tais como de criação, leitura, atualização e remoção (da sigla CRUD, acrônimo de *create, read, update and delete*).

Como o SOAP é capaz de trocar as chamadas RPC em uma variedade de protocolos subjacentes, as implementações iniciais buscaram utilizar os padrões de Serviços Web SOAP como serviços de dados, tal como a utilização do Apache Axis integrado a motores de transformação de dados para realização desses serviços (Edmonson, et al., 2004).

Porém, os inúmeros intermediários do modelo de transporte SOAP, conforme ilustrado na Figura 2-10, afetam profundamente o desempenho desse modelo:

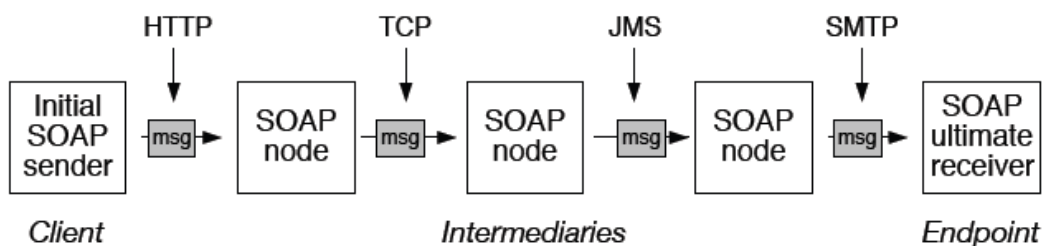


Figura 2-10 – SOAP do consumidor ao provedor de serviços
Fonte: Skonnard, 2003.

Em meio a discussões sobre a melhoria no desempenho do modelo de entrega de dados via protocolo SOAP, foi introduzido por Fielding (2000) em sua tese de doutorado, um novo modelo de acesso à Serviços Web chamado *Representational State Transfer* (REST). Utilizando explicitamente os quatro principais métodos HTTP - POST, GET, PUT e DELETE - o REST implementa as quatro operações básicas do CRUD (respectivamente). Para o uso do REST, é necessário apenas definir uma chamada HTTP, através de um simples Identificador Uniforme de Recursos - URI (*Uniform Resource Identifier*) para identificar o recurso a ser acessado. Dada essa simplicidade, o REST “vem substituindo rapidamente o uso do SOAP e do WSDL devido ao seu estilo simples” (Rodriguez, 2008).

Porém o REST ainda utiliza o XML como padrão na recepção dos dados oriundos da solicitação de serviços. Neste sentido, o uso de modelos de compactação sob esquemas XML binários (*XML Binary*) e de cache no roteamento (*proxy*) de serviços de dados vem sendo estudados, e serão tratados junto ao Capítulo 3 - Estado da Arte.

3 - ESTADO DA ARTE

Neste Capítulo será apresentado o estado da arte nos domínios das soluções sistêmicas para ontologias e serviços de dados, bem, como para o elemento utilizado para integrar esses ambientes - o núcleo central da arquitetura proposta - as soluções de BI.

Porém, para situar o uso proposto de ontologias e de serviços de dados integrados em ambientes de BI, este trabalho descreve a proposta que o inspirou: o modelo de duas camadas de Berthold et al. (2010).

3.1 ESPAÇO DE INFORMAÇÕES E ESPAÇO DE DADOS

Diversos trabalhos, destacando-se os de Franklin et al. (2005) e Berthold et al. (2010), analisam BI em seus aspectos particulares, os aspectos qualitativos e quantitativos, nominando-os e desacoplando-os distintamente, sob a terminologia de Infospace (Espaço de Informação) e Dataspace (Espaço de Dados).

Sob essa ótica, Berthold et. al. (2010) propõem arquiteturas corporativas para BI baseadas em “um modelo de dados global que consiste na sua separação em dois conceitos: o gerenciamento dos dados e o gerenciamento do modelo de relações que fornece informações de contexto aos dados (...)”.

O objetivo principal dessa proposta é “(...) de se ter uma separação clara do ponto de vista informacional a partir do ponto de vista dos resultados empresariais”.

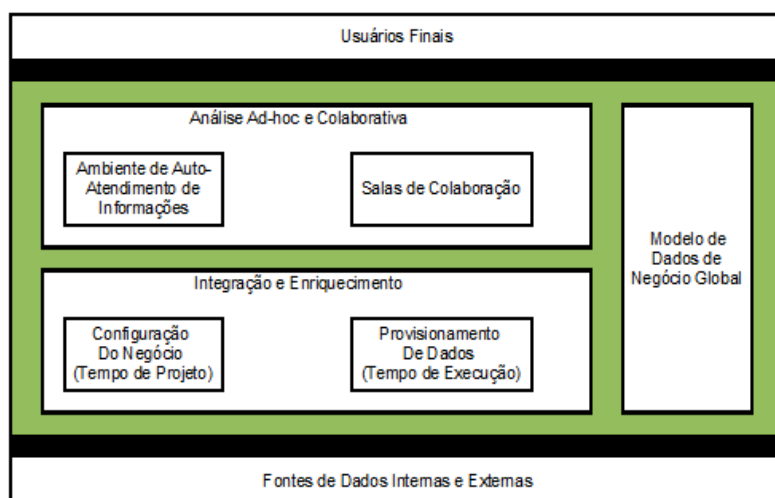


Figura 3-1 - Arquitetura de Berthold.

Fonte: Berthold, et al., 2010.

A Figura 3-1 apresenta os elementos dessa proposta dos autores.

Esses autores propõem como conteúdo para cada um desses aspectos em:

- “(...) *Dataspaces*: que se baseiam na centralização de fontes de dados gerenciadas pela TI, tais como: fontes de dados transacionais, fontes oriundas de modelos de BI (*data warehouses* e *datamarts*), fontes de dados locais descentralizadas que podem ser adicionadas pelos usuários de negócios, e dados externos (por exemplo, *feeds* de RSS externos). Assim, tanto os dados centralizados, locais ou externos, são entendidos como partes do *Dataspace* (espaço dos dados, grifo do autor).
- *Infospaces*: que são representações de negócios abstraídos a partir de conceitos técnicos, como formatos de dados ou esquemas. Eles são usados para qualificar a integração de dados de diferentes fontes para as necessidades de grupos de usuários específicos (espaço da informação, grifo do autor). “Esforços atuais de pesquisa concentram na introdução de abordagens com base ontológica (nos *Infospaces*, grifo do autor) para melhorar a tomada de decisão (...)”

“Fornecer uma estrutura final baseada na captura da visão empresarial”, segundo esses autores, “é garantir um desacoplamento (dissociação) declarativo e flexível entre fontes de dados e modelos orientados aos negócios”.

Porém, todo o tratamento de BI se dá através das “tabelas fato”, que são repositórios unificados da representação dos negócios (conceitos) e dos dados, o que torna rígido o modelo em si, pois os dois estão representados mesma tabela física restrita a um contexto específico.

Neste modelo tradicional, as interfaces de pesquisa atuam junto aos “cubos multidimensionais”, que são fisicamente ou logicamente representados por tabelas de “fato”, selecionando ou filtrando conceitos dentro de dimensões de forma a agregar suas métricas (dados), e formatando saídas aos usuários conforme a navegação nas dimensões (Kimball e Inmon). Por esse aspecto, os ambientes de BI tradicionais, e seus modelos, apresentam forte acoplamento entre dados e os conceitos que os indexam, filtram e controlam.

Segundo Lafuente (2011), “ao lado das soluções de armazenamento de dados, ou *data warehousing*, as arquiteturas orientadas a serviços sobre as quais os aplicativos atuais de BI funcionam até agora foram os protagonistas no reino da análise de dados. Entretanto o surgimento dos padrões da Web Semântica (...) - está abalando o setor”. Assim, não há tecnologia de gestão do conhecimento comparável em flexibilidade e potencial que os formatos semânticos, "pois esses permitem vincular todo tipo de elementos (dados, metadados, mapas conceituais ou ontologias, serviços, dispositivos e aplicativos) e descrevê-los conceitual e contextualmente para consulta posterior por sistemas inteligentes, capazes de relacionar essas entidades de maneira como nunca tinham sido vinculadas e descobrir, novas conexões entre os dados”.

Logo, a proposta de Berthold (2010) encontra um suporte, no que se refere a construção de uma Espaço de Informações, nos princípios da Web Semântica. Em adição, o suporte a dados como serviços apóia a formação do Espaço de Dados.

Para atingir esse fim, serão utilizadas tecnologias modernas conforme ilustrado por Pollock (apud Lafuente, 2011) que é apresentada na Figura 3-2:

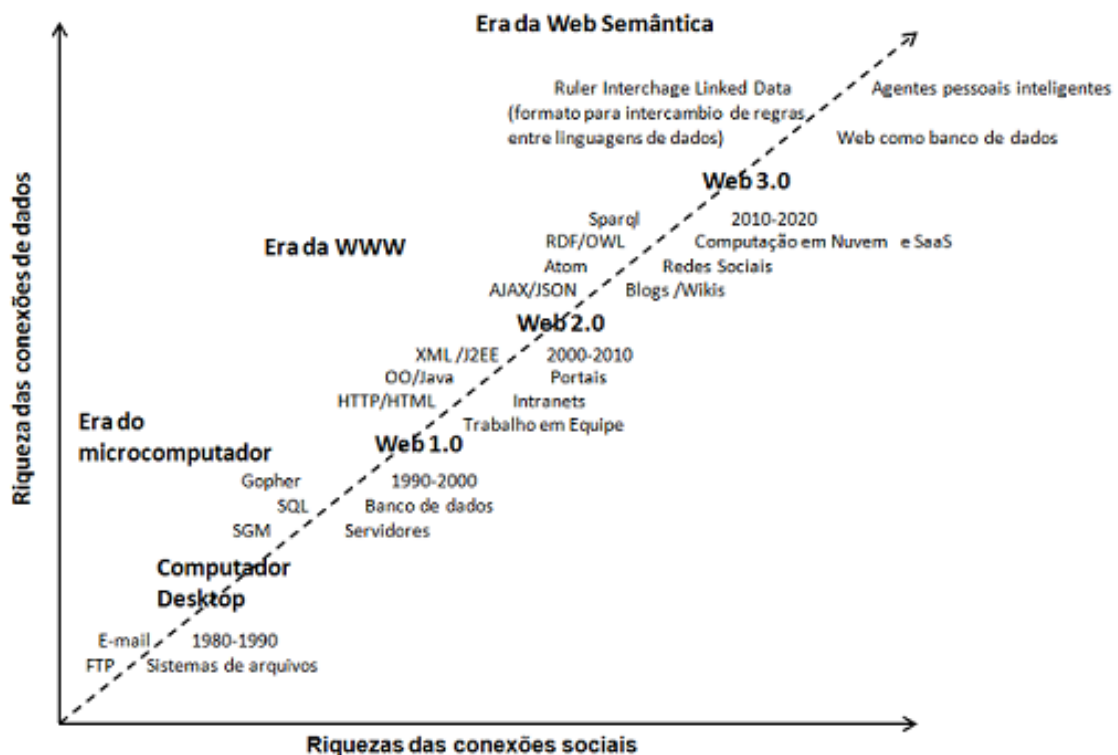


Figura 3-2 - Evolução dos ecossistemas tecnológicos
 Fonte: Pollock (apud Lafuente, 2011)

Neste sentido, este trabalho apresenta o Estado da Arte nos elementos que podem compor ambientes aderentes aos “Espaços” de Berthold (2010) e suas relações com as mais modernas tecnologias indicadas na Figura 3-2, tal como RDF/OWL, Web como banco de dados, ATOM e AJAX/JSON. Essas terminologias serão observadas e apresentadas nos tópicos seguintes.

3.2 ESPAÇO DE INFORMAÇÕES: FERRAMENTAS DE ONTOLOGIAS

Nos estudos sobre ontologias em geral, são citadas ferramentas sistêmicas distintas, em geral em pares, tanto para o desenvolvimento (modelagem) de uma ontologia própria local quanto para alinhamento e guarda da ontologia final comum (repositório). Porém, para o entendimento desses processos se faz necessário o entendimento dos modelos das linguagens representação de ontologias, como tratado.

3.2.1 Linguagem para Representação de Ontologias

Uma linguagem para representação de ontologia é uma linguagem formal utilizada para codificar a ontologia. Neste sentido, diversas linguagens são apresentadas para descrever e guardar ontologias de forma padrão.

O aspecto desafiador do problema é a existência de diversos padrões de metalinguagem para representação do conhecimento. Neste sentido, são considerados como melhores candidatos os padrões definidos pela ISO, W3C e OGC.

Segundo Lafuente (2011), as primeiras inovações de linguagens para representação do conhecimento, “que hoje podem ser definidas como padrões, surgiram do projeto Darpa Agent Markup Language (DAML)”, programa resultante de esforços do Departamento de Defesa do EUA em 2000. Esse projeto deu à luz para criação do Marco para Descrição de Recursos (*Resource Description Framework - RDF*) e, que redundou posteriormente na Linguagem de Ontologia para Web (*Web Ontology Language - OWL*).

O RDF é uma família de especificações oriundas do *World Wide Web Consortium - W3C* (W3C, 1999), tendo em vista o projeto de modelos de metadados, e que passaram a ser utilizadas como um método geral para a descrição conceitual ou modelagem de informação, usando uma variedade de formatos de sintaxe.

A especificação do RDF proposta pela W3C em 1999 aponta que a mesma pode ser usada "como recurso de descoberta para fornecer melhores capacidades para motores de pesquisa, na catalogação para descrever o conteúdo e as relações de conteúdo disponível em um determinado sítio, página ou biblioteca digital, para facilitar a partilha de conhecimentos e troca usando agentes de software inteligentes, na classificação de conteúdos, ao descrever coleções de páginas que representam um único documento lógico, para descrever os direitos de propriedade intelectual de páginas da Web, e para expressar as preferências de privacidade de um usuário, bem como as políticas de privacidade um sítio Web" (W3C, 1999).

Já a OWL é uma linguagem de marcação semântica para definir e instanciar ontologias na Web tomando por base os elementos constituintes de uma ontologia já apresentados (classes, axiomas, relações e instâncias), e foi desenvolvida como uma extensão do vocabulário RDF. A OWL se tornou uma recomendação W3C em 2004 (W3C, 2004).

O RDF e o OWL fazem parte de uma camada arquitetural de padrões definidos pela W3C, especificados da sintaxe até a semântica, conforme representado na Figura 3-3:

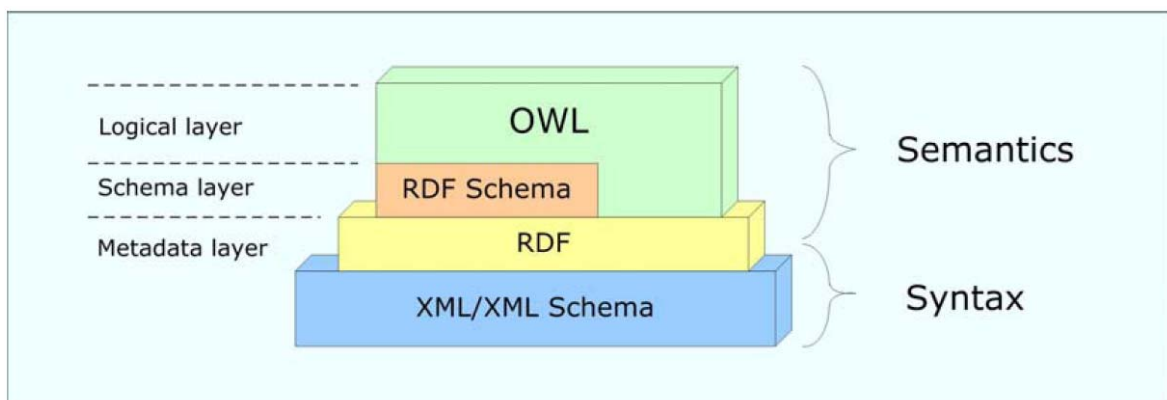


Figura 3-3 - OWL e RDF na arquitetura da Web Semântica.

Fonte: Djuric et al., 2005.

Da especificação W3C (W3C, 2004), este trabalho extrai a definição dos padrões que compõem a pilha de recomendações ilustrada na Figura 3-1:

- O XML fornece a sintaxe básica para documentos estruturados, mas não impõe restrições semânticas sobre o significado desses documentos.
- O *XML Schema* é uma linguagem para restringir a estrutura de documentos XML, acrescentando à sintaxe dessa o conceito de tipos de dados.

- O RDF é modelo de dados para objetos e declaração das relações entre eles, fornecendo uma semântica simples e uma representação através de sintaxe XML.
- O *RDF Schema* é um vocabulário para descrever propriedades e classes de recursos RDF, com uma semântica para hierarquias de generalização de tais propriedades e classes.
- O OWL é uma camada superior (*follow-on*) do RDF que incorpora mais riqueza de semântica ao modelo ("mais vocabulário") por adicionar outras propriedades às classes, aos seus elementos, nas relações entre as classes, como informações de cardinalidade (ex. "somente um"), como tipagens mais ricas, como características específicas (ex. simetria), e como o uso de classes enumeradas.

A Figura 3-4 ilustra a linha no tempo da evolução dessas linguagens e das versões desses padrões:

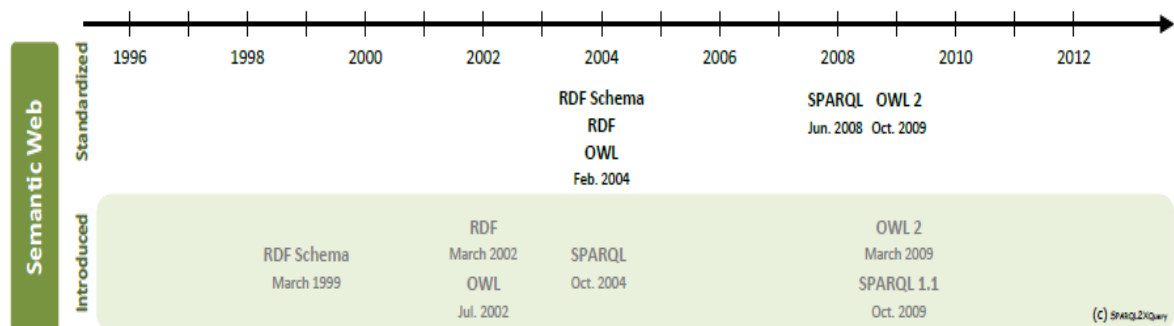


Figura 3-4 - Evolução no Tempo das Linguagens de Representação.
 Fonte: Bikakis et al., 2012.

Nessa evolução destaca-se o OWL por ter mais versatilidade dada sua maior capacidade de para expressar significado e semântica do que o XML, o RDF e o RDF Schema (Djuric et al., 2005). A especificação W3C estabelece ainda três principais modelos de sub-linguagens estendidos do OWL segundos requisitos ou interesses diferenciados de comunidades: a OWL Lite, que suporta principalmente as necessidades de definição de uma hierarquia de classificação e de restrições simples; a OWL DL, que suporta o máximo de expressividade no modelo, mantendo completude computacional, isto é, onde todas as conclusões podem ser avaliadas computacionalmente, e a decidibilidade, isto é, onde todos os processos de computações terminarão em tempo finito; e a OWL Full, que é destinado a usuários que querem o máximo de expressividade e da liberdade sintática do RDF sem

garantias computacionais, isto é, onde o tratamento computacional de todas as possibilidades é improvável (W3C, 2004).

O RDF e a OWL são linguagens de representação do conhecimento que mais influenciaram os modelos de construção da Web Semântica, sendo o RDF “utilizado como linguagem semântica para descrever dados, metadados e outras linguagens”, e o OWL “como extensão do RDF para representar conhecimentos mais ricos e complexos sobre informações, grupos de informações e relações entre elas” (Lafuente, 2011).

Com esses modelos é possível a navegação dentro de um ambiente de conceitos até seus conteúdos, de forma que esses possam ser pesquisados ou estruturados a fim de obter novos conhecimentos.

Outras linguagens para formalizar conhecimento tais como CycL, F-Logic, LOOM, KIF, Ontolingua também são representativas (Su and Iiebrekke, 2006) porém não serão consideradas nesse trabalho.

Assim, o conhecimento pode ser representado de forma ontológica através da semântica oferecida pelos modelos RDF e OWL em geral. Ainda, o uso padrões de representação facilita o processo de fusão de ontologias pois permite a interoperabilidade entre as ferramentas de desenvolvimento de ontologias junto a esses padrões.

A seguir serão analisadas as ferramentas especificamente que usam ou realizam os processos de alinhamento e fusão de ontologia segundo os padrões de linguagem RDF e OWL.

3.2.2 Ferramentas de Ontologia

Para a construção de ontologias, existem ferramentas conhecidas, cujos aspectos principais são a capacidade de capturar ontologias externas, a edição de ontologias através de linguagens de representação padrões, a geração e registro de ontologias de domínio ou de nível superior em um repositório centralizado, e a manutenção dessas ontologias em repositórios com opções para alinhamento e fusão (Noy e Musen, 2002).

Neste sentido, sobre ferramentas para ontologia, deve ser considerado dois aspectos: o da construção de novas ontologias e da junção de várias ontologias em uma ontologia comum (*upper-level ontology*) em um repositório sistêmico centralizado.

Segundo Noy (2002), pode se considerar então dois tipos de ferramentas para uso:

- As ferramentas para o desenvolvimento de ontologias, que são editores de ontologias que permitem aos usuários definir novos conceitos, relações e instâncias. Estas ferramentas geralmente têm capacidades para importação e extensão de ontologias existentes e podem incluir o desenvolvimento gráfico, recursos de busca e de verificação de restrição. As ferramentas para o desenvolvimento de ontologias têm como aspecto fundamental o uso de padrões de linguagem de representação que permitem que a mesma seja interoperável com outras ferramentas de desenvolvimento e a capacidade de interligar as ferramentas de guarda de ontologias de forma a realizar os processos de alinhamento e fusão.
- As ferramentas para manutenção de ontologia em um repositório, com opções de alinhamento e fusão de ontologias como extensões das ferramentas de desenvolvimento. São ferramentas que procuram manter o registro de ontologias e procedem as opções de alinhamento e fusão de forma automática ou semiautomática, buscando e analisando correspondências potenciais segundo seus algoritmos (*matchers*) específicos.

Existe uma forte relação na escolha de ferramentas para estes dois processos, pois em geral as ferramentas de desenvolvimento e para formação do repositório apresentam certo grau de integração, tornando dependente as suas escolhas. Mesmo que as teorias sobre como avaliar esses tipos de ferramentas não estejam bem articuladas, já existem vários *frameworks* para avaliar ferramentas de desenvolvimento de ontologias (Noy, 2002).

Membros do Grupo de Interesse Especial em Ferramentas para Ontologia (*Special Interest Group on Tools/Ontology – SIG Ontology*) projetaram na década de 2000 um extenso conjunto de critérios para avaliar as ferramentas para ontologia, que foram aplicados por Noy (2002) para comparar uma série de projetos. De forma prática, esses aspectos que estes frameworks comparam incluem:

- A interoperabilidade com outras ferramentas e a capacidade de importar e exportar ontologias em linguagens de representação diferentes;
- A expressividade do modelo de conhecimento;
- A escalabilidade e extensibilidade;

- A disponibilidade e capacidades de serviços de inferência;
- A usabilidade das ferramentas.

Esses serão alguns dos fatores subjetivos que serão tratados quando da análise de ferramentas para ambientes ontológicos junto a Prova de Conceito deste trabalho.

Noy e Musen (2002) e Abels et al. (2005) apresentam estudos sobre a avaliação de ferramentas para a construção, alinhamento e fusão de ontologias. Mas na análise desses estudos algumas ferramentas apresentadas já foram descontinuadas ou substituídas por projetos com outras designações. Em outros casos, as ferramentas se encontram em domínio que não permite o uso público da mesma para uma prova de conceito.

Atualizando os diversos estudos citados, é apresentado na Tabela 3-1 com um resumo das ferramentas que permitem a edição e alinhamento de ontologias em ambientes próprios ou que são apenas *plug-ins* de ferramentas existentes, e suas características principais:

Tabela 3-1 – Ferramentas de Alinhamento e Fusão
Fonte: Ehrig e Staab, 2004; sítios específicos

Abordagem / Solução	Linguagens Utilizadas	Estratégia Utilizada	Elementos Estruturais Utilizados	Nível de Automação	Vantagens e Desvantagens
HMatch (2.0)	OWL	Lingüística Contextual	classes, propriedades	semi-automático ou automático	Configuração Dinâmica. Plug-in do Protege.
S-Match	RDF, OWL	Lingüística Raciocínio	classes	semi-automático ou automático	Foco em esquemas de dados ou modelos taxonômicos. Diversos algoritmos.
MapOnto	OWL	Lingüística	classes, instâncias, <i>data models</i>	semi-automático	Foco em modelo de dados e esquema de dados
COMA 3.0	RDF, OWL	Lingüística Heurística Raciocínio	classes, propriedades, instâncias	semi-automático	Multiplicidade de matchers e novos matchers
Neon Toolkit / Align API	RDF, OWL	Lingüística Heurística Raciocínio	classes, propriedades, instâncias	semi-automático ou automático	Conjunto de ferramentas integrado. Possibilita o uso em várias de ontologias.

Protégé OWL / Protégé-OWL API 4.0	RDF, OWL	Lingüística Heurística Raciocínio	classes, propriedades, instâncias	semi- automático ou automático	Permite: editar e visualizar as classes, propriedades e regras no formato; executar analisadores; registrar elementos OWL com marcadores de Web Semântica; e mesclar ontologias.
--	-------------	---	---	---	--

No detalhamento de informações adicionais específicas das ferramentas listadas na Tabela 3-1 temos:

- **HMatch:** O HMatch é uma ferramenta para a alinhamento dinâmico de ontologias distribuídas em diferentes níveis de profundidades, e que permite a combinação de algoritmos diferentes de forma a atender um processo intensivo de alinhamento com uma ampla gama de métricas adequadas para lidar com diferentes cenários. Uma função de afinidade contextual é fornecida pelo HMatch e que garante uma medida de similaridade tendo em conta as características contextuais dos conceitos da ontologia, que incluem propriedades, relações semânticas com outros conceitos, e valores de propriedade. Outra função implementa a afinidade linguística, pela similaridade de textos nos conceitos (classes). O HMatch tem se desenvolvido no âmbito da projeto Helios visando apoiar o compartilhamento do conhecimento através de ontologias endereçáveis com a recuperação de conteúdo em redes ponto-a-ponto de ontologias. (HMatch, 2006).
- **S-Match:** O S-Match é um framework de alinhamento semântico que utiliza o RDF como linguagem básica e fornece vários algoritmos de correspondência e facilidades para construções de novas ontologias. Atualmente S-Match contém várias implementações de algoritmos de correspondência semântica e apresenta dois modelos de saída: um simplificado (compacto) para dar mais facilidade aos usuários e outro retratando uma análise mais completa. (S-Match, 2010).
- **MapOnto:** O MapOnto é um projeto implementado com o foco no mapeamentos semânticos entre diferentes modelos de dados, por exemplo, esquemas de banco de dados, esquemas conceituais e ontologias. É uma ferramenta que opera interativamente e de forma semiautomática. Começando com um conjunto de simples atributos para correspondências, a ferramenta analisa semântica nos dois modelos de entrada e, em seguida, gera um conjunto de representações da relação

semântica entre os dois modelos. A lista final apresenta as lógicas ordenadas pelo mapeamento considerado "mais razoável". (MapOnto, 2005)

- COMA 3.0: A ferramenta COMA 3.0 é uma ferramenta que estende o protótipo anterior COMA de alinhamento de ontologias e utiliza hierarquias de classe OWL e RDF para suportar esse processo. A semelhança entre dois elementos, sua similaridade, é expressa como a “distância na árvore taxonômica”. Apresenta diversos algoritmos de alinhamento (*matchers*) que podem ser utilizados de maneira combinada. Tem um forte potencial, se bem configurado (Aumuller, 2005; COMA 3.0, 2005).
- Neon Toolkit / Align API: A Align é uma API Java que permite o alinhamento e compartilhamento de ontologias, utilizando todos os elementos básicos da ontologia (tal como classes, instâncias, propriedades) e a linguagem de representação RDF. Uma das suas implementações correlatas mais modernas são o conjunto de ferramenta do projeto Neon, um moderno ambiente para ontologias, baseado na plataforma Eclipse e que usa modelos do Align API. (Neon, 2006).
- Protégé OWL 4.0: O Protégé é a mais recente versão da ferramenta desenvolvida na Universidade de Stanford para a aquisição de conhecimento. Ele tem milhares de usuários em todo o mundo e está disponível gratuitamente para download e fornece uma interface gráfica e interativa para projeto de ontologias e formação de base de conhecimentos. Os desenvolvedores de ontologias podem acessar informações relevantes rapidamente sempre que necessário, e pode usar a manipulação direta de navegar e gerenciar uma ontologia. Sua gestão se dá através de árvores permitem a navegação rápida e simples através de uma hierarquia de classes. O modelo de representação do conhecimento inclui suporte para as classes e a hierarquia de classes com múltiplas heranças e especificação para modelos predefinidos e arbitrários. Além de interface amigável, duas outras características importantes distinguem o Protégé da maioria dos ambientes de edição de ontologia: a sua escalabilidade e extensibilidade. Seu uso envolve dois componentes principais: (1) uma retaguarda de banco de dados para armazenar e consultar as ontologias e (2) um mecanismo de cache para habilitar o carregamento de novos modelos uma vez que o número de modelos na memória excedeu o limite de memória.

No que se refere às ferramentas de infraestrutura em geral, isto é, da construção a guarda e disponibilização em repositórios, foi utilizado o mesmo princípio de análise de referencial e atualização das ferramentas de alinhamento e fusão de ontologias.

Nessa análise foram utilizados os referenciais de Noy and Musen (2002), OntoWeb (2002), Neon Toolkit (2006), Align Api (2010), Mike 2.0 (2010), e o resultado revisto é apresentado na Tabela 3-2:

Tabela 3-2 – Infraestrutura - Ferramentas de Edição e de Repositório
 Fonte: Noy, 2002; OntoWeb, 2002; Neon Toolkit, 2006; Align Api, 2010; Mike 2.0, 2010

Ferramenta	Vantagens e Desvantagens
Alignment API e Server / Neon Project	<p>O Align API junto com Align Server são ferramentas de alinhamento e implementação para expressar e compartilhar ontologias. Ele foca de forma privilegiada o atingimento da interoperabilidade entre sistemas heterogêneos dentro da Web Semântica. Sua linguagem é o RDF, o que o torna extensível.</p> <p>O Align API define cinco interfaces principais, o <i>Ontology Networks</i>, o <i>Alignment</i>, o <i>Cell</i>, o <i>Relation</i> e o <i>Evaluator</i>; interfaces essas que propõem executar em conjunto os serviços de: armazenamento, localização e compartilhamento dos alinhamentos realizados; enfileiramento de algoritmos de alinhamento melhorando sucessivamente o resultado; a manipulação de ontologias por <i>thresholding</i> e <i>hardening</i>; a geração de saídas desse processamento com a apresentação das necessidades de transformações, axiomas, regras; e a geração de testes e comparação dos níveis de alinhamento usando os conceitos de diferenciação, <i>precision</i> e <i>recall</i>. (Align API 4.3, 2011)</p>
COMA 3.0	<p>COMA 3.0 é uma ferramenta de ontologia focada no alinhamento e fusão de ontologias. Ela estende os protótipos anteriores COMA e COMA++ com uma gestão do fluxo de trabalho e recursos adicionais como de fusão de ontologias.</p> <p>Além disso, oferece uma infraestrutura abrangente para resolver grandes problemas de alinhamento. A interface gráfica oferece uma variedade de interações, permitindo ao usuário influenciar todo processo de realização de processo. (COMA 3.0, 2005)</p>
Protégé OWL / Protégé-OWL API 4.0	<p>Uma das vantagens principais da arquitetura do Protégé é que o sistema é construído de uma forma aberta e modular. Sua arquitetura baseada em componentes permite que os desenvolvedores de sistemas adicionem novas funcionalidades através da criação de plug-ins adequados, criados no Protégé Plugin Library³. A Universidade de Stanford oferece ainda o Protégé-OWL API, uma biblioteca <i>open source</i> Java para a RDF e OWL. A API fornece classes e métodos para carregar e salvar arquivos OWL, para consultar e manipular modelos OWL de dados, e realizar os processos (de raciocínio). Além disso, a API é otimizada para a implementação de interfaces gráficas de usuário (Protégé, 2012).</p>
Apollo	<p>O Apollo é um aplicativo amigável para ontologias implementado em Java, que suporta todos os elementos básicos de modelagem de conhecimento através de ontologias: classes, instâncias, funções e relações. Uma representação hierárquica de ontologias (classes e instâncias) é controlada em um menu em árvore em que, uma vez selecionado um item, esse é apresentado em detalhes. Uma entrada do tipo "folha de cálculo" permite a adição de instâncias.</p> <p>A verificação de consistência é feito durante a edição, como por exemplo, o uso de classes não definidas previamente (Apollo, 2012).</p>

Cabe citar que neste trabalho, para dar suporte a análise do conjunto de plataforma, é necessário que alguns os pontos importantes sejam atendidos:

- No seu desenvolvimento, o ambiente deve permitir a especificação de princípios e regras que serão estabelecidos;
- Na sua guarda em repositório o ambiente deve permitir de forma integrada o alinhamento e fusão de ontologias e, se for possível, o estabelecimento de diferentes algoritmos (*matchers*) para realizar esse processo

Além disso, foram consideradas preferencialmente ferramentas que suportem altos princípios de inferência associados com uma interface semiautomática de integração das ontologias. Neste sentido, os aspectos de disponibilidade e capacidades de serviços de inferência e de usabilidade das ferramentas através de uma interface amigável são considerados essenciais para a viabilidade e eficácia de um sistema que permita realizar a atividade de correspondência no processo de fusão de ontologias.

3.3 ESPAÇO DE DADOS: SERVIÇOS DE DADOS

Como modelos de entrega de dados distribuídos, os serviços de dados se apresentam como um conjunto de repositórios distribuídos de dados, orientado segundo padrões interoperáveis, e utilizável por diversas organizações. Ele dispõe os serviços de dados cumprindo todos os aspectos de geração de registros de informação segundo requisitados por modelos de metadados. (Mochmann, 2009).

Chaudhuri et al. (2011) aponta que os processos baseados em serviços de dados, que inicialmente consistiam de um simples armazenamento de chaves e valores, foram melhorados para suportar as funcionalidades de bancos de dados relacionais, na forma de um serviço hospedado ou no fornecimento de serviço de dados. Como fortes exemplos temos o Microsoft SQL Azure e os serviços de dados em nuvem do Amazon EC2.

Utilizando os princípios do protocolo REST, algumas organizações têm desenvolvido serviços genéricos para a realização de consulta de dados tabulares. Como exemplos temos a Google com o Google Data - GData, a Microsoft com o Open Access Data Protocol - OData e o Yahoo com o DataRSS (Kansa e Bissell, 2010). Esses padrões são baseados em tecnologias XML e estão associados com as iniciativas de computação em nuvem.

Destaque-se dos exemplos citados a implementação da Microsoft, o Protocolo Aberto de Acesso a Dados (*Open Data Protocol – OData*).

Em sua mais recente implementação para ajudar a suportar a visão de plataforma de dados, a Microsoft juntou-se Citrix, IBM, Progress Software, SAP e WSO2 em conjunto apresentar uma proposta para a OASIS, visando padronizar um modelo de entrega interoperável de dados via Web, o *Open Data Protocol - OData*, um padrão aberto baseado no protocolo de acesso a dados REST (Microsoft SQL Server Team, 2012)

Open Data Protocol (OData) é um protocolo Web para consulta e atualização de dados que fornece a possibilidade de liberar o acesso a dados dos silos existentes em aplicações, aplicando e desenvolvendo tecnologias da Web, como HTTP, *Atom Publishing Protocol* (AtomPub) - um elemento popular nas soluções ao estilo da Web 2.0 - e JavaScript Object Notation - JSON - um formato de troca de dados de baixo impacto desenvolvido para uso em soluções AJAX como uma alternativa para XML; de forma a fornecer acesso à informação a partir de uma variedade de aplicações, serviços e repositórios.

Segundo o sitio OData, esse protocolo proposto pela Microsoft “é consistente com a forma como a Web funciona pois ele associa o acesso a dados com URIs para a identificação e recuperação de recursos, mantendo o compromisso com o HTTP e com o protocolo REST visando estabelecer um mais alto nível de integração de dados e interoperabilidade entre clientes, servidores, serviços e ferramentas”.

Fatland (2011) aponta que o OData pode ser observado como uma escolha de formato de dados com valor intrínsecos e extrínsecos no contexto dos desenvolvimentos atuais da tecnologia da informação.

O OData está encontrando uma ampla adoção como candidato a exposição de dados em situações de consulta e recuperação. Das principais utilizações citadas no sitio OData, destaca-se o seu uso pelo *Open Government Data Initiative – OGDI*, expondo uma coleção de dados de agências de governo dos EUA, pelo *Open Science Data Initiative - OSDI* que associado ao OGDI usa a plataforma de serviços Azure (Azure Services Platform) para facilitar a publicação de dados científicos das agências governamentais, pelo catálogo do eBay, e pelos projetos *vanGuide* e o *Vancouver Street Parking* que disponibilizam dados sociais e das ruas de Vancouver no Canadá.

Segundo a definição da especificação da Microsoft para o protocolo, O OData funciona com mecanismos de requisição baseados no protocolo REST, onde a URI usada é composta de três partes: a rota do serviço (*Service Root URI*), a rota do recurso (*Resource Path*) e as opções de consulta (*Query String Options*). A Figura 3-5 ilustra essa representação:

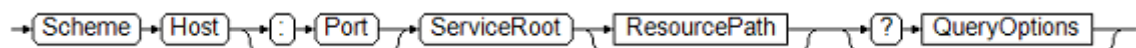


Figura 3-5 - URI OData
Fonte: Sitio OData.org

Na Rota do Serviço está especificado o serviço específico, que geralmente está associado a um banco de dados específico. A Rota de Recursos detalha esta especificação apontando uma tabela ou uma coleção específica, podem especificar um predicado de pesquisa.

A opções de consulta suporta 3 tipos de opções / parâmetros: opções de sistema (*System Query Option*), opções de configuração (*Custom Query Options*) e de parâmetros operacionais para o serviço (*Service Operation Parameters*).

Desses, a recuperação de dados (consulta) pode ser especificada utilizando-se apenas as opções de sistema como será detalhado.

As opções de sistema para requisição de dados OData, que são repassados como parâmetros da URI, são apresentadas na Tabela 3-3:

Tabela 3-3 – Parâmetros de Query do OData
Fonte: Sitio Microsoft OData

Tipo	Parâmetro	Operação
Order By	\$orderby	Especifica o(s) campo(s) a ser(em) utilizado(s) na ordenação da coleção de entradas.
Top	\$top	Identifica uma quantidade de elementos no subconjunto das entradas a serem apresentados.
Skip	\$skip	Indica quais as entradas associadas devem ser representadas in-line
Filter	\$filter	Especifica que um filtro na seleção em um modelo específico do protocolo, diferente do formato SQL.
Expand	\$expand	Especifica que um campo deve ser desdobrado em seus valores relacionados.
Format	\$format	Especificado o formato de saída no qual são aceitos os valores Atom (application/atom+xml), XML (application/xml), JSON (application/json), ou outros.
Select	\$select	Especifica os campos que deverão constar na saída XML.

Uma chamada exemplo para um serviço de dados no formato OData - serviço este definido como "Vendas", listando a tabela de "Produtos", ordenada pelo "Nome", somente os 10 primeiros, com os registros cujo campo de valores sejam maiores que 200, apresentado apenas os campos de "Nome" e de "Valor", e cuja saída seja no formato JSON - tem a seguinte formação: [http://services.odata.org/OData/Vendas.svc/Produtos?\\$orderby=Nome&\\$top=10&\\$filter=Valor ge 200&\\$select=Nome,Valor&\\$format=json](http://services.odata.org/OData/Vendas.svc/Produtos?$orderby=Nome&$top=10&$filter=Valor ge 200&$select=Nome,Valor&$format=json)

Como exposto na Tabela 3-3, os formatos de saída admitidos por provedores de serviços de dados podem ser o Atom, que tem o foco na apresentação de várias informações para cada dado apresentado; o JSON, um formato de saída mais simples interoperável com ambientes de Serviços Web do Apache; e o XML, na forma de texto plano e direto, campo a campo.

Assim, um modelo de atendimento moderno para diversas organizações deve prover o consumo de seus dados conforme todos esses padrões. Porém, indiferente ao uso desses padrões ou outro qualquer, no momento a maioria das soluções tradicionais ainda suportam apenas a entrega de dados como serviços por saídas no formato XML plano, segundo esquemas definidos pelas organizações que os fornecem (Pan et al., 2011).

Utilizando esses princípios de interoperabilidade do ATOM, JSON e/ou XML, temos como exemplo o Microsoft Azure SQL Server (Microsoft SQL Server Team, 2012) E códigos em PHP de um provedor de serviços ODATA disponibilizados no sitio de disponibilização de códigos no regime *open-source* Codeplex.com sob a designação de PHP OData Producer Library (Codeplex.com, 2012) e que atende a todos os três formatos de saída.

Casters (2011), apresenta uma proposição de geração de saídas de dados como Serviços Web utilizando a ferramenta de ETL da suíte do Pentaho, o Pentaho Data Integration - PDI. Esse exemplo utiliza o módulos de disponibilização de execução de processos do PDI via Web, o programa Carte, para acessar fontes de dados segundo o padrão da plataforma, isto é, por JDBC, e disponibilizá-las como um Serviço Web no formato XML ou JSON.

Porém, a escala de consultas *ad-hoc* exige altos requisitos de desempenho nos provedores de serviço de dados, pois as consultas SQL podem complexas ao ponto de tornar esse um recurso massivo e intensivo. Nesse sentido será abordado os aspectos de desempenho estudados que são intrínsecos a esse modelo de serviços.

3.3.1 Aspectos de Desempenho em Serviço de Dados

Mesmo que algumas pessoas acreditam que o desempenho do consumo de dados de um XML plano é indefinível (King, 2002), os processadores atuais “sofrem” quando o tamanho do documento XML se torna muito grande ou existe um maior número de referências de nós.

King (2002) apresenta em seu trabalho sua experiência com serviços de dados trabalhando com transformações de XML para registros de dados que reporta uma sobrecarga de processamento de XML entre 0,1ms à 0,5ms por registro em relação a carga de registros usando o modelo JDBC.

Em 2005, Jeff Lamb (apud Geer, 2005 e LaMonica, 2005), chefe de tecnologia da Leader Technologies, cita que o "XML é extremamente perdulário na quantidade de espaço que ela precisa usar para a quantidade de dados verdadeiros que ele está enviando".

Esse problema é inerente ao desempenho de serviços, seja baseado em SOAP ou em REST, ainda ocorre, principalmente quando o pacote de mensagem a ser transportado contém um volume excessivo de informações (Edmonson, et al., 2004). Assim, o consumo de dados deve buscar que o menor volume de dados com estruturas menos complexas e profundas, isto é, um XML mais “eficiente”.

Segundo Dav and Padmanabhuni (2006), técnicas de compressão XML para superar problemas de desempenho das arquiteturas orientadas a serviços se concentram:

- na existência de hardware dedicado para processamento XML,
- na representação do XML cache,
- na compactação por software e no uso de padrões para XML binário.

Das representações em cache, Küngas e Dumas (2011) apresentam uma proposta de cache no provisionamento de serviços de dados Web baseados em SOAP (que também é válido para REST, grifo do autor). A solução proposta se baseia em "um esquema de cache singular que identifica eficientemente acertos de cache e oferece recursos refinados de configuração no apoio a definição de políticas de cache no serviço e nos níveis de operação".

Da compactação de dados, trabalhos foram desenvolvidos por Ogbuji (2004), Geer (2005) e LaMonica (2005) utilizando os princípios de XML Binários, para incorporar ao processamento XML rotinas de compactação (como exemplos temos o GNU gzip e o PK-ZIP) nos pacotes SOAP.

Visando manter o conteúdo semântico do cabeçalho do pacote SOAP, e assim, garantir os princípios de interoperabilidade desses pacotes, Ogbuji (2004) apresenta somente duas opções: o uso das facilidades de *XML Binary* em anexos e/ou o uso de tipagens codificadas específicas, tal como a Base64, para a inclusão no corpo principal da mensagem. Infelizmente, a codificação Base64 desfaz alguns dos efeitos da compressão pois ele é maior do que o original gerado por modelos binários puros na proporção de 4:3 (Ogbuji, 2004).

3.3.2 Aspectos de Segurança de Serviços de Dados

Tal como acontecem com todos os serviços sensíveis, normas de segurança devem ser aproveitadas para autenticar o cliente e proteger a integridade e a confidencialidade das mensagens. Além de proteger a própria mensagem, é necessário também proteger os recursos através de serviços de habilitação.

Isto é realizado permitindo que os clientes diferentes tenham diferentes capacidades com base no nível do prestador de serviços de confiança no cliente. Quanto maior o nível de confiança, mais liberdade o provedor pode permitir que o cliente utilize seus recursos. Os níveis de confiança incluem não só o direito do cliente para ver dados específicos, mas também a confiança no cliente que não vai fazer mal para o provedor, intencionalmente ou por ignorância.

Porém, se dados como serviços devem tratados sob os princípios de segurança que são afetos ao uso de recursos modernos de serviços, esses são em geral embutidos nas plataformas de serviços de dados e não geram consequências maiores em testes de sua performance. Assim, na prova de conceito estabelecida para esse trabalho, não serão tratados esses aspectos.

3.4 NÚCLEO DA PLATAFORMA: AMBIENTE DE BI

Em arquiteturas de BI existentes, os principais componentes responsáveis pelas respostas às consultas que acessam as estruturas de dados multidimensionais é o servidor OLAP e a interface analítica, como representado na Tabela 2 dentro da camada *Application* e *Presentation* respectivamente, que será tratado.

3.4.1 Ambientes de Interface Analítica de BI

Nos fins de 1996, “o setor corporativo passou a se interessar por soluções de BI de forma mais expressiva, quando o conceito começou a ser espalhado como um processo de evolução dos Sistemas de Informações Empresariais (*Executive Information Systems - EIS*), um sistema criado no final da década de 70, a partir de trabalhos desenvolvidos pelos pesquisadores do MIT (*Massachusetts Institute of Technology - EUA*)” (Primak, 2008).

Com a evolução natural da tecnologia, ambientes de BI ganharam maior abrangência e importância, embutindo uma série de ferramentas aglutinadas ao próprio EIS e classificadas como Sistemas de Suporte a Decisão (*Decision Support Systems – DSS*), agregando ainda recursos de planilhas eletrônicas autogeradas, geradores de consultas e relatórios manipuláveis diretamente pelos usuários finais, mineração e pesquisa de dados com suporte estatístico, ferramentas de análise com realização de operações multidimensionais *on-line*, múltiplos gráficos, entre outras.

Porém, indiferente da solução desenvolvida, um fator comum nas interfaces de solução de BI (interfaces analíticas) é o de prover um modelo de navegação através de operações multidimensionais sob as dimensões do “cubo multidimensional” de forma a organizar, filtrar e/ou definir o nível de agrupamento das informações selecionadas segundo as métricas a serem analisadas.

Neste sentido, as ferramentas de processamento analítico junto as soluções de BI permitem a criação de análises (i.e., relatórios e cubos de dados) obedecendo a diferentes perspectivas sobre os dados disponíveis, porém, as funcionalidades exploratórias oferecidas por essas aplicações analíticas limitam-se a um conjunto de funcionalidades genéricas de navegação e filtragem de dados. (Primak, 2008).

Como proposta para reformatação do modelo básico de resolução de consultas multidimensionais, além da reformulação no tratamento da navegação nas interfaces da solução de BI, é necessário o entendimento de outro componente presente nas soluções de BI, os servidores OLAP, que resolvem e “apresentam as informações multidimensionais (resultados das operações analíticas nos cubos multidimensionais, grifo do autor) para as ferramentas de acesso, análise, geradores de relatórios, planilhas e ferramentas de mineração de dados” (MacDonald, 2007). Sendo esse componente que realiza a lógica de solicitação as fontes de dados, se faz necessário dar atenção ao seu funcionamento para que se entenda o desacoplamento correto do acesso a dados.

Se as interfaces analíticas de soluções de BI são importante contribuição no processo de apoio a análise das informações para suporte a tomada de decisões gerenciais, as ferramentas OLAP (servidores OLAP) são os mais importantes componente dessas soluções.

3.4.2 Servidores OLAP

Os servidores OLAP são o recurso tradicional de implementação utilizados para resolução dos modelos multidimensionais. Esses são desenvolvidos como sistemas gerenciadores de banco de dados - SGBD, tal qual os SGBDs tradicionais que atuam sobre as bases de dados relacionais (Mahboubi et al., 2011).

O termo OLAP foi usado pela primeira vez por Edgar Frank Codd em 1993 visando definir um modelo de armazenamento de dados eficiente para atender as necessidades das aplicações de BI já que essas necessidades “não seriam solucionadas por bancos de dados relacionais”. Seu uso atual é muito difundido e representa um conjunto de produtos que tem a “a característica de trabalhar com dados através do uso de operadores dimensionais, possibilitando uma forma múltipla e combinada de análise sobre dados” (Barbieri, 2001; Primak, 2008).

Em geral, na representação das dimensões dentro do OLAP, os conceitos são organizados em uma hierarquia que caracteriza os diferentes níveis de detalhe das informações contidas. A funcionalidade OLAP é caracterizada pela análise dinâmica e multidimensional dos dados consolidados permitindo que as atividades do usuário final sejam analíticas e navegacionais.

Do seu funcionamento, servidores OLAP apresentam duas características básicas: a linguagem usada para interpretação e sua arquitetura. No que se refere a linguagem de interpretação, comumente os servidores OLAP tratam o SQL tradicional e/ou as consultas multidimensionais.

A Linguagem de Consultas Estruturadas (*Structured Query Language- SQL*) é um modelo tradicional “responsável pela entrada e saída das instruções executadas sobre o banco de dados” (Compolt, 1999). As consultas multidimensionais são realizadas através de uma "linguagem de expressões" que são denominadas de Expressões Multidimensionais (*Multidimensional Expression - MDX*). O MDX foi projetada pela Microsoft em 1998 como um padrão para consultas e troca de dados em fontes de dados multidimensionais e começou a ser utilizado comercialmente para dar suporte as consultas em bases de dados OLAP. (Blakeley and Pizzo, 1998).

Em relação a SQL, a linguagem de consulta MDX tem diferenças importantes:

- o SQL tradicional usa tabelas como fonte de dados e a saída é em forma de colunas.
- o MDX usa um cubo ou tabelas como fonte de dados e a saída da consulta é um cubo, com colunas e linhas de dados.

É comum que as soluções OLAP que utilizem a linguagem MDX sobre banco de dados tradicionais transformem a expressão MDX em uma série de comandos SQL tradicionais para recuperar e compor as respostas às consultas dimensionais como cubos (Pentaho, 2010).

Das arquiteturas de soluções Carvalho Jr (2006) elenca os diversos tipos conforme suas categorias:

- ROLAP - OLAP Relacional: Esse tipo de arquitetura utiliza banco de dados relacional, resultando em um maior acoplamento com fontes de dados OLTP. A consulta é enviada ao servidor de base de dados relacional e processada no mesmo (tabela de fato). Nota-se que o processamento OLAP se dá exclusivamente no servidor. Utiliza um SGBD relacional. É mais tolerante às mudanças de fontes de dados originais quando elas são OLTP, pois também utiliza OLTP.
- MOLAP - OLAP multidimensional: Aquela cujos dados são armazenados em um

banco de dados multidimensional em uma estrutura do tipo *Array* para prover um melhor desempenho das consultas. A implementação varia de acordo com a ferramenta OLAP utilizada, mas frequentemente é implementado em um banco de dados relacional, porém não na terceira forma normal. O acesso aos dados ocorre diretamente no banco de dados do servidor multidimensional, o utilizador trabalha, constrói e manipula os dados do cubo diretamente no servidor. (SGBD multidimensional)

- **HOLAP - OLAP híbrido:** Utiliza um processamento híbrido cuja forma de acesso aos dados é uma mistura de tecnologias onde há uma combinação entre ROLAP e MOLAP. Consegue combinar a capacidade e a escalabilidade das ferramentas ROLAP com o desempenho superior dos bancos de dados multidimensionais.

O mesmo autor ainda aponta outras implementações de menor utilização, a saber: o DOLAP ou Desktop OLAP que é uma variação de arquitetura OLAP criada para fornecer portabilidade dos dados e se obter uma redução do tráfego na rede; o WOLAP ou Web OLAP que é uma variação que utiliza um browser para acessar a ferramenta, o JOLAP que é uma API Java para servidores e aplicações OLAP Orientada a objetos; e o SOLAP ou *Spatial OLAP* que é uma união de aplicação de OLAP com Sistemas de Informações Geográficas (SIG).

Para suporte a seleção do elemento “Serviço de geração e resolução de estruturas OLAP” é elencado na Tabela 3-4 os servidores OLAP conhecidos:

Tabela 3-4 - Resumo de Arquiteturas destacadas de Soluções de BI.
Fonte: Holler, 2010; BI-Insider, 2011.

Ambiente de BI	Arquitetura do Servidor	Linguagem de Interrogação
Pentaho BI Suíte	ROLAP e MOLAP Obs.: A suíte é disponibilizada em uma versão da comunidade como software livre e uma versão Enterprise que é licenciada com custos.	SQL, MDX
Palo BI Server	MOLAP Obs.: os dados para uso no Palo são armazenados como cubos no servidor Palo MOLAP.	SQL, MDX
Jaspersoft	ROLAP	SQL, MDX
Excel / SQL Server OLAP	ROLAP e MOLAP	SQL, MDX
SAP Business Objects	ROLAP e MOLAP	SQL, MDX
Oracle BI	ROLAP, MOLAP e HOLAP	SQL e PL/SQL; OLAP DML
Microstrategy	ROLAP e MOLAP	SQL

Como já exposto, não existe uma definição rígida que limite o que é solução de BI, assim diversas soluções são consideradas nessa categoria. Da análise do mercado ferramentas existentes para a construção de soluções de BI, Holler (2010) aponta uma lista que, agregada ao posicionamento de mercado do site BI-Insider (2011), são elencadas na Tabela 3-5:

Tabela 3-5 - Resumo das mais destacadas soluções de BI
Fonte: Holler, 2010; BI-Insider, 2011.

Ambiente de BI	Descrição	Interface Analítica
Pentaho BI Suite (Jpivot / Mondrian)	Ferramenta <i>open source</i> de Business Intelligence, que fornece relatórios, análises de OLAP, integração de dados, mineração de dados e uma plataforma BI completa. Nessa ferramenta o Jpivot e o Mondrian OLAP Server já funcionam de forma integrada. (JPivot, 2010; Pentaho/Mondrian, 2011)	JPivot, STPivot, Saiku
Palo BI Suite	É composto do Palo OLAP Server, Worksheet Server and ETL Server e é normalmente é usado como uma ferramenta de BI atuando como interface do usuário através de interface com software de planilha. O servidor está Dito como disponível como <i>open source</i> , apresenta parte do seu código fechado. (Palo, 2011)	Palo for Excel (é um <i>add-on</i> para o Microsoft Excel em regime de software livre)
JasperSoft	JasperSoft OLAP é uma ferramenta de suporte à análise de dados de grandes conjuntos de dados e realização de consultas complexas. Os usuários podem interagir com os seus dados através de uma interface web amigável ou usando o Excel. (Japersoft, 2011)	Jasper Analysis
SQL Server 2005/2008	Ferramenta proprietária da Microsoft, associada ao seu banco de dados SQL Server 2005/2008, interpõem um servidor analítico (SQL Server 2005/2008 Analysis Services), que além de facilitar a criação de cubos (OLAP) armazena indicadores de desempenho (Microsoft 2010).	Excel Plugi
SAP Business Object	A solução SAP Business Objects - Business Intelligence (BI) pode alcançar resultados avançados, permitindo que qualquer pessoa na organização tenha o acesso de auto-serviço de informações relevantes. Um dos componentes é o SAP BusinessObjects Analysis, modelo OLAP que permite a capacitação de seus analistas de negócios para realizar a análise de dados multidimensional através de uma interface Web intuitiva por uma gama completa de funções analíticas. (SAP Business Objects, 2010).	Proprietária e add-on para Microsoft Excel
Oracle Business Intelligence Suite Enterprise	Oracle Business Intelligence Suite Enterprise é um conjunto abrangente de produtos para BI empresarial, e permite entregar toda a gama de capacidades de BI, incluindo painéis interativos, resultados de pesquisas <i>ad-hoc</i> , inteligência proativa, e alerta. Tudo em tempo real com inteligência preditiva, análise desconectada, e muito mais. Além de fornecer toda a gama de funcionalidades de BI, essa plataforma é baseada em uma arquitetura moderna Web Orientada a Serviços. (Oracle, 2011)	Proprietária

Microstrategy Suite (Analysis Services)	É uma interface integrada para o usuário. para alavancar informações, acionável pelo modelo relacional do Microsoft SQL Server e fontes de dados de cubos (multidimensionais). (Microstrategy, 2011).	Proprietária
---	---	--------------

Dessa relação e junto ao requisito de fácil adaptação destaca-se as plataformas existentes de código aberto: o Pentaho BI, Spago BI e Palo BI (Liu e Lou, 2010). Assim, neste trabalho serão utilizadas para seleção de plataformas de servidores OLAP e para os ambientes de interfaces analíticas apenas os elementos existentes nessas plataformas apontadas.

4 - PROPOSTA DE ARQUITETURA

Inicialmente, reportamo-nos aos desafios e motivação da proposta, que analisa que: se diversas organizações necessitam realizar o intercâmbio de conceitos e dados entre si, isso se dá através de mecanismos de carga (processos de ETL), partindo de organizações como origem e chegando a várias organizações de destino, de forma pouco amigável.

A Figura 4-1 ilustra a situação de quatro organizações, com suas camadas de BI convencional locais, realizando o intercâmbio de conceitos e dados de forma a alimentar suas arquiteturas locais de BI:

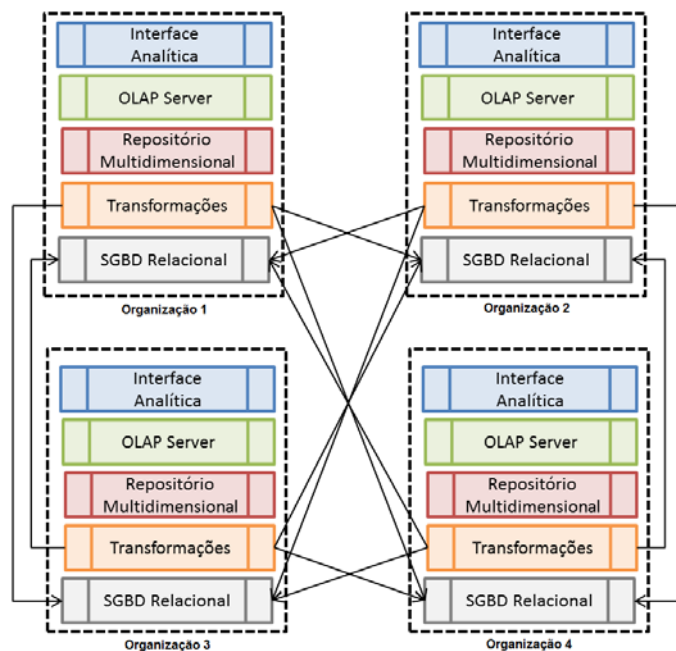


Figura 4-1 – Caso crítico de troca de informações e dados entre quatro organizações.

Dessa Figura, observa-se que: se uma organização altera a estrutura, a semântica, a sintaxe ou as instâncias de seus conceitos organizacionais e / ou alteram o formato ou a aditividade de seus dados, erros podem se propagar "em cascata" obrigando a interlocução entre todas as organizações de forma a recompor seus processos de carga.. Tanto maior é o impacto se o modelo for acrescido de novas organizações, que replica exponencialmente erros e inconsistências.

Além disso, se as cargas de dados ocorrem em momentos diferenciados é alta a probabilidade de que os dados sejam apresentadas com valores diferentes por cada BI convencional local em um mesmo momento, o que pode afetar a confiança dos

consumidores na qualidade de informações disponibilizadas pelas organizações envolvidas.

Assim, para enfrentar esses desafios apresentados e com base nos conceitos introduzidos por Berthold et al. (2010), Böhringer et al. (2010) e Nelson (2010), propõem-se um modelo de colaboração composto de dois ambientes – um ontológico e outro de serviço de dados - que serão integrados utilizando-se de mecanismos básicos de consultas genéricas em BI - um servidor OLAP e uma plataforma de interface analítica – compondo um ambiente único no qual diversos parceiros e / ou organizações podem construir comunidades de informação com o mínimo esforço.

Inspirado no modelo de duas camadas, este trabalho propõe uma arquitetura de *Business Intelligence* (BI) que segue o paradigma de *Dataspace* e *Infospace* de forma a melhorar a estrutura de aplicações convencionais de BI. Empregando ontologias na arquitetura proposta nesse trabalho é possível a especificação de uma solução colaborativa que permite a formação do *Infospace*. Ao mesmo tempo, a utilização de serviços de dados fornece uma solução para um *Dataspace* descentralizado que pode ser suportado por ambientes de computação em nuvem.

Com este desacoplamento, entende-se que benefícios podem ser atingidos, tais como:

- O atendimento unificado no cruzamento de métricas entre organizações;
- A construção automatizada dos modelos de disponibilização de informações de BI em nuvem para as organizações que participem de uma comunidade;
- O tratamento correto de dados e informações entre as organizações ou por terceiros, com a sua gestão dentro do limite das organizações sem replicações ou entendimento errôneos da informação;
- A redução no atraso associado com a evolução da semântica das métricas e dimensões de negócios de determinada organização evitando o impacto para as outras entidades participantes; e
- O aumento das possibilidades de associação de dados disponibilizados em ambientes distribuídos de forma pública ou comunitária.

Para apoiar a apresentação do modelo do ambiente integrado proposto e a exposição do desenvolvimento dessa arquitetura é representado a seguir os ambientes envolvidos e os elementos que devem ser adaptadas a partir de soluções existentes, bem como são tratadas as camadas tecnológicas e o ciclo de vida da solução proposta .

4.1 REPRESENTAÇÃO DA ARQUITETURA PROPOSTA

A arquitetura proposta considera o uso de plataformas de tratamento de ontologias e de serviços de dados como soluções para estabelecer o *Infospace* e *Dataspace*, respectivamente. Essas plataformas são integradas através de componentes comuns de uma plataforma de BI convencional adaptada para tal fim.

4.1.1 Ambientes

Na Figura 4-2 é ilustrado o limites do ambientes envolvidos, posicionado os componentes na estrutura geral da arquitetura proposta. Nesta figura, pode-se distinguir três ambientes: o **Ambiente do Usuário**, o **Ambiente de BI Colaborativo** e **Ambiente Organizacional**.

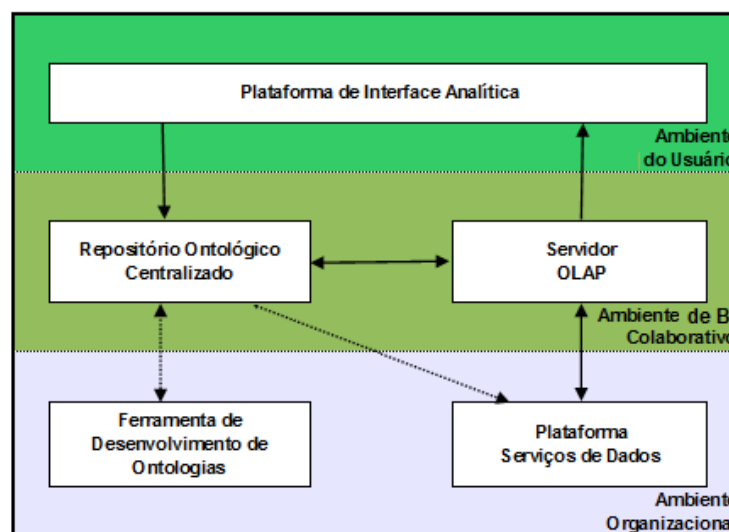


Figura 4-2 – Componentes da Arquitetura.

O **Ambiente do Usuário** é representada pela aplicação web cliente, é usado para realizar pesquisas de informação, associada com a camada *Presentation* na Tabela 2-2.

O **Ambiente de BI Colaborativo** é o ambiente prestador de serviços central da solução proposta. Este ambiente é associado com a camada de *Application* na Tabela 2-2.

O **Ambiente Organizacional** é composta pelas ferramentas de desenvolvimento de ontologias e de disponibilização de serviços de dados, que são providos por cada um dos parceiros. Desta forma, diferentemente da arquitetura de BI convencional, que acessa diretamente o repositório local gerado pelas intempestivas cargas internas e externas, propõem-se que os conceitos (dimensões da informação) sejam atualizados e os dados sejam disponibilizados unicamente dentro dos limites de cada organização.

Neste sentido, as camadas de *Data Management e de Transformation* da Tabela 2-2 não estão incluídos no âmbito deste trabalho, uma vez que elas estão posicionadas dentro do limite do parceiro e/ou organização, repassando assim a responsabilidade pela gestão da qualidade, disponibilidade, temporalidade, desempenho e interpretação dos dados e informações para cada organização.

4.1.2 Componentes

Na análise da Figura 4-2 ainda é possível identificar que cada ambiente pode ter um ou mais componentes, que serão apontados a seguir.

O **Ambiente de Usuário** tem apenas um componente que é a **Plataforma de Interface Analítica**, componente recorrente nas plataformas de BI em geral. É a **Plataforma de Interface Analítica** que permita a seleção “do que se deseja” e de “como se deseja” o resultado na forma de uma expressão multidimensional (MDX), que é repassada ao “Servidor OLAP”.

No **Ambiente de BI Colaborativo**, existem dois componentes: o **Repositório Ontológico Centralizado** e o **Servidor OLAP**.

É o **Repositório Ontológico Centralizado** que permite a guarda e manutenção dos modelos hierárquicos representativos das dimensões de cada organização e do modelo global da ontologia superior. Este elemento prevê mecanismos que, em conjunto com o "Ferramenta de Desenvolvimento de Ontologias", permitem o alinhamento semiautomático das árvores de conceitos que lhe são expostas sob a forma de metadados, de suas instâncias (valorações), e de outros atributos que são necessários para recuperação dos dados nas suas origens e criação de informações consistentes.

Como o processo de alinhamento e fusão de conceitos infere a análise de similaridades, é necessária a definições de padrões de construção e apresentação dos modelos pelas organizações, e a implantação de mecanismos de troca de mensagens entre o repositório centralizado e as organizações, de forma a facilitar a realização de um “acordo” de aceitação entre as partes do grau de similaridade alcançado entre os conceitos e instâncias apresentadas junto aos já existentes.

Para isso, cada participante da solução proposta deve inicialmente analisar o impacto de seus conceitos relevantes e suas relações com o modelo ontológico superior existente no repositório centralizado. Essa análise facilita o processo de fusão visando a melhor adição, remoção ou atualização dos seus conceitos junto ao modelo superior. Como resultado dessa análise, conceitos e termos estarão em sincronia com as proposições globais e do interessado, bem como no que se refere aos dados, que serão providos por cada organização em um modelo de “dados como serviço”.

Assim, o **Repositório Ontológico Centralizado** permite o registro e captura posterior das ontologias dos participantes e da ontologia global, em um processo de publicação e de recuperação (*upload* e *download*).

O Ambiente de BI Colaborativo contém ainda o **Servidor OLAP**, que permite a recuperação dos modelos multidimensionais equivalentes as definições contidas no modelo global de ontologia superior presente no componentes “Repositório Ontológico Centralizado“. Além disto, esse Servidor acessa diretamente - ou indica a um esquema de cache a necessária rota de acesso - os serviços de dados remotos como fontes de dados para formação dos resultados finais conforme solicitado.

No **Ambiente Organizacional**, apresentam-se outros dois componentes: a **Ferramenta de Desenvolvimento de Ontologias** e a **Plataforma de Serviços de Dados**.

A **Ferramenta de Desenvolvimento de Ontologias** permite a criação e/ou atualização do modelo de informações pelos parceiros. Esta ferramenta é baseada em técnicas e padrões que permitem o alinhamento de ontologias sucessivas, de forma a estabelecer em uma nova ontologia de nível superior.

Novamente é disposta a necessidade de se impor padrões na construção e apresentação de modelos ontológicos (conceituais) pela organização. Esse modelo padrão para

apresentação do negócio contém, além do modelo de metadados sob a forma de uma árvore taxonômica de conceitos, também outros metadados tais como as métricas que serão disponibilizados pela organização interessada e as rotas dos serviços de dados correspondentes.

Para garantir a aderência final dos modelos dos serviços de dados a serem publicados pela organização, serviços adjacentes fornecem as propostas de serviços de dados que devem ser disponibilizadas por cada organização, de forma a atender as solicitações de dados oriundas do "Servidor OLAP".

Como outro componente desse ambiente, a **Plataforma de Serviços de Dados** permite que o Ambiente de BI Colaborativo acesse os dados nos ambientes dos parceiros, em conformidade com os padrões ora existentes de serviços de dados SOA, tal como REST.

Essa plataforma opera segundo os padrões e tecnologias especificados para ambientes de serviços de dados estabelecidos junto às arquiteturas orientadas a serviços, e devem ser implantados em infraestrutura de armazenamento de dados eficientes, flexíveis e escaláveis. Alinhado a ótica de Serviços de Dados (Data Services) de SOA, esse elemento tem por princípio a disponibilização de dados em padrões interoperáveis em XML, e sua existência nas organizações participantes é fundamental aos requisitos dessa arquitetura.

A fim de apresentar o comportamento da arquitetura, a Figura 4-3 associa o posicionamento de seus componentes, detalhando as camadas tecnológicas e ciclos de atividades necessárias para sua operacionalização

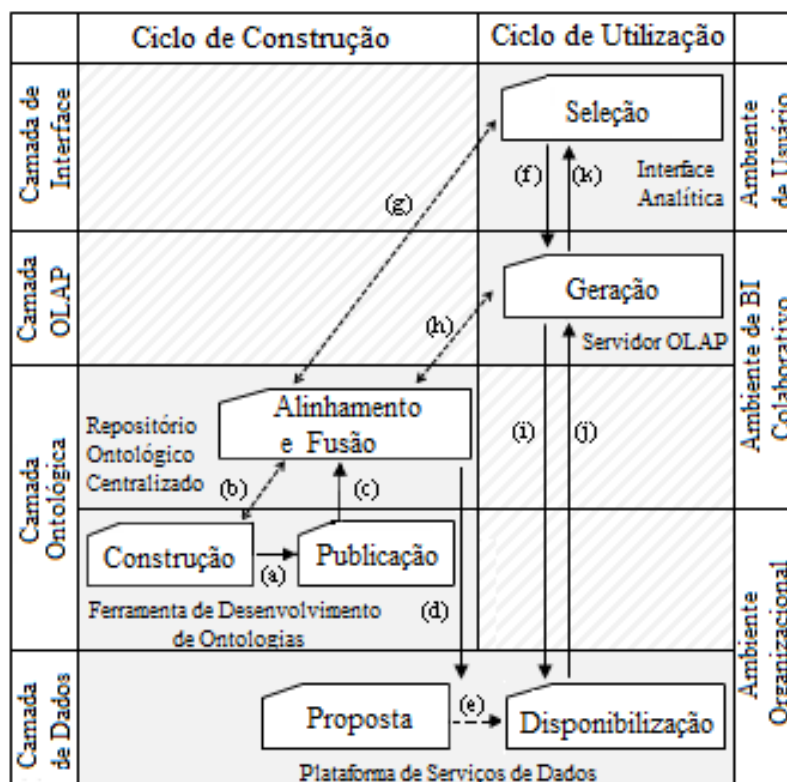


Figura 4-3 - Ambientes, Ciclos, Camadas e Processos da Arquitetura Proposta.

4.1.3 Camadas

Da mesma forma que um ambiente de BI convencional, este trabalho também divide a arquitetura proposta em camadas, porém, diferenciadamente do BI convencional, são consideradas quatro camadas, que são nomeadas como: **Camada de Interface**, **Camada OLAP**, **Camada Ontológica** e **Camada de Dados**.

A **Camada de Interface** utiliza-se da ontologia superior gravada no repositório ontológico centralizado para assegurar a formação de um modelo multidimensional consistente. O repositório ontológico centralizado armazena as relações entre as classes, instâncias e métricas, bem como entre métricas e suas fontes. A grande preocupação encontrada nessa camada é manter o controle da manipulação do usuário dada as restrições impostas por essas relações.

A **Camada OLAP** recebe a necessidade de informações oriundas da Camada de Interface e constrói a resposta necessária em cubos multidimensionais. Não há grandes preocupações sobre a integridade entre as hierarquias e as métricas selecionadas porque essa é assegurada anteriormente pela camada de interface e pelos recursos ora existentes nos Servidores OLAP existentes, componente da mesma.

A **Camada Ontológica** se materializa tanto dentro dos limites das organizações no papel de cliente bem como dentro do ambiente de BI colaborativo atuando como um servidor de guarda da ontologia de cada participante e da ontologia superior. Desta forma, a arquitetura proposta oferece elementos que podem ser combinados em diferentes abordagens para a integração de ontologias heterogêneas.

A **Camada de Dados** é usada para publicar e fornecer dados como serviço. O objetivo dessa camada consiste em dissociar os conceitos do parceiro das suas fontes de dados, através da aplicações de dois padrões de projeto de SOA: o baixo acoplamento e a abstração de serviços

Revisitando o modelo de ambiente convencional Chaudhuri and Dayal (2011) ilustrado na Figura 2-1, observa-se que as camadas apontadas para arquitetura proposta se interrelacionam com um menor acoplamento pela formação do Infospace e Dataspace. A Figura 4-4 ilustra os relacionamentos dessas e as tecnologias envolvidas, já discutidas nos Capítulos de Conceituação e Estado da Arte:

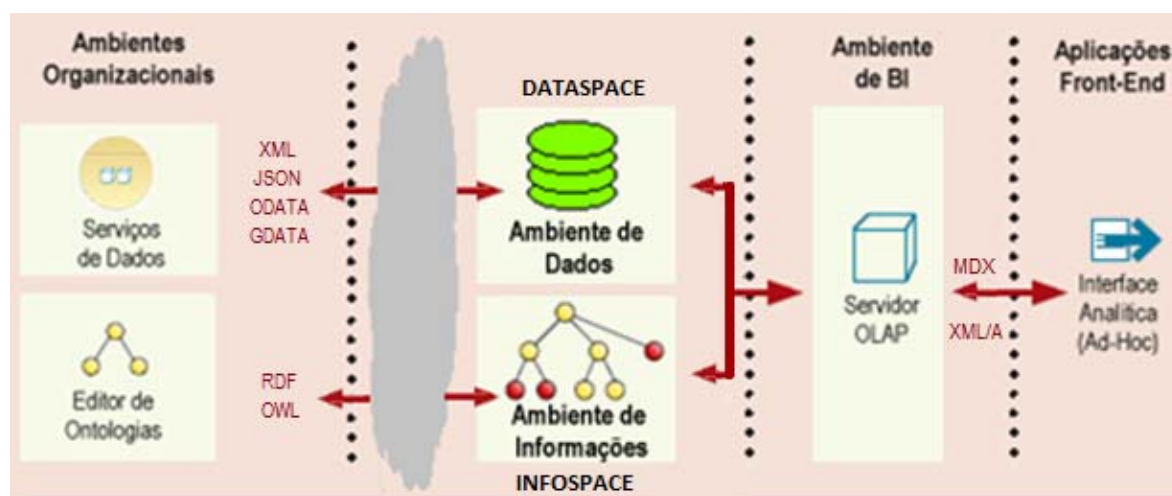


Figura 4-4 – Modelo em Camadas e Funcionamento

4.1.4 Ciclos

Em relação ciclo de vida das aplicações baseadas na arquitetura proposta, o processo se dá de forma semelhante a geração e utilização de um ambiente convencional BI, e assim, é realizada nos mesmos dois ciclos, que são nominados neste trabalho como Ciclo de Construção e Ciclo de Utilização, estando representados nas segunda e terceira colunas da Figura 4-2.

O **Ciclo de Construção** envolve ferramentas para a construção e integração de modelos conceituais dentro do repositório ontológico centralizado e a disponibilização pelo parceiro dos serviços de dados necessários usando sua plataforma de serviços de dados.

No **Ciclo de Utilização** se inicia pela ação de um usuário de BI junto a interface analítica, que interage a seguir com o servidor OLAP. Nessa interação, o servidor OLAP requisita os necessários dados como serviço dos parceiros e, em seguida, gera e fornece a resposta multidimensional de volta para a interface analítica.

Com base nos dois ciclos, de Construção e de Utilização, é apresentado a seguir uma descrição dos processos e como se dá a interação entre os componentes envolvidos, bem como é identificado os perfis básicos de usuários envolvidos na solução proposta.

Perfis de Usuários

- Administrador:

Operam junto ao Ambiente de BI Colaborativo realizando fundamentalmente o passo final de fusão de uma ontologia proposta por um parceiro. Neste sentido, o processo de fusão de ontologias é somente realizado pelo mesmo, o que garante uma visão final e madura do modelo superior a seu utilizado pela solução.

Ao final do processo de fusão, esse Administrador disponibiliza a Organização Participante o modelo do serviços de dados necessário (Proposta de Serviços de Dados) para que a mesma possa implementá-lo.

Atividades suplementares de construção de indicadores e de atualização de alguns domínios gerais (tais como o de tempo e de espaço) também são consideradas como atividades do Administrador.

Outra atividade adicional desse perfil pode envolver a avaliação contínua dos tempos médios de respostas dos serviços de dados pelas organizações participantes, mesmo que a latência do uso de padrões interoperáveis em XML sejam reduzidas pela presença de um mecanismo de cache.

- Organizações Participantes:

Atuam durante o Ciclo de Construção na formação do modelo ontológico de conceitos usando a Ferramenta de Desenvolvimento de Ontologias. Ao final da etapa de formação da sua ontologia de domínio, acessa o Repositório Ontológico Centralizado e submete sua ontologia ao processo de fusão pelo Administrador. Durante o Ciclo de Construção, esse perfil deve buscar auxiliar as atividades do Administrador em atendimento ao princípio de obtenção do maior grau de similaridade possível dos conceitos de sua ontologia junto à ontologia superior.

- Usuários:

Operam o ambiente utilizando plataformas de Interface Analíticas na Internet, realizando suas consultas genéricas ao modelo global de informações. Em geral, elementos das Organizações Participantes serão considerados como Usuários em potencial da solução.

4.2 PROCESSOS E COMPONENTES ENVOLVIDOS

4.2.1 Processos e Componentes e Etapas envolvidas no Ciclo de Construção

Usando uma **Ferramenta de Desenvolvimento de Ontologias**, um novo parceiro realiza a construção de seu modelo conceitual de informações, resultando em uma ontologia de domínio **(a)**. Este processo deve ser precedido de uma captura inicial da ontologia superior existente no **Repositório Ontológico Centralizado** para uso como base para a construção de sua ontologia específica **(b)**.

Acessando o **Repositório Ontológico Centralizado**, o parceiro executa a publicação de sua ontologia, isto é, executa atividades de troca de informações (*handshaking*) com o Administrador até que o seu modelo seja aceito e integrado **(c)**. Depois disso, uma aplicação subjacente de uso exclusivo do Administrador gera a proposta de serviços de dados que deve ser disponibilizada pelo parceiro **(d)**. Assim, cada organização usa essa proposta como modelo para construção e disponibilização de seus dados para serem acessados pelo Ambiente de BI Colaborativo **(e)**.

No final dessa tarefa de publicação, uma aplicação adicional subjacente pode ser usada para avaliar requisitos mínimos de performance dos serviço de dados publicados a fim de assegurar o desempenho global do modelo.

Assim, no Ciclo de Construção temos as atividades de:

- Captura da ontologia central:

Etapa na qual uma Organização Participante recupera o conteúdo do modelo ontológico global ou de outros modelos de domínio disponíveis do Repositório Ontológico Centralizado para dar suporte a construção de sua ontologia. Após essa recuperação, a Organização pode abandonar o Repositório Ontológico Centralizado e trabalhar localmente com a Ferramenta de Desenvolvimento de Ontologias.

- Elaboração do modelo ontológico local (de domínio):

Utilizando-se da Ferramenta de Desenvolvimento de Ontologias, a Organização Participante realiza processos de edição em geral até a formação de sua ontologia, em geral, uma ontologia de domínio. Neste sentido, a responsabilidade de manutenção do seu modelo específico (de domínio) é da mesma, sendo desnecessária a intervenção de outras Organizações Participantes no caso de mudança nas classes (conceitos) ou nas instâncias (valores dos conceitos) de seu modelo ontológico.

- Análise do alinhamento com o modelo global:

Utilizando-se da Ferramenta de Desenvolvimento de Ontologias e do modelo da ontologia superior capturada do Repositório Ontológico Centralizado, a Organização Participante busca alinhar seu modelo local com a cópia da ontologia global capturada. Ao final da análise, a Organização deve publicar essa proposta de modelo ontológico junto ao Repositório Centralizado para conhecimento do Administrador. Mecanismos ou mensagens devem dar suporte ao Administrador para conhecimento de quais novas ontologias foram publicadas pelas Organizações, sejam elas novas propostas ou alteração sobre propostas já existentes.

- Fusão das ontologias (pelo Administrador):

Operada pelo Administrador, este processo envolve a execução dos processos sistêmicos específicos de análise da ontologia proposta com a versão existente de ontologia superior até que o Administrador entenda que a fusão de conceitos não irá afetar nenhum outro partícipe no que se refere ao seu entendimento e dos conceitos globais e suas valorações.

Um processo final e determinístico processo gera e/ou atualiza arquivos de configuração que irão permitir o ambiente proposto sua operação, em destaque: a nova versão de ontologia superior; o novo modelo taxonômico de conceitos; a atualização das dimensões necessárias; os valores das chaves substitutas para as dimensões; e a proposta de serviços de dados .

- Criação e publicação dos Serviços de Dados:

Pelo recebimento e / ou acesso pela Organização Participante da Proposta de Serviço de Dados resultante do processo de fusão, essa elabora modelos de consulta de dados SQL (*queries*) sob a forma de serviços de dados que tenham a mesma representação de saída da proposta, não só com o mesmo modelo de dados como com o nome de campos repassados nos serviços de dados e seus valores de chave substitutas.

Com os elementos necessários, consultas SQL devidamente adaptadas, rotas de dados e de serviços declarados, a Organização que submete a solicitação de fusão publica o serviço de dados correspondente.

Uma atividade adicional pode ser realizada pela Organização Participante e pelo Administrador para avaliar o limite do desempenho na consulta ao menor grão de informação oriundo da organização de forma a garantir o desempenho geral do modelo.

Assim, cabe tanto ao Administrador como a Organização Participante validar os resultados obtidos bem como analisar o impacto da inclusão do novo modelo segundo fatores de performance, isto é: do tempo de resposta global, do tempo de resposta específico do serviço de dados, e do impacto na infraestrutura do

Ambiente de BI Colaborativo no que se refere ao uso de processadores, de memória e de tráfego de rede.

- Outras etapas intempestivas:

Em atenção à proposta de Berthold (2010), passos adicionais e intempestivos podem ser realizados pelo Administrador visando a criação de novas métricas de nível superior (relações entre métricas), formando indicadores para uso geral, através do uso de recursos de "campos calculados" junto ao servidor OLAP.

4.2.2 Processos, Componentes e Etapas envolvidos no Ciclo de Utilização

O Usuário, utilizando a **Plataforma de Interface Analítica**, executa o processo de seleção de informação, isto é, o usuário seleciona e /ou filtra conceitos e métricas, bem como ajusta o modelo de apresentação em colunas e linhas segundo sua necessidade (**f**). A utilização desta interface é controlada pela representação taxonômica da ontologia superior, formada a partir do **Repositório Ontológico Centralizado** (**g**).

Essas seleções do usuário são enviadas ao **Servidor OLAP** como requisitos de dados. O servidor OLAP elabora as consultas SQL aos dados correspondentes a estrutura multidimensional a partir da representação taxonômica da ontologia superior e monta o modelo multidimensional necessário como resposta (**h**).

Em seguida, o **Servidor OLAP** direciona as solicitações de dados necessárias para a **Plataforma de Serviços de Dados** nos parceiros envolvidos para a formação do cubo multidimensional ou ao repositório de cache se estabelecido (**i**). Quando todos os dados são retornados, o servidor OLAP executa a geração de cubo multidimensional (**j**) e encaminha o resultado para a **Plataforma de Interface Analítica** (**k**).

Assim, no Ciclo de Utilização temos as etapas de:

- Seleção de dimensões e medidas junto a Interface Analítica:

Refere-se ao uso da Interface Analítica para seleção das dimensões e métricas no formato desejado, utilizando-se da interface para realizar operações multidimensionais sobre o cubo global. Ao final, a interface analítica gera uma expressão multidimensional (MDX) correspondente as necessidades de

informação e sua forma de representação, que é encaminhada ao Servidor OLAP para geração do cubo multidimensional desejado.

Na arquitetura proposta, integrados à Interface Analítica, recursos serão utilizados para garantir que a situação desejada não gere desambiguidade ou a fatoração de medidas segundo dimensões independentes ou não vinculadas (*unrelated dimensions*). Essa situação será detalhada junto a Prova de Conceito.

- Formação das consultas SQL (OLTP) e requisição de serviços de dados:

Ação interna do servidor OLAP que, utilizando o modelo taxonômico da ontologia superior como modelo multidimensional superior, define uma série de acessos aos bancos de dados através de chamadas SQL. Estes acessos representam as necessidades de dados que o servidor OLAP precisa para construir o cubo multidimensional correspondente ao MDX encaminhado.

Um processo adicional da arquitetura proposta é o tratamento de exceções de falhas no acesso aos serviço de dados em cada Ambiente Organizacional. Porém, se é utilizado um serviço de cache, esse tratamento pode mitigar essa necessidade.

- Construção do cubo de dados a partir dos dados recebidos e seu encaminhamento para interface dos resultados (cubo de dados):

Ação interna entre o Servidor OLAP e a Interface Analítica com a apresentação dos resultados. Com escolha de uma Interface Analítica convenientemente relacionada com a escolha do Servidor OLAP é possível o uso dessa combinação de elementos existentes sem nenhuma necessidade adicional para o desenvolvimento da arquitetura proposta.

4.3 DA PROVA DE CONCEITO

Para execução da prova de conceito, a utilização individual de cada um desses componentes da arquitetura é analisada de forma isolada, com o posterior detalhamento do ambiente integrado, de forma a cumprir o objetivo desse trabalho.

O resultado a ser obtido nessa integração é um ambiente colaborativo e comunitário de informações interoperáveis, isto é, um ambiente com baixo acoplamento tanto entre os participantes como entre as camadas da arquitetura.

Esse trabalho determina ainda como meta adicional para a Prova de Conceito a geração do menor impacto possível na disponibilização dos modelos conceituais e de dados pelos participantes, bem como a obtenção de um desempenho aceitável durante a realização de consultas *ad-hoc* que envolvam conceitos e métricas disponibilizadas por várias organizações fisicamente isoladas.

Para alcançar esses objetivos, alguns mecanismos e padrões associados com os componentes da arquitetura ainda precisam ser definidos, contextualizados e analisados. Porém, como esses detalhes envolvem particularidades dos componentes selecionados, essas serão realizadas junto a Prova de Conceito.

Ultimados esforços da implementação como proposto, o uso desta arquitetura antevê um ganho crescente com as consecutivas agregações de novos domínios de informações (participantes), que são incrementalmente alinhados ao modelo global. Assim, a arquitetura proposta dá suporte como solução ao desafio do BI 2.0 - de democratizar o BI - por permitir a disponibilização crescente e integrada de informações para uma comunidade ou uso público sem a necessidade de conhecimento dos modelos ou das fontes de origem de seus dados, utilizando como interface aos usuários uma única e simples solução de BI.

5 - PROVA DE CONCEITO E RESULTADOS OBTIDOS

5.1 SOLUÇÕES PROPOSTAS JUNTO AO ESTADO DA ARTE

Como é de interesse que o ambiente de integração tenha uma forte associação com a plataforma de repositório ontológico centralizado, a seleção desse elemento é considerada prioritária de forma a que os outros componentes se adaptem ao modelo da solução de BI elencada.

5.1.1 Plataforma de Integração: Solução de BI

Na seção 3.4.2. do Capítulo "Estado da Arte" foram apontadas três alternativas de plataformas de BI que são oferecidas no regime de software livre: Pentaho BI, Spago BI e Palo BI.

Um dos pontos relevante na escolha da plataforma foi a adoção em 2010 pelo Serpro, empresa principal de fornecimentos de serviços de TI para os órgão do Poder Executivo no Brasil, do Pentaho BI Suite como ferramenta essencial de *Business Intelligence* (Serpro, 2010).

O Pentaho BI Suite é uma ferramenta completa de BI, que atende a diversos princípios de padronização e interoperabilidade. Essa ferramenta utiliza de forma integrada o servidor OLAP Mondrian e a ferramenta de interface analítica JPivot.

O Servidor Mondrian favorece o uso de interfaces que solicitem seus serviços através de expressões MDX convencionais e que, a partir dessas expressões, esse servidor gera um conjunto de consultas relacionais no padrão SQL que atendem as suas necessidades na formação do resultado - cubo multidimensional - a ser entregue à Interface.

Neste sentido, a Interface Analítica associada dentro da suíte do Pentaho, o JPivot, é fortemente aderente a geração de expressões MDX em consonância com as operações multidimensionais padrões que essa realiza, o que diminui qualquer esforço de implementação.

Além disso, a Plataforma Pentaho BI Suíte é disponibilizada integralmente em regime de *open source*, o que permite uma adaptação sem questões legais ou administrativas.

Porém, mesmo que o ambiente de interface analítica tenha uma interface amigável na seleção de hierarquias e seus níveis bem como das métricas ou combinação das mesmas gerando uma consulta MDX para o Servidor Mondrian, será necessário incorporar um tratamento adicional as operações multidimensionais para garantir conformidade na associação das métricas de origens diferenciadas ao contexto da representação de toda sua chave.

É fundamental ao atingimento do objetivo proposto que se adapte operacionalmente ou estruturalmente a Interface Analítica de forma a se estabelecer a devida consistência entre os conceitos e / ou níveis utilizados pela mesma na realização das operações multidimensionais solicitada em relação as métricas selecionadas.

De forma a reformatar o modelo de funcionamento da Plataforma Pentaho BI Suite, no que se refere ao uso do Servidor OLAP e o JPivot, é apresentado na Figura 5-1 a arquitetura básica do Serviço de Analítico do Pentaho - o Servidor Mondrian:

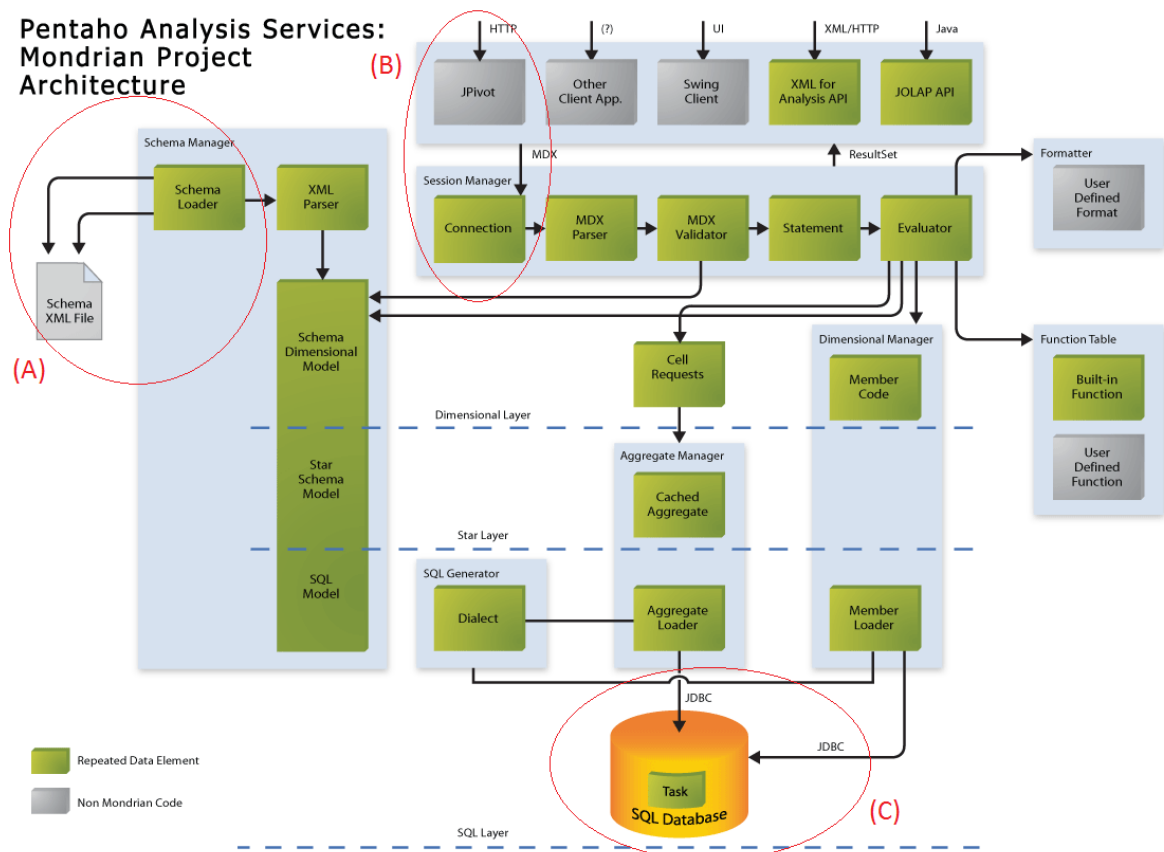


Figura 5-1 – Arquitetura do Servidor OLAP Mondrian
Fonte: Site Pentaho/Mondrian, 2011

Nessa Figura estão em destaque os elementos da arquitetura do Mondrian, e seus adjacentes que irão sofrer adaptações no funcionamento padrão com as implementações pretendidas:

(A) A leitura do modelo multidimensional no formato XML, especificamente um arquivo com a extensão “.mondrian.xml” (*Schema XML File*), realizada pelo servidor OLAP para carga do modelo multidimensional.

Da proposta de utilização do Mondrian como solução para Servidor OLAP, uma implementação recente agrega valor na solução a ser implementada: a possibilidade de uso de Cubos Virtuais.

Segundo o sitio Pentaho/Mondrian (2011), cubos virtuais “ocorrem frequentemente em aplicações do mundo real”, e é representado pela combinação das dimensões e medidas de um ou mais cubos.

Seu uso busca solucionar um dos problemas relacionado a modelagem multidimensional que envolve modelos multidimensionais que tem diversas coincidências dimensionais porém “tem tabelas de fatos de diferentes granularidades (como exemplo, uma medida no nível do dia em relação à outra definida no nível mês) ou tabelas de fatos de dimensionalidades diferentes (como exemplo uma contém as dimensões Tempo, Produto e Cliente, e a outra contém as dimensões Tempo, Produto e Locais) , e que pretende apresentar os resultados para um usuário final que não sabe ou não importa como os dados são estruturados” (Pentaho/Mondrian, 2011).

Um Cubo Virtual utiliza-se das definições de dimensões e medidas provenientes de outros cubos de domínio ("tabelas fato") que são declaradas em conjunto. Nesses cubos de domínio são encontradas as declaração <Dimension> para dimensões (ou <DimensionUsage> para dimensões compartilhadas entre esses) e <Measures> para medidas.

Em contraste a declaração de cubos de domínio, o Cubo Virtual aponta o reuso de dimensões e das medidas oriundas dos cubos usuais através das declarações de <VirtualCubeDimension> e <VirtualCubeMeasure> respectivamente, e com isso, importa dimensões e medidas das declarações dos cubos tradicionais. É óbvio que

essas importações necessitam especificar um atributo identificando o cubo de origem, chamado de <cubeName>.

Em adição, medidas importadas no Cubo Virtual permitem ainda gerar membros (novas medidas) calculados em tempo de execução, no modelo <CalculatedMember> do Mondrian OLAP.

Porém a implementação do modelo taxonômico por cubos virtuais exige a declaração dos cubos de domínio, o que implica que as ontologias de domínio declaradas por cada organização devem ser guardadas no Repositório Ontológico Centralizado mesmo que o interesse da arquitetura seja a ontologia superior. Assim, a guarda das ontologias de domínio e da ontologia superior no Repositório Ontológico Centralizado permitem a construção do arquivo "mondrian.xml" necessário. Os detalhes da implementação e do uso de Cubos Virtuais e de cada ontologia específica na arquitetura proposta será detalhado no tópico "Ferramentas Usadas e suas Implementações " deste Capítulo.

Porém, se a ontologia entregue por cada organização refere-se a uma ontologia de domínio (de um assunto), e que pode ser tratada e exposta por esse domínio como um cubo multidimensional segundo as suas arquiteturas de BI tradicionais executadas no local de cada Organização, este trabalho trata o uso do Cubo Virtual como um equivalente ao aspecto taxonômico de uma ontologia superior e irá ignorar inicialmente as necessárias características de ambiguidade do modelo.

O servidor Mondrian trata essa perspectiva de resolução de ambiguidades com o uso do atributo "*ignoreUnrelatedDimensions*" no seu Cubo Virtual. Esse atributo especifica que as medidas declaradas nos cubos de domínios não terão membros de dimensão afetando o seu uso no nível superior, suportando assim as agregações de medidas em diferentes níveis ou com dimensões diferentes (não relacionadas).

A documentação do MSDN (apud Pentaho/Mondrian, 2011) menciona que "Quando o valor de *IgnoreUnrelatedDimensions* é verdade, dimensões independentes são obrigadas a operar em seu nível mais alto. Quando o valor é falso, as dimensões não são forçados a seu nível mais alto".

O sitio da Pentaho/Mondrian (2011) aponta que essa propriedade está associada ao uso da função *ValidMeasure* junto ao MDX, e para fins de implementação será detalhada junto ao tópico "Ferramentas Usadas e suas Implementações " deste Capítulo.

Ainda segundo o sitio Pentaho/Mondrian (2011), outro recurso de interesse no caso de baixa entropia de dados nas dimensões (poucos valores tabelados) é o uso da degeneração das dimensões e tabelas *inline*, no qual os conceitos são declaradas com seus valores dentro do arquivo que define o esquema multidimensional a ser utilizado por esse servidor OLAP, o arquivo "mondrian.xml". Destaque-se que os valores dos conceitos serão declarados como instâncias de cada ontologia de domínio, e assim, quando do processo de fusão de ontologias, é possível se registrar os valores a serem utilizados nas dimensões no modelo taxonômico global, facilitando a operação do Servidor OLAP.

Em adição, de maneira a aumentar o desempenho do servidor OLAP, pode ser alterada as especificações básicas de memória cache disponível do servidor Mondrian utilizando-se de uma das opções de sua máquina virtual Java, o parâmetro "-Xms". Como exemplo, pode-se configurar o parâmetro como -Xms256m para garantir 256Mb de RAM para sua máquina virtual.

- (B) O acesso que a interface analítica do Pentaho BI, o JPivot, faz ao modelo do servidor OLAP Mondrian.

Todo o processo de tratamento do JPivot como ferramenta é definido junto ao modelo Mondrian que o mesmo recebe do servidor OLAP, na inicialização de uma instância sua de operação. Neste sentido, todo tratamento adicional necessário para a proposta de arquitetura passa pela definição de informações específicas a serem incluídas durante a etapa de fusão de ontologias, quando da formação do arquivo Mondrian correspondente à ontologia superior (modelo taxonômico superior).

- (C) O acesso as fontes de dados pelo servidor OLAP para geração dos cubos multidimensional em conformidade ao MDX apresentado pela interface do JPivot.

Como exposto no Capítulo "Estado da Arte" deste trabalho, na disponibilização de dados é importante o uso de protocolos HTTP/REST para solicitação, e que a mensagem de resposta dos mesmos sejam expressas em formato XML ou JSON. Esse

trabalho considerou que o formato ATOM (AtomPub), proposto pela Microsoft, impõem um grande volume de dados por registro, desnecessários para o modelo proposto e somente é interessante quando do seu uso para operações de *CRUD*, o que não é o interesse direto.

Um aspecto importante da implementação é que o servidor Mondrian, como utilizado na Suite do Pentaho BI, associa todas as referências no modelo taxonômico de estruturas multidimensionais à ser utilizado em uma instância de Interface Analítica associada através de uma única fonte de dados específica, e esse acesso se dá através de um único conector JDBC.

Conectores JDBC são modelos padrões de acesso a fontes de dados, com classes e interfaces escritas em Java, e são utilizados comumente por aplicações, sistemas, soluções em desenvolvimento para acesso a maioria dos bancos de dados relacionais. "Esses conectores atendem aos princípios de interoperabilidade pela padronização do acesso em um conjunto de chamadas que entregam cursores de dados compatíveis, indiferentes do servidor de banco de dados" (Oracle JDBC, 2012).

As configurações básicas do mecanismo de acesso JDBC pela suíte do Pentaho BI são realizadas através do módulo de Administração que acompanha essa (Pentaho/Mondrian, 2011), aonde um administrador da Suite configura suas fontes de dados.

Porém, como a arquitetura prevê um desacoplamento de dados do modelo de conceitos, e assim, do Ambiente de BI Colaborativo, existem várias fonte de dados para construção física dos resultados que representem o modelo superior proposto. Além disso, o acesso a diversas fontes de dados em cada operação multidimensional realizada junto a Interface Analítica acarreta em sobrecarga na transmissão e transformação do acesso XML, afetando drasticamente o desempenho da arquitetura proposta.

No tratamento desse aspectos, é proposto um mecanismo de cache, centralizando o acesso a diversas fontes de dados e aumento do desempenho geral. Neste sentido, de definir os recursos de memória cache para o ambiente de BI Colaborativo de forma a garantir um mínimo de latência com os dados oferecidos como serviços e assim buscar atingir tempos de resposta mais baixos, a arquitetura utiliza o recurso do banco de

dados em memória que é oferecido junto a solução Pentaho, o *HiperSQL Database - HSQLDB*.

O HSQLDB é um motor de banco de dados relacional SQL escrito em Java que suporta tanto o padrão de SQL ANSI-92 como o modelo do SQL:2008. Ele oferece um pequeno motor rápido e *multithread*, com tabelas em memória e ou em disco e com modelo de instanciação por chamada ou como servidor. O HSQLDB tem sido constantemente desenvolvido ao longo de 11 anos é completamente livre para usar e distribuir sob licença com base no padrão BSD sendo totalmente compatível com todas as principais licenças de código aberto (HSQLDB, 2012).

Porém é necessário que qualquer mecanismo de cache intempestivo, imprevisto e de multiacesso deva ser provido de um controle mínimo para evitar contínuos processos de carga desnecessários. Este trabalho propõe um simples mecanismo, publicado em conjunto aos serviços de dados oferecido pelas Organizações Participantes, para oferecer a data da última atualização realizada nas suas fontes de dados, de forma a confrontados com os valores registrados junto ao cache.

Isto permite um sincronismo entre o HSQLDB e as fontes de dados como serviço através do agendamento de tarefas que, mesmo com um baixo limite de tempo, possa anteceder ao pedido temporizado de dados, realize a verificação e comparação das datas entre cada ponto, e assim, garantir melhores opções de desempenho. Porém, para tal, fica reconhecida a necessidade da Organização Participante em manter este recurso atualizado - da data da última atualização de seus dados - como um serviço de dados adicional, e que é considerado necessário para a arquitetura proposta.

Neste sentido, sistemas de sincronismo por agendamento de sistema operacional garantem, por análise prévia, uma necessária atualização nos seus dados locais com baixa latência, pois este ciclo de avaliação da necessidade de sincronismo pode ser realizado dentro de curtos períodos de tempo, tomados os cuidados para que esse processo não caracterize um ataque ao serviço na origem, que então pode sofrer repúdio.

Entende-se então que a utilização de um mecanismo de cache como proposto favorece:

- A formação de um modelo virtual de acesso centralizado aos dados necessários;

- A redução das penalidades impostas pelo uso de XML, pois a carga de dados das origens está desacoplada do acesso a dados pelo serviço de BI; e
- A gestão de falhas, pois mesmo que o acesso as fontes de dados sofram interrupção, o cache garante o funcionamento do modelo

5.1.2 Espaço de Informações: Plataformas para Ontologia

Das ferramentas analisadas no estado da arte, dada as necessidades formalizadas e seu modelo de aplicação, destaca-se desses o Protégé, que oferece recursos avançados de configuração das abordagens de integração e de avaliação de similaridades, bem como, a capacidade de implementação de novas abordagens.

Adicionalmente, como o processo envolve o alinhamento de conceitos por diversas Organizações, faz-se necessária a definições de padrões de construção e entrega dos modelos conceituais como ontologia pelas Organizações para uso no ambiente selecionado. De maneira a não impor informações ou padrões desnecessários ao modelo, é utilizado nessa arquitetura de estruturas nas ontologias com apenas elementos de representação básicos, tal como classes e instâncias, e propriedades previstos na combinação de padrões RDF (<http://www.w3.org/2000/01/rdf-schema#>), RDFS (<http://www.w3.org/1999/02/22-rdf-syntax-ns>) e OWL (<http://www.w3.org/2002/07/owl>).

Assim, é utilizado a definição da classe superior "Thing", as classes "owl:Class", a propriedade "rdf:about" para identificação dos elementos, o *tag* "rdfs:subClassOf" para formar estruturar as propriedades taxonômicas e, com fundamental distinção, a relação "rdfs:isDefinedBy".

No Anexo I deste trabalho é apresentado o código RDF/OWL utilizado no modelo "Educacao.owl" utilizado nessa Prova de Conceito que exemplifica o formato de uma ontologia básica construída conforme a necessidade da arquitetura.

Esta representação textual da ontologia-modelo equivale a ilustrado na Figura 5-2:

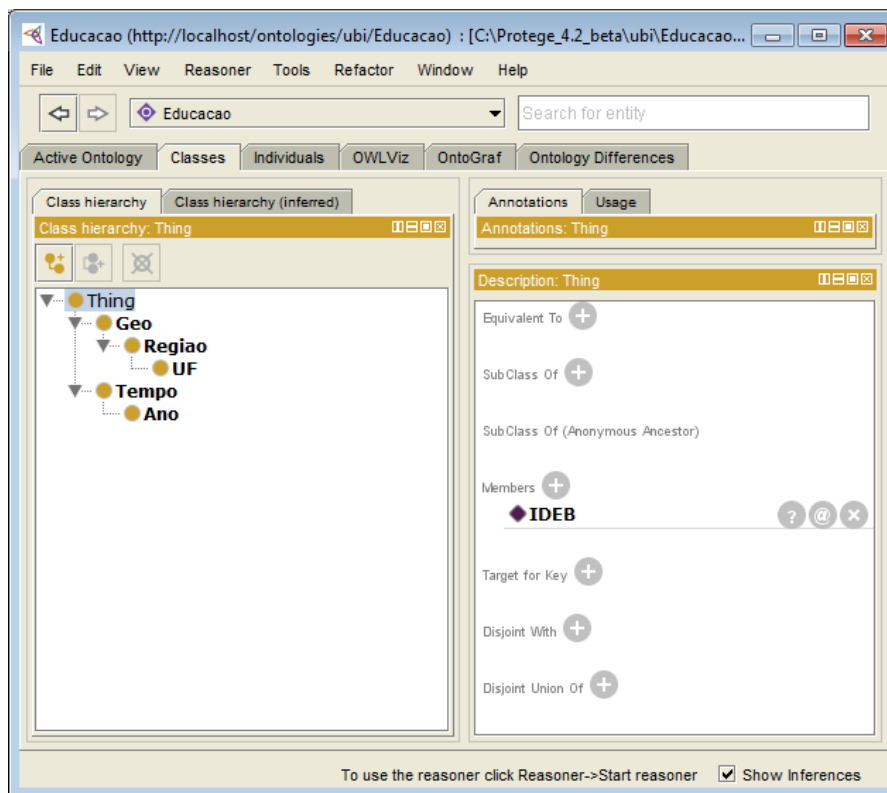


Figura 5-2 – Ontologia-Modelo "Educação" visualizada no Protégé

Para formação dos domínios a serem utilizados na construção da ontologia superior na Prova de Conceito utilizou-se em parte um modelo compatível, no seu primeiro nível, com a classificação funcional definida no Manual Técnico Orçamentário da SOF/MP do Brasil - MTO (2012). Em uma solução final de uso com foco no "interesse do serviço público" é sugerido neste trabalho esse modelo distribuição pois contribui para visão dos domínios na forma de assuntos governamentais e sociais. A elencado na Tabela 5-1 a lista das funções definidas da última versão desse Manual:

Tabela 5-1 - Classificação Funcional da Despesa.
Fonte: Site da MP/SOF - MTO, 2012.

Classificação Funcional do MTO
01 - Legislativa
02 - Judiciária
03 - Essencial à Justiça
04 - Administração
05 - Defesa Nacional
06 - Segurança Pública
07 - Relações Exteriores
08 - Assistência Social
09 - Previdência Social
10 - Saúde
11 - Trabalho
12 - Educação

13 - Cultura
14 - Direitos da Cidadania
15 - Urbanismo
16 - Habitação
17 - Saneamento
18 - Gestão Ambiental
19 - Ciência e Tecnologia
20 - Agricultura
21 - Organização Agrária
22 - Indústria
23 - Comércio e Serviços
24 - Comunicações
25 - Energia
26 - Transporte
27 - Desporto e Lazer
28 - Encargos Especiais

Do processo de elaboração de ontologias de domínio pela Organizações, duas dimensões básicas - tempo e geografia - que são fortemente usadas nos suporte aos sistemas de decisão, e assim, já estarão pré-declaradas em um "*template*" a ser disponibilizado as Organizações para suporte a construção segundo os padrões determinados.

O conceito "Tempo", definido como uma classe no *template*, deve suportar uma incorporação sucessiva de valores conforme a passagem dos anos e meses. O conceito "Geografia" (que foi simplificado para Geo, grifo do autor) não pode sofrer alterações sem a intervenção de um Administrador para validar face à um domínio comum que é regiões, unidades federativas e municípios do Brasil. A carga das instâncias dessas dimensões padrões devem ser sistematizadas, seja automaticamente (Tempo), ou por operações diretas do Administrador (Geo). Assim, essas dimensões já são implementadas *a priori* no domínio da ontologia superior e serão automaticamente carregadas ou geridas pelo Administrador através de recursos sistêmicos suplementares.

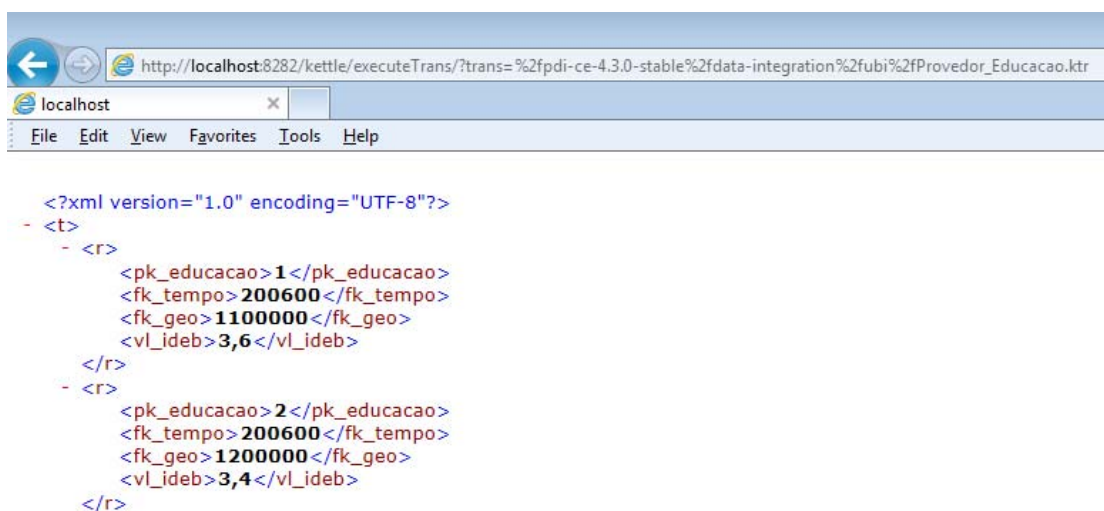
Em adição, de forma a simplificar para as Organizações, as mesmas serão informadas que as subclasses das classes de "Tempo" e "Geo" devem ser declaradas em suas ontologias de domínio, isto é, quais os níveis que serão utilizados em cada uma dessas classes genéricas, porém não devem conter elementos de instância ou, se esses forem declarados, que essas instâncias serão ignoradas na formação do modelo superior.

5.1.3 Espaço de Dados: Plataforma para Serviços de Dados

Como abordado no Capítulo "Estado da Arte" deste trabalho, provedores de serviços de dados em geral - tal como saídas de soluções de BI, ambientes de soluções SOA, servidores de banco de dados modernos, e implementações de serviços Web (Web Services) de entrega de dados baseadas em esquemas de dados no padrão XML - podem ser utilizados indistintamente para atender ao modelo de Plataforma de Serviços de Dados.

Como o provimento de dados para realização da Prova de Conceito, foi utilizado dois simples modelos, a disponibilização de serviços de dados oferecida pelo componente Pentaho Data Integration - PDI da suíte de BI da Pentaho e um modelo simplificado desenvolvido a partir da biblioteca PHP OData Producer Library, utilizando especificações do protocolo REST como mecanismo de requisição.

As Figuras 5-3 e 5-4 ilustram o acesso a serviços de dados pelos mecanismos do PDI e da biblioteca PHP OData Producer Library respectivamente:



```
<?xml version="1.0" encoding="UTF-8"?>
- <t>
  - <r>
    <pk_educacao>1</pk_educacao>
    <fk_tempo>200600</fk_tempo>
    <fk_geo>1100000</fk_geo>
    <vl_ideb>3,6</vl_ideb>
  </r>
  - <r>
    <pk_educacao>2</pk_educacao>
    <fk_tempo>200600</fk_tempo>
    <fk_geo>1200000</fk_geo>
    <vl_ideb>3,4</vl_ideb>
  </r>
```

Figura 5-3 – Fragmento de resposta XML acessando o Pentaho Data Integration.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="true"?>
- <t>
- <r>
  <pk_educacao>1</pk_educacao>
  <fk_tempo>200600</fk_tempo>
  <fk_geo>1100000</fk_geo>
  <vl_ideb>3.6</vl_ideb>
</r>
- <r>
  <pk_educacao>2</pk_educacao>
  <fk_tempo>200600</fk_tempo>
  <fk_geo>1200000</fk_geo>
  <vl_ideb>3.4</vl_ideb>
</r>
```

Figura 5-4 – Fragmento de resposta XML acessando o PHP OData Producer Library.

Esses modelos serão tratados em maiores detalhes no tópico "Modelo Funcional de Integração" deste Capítulo.

5.1.4 Sítio UBI: Repositório Ontológico Centralizado

Para tornar compatível aos objetivos propostos de simplicidade e operacionalização, o uso de repositórios de ontologias como analisado no "Estado da Arte" neste trabalho, obrigaria a disponibilização ou alteração dessas ferramentas além das necessidades previstas.

Assim, para suporte a necessidade de um modelo de gestão das atividades da arquitetura proposta, isto é, de mecanismos de troca de informações entre o Administrador e as Organizações Participantes, em adição as operações de repositório de ontologias presentes nas soluções analisadas, este trabalho propõe e implementa um simples sítio Web, que é denominado de "UBI" (*Upper BI*).

É junto a esse sítio que o Administrador atua e está fortemente vinculado a recursos do Ambiente de BI Colaborativo, além de atender aos princípios de Repositório Ontológico Centralizado e concentra a exposição de todas as necessidades da arquitetura em um só ponto de acesso, focado na disponibilização de mecanismos para comunicação entre as partes e mecanismos de troca de arquivos e de mensagens entre o Administrador e cada Organização Participante. Além disso, sendo um sítio web e não somente um serviço de repositório, permite a fácil implementação de políticas de acesso e segurança na forma como são tratados na disponibilização de outros sítios Web.

A Figura 5-5 ilustra a tela de operações básicas.

Interface da Organização Participante

Indice: */

Pastas OWL.válido | OWL.em.análise | OWL.rejeitado

Nome	Tamanho	Data
 Ontologias_de_Dominio		
 Ontologia_Global		
 Proposta_de_Chaves_Substitutas		
 Proposta_de_Servico_de_Dados		

Tamanho Limite: 128MB

Sitio UBI ... Fusão - Função do Administrador

Fusão:

Figura 5-5 – Representação do Sitio UBI.

Como opção, a Prova de Conceito agrupou as funções sistêmicas da Organização Participante e do Administrador de forma a facilitar seus aspectos de realização, e assim, seus módulos básicos são:

(A) *Apresentação*: apresentação básica do modelo e as tarefas definidas para cada participante bem como a sequencia de tarefas a serem realizadas na formação de uma nova versão da ontologia superior, aonde o processo de identificação do usuário (*login*) e senha podem estão incorporados.

(B) *Download* de Ontologias: opção a ser utilizada pela Organização Participante que prevê a captura no Repositório Ontológico Centralizado da versão corrente da "Ontologia Global", o arquivo UBI.owl, e cada "Ontologia de Domínio", representada por diversos arquivos com extensão ".owl". Essas são divididas em áreas separadas no site para não estabelecer confusão para as Organizações Participantes.

(C) *Upload* de Ontologias de Domínio: opção a ser utilizada pela Organização Participante, permite a disponibilização ao Administrador do Ambiente de BI Colaborativo de uma ontologia de domínio por pretendente ao uso colaborativo previsto na arquitetura.

(D) Fusão: módulo a ser executado pelo Administrador após o uso da ferramenta de análise do alinhamento de ontologias e que prevê, a geração do arquivo UBI.owl - que representa a ontologia superior gerada, bem como a construção dos outros arquivos de controle necessários para o ambiente.

(E) Propostas de Serviços de Dados: expõe como arquivos textuais planos o formato da saída XML que deverá ser ou que foi disponibilizado por cada Organização Participante.

(F) Scripts HSQLDB: expõe como arquivos textuais as chaves substitutas para cada dimensão do modelo e que devem ser usadas pela Organização Participante na construção de seus serviços de dados.

Na implementação da Prova de Conceito, esse sitio utiliza as configurações básicas de atributos de arquivos para correlacionar as respostas do Administrador para as Organizações onde os arquivos em verde estão disponíveis e aderentes, em amarelo estão sob análise do Administrador (isto é, foram solicitadas as participações mas o Administrador ainda não executou/aceitou o modelo publicado por esse participante), e os arquivos em vermelho identificam propostas não aceitas pelo Administrador. Esta representação básica encontra-se ilustrada na Figura 5-6:

Interface da Organização Participante

Índice: */Ontologias_de_Dominio/

Pastas OWL.válido | OWL.em.análise | OWL.rejeitado

Nome	Tamanho	Data
Demografia	4.1 KB	29/06/2012
Educacao	3.1 KB	29/06/2012
Imoveis_Publicos	5.3 KB	29/06/2012
Saude	5.7 KB	29/06/2012

Tamanho Limite: 128MB

Figura 5-6 –Sitio UBI - Ontologias de Domínio.

5.2 MODELO FUNCIONAL DE INTEGRAÇÃO

5.2.1 Arquitetura Operacional

Das plataformas escolhidas, após exposta as características principais a serem adaptadas e que são utilizados na Prova de Conceito, cabe então a apresentação do modelo operacional da arquitetura proposta, conforme ilustrado na Figura 5-7. Esse modelo tecnológico envolve os dois ciclos propostos porém detalha principalmente as operações do Ciclo de Construção, com destaque na geração automática de arquivos de controle compatíveis com os recursos seleccionados de forma a realizar as necessidades da arquitetura proposta.

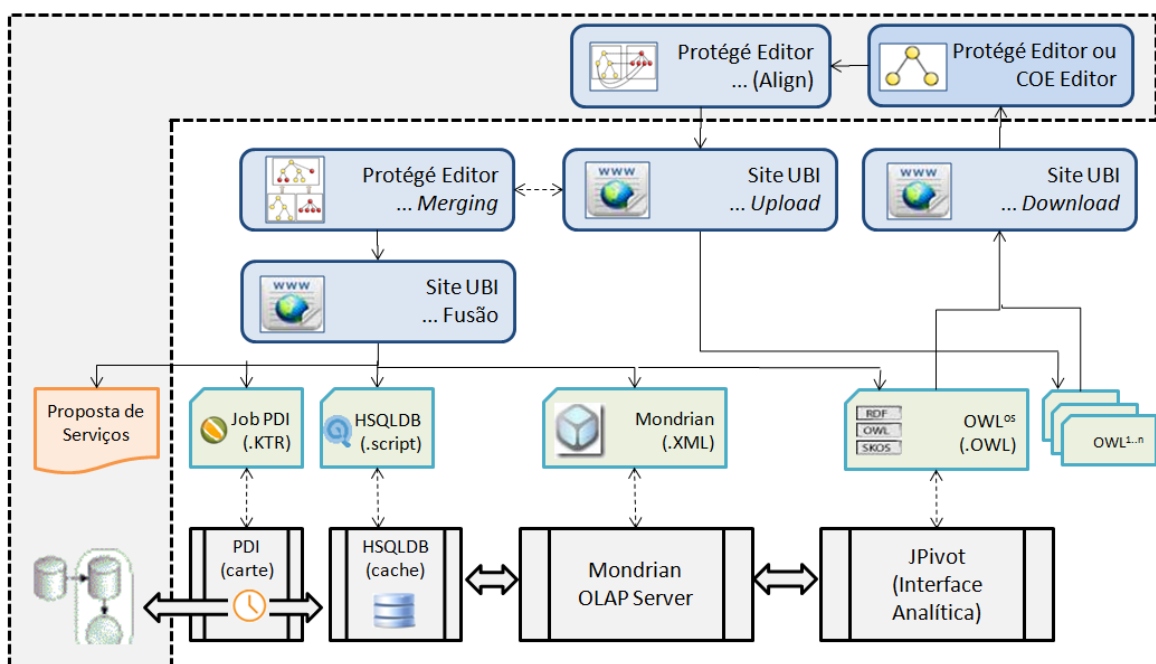


Figura 5-7 – Modelo Operacional da Arquitetura Proposta.

A área em destaque (cinza e pontilhada) representa o limite do Ambiente Organizacional na arquitetura proposta.

5.2.2 Ferramentas Utilizadas como Componentes e suas Implementações

(A) Ferramentas nos Ambientes Organizacionais

- Provedor de Dados XML:

Como exposto no Capítulo "Estado da Arte", diversas soluções já expõem seus resultados como Serviços de Dados em padrões, pelo menos básicos, de XML.

Como especificado, este trabalho não detalha os procedimentos para esses provimentos, somente impõem os requisitos necessários para acesso pelo Ambiente de BI Colaborativo. Desses requisitos, as Organizações Participantes tem a responsabilidade de que os arquivos XML gerados como saídas dos serviços de dados sejam similares em formato com os modelos expostos para mesma junto a "Proposta de Serviços" e utilizem valores de chaves substitutas conforme exposto nas "Propostas de Chaves Substitutas".

Na Prova de Conceito foi utilizado módulos do Pentaho Data Integration - PDI para geração de um ambiente provedor de serviços de dados em XML. Este processo foi realizado pela formação de transformações do PDI para disponibilização de tabelas como XML tendo como requisito de saída a opção de saída como *servlet*, conforme ilustrado na Figura 5-8:

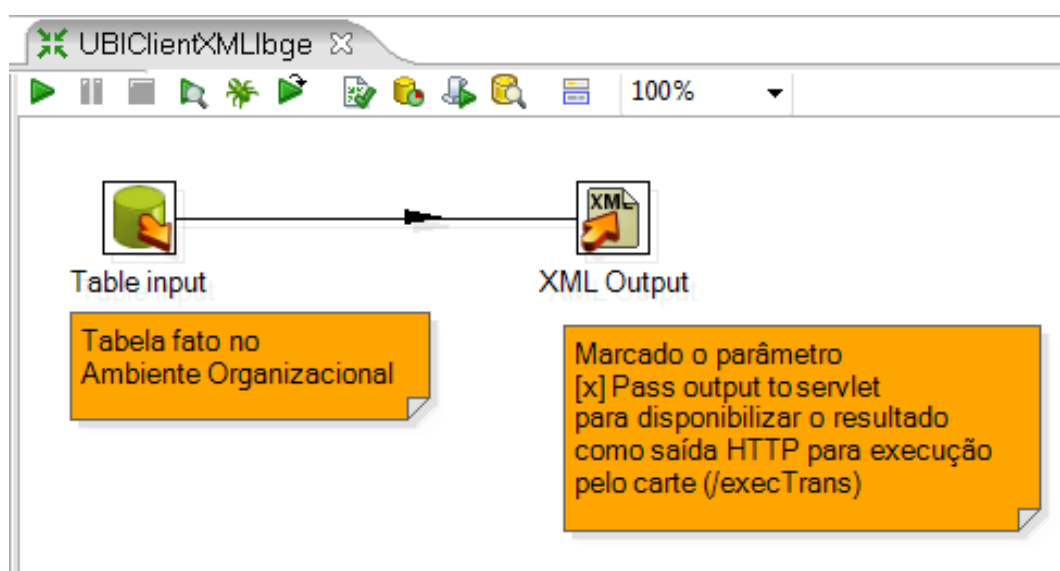


Figura 5-8 – Transformação PDI como Serviços de Dados

Utilizando o programa *Carte*, que é oferecido junto a solução PDI e que fornece um modelo de servidor de recursos HTTP que tem como saída os resultados dessas transformações, o resultado foi configurado para ser entregue conforme o necessário pela arquitetura.

A carga do serviço *Carte* é realizada pela chamada: `carte <servidor> <porta>`. No caso da Prova de Conceito, foi utilizado a chamada: `carte 127.0.0.2 8282`, onde 127.0.0.2 é o nosso "*datahost*", recurso computacional utilizado para simular o Ambiente Organizacional em servidores distintos.

Na execução da Prova de Conceito foi realizada uma carga completa de quatro fontes de dados com quantidades distintas de registros. A Tabela 5-2 apresenta o tamanho em registros tratados e o tempo total de realização:

Tabela 5-2 - Tempo de Resposta da Carga das Quatro Fontes de Dados Exemplo

Fontes de Dados	Número de Registros
Censo	27.825
Educação	131
Saúde	5.438
Imóveis Públicos	6.011
Total de Registros	39.045
Tempo Total	2 min e 10 seg

No "Anexo II - Provedores de Dados XML no Ambiente Organizacional" é apresentado um exemplo dos arquivos do PDI referentes transformações criadas e utilizadas. Neste exemplo destaca-se em sublinhado os parâmetros que sofrem alteração quando do uso nos outros serviços de dados utilizados.

Cabe-nos novamente destacar que o acesso do provimento de dados como XML pela arquitetura é baseada em uma especificação REST simplificada, exigindo apenas uma URI sem parâmetros que será definida junto a ontologia de domínio apresentada por cada Organização.

- Protégé OWL 4.0:

A interface de usuário fácil e intuitiva do Protege 4.0 é a justificativa principal para o uso dessa solução, porém editores mais simples como o CMapTools Ontology Editor - COE ou qualquer editor de texto pode ser usado para editar arquivos OWL.

Protege 4.0 fornece um ambiente integrado para desenvolvimento e manutenção de ontologias OWL, e tem uma arquitetura baseada em duas abstrações: visões (*views*) e guias (*tabs*), aonde cada visão é uma ferramenta especializada otimizada para executar uma tarefa particular enquanto cada guia apresenta um conjunto de ferramentas otimizadas para executar tarefas semelhantes ou relacionados.

Essa solução possui diversas visões, sendo destas são utilizadas na Prova de Conceito: Anotações (Annotation) para definição de relações tal como a fonte de dados de uma medida, as Classes (Class) para definição de hierarquias e níveis, e as

Instâncias (Individual) para definição de instâncias de níveis e para definição de medidas

Neste sentido, a Prova de Conceito usa como padrão de construção:

- Classes diretamente ligadas a "Thing" (classe genérica e global) correspondem no modelo multidimensional a definição de hierarquias. As chaves dessas hierarquias podem conter ou não o atributo "*isDefinedBy*" que declara o nome da chave de banco de dados equivalente. No caso de não serem declaradas, a chave assume o nome da classe acrescido do prefixo "pk_". Estas chaves são fundamentais na construção das relações entre os cubos e dimensões, e por conseguinte, afetam diretamente a formação dos cubos virtuais. Essas implementam ao primeiro nível de representação de informação do Infospace.
- Classes em outros níveis de hierarquia correspondem no modelo multidimensional a definição de níveis de determinada hierarquia. Elas podem conter ou não o atributo "*isDefinedBy*", que declara o nome da chave de banco de dados equivalente. No caso de não serem declaradas, a chave assume o nome da classe acrescido do prefixo "pk_". Essas implementam os diversos outros níveis de representação de informação do Infospace.
- Instâncias-membros referem-se a declaração dos elementos existentes dentro de cada dimensão do modelo multidimensional. A exceção das classes "Tempo" e "Geo" que são automaticamente tratadas, cada nível de cada hierarquia deve conter instâncias-membros que definem sua valoração. A dimensão "Tempo", oriunda da classe correspondente, deverá ser gerada conforme o decorrer do tempo real. A dimensão "Geo" será de gestão manual do Administrador. Essas implementam os as instância de valoração do Infospace.
- As medidas do modelo multidimensional são relacionadas diretamente como membros de "Thing" tendo como atributo essencial a relação "*isDefinedBy*" que define a rota de acesso de dados de cada métrica. Essa rota explícita exatamente o acesso ao serviço de dados conforme explorado no item "Espaço de Dados: Plataforma para Serviços de Dados" deste

Capítulo. Esta referência é que interliga o dado a sua origem e que permite que ontologia superior a ser gerada desacoplando esses dados da hierarquia de conceitos, formando o *DataSpace*.

Na Figura 5-9 é exemplificada a sequencia referente ao processo de construção de classes para atender as hierarquias e níveis do modelo multidimensional (aspecto taxonômico da ontologia), conforme o padrão da arquitetura:

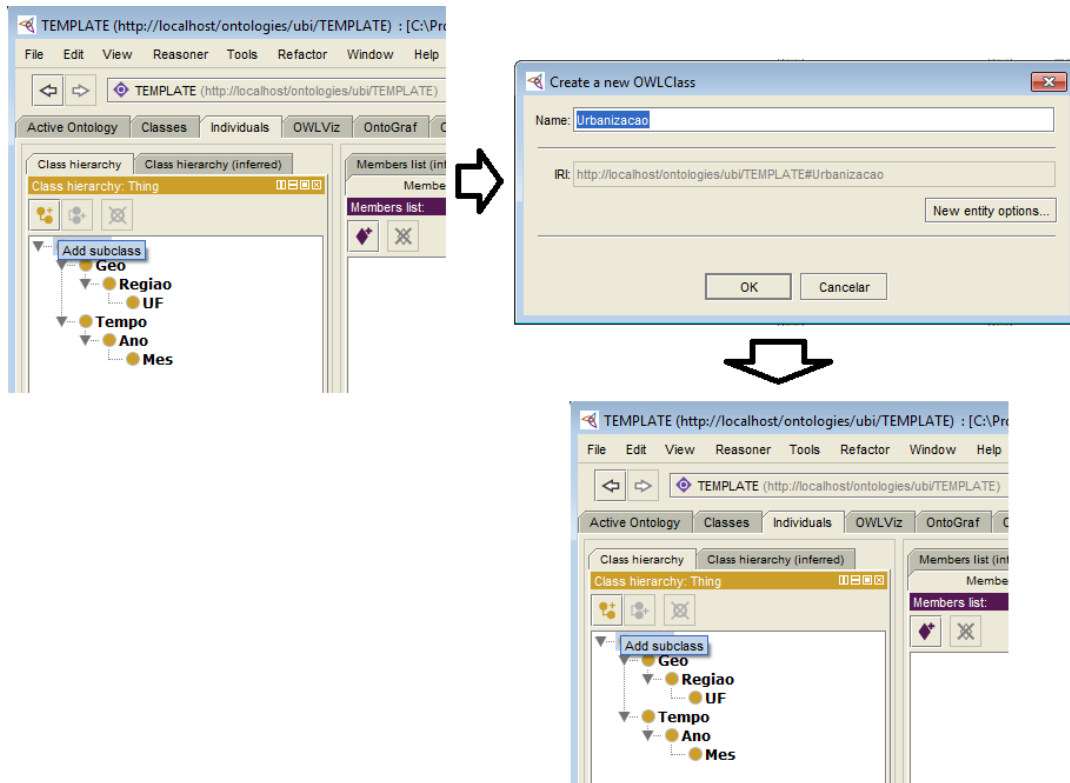


Figura 5-9 – Construção de Classes - Hierarquias e Níveis

Na Figura 5-10 é exemplificada a sequencia referente ao processo de construção de instâncias de classes, referentes a definição dos valores que podem assumir determinados níveis e que se referem aos valores dos nós folhas que classes que atendem aos níveis de uma hierarquia do modelo multidimensional, conforme o padrão da arquitetura:

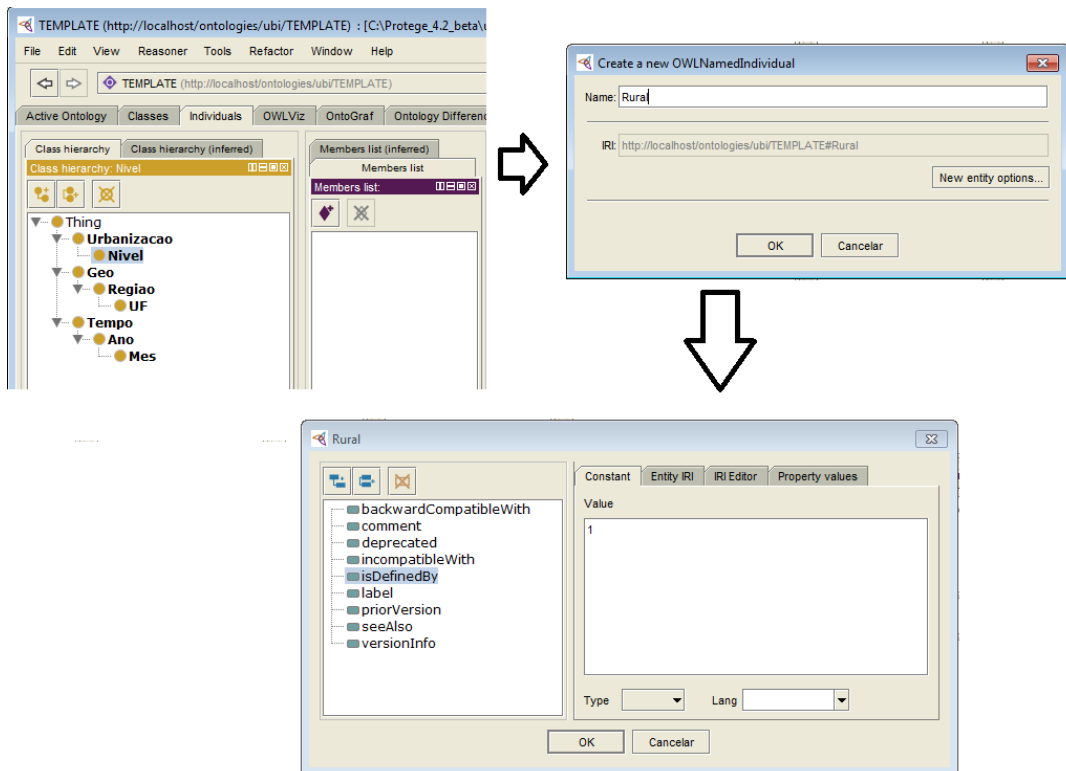


Figura 5-10 – Construção de Instâncias - Valores dos Níveis / Dimensão

Na Figura 5-11 é exemplificada a sequencia referente ao processo de construção de medidas e de seu relacionamento com a fonte de dados:

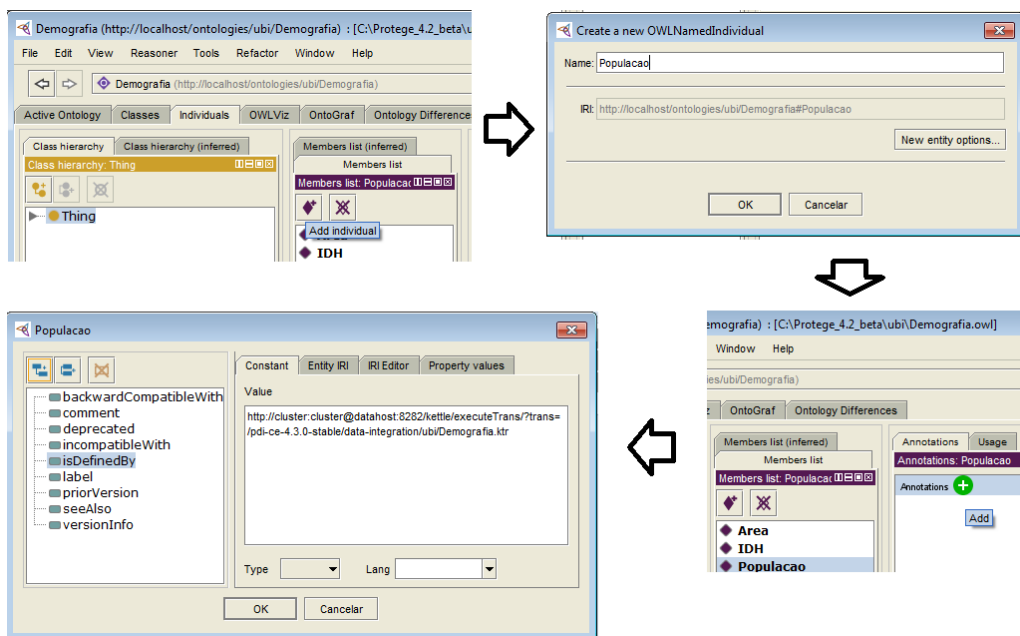


Figura 5-11 – Construção das Medidas e dos acessos aos Serviço de Dados

Como nesta proposta o uso do Protégé é utilizado ao mínimo necessário, esse funcionamento e modo de uso podem ser estudados através dos links disponibilizados no Sítio Wiki do Protégé (Protégé, 2005: Protege4GettingStarted e

Protege4Pizzas10Minutes). Cabe citar uma das opções pouco documentadas do uso do Protégé que é a alteração da sintaxe de classes, instâncias e propriedades: para isso utilize a opção de menu "*Refactor ... Change entity URI*" e altere o termo na caixa de diálogos apresentada.

Como o uso do Protégé como ferramenta de alinhamento é equivalente tanto para as Organizações como para o Administrador junto ao Ambiente de BI Colaborativo, as situações comuns de uso são tratadas junto ao item "(B) Ferramentas no Ambiente de BI Colaborativo" neste Capítulo.

- Sítio UBI (serviços abertos as Organizações Participantes ou Requerentes):

Como visto no item "Sítio UBI: Repositório Ontológico Centralizado" neste Capítulo, as ações do sítio desenvolvido envolvem o *download* da ontologia superior ou do *template* para construção de uma ontologia de domínio e o *upload* da ontologia final pela Organização, bem como permitem modelo para acompanhamento da aceitação da ontologia de domínio apresentada (pelo recurso de cores), da Proposta de Serviço de Dados a ser disponibilizada e das Chaves Substitutas a serem utilizadas.

A Figura 5-5 ilustra o acesso simplificado da disponibilização para download para arquivo e de um recurso adicional para que as Organizações Participantes possam realizar o *upload*.

Especificamente, o recurso de *upload* foi configurado para tolerar a transferência de arquivos unicamente na área de "Ontologias de Domínio" do site de forma a permitir apenas a informação ao Administrador de que uma solicitação para alinhamento e fusão foi apresentada.

(B) Ferramentas no Ambiente de BI Colaborativo

- Pentaho BI Suite

Como ferramenta de utilizada para disponibilizar o ambiente integrado foi utilizado o Pentaho BI Suite na sua versão 4.5.0 da Comunidade (biserver-ce-4.5.0-stable), recolhida na Internet junto ao sítio Pentaho. A Prova de Conceito utiliza o recurso da carga automática do banco de dados HSQLDB provido em seu pacote de

instalação (versão 1.8.0) e que a Pentaho utiliza para instalar seus fontes de dados para seus exemplos (i.e., o exemplo já instalado de *SampleData*).

Para a Prova de Conceito foi incorporado nessa instalação um novo banco de dados no HSQLDB, denominado "*ubi*", acrescido de um modelo de inicialização desse banco de dados junto ao script de carga do Pentaho - similar a carga do seus exemplos - para fornecer os mecanismos necessários para cache dos serviços de dados quando o ambiente é inicializado.

- Cache de Dados HSQLDB

É implementado através de uma nova fonte HSQLDB que é configurado junto a ferramenta de Pentaho através da sua definição nas suas fontes de dados. Esta configuração permite a definição no diretório "/data/hsqldb" do Pentaho dos dois arquivos necessários que tem o mesmo nome da fonte definida (no caso "*ubi*") e com extensões específicas: *ubi.properties* e *ubi.script*.

O arquivo *ubi.properties* contém as propriedades básicas utilizadas por pelo gerenciador para cada banco de dados, destacando-se a definição da propriedade "*hsqldb.default_table_type*" com o valor "*memory*", indicando que as tabelas serão tratadas em memória. A Figura 5-12 ilustra o conteúdo desse arquivo:

```
#HSQL Database Engine 1.8.0.7
#Fri Jan 27 16:06:29 BRT 2012
hsqldb.script_format=0
runtime.gc_interval=0
sql.enforce_strict_size=false
hsqldb.cache_size_scale=8
readonly=false
hsqldb.nio_data_file=true
hsqldb.cache_scale=14
version=1.8.0
hsqldb.default_table_type=memory
hsqldb.cache_file_scale=1
hsqldb.log_size=200
modified=no
hsqldb.cache_version=1.7.0
hsqldb.original_version=1.8.0
hsqldb.compatible_version=1.8.0
```

Figura 5-12 – Arquivo *ubi.properties*

O arquivo *ubi.script* contém a persistência das estruturas de dados das tabelas e propriedades do banco de dados "*ubi*". Ela implementa as declarações em linhas de comando no formato SQL ANSI, não só do modelo de banco de dados, mas também a situação sincronizada dos registros presentes nas tabelas em memória quando o banco é colocado em modo suspenso. A Figura 5-13 ilustra um extrato do conteúdo desse arquivo:

```

CREATE SCHEMA PUBLIC AUTHORIZATION DBA
CREATE MEMORY TABLE DIM_EQUIPAMENTO(PK_EQUIPAMENTO INTEGER NOT NULL PRIMARY KEY,DS_EQUIPAMENTO VARCHAR(255))
CREATE MEMORY TABLE DIM_GEO(PK_GEO INTEGER NOT NULL PRIMARY KEY,DS_REGIAO VARCHAR(25),DS_UF VARCHAR(25),DS_M
CREATE MEMORY TABLE DIM_TEMPO(PK_TEMPO INTEGER NOT NULL PRIMARY KEY,NU_ANO INTEGER,NU_MES INTEGER)
CREATE MEMORY TABLE DIM_URBANIZACAO(PK_URBANIZACAO INTEGER NOT NULL PRIMARY KEY,DS_URBANIZACAO VARCHAR(15))
CREATE MEMORY TABLE FAT_DEMOGRAFIA(PK_DEMOGRAFIA INTEGER NOT NULL PRIMARY KEY,FK_TEMPO INTEGER,FK_GEO INTEGE
CREATE MEMORY TABLE FAT_EDUCACAO(PK_EDUCACAO INTEGER NOT NULL PRIMARY KEY,FK_TEMPO INTEGER,FK_GEO INTEGER,VL
CREATE MEMORY TABLE FAT_IMOVEIS_PUBLICOS(PK_IMOVEIS_PUBLICOS INTEGER NOT NULL PRIMARY KEY,FK_TEMPO INTEGER,F
CREATE MEMORY TABLE FAT_SAUDE(PK_SAUDE INTEGER NOT NULL PRIMARY KEY,FK_TEMPO INTEGER,FK_GEO INTEGER,FK_EQUIP
CREATE USER SA PASSWORD ""
CREATE USER PENTAHO_ADMIN PASSWORD "PASSWORD"
CREATE USER PENTAHO_USER PASSWORD "PASSWORD"
GRANT DBA TO SA
GRANT DBA TO PENTAHO_ADMIN
GRANT ALL ON PUBLIC.DIM_EQUIPAMENTO TO PENTAHO_USER
GRANT ALL ON PUBLIC.DIM_GEO TO PENTAHO_USER
GRANT ALL ON PUBLIC.DIM_TEMPO TO PENTAHO_USER
GRANT ALL ON PUBLIC.DIM_URBANIZACAO TO PENTAHO_USER
GRANT ALL ON PUBLIC.FAT_DEMOGRAFIA TO PENTAHO_USER
GRANT ALL ON PUBLIC.FAT_EDUCACAO TO PENTAHO_USER
GRANT ALL ON PUBLIC.FAT_SAUDE TO PENTAHO_USER
GRANT ALL ON PUBLIC.FAT_IMOVEIS_PUBLICOS TO PENTAHO_USER
SET WRITE_DELAY 500 MILLIS
SET SCHEMA PUBLIC
INSERT INTO DIM_EQUIPAMENTO VALUES(11,'ULTRASSONOGRAFIA E RADIOLOGIA')
INSERT INTO DIM_EQUIPAMENTO VALUES(21,'DIALISE USO DOMICILIAR')
INSERT INTO DIM_EQUIPAMENTO VALUES(22,'DIALISE EM 3 TURNOS')
INSERT INTO DIM_EQUIPAMENTO VALUES(31,'MAMOGRAFO')
INSERT INTO DIM_EQUIPAMENTO VALUES(41,'TOMOGRAFO RAIOS X')
INSERT INTO DIM_EQUIPAMENTO VALUES(51,'OSTEODENSIMETRO')
INSERT INTO DIM_GEO VALUES(1100000,'N','RO',NULL)
INSERT INTO DIM_GEO VALUES(1100015,'N','RO','ALTA FLORESTA D.OESTE')
INSERT INTO DIM_GEO VALUES(1100023,'N','RO','ARIQUEMES')
INSERT INTO DIM_GEO VALUES(1100031,'N','RO','CABIXI')

```

Figura 5-13 – Arquivo ubi.script

O acesso ao banco de dados "ubi" se dará através de recursos JDBC diretamente na memória, porém o conteúdo desse arquivo garante que, na interrupção normal do serviço do Pentaho (*shutdown*), o modelo e conteúdo das tabelas de banco de dados existentes no momento da parada em memória sejam sincronizados com a versão em disco para, no caso de uma recarga do serviço do SGBD, possa ser restabelecida o estado original. Assim, o HSQLDB apresenta características interessantes quando se trata da implementação de arquiteturas de cache rápido para acesso a tabelas específicas.

A atualização dos campos de dados das tabelas em memória são realizados pelo processo de sincronismo previsto na arquitetura e as cargas das dimensões do modelo multidimensional se darão pela leitura dos membros-instâncias dos níveis-classes das ontologias de domínio apresentadas.

Neste sentido, os dados nas tabelas de fato de cada cubo de domínio são carregados por processos de acesso aos serviços de dados conforme será tratado no item seguinte.

- Pentaho Data Integration - PDI (ex-Kettle)

Como ferramenta de suporte a atualização de dados no cache utiliza-se o Pentaho Data Integration 4.3.0 da Comunidade (pdi-ce-4.3.0-stable) capturado da Internet

junto com a suíte de BI Pentaho. Esta ferramenta dá suporte a produção de consumidores o serviços de dados de forma agendada, porém a arquitetura cria esses processos da ferramenta de forma automática, não sendo utilizado nenhum de seus programas manual de criação de transformações e tarefas.

A Figura 5-14 ilustra e destaca os mecanismos envolvidos na chamada de dados como serviços que utiliza um modelo de carga gerado no formato de uma transformação do PDI:

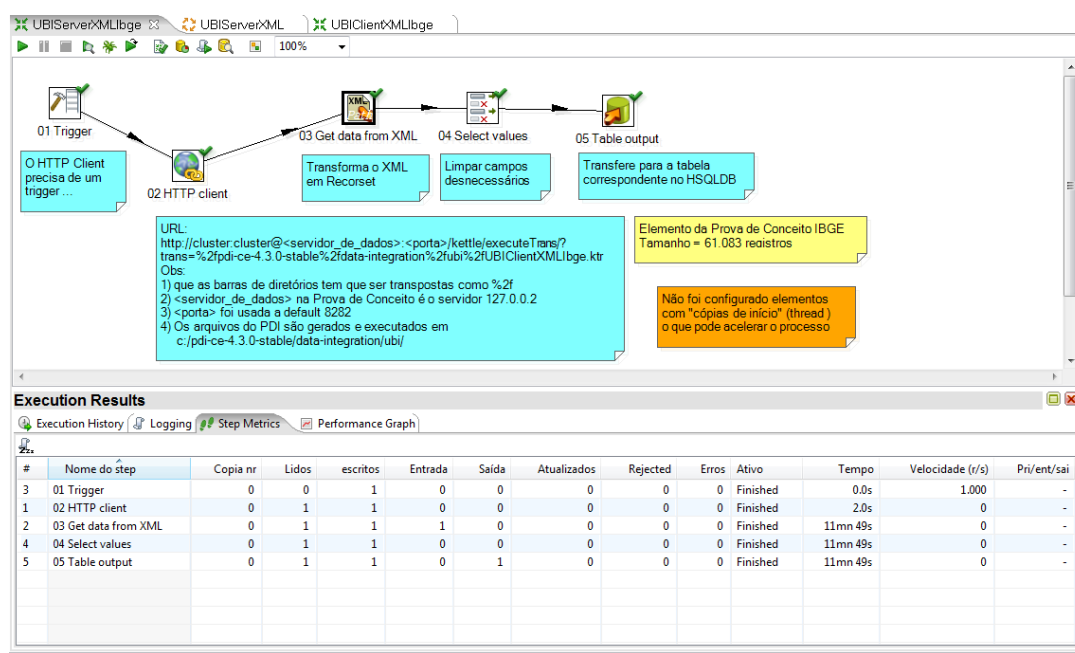


Figura 5-14 – Transformação PDI como Serviços de Dados

Para cada ontologia de domínio integrada, um arquivo similar ao apresentado acima será gerado e seu conjunto será agrupado em um processo de auto-execução devidamente agendado em serviços do sistema operacional. Uma característica em destaque é que o PDI permite a execução de várias transformações paralelamente. A chamada básica de cada domínio será criada como:

../pan.bat /file:"C:\pdi-ce-4.3.0-stable\data-integration\ubi\<dominio>.ktr

- Protégé OWL 4.0 (com o OWL-API incorporado):

O uso do Protégé OWL 4.0 pelas Organizações Participantes e pelo Administrador diferem-se no processo de edição, exclusivo dos participantes, porém são equivalentes no processo de alinhamento de ontologias na medida que atuam utilizando a mesma opção de menu "Show all loaded ontologies" e aciona o mesmo

processo de "*Merge Ontologies ...*" dessa solução sistêmica. Porém, fatores subjetivos e objetivos definem a diferença nesse processo.

Subjetivamente, uso dessa solução sistêmica pelo Administrador envolve uma maior responsabilidade no que se refere à uma análise criteriosa das classes e os impactos que a fusão poderá acarretar no modelo global.

Dos fatores objetivos, cabe salientar que apenas o Administrador pode executar o processo de fusão final, utilizando o seu acesso privilegiado à funcionalidade de "Fusão" disponibilizada no site, e assim, gerando todos os arquivos de controle que permitem o funcionamento integrado da arquitetura.

As Figuras 5-15 e 5-16 ilustram a visão do Protégé no que se refere aos seus itens de menu "Show all loaded ontologies" e "Merge Ontologies..." respectivamente:

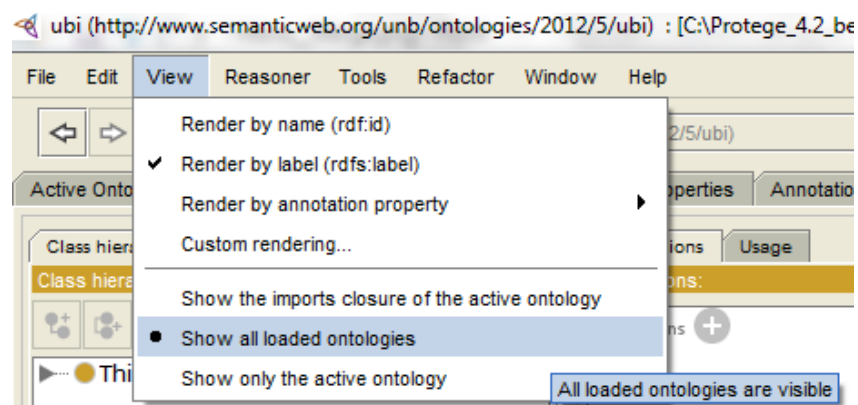


Figura 5-15 – Opção "*Show all loaded ontologies*" do Protégé OWL 4.0.

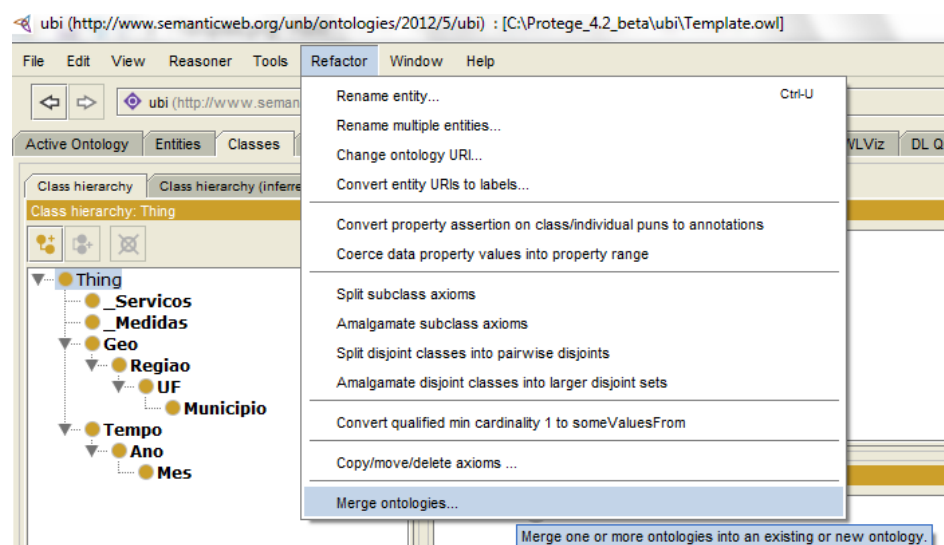


Figura 5-16 – Opção "*Merge Ontologies ...*" do Protégé OWL 4.0.

Durante o processo de alinhamento e fusão, a relação entre a ontologia superior e a ontologia de domínio pode apresentar três situações distintas que merecem destaque:

- Essas apresentam as mesmas hierarquias e as mesmas tem os mesmos níveis de granularidade porém apontam para medidas diferentes: ocorre a formação de cubos isolados que são integrados pelo Cubo Virtual superior. Esta situação é resolvida no múltiplo acesso a fontes de dados pelos cubos específicos de domínio.
- Essas apresentam certas hierarquia diferenciadas: ocorre a formação de dimensões não relacionadas, que serão impostas no Cubo Virtual superior. Esta situação define as restrições do uso de operações multidimensionais pelo usuário final.
- Essas apresentam as mesmas hierarquia porém em níveis diferenciados: ocorre a formação de granularidades não relacionadas. Esta situação obriga a construção de hierarquias diferenciadas em níveis no domínio dos dados.

O tratamento dessas características se dá especificamente da funcionalidade de "Fusão" apresentada no item seguinte.

- **Sítio UBI (Módulos do Administrador):**

Para favorecer a troca de informações entre o Administrador no Ambiente de BI Colaborativo e os Ambientes Organizacionais, foi construído um sítio Web. Em destaque, nesse sítio está prevista a funcionalidade de "Fusão" como uma operação específica a ser realizada pelo Administrador. Para seu uso, o Administrador deve apontar qual o modelo que foi encaminhado por um participante e que o Administrador irá utilizar para formação da nova ontologia superior.

Neste sentido, é reforçado o compromisso de atendimento da sequência de processos estabelecidos na Figura 5-7, aonde o passo "Sítio UBI ... Fusão" ocorre posteriormente ao processo de "Protégé Editor ... Merging", sendo necessário que os dois processos sejam realizados em sequência pelo Administrador.

Assim, nesse passo seguinte final, é através do acesso ao sítio UBI pelo Administrador que é realizado o processo final de alinhamento e fusão, sendo o mais complexo a ser detalhado.

Através da leitura do arquivo de ontologia superior no formato da Linguagem de Ontologia para Web (OWL) gerado na tarefa anterior (Protégé Editor ... Merging), um módulo suplementar do Sítio UBI, denominado de "Fusão", executa uma série de tarefas sistêmicas conforme ilustrado na Figura 5-17 de fluxo de processo:

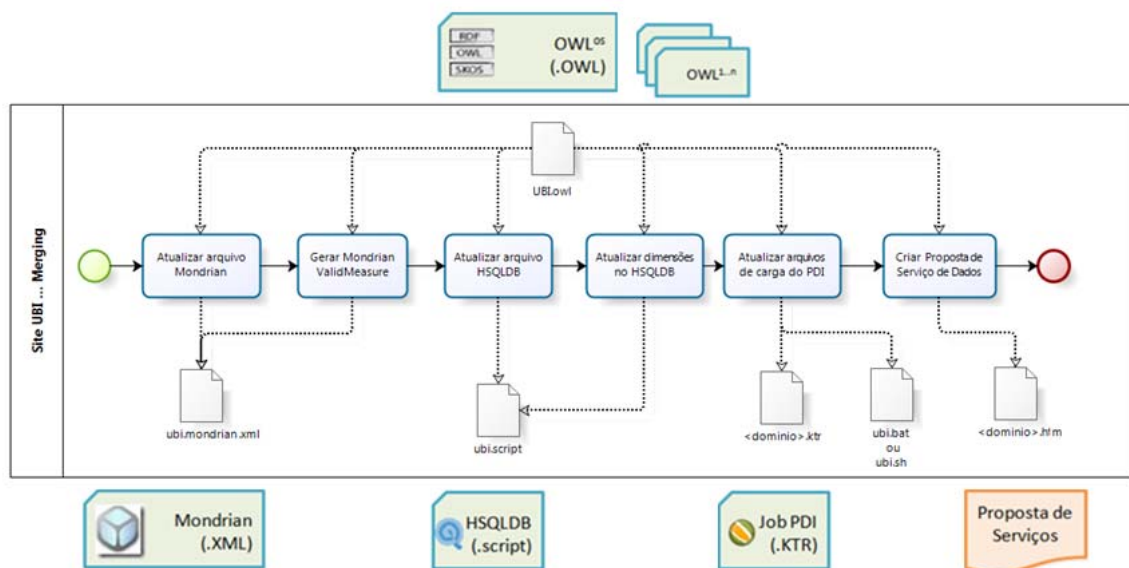


Figura 5-17 – Fluxo da Tarefa Sistêmica "Sítio UBI ... Fusão"

As lógicas envolvidas em cada passo desse processo são:

- o Atualizar Arquivo Mondrian:

Envolve a leitura da ontologia apresentada e a ontologia superior já alinhada para construção de uma árvore com as principais das necessidades do modelo, que estão implementadas dentro da ontologia apresentada. Essa árvore de detalhes é construída observando os aspectos taxonômicos que o modelo multidimensional necessita e agrega, no caso de novo participante, as hierarquias como nova dimensões compartilhadas do modelo Mondrian, e o acesso a dados do domínio como um novo cubo de domínio.

Três simplificações são usadas durante a geração do modelo: (i) o nome do arquivo de ontologia será usado para definir o nome do cubo de domínio bem como da tabela de fato correspondente para carga; (ii) o nome de cada

classe-hierarquia, classe-nível e medidas declaradas - seus termos no editor de ontologias - serão usado para definir o nome dos campos usados nas tabelas de fato e dimensão; e (iii) as relações "*isDefinedBy*" declaradas para as instâncias-membro definem as chaves de valores das dimensões a serem geradas e/ou atualizadas.

Ao final da formação da taxonomia dos cubos de domínio é anexado o Cubo Virtual, agregando as hierarquias e medidas (*Measures*, segundo a especificação Mondrian) e identificando as dimensões e granularidades não relacionadas.

Assim, em parte, as situações de relação entre a ontologia superior e a ontologia de domínio são resolvidas no modelo multidimensional superior.

- Gerar Mondrian ValidMeasure

Tão somente a intervenção do modelo Mondrian com os Cubos Virtuais e as dimensões não relacionadas podem não solucionar todas as situações de alinhamento dos cubos de domínio. Em certas situações, como de granularidade não relacionadas, soluções adicionais devem ser adicionadas ao modelo multidimensional final, no que se refere ao uso de recursos de *ValidMeasure*. O *ValidMeasure* é implementado dentro da configuração básica do MDX utilizado pela Interface Analítica. Este recurso impõe à essa Interface as limitações de uso no que se refere as combinações possíveis de medidas que podem ser utilizadas pelos usuários nas operações multidimensionais durante o Ciclo de Utilização.

- Atualizar Arquivo HSQLDB

Após a formação da especificação de cubos e dimensões do Mondrian, esse processo gera uma especificação de modelo de dados estruturada no formato de uma tabela de fato e das dimensões que são necessárias para o domínio em processamento. Essas tabelas são criadas junto ao cache HSQLDB, alterando o modelo de dados definido no arquivo *ubi.script* para posterior incorporação de registros de inclusão nas tabelas de fato, pelo processo

agendado de acesso aos serviços de dados, assim como dos dados das dimensões, que serão atualizados como detalhado no passo seguinte.

- Atualizar Dimensões no HSQLDB

Como as especificações dos valores de dimensões são declarados junto as ontologias de domínio como membros-instâncias de cada classe-nível, e todo esse processo se inicia pela leitura da ontologia apresentada e a ontologia superior, comandos de *"insert"* são declarados dentro do modelo HSQLDB em memória, sendo sincronizados com o modelo em disco do cache do banco de dados *"ubi"*.

- Atualizar (ou Criar) Arquivos de Carga do PDI

Assim como para a atualização dos arquivo HSQLDB, o processo em destaque gera o processo de carga, no caso da Prova de Conceito, um arquivo de transformação PDI. Esse processo tem um modelo padrão, segundo apresentado no "Anexo III - Consumidores de Dados XML", \cujo conteúdo é copiado e alterado nos pontos indicados no Anexo referenciado.

- Criar Proposta de Serviço de Dados

Ao final, conhecido os modelos de dados e as dimensões referentes especificamente da ontologia de domínio apresentada, é gerado um arquivo de mensagem, a "Proposta de Serviço de Dados", com um fragmento do XML que deverá ser gerado no serviço de dados, e a "Proposta de Chaves Substitutas", com os valores das dimensões alinhadas com o modelo global. Este trabalho opta pelo uso do padrão HTML para formação dessa "Proposta de Serviço de Dados" e para "Proposta de Chaves Substitutas", e seu formato é ilustrado nas Figura 5-18 e 5-19, respectivamente.

Proposta de Serviço de Dados para Demografia

```
<?xml version="1.0" encoding="UTF-8"?>
<id>
</id>
<pk_educacao>1</pk_educacao>
<fk_tempo>200600</fk_tempo>
<fk_geo>1100000</fk_geo>
<vl_ideb>3,6</vl_ideb>
</t>
</id>
```

Dimensão Tempo:

pk_tempo	ds_ano
200600	2006
200700	2007
200800	2008
200900	2009
201000	2010

Dimensão Geo:

pk_geo	ds_regiao	ds_uf	ds_municipio
1100015	N	RO	ALTA FLORESTA D.OESTE
1100023	N	RO	ARIQUEMES
1100031	N	RO	CABIXI
1100049	N	RO	CACOAL
1100056	N	RO	CEREJEIRAS

Figura 5-18 – Exemplo de uma Proposta de Serviço de Dados

```
\c true
DROP TABLE PUBLIC.DIM_EQUIPAMENTO;
\c false
CREATE MEMORY TABLE DIM_EQUIPAMENTO(PK_EQUIPAMENTO INTEGER NOT NULL PRIMARY KEY, DS_TIPO VARCHAR(255));
GRANT ALL ON PUBLIC.DIM_EQUIPAMENTO TO PENTAH0 USER;
INSERT INTO DIM_EQUIPAMENTO VALUES (22,'DIALISE_3_TURNOS');
INSERT INTO DIM_EQUIPAMENTO VALUES (21,'DIALISE_USO_DOMICILIAR');
INSERT INTO DIM_EQUIPAMENTO VALUES (31,'MAMOGRAFO');
INSERT INTO DIM_EQUIPAMENTO VALUES (51,'OSTEO-DENSIMETRO');
INSERT INTO DIM_EQUIPAMENTO VALUES (11,'ULTRASSONOGRRAFIA_E_RADIOLOGIA');
commit;
```

Figura 5-19 – Exemplo de uma Proposta de Chaves Substitutas

Assim, na realização de todos os passos do processo de "Fusão", toda configuração para construção ou atualização dos arquivos necessários para o Servidor OLAP, para Interface Analítica, e para formação do serviço de dados e seus caches são gerados automaticamente, a partir da leitura da ontologia superior e dos arquivos de ontologia de domínio. Cabe, ao final, esse processo alterar as características da ontologia entregue no próprio site sinalizando a Organização Participante que seu modelo foi aceito, isto é, a mudança de atributo do arquivo que impõem uma mudança de "cor" do item.

(C) Ferramentas no Ambiente de Usuários

- Navegador Web:

A restrição de uso de navegadores é explicitamente relacionada com a versão utilizado do Pentaho BI. Como especificação dos navegadores web que podem ser utilizados em conjunto com a solução implementada na Prova de Conceito, segundo o documento "*Getting Started*" da Pentaho, essa versão do Pentaho BI Suite é compatível com as seguintes versões dos navegadores mais conhecidos: Internet Explorer 7.x, 8.x, 9.x; Mozilla FireFox 1.5, 2.0; Opera 9.x; Netscape Navigator 7.x, 8.x e Google Chrome 10.x.

5.3 ETAPAS DA PROVA DE CONCEITO

Utilizando-se dos aspectos dos Ciclos de Construção e Utilização definidos na Figura 5-7 será tratada a implementação da Prova de Conceito reportando aos processos e tarefas envolvidas nesses ciclos.

Para execução da prova de Conceito foi necessária a simulação de quatro diferentes ontologias de organizações distintas, além de uma ontologia básica com os conceitos de "Tempo" e "Geo", sendo impostas nessas ontologias simuladas situações específicas como destacadas na "Motivação" desse trabalho.

Essas quatro referem-se a à exemplos dos domínios:

- Demografia: População, Área (em km²) e valor do IDH por Ano e Município (agregando por UF e região).
- Educação: valor do IDEB por Ano e UF (agregando por Região).
- Imóveis Públicos: Quantidade, Área e Valor de Imóveis disponibilizados por mês (agregando por Ano), por Município (agregando por UF e Região) e o seu nível de Urbanização (podendo ser urbano ou rural).
- Saúde: Quantidade de Equipamentos Recomendados e Atendidos por Ano e por Município (agregando por UF e Região) e pelo Tipo de Equipamento (cinco tipos diferentes instanciados dentro da ontologia específica).

São ilustradas nas Figuras 5-20, 5-21, 5-22, 5-23 e 5-24 esses conteúdos ontológicos à serem experimentados:

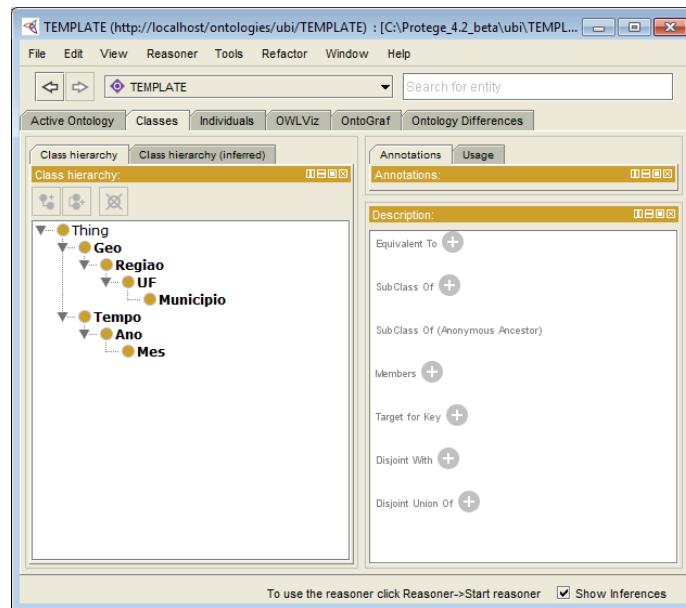


Figura 5-20 – Ontologia "Template.owl"

A ontologia "Template", ilustrada na Figura 5-20, apresenta o que é considerado como níveis máximos para as necessidades das dimensões "Tempo" e "Geo" (Tempo até Mês e Geo até Município).

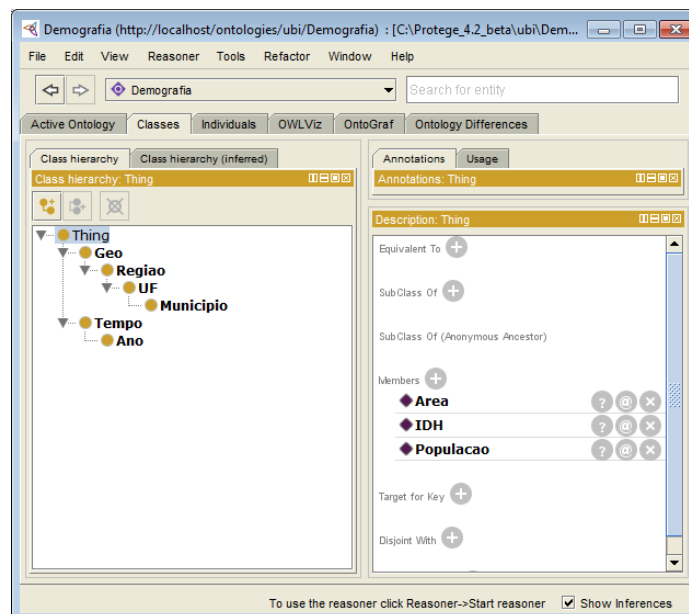


Figura 5-21 – Ontologia "Demografia.owl"

A ontologia "Demografia", ilustrada na Figura 5-21, utiliza também as dimensões "Tempo" e "Geo" porém aponta o problema de "granularidades não relacionadas" no que se refere a

dimensão "Tempo" - pois tem como nível maior o "Ano", e acrescenta as medidas "Área", "IDH" e "População" ao modelo global.

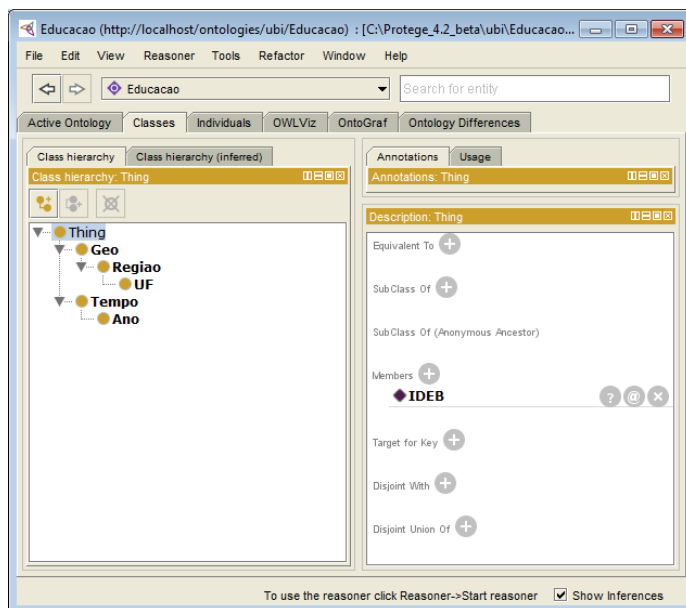


Figura 5-22 – Ontologia "Educação.owl"

A ontologia "Educação", ilustrada na Figura 5-22, utiliza também as dimensões "Tempo" e "Geo" porém aponta o problema de "granularidades não relacionadas" no que se refere a dimensão "Tempo" - pois tem como nível maior o "Ano" - e "Geo" - pois tem como nível maior a "UF", e acrescenta a medida IDEB ao modelo global.

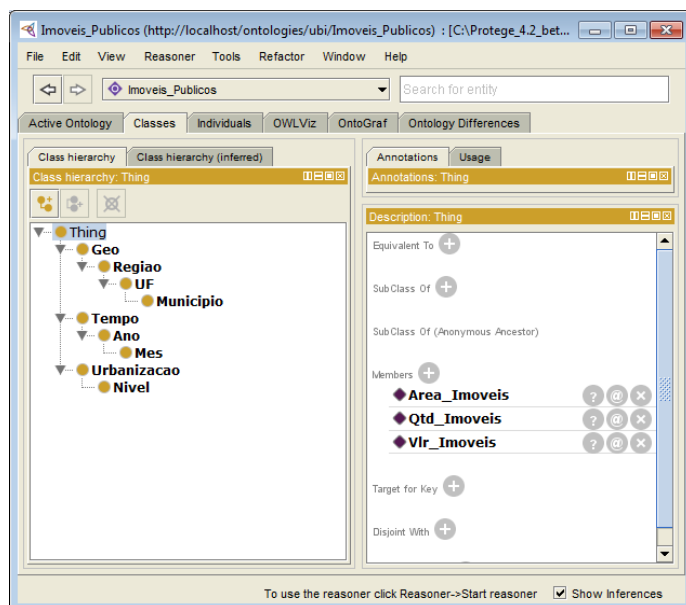


Figura 5-23 – Ontologia "Imoveis_Publicos.owl"

A ontologia "Imóveis Públicos", ilustrada na Figura 5-23, utiliza também as dimensões "Tempo" e "Geo", porém aponta o problema de "dimensão não relacionada" por agregar a

classe-hierarquia "Urbanização", e acrescenta as medidas "Área Imóveis", "Qtd Imóveis" e "Ver Imóveis" ao modelo global.

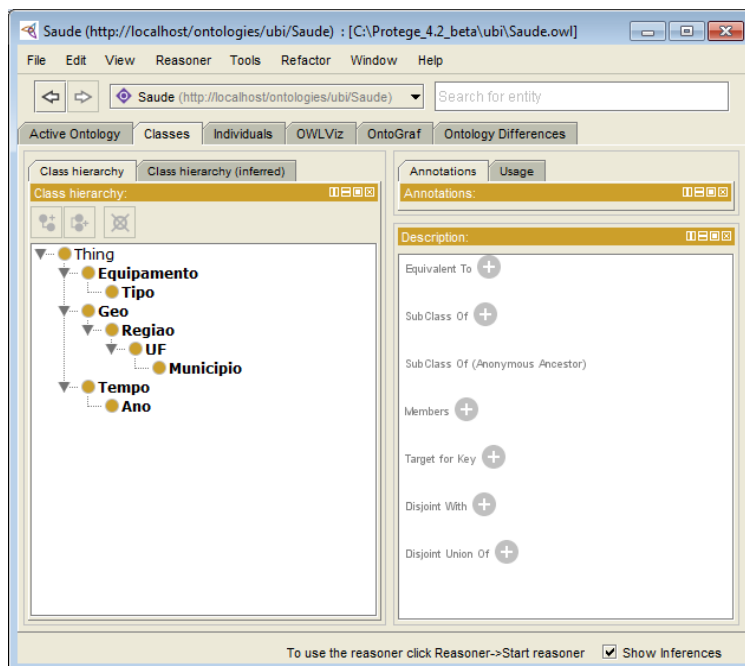


Figura 5-24 – Ontologia "Saude.owl"

A ontologia "Saude", ilustrada na Figura 5-24, utiliza também as dimensões "Tempo" e "Geo" porém aponta o problema de "granularidades não relacionadas" no que se refere a dimensão "Tempo", o problema de "dimensão não relacionada" por agregar a classe-hierarquia "Equipamento", e acrescenta as medidas "Qtd Atendida" e "Qtd Recomendada" ao modelo global.

5.3.1 Ciclo de Construção e suas Implementações

O usuário acessando o sítio na opção "Ontologia Global", ilustrada na Figura 5-4, realiza o download do "Template.owl" e do "UBI.owl" para suporte a sua construção.

As Figuras 5-25 e 5-26 ilustram a representação dessas ontologias junto a ferramenta Protégé, sendo que a Figura 5-26 ilustra o ponto final da ontologia superior após o processo de alinhamento e fusão das ontologias de domínio demonstradas nas Figuras 5-20, 5-21, 5-22, 5-23 e 5-24 já expostas.

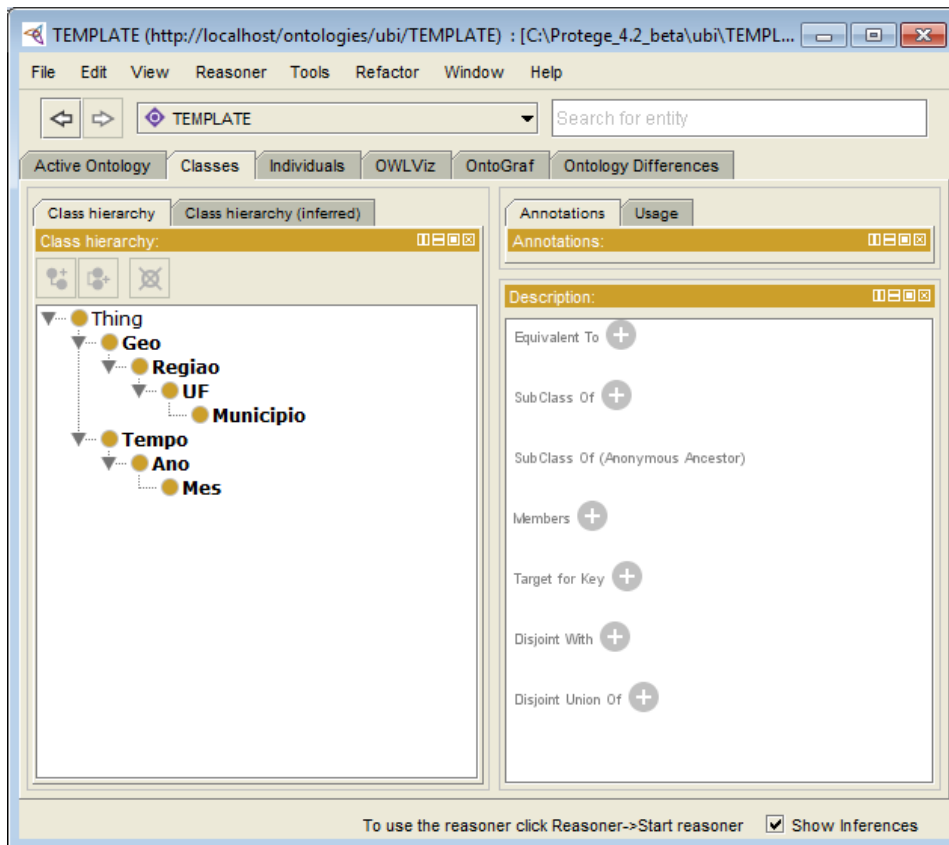


Figura 5-25 – Ontologia "Template.owl" somente com as classes Tempo e Geo

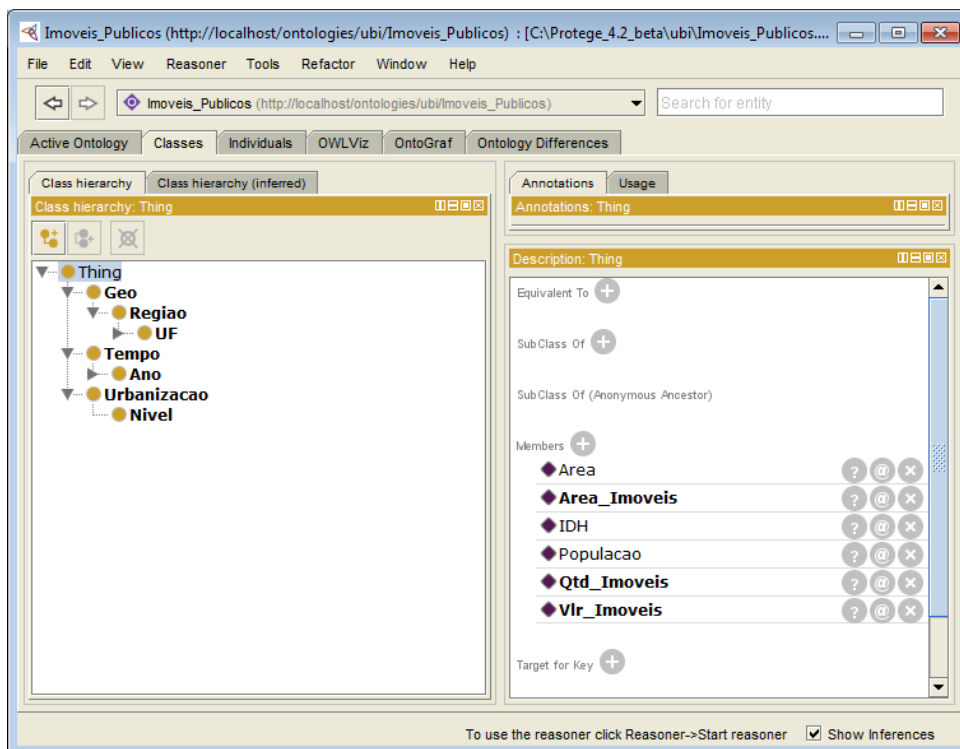
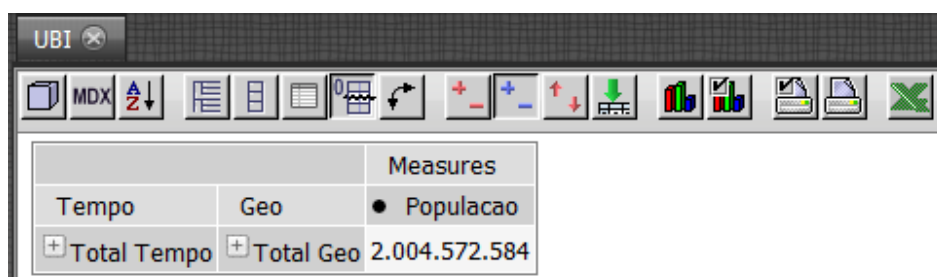


Figura 5-26 – Ontologia "UBI.owl" já certos domínios incorporados

5.3.2 Ciclo de Utilização e suas Implementações

O Ciclo de Utilização envolve o uso da plataforma Pentaho e seu componente JPivot para realizar análises de dados dentro do ambiente de BI. Ao solicitar a solução padrão que acessa o modelo UBI.mondrian.xml, o sistema apresenta a tela padrão do Pentaho, conforme ilustrado na Figura 5-27:

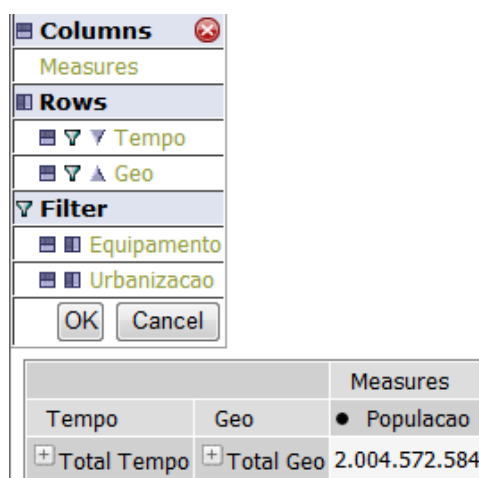


The screenshot shows the Pentaho JPivot interface. At the top, there is a toolbar with various icons for navigation and analysis. Below the toolbar is a pivot table with the following structure:

		Measures
Tempo	Geo	● Populacao
+ Total Tempo	+ Total Geo	2.004.572.584

Figura 5-27 – Acesso com a medida "População" como padrão.

Utilizando a opção de realização de operações multidimensionais, pode-se observar que o Cubo Virtual está operando com todas as dimensões que foram agregadas nos processos de alinhamento e fusão, isto é, Tempo, Geo, (Tipo de) Equipamento (de Saude) e (Nível de) Urbanização (de Imóveis Públicos), conforme ilustrado na Figura 5-28:



The screenshot shows the 'Columns' dialog box in JPivot. It has several sections:

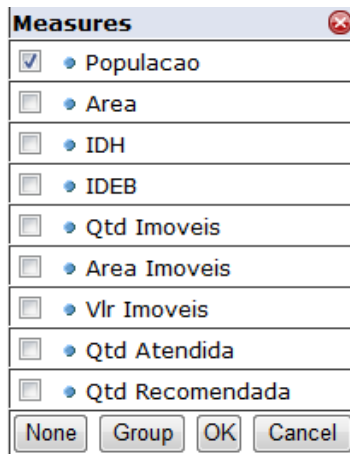
- Measures:** Populacao
- Rows:** Tempo, Geo
- Filter:** Equipamento, Urbanizacao

At the bottom, there are 'OK' and 'Cancel' buttons. Below the dialog box, a preview of the pivot table is shown:

		Measures
Tempo	Geo	● Populacao
+ Total Tempo	+ Total Geo	2.004.572.584

Figura 5-28 – Hierarquias / Dimensões presentes no modelo superior.

Além disso, na declaração de medidas, o *link* "Measures" da Figura 5-28, todas as medidas agregadas pelo processo de fusão se encontram presentes, conforme ilustrado na Figura 5-29:



		Measures
Tempo	Geo	● Populacao
+ Total Tempo	+ Total Geo	2.004.572.584

Figura 5-29 – Medidas no modelo UBI

Esta situação final comprova que o processo semiautomatizado de construção de soluções de BI em nível superior através dos processos especificados na arquitetura proposta suporta as necessidades da mesma, e que não impõem longas intermediações entre organizações interessadas.

Porém, o processo de integração de modelos com situações de "dimensões não relacionadas" e de "granularidades não relacionadas" precisa de uma comprovação junto a Prova de Conceito.

Neste sentido, o acionamento de determinadas ações na Interface Analítica realizam operações multidimensionais, desdobrando parcialmente o domínio "Tempo" e "Geo" com todas as medidas presentes. Mesmo no caso das medidas estarem sob domínios de "Tempo" e "Geo" diferentes as operações multidimensionais, a Interface Analítica permite a apresentação das medidas selecionadas, conforme ilustrado na Figura 5-30:

		Measures								
Tempo	Geo	● Populacao	● Area	● IDH	● IDEB	● Qtd Imoveis	● Area Imoveis	● Vlr Imoveis	● Qtd Atendida	● Qtd Recomendada
[-] Total Tempo	[+] Total Geo	2.004.572.584	93.461.093,00	0,00	3,30	861.410	8.730.235.133	18.949.235.842,24	867.934	640.697
[+] 2005	[+] Total Geo	184.119.666	8.496.463,00	0,00		84.656	317.453.309	1.972.878.575,36		
[+] 2006	[+] Total Geo	186.705.182	8.496.463,00	0,00	3,10	85.943	933.177.602	1.728.346.530,02		
[+] 2007	[+] Total Geo	186.606.849	8.496.463,00	0,00	3,60	78.012	729.732.925	1.795.345.620,86	96.421	96.494
	[+] CO	13.307.960	1.606.142,00	0,00	4,20	539	15.855.053	2.821.359,86	5.230	4.933
	[+] N	15.082.496	3.853.107,00	0,00	3,40	308	360.362	2.434.670,26	3.046	2.149
	[+] NE	49.379.621	1.551.616,00	0,00	3,00	37.961	495.210.518	603.263.893,43	15.008	40.578
	[+] S	31.134.993	561.902,00	0,00	4,30	12.507	15.444.949	373.673.624,61	25.772	9.989
	[+] SE	77.701.779	923.696,00	0,00	4,20	26.697	202.862.044	813.152.072,70	47.365	38.846

Figura 5-30 – Todas as medidas no domínio de "Tempo" e "Geo".

Em continuidade a Prova de Conceito, para validação, incorpora-se uma dimensão específica ("Equipamento" do domínio "Saude") junto com uma medida externa ("População"). Ainda assim, o modelo responde comprovando a possibilidade de uso de "dimensões não relacionadas", conforme ilustrado na Figura 5-31:

Tempo	Equipamento	Measures		
		● Qtd Atendida	● Qtd Recomendada	● Populacao
[-] Total Tempo	[+] Total Equipamento	867.934	640.697	2.004.572.584
[+] 2007	[-] Total Equipamento	96.421	96.494	186.606.849
	DIALISE EM 3 TURNOS	17.961	8.517	
	DIALISE USO DOMICILIAR	1.784	10.267	
	MAMOGRAFO	4.329	1.137	
	OSTEODENSIMETRO	2.489	12.572	
	TOMOGRAFO RAO X	3.391	1.469	
	ULTRASSONOGRAFIA E RADIOLOGIA	66.467	62.532	
[-] 2008	[-] Total Equipamento	241.583	148.844	189.535.546
	DIALISE EM 3 TURNOS	49.111	26.658	
	DIALISE USO DOMICILIAR	2.281	17.990	
	MAMOGRAFO	9.184	2.501	
	OSTEODENSIMETRO	4.585	25.629	
	TOMOGRAFO RAO X	5.680	2.738	
	ULTRASSONOGRAFIA E RADIOLOGIA	170.742	73.328	

Figura 5-31 – Dimensão e Medidas de "Saude" com uma medida de "Demografia".

A Figura 5-32 apresenta, uma última validação: a adição de medida no escopo de outro domínio, o de "Imóveis Públicos".

Tempo	Equipamento	Measures			
		● Qtd Atendida	● Qtd Recomendada	● Populacao	● Qtd Imoveis
[-] Total Tempo	[-] Total Equipamento	867.934	640.697	2.004.572.584	861.410
[+] 2007	[-] Total Equipamento	96.421	96.494	186.606.849	78.012
	DIALISE EM 3 TURNOS	17.961	8.517		
	DIALISE USO DOMICILIAR	1.784	10.267		
	MAMOGRAFO	4.329	1.137		
	OSTEODENSIMETRO	2.489	12.572		
	TOMOGRAFO RAO X	3.391	1.469		
	ULTRASSONOGRAFIA E RADIOLOGIA	66.467	62.532		

Figura 5-32 – Outra medida adicional oriunda do domínio "Imoveis Públicos".

Essa sequencia de ações comprova que medidas e hierarquias envolvendo soluções de domínios diversos são capazes de serem implementadas nas arquiteturas proposta.

5.4 AVALIAÇÕES PRELIMINARES

Quanto aos serviços de dados, a Tabela 5-2 aponta restrições na quantidade de dados oferecidos por cada usuário. Mesmo que os processos de carga ocorram de maneira simultânea, seja em um modelo utilizando diretamente aos serviços de dados ou através de mecanismos de cache, se as Organizações Participantes apresentarem modelos de informação complexos ou com um alto volume de registros, o uso desta arquitetura apresenta características restritivas quanto das questões de desempenho.

Para execução da Prova de Conceito, o modelo de ontologias de domínio foram simplificados ao estritamente necessário. Entende-se que a adição de outras relações ontológicas, tal como definição textual dos conceitos e sinônimos para elementos da hierarquia, podem enriquecer o modelo de BI durante o seu Ciclo de Uso, com telas específicas de suporte no entendimento dos conceitos pelos usuários finais.

A decisão de construção de um site próprio em detrimento ao uso de repositórios ontológicos existentes (i.e., Protégé Server, Align API, entre outros) teve como princípio o favorecimento não só do processo de carga e publicação de ontologias mas também da oferta de um modelo simples quanto aos mecanismos de troca de informações entre Organizações Participantes e a Administração do Ambiente de BI Colaborativo, tal como a disponibilização da "Proposta de Serviços de Dados".

Da interface do ambiente de BI, outros padrões gerados pela comunidade livre, tal como Saiku e STPivot são facilmente adaptáveis, já que as soluções das situações de "não relacionamentos" remetem ao controle das opções do servidor OLAP (Cubos Virtuais e *ValidMeasures*).

Cabe citar que, durante a execução da Prova de Conceito, alguns limites ainda se apresentam no que se refere ao tratamento de todas as situações baseadas em Cubos Virtuais, servidor OLAP Mondrian e JPivot. Estudos adicionais apontam que o recurso de Cubos Virtuais está sendo otimizado e melhorado em comunidades de soluções existentes.

5.5 EXEMPLOS DE USO NO INTERESSE PÚBLICO

De um dos benefícios apontados no uso desta arquitetura destacamos o uso no interesse público (Ludwig, 2005). Esse uso, para gestores de mais alto nível que precisam realizar análise pelo cruzamento de medidas entre organização e definição de indicadores.

Quando as medidas a serem analisadas tem origens diferentes mas são associadas ao mesmo modelo de conceitos, a situação é tratada como um modelo de BI convencional com as duas medidas. Porém, dada a arquitetura, a origem dos dados não é realizada através de trocas intempestivas de dados, mas em um acesso único aos gestores específicos do dado, conforme ilustrado na Figura 5-33 que relaciona o dado IDH oriundo da organização que trata "Demografia" com o dado IDEB que é tratado pela organização gestora do assunto "Educação".

		Measures		
Tempo	Geo	● Relacao Indices	● IDEB	● IDH
+ 2009	- CO	0,06	16,40	297,22
	+ DF	5,83	4,90	,84
	+ MS	0,07	3,60	53,04
	+ MT	0,05	3,70	81,49
	+ GO	0,03	4,20	161,85
	- N	0,09	23,70	264,60
	+ RR	0,40	3,80	9,58
	+ AP	0,35	3,20	9,27

Figura 5-33 – Extrato do resultado do indicador e dos valores de IDEB e IDH em 2009.

Quando a relação necessita da observação através de dimensões "não relacionadas" no caso de uma análise da área de "Imóveis Públicos" em relação a área da "Demografia com o suporte a informações de nível de "Urbanização", um campo calculado de "Relação Áreas" é adicionado, como ilustrado na Figura 5-34, e que permite o relacionamento ignorando adequadamente o desdobramento do nível de Urbanização.

		Measures				
		Relacao Areas		Area Imoveis		Area
		Urbanizacao		Urbanizacao		Urbanizacao
Tempo	Geo	● Total Urbanizacao	● Total Urbanizacao	● RURAL	● URBANO	● Total Urbanizacao
[-] Total Tempo	[+] Total Geo	205,44	8730235132,52	5526833656,63	3203401475,89	42496056,80
[+] 2009	[-] Total Geo	83,36	708470260,96	308732492,09	399737768,87	8499211,36
	[+] S	310,53	174666635,99	20496679,72	154169956,27	562481,71
	[+] NE	213,13	330895897,31	131168347,86	199727549,45	1552519,32
	[+] SE	178,92	165418024,42	155711035,92	9706988,50	924511,42
	[+] CO	22,40	35984658,82	25992,12	35958666,70	1606371,61
	[+] N	0,39	1505044,42	1330436,47	174607,95	3853327,30

Figura 5-34 – Relação de Área de imóveis e de Regiões por Nível de Urbanização.

Em outro exemplo, um exemplo extremo no qual falta um suporte de relacionamento em dimensões e em nível de granularidade entre domínios diferentes, a arquitetura ainda permite uma visão de relações, porém mais genérica e global. A Figura 5-35 ilustra uma associação de medidas dentro de elementos do domínio de Equipamento (domínio Saúde) com uma medida externa, a "População", oriunda da "Demografia". Um indicador geral - "Qtd Atendida/Qtd Recomendada/População" - é populado, porém limitado ao seu valor global.

		Measures			
Equipamento		● Atendida/Recomendada/Populacao	● Populacao	● Qtd Atendida	● Qtd Recomendada
[-] Total Equipamento		5,2239	191429257,00	236435,00	153247,21
DIALISE_3_TURNOS				65353,00	30281,84
DIALISE_USO_DOMICILIAR				3878,00	22686,89
MAMOGRAFO				8239,00	1973,29
OSTEO-DENSIMETRO				3419,00	17056,39
ULTRASSONOGRAFIA_E_RADIOLOGIA				149460,00	78143,86

Slicer: [Ano=2009]

Figura 5-35 – População dentro do domínio da Saúde.

Porém sob a ótica de análise desse indicador específico, seria de interesse público uma análise desta valoração no tempo, que permite ao gestor inferir se o atendimento de equipamentos (ou de tipos de equipamentos específicos) está correspondendo no tempo com o crescimento populacional referente. A Figura 5-36 ilustra esta situação.

	Measures			
Tempo	● Atendida/Recomendada/Populacao	● Populacao	● Qtd Atendida	● Qtd Recomendada
+ 2005				
+ 2006		186705182,00		
+ 2007	5,3589	186606849,00	96421,00	96494,34
+ 2008	5,2761	189535546,00	241583,00	148844,09
+ 2009	5,2239	191429257,00	236435,00	153247,21
+ 2010	5,1314	194880355,00	293495,00	242111,65

Figura 5-36 – Atendimento de Equipamentos em relação a População no Tempo.

Como visto através desses exemplos, o uso no interesse público deve apresentar um foco essencial na relação entre métrica, assim, as tarefas adicionais de criação de indicadores pré-definidos dentro do repositório ontológico centralizado para uso geral é passível de implementação na arquitetura proposta e atende as necessidades apresentadas por Berthold et al. (2010).

Por fim, cabe-nos citar que os números presentes neste trabalho são, em geral, de origem aleatória, sendo utilizados meramente com caráter de comprovação da realização da Prova de Conceito estabelecida.

6 - CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresenta uma proposta arquitetural de um ambiente colaborativo de BI no qual a realização de consultas analíticas baseia-se na integração de componentes: de um repositório ontológico centralizado de conceitos e no uso de serviços de dados interoperáveis descentralizados. Esses componentes são integrados por elementos básicos de uma solução sistêmica de BI devidamente adaptada.

Neste sentido, busca-se descrever a importância de se desenvolver, através do uso de padrões do domínio, uma arquitetura-modelo com baixo acoplamento entre todas as camadas de um ambiente de BI, e que esse desacoplamento possa ser facilmente adaptável junto às soluções existentes, focando nos princípios de uso de ontologias para representação de conceitos, e de serviços de dados distribuídos para resposta a dados com baixa latência.

A abordagem apresentada se afasta de um ambiente de BI convencional em diversos aspectos, mas principalmente seus dois princípios fundamentais. Em primeiro lugar, esta proposta baseia-se na integração de heterogêneos conceitos semânticos, de modo a compor ontologias superiores. Além disso, a recuperação da fonte de dados é realizada de acordo com práticas fundamentais de SOA, tais como baixo acoplamento e abstração.

Desta forma, entende-se que a arquitetura proposta constitui uma alternativa melhorada para a estrutura convencional de BI, oferecendo uma solução que é capaz de integrar conceitos heterogêneos de informação e entregando a responsabilidade desses e dos dados correlatos para os parceiros, garantindo uma independência funcional do ambiente de disponibilização de serviços de BI da gestão dos conceitos e dos dados, que são realizados dentro dos limites de cada organização.

6.1 APLICAÇÕES DA ARQUITETURA PROPOSTA

Com a arquitetura proposta, redes de informações estratégicas podem ser criadas vinculando conceitos e dados de vários parceiros. Entende-se que essa integração pode ser melhor alcançada se a camada de serviço de dados de cada parceiro envolvido está em conformidade com uma arquitetura de BI, isto é, já tiver seus dados agregados e tratados em soluções locais de BI. Além disso, o uso de outras plataformas sistêmicas com maior completude pode simplificar a construção de um ambiente.

Tarefas adicionais ainda podem ser executadas de modo a criar indicadores pré-definidos dentro do repositório ontológico centralizado para uso geral. Essas tarefas satisfazem os requisitos propostos por Berthold et al. (2010).

6.2 ATINGIMENTO DE METAS DA MOTIVAÇÃO

Das quatro situações apontadas na motivação, como soluções atingidas destaca-se:

a) A centralização de conceitos, que interpõem a responsabilidade pela manutenção dos modelos de conceitos a cada organização.

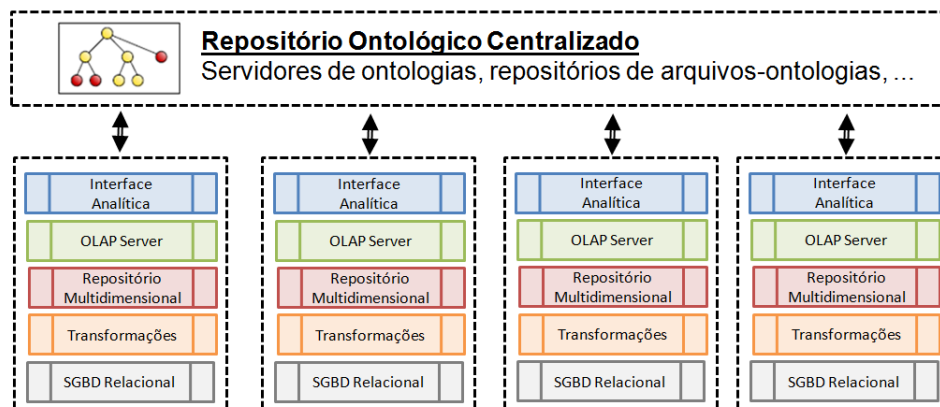


Figura 6-1 – Motivação para Centralização de Conceitos

b) O acesso de dados como serviço, que delega a responsabilidade na conceituação e na forma dos dados à cada organização além de equalizar a latência dos mesmos.

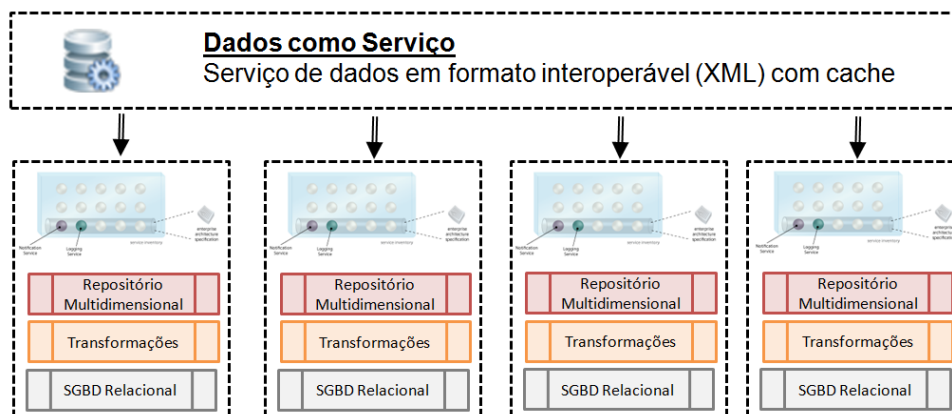


Figura 6-2 – Motivação para o uso de Dados como Serviços

c) Cada organização e a Administração observam, durante um Ciclo de Construção, os problemas de alinhamento de conceito sem afetar os modelos dos outros participantes.

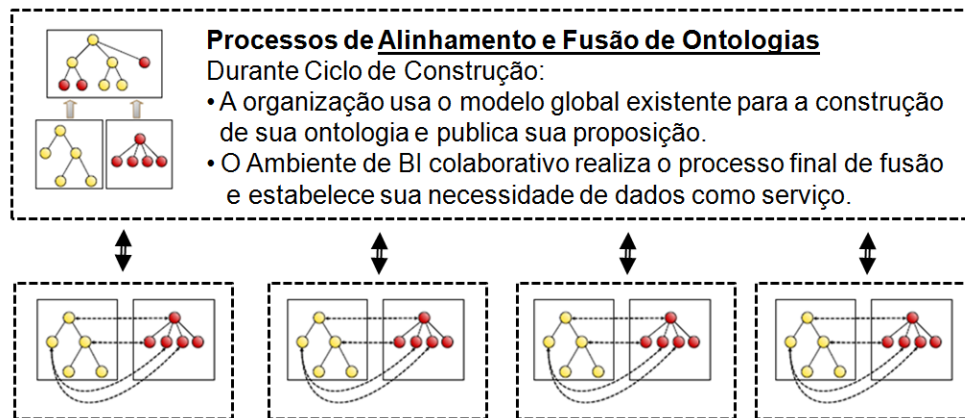


Figura 6-3 – Motivação para o uso de processos de Alinhamento e Fusão

d) Processos automáticos usam o desacoplamento de dados e de conceitos para identificar as dimensões não relacionadas, disponibilizando aos usuários finais essas considerações.

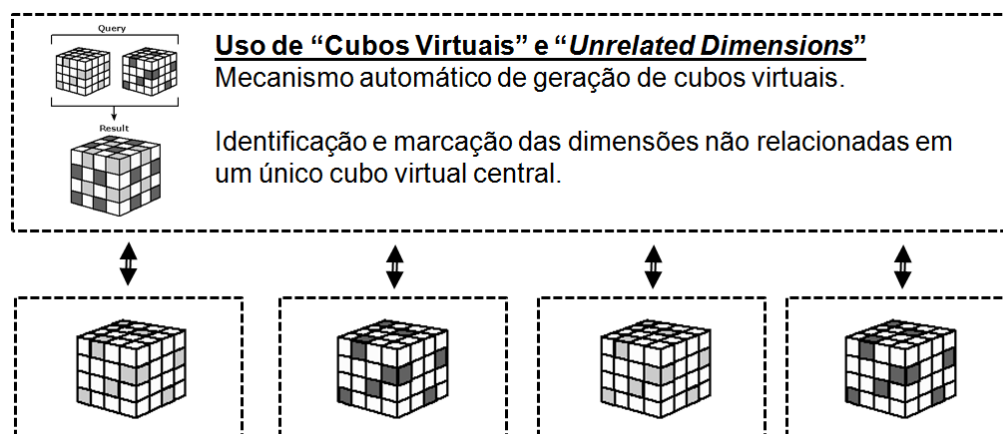


Figura 6-4 – Solução dos problemas de "não relacionamentos" de modelos

6.3 ATINGIMENTO DOS OBJETIVOS

Dos objetivos gerais, com o uso do Protégé e a implementação do Repositório Centralizado - Site UBI, este trabalho permitiu a construção de um ambiente sistêmico com padrões que permitiram os sucessivos passos de alinhamento de modelos exemplo em um único modelo.

Com o suporte da "Proposta de Serviço de Dados" e a geração automática de consumidores de dados, este trabalho considera ter elaborado um modelo de fácil integração - no que se refere a dados - de cada Organização Participante com o ambiente colaborativo de BI proposto. Como exposto na seção, mecanismos tecnológicos que carreguem dados de tabelas e disponibilizem via Internet como arquivos XML podem ser facilmente implementados ou podem ser utilizados quaisquer serviços de provimento de dados de

soluções existentes, já que a própria Organização Participante é responsável por apontar a URI de seus serviços de dados junto com a ontologia de domínio que a mesma expõe.

Isto se dá pois o processo central de "Fusão" do site UBI permite a geração de toda configuração para o servidor analítico OLAP, desde que, como premissa, o mesmo apresente recursos de Cubos Virtuais e de "dimensões não relacionadas" (ambientes de BI Pentaho e da Microsoft já apresentam esses recursos).

No cumprimento do objetivo específico, pode-se considerar que:

(a) Nos Capítulos "Conceituação" e "Estado da Arte" são apresentados os estudos e definições necessários na elaboração e teste da arquitetura proposta.

(b) No Capítulo "Proposta de Arquitetura" e "Estado da Arte" é apresentado o projeto de modelo arquitetural, e identificado os elementos passíveis de serem utilizados para formação da arquitetura, bem como é apontada as atividades necessárias para todo ciclo de vida de incorporação de um requerente à arquitetura;

(c) Com base no Capítulo "Estado da Arte", no Capítulo "Prova de Conceito e Resultados Obtidos", este trabalho propôs, junto às soluções de software existentes, plataformas a serem utilizadas, e detalha os mecanismos e mensagens utilizados nessa integração;

(d) Ainda no Capítulo "Prova de Conceito e Resultados Obtidos" foi colocado sob prova a proposta quatro cenários - Demografia, Educação, Imóveis Públicos e Saúde - simulando organizações distintas com informações em hierarquias diferenciadas, e executando situações de análise de dados *ad-hoc* (operações multidimensionais) específicas das questões a serem tratadas, comprovando a realização pela apresentação dos resultados obtidos.

Além disso, os resultados apontados na Tabela 5-2 demonstram que a solução favorece o uso em estruturas e quantidades de dados com foco estratégico (baixo número de registros), para suporte a um tempo de resposta razoável. Assim, é importante uma análise aprofundada do envolvimento de dimensões com uma alta distribuição de informações, tal como município (com 5.565 instâncias-membro), aonde ocorre uma degradação natural do tempo de resposta do BI acima do valor da maior carga realizada.

6.4 TRABALHOS FUTUROS

Apesar da arquitetura proposta ter um modelo de fácil implementação algumas questões relacionadas ao desempenho são identificadas. Assim, destacam-se algumas melhorias futuras para resolver essas questões: o uso de mecanismos otimizados de transporte de dados com a evolução dos padrões de SOA e de *Cloud Computing*, a implementação de esquemas de *cache* no servidor OLAP e nos serviços de dados, o estabelecimento de restrições a quantidade de dados que podem ser gerados, a implementação de melhores e mais eficientes técnicas de compressão para transferir dados dos serviços de dados para o servidor OLAP.

Um aspecto a ser observado é a provável capacidade de automatizar tarefas nos processos de alinhamento e fusão. No modelo proposto é necessário um Administrador para verificar se o nível de consistência mínimo foi atingido, pois uma solução totalmente automatizada pode afetar modelos existentes de outros parceiros.

O compromisso de cumprir plenamente a automação em um único passo ainda não parece ser razoável, uma vez que este processo exige troca de informações sobre similaridade entre ontologias em graus indefinidos, o que entende-se como uma atividade que necessita ainda da intervenção humana. No entanto, esse esforço pode ser minimizado se os elementos selecionados para a camada de ontologia (cliente e servidor) forem capazes de usar uma rica combinação de diferentes abordagens para a integração de ontologias.

Em outro ponto de destaca, como herança consistente das preocupações da aplicação de Arquitetura Orientada a Serviços - SOA e dos seus ambientes de implementação em tecnologias de Computação em Nuvem, é que a arquitetura proposta necessita de trabalhos futuros voltados ao tratamento de problemas de segurança na comunicação entre todos os ambientes dessa arquitetura através um modelo de confiança, para estabelecer uma relação segura entre as organizações participantes do modelo unificado e centralizado.

Dos aspectos de desempenho que são claramente observados desde a proposição, que foram convalidados junto a prova de conceito, destaca-se algumas hipóteses e recursos futuros para o tratamento específico desse aspecto:

- O processamento específico e otimizado de mensagens SOAP ou REST em evolução nas modernas plataformas de SOA podem diminuir algum dos impactos negativos existentes.
- Uso de mecanismos mais complexos e eficientes de cache nos servidores e de compactação na transferência de dados.
- Associação ao cache de dados aos processos de serviços de dados nos parceiros por troca de informações sobre alterações realizadas na origem e que necessitam ser carregadas (*handshaking*).
- O uso adicional de uma abordagem baseada em situações de consultas comuns, que fornece modelos de consultas pré-definidos e pré-ligadas que preenchem as necessidades de cenários explícitos ou de uso recorrente.

Por fim, um trabalho futuro de grande interesse seria a modelagem de um sistema totalmente automático de integração de ontologias baseados em agentes que pudessem utilizar “serviços de conhecimento” - que expõem ontologias como serviços - e realizar o tratamento *on-line* do alinhamento ontológico. Isto posto, seria possível a criação de redes *ad-hoc* de conhecimento e dados, através de agentes que podem implementar os serviços de BI necessários.

TRABALHOS PUBLICADOS

1. **V. A. Martins**, J. P. C. L. da Costa and R. T. de Sousa Jr., "**Architecture of a Collaborative Business Intelligence Environment based on an Ontology Repository and Distributed Data Services**", in Proc. 4th International Conference on Knowledge Management and Information Sharing (KMIS'2012), Barcelona, Spain, Oct. 2012.
2. A. A. Fernandes, L. C. Amaro, J. P. C. L. da Costa, **V. A. Martins**, A. M. R. Serrano and R. T. de Sousa Jr., "**Construction of Ontologies by using Concept Maps: a Study Case of Business Intelligence for the Federal Property Department**", in Proc. International Conference on Business Intelligence and Financial Engineering (BIFE'2012), (Lanzhou & Tunhuang, China), Aug. 2012.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Abels, S., Haak, L., Hahn, A. Identification of Common Methods Used for Ontology Integration Tasks. University of Oldenburg - Business Information Systems- Oldenburg, Germany. 2005. Disponível em: <<http://dl.acm.org/citation.cfm?id=1096983&dl=ACM&coll=DL&CFID=70638768&CFTOKEN=65789288>>
- [2] Abolhassani, H., Hariri, B., Haeri, S. On Ontology Alignment Experiments. Webology, Volume 3, Number 3, September, 2006 Disponível em: <<http://www.webology.org/2006/v3n3/a28.html>>
- [3] Amara, Y. Business Intelligence Software Evaluation: Testing the SSAV Model, Master Thesis in Business Administration. Blekinge Institute Of Technology. 10/06/2008. Disponível em: <<http://studentyogi.com/wp-content/uploads/2010/02/Business-Intelligence-Software-Evaluation-Testing-the-SSAV-Model.doc>>
- [4] Andrejko, A., Barla, M., Tavarozek, M. Comparing Ontological Concepts to Evaluate Similarity. In: Tools for Acquisition, Organisation and Presenting of Information and Knowledge, Rese'arch Project Workshop Bystrá dolina, Nízke Tatry, Slovak University of Technology. 2006. pp. 71-78. - Bratislava, Slovakia. Disponível em: <<http://nazou.fiit.stuba.sk/home/misc/itat2006-andrejko-et-al.pdf>>
- [5] Anzanello, C. A. Olap Conceitos e Utilização. Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS) Porto Alegre - RS - Brasil. 2005. Disponível em: <[http://www.fag.edu.br/professores/limanzke/Administração de Sistemas de Informação/OLAP.pdf](http://www.fag.edu.br/professores/limanzke/Administração%20de%20Sistemas%20de%20Informação/OLAP.pdf)>.
- [6] Aumueller, D., Do, H.-H., Massmann, S., Rahm, E. Schema and Ontology Matching with COMA++. Department of Computer Science, University of Leipzig, Augustusplatz 10/11, Leipzig 04103, Germany. 2005. Publicação: <<http://disi.unitn.it/~p2p/RelatedWork/Matching/COMA++-SIGMOD05.pdf>>
- [7] Barbieri, C.. Business Intelligence: Modelagem & Tecnologia. Rio de Janeiro: Axcel Books, 2001. 424 p.

- [8] Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., Munro, M. (2000) Service-based software: the future for flexible software. Asia-Pacific Software Engineering Conference, vol. 0, p. 214.
- [9] Berners-Lee, T., Lassila, O., HENDLER, J. The semantic web. Scientific America. In.: Scientific American Magazine Maio 2001. Disponível em: <<http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>>.
- [10] Berthold, H., Rösch, P., Zöller, S., Wortmann, F., Carenini, A., Campbell, S., Bisson, P., Strohmaier, F. An architecture for ad hoc and collaborative business intelligence. ACM International Conference Proceeding Series, ACM, 2010. Disponível em: <http://www.edbt.org/Proceedings/2010-Lausanne/workshops/beweb/papers/edbt_2010_submission_565.pdf>
- [11] Bikakis, N., Tsinaraki, C., Gioldasis, N., Stavrakantonakis, I., Christodoulakis, S. The XML and Semantic Web Worlds: Technologies, Interoperability and Integration. A survey of the State of the Art In Semantic Hyper/Multi-media Adaptation: Schemes and Applications, Springer 2012.
- [12] Blakeley, J. A., Pizzo, M. J. Microsoft universal data access platform. Proceedings of the 1998 ACM SIGMOD international conference on Management of data - SIGMOD '98. Disponível em: <<http://dl.acm.org/citation.cfm?id=276354>>
- [13] Borst, W. N. Construction Of Engineering Ontologies For Knowledge Sharing And Reuse. Thesis Enschede - ISBN: 90-365-0988-2 - ISSN: 1381-3617 (CTIT Ph. D-series No. 97-14) - W. N. Borst, Enschede, The Netherlands - CTIT Ph. D-thesis series No. 97-14 - The Netherlands Information Technology>.
- [14] Bruijn, Jos d., Ehrig, M., Feier, C., Martín-Recuerda, F., Scharffe, F., Weiten, M. Ontology mediation, merging and aligning. In Semantic Web Technologies. Wiley, 2006. May 20, 2006
- [15] Böhringer, M., Gluchowski, P., Kurze, C., Schieder, C. A Business Intelligence Perspective on the Future Internet. Americas Conference on Information Systems

- (AMCIS). AMCIS 2010 Proceedings. Paper 267, Disponível em: <<http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1263&context=amcis2010>>
- [16] Caldwell, F., Knox, R. E., Logan, D. Taxonomies Generates Return on Investment, Research Report LE-20-9654, Gartner, Inc., Sept 10th, 2003.
- [17] Calvanese, D, Giacomo, G., Lenzerini. M. A framework for ontology integration. In Proceedings of the 1st Internationally Semantic Web Working Symposium (SWWS), Stanford, CA, USA, August 2001[a].
- [18] Calvanese, D, Giacomo, G., Lenzerini. M. Ontology of integration and integration of ontologies. In Proceedings of the 9th International Conference on Conceptual Structures (ICCSf01), Stanford, CA, USA, August 2001[b].
- [19] Cao, L., Zhang, C., Liu, J. Ontology-based integration of business intelligence. Web Intelligence and Agent Systems, 4(3):313-325, 2006.
- [20] Cardoso, J. On the Move to Semantic Web Services. World Academy of Science, Engineering and Technology 8 2005.
- [21] Carvalho Jr, A. F. Data Warehouse: Melhorando o Desempenho Acadêmico. Universidade Federal da Bahia - Instituto De Matemática - Departamento De Ciências Da Computação - Monografia de Graduação - Salvador - Ba, Julho/2006. Disponível em: <<https://twiki2.dcc.ufba.br/pub/MATA67/TrabalhosSemestre20061/Projeto.VersaoFinal.PDF>>
- [22] Casters, M. PDI data over web services. Sitio da Pentaho, Seção do Pentaho Data Integration. 2011. Disponível em: <<http://wiki.pentaho.com/display/EAI/PDI+data+over+web+services>>
- [23] Chaudhuri, S., Dayal, U. An overview of data warehousing and OLAP technology, ACM SIGMOD Record, vol.26, no.1, pp.65-74, March 1997, doi: 10.1145/248603.248616. Pages 65-74. ACM New York, NY, USA Disponível em: <http://delivery.acm.org/10.1145/250000/248616/P065.pdf?ip=164.41.46.30&acc=ACTIVE%20SERVICE&CFID=110880164&CFTOKEN=47387419&__acm__=1339609472_5ccef062ea8eebb04eeef00afde1143f>

- [24] Chaudhuri, S., Dayal, U., Narasayya, V. An overview of business intelligence technology. *Commun. ACM* 54, August 2011, 88-98. DOI=10.1145/1978542.1978562.
- [25] Clear, A.K., Knox, S., Ye, J., Coyle, L., Dobson, S. Integrating multiple contexts and ontologies in a pervasive computing framework. 2006. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.818&rep=rep1&type=pdf>>
- [26] Codd, E.F., Codd, S. B., Salley, C. T. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. E.F. Codd Associates white paper, commissioned by Arbor Software (now Hyperion Solutions). 2005. Disponível em: <http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf>
- [27] Cormen, T., Leiserson, C., Rivest, R., Stein, C. Introduction to Algorithms, Second Edition. The MIT Press - 2001. Disponível em <[http://www.civilnet.cn/book/program/Introduction To Algorithms 2th Edition/DDU0251.html](http://www.civilnet.cn/book/program/Introduction%20To%20Algorithms%202th%20Edition/DDU0251.html)>
- [28] Curbera, F. Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. March/April 2002.
- [29] Dav, A. S., Padmanabhuni, S. XML Compression and its Role in SOA Performance. In: *Jornal SOAWebServices*, n7 v6 p48 (2) Data de publicação: Jul 2006 - ISSN: 1535-6906. Disponível em: <<http://www.tectrends.com/tectrends/article/00162194.html>>.
- [30] Davenport, T. H., Harris, J. G. *Competing on Analytics: The New Science of Winning*. Harvard Business School Press, 2007, pp 240.
- [31] Delis, A., Faloutsos, C., Ghandeharizadeh, S. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data: SIGMOD '99*, Philadelphia, PA, USA, June 1-3, 1999, Volume 28
- [32] Delugach, H. ISO/IEC WD 24707 Information technology - Common Logic (CL): A Framework for a Family of Logic-Based Languages. Pacific Northwest National Laboratory, Chantilly, VA, 7 June 2004.

- [33] Djuric, D., Gasevic, D., Devedzic, V. Ontology Modeling and MDA. Journal of Object Technology. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2005, Vol. 4, No. 1, January-February 2005. Disponível em: <http://www.jot.fm/issues/issue_2005_01/article3>
- [34] Ehrig, M., Staab, S. QOM: Quick Ontology Mapping: The State of Art. The Knowledge Engineering Review, 2003.
- [35] Ehrig, M., Staab, S., Sure, Y. Bootstrapping ontology alignment methods with APFEL. In: Proceedings of the 4th International Semantic Web Conference (ISWC-2005), ser. Lecture Notes in Computer Science, Y. Gil, E. Motta, and R. Benjamins, Eds., 2005, pp. 186-200.
- [36] Ehrig, M., Sure, Y. Ontology mapping - an integrated approach. In: Proceedings of the European Semantic Web Symposium (ESWS), May 2004, pp. 76-91.
- [37] Erl, T. Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005
- [38] Euzenat, J., Barrasa, J., Bouquet, P., Bo, J.D., et al. (2004). State of the Art on Ontology Alignment. Knowledge Web, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, deliverable 2.2.3, 2004.
- [39] Euzenat, J., Valtchev, P. An integrative proximity measure for ontology alignment. In: Proceedings of Semantic Integration workshop at ISWC, 2003.
- [40] Euzenat, J., Valtchev, P. Similarity-based ontology alignment in OWL Lite. In: Proceedings of ECAI, 2004, pp. 333-337.
- [41] Evelson, B., Moore, C., Karel, R., Kobielus, J., Coit, C. Forrester's BI Maturity Assessment Tool. September 11, 2009, Forrester Research, Inc. Disponível em : <<http://pt.scribd.com/doc/76597995/Forresters-Bi-Maturity-Assessment-Tool>>
- [42] Fatland, D., Open Data Protocol (ODATA): A Practical Web Protocol For Data Query And Retrieval. In: 2011 GSA Annual Meeting in Minneapolis. The Geological Society of America -GSA. 9-12 October 2011.

- [43] Ferreira, M., Baptista, A. A. The use of Taxonomies as a way to achieve Interoperability and improved Resource Discovery in DSpace-based Repositories. 2005. Disponível em: <<http://repositorium.sdum.uminho.pt/bitstream/1822/873/1/paper-25.pdf>>, Apresentação disponível em: <<http://repositorium.sdum.uminho.pt/bitstream/1822/873/2/presentation.pdf>>
- [44] Foshay , N., Mukherjee, A., Taylor, A. Does data warehouse end-user metadata add value?. *Communications of the ACM*, 50(11):77, 2007.
- [45] Franklin , M., Halevy , A., Maier , D. From Databases to Dataspaces: A New Abstraction for Information Management. *ACM SIGMOD Record* December 2005
- [46] Galatis, G., Tsekeridou, S. Business Intelligence On 2.0 Era: A Framework Described With Emphasis On Opinion Mining, Thesis submitted for Master Of Science. Management of Business, Innovation & Technology (MBIT), Athens, 2011, Unpublished.
- [47] Garshol, L. M. Metadata? Thesauri? Taxonomies? Topic Maps!, *Ontopia*, 2004.
- [48] Geer, D. Will Binary XML Speed Network Traffic? Published by the IEEE Computer Society. April 2005 (Vol. 38, No. 4) pp. 16-18. 0018-9162© 2005 IEEE
- [49] Gilchrist, A. Thesauri, taxonomies and ontologies - an etymological note, *Journal of Documentation*, Vol. 59 Iss: 1 pp. 2003. Disponível em: <<http://dx.doi.org/10.1108/00220410310457984>>
- [50] Giunchiglia, F., Zaihrayeu, I. Lightweight Ontologies. Technical Report DIT-07-071, *Ingegneria e Scienza dell'Informazione*, University of Trento. 2007.
- [51] Goldstone, R. L., Rogosky, B. J. Using relations within conceptual systems to translate across conceptual systems. 2002. *Cognition* 295-320.
- [52] Golfarelli, M., Rizzi, S., Cella, I. Beyond data warehousing: what's next in business intelligence? In *Proceedings of the 7th ACM international workshop on Data warehousing and OLAP*, 2004, pages 1-6. ACM New York, NY, USA.
- [53] Gruber, T. What is an Ontology? *Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009.

- [54] Gruber, T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, p.907-928, 1993.
- [55] Guarino, N. Understanding, building and using ontologies. In: *Proceedings of Knowledge Acquisition for knowledge-based systems workshop. Knowledge Acquisition Workshop - KAW'96*, Banff, Alberta, Canada, 1996.
- [56] Guarino, N., Giaretta, P. Ontologies and Knowledge Bases Towards a Terminological Clarification. In Mars, N., editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. 1995. IOS Press. Pages 25-32, Amsterdam, NL. IOS Press. Disponível em <<http://www.loa.istc.cnr.it/Papers/KBKS95.pdf>>
- [57] Herrman, M., Aslam, M. A., Dalferth, O. (). Applying Semantics (WSDL, WSDL-S, OWL) in Service Oriented Architectures (SOA). Sindelfingen and Universität Leipzig, Germany.
- [58] Holler, C. Estudo Comparativo Entre Diferentes Tecnologias Para Concepção de uma Solução de BI Aplicada a uma Empresa de Desenvolvimento de Software. Universidade FEEVALE - Instituto de Ciências Exatas e Tecnológicas - Curso de Ciência da Computação - Anteprojeto de Trabalho de Conclusão - Novo Hamburgo, setembro de 2010.
- [59] Hui, M., Jiang, D., Li, G., Zhou, Y. Supporting database applications as a service. In: *ICDE '09: Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 832-843. IEEE Computer Society.
- [60] Inmon, W. H. *Como construir o Data Warehouse*. Campus. Rio de Janeiro. 1997.
- [61] JPivot Disponível em <<http://jpivot.sourceforge.net/>>. 2011.
- [62] Jacobson, R., Misner, S. *Microsoft SQL Server 2005 Analyses Services: Passo a passo*. Hitachi Consulting.. Porto Alegre, Bookman, 2007. 352p.
- [63] Jasper, R., Uschold, M. A framework for understanding and classifying ontology applications. *IJCAI-99, Ontology Workshop, Stockholm, S. 1. : s./n*, 1999.

- [64] Kalfoglou, Y., Hu, B. Cross mapping system (CMS) results of the 2005 ontology alignment contest. In: Proceedings of K-Cap'05 Integrating Ontologies workshop, 2005, pp. 77-85.
- [65] Kalfoglou, Y., Schorlemmer, M. Information-flow-based ontology mapping. In On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE. Lecture Notes in Computer Science 2519, pages 1132.1151. Springer, 2002.
- [66] Kansa, E. C., Bissell, A. Web Syndication Approaches For Sharing Primary Data in Small Science Domains. School of Information, Data Science Journal, Volume 9, 8 July 2010.
- [67] Kempf, J., Nikander, P., Green, H. Innovation and the Next Generation Internet. INFOCOM IEEE Conference on Computer Communications Workshop, March 2010, pp.1-6, 15-19, doi: 10.1109/INFCOMW.2010.5466657.
- [68] Kimball , R., Ross, M., Thornthwaite, W., Mundy, J. The Data Warehouse Lifecycle Toolkit. Wiley & Sons, 2008.
- [69] Kimball, R. Data Warehouse Toolkit. São Paulo: Makron, 1998, 388p.
- [70] Kimball, R. The Data Warehousing Lifecycle Toolkit: expert methods for designing, developing, and deploying data warehouses. New York: John Wiley & Sons, 1998.
- [71] King, C. Managing Data Sources with XSLT. - Artigo na Revista XMLJournal. Maio de 2002. Disponível em: <<http://xml.sys-con.com/node/40411>>
- [72] Kiryakov, A., Simov , K., Dimitrov , M. OntoMap: Portal for Upper-Level Ontologies. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOISf01), Ogunquit, Maine, USA, October 2001.
- [73] Küngas, P., Dumas, M. Configurable SOAP proxy cache for data provisioning web services. In: Proceedings of the 2011 ACM Symposium on Applied Computing. Pages 1614-1621. 2011. Disponível em: <http://dl.acm.org/citation.cfm?id=1982523>

- [74] LaMonica, M. Putting XML in the fast lane. CNET News.com. January 13, 2005. Disponível em <<http://www.techrepublic.com/article/putting-xml-in-the-fast-lane/5534881>>
- [75] Lafuente, F. Dossiê Web 3.0. Revista HSM Management. Maio/Junho de 2011.
- [76] Levy, P., Authier, M. Árvores do Conhecimento. Editora Escuta - 1995. ISBN-10: 8571370923. ISBN-13: 9788571370920. 190p.
- [77] Liu, X., Lou, X. A Data Warehouse Solution For E-Government. International Journal of Research and Reviews in Applied Sciences - IJRRAS, vol.4, issue 1. July 2010.
- [78] Ludwig, L. Business Intelligence und das Semantic Web, Business Intelligence und das Semantic Web: ein Traumpaar, 2005.
- [79] MacDonald, G. C., Rubik, J. R., Pesquisa e Seleção de Ferramentas Livres e Baseadas em Padrões de Sistemas Abertos para a Elaboração de Interfaces - OLAP sobre a Web. Universidade de Santa Catarina - Florianópolis - SC - 2007 / 1 - TCC-CarusoRubik-20070716.doc>
- [80] Machado, F. N. R. Projeto de Data Warehouse: Uma Visão multidimensional. São Paulo: Érica, 2000, 248p.
- [81] Madhavan, J., P.A. Bernstein, A.H. Doan, A.Y. Halevy. Corpus based Schema Matching. Int. Conf. of Data Engineering (ICDE). 2005.
- [82] Mahboubi, H., Bimonte S., Deffuant. G. Analyzing Demographic and Economic Simulation Model Results: A Semi-automatic Spatial OLAP Approach. Lecture Notes in Computational Science and Its Applications - ICCSA 2011, 2011, Volume 6782/2011, 17-31, DOI: 10.1007/978-3-642-21928-3_2. Disponível em: <<http://www.springerlink.com/content/w386623r5r1q823q/>>
- [83] Manual Técnico Orçamentário. Versão 2013. liberada em 2012. Disponível em <https://www.portalsof.planejamento.gov.br/bib/MTO/MTO_2013_4.pdf>
- [84] Marchetti, A., Ronzano, F., Tesconi, M., Minutoli, M. Formalizing Knowledge by Ontologies: OWL and KIF. Relatório apresentado L'Istituto di Informatica e

- Telematica (IIT). Consiglio Nazionale delle Ricerche (CNR). Italia.. 2008. Disponível em: <http://www.iit.cnr.it/sites/default/files/TechnicalReport_IIT_FormalizingKnowledgeByOntologies.pdf>
- [85] Martins, D. Aplicação de Técnicas de Distribuição e Paralelismo em Ambientes de BI 2.0 para suporte à Qualidade de Dados. Dissertação de Mestrado. Programa de Pós.Graduação em Informática. Universidade Federal do Estado do Rio de Janeiro (UNIRIO). RIO de Janeiro, RJ. Brasil. Disponível em: <<http://www.cin.ufpe.br/~lma4/DW/qualidadeintro.pdf>>.
- [86] McGuinness, D.L., Fikes, R., Rice, J., Wilder, S. An Environment for Merging and Testing Large Ontologies: Principles of Knowledge Representation and Reasoning. Proceedings of the Seventh International Conference (KR2000). A. G. Cohn, F. Giunchiglia and B. Selman, editors. San Francisco, CA, Morgan Kaufmann Publishers. 2000.
- [87] Mochmann, E. E-Science Infrastructure for the Social Sciences. July 1, 2009. RatSWD_WP_ 115. Mochmann, Ekkehard, E-Science Infrastructure for the Social Sciences (July 1, 2009). RatSWD_WP_ 115. Available at SSRN: <http://ssrn.com/abstract=1460721> or <http://dx.doi.org/10.2139/ssrn.1460721>. Disponível em: <<http://dx.doi.org/10.2139/ssrn.1460721>>
- [88] Nelson, G. S. Business Intelligence 2.0: Are we there yet?. SAS Global Forum 2010 / Business Intelligence/Analytics. Paper 040-2010. Disponível em: <<http://support.sas.com/resources/papers/proceedings10/040-2010.pdf>>.
- [89] Noy, N. Ontology Mapping and Alignment. Fifth International Workshop on Ontology Matching collocated with the 9th International Semantic Web Conference ISWC-2010, Shangai, China.
- [90] Noy, N., McGuinness, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. 2000. Stanford University, Stanford, CA, 94305
- [91] Noy, N., Musen, M. A. Algorithm and Tool for Automated Ontology Merging and Alignment. From: AAAI-00 Proceedings. 2000. AAAI (www.aaai.org).

- [92] Noy, N., Musen, M. Evaluating Ontology-Mapping Tools: Requirements and Experience. Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002). 2002. Disponível em: <http://km.aifb.kit.edu/ws/eon2002/EON2002_Noy.pdf>
- [93] Noy, N., Musen, M. Anchor-prompt: using non-local context for semantic matching. In: Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), 2001, pp. 63-70.
- [94] Ogbuji, U. Compress XML files for efficient transmissionWho needs binary XML when we have good compression?. 09 Apr 2004. Disponível em <<http://www.ibm.com/developerworks/xml/library/x-tipcomp/index.html>>
- [95] Oliveira, K. Modelo para construção de ambientes de desenvolvimento de software orientados a domínio. 222f. Tese (Doutorado em Engenharia de Sistemas e Computação). COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro. 1999.
- [96] OntoWeb. Deliverable 1.3: A survey on ontology tools. 31st May, 2002.
- [97] Pan, W., Tang, T., Qiu, C. Web Service-Based Enterprise Data Service Center for Exchanging Common Data of Enterprise. Advanced Research on Computer Education, Simulation and Modeling. Communications in Computer and Information Science, 2011, Volume 176, 193-199, DOI: 10.1007/978-3-642-21802-6_31. Disponível em: <<http://www.springerlink.com/content/vn7230514w40711g/>>
- [98] Park, J., Oh, S., Ahn J. Ontology Selection Ranking Model for Knowledge Reuse, Expert Systems With Applications, 38(5), pp. 5133-5144, 2011, (ISSN: 0957-4174).
- [99] Pendse, N. OLAP Report., Principal of OLAP Solutions and Co-author of the OLAPreport.com. July 20,2002, Acessado em 10 Nov. 2011. Disponível em: <http://www.bi-verdict.com/fileadmin/dl_temp/16a5aaab1f60441386cce1a7b3476794/fasmi.htm>
- [100] Petrini et al. Inteligência de Negócios ou Inteligência Competitiva? Noivo Neurótico, Noiva Nervosa.. In: ENANPAD, 2006, Salvador. Anais do 30º Encontro

da ANPAD Associação Nacional dos Programas de Pós-graduação em Administração, 2006.

- [101] Primak, F. V. Decisões com B.I.. Business Intelligence. Editora Ciência Moderna - 2008 - 1a. Edição - ISBN: 8573937149 - ISBN-13: 9788573937145.
- [102] Rahm, E., Bernstein, P. A survey of approaches to automatic schema matching. VLDB Journal, 10(4):334-350, 2001.
- [103] Ramalho, R. Desenvolvimento e utilização de ontologias em Bibliotecas Digitais: uma proposta de aplicação. Tese apresentada ao Programa de Pós-Graduação em Ciência da Informação da Faculdade de Filosofia e Ciências da Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP, Campus de Marília, como requisito para obtenção do título de Doutor em Ciência da Informação. 2010. Disponível em: <http://www.athena.biblioteca.unesp.br/exlibris/bd/bma/33004110043P4/2010/ramalho_ras_dr_mar.pdf>
- [104] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), 1995. Disponível em: <<http://dl.acm.org/citation.cfm?id=1625914>>
- [105] Robredo, J. Ciência da informação e Web Semântica: Linhas convergentes ou linhas paralelas? Brasília: IBICT, 2010. Disponível em: <<http://hdl.handle.net/10482/7952>>. Acesso em: 20 maio 2011
- [106] Rodriguez, A. RESTful Web services: The basics. 06 Nov 2008. Disponível em: <<http://www.ibm.com/developerworks/webservices/library/ws-restful/>>
- [107] Siciliano, L. Data warehouse - Parte 1 e 2. 22 de abril de 2012. Disponível em <<http://leandrosiciliano.blogspot.com.br/2012/04/data-warehouse-parte-1.html>>
- [108] Silva Filho, A. M. Espaço Acadêmico. Revista Espaço Acadêmico. No. 58. Março de 2006. ISSN 1519.6186. Ano V. Disponível em: <<http://www.espacoacademico.com.br/058/58silvafilho.htm>>

- [109] Silva, E. Integração de Padrões de Análise e Ontologias de Domínio: Um Estudo de Caso no Domínio da Gestão Pública. Dissertação de Mestrado. Universidade de Viçosa. MG. 2008.
- [110] Singh, Harry S. Data Warehouse. São Paulo: Makron Books, 2001. 382p.
- [111] Sitio Align Api 4.3. 2011. Disponível em: <<http://alignapi.gforge.inria.fr/neontk/>>
- [112] Sitio BI-INSIDER.com. BI Market: Vendors and Products (2011). Sitio comercial. Artigo publicado em 20 de julho 2011. Disponível em : <<http://bi-insider.com/portfolio/bi-market-vendors-and-products-2011/>>
- [113] Sitio COMA 3.0. 2005. Disponível em: <<http://dbs.uni-leipzig.de/Research/coma.html>>
- [114] Sitio Codeplex.com. 2012. CodePlexProject Hosting for Open Source Software. Disponível em: <<http://odataphpproducer.codeplex.com/>>
- [115] Sitio H-Match. 2006. Disponível em <<http://islab.dico.unimi.it/hmatch/news.php>>
- [116] Sitio HSQLDB. Disponível em <<http://hsqldb.org>>. 2012.
- [117] Sitio IBM. Introduction to BI Architecture Framework and Methods. IBM Group. DB2 Data Management Software. IBM Corporation. 2004.
- [118] Sitio Jaspersoft OLAP. Disponível em: <<http://www.jaspersoft.com/jaspersoft-olap>>
- [119] Sitio MapOnto. 2005. Disponível em: < <http://www.cs.toronto.edu/semanticweb/maponto/index.html>
- [120] Sitio Microsoft (2006). Business Intelligence Guidelines Conceptual Framework. Microsoft Corporation.
- [121] Sitio Microsoft SQL Server Team. Microsoft, SAP, IBM, Citrix, Progress Software and WSO2 co-submit OData to OASIS for standardization. Artigo no TechNet Blogs do SQL Server Team Blog. 24 May 2012. Disponível em: <<http://blogs.technet.com/b/dataplatforminsider/archive/2012/05/24/microsoft-sap>

ibm-citrix-progress-software-and-wso2-co-submit-odata-to-oasis-for-standardization.aspx>

- [122] Sitio Microsoft. Library. Microsoft Corporation. Disponível em: <<http://msdn.microsoft.com/pt-br/library/>>. Acesso em: Novembro de 2011.
- [123] Sitio Microstrategy Analysis Server. Componente da MicroStrategy Reporting Suite. 2010. Disponível em: <<http://www.microstrategy.com/freereportingsoftware/learnmore/microsoft-analysis-services-ssas.asp>>.
- [124] Sitio Mike 2.0. Disponível em: <http://mike2.openmethodology.org/wiki/Ontology_Tools>
- [125] Sitio Moodle/UFBA Taxonomia e a gestão do conhecimento explícito. Sitio de Educação a distância. EDC/20072 > Livros > Módulo II. 2007. Disponível em <<http://www.moodle.ufba.br/mod/book/view.php?id=13137&chapterid=10654>>.
- [126] Sitio Neon Toolkit. 2006. Disponível em: <http://neon-toolkit.org/wiki/Main_Page>
- [127] Sitio Ontology Matching. Disponível em: <<http://www.ontologymatching.org/projects.html> >
- [128] Sitio Oracle Oracle Olap. 2011. Disponível em <http://www.oracle.com/br/products/database/options/olap/index.html> Acesso em : Novembro de 2011.
- [129] Sitio Oracle Business Intelligence Suite Enterprise. Disponível em: <<http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html>>
- [130] Sitio Palo OLAP Server. 2011. Disponível em: <<http://www.jedox.com/pt/produos/palo-suite/palo-olap-server.html>>. Acessado em Novembro de 2011.
- [131] Sitio Pentaho. 2011. Open Source BI. Disponível em: <<http://www.pentaho.com/about/>>. Acesso em : Novembro de 2011.

- [132] Sitio Pentaho/Mondrian. 2011. Mondrian OLAP Server. Disponível em: <<http://mondrian.pentaho.org>>. Acesso em novembro 2011.
- [133] Sitio Protégé. 2005. Disponível em: <<http://protege.stanford.edu>>. Acesso em novembro de 2011.
- [134] Sitio S-Match. 2010. Disponível em: <<http://semanticmatching.org/s-match.html>>
- [135] Sitio SAP Business Object. Disponível em: <<http://www.sap.com/solutions/sapbusinessobjects/large/business-intelligence/reporting-analysis/analysis-olap/index.epx>>
- [136] Sitio Serpro. Artigo Pentaho é a nova ferramenta de BI do Serpro. Comunicação Social do Serpro - Brasília, 22 de março de 2010. Sitio do Serpro. Disponível em: <<https://www.serpro.gov.br/noticias/pentaho-e-a-nova-ferramenta-de-bi-do-serpro>>
- [137] Sitio XML Binary Characterization Working Group Public Page. 2004-2005. Disponível em <<http://www.w3.org/XML/Binary/>>
- [138] Sitio do OData.org. Sitio colaborativo para o OData. 2012. Disponível em <www.odata.org>
- [139] Sitio do Oracle JDBC. 2012. Disponível em: <<http://www.oracle.com/technetwork/java/overview-141217.html>>
- [140] Sitio do Protégé. 2012. Disponível em: <<http://protege.stanford.edu/>>
- [141] Sitio do W3Schools.com. SOAP Tutorial. 2012. Disponível em <<http://www.w3schools.com/soap/default.asp>>
- [142] Skonnard, A. Understanding SOAP. MSDN, Web Services Specifications, Technical Articles. Microsoft, March 2003. Disponível em <<http://msdn.microsoft.com/en-us/library/ms995800.aspx>>
- [143] Spago BI. (2011). Disponível em: <<http://www.spagoworld.org/xwiki/bin/view/SpagoBI/>>. Acessado em Novembro de 2011.
- [144] Staudt, D. L. (2008). Proposta de uma solução Business Intelligence com foco em Gerência de Projetos. Trabalho de Conclusão, Feevale, Novo Hamburgo, 2008.

- [145] Stodder, D. The State of Business Intelligence. Artigo publicado em 30/04/2008, no sitio da Intelligent Enterprise. Disponível em: <<http://www.intelligententerprise.com/showArticle.jhtml?articleID=198701576&pgno=1>>.
- [146] Su, X., Ilebrikke, L. A comparative study of ontology languages and tools. Advanced Information Systems Engineering, Lecture Notes in Computer Science, 2006, Volume 2348/2006, 761-765, DOI: 10.1007/3-540-47961-9_62, Springer. Disponível em: <<http://www.springerlink.com/content/0pgul6j9yhjdqrn/>>
- [147] Szwarcfiter, J. L. Grafos e algoritmos computacionais. Rio de Janeiro: Campus, 1988. pp. 43-45. ISBN 85-7001-341-8.
- [148] Uschold, M., Gruninger, M. Ontologies: Principles, Methods and Applications. AIAI-TR-191 February 1996. To appear in Knowledge Engineering Review Volume 11 Number 2, June 1996.
- [149] Uschold, M., Gruninger, M. Ontologies and semantics for seamless connectivity. 2004. SIGMOD Rec., 33(4):58-64.
- [150] Valtchev, P. Construction automatique de Taxonomies pour laide la representation de connaissances par objets. Ph.D. Dissertation, Universite Grenoble.
- [151] Valtchev, P., Euzenat, J. Dissimilarity measure for collections of objects and values, ser. Lecture Notes in Computer Science. London, UK: Springer, 1997, vol. 1280.
- [152] Vital, L. P.. Ontologias e Taxonomias: convergências e divergências. Universidade Federal de Pernambuco - XXIV Congresso Brasileiro de Biblioteconomia, Documentação e Ciência da Informação - Sistemas de Informação, Multiculturalidade e Inclusão Social - Maceió, Alagoas, 07 a 10 de Agosto de 2011.
- [153] W3C. Resource Description Framework (RDF) Model and Syntax Specification. 05 January 1999. Disponível em <<http://www.w3.org/TR/PR-rdf-syntax/>>
- [154] W3C. Web Ontology Language Guide: W3C Recommendation 10 February 2004. (2004-02-10). Disponível em: < <http://www.w3.org/2004/OWL/> >

- [155] W3C. Web Service Architecture. World-Wide Web Consortium - W3C. Disponível em: <www.w3.org/TR/ws-arch/>. 2004.
- [156] Woods, D., Mattern, T. Enterprise SOA. O'Reilly Media, Inc. 2006. ISBN-13: 978-0-596-10238-8.

ANEXOS

ANEXO I - ONTOLOGIA-MODELO "EDUCAÇÃO.OWL" UTILIZADA

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY daml "http://www.daml.org/2001/03/daml+oil#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://localhost/ontologies/ubi/Educacao#"
  xml:base="http://localhost/ontologies/ubi"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#">
  <owl:Ontology rdf:about="http://localhost/ontologies/ubi/Educacao"/>
  <!--
  // Classes
  -->

  <!-- http://localhost/ontologies/ubi#Ano -->
  <owl:Class rdf:about="http://localhost/ontologies/ubi#Ano">
    <rdfs:subClassOf rdf:resource="http://localhost/ontologies/ubi#Tempo"/>
  </owl:Class>

  <!-- http://localhost/ontologies/ubi#Geo -->
  <owl:Class rdf:about="http://localhost/ontologies/ubi#Geo"/>

  <!-- http://localhost/ontologies/ubi#Regiao -->
  <owl:Class rdf:about="http://localhost/ontologies/ubi#Regiao">
    <rdfs:subClassOf rdf:resource="http://localhost/ontologies/ubi#Geo"/>
  </owl:Class>

  <!-- http://localhost/ontologies/ubi#Tempo -->
  <owl:Class rdf:about="http://localhost/ontologies/ubi#Tempo"/>

  <!-- http://localhost/ontologies/ubi#UF -->
  <owl:Class rdf:about="http://localhost/ontologies/ubi#UF">
    <rdfs:subClassOf rdf:resource="http://localhost/ontologies/ubi#Regiao"/>
  </owl:Class>

  <!--
  // Individuals
  -->

  <!-- http://www.co-ode.org/ontologies/ont.owl#IDEB -->
  <owl:Thing rdf:about="http://www.co-ode.org/ontologies/ont.owl#IDEB">
    <rdf:type rdf:resource="&owl;NamedIndividual"/>
    <rdfs:isDefinedBy>http://cluster:cluster@datahost:8282
/kettle/executeTrans/?trans=/pdi-ce-4.3.0-stable/data-integration/
ubi/Educacao.ktr</rdfs:isDefinedBy>
  </owl:Thing>
</rdf:RDF>

<!-- Generated by the OWL API (version 3.3.1957)
http://owlapi.sourceforge.net -->
```

ANEXO II - PROVEDORES DADOS XML

```
<?xml version="1.0" encoding="UTF-8"?>
<transformation>
  <info>
    <name>Provedor Demografia</name>
    <description/>
    <extended_description/>
    <trans_version/>
    <trans_type>Normal</trans_type>
    <trans_status>0</trans_status>
    <directory>&#47;</directory>
    <parameters>
    </parameters>
    <log>
<trans-log-table>
<connection/>
<schema/>
<table/>
<size_limit_lines/>
<interval/>
<timeout_days/>
</trans-log-table>
<perf-log-table>
<connection/>
<schema/>
<table/>
<interval/>
<timeout_days/>
</perf-log-table>
<channel-log-table>
<connection/>
<schema/>
<table/>
<timeout_days/>
<step-log-table>
<connection/>
<schema/>
<table/>
<timeout_days/>
</trans-log-table>
<perf-log-table>
  </log>
  <maxdate>
    <connection/>
    <table/>
    <field/>
    <offset>0.0</offset>
    <maxdiff>0.0</maxdiff>
  </maxdate>
  <size_rowset>10000</size_rowset>
  <sleep_time_empty>50</sleep_time_empty>
  <sleep_time_full>50</sleep_time_full>
  <unique_connections>N</unique_connections>
  <feedback_shown>Y</feedback_shown>
  <feedback_size>50000</feedback_size>
  <using_thread_priorities>N</using_thread_priorities>
  <shared_objects_file/>
  <capture_step_performance>N</capture_step_performance>
  <step_performance_capturing_delay>1000</step_performance_capturing_delay>
  <step_performance_capturing_size_limit/>
  <dependencies>
</dependencies>
<partitionschemas>
```

```

    </partitionschemas>
    <slaveservers />
    <clusterschemas>
    </clusterschemas>
<created_user/>
<created_date>2012&#47;06&#47;21 07:53:50.986</created_date>
<modified_user>-</modified_user>
<modified_date>2007&#47;11&#47;16 20:11:28.571</modified_date>
</info>
<connection>
  <name>demografia</name>
  <server>localhost</server>
  <type>POSTGRESQL</type>
  <access>Native</access>
  <database>demografia</database>
  <port>5432</port>
  <username>postgres</username>
  <password>Encrypted 2be98afc86aa7f2e4bb16bd64d980aac9</password>
  <servername/>
  <data_tablespace>postgres</data_tablespace>
  <index_tablespace>postgres</index_tablespace>
  <attributes>
    <attribute><code>PORT_NUMBER</code><attribute>5432</attribute></attribute>
  </attributes>
</connection>
<order>
<hop> <from>Table input</from><to>XML Output</to><enabled>Y</enabled> </hop>
</order>
<step>
  <name>Table input</name>
  <type>TableInput</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>
    <schema_name/>
  </partitioning>
  <connection>demografia</connection>
  <sql>SELECT pk demografia, fk tempo, fk geo, qt populacao, qt area, vl idh
  FROM fat demografia</sql>
  <limit>0</limit>
  <lookup/>
  <execute_each_row>N</execute_each_row>
  <variables_active>N</variables_active>
  <lazy_conversion_active>N</lazy_conversion_active>
  <cluster_schema/>
  <remotesteps>
    <input></input>
    <output></output>
  </remotesteps>
  <GUI>
    <xloc>38</xloc>
    <yloc>29</yloc>
    <draw>Y</draw>
  </GUI>
</step>
<step>
  <name>XML Output</name>
  <type>XMLOutput</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>
    <schema_name/>
  </partitioning>

```

```

<encoding>UTF-8</encoding>
<name_space/>
<xml_main_element>t</xml_main_element>
<xml_repeat_element>r</xml_repeat_element>
<file>
  <name>${Internal.Transformation.FileName.Directory}&#47;output&#47;
    ${Internal.Transformation.Name}</name>
  <extention>xml</extention>
  <servlet_output>Y</servlet_output>
  <do_not_open_newfile_init>N</do_not_open_newfile_init>
  <split>N</split>
  <add_date>N</add_date>
  <add_time>N</add_time>
  <SpecifyFormat>N</SpecifyFormat>
  <omit_null_values>N</omit_null_values>
  <date_time_format/>
  <add_to_result_filenames>N</add_to_result_filenames>
  <zipped>N</zipped>
  <splitevery>0</splitevery>
</file>
<fields>
  <field>
    <name>pk_demografia</name>
    <element/>
    <type>Integer</type>
    <format/>
    <currency/>
    <decimal/>
    <group/>
    <>nullif/>
    <length>-1</length>
    <precision>-1</precision>
  </field>
  <field>
    <name>fk_tempo</name>
    <element/>
    <type>Integer</type>
    <format>#####</format>
    <currency/>
    <decimal/>
    <group/>
    <>nullif/>
    <length>-1</length>
    <precision>-1</precision>
  </field>
  <field>
    <name>fk_geo</name>
    <element/>
    <type>Integer</type>
    <format>#####</format>
    <currency/>
    <decimal/>
    <group/>
    <>nullif/>
    <length>-1</length>
    <precision>-1</precision>
  </field>
  <field>
    <name>qt_populacao</name>
    <element/>
    <type>BigNumber</type>
    <format>#####</format>
    <currency/>
    <decimal/>
    <group/>
    <>nullif/>
    <length>-1</length>
    <precision>-1</precision>
  </field>

```

```

</field>
<field>
  <name>qt_area</name>
  <element/>
  <type>BigNumber</type>
  <format>#####.##</format>
  <currency/>
  <decimal/>
  <group/>
  <nullif/>
  <length>-1</length>
  <precision>-1</precision>
</field>
<field>
  <name>vl_idh</name>
  <element/>
  <type>BigNumber</type>
  <format>###.##</format>
  <currency/>
  <decimal/>
  <group/>
  <nullif/>
  <length>-1</length>
  <precision>-1</precision>
</field>
</fields>
<cluster_schema/>
<remotesteps>
  <input></input>
  <output></output>
</remotesteps>
<GUI>
  <xloc>256</xloc>
  <yloc>29</yloc>
  <draw>Y</draw>
</GUI>
</step>
<step_error_handling>
</step_error_handling>
<slave-step-copy-partition-distribution>
</slave-step-copy-partition-distribution>
  <slave_transformation>N</slave_transformation>
</transformation>

```

ANEXO III - CONSUMIDORES DE DADOS XML

```
<?xml version="1.0" encoding="UTF-8"?>
<transformation>
  <info>
    <name>Consumidor_Demografia</name>
    <description/>
    <extended_description/>
    <trans_version/>
    <trans_type>Normal</trans_type>
    <trans_status>0</trans_status>
    <directory>&#47;</directory>
    <parameters>
    </parameters>
    <log>
<trans-log-table><connection/>
<schema/>
<table/>
<size_limit_lines/>
<interval/>
<timeout_days/>
<perf-log-table><connection/>
<schema/>
<table/>
<interval/>
<timeout_days/>
</perf-log-table>
<channel-log-table><connection/>
<schema/>
<table/>
<timeout_days/>
</channel-log-table>
<step-log-table><connection/>
<schema/>
<table/>
<timeout_days/>
</step-log-table>
    </log>
    <maxdate>
      <connection/>
      <table/>
      <field/>
      <offset>0.0</offset>
      <maxdiff>0.0</maxdiff>
    </maxdate>
    <size_rowset>10000</size_rowset>
    <sleep_time_empty>50</sleep_time_empty>
    <sleep_time_full>50</sleep_time_full>
    <unique_connections>N</unique_connections>
    <feedback_shown>Y</feedback_shown>
    <feedback_size>50000</feedback_size>
    <using_thread_priorities>N</using_thread_priorities>
    <shared_objects_file/>
    <capture_step_performance>N</capture_step_performance>
    <step_performance_capturing_delay>1000</step_performance_capturing_delay>
    <step_performance_capturing_size_limit/>
    <dependencies>
    </dependencies>
    <partitionschemas>
    </partitionschemas>
    <slaveservers/ >
    <clusterschemas>
    </clusterschemas>
    <created_user/>
    <created_date>2012&#47;06&#47;21 07:55:10.466</created_date>
    <modified_user>-</modified_user>
    <modified_date>2007&#47;11&#47;16 20:11:28.571</modified_date>
```

```

</info>
<connection>
  <name>ubi</name>
  <server>localhost</server>
  <type>HYPERSONIC</type>
  <access>Native</access>
  <database>ubi</database>
  <port>9001</port>
  <username>pentaho_user</username>
  <password>Encrypted 2be98afc86aa7f2e4bb18bd63c99dbdde</password>
  <servername/>
  <data_tablespace/>
  <index_tablespace/>
  <attributes>
    <attribute>
      <code>FORCE_IDENTIFIERS_TO_LOWERCASE</code>
      <attribute>N</attribute>
    </attribute>
    <attribute>
      <code>FORCE_IDENTIFIERS_TO_UPPERCASE</code>
      <attribute>N</attribute>
    </attribute>
    <attribute>
      <code>IS_CLUSTERED</code>
      <attribute>N</attribute>
    </attribute>
    <attribute>
      <code>PORT_NUMBER</code>
      <attribute>9001</attribute>
    </attribute>
    <attribute>
      <code>QUOTE_ALL_FIELDS</code>
      <attribute>N</attribute>
    </attribute>
    <attribute>
      <code>SUPPORTS_BOOLEAN_DATA_TYPE</code>
      <attribute>N</attribute>
    </attribute>
    <attribute>
      <code>USE_POOLING</code>
      <attribute>N</attribute>
    </attribute>
  </attributes>
</connection>
<order>
  <hop><from>02 HTTP client</from>
  <to>03 Get data from XML</to>
  <enabled>Y</enabled></hop>
  <hop><from>03 Get data from XML</from>
  <to>04 Select values</to>
  <enabled>Y</enabled></hop>
  <hop><from>01 Trigger</from>
  <to>02 HTTP client</to>
  <enabled>Y</enabled></hop>
  <hop><from>04 Select values</from>
  <to>05 Table output</to>
  <enabled>Y</enabled></hop>
</order>
<step>
  <name>01 Trigger</name>
  <type>ExecSQL</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>
    <schema_name/>
  </partitioning>

```



```

<connection>ubi</connection>
<execute_each_row>N</execute_each_row>
<single_statement>Y</single_statement>
<replace_variables>N</replace_variables>
<sql>DELETE FROM PUBLIC.FAT DEMOGRAFIA;</sql>
<insert_field/>
<update_field/>
<delete_field/>
<read_field/>
<arguments>
</arguments>
<cluster_schema/>
<remotesteps>
  <input> </input>
  <output> </output>
</remotesteps>
<GUI>
  <xloc>45</xloc>
  <yloc>30</yloc>
  <draw>Y</draw>
</GUI>
</step>
<step>
  <name>02 HTTP client</name>
  <type>HTTP</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>
    <schema_name/>
  </partitioning>
  <url>http://cluster:cluster@localhost:8282/kettle/executeTrans/? trans=%2fpdi-ce-4.3.0-stable%2fdata-integration%2fubi%2fProvedor_Demografia.ktr</url>
  <urlInField>N</urlInField>
  <urlField/>
  <encoding>UTF-8</encoding>
  <httpLogin>cluster</httpLogin>
  <httpPassword>cluster</httpPassword>
  <proxyHost/>
  <proxyPort/>
  <lookup>
  </lookup>
  <result>
    <name>result</name>
    <code>status</code>
    <response_time>ts</response_time>
  </result>
  <cluster_schema/>
  <remotesteps>
    <input> </input>
    <output> </output>
  </remotesteps>
  <GUI>
    <xloc>168</xloc>
    <yloc>97</yloc>
    <draw>Y</draw>
  </GUI>
</step>
<step>
  <name>03 Get data from XML</name>
  <type>getXMLData</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>

```

```

        <schema_name/>
        </partitioning>
<include>N</include>
<include_field/>
<rownum>N</rownum>
<addressresultfile>N</addressresultfile>
<namespaceaware>N</namespaceaware>
<ignorecomments>N</ignorecomments>
<readurl>N</readurl>
<validating>N</validating>
<usetoken>N</usetoken>
<IsIgnoreEmptyFile>N</IsIgnoreEmptyFile>
<doNotFailIfNoFile>Y</doNotFailIfNoFile>
<rownum_field/>
<encoding>UTF-8</encoding>
<file>
  <name/>
  <filemask/>
  <exclude_filemask/>
  <file_required>N</file_required>
  <include_subfolders>N</include_subfolders>
</file>
<fields>
  <field>
    <name>pk_demografia</name>
    <xpath>pk_demografia</xpath>
    <element_type>node</element_type>
    <result_type>valueof</result_type>
    <type>Number</type>
    <format/>
    <currency/>
    <decimal/>
    <group/>
    <length>-1</length>
    <precision>-1</precision>
    <trim_type>both</trim_type>
    <repeat>N</repeat>
  </field>
  <field>
    <name>fk_tempo</name>
    <xpath>fk_tempo</xpath>
    <element_type>node</element_type>
    <result_type>valueof</result_type>
    <type>Integer</type>
    <format/>
    <currency/>
    <decimal/>
    <group/>
    <length>-1</length>
    <precision>-1</precision>
    <trim_type>both</trim_type>
    <repeat>N</repeat>
  </field>
  <field>
    <name>fk_geo</name>
    <xpath>fk_geo</xpath>
    <element_type>node</element_type>
    <result_type>valueof</result_type>
    <type>Integer</type>
    <format/>
    <currency/>
    <decimal/>
    <group/>
    <length>-1</length>
    <precision>-1</precision>
    <trim_type>both</trim_type>
    <repeat>N</repeat>
  </field>
</fields>

```

```

<field>
  <name>qt_populacao</name>
  <xpath>qt_populacao</xpath>
  <element_type>node</element_type>
  <result_type>valueof</result_type>
  <type>Integer</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <length>-1</length>
  <precision>-1</precision>
  <trim_type>both</trim_type>
  <repeat>N</repeat>
</field>
<field>
  <name>qt_area</name>
  <xpath>qt_area</xpath>
  <element_type>node</element_type>
  <result_type>valueof</result_type>
  <type>String</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <length>-1</length>
  <precision>-1</precision>
  <trim_type>both</trim_type>
  <repeat>N</repeat>
</field>
<field>
  <name>vl_idh</name>
  <xpath>vl_idh</xpath>
  <element_type>node</element_type>
  <result_type>valueof</result_type>
  <type>String</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <length>-1</length>
  <precision>-1</precision>
  <trim_type>both</trim_type>
  <repeat>N</repeat>
</field>
</fields>
<limit>0</limit>
<loopxpath>&#47;t&#47;r</loopxpath>
<IsInFields>Y</IsInFields>
<IsAFile>N</IsAFile>
<XmlField>result</XmlField>
<prunePath/>
<shortFileName/>
<pathFieldName/>
<hiddenFieldName/>
<lastModificationTimeFieldName/>
<uriNameFieldName/>
<rootUriNameFieldName/>
<extensionFieldName/>
<sizeFieldName/>
<cluster_schema/>
<remotesteps>
  <input> </input>
  <output> </output>
</remotesteps>
<GUI>
  <xloc>348</xloc>
  <yloc>36</yloc>

```

```

    <draw>Y</draw>
  </GUI>
</step>
<step>
  <name>04 Select values</name>
  <type>SelectValues</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
    <partitioning>
      <method>none</method>
      <schema_name/>
    </partitioning>
  <fields>
    <select_unspecified>N</select_unspecified>
    <remove><name>result</name></remove>
    <remove><name>status</name></remove>
    <remove><name>ts</name></remove>
    <meta>
      <name>pk_demografia</name>
      <rename>pk_demografia</rename>
      <type>Integer</type>
      <length>8</length>
      <precision>0</precision>
      <conversion_mask/>
      <date_format_lenient>>false</date_format_lenient>
      <encoding/>
      <decimal_symbol>0</decimal_symbol>
      <grouping_symbol/>
      <currency_symbol/>
      <storage_type/>
    </meta>
    <meta>
      <name>fk_tempo</name>
      <rename>fk_tempo</rename>
      <type>Integer</type>
      <length>6</length>
      <precision>0</precision>
      <conversion_mask/>
      <date_format_lenient>>false</date_format_lenient>
      <encoding/>
      <decimal_symbol>0</decimal_symbol>
      <grouping_symbol/>
      <currency_symbol/>
      <storage_type/>
    </meta>
    <meta>
      <name>fk_geo</name>
      <rename>fk_geo</rename>
      <type>Integer</type>
      <length>7</length>
      <precision>0</precision>
      <conversion_mask/>
      <date_format_lenient>>false</date_format_lenient>
      <encoding/>
      <decimal_symbol>0</decimal_symbol>
      <grouping_symbol/>
      <currency_symbol/>
      <storage_type/>
    </meta>
    <meta>
      <name>qt_populacao</name>
      <rename>qt_populacao</rename>
      <type>Number</type>
      <length>17</length>
      <precision>2</precision>
      <conversion_mask/>
      <date format lenient>>false</date format lenient>

```

```

    <encoding/>
    <decimal_symbol>2</decimal_symbol>
    <grouping_symbol/>
    <currency_symbol/>
    <storage_type/>
  </meta>
  <meta>
    <name>qt_area</name>
    <rename>qt_area</rename>
    <type>Number</type>
    <length>17</length>
    <precision>2</precision>
    <conversion_mask/>
    <date_format_lenient>>false</date_format_lenient>
    <encoding/>
    <decimal_symbol>2</decimal_symbol>
    <grouping_symbol/>
    <currency_symbol/>
    <storage_type/>
  </meta>
  <meta>
    <name>vl_idh</name>
    <rename>vl_idh</rename>
    <type>Number</type>
    <length>17</length>
    <precision>2</precision>
    <conversion_mask/>
    <date_format_lenient>>false</date_format_lenient>
    <encoding/>
    <decimal_symbol>2</decimal_symbol>
    <grouping_symbol/>
    <currency_symbol/>
    <storage_type/>
  </meta>
</fields>
<cluster_schema/>
<remotesteps>
  <input> </input>
  <output> </output>
</remotesteps>
<GUI>
  <xloc>479</xloc>
  <yloc>36</yloc>
  <draw>Y</draw>
</GUI>
</step>
<step>
  <name>05 Table output</name>
  <type>TableOutput</type>
  <description/>
  <distributed>Y</distributed>
  <copies>1</copies>
  <partitioning>
    <method>none</method>
    <schema_name/>
  </partitioning>
  <connection>ubi</connection>
  <schema/>
  <table>FAT_DEMOGRAFIA</table>
  <commit>1000</commit>
  <truncate>Y</truncate>
  <ignore_errors>N</ignore_errors>
  <use_batch>N</use_batch>
  <specify_fields>Y</specify_fields>
  <partitioning_enabled>N</partitioning_enabled>
  <partitioning_field/>
  <partitioning_daily>N</partitioning_daily>
  <partitioning_monthly>Y</partitioning_monthly>

```

```

<tablename_in_field>N</tablename_in_field>
<tablename_field/>
<tablename_in_table>N</tablename_in_table>
<return_keys>N</return_keys>
<return_field/>
<fields>
  <field>
    <column_name>pk_demografia</column_name>
    <stream_name>pk_demografia</stream_name>
  </field>
  <field>
    <column_name>fk_tempo</column_name>
    <stream_name>fk_tempo</stream_name>
  </field>
  <field>
    <column_name>fk_geo</column_name>
    <stream_name>fk_geo</stream_name>
  </field>
  <field>
    <column_name>qt_populacao</column_name>
    <stream_name>qt_populacao</stream_name>
  </field>
  <field>
    <column_name>qt_area</column_name>
    <stream_name>qt_area</stream_name>
  </field>
  <field>
    <column_name>vl_idh</column_name>
    <stream_name>vl_idh</stream_name>
  </field>
</fields>
<cluster_schema/>
<remotesteps>
  <input> </input>
  <output> </output>
</remotesteps>
<GUI>
  <xloc>621</xloc>
  <yloc>37</yloc>
  <draw>Y</draw>
</GUI>
</step>
<step_error_handling>
</step_error_handling>
<slave-step-copy-partition-distribution>
</slave-step-copy-partition-distribution>
<slave_transformation>N</slave_transformation>
</transformation>

```

ANEXO IV - CÓDIGO-FONTE DO PROCESSO "SITE UBI ... FUSÃO"

```
<?php

$rota_site = "C:/xampp/htdocs/UBI/";
$rota_java = "C:/jdk1.6.0_25/bin/";
$rota_pdi = "C:/pdi-ce-4.3.0-stable/data-integration/";
$rota_mondrian = "C:/biserver-ce-4.5.0-stable/biserver-ce/pentaho-
solutions/ubi/analysis/";
$hsqldblib = "C:/biserver-ce-4.5.0-stable/biserver-ce/tomcat/lib/hsqldb-
1.8.0.7.jar";
$server_ip = "127.0.0.1";

error_reporting(0);
set_time_limit(0);
session_start();

require("dirLIST_files/config.php");
require("dirLIST_files/functions.php");

$url_folder = base64_decode(trim($_GET['folder']));
if(!empty($_GET['folder']))
    $dir_to_browse .= $url_folder."/";

if(!empty($_REQUEST['ontologia_dominio'])) {
    $ontologia_dominio = $_REQUEST['ontologia_dominio'];
}

$ubi = OWL2Array($dir_to_browse."Ontologia_Global/UBI.owl");
$aux = array_copy($ubi);

$dominios_validos = array();
$dir_content = get_dir_content($dir_to_browse."Ontologias_de_Dominio/");
$files = array();
$files['name'] = $dir_content['files']['name'];
$files['attr'] = $dir_content['files']['attr'];
$files['link'] = $dir_content['files']['link'];
foreach($files['name'] as $key => $val) {
    $ext = explode(".", $files['link'][$key]);
    $is_banned = in_array(".$ext[count($ext)-1], $banned_file_types);
    if (($files['attr'][$key]==1) && (!$is_banned)) {
        array_push($dominios_validos, $files['name'][$key]);
    }
}
?>
<!-- Output basic HTML code -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>dirLIST - <?php if(empty($url_folder)) echo "Index of: /"; else echo
"Index of: */".$url_folder."/"; ?></title>
<style type="text/css">
<!--
<?php echo 'body,td,th {font-family: Tahoma, Verdana;font-size: 10pt;}
a:link {text-decoration: none;color: '.$color_scheme['link_content']['link'].';}
a:visited {text-decoration: none;color:
'.$color_scheme['link_content']['visited'].';}
a:hover {text-decoration: underline;color:
'.$color_scheme['link_content']['hover'].';}
a:active {text-decoration: none;color:
'.$color_scheme['link_content']['active'].';}
a.sort:link{text-decoration: underline;color:
'.$color_scheme['link_sort']['link'].';}
a.sort:visited{text-decoration: underline;color:
```

```

'. $color_scheme['link_sort']['visited'].';}
a.sort:hover{text-decoration: underline;color:
'. $color_scheme['link_sort']['hover'].';}
a.sort:active{text-decoration: none;color:
'. $color_scheme['link_sort']['active'].';}
.top_row {color: '. $color_scheme['top_row']['color'].';font-weight: bold;font-
size: 14px;background-color: '. $color_scheme['top_row']['bg'].';}
.folder_bg {background-color: '. $color_scheme['main_table']['folder_bg'].';}
.file_bg1 {background-color: '. $color_scheme['main_table']['file_bg1'].';}
.file_bg2 {background-color: '. $color_scheme['main_table']['file_bg2'].';}
.file_bg3 {background-color: '. $color_scheme['main_table']['file_bg3'].';}';}
?>
.table_border {border: 1px dashed #666666;}
.path_font {font-family: "Courier New", Courier, monospace;}
.banned font {font-size: 9px;}
.error {border-top-width: 2px;border-bottom-width: 2px;border-top-style:
solid;border-bottom-style: solid;border-top-color: #FF666A;border-bottom-color:
#FF666A;}
#color_scheme {cursor:pointer;}
.option_style {font-family: Verdana, Tahoma;font-size: 11px;}
.language_selection {height: 22px; width: 182px;background-color:<?php echo
$color_scheme['main_table']['file_bg1']; ?>; border: 1px dashed #666666;}
.selected_lang {background-color:<?php echo
$color_scheme['main_table']['file_bg2']; ?>;}
#file_edit_box {position:absolute;width: 150px;display:none;}
-->
</style>
<?php
print "<html>";
print "<body>";
print "===== INICIO =====<br>";

// ---- Gerar arquivo Mondrian.xml
// ----- em: <pentaho_solutions>/ubi/analysis/UBI.mondrian.xml
flush();
print "<h3>Gerando arquivo Mondrian UBI.mondrian.xml</h3><br>";
$str_mondrian='';
$str_mondrian.='<?xml version="1.0"?>'.chr(13).chr(10);
$str_mondrian.='<Schema name="ubi">'.chr(13).chr(10);
$str_mondrian.=''.chr(13).chr(10);
$dimensoes_unicas = array();
$valores_unicos = array();
reset($ubi);
foreach ($ubi as $adata) {
    foreach ($adata as $atyp => $aline) {
        if ($atyp=="01_HIERARQUIA") {
            $nome_dimensao = $aline['CLASSE'];
            $nome_dimensao_upper = strtoupper($nome_dimensao);
            print "Dimensão: ".$nome_dimensao."<br>";
            $str_mondrian.=' <Dimension name="'.$nome_dimensao.'">'.chr(13).chr(10);
            $str_mondrian.=' <Hierarchy hasAll="true" allMemberName="Total
'. $nome_dimensao.'" primaryKey="PK_'. $nome_dimensao_upper.'">'.chr(13).chr(10);
            $str_mondrian.=' <Table
name="DIM_'. $nome_dimensao_upper.'">'.chr(13).chr(10);
            $niveis_unicos = Hierarquia2ArrayNiveis($ubi, $nome_dimensao_upper, true);
            reset($niveis_unicos);
            $cnt = count($niveis_unicos);
            $i = 1;
            foreach ($niveis_unicos as $nivel_desejado) {
                reset($ubi);

```



```

        foreach ($ubi as $adata1) {
            foreach ($adata1 as $atypel => $aline1) {
                if (($atypel=="02_NIVEL") &&
                (strtoupper($aline1['CLASSE'])==$nivel_desejado)) {
                    $nome_nivel = $aline1['CLASSE'];
                    $nome_nivel_upper = strtoupper($nome_nivel);
                    if ($cnt == $i) {
                        $str_mondrian.='        <Level name="' . $nome_nivel . '"
column="DS_' . $nome_nivel_upper . '" uniqueMembers="true"/>' . chr(13) . chr(10);
                    }
                    else {
                        $str_mondrian.='        <Level name="' . $nome_nivel . '"
column="DS_' . $nome_nivel_upper . '" uniqueMembers="false"/>' . chr(13) . chr(10);
                    }
                }
            }
            $i++;
        }
        array_push($dimensoes_unicas, array($nome_dimensao_upper =>
$ niveis_unicos));
        $str_mondrian.='    </Hierarchy>' . chr(13) . chr(10);
        $str_mondrian.=' </Dimension>' . chr(13) . chr(10);
        $str_mondrian.=' ' . chr(13) . chr(10);
    }
    if ($atypel=="03_VALORES") {
        $nome_dimensao = $aline1['SUBCLASSE'];
        $nome_dimensao_upper = strtoupper($nome_dimensao);
        $nome_instancia = $aline1['CLASSE'];
        $nome_instancia_upper = strtoupper($nome_instancia);
        $valor_instancia = $aline1['DEFINICAO'];
        if ($nome_instancia_upper != "THING") {
            array_push($valores_unicos, array($nome_dimensao_upper =>
array($nome_instancia_upper, $valor_instancia)));
        }
    }
}
}

$fatos = array();
$dimensoes = array();
$medidas = array();
$valores = array();
foreach ($dominios_validos as $key => $owl_file) {
    $cubos = OWL2Array($dir_to_browse . "Ontologias_de_Dominio/" . $owl_file);
    $aux = array_copy($cubos);
    $ary_dominio = explode("/", $owl_file);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $dominio_upper = strtoupper($dominio);
    array_push($fatos, $dominio_upper);
    print "Cubo: " . $dominio_upper . "<br>";
    $str_mondrian.='    <Cube name="' . $dominio_upper . '">' . chr(13) . chr(10);
    $str_mondrian.='        <Table name="FAT_' . $dominio_upper . '" />' . chr(13) . chr(10);
    reset($cubos);
    foreach ($cubos as $adata) {
        foreach ($adata as $atypel => $aline) {
            if ($atypel=="01_HIERARQUIA") {
                $nome_dimensao = $aline['CLASSE'];
                $nome_dimensao_upper = strtoupper($nome_dimensao);
                print "-- Dimensão: " . $nome_dimensao . "<br>";
                $str_mondrian.='        <DimensionUsage name="' . $nome_dimensao . '"
source="' . $nome_dimensao . '" foreignKey="FK_' . $nome_dimensao_upper . '"
/>' . chr(13) . chr(10);
                $niveis = array();
            }
        }
    }
}

```

```

        foreach ($aux as $adata1) {
            foreach ($adata1 as $atypel => $aline1) {
                if ($atypel=="02_NIVEL") {
                    if (strtoupper($aline1['SUBCLASSE'])==$nivel_anterior_upper) {
                        $nome_nivel_upper = strtoupper($nome_nivel);
                        array_push($niveis, array($nome_nivel_upper =>
$nivel_anterior_upper));
                    }
                }
            }
        }
        array_push($dimensoes, array($dominio_upper =>
array($nome_dimensao_upper => $niveis)));
    }
}
reset($cubos);
foreach ($cubos as $adata) {
    foreach ($adata as $atypel => $aline) {
        if ($atypel=="04_MEDIDA") {
            $nome_medida = $aline['CLASSE'];
            $nome_medida_upper = strtoupper($nome_medida);
            $eh_definido_em = $aline['DEFINICAO'];
            $str_mondrian.=' <Measure name="'.str_replace(" ", " ",
$nome_medida).' " column="VL_'.$nome_medida_upper.'" aggregator="sum"
formatString="#.00"/>' .chr(13).chr(10);
            print "-- Medida: ".$nome_medida ."<br>";
            array_push($medidas, array($dominio_upper => array($nome_medida =>
$eh_definido_em)));
        }
    }
}
reset($cubos);
foreach ($cubos as $adata) {
    foreach ($adata as $atypel => $aline) {
        if ($atypel=="03_VALORES") {
            $nome_dimensao = $aline['SUBCLASSE'];
            $nome_dimensao_upper = strtoupper($nome_dimensao);
            $nome_instancia = $aline['CLASSE'];
            $nome_instancia_upper = strtoupper($nome_instancia);
            $valor_instancia = $aline['DEFINICAO'];
            array_push($valores, array($dominio_upper => array($nome_dimensao_upper
=> array($nome_instancia_upper => $valor_instancia))));
        }
    }
}
$str_mondrian.=' </Cube>' .chr(13).chr(10);
$str_mondrian.=' ' .chr(13).chr(10);
}

print "Cubo Virtual: UBI<br>";
$str_mondrian.=' <VirtualCube name="UBI">' .chr(13).chr(10);
$str_mondrian.=' <CubeUsages>' .chr(13).chr(10);
foreach ($dominios_validos as $key => $owl_file) {
    $ary_dominio = explode("/", $owl_file);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $dominio_upper = strtoupper($dominio);
    $str_mondrian.=' <CubeUsage cubeName="'.$dominio_upper.'"
ignoreUnrelatedDimensions="true"/>' .chr(13).chr(10);
    print "-- CubeUsage: ".$dominio_upper ."<br>";
}
}

```

```

$str_mondrian.=' </CubeUsages>'.chr(13).chr(10);
reset($ubi);
foreach ($ubi as $adata) {
    foreach ($adata as $atype => $aline) {
        if ($atype=="01_HIERARQUIA") {
            $nome_dimensao = $aline['CLASSE'];
            $nome_dimensao_upper = strtoupper($nome_dimensao);
            $str_mondrian.=' <VirtualCubeDimension
name="'. $nome_dimensao.'" />'.chr(13).chr(10);
            print "-- VirtualDimension: ". $nome_dimensao."<br>";
        }
    }
}
reset($medidas);
foreach ($medidas as $adata) {
    foreach ($adata as $key => $value) {
        $nome_medida = $aline['CLASSE'];
        $nome_medida_upper = strtoupper($nome_medida);
        print "-- VirtualCubeMeasure: [". $value." ] ". $key."<br>";
        foreach ($value as $nome_medida => $eh_definido_por) {
            $str_mondrian.=' <VirtualCubeMeasure cubeName="'. $key.'"
name="[Measures].['.str_replace("_", " ", $nome_medida).']" />'.chr(13).chr(10);
        }
    }
}

$str_mondrian.=' </VirtualCube>'.chr(13).chr(10);
$str_mondrian.=' </Schema>'.chr(13).chr(10);
$handle = @fopen($rota_mondrian."UBI.mondrian.xml", "w+");
fputs($handle, $str_mondrian);
fclose($handle);
print "<br>&nbsp;<br>";

// ---- Gerar Mondrian ValidMeasure
// ----- em: em: <pentaho_solutions>/ubi/analysis/UBI.xaction
flush();
print "<h3>Gerando Mondrian ValidMeasure</h3><br>";
print "<br>&nbsp;<br>";

// ---- Atualizar arquivos HSQLDB
// ----- criar script no site em Scripts_HSQLDB
// ----- executar todos os script com java -jar $HSQLDB_HOME/lib/hsqldb.jar
ubi filepath1.sql...
flush();
print "<h3>Atualizando arquivos HLQLDB</h3><br>";

$hsqldb_files = array();
array_push($hsqldb_files, "Scripts_HSQLDB/DIM_TEMPO.sql");
array_push($hsqldb_files, "Scripts_HSQLDB/DIM_GEO.sql");
reset($dimensoes_unicas);
foreach ($dimensoes_unicas as $dimensoes_array) {
    foreach ($dimensoes_array as $dimensao => $dimensao_data) {
        if (($dimensao != "TEMPO") && ($dimensao != "GEO")) {
            $hsqldb_string = "";
            array_push($hsqldb_files, "Scripts_HSQLDB/DIM_". $dimensao." .sql");
            print "Scripts_HSQLDB/DIM_". $dimensao." .sql<br>";
            $handle = @fopen($rota_site."Scripts_HSQLDB/DIM_". $dimensao." .sql", "w+");
            $hsqldb_string.= "\c true".chr(13).chr(10);
            $hsqldb_string.= "DROP TABLE PUBLIC.DIM_". $dimensao." ;".chr(13).chr(10);
            $hsqldb_string.= "\c false".chr(13).chr(10);
            $hsqldb_string.= "CREATE MEMORY TABLE DIM_". $dimensao." (PK_". $dimensao."
INTEGER NOT NULL PRIMARY KEY, ";
            $nome_dimensao_upper = strtoupper(dimensao);
            $niveis_unicos = Hierarquia2ArrayNiveis($ubi, $dimensao, false);

```

```

        reset($niveis_unicos);
        $cnt = count($niveis_unicos);
        $i = 1;
        foreach ($niveis_unicos as $nivel_desejado) {
            reset($ubi);
            foreach ($ubi as $adata1) {
                foreach ($adata1 as $atyp1 => $aline1) {
                    if (($atyp1=="02_NIVEL") &&
(strtoupper($aline1['CLASSE'])==$nivel_desejado)) {
                        $nome_nivel = $aline1['CLASSE'];
                        $nome_nivel_upper = strtoupper($nome_nivel);
                        if ($cnt == $i) {
                            $hsqldb_string.= "DS_".$nome_nivel_upper."
VARCHAR(255));".chr(13).chr(10);
                        }
                        else {
                            $hsqldb_string.= "DS_".$nome_nivel_upper." VARCHAR(255)";
                        }
                    }
                }
            }
            $i++;
        }
        $hsqldb_string.= "GRANT ALL ON PUBLIC.DIM_".$dimensao." TO
PENTAHO_USER;".chr(13).chr(10);
        foreach ($dimensao_data as $indice => $hierarquia_nome) {
        }
        foreach ($valores_unicos as $indice => $valores_lista) {
            foreach ($valores_lista as $nivel_upper => $array_valor) {
                if ($nivel_upper == $hierarquia_nome) {
                    $hsqldb_string.= "INSERT INTO DIM_".$dimensao." VALUES
(".$array_valor[1].",".$array_valor[0].");".chr(13).chr(10);
                }
            }
        }
        $hsqldb_string.= "commit;".chr(13).chr(10);
        fputs($handle, $hsqldb_string);
        fclose($handle);
    }
}

reset($dominios_validos);
foreach ($dominios_validos as $key => $owl_file) {
    $hsqldb_string = "";
    $ary_dominio = explode("/", $owl_file);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $dominio_upper = strtoupper($dominio);
    array_push($hsqldb_files, "Scripts_HSQLDB/FAT_".$dominio_upper.".sql");
    print "Scripts_HSQLDB/FAT_".$dominio_upper.".sql<br>";
    $handle = @fopen($rota_site."Scripts_HSQLDB/FAT_".$dominio_upper.".sql",
"w+");
    $hsqldb_string.= "\c true".chr(13).chr(10);
    $hsqldb_string.= "DROP TABLE PUBLIC.FAT_".$dominio_upper.";".chr(13).chr(10);
    $hsqldb_string.= "\c false".chr(13).chr(10);
    $hsqldb_string.= "CREATE MEMORY TABLE
FAT_".$dominio_upper."(PK_".$dominio_upper." INTEGER NOT NULL PRIMARY KEY, ";
    $cubos = OWL2Array($dir_to_browse."Ontologias_de_Dominio/"."$owl_file);
    reset($cubos);
    foreach ($cubos as $adata) {
        foreach ($adata as $atyp => $aline) {
            if ($atyp=="01_HIERARQUIA") {
                $nome_dimensao = $aline['CLASSE'];
                $nome_dimensao_upper = strtoupper($nome_dimensao);
                $hsqldb_string.= "FK_".$nome_dimensao_upper." INTEGER, ";
            }
        }
    }
}

```

```

    }
  }
  reset($cubos);
  foreach ($cubos as $adata) {
    foreach ($adata as $atipe => $aline) {
      if ($atipe=="04_MEDIDA") {
        $nome_medida = $aline['CLASSE'];
        $nome_medida_upper = strtoupper($nome_medida);
        $hsqldb_string.= "VL_". $nome_medida_upper. " NUMERIC(17,2), ";
      }
    }
  }
  $hsqldb_string = trim($hsqldb_string);
  $hsqldb_string = substr($hsqldb_string, 0, strlen($hsqldb_string)-1);
  $hsqldb_string.= " );".chr(13).chr(10);
  $hsqldb_string.= "GRANT ALL ON PUBLIC.FAT_". $dominio_upper. " TO
PENTAHO_USER;".chr(13).chr(10);
  $hsqldb_string.= "commit;".chr(13).chr(10);
  fputs($handle, $hsqldb_string);
  fclose($handle);
}
print "<br>&nbsp;<br>";

// ---- Atualizar Dimensões no HSQLDB
// ----- executar todos os script com
flush();
print "<h3>Atualizando Dimensões no HSQLDB</h3><br>";
$st_comandos = "";
ob_start();
reset($hsqldb_files);
foreach ($hsqldb_files as $hsqldb_file) {
  $comando = $rota_java."java.exe -jar ".$hsqldblib." ubi
".$rota_site.$hsqldb_file;
  $output = "";
  $st_comandos.=$comando."<br>";
  $output = system($comando);
}
ob_end_clean();
print $st_comandos;
print "<br>&nbsp;<br>";

// ---- Atualizar (ou Criar) Arquivos de Carga do PDI
// ---- em: <rota_pdi>/ubi
// ---- ler->gravar Consumidor_TEMPLATE.ktr --> Consumidor_(DOMINIO)
// ---- obs: executa a carga ao final
flush();
print "<h3>Atualizando Arquivos de Carga do PDI</h3><br>";
reset($dominios_validos);
foreach ($dominios_validos as $key => $owl_file) {
  $ary_dominio = explode("/", $owl_file);
  $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
  $dominio_upper = strtoupper($dominio);
  $cubos = OWL2Array($dir_to_browse."Ontologias_de_Dominio/".$owl_file);
  print "_pdi_/ubi/Consumidor_". $dominio_upper. ".ktr<br>";
  $handle = @fopen($rota_pdi."ubi/Consumidor_TEMPLATE.ktr", "r");
  $handleo = @fopen($rota_pdi."ubi/Consumidor_". $dominio_upper. ".ktr", "w+");
  if ($handle) {
    while (!feof($handle)) {
      $buffer = fgets($handle, 4096);
      $buffer = str_replace('${DOMINIO}', $dominio_upper, $buffer);
      $buffer = str_replace('${SERVER_IP}', $server_ip, $buffer);
      $ktr_string = "";
      if (stristr($buffer, '<!-- @FIELD BLOCK -->') == TRUE) {

```

```

        $ktr_string.= "        <field>".chr(13).chr(10);
        $ktr_string.= "
<name>PK_".${dominio_upper}."</name>".chr(13).chr(10);
        $ktr_string.= "
<xpath>PK_".${dominio_upper}."</xpath>".chr(13).chr(10);
        $ktr_string.= "
<element_type>node</element_type>".chr(13).chr(10);
        $ktr_string.= "
<result_type>valueof</result_type>".chr(13).chr(10);
        $ktr_string.= "        <type>Integer</type>".chr(13).chr(10);
        $ktr_string.= "        <format/>".chr(13).chr(10);
        $ktr_string.= "        <currency/>".chr(13).chr(10);
        $ktr_string.= "        <decimal/>".chr(13).chr(10);
        $ktr_string.= "        <group/>".chr(13).chr(10);
        $ktr_string.= "        <length>-1</length>".chr(13).chr(10);
        $ktr_string.= "        <precision>-1</precision>".chr(13).chr(10);
        $ktr_string.= "        <trim_type>both</trim_type>".chr(13).chr(10);
        $ktr_string.= "        <repeat>N</repeat>".chr(13).chr(10);
        $ktr_string.= "    </field>".chr(13).chr(10);
    reset($cubos);
    foreach ($cubos as $adata) {
        foreach ($adata as $attype => $aline) {
            if ($attype=="01_HIERARQUIA") {
                $nome_dimensao = $aline['CLASSE'];
                $nome_dimensao_upper = strtoupper($nome_dimensao);
                $ktr_string.= "        <field>".chr(13).chr(10);
                $ktr_string.= "
<name>FK_".${nome_dimensao_upper}."</name>".chr(13).chr(10);
                $ktr_string.= "
<xpath>FK_".${nome_dimensao_upper}."</xpath>".chr(13).chr(10);
                $ktr_string.= "
<element_type>node</element_type>".chr(13).chr(10);
                $ktr_string.= "
<result_type>valueof</result_type>".chr(13).chr(10);
                $ktr_string.= "        <type>Integer</type>".chr(13).chr(10);
                $ktr_string.= "        <format/>".chr(13).chr(10);
                $ktr_string.= "        <currency/>".chr(13).chr(10);
                $ktr_string.= "        <decimal/>".chr(13).chr(10);
                $ktr_string.= "        <group/>".chr(13).chr(10);
                $ktr_string.= "        <length>-1</length>".chr(13).chr(10);
                $ktr_string.= "        <precision>-1</precision>".chr(13).chr(10);
                $ktr_string.= "
<trim_type>both</trim_type>".chr(13).chr(10);
                $ktr_string.= "        <repeat>N</repeat>".chr(13).chr(10);
                $ktr_string.= "        </field>".chr(13).chr(10);
            }
        }
    }
    reset($cubos);
    foreach ($cubos as $adata) {
        foreach ($adata as $attype => $aline) {
            if ($attype=="04_MEDIDA") {
                $nome_medida = $aline['CLASSE'];
                $nome_medida_upper = strtoupper($nome_medida);
                $ktr_string.= "        <field>".chr(13).chr(10);
                $ktr_string.= "
<name>VL_".${nome_medida_upper}."</name>".chr(13).chr(10);
                $ktr_string.= "
<xpath>VL_".${nome_medida_upper}."</xpath>".chr(13).chr(10);
                $ktr_string.= "
<element_type>node</element_type>".chr(13).chr(10);
                $ktr_string.= "
<result_type>valueof</result_type>".chr(13).chr(10);

```

```

        $ktr_string.= "          <type>Number</type>".chr(13).chr(10);
        $ktr_string.= "          <format/>".chr(13).chr(10);
        $ktr_string.= "          <currency/>".chr(13).chr(10);
        $ktr_string.= "          <decimal/>".chr(13).chr(10);
        $ktr_string.= "          <group/>".chr(13).chr(10);
        $ktr_string.= "          <length>-1</length>".chr(13).chr(10);
        $ktr_string.= "          <precision>-1</precision>".chr(13).chr(10);
        $ktr_string.= "
    <trim_type>both</trim_type>".chr(13).chr(10);
        $ktr_string.= "          <repeat>N</repeat>".chr(13).chr(10);
        $ktr_string.= "          </field>".chr(13).chr(10);
    }
}
}
}
$buffer = $ktr_string;
} else {
    if (stristr($buffer, '<!-- @META BLOCK -->') == TRUE) {
        $ktr_string.= "          <meta>".chr(13).chr(10);
        $ktr_string.= "
<name>PK_". $dominio_upper. "</name>".chr(13).chr(10);
        $ktr_string.= "
<rename>PK_". $dominio_upper. "</rename>".chr(13).chr(10);
        $ktr_string.= "          <type>Integer</type>".chr(13).chr(10);
        $ktr_string.= "          <length>8</length>".chr(13).chr(10);
        $ktr_string.= "          <precision>0</precision>".chr(13).chr(10);
        $ktr_string.= "          <conversion_mask/>".chr(13).chr(10);
        $ktr_string.= "
<date_format_lenient>>false</date_format_lenient>".chr(13).chr(10);
        $ktr_string.= "          <encoding/>".chr(13).chr(10);
        $ktr_string.= "
<decimal_symbol>0</decimal_symbol>".chr(13).chr(10);
        $ktr_string.= "          <grouping_symbol/>".chr(13).chr(10);
        $ktr_string.= "          <currency_symbol/>".chr(13).chr(10);
        $ktr_string.= "          <storage_type/>".chr(13).chr(10);
        $ktr_string.= "          </meta>".chr(13).chr(10);
        reset($cubos);
        foreach ($cubos as $adata) {
            foreach ($adata as $atype => $aline) {
                if ($atype=="01_HIERARQUIA") {
                    $nome_dimensao = $aline['CLASSE'];
                    $nome_dimensao_upper = strtoupper($nome_dimensao);
                    $ktr_string.= "          <meta>".chr(13).chr(10);
                    $ktr_string.= "
<name>FK_". $nome_dimensao_upper. "</name>".chr(13).chr(10);
                    $ktr_string.= "
<rename>FK_". $nome_dimensao_upper. "</rename>".chr(13).chr(10);
                    $ktr_string.= "          <type>Integer</type>".chr(13).chr(10);
                    $ktr_string.= "          <length>8</length>".chr(13).chr(10);
                    $ktr_string.= "
<precision>0</precision>".chr(13).chr(10);
                    $ktr_string.= "          <conversion_mask/>".chr(13).chr(10);
                    $ktr_string.= "
<date_format_lenient>>false</date_format_lenient>".chr(13).chr(10);
                    $ktr_string.= "          <encoding/>".chr(13).chr(10);
                    $ktr_string.= "
<decimal_symbol>0</decimal_symbol>".chr(13).chr(10);
                    $ktr_string.= "          <grouping_symbol/>".chr(13).chr(10);
                    $ktr_string.= "          <currency_symbol/>".chr(13).chr(10);
                    $ktr_string.= "          <storage_type/>".chr(13).chr(10);
                    $ktr_string.= "          </meta>".chr(13).chr(10);
                }
            }
        }
    }
}
}
}

```

```

        reset($cubos);
        foreach ($cubos as $adata) {
            foreach ($adata as $atipe => $aline) {
                if ($atipe=="04_MEDIDA") {
                    $nome_medida = $aline['CLASSE'];
                    $nome_medida_upper = strtoupper($nome_medida);
                    $ktr_string.= "        <meta>".chr(13).chr(10);
                    $ktr_string.= "
<name>VL_\".$nome_medida_upper."</name>".chr(13).chr(10);
                    $ktr_string.= "
<rename>VL_\".$nome_medida_upper."</rename>".chr(13).chr(10);
                    $ktr_string.= "        <type>Number</type>".chr(13).chr(10);
                    $ktr_string.= "        <length>17</length>".chr(13).chr(10);
                    $ktr_string.= "
<precision>2</precision>".chr(13).chr(10);
                    $ktr_string.= "        <conversion_mask/>".chr(13).chr(10);
                    $ktr_string.= "
<date_format_lenient>>false</date_format_lenient>".chr(13).chr(10);
                    $ktr_string.= "        <encoding/>".chr(13).chr(10);
                    $ktr_string.= "
<decimal_symbol>2</decimal_symbol>".chr(13).chr(10);
                    $ktr_string.= "        <grouping_symbol/>".chr(13).chr(10);
                    $ktr_string.= "        <currency_symbol/>".chr(13).chr(10);
                    $ktr_string.= "        <storage_type/>".chr(13).chr(10);
                    $ktr_string.= "        </meta>".chr(13).chr(10);
                }
            }
        }
        $buffer = $ktr_string;
    }
    else {
        if (strstr($buffer, '!-- @TABLE BLOCK -->') == TRUE) {
            $ktr_string.= "        <field>".chr(13).chr(10);
            $ktr_string.= "
<column_name>PK_\".$dominio_upper."</column_name>".chr(13).chr(10);
            $ktr_string.= "
<stream_name>PK_\".$dominio_upper."</stream_name>".chr(13).chr(10);
            $ktr_string.= "        </field>".chr(13).chr(10);
            reset($cubos);
            foreach ($cubos as $adata) {
                foreach ($adata as $atipe => $aline) {
                    if ($atipe=="01_HIERARQUIA") {
                        $nome_dimensao = $aline['CLASSE'];
                        $nome_dimensao_upper = strtoupper($nome_dimensao);
                        $ktr_string.= "        <field>".chr(13).chr(10);
                        $ktr_string.= "
<column_name>FK_\".$nome_dimensao_upper."</column_name>".chr(13).chr(10);
                        $ktr_string.= "
<stream_name>FK_\".$nome_dimensao_upper."</stream_name>".chr(13).chr(10);
                        $ktr_string.= "        </field>".chr(13).chr(10);
                    }
                }
            }
        }
        reset($cubos);
        foreach ($cubos as $adata) {
            foreach ($adata as $atipe => $aline) {
                if ($atipe=="04_MEDIDA") {
                    $nome_medida = $aline['CLASSE'];
                    $nome_medida_upper = strtoupper($nome_medida);
                    $ktr_string.= "        <field>".chr(13).chr(10);
                    $ktr_string.= "
<column_name>VL_\".$nome_medida_upper."</column_name>".chr(13).chr(10);

```



```

        $ktr_string.= "
<stream_name>VL_".$nome_medida_upper."</stream_name>".chr(13).chr(10);
        $ktr_string.= "        </field>".chr(13).chr(10);
    }
        }
        }
        $buffer = $ktr_string;
    }
}
    }
    fputs($handleo, $buffer);
}
}
fclose($handleo);
fclose($handle);
}

$str_bat = "";
reset($dominios_validos);
foreach ($dominios_validos as $key => $owl_file) {
    $ary_dominio = explode("/", $owl_file);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $str_bat.= "call pan.bat /file:". $rota_pdi."ubi/Consumidor_". $dominio." .ktr
/level:Basic".chr(13).chr(10);
}
$handle = @fopen($rota_pdi."ubi.bat", "w+");
fputs($handle, $str_bat);
fclose($handle);
print "<br>&nbsp;<br>";

// ---- Criar Proposta de Serviço de Dados
flush();
print "<h3>Criando Proposta de Serviço de Dados</h3><br>";
reset($dominios_validos);
foreach ($dominios_validos as $key => $owl_file) {
    $sdados_string = "";
    $ary_dominio = explode("/", $owl_file);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $dominio_upper = strtoupper($dominio);
    print "Proposta: ". $dominio_upper." .txt<br>";
    $handle =
@fopen($rota_site."Proposta_de_Servico_de_Dados/Proposta_". $dominio_upper." .txt",
"w+");
    $sdados_string.= '<?xml version="1.0" encoding="UTF-8"?>'.chr(13).chr(10);
    $sdados_string.= '<t>'.chr(13).chr(10);
    $sdados_string.= ' <r>'.chr(13).chr(10);
    $sdados_string.= ' <PK_'. $dominio_upper.'>'.chr(13).chr(10);
    $cubos = OWL2Array($dir_to_browse."Ontologias_de_Dominio/". $owl_file);
    reset($cubos);
    foreach ($cubos as $adata) {
        foreach ($adata as $atype => $aline) {
            if ($atype=="01_HIERARQUIA") {
                $nome_dimensao = $aline['CLASSE'];
                $nome_dimensao_upper = strtoupper($nome_dimensao);
                $sdados_string.= '
<FK_'. $nome_dimensao_upper.'>1</FK_'. $nome_dimensao_upper.'>'.chr(13).chr(10);
            }
        }
    }
}
reset($cubos);
foreach ($cubos as $adata) {
    foreach ($adata as $atype => $aline) {
        if ($atype=="04_MEDIDA") {
            $nome_medida = $aline['CLASSE'];

```

```

        $nome_medida_upper = strtoupper($nome_medida);
        $sdados_string.= '
<VL_' . $nome_medida_upper . '>1.23</VL_' . $nome_medida_upper . '>' . chr(13) . chr(10);
    }
}
}
$sdados_string.= ' </r>' . chr(13) . chr(10);
$sdados_string.= '</t>' . chr(13) . chr(10);
$sdados_string.= chr(13) . chr(10) . chr(13) . chr(10);
foreach ($cubos as $adata) {
    foreach ($adata as $atype => $aline) {
        if ($atype=="01_HIERARQUIA") {
            $nome_dimensao = $aline['CLASSE'];
            $nome_dimensao_upper = strtoupper($nome_dimensao);
            $sdados_string.= 'Dimensão: ' . $nome_dimensao_upper . chr(13) . chr(10);
            $sdados_string.= "Em: Scripts_HSQLDB, arquivo
DIM_" . $nome_dimensao_upper . ".sql" . chr(13) . chr(10) . chr(13) . chr(10);
        }
    }
}
}
fputs($handle, $sdados_string);
fclose($handle);
}
print "<br>&nbsp;<br>";

// -- Carga imediata
flush();
print "<h3>... carga imediata ...</h3><br>";
ob_start();
$output = "";
$comando = "cmd /c ".$rota_pdi."pan.bat /file:C:\pdi-ce-4.3.0-stable\data-
integration\ubi\Consumidor_DEMOGRAFIA.ktr /level:Basic";
$output = system($comando);
$comando = "cmd /c ".$rota_pdi."pan.bat /file:C:\pdi-ce-4.3.0-stable\data-
integration\ubi\Consumidor_EDUCACAO.ktr /level:Basic";
$output = system($comando);
$comando = "cmd /c ".$rota_pdi."pan.bat /file:C:\pdi-ce-4.3.0-stable\data-
integration\ubi\Consumidor_IMOVEIS_PUBLICOS.ktr /level:Basic";
$output = system($comando);
$comando = "cmd /c ".$rota_pdi."pan.bat /file:C:\pdi-ce-4.3.0-stable\data-
integration\ubi\Consumidor_SAUDE.ktr /level:Basic";
$output = system($comando);

$output = system($comando);
ob_end_clean();
print "<br>&nbsp;<br>";

print "===== FIM =====<br>";
print "<a href='diretorio.php'>Volta</a><br>";
print "</body>";
print "</html>";

function OWL2Array($filename) {
    $ary_dominio = explode("/", $filename);
    $dominio = remove_ext($ary_dominio[count($ary_dominio)-1]);
    $dominio_upper = strtoupper($dominio);
    $vart = "";
    $var1 = "";
    $var2 = "";
    $var3 = "";
    $buffer1 = "";
    $buffer2 = "";
}

```

```

$result_array = array();

$handle = @fopen($filename, "r");

if ($handle) {
    while (!feof($handle)) {
        $vart = "";
        $buffer = fgets($handle, 4096);
        /* <owl:Class rdf:about="http://localhost/ontologies/ubi#Ano"> */
        /* <owl:Class rdf:about="http://localhost/ontologies/ubi#Geo"/> */
        if (strstr($buffer, '<owl:Class') == TRUE) {
            $vart = "01_HIERARQUIA";
            $var2 = "Thing";
            if (($x_pos = strpos($buffer, '#')) != FALSE) {
                $var1 = substr($buffer, $x_pos + + strlen("#"));
            }
            else {
                $var1 = "Thing";
            }
            $buffer1 = fgets($handle, 4096);
            /* <rdfs:subClassOf
rdf:resource="http://localhost/ontologies/ubi#Tempo"/> */
            if (strstr($buffer1, '<rdfs:subClassOf') == TRUE) {
                if (($x_pos = strpos($buffer1, '#')) != FALSE) {
                    $vart = "02_NIVEL";
                    $var2 = substr($buffer1, $x_pos + + strlen("#"));
                }
                else {
                    $var2 = "Thing";
                }
            }
            /* se não está definido subClassOf eh subClassOf de Thing */
            else {
                $var2 = "Thing";
            }
        }

        /* <owl:NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#Urbano"> */
        if (strstr($buffer, 'owl:NamedIndividual') == TRUE) {
            $vart = "03_VALORES";
            $var2 = "Thing";
            if (($x_pos = strpos($buffer, '#')) != FALSE) {
                $var1 = substr($buffer, $x_pos + + strlen("#"));
            }
            else {
                $var1 = "Thing";
            }
            $buffer1 = fgets($handle, 4096);
            /* <rdf:type rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#Nivel"/> */
            if (strstr($buffer1, '<rdf:type') == TRUE) {
                if (($x_pos = strpos($buffer1, '#')) != FALSE) {
                    $var2 = substr($buffer1, $x_pos + + strlen("#"));
                }
                else {
                    $var2 = "Thing";
                }
            }
            $buffer1 = fgets($handle, 4096);
        }
    }
}

```

```

/* <rdfs:isDefinedBy>2</rdfs:isDefinedBy> */
if (stristr($buffer1, '<rdfs:isDefinedBy>') == TRUE) {
    if (($x_pos = strpos($buffer1, '<rdfs:isDefinedBy>')) != FALSE) {
        $var3 = substr($buffer1, $x_pos + + strlen("<rdfs:isDefinedBy>"));
    }
    else {
        $var3 = $var1;
    }
}
else {
    $var3 = $var1;
}
}

/* <owl:Thing rdf:about="http://www.co-ode.org/ontologies/ont.owl#Area"> */
if (stristr($buffer, '<owl:Thing>') == TRUE) {
    $vart = "04_MEDIDA";
    if (($x_pos = strpos($buffer, '#')) != FALSE) {
        $var1 = substr($buffer, $x_pos + + strlen("#"));
    }
    else {
        $var1 = "Thing";
    }
    $buffer1 = fgets($handle, 4096);
    /* <rdf:type rdf:resource="&owl;NamedIndividual"/> */
    if (stristr($buffer1, '<rdf:type>') == TRUE) {
        if (($x_pos = strpos($buffer1, '#')) != FALSE) {
            $var2 = substr($buffer1, $x_pos + + strlen("#"));
        }
        else {
            $var2 = "Thing";
        }
    }
    $buffer1 = fgets($handle, 4096);
    /*
<rdfs:isDefinedBy>http://cluster:cluster@datahost:8282/kettle/executeTrans/?trans=...
*/
    if (stristr($buffer1, '<rdfs:isDefinedBy>') == TRUE) {
        if (($x_pos = strpos($buffer1, '<rdfs:isDefinedBy>')) == TRUE) {
            $var3 = substr($buffer1, $x_pos + strlen("<rdfs:isDefinedBy>"));
        }
        else {
            $vart = "";
        }
    }
}

$var1 = trim(str_replace('</>', '', $var1));
$var1 = trim(str_replace('<>', '', $var1));
$var1 = trim(str_replace('</rdfs:isDefinedBy>', '', $var1));
$var2 = trim(str_replace('</>', '', $var2));
$var2 = trim(str_replace('<>', '', $var2));
$var2 = trim(str_replace('</rdfs:isDefinedBy>', '', $var2));
$var3 = trim(str_replace('</>', '', $var3));
$var3 = trim(str_replace('<>', '', $var3));
$var3 = trim(str_replace('</rdfs:isDefinedBy>', '', $var3));

switch ($vart) {
    case "01_HIERARQUIA":
    case "02_NIVEL":
    case "03_VALORES":
    case "04_MEDIDA":

```

```

        $line = array($vart => array('CLASSE' => $var1, 'SUBCLASSE' => $var2,
'DEFINICAO' => $var3));
        array_push($result_array, $line);
        break;
    }
}

fclose($handle);

}
return $result_array;
}

function array_copy($arr) {
    foreach ($arr as $key => $value) {
        if (is_array($value)) {
            $newArray[$key] = array_copy($value);
        }
        else {
            if (is_object($value)) {
                $newArray[$key] = clone $value;
            }
            else {
                $newArray[$key] = $value;
            }
        }
    }
    return $newArray;
}

function Hierarquia2ArrayNiveis($arr, $nome_dimensao_funcao, $virtual) {
    $stem_mais = true;
    $level_array = array();
    $nivel_anterior_upper = $nome_dimensao_funcao;
    while ($stem_mais) {
        $stem_mais = false;
        reset($arr);
        foreach ($arr as $adatal) {
            foreach ($adatal as $atypel => $aline1) {
                if ($atypel=="02_NIVEL") {
                    if (strtoupper($aline1['SUBCLASSE'])==$nivel_anterior_upper) {
                        $stem_mais = true;
                        $nome_nivel = $aline1['CLASSE'];
                        $nome_nivel_upper = strtoupper($nome_nivel);
                        $nivel_anterior_upper = $nome_nivel_upper;
                        array_push($level_array, strtoupper($nome_nivel_upper));
                    }
                }
            }
        }
    }
    return $level_array;
}
?>

```