SPARQL2MDX: Um Componente de Tradução de Consultas em Ontologia para *Data Warehousing*

Fabiola de L. Moreira[1], **Eduardo M. de Freitas Jorge**[2]

[1]Universidade do Estado da Bahia (UNEB), Salvador, BA, Brasil

[2]Instituto Recôncavo de Tecnologia, Salvador, BA, Brasil

faumoreira_flm@yahoo.com.br,emjorge1974@gmail.com

Abstract. *In the contemporary market, it is common sense that information is valuable for organizations' decision process. Data Warehousing is among the technologies used for information management support. This work intends to propose and develop a component capable of building, from SPARQL queries over ontologies, MDX queries for Data Warehousing solutions. Once these objectives are achieved, end-users can access data stored in the Data Warehousing from a query based on organisation's knowledge, represented on the ontology.*

Resumo. *No mercado contemporâneo, é de senso comum que a informação tem um grande valor no processo decisório das organizações. Dentre as tecnologias utilizadas para apoiar o gerenciamento de informações está o Data Warehousing. Este trabalho busca propor e implementar um componente capaz de construir consultas MDX para soluções de Data Warehousing a partir de consultas de ontologias escritas em SPARQL. Uma vez cumprido o objetivo, o usuário final poderá acessar os dados armazenados no Data Warehousing a partir de uma consulta baseada nos conceitos da organização, representados na ontologia.*

1. Introdução

No mercado contemporâneo, é de senso comum que a informação tem um grande valor no processo decisório das organizações, tornando-se um recurso-chave de competitividade efetiva e lucratividade.

Diversas tecnologias são utilizadas para apoiar o gerenciamento das informações de uma organização. Neste trabalho, duas delas serão utilizadas: ontologias e *Data Warehousing*.

Ontologia é a definição de um conjunto de primitivas com as quais se modela um domínio de conhecimento ou universo de discurso. As primitivas são, tipicamente, classes, atributos e as relações entre essas classes [Gruber 2008] e o universo de discurso pode ser qualquer domínio de conhecimento que possa ser representado por um formalismo declarativo [Gruber 1993]. Uma organização ou parte dela pode ser considerada um universo de discurso, e os seus termos e as relações entre eles fazem o papel das primitivas. Enfim, ontologia é uma forma de representar o conhecimento.

Data Warehousing é uma coleção de tecnologias de suporte à decisão que visa auxiliar o analista de negócio a tomar decisões melhores e mais rápidas [Chaudhuri and Dayal 1997]. Uma dessas tecnologias é o *Data Warehouse*, um conjunto

de dados de apoio às decisões gerenciais orientado a assuntos, integrado, não volátil e variável em relação ao tempo [Inmom 1997]. Para realizar consultas no conjunto de dados de um *Data Warehouse*, a linguagem de consulta mais usada é a MDX (*Multi-dimensional Expressions*).

Para construir consultas MDX é preciso ter conhecimentos técnicos em *Data Warehousing* e da sintaxe da linguagem. Apesar do auxílio de ferramentas visuais, há uma tendência que esta função fique a cargo das pessoas especializadas na área de computação ao invés do usuário final, criando uma barreira entre os dados e quem entende do negócio.

A fim de solucionar o problema destacado, este projeto tem por objetivo propor e implementar um componente que constrói consultas MDX para soluções de *Data Warehousing* a partir de consultas de ontologias escritas em SPARQL.

O uso de SPARQL para a criação de consultas em um *Data Warehousing* possibilita que os usuários finais acessem os dados armazenados a partir de uma consulta baseada nos conceitos da organização, representados na ontologia.

Este trabalho está organizado da seguinte forma: as seções 2 e 3 discutem, respectivamente, sobre *Data Warehousing* e ontologia; a seção 4 apresenta o componente, denominado SPARQL2MDX, apresentando a sua estrutura, processo de tradução, tecnologias utilizadas e os testes realizados; por fim, as considerações finais e os projetos futuros, na seção 5.

2. Data Warehousing

Data Warehousing é uma coleção de tecnologias de suporte à decisão que visa auxiliar o analista de negócio a tomar decisões melhores e mais rápidas [Chaudhuri and Dayal 1997].

Dentre essas tecnologias está o *Data Warehouse*: um repositório onde são armazenados todos os dados relevantes de uma organização e a partir do qual surgem as informações e o conhecimento necessário para gerir eficazmente a organização [March and Hevner 2007, apud Watson 2002].

Para facilitar análises complexas e a visualização, os dados em um *Data Warehouse* são geralmente modelados seguindo o modelo multidimensional [Chaudhuri and Dayal 1997]. A idéia central desse modelo de dados é a noção de cubo, no qual as dimensões determinam o contexto qualitativo dos fatos; o fato é o elemento atômico de informação em um banco de dados multidimensional, o assunto a qual o cubo se trata [Hüsemann et al. 2000]. Os dados referente ao fato são as medidas; é o objeto de análise no modelo multidimensional [Chaudhuri and Dayal 1997].

Cada dimensão é descrita por um conjunto de atributos e, freqüentemente, esses atributos são relacionados via hierarquia de relacionamentos [Chaudhuri and Dayal 1997]. Na figura 1, a dimensão *Loja* pode ser organizada em uma hierarquia Logradouro-Cidade-Estado, por exemplo.

Não basta somente ter os dados bem estruturados, é preciso uma boa ferramenta para análise. As ferramentas mais utilizadas são as baseadas no Processamento Analítico On-line (também conhecido como OLAP - *On-line Analytical Process*).

Além de responsáveis pela análise dos dados, os sistemas OLAP são responsáveis

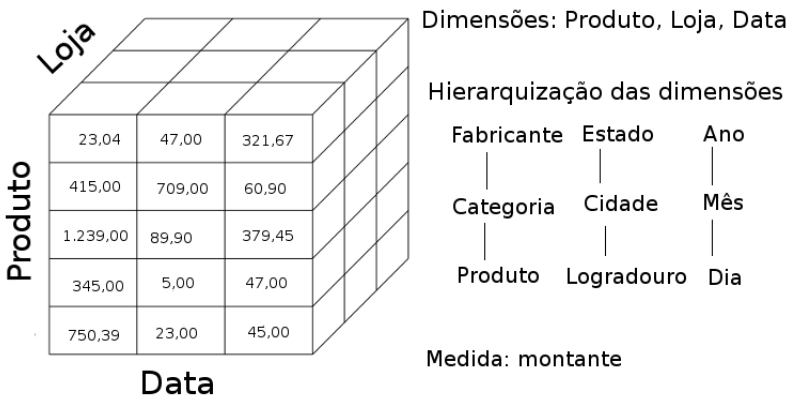

Figura 1. Cubo Multidimensional. Adaptada de [Chaudhuri and Dayal 1997]

também pela execução das consultas ao *Data Warehouse*. Para isso, a linguagem de consulta mais usada é a MDX (*Multi-Dimensional Expressions*). A estrutura da consulta MDX está descrita abaixo:

```
SELECT [<axis_specification>
       [, <axis_specification>...]]
  FROM [<cube_specification>]
WHERE [<slicer_specification>]]
```

Cada `<axis_specification>`descreve como produzir uma das dimensões do resultado; `<cube_specification>` determina qual fonte de dados multidimensional deve ser usada para extrair os dados e popular o resultado; e o `<slicer_specification>` permite restringir os resultados [Niemi et al. 2001].

Uma simples consulta MDX é representada abaixo:

```
SELECT {([Data].[1998],[Loja].[All Lojas])} ON ROWS,
       {[Measures].[montante]} ON ROWS
FROM [Venda]
```

A consulta acima retorna um cubo de duas dimensões com o montante de vendas realizadas no ano de 1998 em todas lojas.

3. Ontologia

Segundo [Gruber 1993], *"uma ontologia é uma especificação explícita de uma conceitualização"*, sendo a conceitualização uma visão simplificada do mundo que desejamos representar para algum propósito. Segundo o consórcio W3C, uma ontologia define os conceitos e suas relações usadas para descrever e representar uma área de conhecimento. Em relação à banco de dados, ontologia pode ser vista como um nível de abstração do modelo de dados, tanto para os modelos hierárquico e relacional, voltado para a modelagem do conhecimento sobre os indivíduos, seus atributos e suas relações com outros indivíduos [Gruber 2008].

Uma das linguagens de representação de ontologias é a RDF (*Resource Data Framework*), que foi criada para representar informações sobre recursos na web. O modelo de representação do RDF é uma tripla formada pelos seus princípios fundamentais: sujeito ou recurso, predicado e objeto [Breitman 2005]. O sujeito identifica sobre o que a

tripla descreve, podem ser pessoas, objetos, etc; o predicado é a propriedade ou característica do sujeito, por exemplo, “escrito por” e “idade”; e o objeto representa o valor da propriedade.

RDF Schema é um framework no qual é possível escrever classes e propriedades [Breitman 2005]: as classes são grupos de recursos com mesmas características – os membros de uma classe são chamadas de instância dessa classe – e as propriedades são as relações entre dois recursos, um no papel do sujeito e outro do objeto [Brickley and Guha 2004].

Outra linguagem de representação de ontologias é a OWL (*Ontology Web Language*): é uma linguagem para definição e instanciação de ontologias Web [Smith et al. 2004] capaz de formalizar um domínio, definindo classes e suas propriedades, além dos indivíduos e as informações sobre eles [Lima and Carvalho 2005].

Os elementos básicos para a construção de uma ontologia OWL são [Lima and Carvalho 2005]: as *classes* que descrevem os conceitos mais básicos de um domínio; os *indivíduos*,os membros das classes, e as *propriedades*, relações binárias que permitem afirmar fatos gerais sobre os indivíduos. Existem dois tipos de propriedades:

Propriedades de objetos (*object properties*) relação entre os indivíduos de duas classes.

Propriedades de dados tipados (*datatype properties*) relação entre indivíduos e valores de dados, como strings e inteiros, por exemplo.

Para realizar consultas nas informações presentes em uma ontologia, a W3C recomenda a linguagem de consulta SPARQL (*Sparql Protocol and Rdf Query Language*).

Vejamos uma simples consulta SPARQL:

```
1 PREFIX p: <http://www.owl-ontologies.com/venda.owl/#>
2 SELECT ?descricao
3 FROM p:
4 WHERE {?produto p:descricao ?descricao}
```

Na linha 1, aparece a palavra reservada PREFIX . Ela associa um marcador a uma URI, evitando-se a sua escrita várias vezes na consulta.

Assim como na linguagem SQL, as cláusulas SELECT e FROM definem, respectivamente, quais dados serão retornados e sob quais dados a consulta será realizada.

Na cláusula WHERE são definidas as *triple patterns*. Uma SPARQL *triple pattern* é equivalente à tripla do RDF, também consiste numa tripla formada por um sujeito, um predicado e um objeto, sendo que, no SPARQL, qualquer elemento da tripla pode ser substituído por uma variável. Na linha 4 da consulta anterior, o sujeito e o objeto da *triple pattern* foram substituídos pelas variáveis *?produto* e *?descricao*, respectivamente.

Para restringir os resultados de uma consulta, basta substituir um elemento da *triple pattern* por um valor correspondente. Executando a consulta abaixo, serão retornados todos os fabricantes que produzem um produto cuja descrição “fogao”:

```
1 SELECT ?fabricante
2 FROM p
3 WHERE {
```

```
4         ?produto p:fabricado_por ?fabricante.
5         ?produto p:descricao "fogao".}
```

4. SPARQL2MDX

O objetivo deste trabalho é propor e implementar um componente que realiza a tradução de consultas escritas em SPARQL para MDX. Optou-se por criar um componente - denominado SPARQL2MDX - tendo em vista prover uma ferramenta acoplável a outras soluções de escopo maior que apóiem o usuário final no processo de análise dos dados.

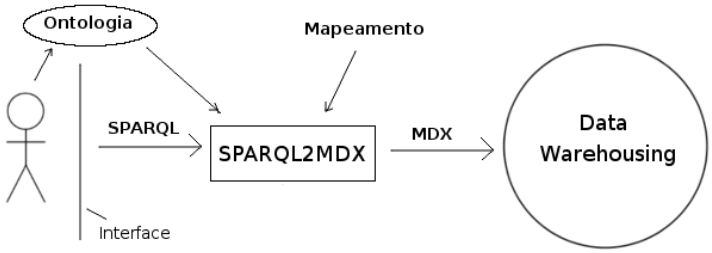

Figura 2. SPARQL2MDX: uma ponte entre ontologias e *Data Warehousing*

A Figura 2 apresenta um cenário de utilização do SPARQL2MDX. Nesse cenário, o usuário final, com acesso à ontologia e o apoio de uma interface, envia uma consulta SPARQL para o componente. Durante a tradução da consulta recebida, o componente acessa a ontologia e um arquivo de mapeamento (a função da ontologia e do mapeamento no processo de tradução será explicado nas próximas seções). Finalmente, a consulta MDX gerada pelo componente é executada em uma solução de *Data Warehousing*.

4.1. Mapeamento

Ontologias e cubos multidimensionais têm grandes diferenças na sua modelagem, e, para relacioná-los, foi criado um RDF Schema que define a estrutura de um mapeamento que associa uma ontologia a um cubo multidimensional.

O RDF Schema possui 5 classes: *Cube*, *Measure*, *Dimension*, *Hierarchy* e *Level*, representando, respectivamente, o cubo, as medidas, as dimensões, as hierarquias e os níveis do cubo multidimensional. As propriedades dessas classes são:

name propriedade pertencente a todas as classes; determina o nome do elemento do cubo a ser mapeado.

uri URI do elemento da ontologia que representa o elemento do cubo a ser mapeado. Todas as classes usam essa propriedade, exceto a classe *Hierarchy*. Em um cubo, a hierarquia de uma dimensão é somente um elemento que define e nomeia um conjunto de níveis de agregação. Por não representar dados, a hierarquia de um cubo também não é representativa em uma ontologia e, consequentemente, a classe *Hierarchy* não tem essa propriedade.

all_member Propriedade da classe *Hierarchy*; define o nome do nível mais alto de agregação - nível "All"- de uma hierarquia.

aggregation_level Propriedade da classe *Level*; determina em qual posição da hierarquia o nível a ser mapeado se encontra. O nível mais alto é representado pelo número 1, o próximo pelo número 2 etc.

No modelo de mapeamento proposto, o cubo e as dimensões são representados por classes, e as medidas e os níveis por propriedades pois, respectivamente, são os elementos conceituais e caracterizadores das instâncias dos conceitos, tanto no modelo multidimensional quanto numa ontologia. Por isso, o valor da propriedade URI, quando usada pelas classes *Cube* e *Dimension*, deverá ser a URI de uma classe e, quando usada pelas classes *Level* e *Measure*, deverá ser uma URI de uma propriedade, podendo ser tanto uma propriedade de dado quanto de objeto.

4.2. Processo de tradução

O processo de tradução de uma consulta SPARQL para MDX foi dividido em três fases: compilação da consulta SPARQL, a tradução propriamente dita e a formatação da consulta MDX.

Na fase de compilação da consulta SPARQL, são feitas as análises léxica, sintática e semântica, sendo que, na análise semântica, é preciso acessar a ontologia para a qual a consulta foi feita para: (1) definir o tipo das variáveis, isto é, determinar qual classe a variável representa e (2) verificar se as *triple patterns* estão de acordo com as classes e propriedades definidas na ontologia.

Na fase de tradução, as URIs dos elementos das *triple patterns* – sujeito, predicado e objeto – são consultadas no mapeamento e, partindo do que o elemento representa, são definidos o cubo, as dimensões e as medidas da consulta MDX. O cubo e as dimensões são representadas pelo sujeito ou objeto e as medidas e os níveis pelo predicado da *triple pattern*.

Ainda na fase de tradução, são definidas quais dimensões e medidas farão parte da cláusula SELECT e quais farão parte da cláusula WHERE da consulta MDX: se a variável que representa uma dimensão ou medida numa *triple pattern* estiver presente na cláusula SELECT, esta dimensão ou medida também fará parte da cláusula SELECT da consulta MDX; se não estiver presente, a dimensão ou medida fará parte da cláusula WHERE. Caso a variável que representa um nível de uma dimensão esteja presente na cláusula SELECT da consulta SPARQL, a dimensão a qual o nível pertence fará parte da cláusula SELECT da consulta MDX. Vejamos a consulta SPARQL abaixo:

```
1 PREFIX p: <http://www.owl-ontologies.com/Venda.owl#>
2 SELECT ?produto
3 FROM p:
4 WHERE {?venda p:produto_vendido ?produto.
5        ?venda p:realizada_na_loja ?loja}
```

Supondo que *?produto* e *?loja* representem as dimensões *Produto* e *Loja* de um cubo de vendas, representado pela variável *?venda*, a consulta MDX gerada terá somente a dimensão *Produto* na cláusula SELECT, já que a variável *?loja* não está presente na cláusula SELECT da consulta SPARQL:

```
1 SELECT {([Produto].[All Produtos])} ON COLUMNS
2 FROM [Venda]
3 WHERE [Loja].[All lojas]
```

Com a definição dos elementos das cláusulas SELECT, FROM e WHERE da consulta MDX, é preciso formatar o cubo de resultado, definindo em quais eixos estarão as dimensões. Na fase de formatação, são definidos os eixos das dimensões que formam o cubo de resultado. As dimensões que não tiverem o eixo definido, farão parte do eixo COLUMNS.

4.3. Contexto de Desenvolvimento

A arquitetura do SPARQL2MDX segue o padrão *Facade*. Este padrão foi escolhido para dividir a implementação em 3 camadas, de modo que cada uma delas correspondesse a uma das fases do processo de tradução descritas na seção 4.2.

Durante a análise semântica da consulta SPARQL, é preciso acessar a ontologia para a qual a consulta foi feita e verificar se as *triple patterns* estão de acordo com as classes e propriedades definidas. Uma ontologia pode ser escrita em várias linguagens, com maneiras diferentes de organizar suas informações. Para que o componente fosse extensível, na análise semântica, foi utilizado o padrão *Strategy*, permitindo o acoplamento de algoritmos de acesso as ontologias para cada linguagem.

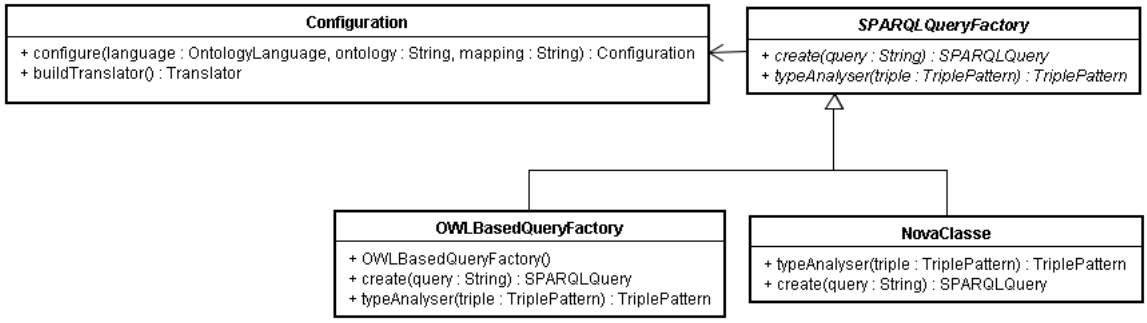

Figura 3. Aplicação do padrão *Strategy* no SPARQL2MDX

Nesta versão, o componente somente suporta ontologias na linguagem OWL. Para dar suporte a outras linguagens, basta criar uma classe que herde de *SparqlQueryFactory*, implementar seus métodos abstratos e atualizar o método *buildTranslator* da classe *Configuration* de modo que ele instancie a classe filha de *SparqlQueryFactory* correta para cada linguagem em que a ontologia foi definida. Essa operação é exemplificada pela Figura 3, com a criação de uma nova classe – *NovaClasse* – que dê suporte a outra linguagem de representação de ontologias.

As principais classes do projeto são:

Configuration Responsável pela configuração do componente: possui métodos que recebem os caminhos de onde estão os arquivos da ontologia e do mapeamento e criam um objeto Translator que possuirá uma instância de uma classe que herda de SparqlQueryFactory, a qual possui o algoritmo correto para a compilação da consulta SPARQL;

SparqlQuery Encapsula os *tokens* que formam as *triple patterns* e as variáveis encontradas na cláusula SELECT da consulta SPARQL;

MdxQuery Encapsula as definições do cubo e das dimensões da consulta MDX;

SparqlQueryFactory Classe base da fase de compilação da consulta SPARQL: possui métodos para realizar as análises léxica e sintática e define métodos abstratos, que serão implementados pelas classes filhas, que farão a análise semântica da linguagem de acordo com cada linguagem de representação de ontologias;

OWLBasedQueryFactory Classe que herda de *SparqlQueryFactory* que tem definido o algoritmo para realização da análise semântica em ontologias escritas em OWL.

Translate Responsável pela tradução das consultas; a partir do objeto da classe *SparqlQuery*, gera um objeto *MdxQuery*;

MdxSolution Encapsula o objeto *MdxQuery* gerado e possui métodos que realizam a formatação da consulta MDX.

4.4. Testes e resultados

O testes realizados no componente tem o objetivo de garantir que a consulta MDX gerada não tenha erros de sintaxe e que todas as dimensões e medidas do cubo, as quais foram representadas pelos elementos que formam as *triple patterns* da consultas SPARQL, estejam na consulta MDX.

Para realizar os testes, foi preciso criar um ambiente de validação. Este ambiente é composto de uma ontologia, e um cubo multidimensional em uma solução de *Data Warehousing* (como descrito na Figura 1), representando as vendas realizadas por uma organização fictícia.

O mapeamento foi criado associando-os da seguinte forma: O *cubo* Venda é representado pela *classe* Venda e a *medida* montante pela *propriedade* montante; a *dimensão* Loja é representada pela *classe* Loja e os seus atributos (logradouro, cidade e estado) pelas propriedades situada_no_endereco, pertence_a_cidade, pertence_a_estado; a *dimensão* Data é representada pela *classe* Data e os seus atributos pelas propriedades referente_ao_dia, referente_ao_mes, referente_ao_ano e a *dimensão* Produto é representada pela *classe* Produto e os seus atributos pelas propriedades descricao, pertence_a_categoria, fabricado_por. Todas as propriedades citadas acima são do tipo *object*, com exceção das propriedades *montante* e *descricao*, que são do tipo *datatype*.

Também foram adicionadas ao mapeamento a definição das hierarquias *Produto*, *Data* e *Localizacao* para as dimensões *Produto*, *Data* e *Loja*, respectivamente. O nível "All" dessas hierarquias anteriores são, respectivamente, *All Produtos*, *All Loja.Localizacaos* e *All Datas*.

A ontologia foi escrita com o apoio da ferramenta Protégé[1] - versão 3.4.1, com licença MPL (*Mozilla Public License*) - e a solução de *Data Warehousing* escolhida foi a Pentaho[2] - versão 3.5.0, com licença GPLv2 (*GNU General Public License Version 2*).

Todas as consultas SPARQL utilizadas nesse teste, assim como as consultas MDX geradas pelo componente foram executadas, na ferramenta Protégé e na solução *Pentaho*, respectivamente, a fim de comprovar sua corretude.

[1]Mais informações em *http://protege.stanford.edu*
[2]Mais informações em *http://www.pentaho.com*

A seguir, serão apresentadas as consultas SPARQL, juntamente com a descrição de como as dimensões foram formatadas, e a consulta MDX gerada pelo componente, partindo dessas informações. Considere que todas as consultas SPARQL têm o prefixo ':' definido.

Na consulta abaixo, as *triple patterns* representam a dimensão *Produto* e a medida *montante* através das variáveis *?produto* e *?montante*, respectivamente, as quais estão presentes na cláusula SELECT. Nesta consulta, os níveis da dimensão *Produto* foram definidos. Os valores dos níveis *fabricante*, *categoria* e *descrição* são, respectivamente, "DICASA", "MÓVEIS" e "Cama casal".

```
SELECT ?produto ?montante
FROM :
{ ?venda :montante ?montante.
  ?venda :produto_vendido ?produto.
  ?produto :fabricado_por :DICASA.
  ?produto :pertence_a_categoria :MÓVEIS.
  ?produto :descricao 'Cama casal'}
```

Definindo a medida *montante* para o eixo ROWS, a consulta MDX gerada é:

```
SELECT {([Produto].[DICASA].[MÓVEIS].[Cama casal])} ON COLUMNS,
  {([Measures].[montante])} ON ROWS
FROM [Venda]
```

Na consulta abaixo, as *triple patterns* representam as dimensões *Loja*, *Produto* e *Data* através das variáveis *?loja*, *?produto* e *?data*, respectivamente, as quais também estão presentes na cláusula SELECT.

```
SELECT ?produto ?loja
FROM :
{ ?venda :produto_vendido ?produto.
  ?venda :realizada_na_loja ?loja.
  ?venda :realizada_na_data ?data.
  ?data :referente_ao_dia ?dia.
  ?dia :referente_ao_mes ?mes.
  ?mes :referente_ao_ano :1998}
```

Definindo as dimensões *Produto* para o eixo COLUMNS e *Loja* para o eixo ROWS, a consulta MDX gerada é:

```
SELECT {([Produto].[All Produtos])} ON COLUMNS, {([Loja].[All Loja.
    Localizacaos])} ON ROWS
FROM [Venda]
WHERE [Data].[1998]
```

As consultas anteriores utilizadas para o teste do componente mostram a capacidade do mesmo de criar consultas MDX com dimensões e medidas, dimensões com e sem níveis definidos – nesse último, utilizando o nível "All" – e definindo dimensões e/ou medidas também na cláusula WHERE. Durante o teste, não foram encontrados erros de lógica. Diante do exposto, pode-se concluir que os resultados foram positivos.

5. Considerações Finais

Este trabalho apresentou um componente que constrói consultas MDX para soluções de Data Warehousing a partir de consultas de ontologias escritas em SPARQL.

Para a realização do projeto, foram definidas quais tecnologias seriam utilizadas na construção do componente, quais comandos SPARQL e MDX seriam suportados, e um ambiente de validação para realizar testes e avaliar a solução criada.

Além da preocupação em atingir o objetivo proposto, também houve empenho para tornar o componente fácil ao uso de outras aplicações e que fosse extensível. Para que ele suportasse outras linguagens de ontologia além de OWL, foi utilizado o padrão *Strategy*. No entanto, aumentar a gramática da linguagem SPARQL demanda um esforço maior, já que é necessário conhecer e realizar atualizações no código em todas as fases do processo de tradução.

Como desenvolvimentos futuros, sugere-se criar uma interface gráfica que auxilie o usuário na criação de consultas SPARQL, e ampliar a gramática da consulta SPARQL suportada pelo componente a fim de construir consultas MDX mais complexas e ampliar as linguagens de representação de ontologias suportadas.

Referências

Breitman, K. (2005). *Web Semântica: a Internet do Futuro*. LTC.

Brickley, D. and Guha, R. V. (2004). Rdf vocabulary description language 1.0: Rdf schema.

Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *ACM SIGMOD Record*, 26:65–74.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220.

Gruber, T. (2008). Ontology (computer science) - definition in encyclopedia database systems.

Hüsemann, B., Lechtenbörger, J., and Vossen, G. (2000). Conceptual data warehouse design. In *International Workshop On Design And Management Of Data Warehouses*, pages 6.1–6.11.

Inmom, W. H. (1997). *Como Construir o Data WareHouse - tradução da segunda edição*. Campus.

Lima, J. C. and Carvalho, C. L. (2005). Ontologias - (web ontology language). Technical report, Instituto de Informática - Universidade Federal de Goiás.

March, S. T. and Hevner, A. R. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems*, 43:1031–1043.

Niemi, T., Nummenmaa, J., and Thanisch, P. (2001). Constructing olap cubes based on queries. In *DOLAP '01: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP*, pages 9–15, New York, NY, USA. ACM.

Smith, M. K., Welty, C., and McGuinness, D. L. (2004). Owl web ontology language guide.