Pontifícia Universidade Católica
do Rio de Janeiro

Marcia Lucas Pesce

RdXel:

Um conjunto de ferramentas para manipulação de dados estatísticos em RDF por meio de planilhas

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Prof.ª Karin Breitman

Rio de Janeiro

Setembro de 2012

Pontifícia Universidade Católica
do Rio de Janeiro

Marcia Lucas Pesce

RdXel:

Um conjunto de ferramentas para manipulação de dados estatísticos

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof.ª Karin Breitman
Orientador
Departamento de Informática - PUC-Rio

Prof. Marco Antonio Casanova
Departamento de Informática – PUC-Rio

Prof. José Viterbo Filho
Departamento de Informática - UFF

Prof. Antonio Luz Furtado
Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 17 de Setembro de 2012

Marcia Lucas Pesce

Marcia Lucas Pesce graduou-se em Informática pela PUC-RIO em 2006. Desde então trabalha como analista de sistemas na Fundação Getúlio Vargas, se envolvendo em diversos projetos do IBRE (Instituto Brasileiro de Economia). Possui interesse acadêmico e profissional nas áreas ligadas à Web Semântica e Linked Data.

Ficha Catalográfica

Pesce, Marcia Lucas

RdXel: um conjunto de ferramentas para manipulação de dados estatísticos em RDF por meio de planilhas / Marcia Lucas Pesce ; orientador: Karin Breitman. – 2012.

79 f. : il. (color.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2012.

Inclui bibliografia

1. Informática – Teses. 2. Web semântica. 3. Linked data. 4. Triplificação. 5. RDF. 6. Data cube vocabulary. 7. Excel. I. Breitman, Karin. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD:004

Dedico este trabalho a todas as pessoas que contribuíram de alguma forma para a minha formação, em especial a minha família e amigos.

Agradecimentos

À minha orientadora Professora Karin Breitman pelo apoio, incentivo e confiança no meu trabalho.

A todos os professores e funcionários do Departamento pelos ensinamentos e ajuda.

A todos os amigos feitos durante esse percurso, em especial: Edgard Marx, Gustavo Miranda, Lívia Ruback, Olisses Baggio, Percy Salas e Sergio Ortiga.

A todos os amigos, que de alguma forma me incentivaram e apoiaram.

Aos meus colegas de trabalho, em especial ao meu chefe, João Luís, que sempre me apoiou.

A toda a minha família pelo apoio incondicional.

À minha mãe, minha maior inspiração.

Resumo

Pesce, Marcia Lucas; Breitman, Karin. **RdXel: Um conjunto de ferramentas para manipulação de dados estatísticos em RDF por meio de planilhas.** Rio de Janeiro, 2012. 79p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Dados estatísticos são uma das mais importantes fontes de informação para atividades humanas e organizações. No entanto, o acesso, consulta e correlação deste tipo de dados demanda grande esforço, principalmente em situações que envolvem diferentes organizações. Soluções que facilitem o acesso e a integração de grandes bases de dados analíticos, desta forma, agregam muito valor a este cenário. Neste trabalho propomos um arcabouço de software que permite com que dados estatísticos sejam eficientemente transformados e representados no formato de triplas RDF. Utilizando como base o *DataCube Vocabulary*, padrão W3C para o processo de triplificação de informações, a solução proposta facilita a consulta, análise, e reuso dos dados quando no formato RDF. O processo inverso, RDF para Excel, também é suportado, de modo a oferecer uma solução para a integração e consumo de dados RDF a partir de planilha.

Palavras-Chave

Web Semântica; Linked Data; Triplificação; RDF; Data Cube Vocabulary; Excel.

Abstract

Pesce, Marcia Lucas; Breitman, Karin (Advisor). **RdXel: A toolkit for RDF statistical data manipulation through spreadsheets.** Rio de Janeiro, 2012. 79p. MSc. Dissertation – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Statistical data represent one of the most important sources of information both for humans and organizations alike. However, accessing, querying and correlating statistical data demand a great deal of effort, especially in situations that involve different organizations. Therefore, solutions to facilitate the manipulation and integration of large statistical databases add value to this scenario. In this dissertation we propose a framework that allows statistical data to be efficiently processed and represented as RDF triples. Based on the DataCube Vocabulary, W3C's triplification standard, the proposed solution makes it easy to query, analyze, and reuse statistical data in RDF format. The reverse process, RDF for Excel, is also supported, so as to offer a solution for the integration and use of RDF data in spreadsheets.

Keywords

Semantic Web; Linked Data; Triplification; RDF; Data Cube Vocabulary; Excel.

Sumário

Lista de Figuras

Lista de Quadros

Lista de Tabelas

1
Introdução

Segundo um dos idealizadores da Web Semântica, Tim Bernes-Lee (Bernes-Lee, et al., 2001), a mesma não se resume a apenas disponibilizar dados e meta dados na Web, mas sim possibilitar que pessoas e máquinas possam explorar esses dados através de links. O termo *Linked Data*[1] refere-se a um conjunto de regras, princípios e melhores práticas para se publicar e interligar dados estruturados na Web possibilitando a melhor exploração desses dados (Bernes-Lee, 2007).

Com a adoção do padrão definido pela Web Semântica (Breitman, et al., 2006), provedores de conteúdo tornaram-se capazes de publicar informações utilizando-se de vocabulários específicos e de criar *interfaces* de consulta para acessar esses dados. A W3C (World Wide Web Consortium) recomenda a utilização do padrão *Linked Data* (Bizer, et al., 2007) baseado no conjunto de triplas RDF, para esta tarefa.

A efetiva disseminação, e consequente adoção, da Web Semântica depende de um pré-requisito fundamental: a existência de grandes quantidades de dados em formas de triplas RDF, interconectados na *Web*. O projeto LOD[2] (*Linking Open Data*) tem disponibilizado diversas bases de dados abertos no formato RDF na *Web* e desenvolvido mecanismos automatizados para interconectar estes dados de acordo com as práticas *Linked Data*. A Figura 1, conhecida como nuvem LOD[3], é mantida por Richard Cyganiak[4] e Anja Jentzsch[5]. A figura mostra os repositórios de dados e suas interligações, publicados segundo as práticas *Linked Data* pelos membros da comunidade

[1] http://linkeddata.org

[2] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[3] http://richard.cyganiak.de/2007/10/lod

[4] http://richard.cyganiak.de/#/me

[5] http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/team/JentzschAnja.html

Linking Open Data (Cyganiak, et al., 2011) composta de diversas organizações e indivíduos.

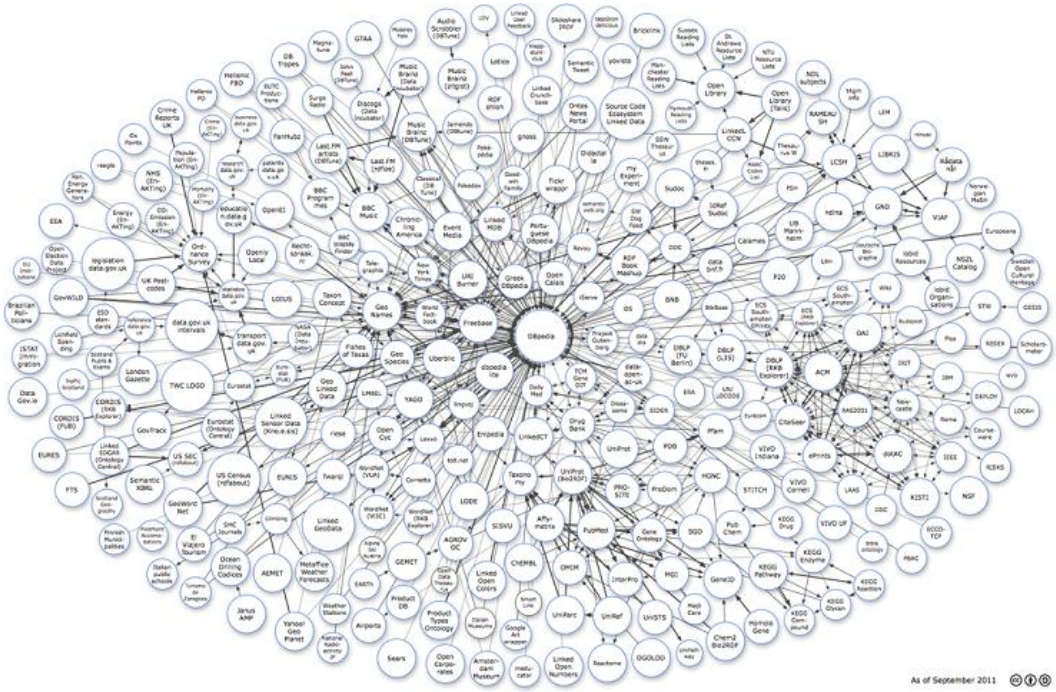


Figura 1 - Nuvem LOD (*Lod Cloud Diagram*) em Setembro 2011

Comparando a Nuvem da LOD atual (Figura 1) com a Nuvem da LOD de 5 anos atrás (Figura 2), podemos observar claramente o grande crescimento do número de fontes de dados em formato RDF. Um estudo realizado com base em uma massa de dados de 12 bilhões de páginas indexadas pelo Yahoo, mostra que a utilização do RDF cresceu 510% entre março de 2009 e Outubro de 2010. No entanto, o percentual de páginas que utilizam a Web Semântica ainda continua pequeno (3,6%).

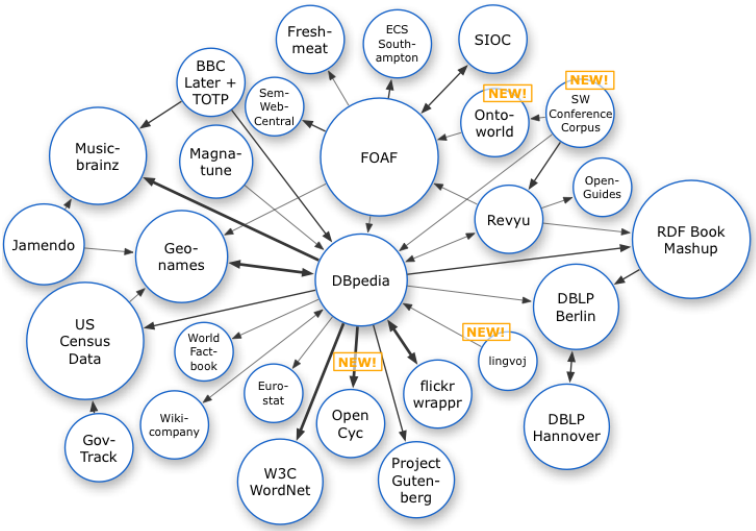


Figura 2 – Nuvem LOD (*Lod Cloud Diagram*) em Setembro de 2007

1.1 Motivação

Dados estatísticos formam uma das mais importantes fontes de informação utilizadas tanto por pessoas quanto por organizações. No domínio governamental, dados estatísticos fornecem uma visão da sociedade delineando os pontos fortes e fracos do governo, servindo de *input* para decisões futuras. Na Ciência, dados estatísticos que representam observações ou medições servem de base para verificar ou refutar teorias científicas. No domínio empresarial, dados estatísticos sobre as vendas de seus produtos, sobre a evolução do mercado ou sobre indicadores econômicos são uma contribuição crucial para decisões estratégicas de gestão (Salas, et al., 2012).

O levantamento de dados estatísticos, de maneira geral, gera um grande volume de dados e demanda muito tempo, principalmente em situações envolvendo diferentes organizações. Para que seja possível agregar e integrar dados estatísticos é de suma importância que os critérios utilizados sejam descritos com a semântica adequada, e ligados a ontologias bem reputadas.

Com base nestes requisitos, percebemos grande valor em possibilitar que grandes bases de dados analíticos sejam eficientemente transformadas e representadas de acordo o paradigma *Online Analytical Processing* (OLAP) (Thomsen, 1997), no formato de triplas RDF. Para tal o W3C disponibiliza um vocabulário específico, o *Data Cube Vocabulary*, baseado no padrão SDMX (SDMX, 2012). Este vocabulário foi concebido para representar dados

estatísticos multidimensionais, utilizando-se RDF. Esse vocabulário também faz uso de recursos do *content oriented guidelines* (COG), que definem um conjunto de conceitos estatísticos comuns e listas de códigos associados que podem ser reutilizados em todos os conjuntos de dados (SDMX, 2009).

Com o objetivo de permitir ao usuário filtrar e visualizar dados estatísticos armazenados em um catálogo através de transformações nos cubos de dados disponíveis, criamos o Framework *OLAP2DataCube Catalog On Demand*, que é composto dos seguintes módulos: Client Application (Application Code), Mediator (Integrated Web Data), Catalog (Web Data Acces Modulo) e Wrapper. A Figura 3 mostra a arquitetura do Framework, e o modo com que suas camadas se comunicam:

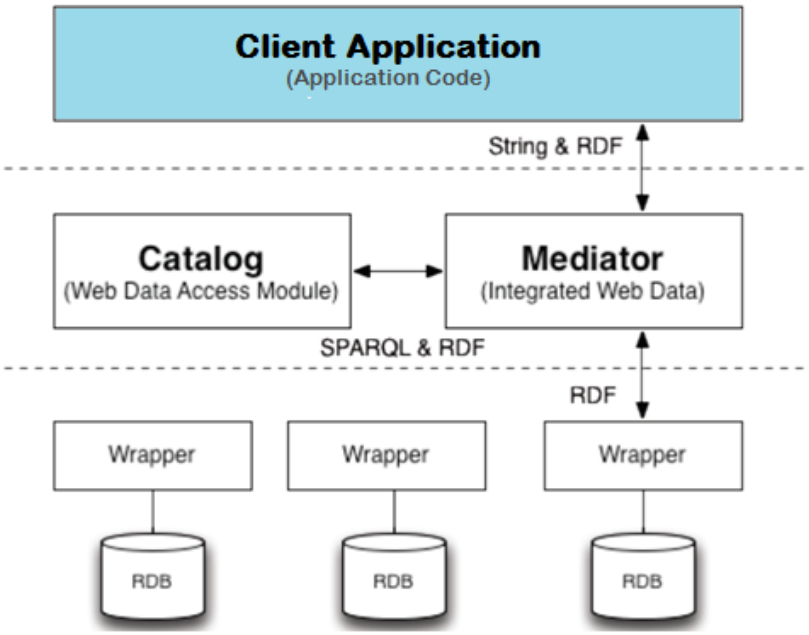


Figura 3 – Arquitetura do Framework *OLAP2DataCube Catalog On Demand*

Em linhas gerais o Framework é composto dos seguintes elementos, que serão apresentados em detalhe no Capítulo 4 desta dissertação:

1. Camada de Aplicação (*Client Application*), permite com que usuários interajam com o Framework informando que dados desejam buscar. Esta informação é repassada para a camada inferior, o Mediador (*Mediator*);

2. Camada de Mediação é responsável por chamar o Catálogo (*Catalog*), para que ele retorne os cubos que atendem à busca feita pelo usuário, e por repassar esse arquivo para a Aplicação. A Aplicação, então, exibe uma lista contendo todos os cubos retornados, para que o usuário faça sua escolha. Após escolhido um dos cubos, o usuário seleciona dimensões e métricas de seu interesse. Desta forma a camada de aplicação fica responsável por montar um arquivo no formato RDF, que codifica as escolhas feitas pelo usuário, e por chamar a camada de mediação (Mediador) requisitando os dados correspondentes;
3. Camada do Wrapper tem como responsabilidade acessar o banco de dados e gerar triplas RDF. O Mediador repassa o arquivo gerado pelo *Wrapper* para a Aplicação, que exibe esses dados para o usuário.

Essa dissertação se propõe a detalhar, especificar e desenvolver a camada de Aplicação (*Client Application*) do Framework. O aplicativo de software escolhido como plataforma para o desenvolvimento para esta solução foi o MS Excel, pois constitui uma plataforma amplamente difundida, com mais de 500 milhões de usuários[6], e que oferece versões proprietárias para vários sistemas operacionais, bem como open source (Open Office) (Apache, 2012). A escolha da plataforma foi largamente influenciada pela facilidade que a mesma oferece junto a sua base de usuários. Desta forma, julgamos que a melhor forma de interagir com o usuário seria desenvolvendo um plug-in para o MS Excel pois, desta forma, ele poderá extrair diretamente os dados resultantes de sua busca para uma ferramenta que já está habituado a trabalhar. Acreditamos que esta escolha contribui na usabilidade da solução aqui proposta.

O plug-in desenvolvido nesta dissertação também é capaz de realizar a conversão no sentindo inverso, ou seja, de ler os dados disponíveis em uma planilha MS Excel para o formato de triplas RDF, utilizando o *Data Cube Vocabulary*.

---

[6] http://blogs.technet.com/b/office2010/archive/2009/10/07/new-ways-to-try-and-buy-microsoft-office-2010.aspx

1.2 Objetivo

Nesta dissertação propomos um plug-in para o MS Excel que possibilita a conversão bidirecional de dados para o formato de triplas RDF, i.e., a conversão de dados de planilhas MS Excel para RDF, bem como dados RDF, anotados utilizando-se o *Data Cube Vocabulary*, para o formato MS Excel.

1.3 Contribuições

As principais contribuições desta dissertação são:

- ✓ Especificação, desenvolvimento e teste de dois plug-ins para o MS Excel. O primeiro permite a conversão de dados OLAP para o formato RDF; o segundo permite a operação inversa, i.e. a conversão de triplas RDF (desde de que anotadas utilizando-se o *Data Cube Vocabulary*), para o formato MS Excel.
- ✓ Uma interface gráfica que guia os usuários durante o processo de conversão dos dados em ambos sentidos, RDF2Excel e Excel2RDF, permitindo a seleção e manipulação dos dados.

1.4 Organização do trabalho

O restante da dissertação foi estruturado da seguinte forma:

- ✓ Capítulo 2 – Fundamentos: são revisados os conceitos básicos sobre os quais este trabalho está baseado;
- ✓ Capítulo 3 – Trabalhos Relacionados: são apresentados os principais trabalhos relacionados ao tópico abordado por esta dissertação. Estes estão subdivididos em Ferramentas de conversão de um Formato Específico para RDF e Ferramenta de conversão de RDF para Modelos Multidimensionais;
- ✓ Capítulo 4 – Framework *OLAP2DataCube Catalog On Demand*: descreve o Framework, detalhando sua arquitetura, camadas e processo;

- ✓ Capítulo 5 – RdXel: descreve a ferramenta RdXel, em dois módulos: RDF2Excel e Excel2RDF. Para cada módulo detalhamos sua arquitetura e processo;
- ✓ Capítulo 6 – Conclusão: são apresentadas as conclusões finais, limitações da ferramenta, e trabalhos futuros.

1.5 Resumo Capítulo 1

Neste capítulo introduzimos os temas que serão abordados nesta dissertação, descrevendo os principais motivos e objetivos que nos levaram a desenvolvê-la. Listamos as contribuições esperadas, bem como apresentamos a forma como o trabalho está organizado.

2
Fundamentos

2.1
Web Semântica

A Web semântica é uma extensão da Web atual, que permite com que computadores e humanos trabalhem em cooperação.[7] A Web semântica interliga significados de palavras e, neste âmbito, tem como finalidade atribuir significado (sentido) aos conteúdos publicados na Internet de modo que sejam perceptíveis tanto pelo humano como por computadores.

O conceito da Web Semântica surgiu em 2001, quando Tim Berners-Lee, James Hendler e Ora Lassila publicaram um artigo, na revista *Scientific American*, intitulado: "Web Semântica: um novo formato de conteúdo para a Web que tem significado para computadores vai iniciar uma revolução de novas possibilidades." (Berners-Lee, et al., 2001)

A W3C define a Web Semântica como uma Web de dados[8]. Ela afirma que existe uma grande quantidade de dados que usamos todos os dias, e que não fazem parte da Web atual, pois os dados são controlados por aplicações, e cada aplicação os mantém de forma isolada. Para a W3C o objetivo da Web Semântica é estender os princípios utilizados atualmente nos documentos disponíveis na Web, para os dados disponíveis na Web. Para concretizar esse objetivo se faz necessário a criação de um *framework* que permita com que os dados sejam compartilhados e reutilizados pelas aplicações e organizações, e para que esses dados possam ser processados automaticamente por ferramentas.

Dentre os resultados dos esforços para a criação da Web Semântica podemos destacar o padrão RDF[9], as linguagens para definição de vocabulários

[7] http://www.w3.org/2001/sw/SW-FAQ#What1

[8] http://www.w3.org/2001/sw/SW-FAQ#swgoals

[9] http://www.w3.org/standards/techs/rdf#w3c_all

RDFS[10] e OWL[11], a linguagem de consultar SPARQL[12], as práticas *Linked Data* e o projeto *Linked Open Data*, que serão discutidos nas seções seguintes.

2.2 RDF

O *Resource Description Framework* (RDF) é uma linguagem para a representação de informações na WWW. O RDF foi particularmente projetado para representar metadados sobre recursos da Web, como o título, autor, data de alteração de uma página Web, direitos autorais, licenciamento sobre um documento na Web, cronograma de disponibilidade de algum recurso compartilhado, ou a descrição das preferências de um usuário da Web para entrega de informação. Entretanto, pela generalização do conceito "recurso da Web", o RDF pode ser usado para representar informações sobre qualquer coisa que possa ser identificada por meio de uma URI, mesmo quando não pode ser diretamente recuperada pela Web. Exemplos incluem a informação de itens disponíveis em uma página de comércio eletrônico (e.g., informações sobre preços, editores e disponibilidade de livros ou CDs) ou a descrição das preferências de um usuário da Web para entrega de informação.

O RDF foi projetado para situações em que a informação precisa ser processada por aplicações, em vez de simplesmente ser mostrada para pessoas. O RDF fornece um *framework* comum para expressar esta informação de modo que possa ser trocada entre aplicações sem perda de significado. Como fornece um *framework* comum, projetistas de aplicações podem aproveitar a disponibilidade de ferramentas comuns para processamento e análise de informações descritas em RDF. A capacidade de troca de informações entre diferentes aplicações significa

[10] http://www.w3.org/TR/rdf-shcema

[11] http://www.w3.org/TR/owl-guide

[12] http://www.w3.org/TR/rdf-sparql-query

> que as informações podem ser disponibilizadas para outras aplicações que não aquelas para as quais foram originalmente criadas.
>
> (Manola, et al., 2004)

O RDF é baseado na ideia da identificação de itens de interesse, chamados de recursos, usando identificadores da Web (chamados de *Uniform Resource Identifiers*, ou URIs[13]), e na descrição destes recursos em termos de propriedades e seus valores. Isto permite representar declarações simples sobre recursos sob a forma de triplas <*sujeito*, *propriedade*, *valor*>, onde o URI do recurso de interesse é o *sujeito* da tripla. Como o valor de uma propriedade pode ser o URI de outro recurso, um conjunto de triplas desta forma pode ser entendido como um grafo, composto por nós e arcos, que representam recursos, suas propriedades e valores.

O exemplo a seguir, reproduzido de (Manola, et al., 2004), ilustra uma representação em RDF para descrever a sentença: "Existe uma Pessoa identificada por http://www.w3.org/People/EM/contact#me, cujo nome é Eric Miller, seu endereço de e-mail é em@w3.org e seu título é Dr." Poderia ser representado pelo grafo apresentado na Figura 4.

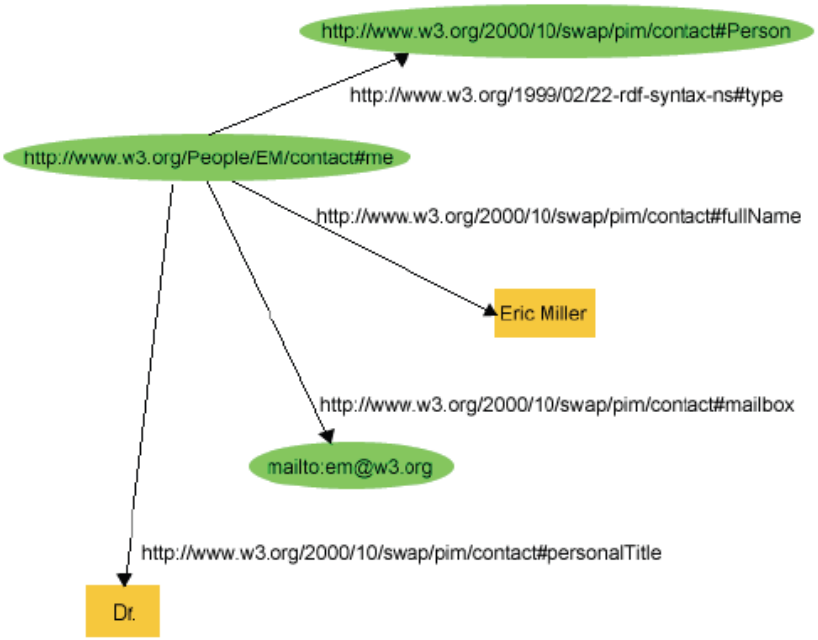


Figura 4 – Um grafo RDF descrevendo Eric Miller (Manola, et al., 2004)

[13] http://www.ietf.org/rfc/rfc2396.txt

A Figura 4 ilustra o RDF utilizando URIs para identificar:

- ✓ Indivíduos, e.g., Eric Miller, identificado por: http://www.w3.org.People/EM/contact#me;
- ✓ Tipos de coisas, e.g., Pessoa, identificada por: http://www.w3.org/2000/10/swap/pim/contact#Person;
- ✓ Propriedades das coisas, e.g., *mailbox*, identificada por http://www.w3.org/2000/10/swap/pim/contaxt#mailbox;
- ✓ Valores das propriedades, e.g., mailto:em@w3..org como valor da propriedade *mailbox* (o RDF também usa cadeia de caracteres como "Eric Miller", e valores de outros tipos como inteiros e datas, como valores de propriedades).

O RDF também possui uma sintaxe baseada em XML (RDF/XML) para representação destes grafos. O Quadro 1 mostra um trecho de RDF na notação RDF/XML, que corresponde a parte do grafo ilustrado na Figura 4.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

Quadro 1 – RDF descrevendo Eric Miller

Além da notação RDF/XML, um grafo RDF pode ser representado em outras notações: N3[14], NTriples[15] e Turtle[16].

O RDF fornece uma maneira flexível para descrever as informações e recursos, e como eles se relacionam uns com os outros. Ele permite publicar

[14] http://www.w3.org/DesignIssues/Notation3

[15] http://www.w3.org/2001/sw/RDFCore/ntriples

[16] http://www.w3.org/TeamSubmission/turtle

informações na Web de uma forma que os outros possam descobrir e reutilizar (Heath, et al., 2011).

Através da montagem de *hiperlinks* explícitos, as ferramentas capazes de ler RDF se tornam capazes seguir os mesmos de forma a descobrir mais dados na Web. Dados publicados e interligados através do RDF se tornam significativamente mais fáceis de encontrar, e, portanto, mais utilizáveis.

2.3
RDF Schema

Para descrever tipos ou classes específicas de recursos utilizando-se RDF precisamos de meios para definir os vocabulários (termos) que pretendemos utilizar.

Estas classes e propriedades devem ser descritas através de um vocabulário específico, utilizando extensões do RDF fornecidas pela Linguagem de Descrição de Vocabulários RDF ou RDF Schema (RDFS) (Brickley, et al., 2004).

O RDFS não fornece um vocabulário para classes de um domínio específico. Na verdade, ele compõe um conjunto de classes que fornecem os elementos básicos para a descrição de ontologias, também chamados vocabulários RDF, destinadas a estruturar os recursos RDF.

A primeira versão do RDF Schema foi publicada pela W3C em abril de 1998. Muitos componentes do RDFS foram incluídos na linguagem *Web Ontology Language* (OWL), detalhada na próxima seção.

2.4
OWL

A *Web Ontology Language* (OWL) é uma linguagem para definir e instanciar ontologias na Web (Smith, et al., 2-004). O termo ontologia veio emprestado da Filosofia, e se refere à ciência de descrever os tipos das entidades do mundo e como elas se relacionam. Uma ontologia OWL pode incluir descrições de classes, propriedades e suas instâncias.

A linguagem foi projetada para ser utilizada por aplicações que precisam processar o conteúdo da informação ao invés de apenas apresentá-la aos humanos. Comparada com outras linguagens, como XML e RDF, A OWL torna mais fácil a interpretação do conteúdo Web por fornecer um vocabulário adicional através de uma semântica formal.

Se compararmos o RDFS a OWL, veremos que a OWL é mais expressiva, pois é capaz de representar vários tipos de relacionamentos entre classes e propriedades que não são possíveis de serem representados utilizando-se apenas o RDFS. Recentemente a linguagem passou a ser recomendada pela W3C, que introduziu diversas melhorias.

2.5
SPARQL

SPARQL é um protocolo (Kendall, et al., 2008) e linguagem de consultas para RDF (Prud'Hommeaux, et al., 2008)

A linguagem de consultas SPARQL está para o RDF assim como a linguagem SQL está para os bancos de dados relacionais, sendo esta, uma linguagem específica para consultas em grafos RDF. O Quadro 2 apresenta uma expressão de consulta em linguagem SPARQL, construída para retornar o título do livro identificado pela URI *http://example.org/book/book1*.

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title>
?title .
}
```

Quadro 2 – Exemplo de consulta na linguagem SPARQL

Um *Sparql EndPoint* é um serviço que implementa o protocolo SPARQL. Ele permite que o usuário (humano ou máquina) faça uma consulta a uma base de conhecimento utilizando a linguagem SPARQL. Um exemplo prático de utilização de um *Sparql EndPoint* para explorar a Web Semântica é o projeto LOD que disponibiliza um conjunto de repositórios RDF interconectados, que podem ser acessados por meio de diversos *Sparql EndPoints*.

2.6
Linked Data

O termo *Linked Data* (Bizer, et al., 2009) pode ser definido como um conjunto de regras, princípios e melhores práticas para possibilitar a publicação e interligação de dados estruturados na Web. Seu principal objetivo é facilitar a exploração desses dados (Bernes-Lee, 2007), como ilustrado pelo trecho a seguir.

> "A Web Semântica não se trata apenas de publicar dados na Web. Ela trata sobre como fazer *links*, de modo que uma pessoa ou máquina possa explorar a Web de dados. Com o *linked data*, a partir de um determinado dado, você pode encontrar outros dados relacionados. Como na Web convencional, a Web de dados é formada por documentos disponíveis na Web. No entanto, na Web convencional, as ligações são formadas através das relações entre documentos de hipertexto escritos em HTML, enquanto que, na Web de dados, as ligações são formadas por relações descritas através do padrão RDF."
>
> (Bernes-Lee, 2007)

Os princípios do *Linked Data*, segundo Tim Bernes-Lee[17], são:

- ✓ Utilizar *Uniform Resource Identifiers* (URIs) para identificar todo tipo de coisas;
- ✓ Utilizar URIs HTTP para que seja possível encontrar esses nomes na Web;
- ✓ Quando alguém procurar um URI deve encontrar informações úteis, através dos padrões da Web Semântica (RDF, RDFS);
- ✓ Incluir *links* (elos) para outros URIs, para que seja possível encontrar outras informações através destes;

Os quatro itens acima ficaram conhecidos como os "Princípios *Linked Data*" e são tidos como um guia de como publicar, e conectar dados por meio

[17] http://www.w3.org/DesignIssues/LinkedData.html

da Web. Eles levam em consideração que o principal objetivo do *Linked Data* é permitir que as pessoas compartilhem dados estruturados na Web da mesma forma que compartilham documentos hoje em dia (Heath, et al., 2011).

2.7
Vocabulários Padrão

Vocabulários fornecem termos específicos para descrever classes de objetos no mundo, e como eles se relacionam entre si. Dependendo do seu poder de expressão, vocabulários podem ser classificados como taxonomias ou como ontologias (McGuinness, 2002).

Dependendo do domínio comunidades dão preferência a vocabulários específicos para descrever e publicar seus dados. A Web de dados, portanto, compreende vários vocabulários especializados, utilizados em paralelo. De acordo com (Heath, et al., 2011), é considerado uma boa prática a reutilização de termos de vocabulários conhecidos RDF, sempre que possível.

Se encontrarmos os termos adequados para representar uma dada informação em vocabulários existentes, estes devem ser reutilizados para descrevê-los. A reutilização de termos de vocabulários existentes é altamente desejável, uma vez que maximiza a probabilidade de que os dados sejam consumidos por aplicações que compartilhem esses vocabulários, sem necessidade de alterações no aplicativo. Os vocabulários na lista a seguir cobrem um conjunto de diferentes domínios e são utilizados por muitas comunidades. Para garantir a interoperabilidade, recomenda-se que estes vocabulários sejam reutilizados sempre que possível (Bizer, et al., 2007).

- ✓ *Dublin Core Metadata Initiative* (DCMI)[18] – utilizado para representar metadados como título, autor, data e assunto;
- ✓ *Friend-of-a-Friend* (FOAF)[19] – utilizado para descrever pessoas, suas atividades e seus relacionamentos com outras pessoas e objetos;
- ✓ *Description of a Project* (DOAP)[20] – utilizado para descrever projetos de software, em particular os projetos *Open Source*;

[18] http://dublincore.org/documents/demi-terms

[19] http://xmlns.com/foaf/spec

- ✓ *Music Ontology*[21] - utilizado para descrever diversos aspectos relacionados à música, como artistas, álbuns, faixas, performances e arranjos;
- ✓ *Good Relations Ontology*[22] - utilizado para descrever produtos, serviços e outros aspectos relevantes para aplicações de comércio eletrônico;
- ✓ *Bibliographic Ontology (BIBO)*[23]- utilizado para descrever citações e referências bibliográficas (citações, livros, artigos, etc.);
- ✓ *Basic Geo (WGS84)*[24] – utilizado para descrever termos como latitude e longitude e para descrever localizações geográficas;
- ✓ *Data Cube Vocabulary*[25] - utilizado para descrever dados estatísticos multidimensionais e seus metadados;

Somente devemos definir uma nova terminologia caso os termos desejados não possam ser representados por nenhum dos vocabulários existentes (Bizer, et al., 2007). A W3C fornece um conjunto de orientações para ajudar os usuários na publicação de novos vocabulários, tais como:

"if new terminology is defined, it should be made self-describing by making the URIs that identify terms Web dereferencable. This allows clients to retrieve RDF Schema or OWL definitions of the terms as well as term mappings to other vocabularies."

(Berrueta, et al., 2008).

Traduzido a seguir:

"se uma nova terminologia é definida, ela deve ser auto-descricritiva, tornando as URIs que identificam seus termos derreferenciáveis. Isso permite que os clientes recuperem o *RDF Schema* ou as definições OWL dos termos, bem como os mapeamentos para outros vocabulários."

---

[20] http://trac.usefuline.com/doap

[21] http://musicontology.com

[22] http://purl.org/goodrelations

[23] http://bibliontology.com

[24] http://www.w3.org/2003/01/geo

[25] http://www.w3.org/TR/vocab-data-cube/

Na seção seguinte descrevemos o *Data Cube Vocabulary*, central para este trabalho.

2.7.1
Data Cube Vocabulary

Em muitas situações queremos ter a possibilidade de publicar dados multidimensionais, tais como dados estatísticos, de tal maneira que eles possam ser relacionados a outros conjuntos de dados. O *Data Cube Vocabulary* fornece um meio para publicarmos esses dados através do padrão RDF. O modelo por trás do *Data Cube Vocabulary* é compatível com o modelo SDMX[26] (*Statistical Data and Metadata Exchange*), padrão ISO para intercâmbio e compartilhamento de dados e metadados estatísticos entre as organizações (ISO, 2005). O *Data Cube Vocabulary* permite a extensão do SDMX possibilitando a publicação de outros aspectos dos fluxos de dados estatísticos.

A base de um conjunto de dados estatísticos é formada por um conjunto de valores organizados através de grupos de dimensões e metadados associados. O *Data Cube Vocabulary* foi concebido de forma generalista para que, desta forma, possa ser utilizado por diferentes fontes de dados como planilhas e cubos de dados OLAP.

O *Data Cube Vocabulary* se baseia nos seguintes vocabulários RDF existentes:

- ✓ *Simple Knowledge Organization System* (SKOS)[27] – para definição dos esquemas;
- ✓ *Statistical Core Vocabulary* (SCOVO)[28] – para as estruturas dos dados estatísticos;
- ✓ VoID[29] - para o acesso aos dados;

[26] http://sdmx.org/

[27] http://www.w3.org/2004/02/skos/

[28] http://sw.joanneum.at/scovo/schema.html

- ✓ *Friend-of-a-Friend* (FOAF) – para as organizações;
- ✓ *Dublin Core Terms*[30] - para os metadados;

As URIs geralmente são expressas usando uma notação compacta, onde o nome está escrito na forma - prefixo:nome - onde o prefixo identifica um *namespace* de um URI. O *namesapce* identificado pelo prefixo é anexado ao *localname* para obter o URI completo. Os *namespaces* mais conhecidos são:

- ✓ rdf, rdfs – para os *namespaces* padrões do RDF;
- ✓ dc – Dublin Core;
- ✓ skos – Simple Knowledge Organization System;
- ✓ foaf – Friend Of A Friend;
- ✓ void – Vocabulary of Interlinked Datasets;
- ✓ scovo – Statistical Core Vocabulary;
- ✓ qb – Data Cube Vocabulary.

A seguir descrevemos brevemente as classes que compõe o *Data Cube Vocabulary*, ilustradas na Figura 5. Na seção seguinte apresentaremos um exemplo de representação de dados que ilustra a utilização prática das classes previstas pelo *Data Cube Vocabulary*.

[29] http://www.w3.org/TR/void/

[30] http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=terms

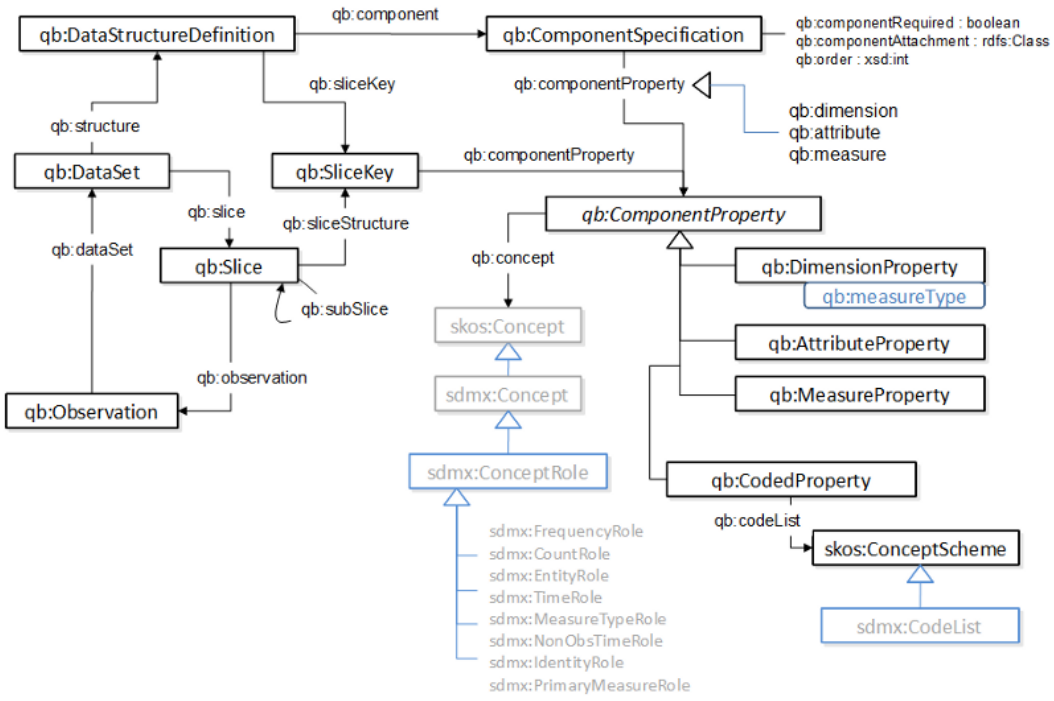


Figura 5 – Estrutura do *Data Cube Vocabulary*

a) *Class:* qb:DataSet *Sub class of:* qb:Attachable *Equivalent to:* scovo:Dataset

Representa uma coleção de observações, possivelmente organizados em vários *slices*, de acordo com uma estrutura multidimensional.

b) *Class:* qb:Observation *Sub class of:* qb:Attachable *Equivalent to:* scovo:Item

Uma única observação no cubo, podendo ter um ou mais valores associados.

c) *Class:* qb:Slice *Sub class of:* qb:Attachable

Denota um subconjunto de dados.

d) *Class:* qb:ComponentProperty *Sub class of:* rdf:Property

Propriedade abstrata que representa as dimensões, atributos e medidas.

e) *Class:* qb:DimensionProperty *Sub class of:* qb:ComponentProperty qb:CodedProperty

Representa as dimensões do cubo.

f) *Class:* qb:AttributeProperty *Sub class of:* qb:ComponentProperty
Representa os atributos do cubo.

g) *Class:* qb:MeasureProperty *Sub class of:* qb:ComponentProperty
Representa os valores medidos do cubo.

h) *Class:* qb:CodedProperty *Sub class of:* qb:ComponentProperty
Superclasse de todos os *ComponentProperties*.

i) *Class:* qb:DataStructureDefinition *Sub class of:* qb:ComponentSet
Define a estrutura de um *DataSet* ou *slice.*

j) *Class:* qb:ComponentSpecification *Sub class of:* qb:ComponentSet
Define as propriedades de um componente (atributo, dimensão, etc.).

k) *Class:* qb:SliceKey *Sub class of:* qb:ComponentSet
Denota um subconjunto das propriedades de um *DataSet*, que são fixadas nos *slices* correspondentes.

2.7.2
Exemplo de Utilização do Data Cube Vocabulary

A Tabela 1 descreve um cubo demonstrativo de Compras por Tipo de Estabelecimento, Data e Características do Cliente. Ela demonstra um exemplo de dados que podem ser representados utilizando-se o *Data Cube Vocabulary.*

		2010			
		Junho		**Julho**	
		R$	**Qtd.**	**R$**	**Qtd.**
Teatro	**Masculino**	150,00	2	200,00	3
	Feminino	180,00	2	230,00	3
Cinema	**Masculino**	60,00	3	60,00	3
	Feminino	40,00	2	60,00	3

Tabela 1 – Cubo Demonstrativo de Compras de Ingressos

Na tabela podemos ver que existem quatro conceitos - ano, mês, tipo de estabelecimento e sexo. Os conceitos ano e mês fazem parte da dimensão Tempo. O conceito Sexo faz parte da dimensão Cliente. As observações são representadas por duas medidas – valor e quantidade. A Figura 6 apresenta diagrama em estrela relativo ao cubo de dados da Tabela 1.

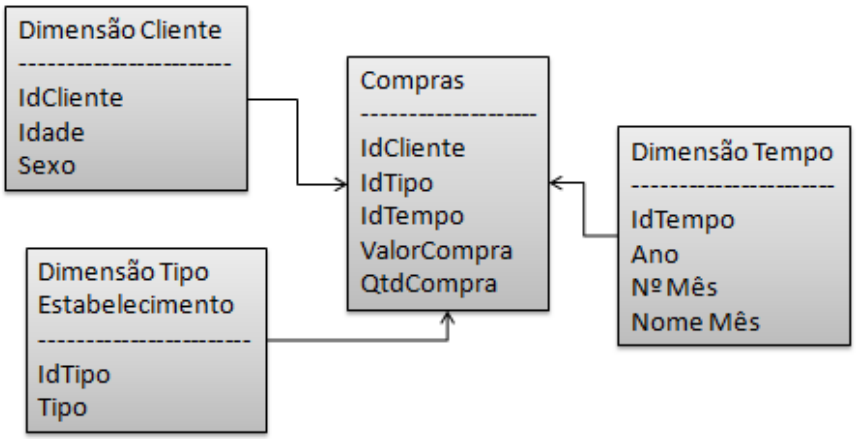


Figura 6 – Diagrama em Estrela da Tabela 1

Através do *Data Cube Vocabulary*, devemos definir as dimensões e atributos da Tabela, conforme podemos ver na Tabela 2:

Dimensão/Atributos	**Representação no *Data Cube Vocabulary***

Dimensão Cliente	`ex:DimCliente rdf:type vocab:tbdimensao;`
	`rdfs:label "Dimensao Cliente"@pt ;`
	`vocab:rdbcorrespondence "TB_DIM_CLIENTE";`
	`skos:inScheme ex:Cube1.`

Atributos	`ex:IdadeCli rdf:type qb:DimensionProperty;`
	`skos:inscheme ex:DimCliente;`
	`rdfs:label "Idade do Cliente"@pt;`
	`qb:concept sdmx-dimension:age.`

	`ex:SexoCli rdf:type qb:DimensionProperty;`
	`skos:inscheme ex:DimCliente;`
	`rdfs:label "Sexo do Cliente"@pt;`
	`qb:concept sdmx-dimension:sex.`

Dimensão Tipo de	`ex:DimTpEstab rdf:type vocab:tbdimensao;`

Estabelecimento	rdfs:label "Dimensao Tipo Estabelecimento"@pt ; vocab:rdbcorrespondence "TB_DIM_TP_ESTAB"; skos:inScheme ex:Cube1.
Atributos	ex:TpEstab rdf:type qb:DimensionProperty; skos:inscheme ex:DimTpEstab; rdfs:label "Tipo do Estabelecimento"@pt.
Dimensão Tempo	ex:DimTempo rdf:type vocab:tbdimensao; rdfs:label "Dimensao Tempo"@pt ; vocab:rdbcorrespondence "TB_DIM_TEMPO"; skos:inScheme ex:Cube1.
Atributos	ex:NuAno rdf:type qb:DimensionProperty; skos:inscheme ex:DimTempo; rdfs:label "Ano da Compra"@pt; qb:concept sdmx-dimension:timeperiod.
	ex:NuMes rdf:type qb:DimensionProperty; skos:inscheme ex:DimTempo; rdfs:label "Mes da Compra"@pt; qb:concept sdmx-dimension:timeperiod.
	ex:NomeMes rdf:type qb:DimensionProperty; skos:inscheme ex:DimTempo; rdfs:label "Nome do Mes da Compra"@pt; qb:concept sdmx-dimension:timePeriod.

Tabela 2 – Representação das Dimensões e Atributos da Tabela com o Data Cube Vocabulary

Devemos definir também as métricas da Tabela, conforme podemos ver na Tabela 3:

Métricas	**Representação no *Data Cube Vocabulary***
Valor Compra	ex:ValCompra rdf:type qb:MeasureProperty; skos:inscheme ex:Cube1; rdfs:label "Valor Total das Compras"@pt; rdfs:range xsd:double.
Qtd. Compra	ex:QtdCompra rdf:type qb:MeasureProperty; skos:inscheme ex:Cube1; rdfs:label "Quantidade Total de Compras

	Realizadas"@pt; rdfs:range xsd:integer.

Tabela 3 – Representação das Métricas da Tabela com o Data Cube Vocabulary

Por fim, devemos definir as observações da Tabela, conforme podemos ver, em parte, na Tabela 4:

Representação das Observações com o *Data Cube Vocabulary*
ex:obs1a a qb:Observation ; qb:dataSet ex:cube1 ; ex:NuAno dbpedia:2010; ex:NomeMes dbpedia:Julho; ex:SexoCli dbpedia:Masculino; ex:TpEstab dbpedia:Teatro; qb:unitMeasure dbpedia:Money ; ex:ValCompra '200,00'^^xsd:double .
ex:obs1b a qb:Observation ; qb:dataSet ex:cube1 ; ex:NuAno dbpedia:2010; ex:NomeMes dbpedia:Julho; ex:SexoCli dbpedia:Masculino; ex:TpEstab dbpedia:Theater; qb:unitMeasure dbpedia:quantity ; ex:QtdCompra '3'^^xsd:integer

Tabela 4 – Representação das Observações da Tabela com o Data Cube Vocabulary

O *Data Cube Vocabulary* tem se mostrado, de fato, o padrão adotado para descrever cubos de dados, pois fornece grande parte dos conceitos necessários para realizar o mapeamentos das informações contidas em um modelo multidimensional. No capítulo seguinte discutiremos alguns dos principais trabalhos relacionados que se baseiam na utilização do mesmo.

2.8
Resumo Capítulo 2

Neste capítulo discutimos os fundamentos teóricos que serviram de base para o desenvolvimento desta dissertação como: o surgimento da Web Semântica, o padrão RDF, os princípios do *Linked Data*, os Vocabulários Padrão adotados nos processos de triplificação e o *Data Cube Vocabulary*, muito utilizado para descrever cubos de dados, dentre outros.

3
Trabalhos Relacionados

Existem várias ferramentas de conversão de diferentes fontes de dados em RDF. As principais ferramentas disponíveis se concentram na conversão bancos de dados relacionais para o formato RDF (Auer, et al., 2008), (Seaborne, et al., 2004) e (Salas, 2011). Outras ferramentas permitem a conversão de dados disponíveis em cubos (Salas, et al., 2012), porém poucas ferramentas permitem a conversão de dados multidimensionais disponíveis em planilhas, para o formato RDF (Salas, et al., 2012), (Zaveri, 2010) e (Langegger, 2009).

A conversão no sentido inverso, ou seja, do RDF para outras fontes de dados (banco relacional, OLAP ou planilhas) ainda se encontra muito pouco explorada (Kämpgen, et al., 2011).

3.1
Ferramentas de conversão de um Formato Específico para RDF

Nesta subseção iremos tratar das ferramentas que realizam a conversão de um determinado formato de dados para o formato RDF. Ela será subdividida em ferramentas que convertem dados disponíveis: em bancos de dados relacionais, bancos de dados multidimensionais, e planilhas MS Excel.

3.1.1
Ferramentas de conversão de Bancos de Dados relacionais para RDF

Nesta subseção apresentamos algumas das ferramentas disponíveis para conversão de dados armazenados em Bancos de Dados relacionais para o formato de triplas RDF.

3.1.1.1
Triplify

Triplify (Auer, et al., 2008) foi desenvolvido para tornar simples a conversão e publicação de informações contidas em bancos de dados relacionais – RDB – em RDF. Para realizar a conversão, é preciso definir um mapeamento que envolve consultas SQL onde as tabelas são mapeadas para classes e as colunas para atributos. Após a conversão, os dados mapeados podem ser acessados através de requisições HTTP. O Triplify possui um bom desempenho e simplicidade de uso em comparação às demais ferramentas e por isso é largamente utilizada na geração de conteúdo em RDF e Linked Data.

3.1.1.2
D2RQ

D2RQ (Seaborne, et al., 2004) é uma plataforma que permite tratar bancos de dados relacionais não nativos como um grafo RDF virtual. Através de uma linguagem declarativa, o D2R Mapping Language[31], é possível definir mapeamentos que possibilitam a criação de RDF-views, que permitem que estes bancos possam ser acessados como RDF e Linked Data através da Web.

A plataforma D2RQ acompanha um plugin, D2RQ Engine, que reescreve as APIs Jena e Sesame de consulta SQL, possibilitando que um banco de dados relacional possa ser acessado e convertido para RDF. Além disso, a plataforma ainda oferece um servidor HTTP – D2R Server[32] – que provê um SPARQL enpoint para realizar consultas.

3.1.1.3
StdTrip

StdTrip (Salas, 2011) é uma ferramenta que propõe facilitar o processo de decisão da representação de esquemas de bancos de dados relacionais em classes e propriedades do vocabulário RDF.

[31] http://www4.wiwiss.fu-berlin.de/bizer/d2rmap/D2Rmap.htm

[32] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

O principal objetivo desta ferramenta é o reuso de vocabulários de forma a assegurar a interoperabilidade dentro do espaço da *Linked Open Data (*LOD).

O processo de triplificação da ferramenta, ilustrado visto na Figura 7, segue a seguinte ordem: conversão, alinhamento, seleção, inclusão, conclusão e saída.

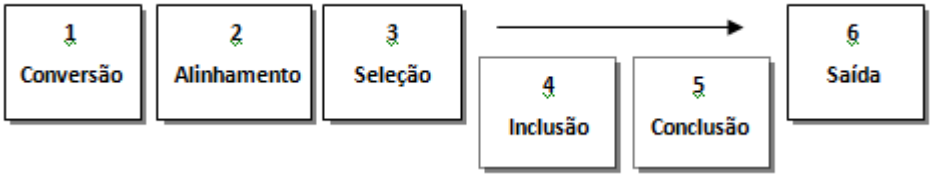


Figura 7 – Arquitetura da Ferramenta StdTrid (Salas, 2011)

3.1.2

Ferramenta de conversão de Bancos de Dados multidimensionais para RDF - OLAP2DataCube

Nesta subseção apresentamos uma ferramenta disponível para conversão de dados armazenados em Bancos de Dados Multidimensionais para o formato de triplas RDF.

OLAP2DataCube (Salas, et al., 2012) é uma ferramenta que permite que grandes bases de dados analíticos, representadas de acordo com o paradigma *Online Analytical Processing* (OLAP), possam ser eficientemente transformados para o formato RDF.

A ferramenta foi desenvolvida como um plug-in para o OntoWiki (Auer, et al., 2006), que suporta a criação colaborativa, manutenção e publicação de bases de conhecimento no formato RDF. A ferramenta também oferece várias interfaces para publicar e consultar dados.

Um banco de dados relacional, representado de acordo com o modelo em estrela (Thomsen, 1997), serve de entrada para a ferramenta OLAP2DataCube que converte esses dados para o formato de triplas RDF, através do DataCubeVocabulary. O processo abrange três fases: (1) extração dos metadados do banco de dados e categorização das tabelas, (2) definição do cubo, e (3) mapeamento para RDF.

3.1.3
Ferramentas de conversão de planilhas para RDF

Nesta subseção apresentamos algumas das ferramentas disponíveis para conversão de dados disponíveis em planilhas para o formato de triplas RDF.

3.1.3.1
CSV2DataCube e Stats2RDF

CSV2DataCube (Salas, et al., 2012) e Stats2RDF (Zaveri, 2010) são ferramentas que possibilitam que dados estatísticos disponíveis no formato CSV possam ser convertidos em RDF. Ambas as ferramentas foram desenvolvidas como plug-ins para o OntoWiki, como a ferramenta OLAP2DataCube.

Dados estatísticos, muitas vezes, também são representados através de planilhas. As ferramentas CSV2DataCube e Stats2RDF permitem a transformação semiautomática de planilhas para o formato RDF.

Quando uma planilha contendo dados estatísticos multidimensionais é importada para a ferramenta OntoWiki, ela passa a ser apresentada em forma de tabela. Esta apresentação dos dados permite que os usuários configurem: as dimensões e seus atributos, selecionando todos os elementos pertencentes a uma dimensão e as medidas dos itens estatísticos. Após a configuração, o plug-in faz a conversão das planilhas para o formato RDF utilizando o *Data Cube Vocabulary*.

3.1.3.2
XLWrap

XLWrap (Langegger, 2009) é uma ferramenta capaz de transformar planilhas para o formato de triplas RDF baseando-se em um mapeamento especificado pelo usuário. A ferramenta suporta a conversão de dados disponíveis em planilhas MS Excel e planilhas do OpenDocument, assim como dados disponíveis em arquivos no formato *Comma Separated Value* (CSV).

A ferramenta não permite que dados multidimensionais sejam convertidos para o formato RDF. O mapeamento das colunas leva em

consideração apenas o vocabulário SCOVO[33], não sendo possível reaproveitar outros vocabulários disponíveis para representar os dados.

3.2 Ferramenta de conversão de RDF para Modelos Multidimensionais

A ferramenta *Transforming Statistical Linked Data for Use in OLAP Systems* (Kämpgen, et al., 2011) se propõe a converter dados estatísticos disponíveis no formato de triplas RDF, descritos utilizando-se o *Data Cube Vocabulary*, para o formato OLAP, chamado de *Multidimensional Model* – Modelo Multidimensional - (MDM).

A Figura 8 mostra a arquitetura da ferramenta, que implementa o seguinte processo: escolha do *dataset* (arquivos no formato RDF) que serão convertidos; construção da consulta SPARQL para extrair os dados do arquivo RDF; a partir dos dados extraídos pela consulta, criação do Modelo Multidimensional (MDM); serialização do Modelo Multidimensional para o formato *XML for Analysis* (XMLA), que implementa a *Multidimensional Query Language*[34] (MDX), que permite operações OLAP comuns.

[33] http://vocab.deri.ie/scovo/

[34] http://msdn.microsoft.com/en-us/library/Aa216767

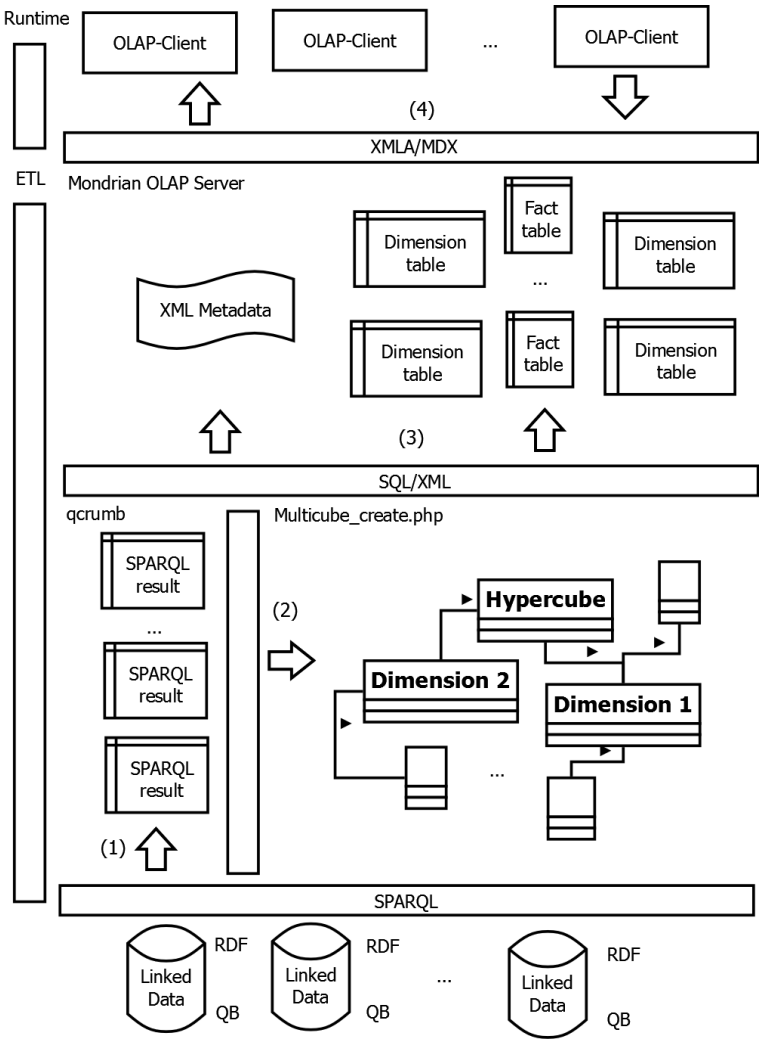


Figura 8 – Arquitetura da ferramenta de conversão do formato RDF para MDM [17]

No capítulo seguinte iremos apresentar o framework *OLAP2DataCube Catalog On Demand*, descrevendo em mais detalhes sua arquitetura, suas camadas, módulos (*Client Application*, *Catalog*, *Mediator* e *Wrapper*) e a forma como elas se comunicam. Iremos detalhar também o processo de conversão de dados por meio do Framework *OLAP2DataCube Catalog On Demand*, que se divide em três etapas (*Search and Choose*, *Production and Request* e *Transform and Respond).*

3.3
Resumo Capítulo 3

Neste capítulo discutimos as principais ferramentas disponíveis para conversão de dados disponíveis em diversos formatos (bancos de dados relacionais, bancos de dados multidimensionais e planilhas) para o padrão RDF. Também discutimos uma ferramenta que realiza a conversão de dados no formato RDF para o formato de Modelos Multidimensionais.

4
O Framework *OLAP2DataCube Catalog On Demand*

Com o objetivo de possibilitar com que dados estatísticos armazenados em um catálogo sejam filtrados e visualizados, através de transformações do tipo *slice and dice* em cubos de dados, criamos o Framework *OLAP2DataCube Catalog On Demand*. Este é composto dos seguintes módulos: *Client Application* (*Application Code*), *Mediator* (*Integrated Web Data*), *Catalog* (Web *Data Acces Module*) e *Wrapper*. O framework proposto permite a visualização de porções de cubos de dados, disponíveis em um catálogo, de acordo com as dimensões de interesse do usuário.

O framework *OLAP2DataCube Catalog On Demand* foi inspirado na arquitetura do padrão *Crawling Data* (Heath, et al., 2011), que se baseia na clássica arquitetura de motores de busca da Web como Google e Yahoo. A Tabela 5 fornece uma visão comparativa entre as duas arquiteturas.

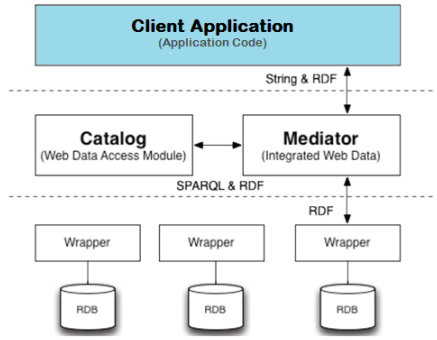
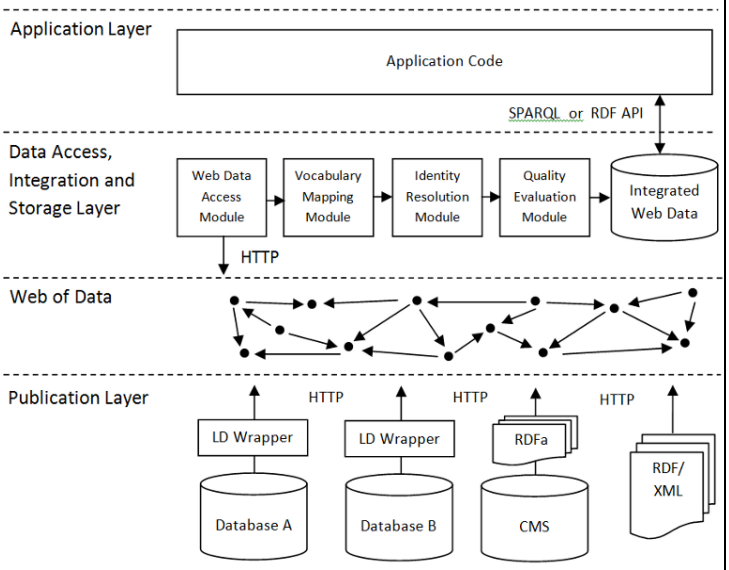
	Arquitetura do Framework proposto		Arquitetura do Padrão Crawling Data
a)		b)	

Tabela 5 – Visão comparativa das arquiteturas do Framework *OLAP2DataCube Catalog On Demand* e do Padrão *Crawling Data* (Heath, et al., 2011)

Analisando as duas arquiteturas, vemos que a principal diferença está na ausência da camada *Web of Data* na arquitetura do Framework *OLAP2DataCube Catalog On Demand*. O padrão *Crawling Data* busca os dados no formato de triplas RDF na Web de Dados, enquanto que o framework proposto recupera os dados no formato RDF diretamente dos bancos de dados (RDB), através dos seus respectivos *Wrappers*. No framework proposto, o módulo *Catalog*, que representa o módulo W*eb Data Access Module*, possui os metadados de todos os cubos armazenados, para tornar possível a conexão do *Wrapper* com o banco de dados relacional.

A grande vantagem do framework proposto, em relação aos já existentes, que também se baseiam no padrão *Crawling Data*, é o foco em mitigar um problema comum em aplicações deste tipo, a replicação dos dados. Seguindo a abordagem *on demand*, em que a integração dos dados ligados é realizada de forma evolutiva e realizando a conversão dos dados em tempo real de acordo com a necessidade, o problema de replicação pode ser contornado. Para isto, a abordagem utiliza a publicação dos dados em formatos que permitem sua

reutilização, e cria uma nova ontologia a partir de algumas outras já conhecidas e amplamente utilizadas.

Podemos ver também que o Padrão *Crawling Data* possui três módulos na camada *Data Access Integration and Storage Layer* que o Framework *OLAP2DataCube Catalog On Demand* não implementa: *Vocabulary Mapping Module* (responsável por realizar o mapeamento dos termos encontrados para os vocabulários padrões); *Identity Resolution Module* (responsável por padronizar as URI's)*; Quality Evaluation Module* (responsável por garantir a qualidade das informações buscadas).

Os módulos *Identity Resolution* e *Quality Evaluation* não são necessários no framework proposto, uma vez que a montagem das URI's já vem padronizada pelo processo de triplificação implementado pelo *Wrapper*. A qualidade da informação já está garantida, pois os dados foram recuperados diretamente dos bancos de dados relacionais selecionados pelo usuário.

O módulo *Vocabulary Mapping* não foi implementado nesta versão do Framework *OLAP2DataCube Catalog On Demand*, porém, futuramente, desejamos desenvolvê-lo e incorporá-lo ao Framework.

4.1
Módulos do Framework *OLAP2DataCube Catalog On Demand*

Nesta subseção iremos detalhar os quatro módulos que compõe o Framework *OLAP2DataCube Catalog On Demand*: *Client Application* (*Application Code*), *Mediator* (*Integrated Web Data*), *Catalog* (Web *Data Acces Module*) e *Wrapper.*

4.1.1
Client Application

Com o objetivo de fornecer uma interface de consulta com o usuário, o módulo *Client Application* se comunica com o *Mediator* para recuperar as informações dos cubos de dados disponíveis, e recuperar os dados disponíveis do cubo escolhido.

As principais funcionalidades dessa camada são:

- ✓ Permitir a escolha do cubo desejado pelo usuário;

- ✓ Permitir operações de *slice and dice* no modelo do cubo escolhido;
- ✓ Permitir operações de *drill up* e *drill down* nos dados apresentados;
- ✓ Exportar para a ferramenta MS Excel os dados filtrados e formatados.

4.1.2 Catalog

O módulo *Catalog* consiste de um catálogo que armazena os cubos disponíveis para manipulações do usuário, e fornece ao *Mediator* as informações necessárias para que as observações solicitadas sejam retornadas.

Para que seja possível o mapeamento entre os cubos e os seus respectivos RDB's, o catálogo contém metadados tanto dos cubos de dados, quanto dos RDB's onde podem ser encontradas observações sobre cada cubo. Individualmente estes metadados incluem informações do mapeamento entre os dois formatos, tais como a tabela-fato e as tabelas-dimensão relacionadas à cada cubo, dados de conexão com o RDB, entre outros.

4.1.3 Mediator

O *Mediator* é o módulo que faz o intermédio das trocas de informações entre os dois módulos descritos anteriormente – *Catalog* e *Client Application* – de modo a permitir com que os cubos disponíveis no catálogo sejam consumidos pelo usuário.

As principais funcionalidades dessa camada são:

- ✓ Realizar a comunicação do módulo *Client Application* com o módulo *Catalog*, buscando no Catálogo os cubos que atendem à consulta do usuário;
- ✓ Realizar a comunicação do módulo *Client Application* com o *Wrapper*, retornando o arquivo no formato RDF.

4.1.4 Wrapper

O *Wrapper* é o módulo responsável por realizar o processo de triplificação dos dados disponíveis nos RDB's. Eles formam interfaces que se

comportam como uma camada de acesso aos RDB's, sendo que todos os RDB's são acessados somente a partir do seu próprio *Wrapper*.

4.2
Etapas do processo de consumo de dados do Framework *OLAP2DataCube Catalog On Demand*

O processo de consumo dos cubos, que se inicia com a entrada das palavras-chave pelo usuário, e termina com a exibição dos dados retornados contendo as observações do cubo selecionado, pode ser dividido em três etapas: *Search and Choose*, *Production and Request, Transform ans Responds*, detalhadas a seguir.

4.2.1
Etapa 1: Search and Choose

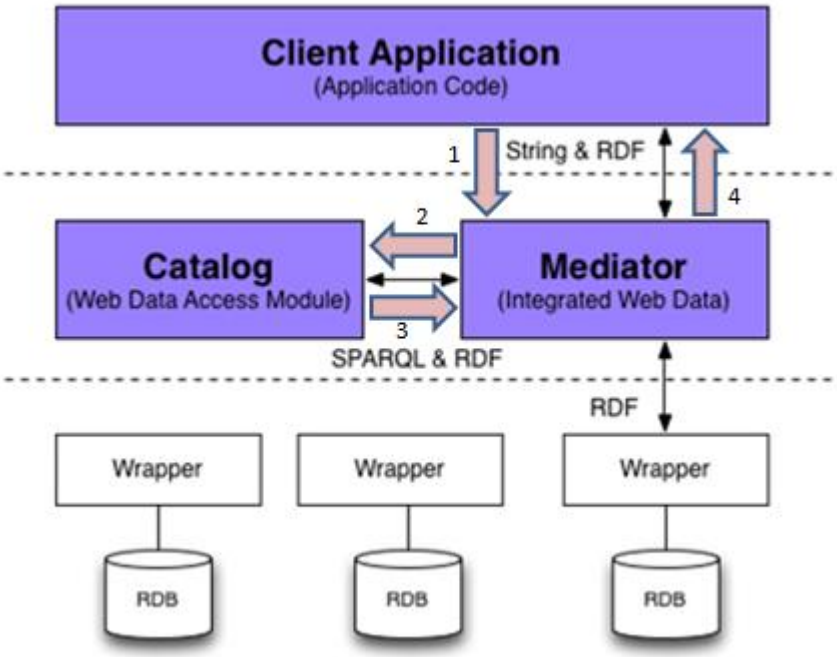


Figura 9 – Etapa *Search and Choose* do Framework *OLAP2DataCube Catalog On Demand*

Essa etapa, ilustrada pela Figura 9, se inicia com o usuário informando a palavra-chave desejada. Esta informação é repassada para o *Mediator* (passo 1), que por sua vez constrói uma consulta em SPARQL e executa esta consulta

sobre o *Catalog* (passo 2), com objetivo de filtrar somente os cubos compatíveis com a palavra-chave informada.

O *Catalog*, então, devolve ao *Mediator* o arquivo com os cubos que atendem à consulta, no formato de triplas RDF (passo 3).

O *Mediator* repassa o arquivo RDF com os cubos para o usuário (passo 4), para que ele possa selecionar apenas um deles para então manipulá-lo selecionando as dimensões e métricas desejadas, através de transformações do tipo *slice*.

4.2.2
Etapa 2: Production and Request

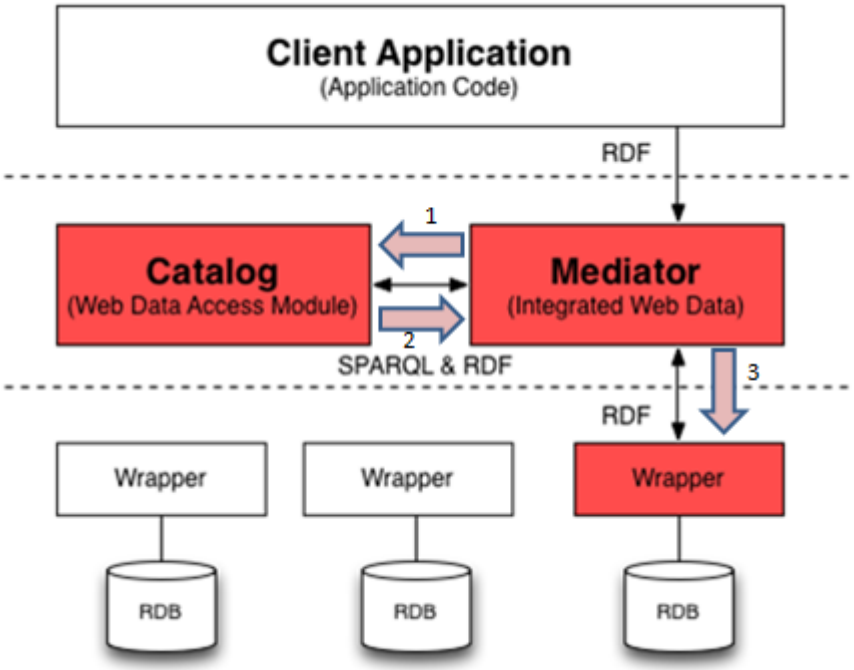


Figura 10 – Etapa *Production and Request* do Framework *OLAP2DataCube Catalog On Demand*

Essa etapa, ilustrada pela Figura 10, se inicia após o *Client Application* devolver o cubo transformado de volta para o *Mediator*, este por sua vez solicita (passo 1) ao *Catalog* dados relativos à conexão com os RDB's selecionados. Estes dados incluem: *hostname*, porta, *database name*, entre outros (passo 2).

O *Mediator*, a partir dos dados de conexão fornecidos pelo *Catalog*, se comunica com o *Wrapper* do respectivo banco de dados relacional, do qual vão ser buscadas as observações (passo 3).

4.2.3
Etapa 3: Transform and Respond

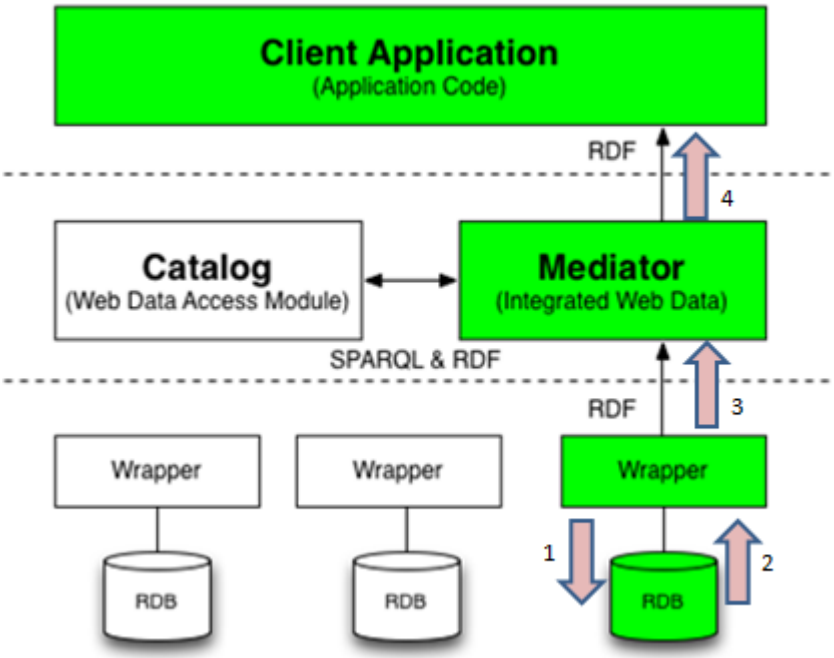


Figura 11 – Etapa *Transform and Respond* do Framework *OLAP2DataCube Catalog On Demand*

A última etapa do processo de consumo dos cubos, ilustrada na Figura 11, envolve as camadas de aplicação e mediação, além dos *Wrappers* e os seus bancos de dados relacionais referentes aos cubos selecionados. A mesma se inicia com a resposta do RDB ao *Wrapper*, referente à solicitação da etapa anterior (passo 1), e objetiva concretizar as visões RDF dos dados relacionais solicitados pelo usuário (passo 2).

Essas visões são repassadas do *Wrapper* para o *Mediator* (passo 3), no formato de arquivo RDF, que por sua vez o repassa para o *Client Application* (passo 4).

A partir do arquivo RDF recebido, o *Client Application* exibe os dados retornados para o usuário, permitindo que este realize operações de *slice and*

dice no nível de dados (restringindo e reorganizando os dados das linhas e colunas, respectivamente), e operações de *drill up* e *drill down* (diminuindo e aumentando o nível de detalhamento dos dados, respectivamente). O *Client Application* também permite que os dados sejam filtrados, e ordenados da maneira desejada.

Por fim, o usuário poderá exportar os dados resultantes para uma planilha MS Excel.

No capítulo seguinte iremos apresentar o rdXel, uma ferramenta desenvolvida para implementar o módulo *Client Application* do Framework *OLAP2DataCube Catalog On Demand* proposto. O framework é capaz de converter os arquivos de triplas RDF para o formato MS Excel. Também realiza a conversão no sentido inverso, do formato MS Excel para o de triplas RDF.

4.3
Resumo Capítulo 4

Neste capítulo apresentamos em mais detalhes o Framework *OLAP2DataCube Catalog On Demand* proposto, comparando-o com o Padrão *Crawling Data*, muito utilizado em motores de busca clássicos da Web, tais como Google e Yahoo.

5
RdXel

O RdXel foi concebido para detalhar, especificar e desenvolver a camada de Aplicação (*Client Application*) do Framework *OLAP2DataCube Catalog On Demand*. O aplicativo de software escolhido como plataforma para o desenvolvimento para esta solução foi o MS Excel, na forma de plug-in (Pesce, et al., 2012).

O processo da camada *Client Application*, que chamamos nesta dissertação de RDF2Excel, se resume em recuperar os dados e metadados dos cubos disponíveis no Catálogo, exibi-los para o usuário, de modo a permitir as operações OLAP tradicionais, i.e., *slice and dice*, *drill up*, *drill down*, filtros e ordenações (Thomsen, 1997).

A ferramenta RdXel também é capaz de realizar o processo de conversão de dados no sentindo inverso, ou seja, de ler os dados disponíveis em uma planilha MS Excel e convertê-los para o formato de triplas RDF, tendo como base o *Data Cube Vocabulary*. A este processo de conversão demos o nome, nesta dissertação, de Excel2RDF. A implementação desse processo inverso é importante pois permite que os dados disponíveis em planilhas MS Excel sejam cruzados com os dados disponíveis nos cubos.

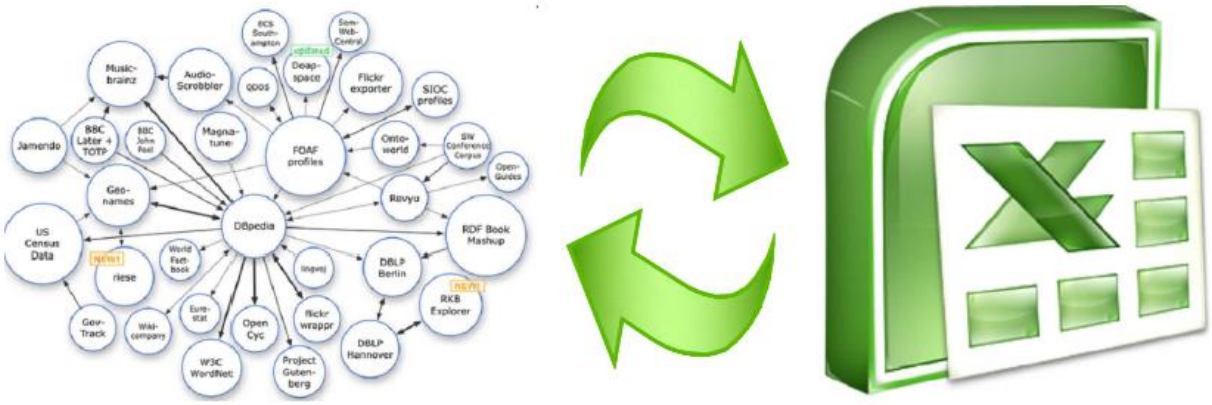


Figura 12 – Funcionalidades da Ferramenta rdXel

A Figura 12 representa de modo abstrato o que a ferramenta rdXel se propõe a fazer, i.e., a conversão de dados RDF para o formato MS Excel e, de forma inversa, a conversão de dados disponíveis em planilhas MS Excel para o formato de triplas RDF.

É importante notar que o arquivo RDF de saída do processo de triplificação Excel2RDF não é exatamente simétrico ao que a ferramenta consome no processo RDF2Excel.

Algumas diferenças, principalmente na formatação da informação, são inevitáveis neste processo, pois, durante o processo de conversão Excel2RDF a ferramenta não dispõe de algumas informações necessárias para garantir a simetria dos dois processos.

A Tabela 6, a seguir, sumariza e exemplifica as diferenças entre o arquivo RDF de saída do processo Excel2RDF e o arquivo RDF de entrada do processo RDF2Excel.

RDF de Saída (Excel2RDF)	RDF de Entrada (RDF2Excel)
	vocab:rdbcorrespondence rdf:type owl:datatypeproperty; rdf:label "Propriedade que une a dimensao com a tab do BD relacional"; rdfs:domain vocab:tbdimensao; rdfs:range xsd:literal.
Esse bloco, que define o vocabulário para correspondência, não é gerado no processo Excel2RDF pois ele não possui informações sobre o banco de bados.	
ex:DimTipoEstabelecimento rdf:type vocab:tbdimensao; rdfs:label "Dimensao TipoEstabelecimento"@pt ; skos:inScheme ex:Cube1.	ex:DimTpEstab rdf:type vocab:tbdimensao; rdfs:label "Dimensao Tipo Estabelecimento"@pt ; **vocab:rdbcorrespondence "TB_DIM_TP_ESTAB";** skos:inScheme ex:Cube1.
Essa linha, que define a correspondência da Dimensão com a tabela no banco de dados, não é gerado no processo Excel2RDF pois ele não possui informações sobre o banco de dados.	
ex:DimSexo rdf:type vocab:tbdimensao; rdfs:label "Dimensao Sexo"@pt ; skos:inScheme ex:Cube1.	#===================================== # Declaracao dos atributos da DimCliente #===================================== ex:SexoCli rdf:type qb:DimensionProperty; skos:inscheme ex:DimCliente; rdfs:label "Sexo do Cliente"@pt;

<table>
<tr><td></td><td>qb:concept sdmx-dimension:sex.</td></tr>
<tr><td colspan="2">Enquanto no processo RDF2Excel a informação ‘Sexo Cliente’ é representada como um atributo da dimensão cliente, no processo Excel2RDF essa mesma informação é representada como uma Dimensão, pois a ferramenta distingue dimensões de atributos.</td></tr>
<tr><td>ex:Quantidade rdf:type
qb:MeasureProperty;
skos:inscheme ex:Cube1;
rdfs:label "Quantidade"@pt;
rdfs:range xsd:double.</td><td>ex:QtdCompra rdf:type qb:MeasureProperty;
skos:inscheme ex:Cube1;
rdfs:label "Quantidade Total de Compras Realizadas"@pt;
rdfs:range xsd:integer.</td></tr>
<tr><td colspan="2">O processo RDF2Excel distingue o tipo exato do dado (integer, double, string, date, char), pois o banco de dados possui essa informação. No entanto, o processo Excel2RDF apenas diferencia os tipos (double, string e date).</td></tr>
<tr><td>ex:obs1a a qb:Observation ;
qb:dataSet ex:cube1 ;
ex:DimAno dbpedia:2010;
ex:DimMes dbpedia:Julho;
ex:DimSexo dbpedia:Masculino;
ex:DimTipoEstabelecimento dbpedia:Teatro;
qb:unitMeasure
ex:Reais '200,00'^^xsd:double .</td><td>ex:obs1a a qb:Observation ;
qb:dataSet ex:cube1 ;
ex:NuAno dbpedia:2010;
ex:NomeMes dbpedia:Julho;
ex:SexoCli dbpedia:Masculino;
ex:TpEstab dbpedia:Teatro;
qb:unitMeasure dbpedia:Money ;
ex:valCompra '200,00'^^xsd:double .</td></tr>
<tr><td colspan="2">O processo RDF2Excel detalha o tipo do dado da Métrica (Money), pois o banco de dados possui essa informação. No entanto, o processo Excel2RDF não possui essa informação.</td></tr>
</table>

Tabela 6 - Diferenças entre o arquivo RDF de saída do processo Excel2RDF e o arquivo RDF de entrada do processo RDF2Excel.

As subseções a seguir irão detalhar o processo RDF2Excel e Excel2RDF, bem como a arquitetura adotada na implementação de cada um. Para exemplificar a utilização da ferramenta, apresentamos um cubo que representa um Demonstrativo de Compras por Tipo de Estabelecimento, Tipo de Transação, Tempo e Características do Cliente.

5.1 RDF2Excel

Este módulo da ferramenta realiza a conversão de dados do formato de triplas RDF para o formato de planilha MS Excel. Para possibilitar a leitura do

RDF, foi definido um modelo específico de arquivo, baseado no *Data Cube Vocabulary*, presente nos exemplos deste trabalho.

Para realizar a conversão citada acima, é necessário que a ferramenta se "conecte" ao Mediador, que irá fornecer os arquivos RDF necessários para a conversão dos dados. A ferramenta somente é capaz de ler os arquivos RDF fornecidos pelo Mediador, e que seguem o padrão RDF pré-estabelecido para troca de dados.

5.1.1 Arquitetura

A arquitetura elaborada para a ferramenta RdXel, conforme vemos na Figura 13, se divide em cinco componentes: Busca e Escolha dos Cubos, Leitura e Interpretação do RDF Cubos, Seleção das Dimensões e Métricas, Leitura e Interpretação do RDF Observações e Output – Planilha MS Excel. O Mediador está fora do escopo desta dissertação e, portanto, não será detalhado.

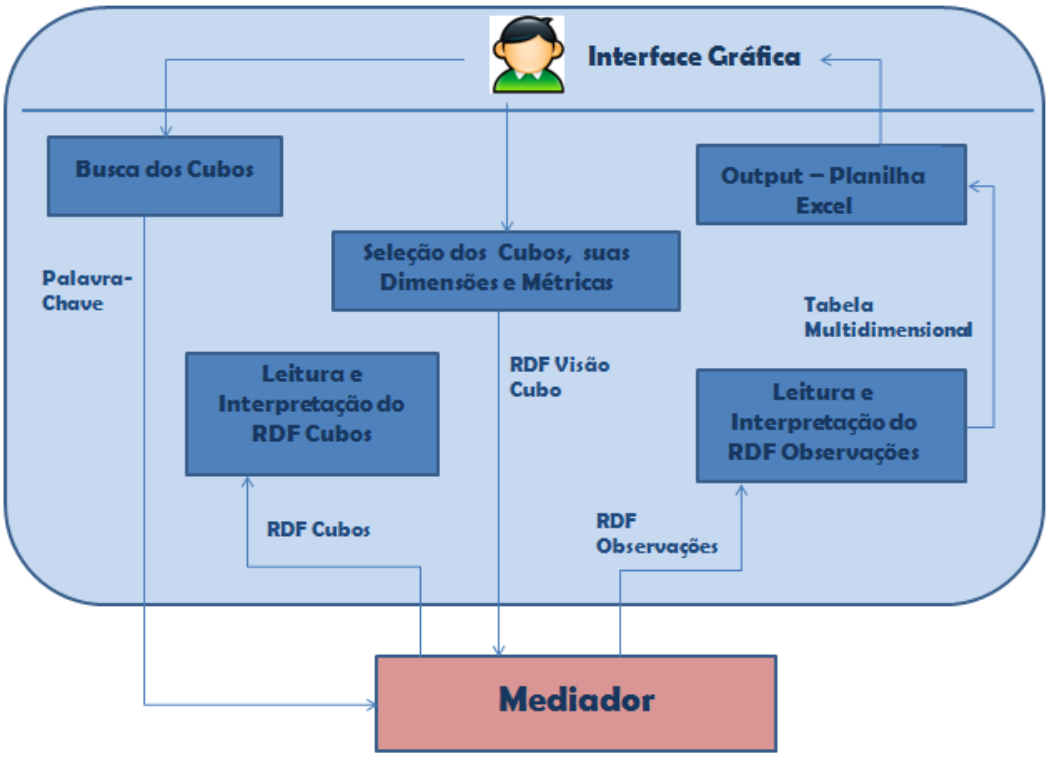


Figura 13 – Arquitetura da Ferramenta RDF2Excel

1. **Busca e Escolha dos Cubos**

Este componente é responsável por chamar o método `BuscarCubos` do Mediador, passando como parâmetro as palavras-chave informadas pelo usuário na Interface Gráfica (IE). O método `BuscarCubos` irá retornar um arquivo RDF contendo os cubos que satisfazem a pesquisa feita pelo usuário.

2. **Leitura e Interpretação do RDF Cubos**

Este componente é responsável por ler e interpretar o arquivo RDF Cubos retornado pelo Mediador. Primeiramente ele busca os cubos disponíveis e apresenta uma lista dos cubos na IE, para escolha do usuário. As dimensões e atributos do cubo escolhido serão dispostos em uma lista de dimensões, enquanto as suas métricas serão dispostas em uma lista de métricas.

3. **Seleção das Dimensões e Métricas**

Este componente é responsável por apresentar as listas contendo os atributos das dimensões e as métricas do cubo escolhido na IE. Ele também é responsável por montar o arquivo RDF com as escolhas (atributos e métricas) realizadas pelo usuário, bem como repassá-las para o Mediador, chamando o método `BuscarObservacoesCubo`.

4. **Leitura e Interpretação do RDF observações**

Este componente é responsável por ler e interpretar o arquivo RDF contendo as observações retornadas pelo Mediador. Conforme o componente realiza a leitura do arquivo, ele monta dinamicamente uma tabela multidimensional, que será apresentada ao usuário.

5. **Output – Planilha MS Excel**

Este componente é responsável por apresentar a tabela multidimensional para o usuário, permitindo que ele realize operações como *slice and dice, drill up, drill down,* filtros, e ordenações sobre a tabela. Ele também possibilita a exportação dos dados trabalhados pelo usuário para sua planilha MS Excel.

5.1.2
Processo de conversão de dados RDF para planilha MS Excel

O processo completo de conversão de dados RDF para planilhas MS Excel, que se inicia com a palavra-chave fornecida pelo usuário, e termina com os dados sendo exportados para o MS Excel, está representado na Figura 14. Será descrito em mais detalhes a seguir.

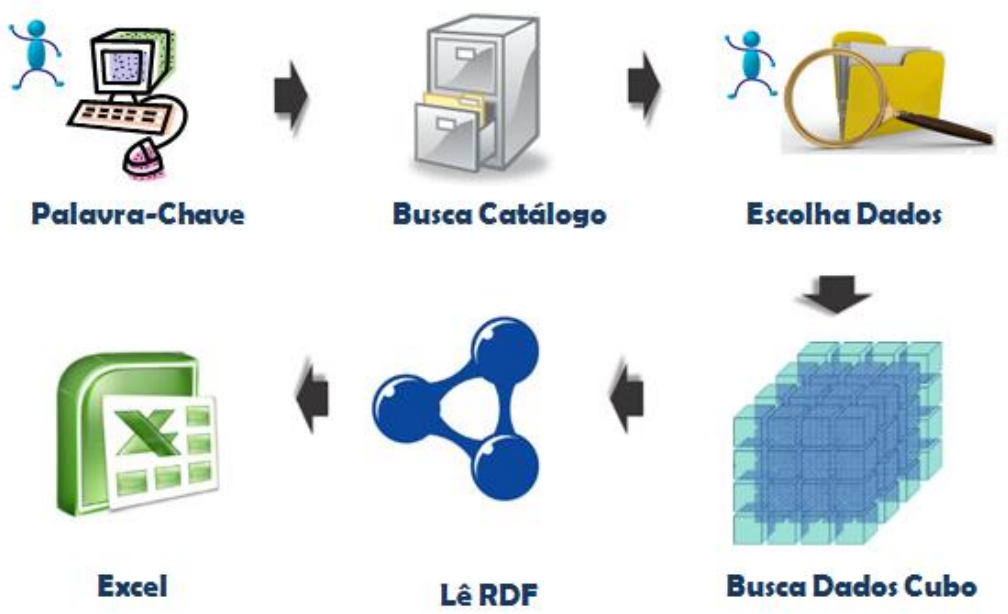


Figura 14 – Processo de conversão RDF2Excel

O processo se inicia com o usuário informando as palavras-chave que deseja buscar. A Figura 15 mostra a interface gráfica apresentada ao usuário nesse momento. No exemplo citado, o usuário pode preencher o campo com a palavra ‘Compras’ e clicar em ‘Buscar Cubos’.

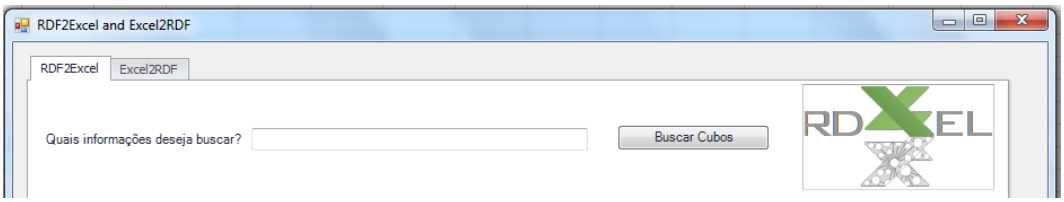


Figura 15 – Usuário informa as palavras-chave que deseja buscar

A ferramenta então chama um método do Mediador, passando como parâmetro as palavras-chave informadas pelo usuário. O Mediador, por sua vez, retorna um arquivo no formato RDF com os metadados dos cubos,

dimensões, atributos e métricas. A Figura 16, a seguir, exemplifica as informações retornadas.

```
@prefix rdf:            <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
.
@prefix rdfs:           <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:            <http://www.w3.org/2002/07/owl#> .
@prefix xsd:            <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:           <http://www.w3.org/2004/02/skos/core#> .
@prefix qb:             <http://purl.org/linked-data/cube#> .
@prefix dcterms:        <http://purl.org/dc/terms/> .
@prefix vocab:          <http://www.inf.puc-rio.br/vocab#>
#================================================================
# Definicões
#================================================================
vocab:tbdimensao rdf:type owl:class;
  rdf:label "Tabela de Dimensao no modelo OLAP".

vocab:rdbcorrespondence rdf:type  owl:datatypeproperty;
   rdf:label    "Propriedade que une a dimensao com a tab do BD
relacional";
   rdfs:domain   vocab:tbdimensao;
   rdfs:range    xsd:literal.
#================================================================
# Declaracao do Modelo Dimensional
#================================================================
#=======================================
# Declaracao do Cubo
# ex:cube1 e igual à ex:dataset1 - Percy
#=======================================
ex:cube1 a qb:DataSet;
    qb:structure ex:dsd-cube ;
    rdfs:label "Cubo Demonstrativo de Compras por Tipo de
Estabelecimento, Tipo de Transacao e Caracteristicas do Cliente"@pt.

ex:cube2 a qb:DataSet;
    qb:structure ex:dsd-cube ;
    rdfs:label "Cubo Demonstrativo de Compras por Tipo de
Estabelecimento, Tipo de Transacao e Caracteristicas do Cliente e
Teste"@pt.
#=======================================
# Declaracao das Dimensões
#=======================================
#=======================================
# Declaracao da Dimensao Cliente
#=======================================
ex:DimCliente   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao Cliente"@pt ;
```

```
  vocab:rdbcorrespondence "TB_DIM_CLIENTE";
  skos:inScheme ex:Cube1.
#=======================================
# Declaracao dos atributos da DimCliente
#=======================================
ex:IdadeCli rdf:type qb:DimensionProperty;
  skos:inscheme ex:DimCliente;
  rdfs:label "Idade do Cliente"@pt;
  qb:concept sdmx-dimension:age.

ex:SexoCli rdf:type qb:DimensionProperty;
  skos:inscheme ex:DimCliente;
  rdfs:label "Sexo do Cliente"@pt;
  qb:concept sdmx-dimension:sex.
#================================================
# Declaracao da Dimensao Tipo de Estabelecimento
#================================================
ex:DimTpEstab   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao Tipo Estabelecimento"@pt ;
  vocab:rdbcorrespondence "TB_DIM_TP_ESTAB";
  skos:inScheme ex:Cube1.
#================================================
# Declaracao dos Atributos da DimTpEstab
#================================================
ex:TpEstab rdf:type qb:DimensionProperty;
  skos:inscheme ex:DimTpEstab;
  rdfs:label "Tipo do Estabelecimento"@pt.

ex:CdPorteEstab rdf:type qb:DimensionProperty;
  skos:inscheme ex:DimTpEstab;
  rdfs:label "Porte do Estabelecimento"@pt.
#=======================================
# Declaracao das Metricas
#=======================================
ex:ValCompra rdf:type qb:MeasureProperty;
  skos:inscheme ex:Cube1;
  rdfs:label "Valor Total das Compras"@pt;
  rdfs:range xsd:double.

ex:QtdCompra rdf:type qb:MeasureProperty;
  skos:inscheme ex:Cube1;
  rdfs:label "Quantidade Total de Compras Realizadas"@pt;
  rdfs:range xsd:integer.
```

Figura 16 – Exemplo de arquivo RDF contendo as definições dos cubos

Com base no arquivo RDF retornado, a ferramenta lê o arquivo, linha a linha, buscando os cubos e separando os dados (metadados, dimensões, atributos, métricas e observações) de cada cubo. Os cubos encontrados são apresentados em uma lista, para que o usuário escolha o cubo desejado, e clique em 'Buscar Dimensões e Métricas do Cubo', conforme na Figura 17.

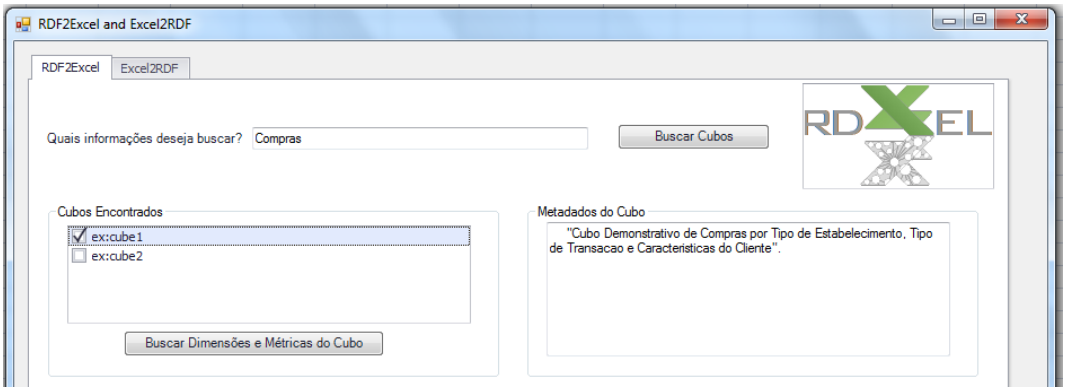


Figura 17 – Usuário escolhe o cubo desejado

Após realizar a busca pelas dimensões e métricas do cubo escolhido, a ferramenta exibe uma lista com todos os atributos das dimensões e uma segunda lista com as métricas do cubo, conforme vemos na Figura 18. O usuário deverá selecionar os atributos das dimensões escolhidas, possibilitando desta forma uma operação de *slice* no cubo retornado. O usuário pode, neste momento, escolher se quer que o atributo seja exibido na horizontal (linha) ou na vertical (coluna), permitindo assim uma operação de *dice* no cubo. Ele deve, também, escolher as métricas que deseja, e clicar em 'Buscar Dados'.

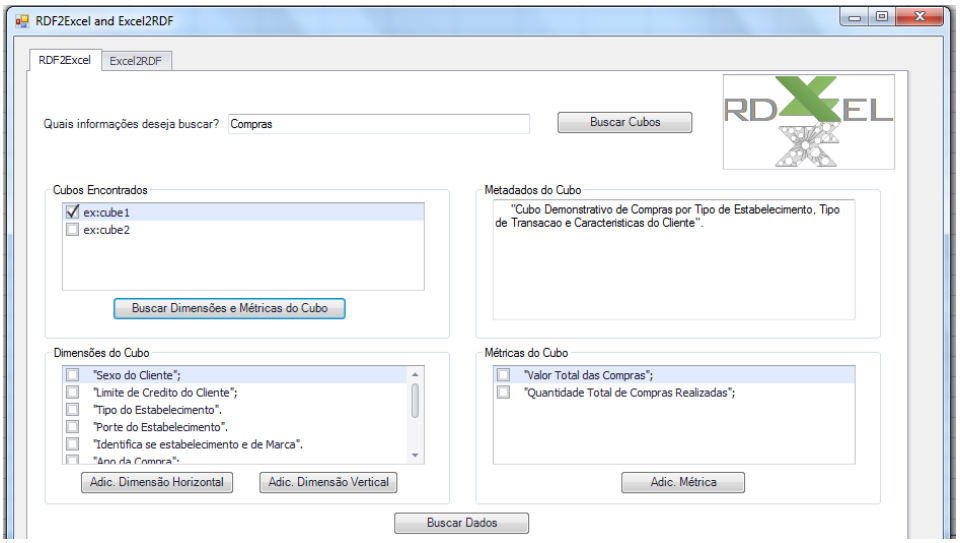


Figura 18 – Usuário escolhe atributos e métricas do cubo

Com base no cubo, atributos e métricas selecionadas, a ferramenta monta um arquivo RDF, conforme podemos observar na Figura 19. A ferramenta então chama um método do Mediador, passando como parâmetro o arquivo montado. O Mediador retorna para a ferramenta um arquivo no formato RDF com os dados solicitados, conforme podemos ver na Figura 20.

```
@prefix rdfs:           <http://www.w3.org/2000/01/rdf-schema#> .
@prefix qb:             <http://purl.org/linked-data/cube#> .
#=======================================
# Declaração do Visão do Cubo
# Definida pelo Usuário
#=======================================
ex:dsd-cube a qb:DataStructureDefinition;
      rdfs:comment "Vendas por cliente por Tipo de Estabelecimento e
Transação"@pt;
      # The dimensions
      qb:component ex:NuAno ;
      qb:component ex:NuMes ;
      qb:component ex:SexoCli;
      qb:component ex:TpEstab;
      qb:component qb:measureType;
      # The measure(s)
      qb:component ex:ValCompra;
      qb:component ex:QtdCompra.
```

Figura 19 – Exemplo de arquivo RDF com a Visão do Cubo definida pelo usuário

```
#=======================================
# Declaração do Dataset
# ex:cube1 é igual à ex:dataset1 - Percy
#=======================================
ex:cube1 a qb:DataSet;
    qb:structure ex:dsd-cube ;
    rdfs:label   "Cubo   Demonstrativo   de   Compras   por   Tipo   de
Estabelecimento, Tipo de Transação e Características do Cliente"@pt.
#=======================================
# Observações retornadas
#=======================================
ex:obs1a  a qb:Observation ;
    qb:dataSet     ex:cube1 ;
    ex:NuAno       dbpedia:2010;
```

```
    ex:NomeMes      dbpedia:Julho;
    ex:SexoCli      dbpedia:Masculino;
    ex:TpEstab      dbpedia:Theater;
    qb:unitMeasure dbpedia:Money ;
    ex:valCompra '200,00'^^xsd:double .

ex:obs1b  a qb:Observation ;
    qb:dataSet      ex:cube1 ;
    ex:NuAno        dbpedia:2010;
    ex:NomeMes      dbpedia:Julho;
    ex:SexoCli      dbpedia:Masculino;
    ex:TpEstab      dbpedia:Theater;
    qb:unitMeasure dbpedia:quantity ;
    ex:QtdCompra '3'^^xsd:integer .
```

Figura 20 – Exemplo de arquivo RDF com os dados do cubo

A ferramenta, então, lê o arquivo RDF retornado pelo Mediador, linha a linha, buscando os blocos de observações que irão compor a tabela multidimensional que será apresentada para o usuário.

Cada bloco de observação é composto pelos valores de cada um dos atributos e das métricas escolhidas pelo usuário. Sendo assim, os dados que irão compor cada uma das linhas da tabela multidimensional podem estar espalhados no documento RDF.

Para contornar esse problema, a ferramenta lê e processa as informações por bloco de observação, no qual o algoritmo realiza um *match* para cada valor dos atributos, verificando se a métrica representada no bloco específico deve compor uma linha já montada da tabela, ou se uma nova linha para representar as informações do bloco deve ser criada.

Após ler todo o documento, a tabela multidimensional montada pela ferramenta é apresentada para o usuário. A partir daí ele poderá realizar operações OLAP de *slice, dice, drill up, drill down,* filtragem e ordenações, sem que seja necessário solicitar um novo arquivo RDF para o Mediador, conforme ilustrado ver na Figura 21.

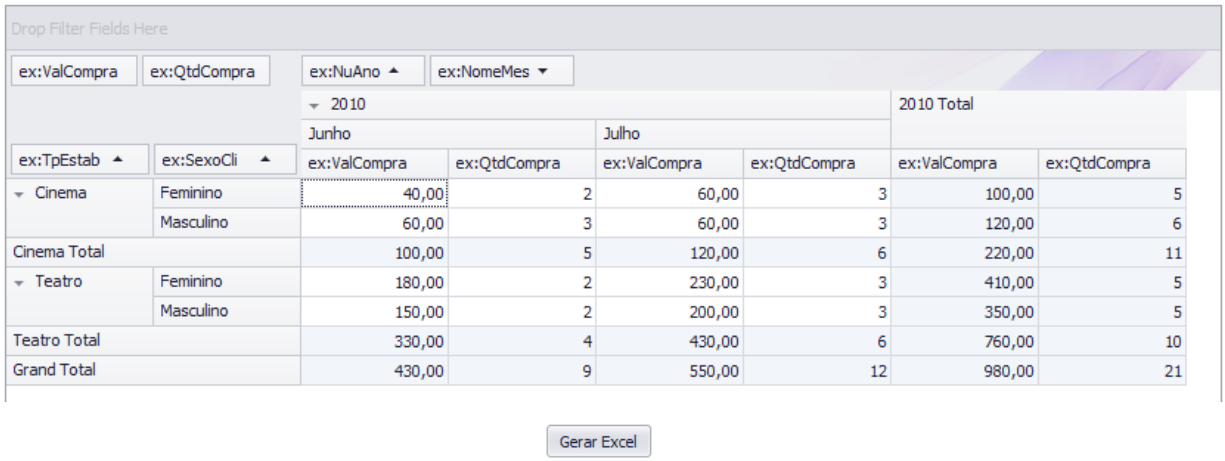


Figura 21 – Exemplo de tabela multidimensional exibida para o usuário

Por fim, após realizar todas as operações desejadas sobre os dados, o usuário poderá exportar a tabela para a planilha MS Excel. Para tal basta acionar o botão "Gerar Excel", ilustrado na parte inferior da Figura 21. O resultado está ilustrado na Figura 22

	A	B	C	D	E	F	G	H	I	J	K	L
1	ex:ValCompra		ex:QtdCompra		ex:NuAno	ex:NomeMes						
2					2010						2010 Total	
3					Junho				Julho			
4	ex:TpEstab		ex:SexoCli		ex:ValCompra		ex:QtdCompra		ex:ValCompra	ex:QtdCompra	ex:ValCompra	ex:QtdCompra
5	Cinema		Feminino		40,00		2,00		60,00	3,00	100,00	5,00
6			Masculino		60,00		3,00		60,00	3,00	120,00	6,00
7	Cinema Total				100,00		5,00		120,00	6,00	220,00	11,00
8	Teatro		Feminino		180,00		2,00		230,00	3,00	410,00	5,00
9			Masculino		150,00		2,00		200,00	3,00	350,00	5,00
10	Teatro Total				330,00		4,00		430,00	6,00	760,00	10,00
11	Grand Total				430,00		9,00		550,00	12,00	980,00	21,00

Figura 22 – Exemplo de planilha MS Excel gerada por meio do processo RDF2Excel

5.2
Excel2RDF

Este módulo da ferramenta realiza a conversão de dados disponíveis no formato de planilha MS Excel para o formato de triplas RDF. A ideia é que este processo sirva com dual do processo descrito na seção anterior. Desta forma, desejamos que o arquivo RDF de saída do processo Excel2RDF seja o mais próximo possível do arquivo RDF de entrada do processo RDF2Excel,

descrito anteriormente, garantindo que o arquivo RDF gerado pela ferramenta poderá ser consumido por outras aplicações capazes de ler o padrão RDF adotado em ambos módulos da ferramenta proposta.

Como sabemos, as planilhas que nos propomos a conveter contém dados e metadados. Em geral, as linhas superiores e as colunas mais à esquerda da planilha contêm metadados, isto é, conceitos (do que a planilha trata) e o miolo contém os dados (instâncias de valores). Durante o processo de triplificação, é muito importante separar os conceitos dos valores, para que ambos sejam devidamente representados por meio do *Data Cube Vocabulary*.

5.2.1 Arquitetura

A arquitetura adotada na ferramenta, conforme vemos na Figura 23, se divide em cinco componentes, além da Interface Gráfica: Seleção de Dados, Leitura das Dimensões, Mapeamento das Dimensões, Geração RDF e Output – Arquivo RDF.

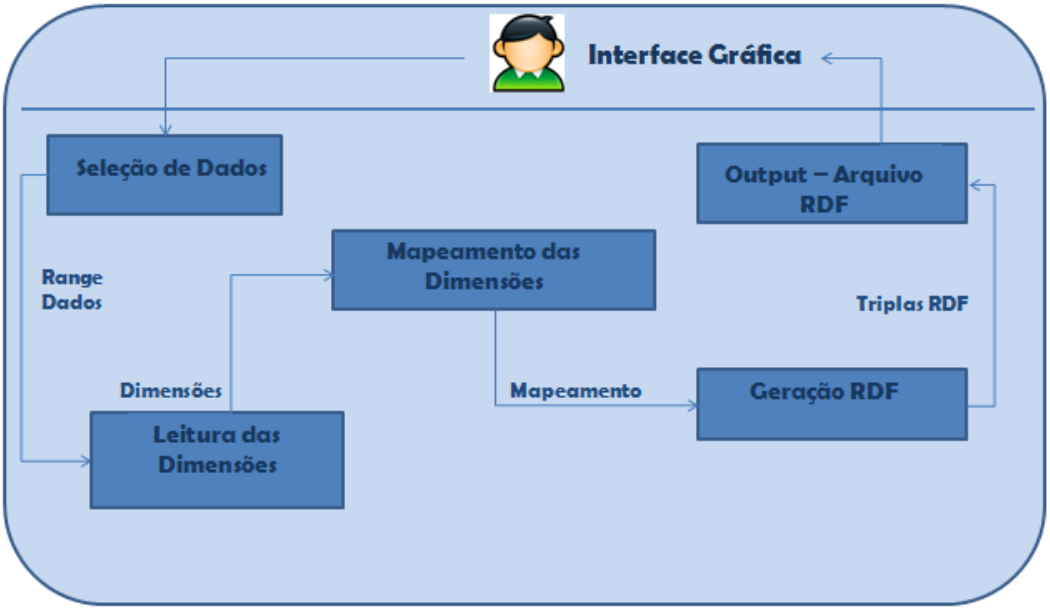


Figura 23 – Arquitetura da Ferramenta Excel2RDF

1. **Seleção Dados**

Este componente é responsável por permitir que o usuário escolha qual o intervalo de dados da sua planilha, ou seja, quais linha e colunas fazem parte dos dados, separando assim, os dados das dimensões (conceitos).

2. **Leitura Dimensões**

Este componente é responsável por ler as dimensões da planilha, sendo necessário separar seus dados de seus conceitos. Internamente, a ferramenta assume que todas as linhas acima, e à esquerda da seleção realizada no componente anterior, compõem as dimensões da planilha.

3. **Mapeamento Dimensões**

Este componente é responsável por realizar o mapeamento dos conceitos encontrados no componente anterior para as dimensões. Como já foi dito anteriormente, a ferramenta mapeia todos os conceitos encontrados em dimensões, sendo esta a principal diferença entre o RDF de entrada do módulo RDF2Excel (que distingue a dimensão dos atributos de cada dimensão) para o RDF de saída do módulo Excel2RDF.

4. **Geração RDF**

Este componente é responsável por converter os dados (observações), e os metadados (dimensões) da planilha, para o formato de triplas RDF, seguindo as definições do *Data Cube Vocabulary* e o padrão de arquivo RDF adotado.

5. **Output – Arquivo RDF**

Este componente é responsável por gerar o arquivo RDF, com o nome definido pelo usuário, com as triplas geradas no componente anterior. O arquivo de saída será gerado em um diretório definido pelo usuário, em um arquivo de configuração.

5.2.2
Processo

O processo de conversão de dados disponíveis no formado de planilhas MS Excel para o formato de triplas RDF, se inicia com a escolha dos dados pelo usuário, e termina com os dados sendo convertidos para o formato RDF, está representado na Figura 24, e será descrito em mais detalhes a seguir.

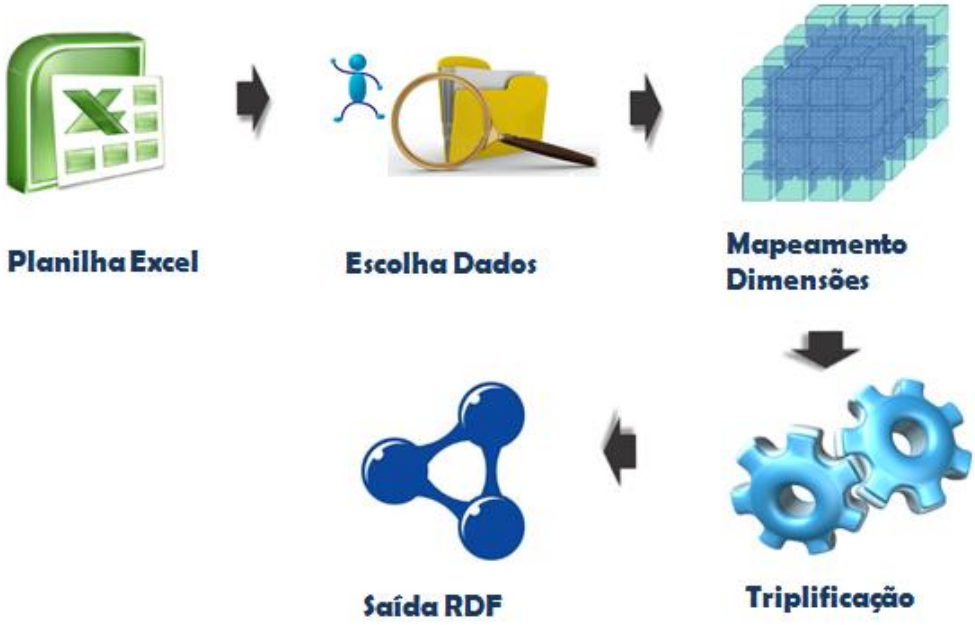


Figura 24 – Processo de conversão Excel2RDF

O processo se inicia com o usuário escolhendo o intervalo dos dados, ou seja, as linhas e colunas que compõe as observações da planilha. A Figura 25 mostra um exemplo de planilha pronta para conversão. Neste caso, o usuário deve selecionar os dados localizados entre a coluna C4 e a coluna F7.

	A	B	C	D	E	F
1			2010			
2			Junho		Julho	
3			R$	Qtd.	R$	Qtd.
4	Teatro	Masculino	150,00	2	200,00	3
5		Feminino	180,00	2	230,00	3
6	Cinema	Masculino	60,00	3	60,00	3
7		Feminino	40,00	2	60,00	3

Figura 25 – Exemplo de planilha a ser convertido para o formato RDF

A Figura 26 mostra a interface gráfica apresentada ao usuário nesse momento.

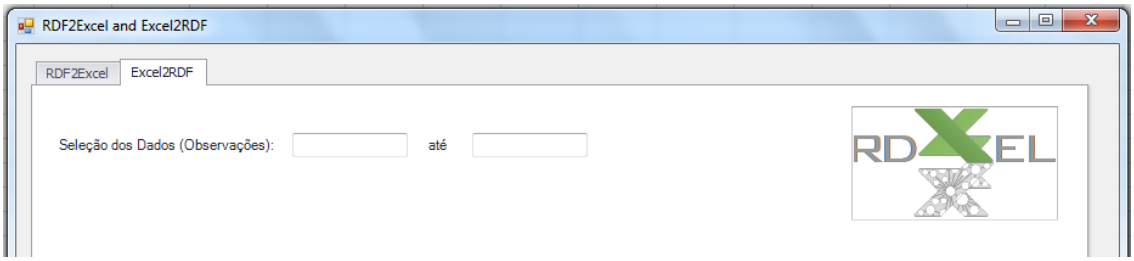


Figura 26 – Usuário seleciona os dados que deseja converter

A ferramenta, então, assume que todas as linhas acima e para a esquerda da seleção formam o cabeçalho da planilha. Estas informações localizadas nas colunas A e B, e nas linhas 1-3, serão utilizadas para a identificação de possíveis dimensões, métricas e agrupamentos.

Os conceitos encontrados são exibidos para o usuário, para que ele faça a distinção entre as dimensões e métricas, e para que ele dê o nome adequado a cada uma destas.

No exemplo acima, imagine que o usuário realizou o seguinte mapeamento:

Conceito	Tipo de Informação	Termo Adequado
R$	Métrica	Reais
Qtd.	Métrica	Quantidade
Cinema, Teatro	Dimensão	Tipo de Estabelecimento
Masculino, Feminino	Dimensão	Sexo
Junho, Julho	Dimensão	Mês
2010	Dimensão	Ano

Tabela 7 – Possível Mapeamento das Dimensões e Métricas para o exemplo da Figura 25

A Figura 27, ilustra a interface gráfica apresentada para o usuário neste momento:

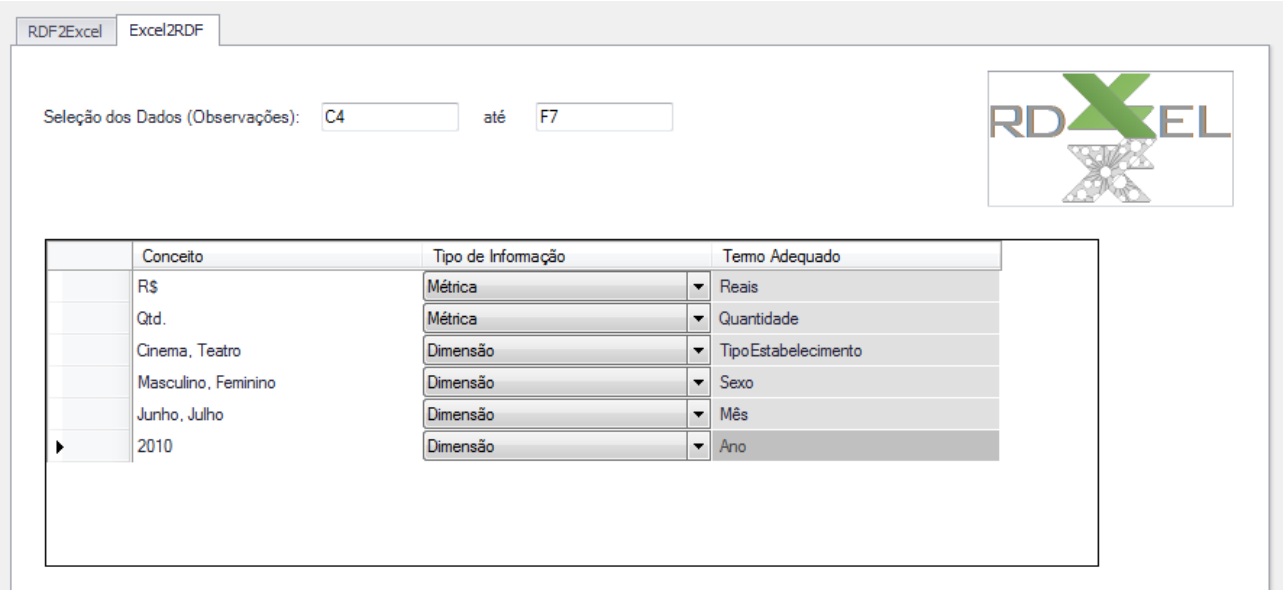


Figura 27 – Interface Gráfica do Mapeamento de Dimensões e Métricas

Após o mapeamento das métricas e dimensões da planilha, o usuário solicita a conversão dos dados. Para isso, ele informa o nome do arquivo, e clica no botão 'Gerar RDF', ilustrado na parte inferior da na Figura 28. A ferramenta verifica o preenchimento do mapeamento, o preenchimento do nome do arquivo, e inicia o processo de triplificação.

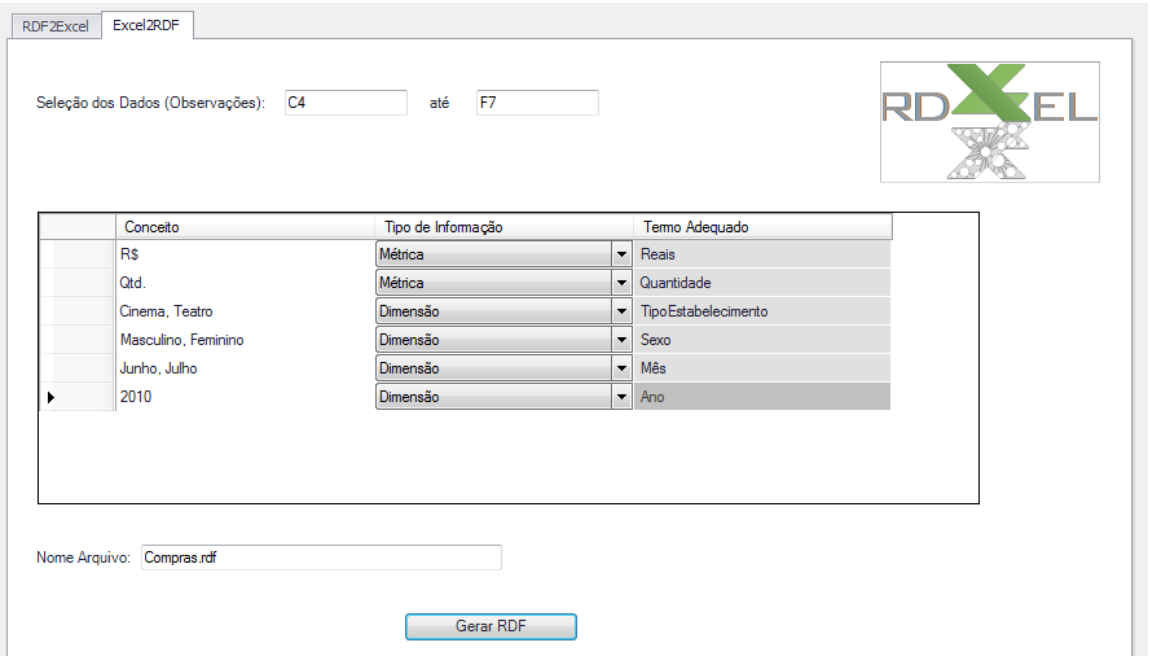


Figura 28 – Interface Gráfica do Processo de Triplificação

A ferramenta, então, inicia o processo de triplificação, mapeando primeiramente as dimensões da planilha, de acordo com o padrão estabelecido pelo *Data Cube Vocabulary*, como podemos observar na Figura 29.

```
@prefix rdf:            <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
.
@prefix rdfs:           <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:            <http://www.w3.org/2002/07/owl#> .
@prefix xsd:            <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:           <http://www.w3.org/2004/02/skos/core#> .
@prefix qb:             <http://purl.org/linked-data/cube#> .
@prefix dcterms:        <http://purl.org/dc/terms/> .
@prefix vocab:          <http://www.inf.puc-rio.br/vocab#>
#============================================================
# Definicões
#============================================================
vocab:tbdimensao rdf:type owl:class;
  rdf:label "Tabela de Dimensao no modelo OLAP".

#=======================================
# Declaracao do Cubo
#=======================================
ex:cube1 a qb:DataSet;
    qb:structure ex:dsd-cube ;
#=======================================
# Declaracao das Dimensões
#=======================================
ex:DimTipoEstabelecimento   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao TipoEstabelecimento"@pt ;
  skos:inScheme ex:Cube1.

ex:DimSexo   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao Sexo"@pt ;
  skos:inScheme ex:Cube1.

ex:DimAno   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao Ano"@pt ;
  skos:inScheme ex:Cube1.

ex:DimMes   rdf:type   vocab:tbdimensao;
  rdfs:label "Dimensao Mês"@pt ;
  skos:inScheme ex:Cube1.
```

Figura 29 – Triplas geradas para as dimensões da planilha

Em seguida, no mesmo documento, são geradas as triplas das métricas da planilha, como podemos observar na Figura 30.

```
#=======================================
# Declaracao das Metricas
#=======================================
ex:Reais rdf:type qb:MeasureProperty;
  skos:inscheme ex:Cube1;
  rdfs:label "Reais"@pt;
  rdfs:range xsd:double.

ex:Quantidade rdf:type qb:MeasureProperty;
  skos:inscheme ex:Cube1;
  rdfs:label "Quantidade"@pt;
  rdfs:range xsd:double.
```

Figura 30 – Triplas geradas para as métricas da planilha

Por fim, são geradas as triplas com as observações dos dados da planilha, como ilustrado na Figura 31.

```
#=======================================
# Observações
#=======================================
ex:obs1a  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Julho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:Reais '200,00'^^xsd:double .

ex:obs1b  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Julho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:Quantidade '3'^^xsd:double .

ex:obs2a  a qb:Observation ;
```

```
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Junho;
    ex:DimSexo              dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:Valor '150,00'^^xsd:double .

ex:obs2b  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Junho;
    ex:DimSexo              dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:Quantidade '2'^^xsd:double .

ex:obs3a  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Julho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:valCompra '230,00'^^xsd:double .

ex:obs3b  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Julho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:QtdCompra '3'^^xsd:double .

ex:obs4a  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Junho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Teatro;
    qb:unitMeasure
    ex:valCompra '180,00'^^xsd:double .

ex:obs4b  a qb:Observation ;
```

```
    qb:dataSet     ex:cube1 ;
    ex:DimAno       dbpedia:2010;
    ex:DimMes     dbpedia:Junho;
    ex:DimSexo     dbpedia:Feminino;
    ex:DimTipoEstabelecimento     dbpedia:Teatro;
    qb:unitMeasure
    ex:QtdCompra '2'^^xsd:double .

ex:obs5a  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Julho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:valCompra '60,00'^^xsd:double .

ex:obs5b  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Julho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:QtdCompra '3'^^xsd:double .

ex:obs6a  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Junho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:valCompra '60,00'^^xsd:double .

ex:obs6b  a qb:Observation ;
    qb:dataSet             ex:cube1 ;
    ex:DimAno                    dbpedia:2010;
    ex:DimMes              dbpedia:Junho;
    ex:DimSexo             dbpedia:Masculino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:QtdCompra '3'^^xsd:double .

ex:obs7a  a qb:Observation ;
```

```
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Julho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:valCompra '60,00'^^xsd:double .

ex:obs7b  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Julho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:QtdCompra '3'^^xsd:double .

ex:obs8a  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Junho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:valCompra '40,00'^^xsd:double .

ex:obs8b  a qb:Observation ;
    qb:dataSet              ex:cube1 ;
    ex:DimAno                       dbpedia:2010;
    ex:DimMes               dbpedia:Junho;
    ex:DimSexo              dbpedia:Feminino;
    ex:DimTipoEstabelecimento   dbpedia:Cinema;
    qb:unitMeasure
    ex:QtdCompra '2'^^xsd:double .
```

Figura 31 – Triplas das observações da planilha

No capítulo seguinte iremos apresentar as nossas conclusões e observações realizadas durante o desenvolvimento desta dissertação. Discutiremos as limitações do framework proposto, bem como as da ferramenta que o implementa. Por fim, apresentamos nossa proposta para trabalhos futuros.

5.3
Resumo Capítulo 5

Neste capítulo apresentamos em mais detalhes a ferramenta rdXel, subdividindo-a em dois módulos: RDF2Excel e Excel2RDF. Para ambos os módulos foram apresentadas as suas arquiteturas e os processos de conversão de dados.

6
Conclusão

Nesta dissertação apresentamos o RdXel, uma ferramenta concebida para detalhar, especificar e desenvolver a camada de aplicação (*Application Layer*) do Framework *OLAP2DataCube Catalog On Demand*. Seu principal objetivo é realizar a conversão de dados disponíveis no formato de triplas RDF, geradas pelo Framework, para o formato de planilha MS Excel. A este processo demos o nome de RDF2Excel.

Observamos, durante o processo de desenvolvimento da ferramenta, que seria importante possibilitar a conversão no caminho inverso, ou seja, do formato de planilha MS Excel para o formato de triplas RDF. Diante disso, incorporamos à ferramenta o módulo Excel2RDF, que realiza a conversão dos dados, que gera um arquivo RDF semelhante ao arquivo RDF de entrada no módulo RDF2Excel.

6.1
Contribuições

A principal contribuição do Framework *OLAP2DataCube Catalog On Demand*, em relação às ferramentas existentes que tratam da mesma questão, é o foco em mitigar um problema comum em aplicações deste tipo, a replicação dos dados.

Seguindo uma abordagem *on demand* (Bizer, 2010), no qual a integração dos dados ligados é realizada de forma evolutiva, e realizando a conversão dos dados em tempo real de acordo com a necessidade, o problema de replicação pode ser contornado. Para isto, a abordagem utiliza a publicação dos dados em formatos que permitem sua reutilização, neste caso particular, o RDF.

Diante deste cenário, nos preocupamos em desenvolver o processo RDF2Excel da ferramenta rdXel de forma que seja possível a conversão dos dados em tempo real, e de acordo com as necessidades de seus usuários, ou seja, permitindo com que o usuário escolha dimensões, atributos e métricas que

deseja, e a forma como deseja organizá-las ao final do processo de conversão. Desta forma, permitimos que o usuário realize operações de *slice and dice* no cubo escolhido, diminuindo assim a quantidade de triplas que terão que ser geradas pelo W*rapper*.

Também com o mesmo objetivo, nos preocupamos em permitir que o usuário trabalhe em cima dos dados convertidos, realizando operações de *slice, dice, drill up, drill down* e filtragem em memória, sem que haja necessidade de uma nova conversão de dados, evitando assim a replicação dos mesmos.

6.2 Limitações

Durante o processo de desenvolvimento da ferramenta, observamos algumas limitações, que serão listadas a seguir:

- ✓ O Framework *OLAP2DataCube Catalog On Demand* não está preparado para realizar o mapeamento dos termos encontrados para outros vocabulários padrão, como *Dublin Core Metadata Initiative* (DCMI), *Friend-of-a-Friend* (FOAF) e *Basic Geo*. O processo de triplificação se baseia somente nos vocabulários *Data Cube Vocabulary*, *Simple Knowledge Organization System* (SKOS) e Vocab (vocabulário definido pelo grupo de pesquisa da PUC-Rio);
- ✓ Para evitar incorporar mais diferenças entre o arquivo RDF de entrada no processo RDF2Excel e o arquivo RDF de saída do processo Excel2RDF, a ferramenta também não está preparada para realizar o mapeamento dos termos encontrados para os vocabulários padrão;
- ✓ A ferramenta RDF2Excel não permite a distinção entre uma dimensão e um atributo de uma dimensão, o que contribui para aumentar as diferenças entre o arquivo RDF de entrada no processo RDF2Excel, e o arquivo RDF de saída do processo Excel2RDF;

6.3
Trabalhos Futuros

Tendo como base as limitações citadas na seção 6.2, podemos destacar como oportunidades para possíveis trabalhos futuros:

- ✓ Desenvolver o módulo *Vocabulary Mapping Module*, previsto no padrão *Crawling Data*, para realizar o mapeamento dos termos encontrados para os vocabulários padrão;
- ✓ Incorporar à ferramenta Excel2RDF o módulo desenvolvido no plug-in Xcel2RDF (Pesce, et al., 2012), que realiza o mapeamento dos termos encontrados para os vocabulários conhecidos;
- ✓ Realizar ajustes na ferramenta RDF2Excel a fim de minimizar as diferenças entre o arquivo RDF de entrada no processo RDF2Excel e o arquivo RDF de saída do processo Excel2RDF;
- ✓ Realizar os ajustes necessários para permitir que o usuário exporte diferentes visões do cubo selecionado para as abas de uma mesma planilha MS Excel.

7
Referências Bibliográficas

Apache. 2012. Open Office. [Online] 2012. http://www.openoffice.org/pt-br/.

Auer, S., Dietzold, S. and Riechert, T. 2006. Ontowiki - A Tool for Social, Semantic Collaboration. *ISWC.* 2006.

Auer, S., et al. 2008. Triplify. [Online] 2008. http://triplify.org/Overview.

Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web. [Online] 2001. http://www.scientificamerican.com/article.cfm?id=the-semantic-web.

Bernes-Lee, T. 2007. Linked Data. *W3C Design Issues.* [Online] Julho 2007. http://www.w3.org/DesignIssues/LinkedData.html.

Bernes-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web. *Scientific American.* 2001.

Berrueta, D. and Phipps, J. 2008. Best Practice Recipes for Publishing RDF Vocabularies. [Online] W3C, 2008. http://www.w3.org/TR/swbp-vocab-pub/.

Bizer, C. 2010. On Demand Data integration on the public Web of Linked Data. *3rd Future Internet Symposium FIS2010.* 2010.

Bizer, C., et al. 2007. Interlinking Open Data on the Web. *ESWC2007 Poster Paper.* 2007.

Bizer, C., Heath, T. and Bernes-Lee, T. 2009. Linked Data - The Story so Far. 2009.

Breitman, K., Casanova, M. and Truszkowski, W. 2006. *Semantic Web: Concepts, Tecjnologies and Applications.* Londres : Springer, 2006.

Brickley, G. and Guha, R. V. 2004. RDF Vocabulary Description Language 1.0: RDF Schema. [Online] W3C, 2004. http://www.w3.org/TR/rdf-schema/.

Cyganiak, R. and Jentzsch, A. 2011. The Linking Open Data Cloud Diagram. [Online] 2011. http://richard.cyganiak.de/2007/10/lod/.

Heath, T. and Bizer, C. 2011. *Linked Data.* s.l. : Morgan & Claypool Publishers, 2011.

—. 2011. *Linked Data: Evolving the Web into a Global Data Space.* 2011.

ISO. 2005. *Statistical data em metadata exchange (sdmx). Technical report.* 2005. ISO/TS 17369:2005.

Kämpgen, B. and Harth, A. 2011. Transforming Statistical Linked Data for Use in OLAP Systems. *ISWC.* 2011.

Kendall, G. C., Feigenbaum, L. and Torres, E. 2008. SPARQL Protocol for RDF. [Online] W3C, 2008. http://www.w3.org/TR/rdf-sparql-protocol/.

Langegger, A. 2009. XLWrap – Spreadsheet-to-RDF Wrapper. [Online] 2009. http://xlwrap.sourceforge.net/.

Manola, F. and Miller, E. 2004. RDF Primer. *W3C Recommendation 10 February 2004.* [Online] W3C, February 2004. http://www.w3.org/TR/rdf-primer/.

McGuinness, D. L. 2002. *Ontologies Come of Age. Spinning the semantic web: bringing the World Wide Web to its full potential.* 2002.

Pesce, M., Breitman, K. and Casanova, M. 2012. Surfacing Scientific and Financial with the Xcel2RDF Plug-in. *ICSE - Topi.* 2012.

Prud'Hommeaux, E. and Seaborne, A. 2008. SPARQL Query Languages for RDF. [Online] W3C, 2008. http://www.w3.org/TR/rdf-sparql-query/.

Salas, P. 2011. StdTrip: An a priori design process for publishing Linked Data. 2011.

Salas, P., et al. 2012. OLAP2DataCube: An Ontowiki Plug-In for Statistical Data Publishing. *TOPI.* 2012.

Salas, P., et al. 2012. Publishing Statistical Data on the Web. 2012.

SDMX. 2009. SDMX Content-Oriented Guidelines. [Online] 2009. http://unstats.un.org/unsd/dnss/docs-nqaf/04_sdmx_cog_annex_4_mcv_2009.pdf.

—. 2012. Statistical Data and Metadata eXchange. *SDMX*. [Online] 2012. http://sdmx.org/.

Seaborne, A. and Bizer, C. 2004. D2RQ - treating non-RDF databases as virtual RDF graphs. *ISWC2004.* 2004.

Smith, M. K., Welty, C. and McGuinness, D. L. 2004. OWL Web Ontology Language. [Online] W3C, 2004. http://www.w3.org/TR/owl-guide/.

Thomsen, E. 1997. *OLAP Solutions: Building Multidimensional Information Systems.* 1997.

Zaveri, J. 2010. Stats2RDF - Representing multi-dimensional statistical data as RDF using the RDF Data Cube Vocabulary. [Online] 2010. http://aksw.org/Projects/Stats2RDF#h13390-6.