

Uma Ferramenta para suportar a publicação de visões RDF de dados relacionais

Vânia Maria P. Vidal¹, José Maria Monteiro¹, Luís Eufrasio T. Neto¹

¹Departamento de Computação – Universidade Federal do Ceará (UFC)
Qd Campus Pici, 1 – Pici – 60455-900 – Fortaleza – CE – Brazil

{vvidal,zemaria,luisseufrasio}@lia.ufc.br

Abstract. *In This paper we propose a web tool for publishing relational databases in the form of RDF ontologies. For this purpose, we use the mapping language adopted by w3c: R2RML (RDB to RDF Mapping Language), which is supported by our solution. Thus, the user will have a friendly interface to create the mapping files.*

Resumo. *Neste artigo propomos uma ferramenta web para publicação de bases relacionais na forma de ontologias RDF. Para tanto, utilizamos a linguagem de mapeamento adotada pela w3c: R2RML (RDB to RDF Mapping Language), que será suportada pela nossa solução. Assim, o usuário terá uma interface amigável para criar seus arquivos de mapeamento.*

1. Introdução

O padrão Linked Data foi estabelecido como a melhor prática para expor, compartilhar e conectar partes de dados, informações e conhecimentos na Web Semântica usando URIs e RDF [1]. No entanto, a grande maioria dos dados corporativos, inclusive dados da Web, permanecem armazenados em SGBDs relacionais. Segundo [2] a Web Semântica não consiste simplesmente em colocar dados na web, mas principalmente da criação de links entre esses dados de tal modo que pessoas e máquinas possam fazer um melhor uso da web de dados. Para tornar as informações armazenadas em bancos relacionais disponíveis na web de dados, é preciso publicá-las usando o modelo RDF. Essa publicação pode ser feita utilizando mapeamentos de bases relacionais para ontologias RDF. Existe uma linguagem padrão da w3c para criação desses mapeamentos que é a R2RML [10].

Abordaremos duas maneiras de criarmos os mapeamentos nesse trabalho: mapeamento direto definido em [3] e mapeamento para vocabulários comuns como motivado em [4]. Os mapeamentos diretos servem como base para criação de mapeamentos mais complexos, que utilizam vocabulários comuns e promovem o reuso.

Neste trabalho, tratamos do problema de gerar automaticamente mapeamentos diretos e suportamos a criação de mapeamentos complexos. Para tanto, propomos uma ferramenta web que disponibiliza para o usuário a criação da configuração de acesso às bases de dados, a geração da ontologia local (mapeamento direto), a importação dos vocabulários utilizados na ontologia de domínio, a criação da ontologia de aplicação usando um vocabulário que é subconjunto do vocabulário da ontologia de domínio e a geração do arquivo de mapeamento R2RML para converter dados armazenados em bancos relacionais em triplas RDF.

O restante deste artigo está organizado da seguinte forma: a seção 2 apresenta trabalhos relacionados. A seção 3 contextualiza o problema introduzindo um passo a passo para resolvê-lo. A seção 4 introduz a linguagem de geração de mapeamentos padrão da W3C: R2RML. A seção 5 apresenta a arquitetura proposta para a ferramenta, seus componentes principais e o estado atual do trabalho. A seção 6 apresenta um estudo de caso. A seção 7 conclui expondo oportunidades de trabalhos futuros que foram gerados com essa pesquisa.

2. Trabalhos Relacionados

Apesar de ser uma área relativamente nova, já existem vários trabalhos que colaboram com a publicação de dados no padrão Linked Data. Dentre os principais estão:

- Lod2: projeto de integração de dados em larga escala. Propõe ferramentas e metodologias corporativas para publicar e gerenciar grandes quantidades de informações estruturadas na web de dados, algoritmos de aprendizagem de máquina para interligar e fundir dados na web de forma automática, padrões e métodos para monitoramento confiável da proveniência, privacidade e segurança dos dados e ferramentas para busca, navegação e autoria de Linked Data. Possui um componente denominado Lod2 Stack que define um ciclo de vida de Linked Data. Nesse ciclo há uma fase de extração e carga que utiliza o D2R [11] para extrair dados SQL.
- Sig.ma: aplicação que realiza mashups de vários datasets e disponibiliza acesso interativo a “Web de Dados” como um todo. É definido como sendo um navegador, um gerador de mashups e uma API para a web de dados. Nela é possível não somente pesquisar mas também compartilhar dados através da submissão de URLs que são incluídas nos mashups.
- OntoWiki: ferramenta que possui modos sofisticados para navegar, visualizar e criar bases de conhecimento baseadas em RDF. Ela serve e consome Linked Data e tem uma API para criação de aplicações web semânticas.
- Babel: Promove a conversão de bases relacionais, planilhas e outras fontes de dados em dados nos formatos OWL, RDFa, RSS, (X)HTML, RDF/XML e N3. Sugere a criação de templates simples para mapear qualquer tipo de fonte de dados para RDF. O argumento principal é que é mais fácil aprender a criar os templates do que aprender a criar os mapeamentos R2RML.

3. Contextualização do Problema

Para publicarmos fontes de dados relacionais na Web, não basta fazermos simples transformações de formato do dado: de Relacional para RDF. É preciso publicar dados que estejam de acordo com os princípios do Linked Data [2]. Para realizar essa tarefa foram criadas linguagens de mapeamentos complexas utilizadas por várias ferramentas que exportam dados RDF na WEB.

A vasta diversidade de linguagens de mapeamento e a complexidade delas, dificulta a exportação dos dados relacionais para RDF. Com a finalidade de tornar mais intuitivo esse trabalho e facilitar a geração dos mapeamentos, definimos os seguintes passos:

1. Primeiro criamos uma ontologia da fonte aplicando mapeamento direto da base relacional para RDF, gerando um esboço dos dados a serem publicados.
2. Então selecionamos uma ontologia de domínio, que modela o domínio da aplicação. De fato, a ontologia de domínio pode ter uma combinação de ontologias cobrindo domínios distintos. Nesse passo são selecionados vocabulários já existentes para que possa ser feito um alinhamento de termos, possibilitando o reuso e facilitando a interoperabilidade futura.
3. Dando prosseguimento, criamos um mapeamento de alto nível *fonte-para-domínio* da ontologia fonte para a ontologia de domínio. Esse mapeamento é criado através de regras de alto nível conforme definido em [5].
4. Em seguida, geramos a ontologia de aplicação e criamos o mapeamento R2RML. Nesse passo a ontologia de aplicação é criada a partir das regras definidas no passo anterior, onde vocabulário utilizado é um subconjunto do vocabulário da ontologia de domínio. Por fim, aplicamos regras de transição entre os mapeamentos criados no passo 1 e as regras que definem a ontologia de aplicação para gerar o mapeamento final R2RML.

Nesse trabalho é definida uma metodologia para geração dos mapeamentos R2RML. A definição dessa metodologia e sua implementação na forma de uma ferramenta são as principais contribuições desse trabalho. Com o uso da ferramenta será possível gerar mapeamentos de uma forma mais fácil e produtiva.

4. Fundamentação Teórica

4.1. Introdução ao D2R Server

O D2R Server é uma ferramenta para publicar o conteúdo de bases de dados relacionais na Web Semântica, um espaço de informação global consistindo de dados ligados.

Ele pode ser acessado de 3 formas diferentes conforme mostra a figura abaixo:

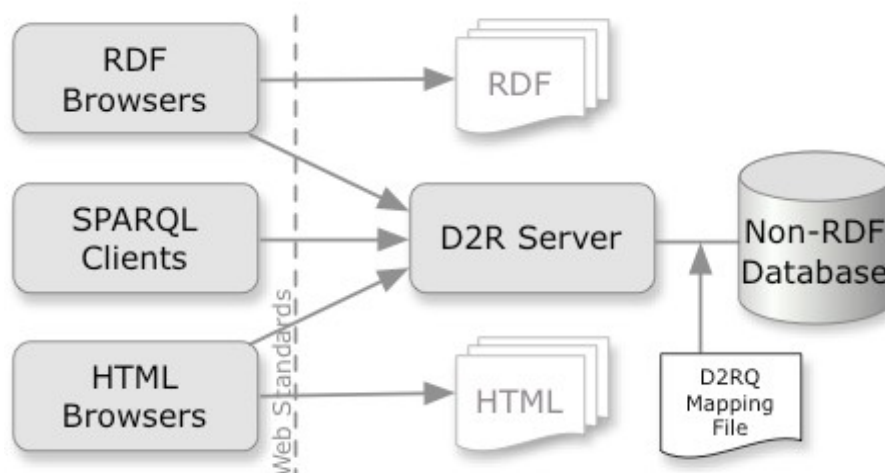


Figura 1. Formas de acesso ao D2R Server

Sua interface Linked Data torna as descrições RDF de recursos individuais disponíveis no protocolo HTTP, a interface SPARQL torna possível que aplicações pesquisem e consultem o banco de dados usando a linguagem de consulta SPARQL sobre o protocolo SPARQL e uma interface HTML que oferece acesso a navegadores Web conhecidos.

Requisições oriundas da Web são reescritas em consultas SQL usando o mapeamento. Essa tradução “on-the-fly” permite a publicação de grandes bases de dados no formato RDF e elimina a necessidade de replicação desses dados.

Nesse trabalho tentamos reutilizar as funcionalidades já implementadas pelo D2R para criação do Endpoint SPARQL.

4.2. Introdução ao R2RML (RDB to RDF Mapping Language)

Uma linguagem para criação de mapeamentos de dados no modelo relacional para dados no formato RDF, possibilitando que o autor do mapeamento escolha vocabulário e estrutura adequados para expressar os dados.

Cada mapeamento R2RML é adaptado para um esquema de banco de dados específico e para um vocabulário alvo. A entrada para um mapeamento R2RML é um banco de dados relacional que está de acordo com esse esquema. A saída é um dataset RDF [SPARQL], conforme definido no SPARQL, que usa predicados e tipos do vocabulário alvo. O mapeamento é conceitual; processadores R2RML são livres para materializar os dados de saída, ou para disponibilizá-los de forma virtual através de uma interface que consulta o banco de dados de origem, ou para oferecer qualquer outra forma de acesso ao dataset RDF resultante.

Um R2RML mapping define um mapeamento a partir de um banco de dados relacional para um dataset RDF. Esse mapeamento é representado como um grafo RDF. Em outras palavras, RDF é usado não apenas como o modelo de dados alvo do mapeamento, mas também como um formalismo para representar o mapeamento R2RML. Um documento de mapeamento R2RML é qualquer documento escrito na sintaxe Turtle [6] RDF, subconjunto da notação N3.

Um mapeamento R2RML faz referência a tabelas lógicas para obter dados do banco de dados de entrada. Uma tabela lógica pode ser dos seguintes tipos:

1. Uma tabela do esquema do banco relacional.
2. Uma visão do esquema do banco relacional.
3. Uma consulta SQL válida chamada de “R2RML view” criada dentro do mapeamento.

Cada tabela lógica é mapeada para RDF usando um “triples map”. O triples map é uma regra que mapeia cada linha da tabela lógica para uma ou mais triplas RDF. A regra tem duas partes principais:

1. Um “subject map” que gera o subject de todas as triplas RDF que serão geradas de uma tabela lógica. Os subjects normalmente são IRIs [7] que são geradas a partir das colunas que são chave primária da tabela.
2. Vários “predicate-object maps” que de fato consistem de “predicate maps” e “object maps” (ou referências para object maps).

As triplas são produzidas pela combinação do subject map com um predicate map e um object map, e aplicando esses três para cada linha da tabela lógica.

Como o R2RML é a linguagem padrão adotada pela W3C, resolvemos gerar nossos mapeamentos nessa notação.

5. A Ferramenta

Nessa seção introduzimos os principais componentes e as funcionalidades da ferramenta web para criação dos mapeamentos R2RML. A seguir definimos a arquitetura da solução e seus principais componentes.

5.1. Arquitetura Proposta

Com o intuito de aproveitar as várias soluções já existentes, resolvemos criar a camada de apresentação utilizando xxx juntamente com o framework JENA [12] para geração das triplas. Criamos uma interface para gerar os mapeamento R2RML.

A figura abaixo descreve de forma gráfica a arquitetura escolhida e as dependências entre seus componentes:

Figura 2. A arquitetura escolhida

5.2. Componentes Principais

Uma das principais contribuições desse trabalho é tornar fácil para o usuário exportar seus dados relacionais em RDF através de uma interface amigável que esconde a complexidade de geração e manutenção dos mapeamentos.

Para tornar isso possível, é preciso criar mecanismos para:

1. Criação de Ontologias de Domínio;
2. Configuração de acesso ao banco de dados via D2R;
3. Geração de mapeamentos R2RML;
4. Tradução de R2RML para D2R Mapping Language;
5. Criação do Endpoint SPARQL via D2R Server;

Cada um desses componentes possui seus algoritmos específicos para implementar o mecanismo ao qual ele está proposto a atender. Destacamos aqui o componente de geração dos mapeamentos R2RML que soluciona como representar as possíveis regras de mapeamento nesse formato, e o componente de tradução de R2RML para D2RM, que define um DE-PARA complexo para tratar tal transformação.

5.3. Estado Atual

Com a arquitetura definida, o trabalho encontra-se na fase de especificação dos algoritmos a serem implementados.

6. Estudo de Caso

Esta seção apresenta a aplicação da nossa metodologia de geração do mapeamento R2RML nos exemplos utilizados por [5] que são ontologias locais do Ebay e da Amazon exportadas para uma ontologia de domínio de lojas virtuais chamada Sales.

Porém, antes de iniciarmos nosso exemplo introduzimos a estratégia proposta por [5] para criação das regras de mapeamento entre ontologias.

A estratégia proposta tem três passos: (1) *matching de vocabulário*, que gera um conjunto de correspondências entre entidades de cada ontologia local e a ontologia de domínio. Nesse passo, a contribuição é um modelo de matching que captura o resultado de um processo de matching já existente. Não é proposto um novo algoritmo para realizar matching; (2) o *post-matching*, que valida as correspondências obtidas no passo (1) e ajuda a ajustar algumas correspondências que não podem ser capturados por algoritmos de matching tradicionais; e (3) *geração das ontologias de aplicação e dos mapeamentos*, que primeiro deduz o conjunto de regras de mapeamento LO-DO (Local Ontology – Domain Ontology) a partir das correspondências obtidas no passo (2), e então usa essas regras para gerar as ontologias de aplicação e um conjunto de mapeamentos (mapeamentos mediados e mapeamentos Local Ontology – Application Ontology).

Essa abordagem não trata somente correspondências homogêneas entre classes e entre propriedades, mas com ela é possível capturar outros tipos de correspondências, resultantes de heterogeneidades estruturais (chamadas em [5] de correspondências de caminho). Essas correspondências são obtidas no passo de post-matching e o resultado delas é usado para gerar o conjunto de *mapeamentos operacionais*.

Nas subseções que seguem apresentamos o formalismo de mapeamento baseado em regras, o exemplo utilizado no estudo de caso, e a aplicação da nossa metodologia de geração de mapeamentos R2RML.

6.1. Formalismo de Mapeamentos baseado em Regras

Seja F um conjunto de símbolos funcionais, B um conjunto de conceitos atômicos e papéis atômicos, e U um conjunto de variáveis. O conjunto de termos sobre F e U é definido recursivamente da seguinte forma:

- i. cada variável v em U é um termo;
- ii. cada constante c em F é um termo;
- iii. Se t_1, \dots, t_n são termos, e f é uma função n -ária em F , então $f(t_1, \dots, t_n)$ é um termo.

Um átomo sobre F , B e U é uma expressão da forma $c(t)$, onde c é um conceito atômico e t é um termo, ou da forma $p(t, u)$, onde p é um papel atômico e t e u são termos.

Faça O_S e O_T serem duas ontologias e R uma linguagem de regras. Em [5], a relação entre um conceito de O_S e um conceito de O_T (chamado de mapeamento de conceito) é especificado por um conjunto de *regras de mapeamento* da forma

$$\beta_1(\omega_1) \leq \alpha_1(v_1), \dots, \alpha_m(v_m), \text{ onde:}$$

$\alpha_1(v_1), \dots, \alpha_m(v_m)$, chamado de corpo da regra de mapeamento, é um átomo ou uma

conjunção de átomos, onde $\alpha_i(v_i)$ é um átomo cujo conceito atômico ou papel atômico ocorre na ontologia fonte O_S ;

$\beta_1(\omega_1)$, chamado de cabeça da regra de mapeamento, é um átomo cujo conceito atômico ou papel atômico ocorre na ontologia alvo O_T .

Esse formalismo baseado em regras suporta *Skolem functions* [8] para criação de novos OID's de classes na O_T de uma ou mais propriedades da O_S . Em [5] Skolem functions são usados simplesmente para geração de URIref. Assim, essas regras de mapeamento permitem a construção de URIrefs para novos objetos em O_T como termos na forma $f(t_1, \dots, t_n)$, onde f é um símbolo de função n -ária e t_1, \dots, t_n é uma sequência de termos de O_S .

6.2. Esquemas Extralite

Nesse trabalho, consideramos a família de ontologias extralite (ou esquemas extralite [9]). Uma ontologia extralite inclui a definição de classes, propriedades do tipo datatype e object, e admite restrições do tipo domain e range, cardinalidades mínima e máxima, subconjuntos e disjunção. Formalmente, uma ontologia extralite é um par $s = (A, C)$ tal que:

- A é um alfabeto, dito vocabulário de s , cujos conceitos atômicos são chamados de classes de s e os papéis atômicos são chamados de propriedades de s .
- C é um conjunto de fórmulas, ditas restrições sobre s , onde cada restrição possui uma das seguintes formas:
 - Domain: $\exists P \subseteq D$ (a propriedade P tem domínio D)
 - Range: $\exists P^- \subseteq R$ (a propriedade P tem imagem R)
 - Cardinalidade Mínima: $D \subseteq (\geq kP)$, onde D é domínio de P
(a propriedade P mapeia cada indivíduo do domínio D em pelo menos k indivíduos distintos)
 - Cardinalidade Máxima: $D \subseteq (\leq kP)$, onde D é domínio de P
(a propriedade P mapeia cada indivíduo do domínio D em no máximo k indivíduos distintos)
 - Subconjunto: $C \subseteq D$ (a classe C é subclasse da classe D)
 - Disjunção: $C \mid D$ (a classe C é disjunta com a classe D)

Por fim, é definido que: uma classe c domina uma classe d no esquema s [9] sse $c = d$, ou existe uma sequência (c_1, c_2, \dots, c_n) tal que $c = c_1$, $d = c_n$ e c_{i-1} subordina c_n , e para cada $i \in [1, n - 2)$, ou

- c_{i+1} e c_i são classes e c_{i+1} é uma subclasse de c_i em s , ou
- c_{i+1} é um object property em s , cujo domínio é c_i , ou
- c_i é um object property em s , cuja imagem é c_{i+1} .

Podemos afirmar que $\pi = (c_1, c_2, \dots, c_n)$ é um caminho de dominância [9] de c para d e $\theta = (pk_1, pk_2, \dots, pk_n)$, a subsequência de π consistindo de object properties que ocorrem em π , é o caminho de propriedade correspondente para $\pi(\theta)$ pode ser uma sequência

vazia).

6.3. Exemplo

Essa seção apresenta o exemplo de um acesso mediado a lojas virtuais que vendem livros na web. Assumimos que temos as bases de dados relacionais criadas e a ontologia de domínio selecionada (veja figura 6.1). No fim teremos a base de dados relacional exportada para a ontologia de domínio no formato RDF.

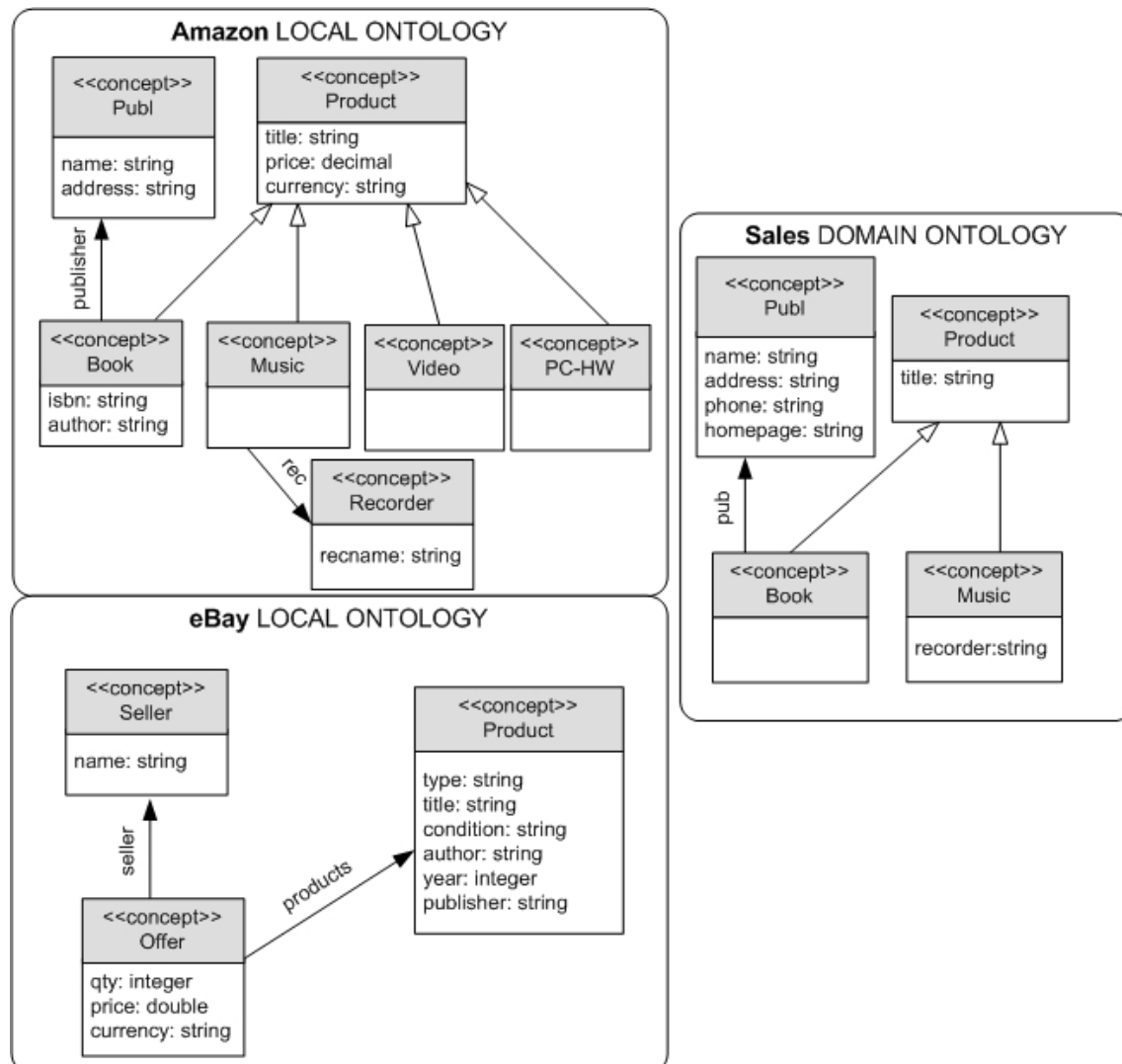


Figura 6.1

Inicialmente criamos o mapeamento R2RML direto das bases de dados para as Ontologias Locais seguindo [3]. Em seguida, definimos os tipos de mapeamentos possíveis entre os objetos da base de dados e os elementos da ontologia local (2),

utilizamos as regras definidas em [2] para mapeamentos entre ontologias local e de domínio (3), identificamos os tipos de mapeamentos entre ontologias (4). Por fim, substituímos em (3) os elementos da ontologia local pelos elementos do banco de dados utilizando os mapeamentos definidos em (1). Resultando em um mapeamento final da base de dados para a Ontologia de Domínio. Desse resultado, extraímos as generalizações de mapeamentos entre bases de dados relacionais e ontologias de domínio.

(1) Mapeamento Direto das bases de dados para as ontologias locais:

Base Amazon

Abaixo listamos o mapeamento direto da tabela lógica “book” para um novo vocabulário local chamado “vocab”:

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix vocab: <http://localhost/vocab/resource/> .

<#TriplesMap1>
  rr:logicalTable [ rr:tableName "book" ];
  rr:subjectMap [
    rr:template "http://www.amazon.com/book/{title}";
    rr:class vocab:Book;
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:title;
    rr:objectMap [ rr:column "title" ];
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:price;
    rr:objectMap [ rr:column "price" ];
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:currency;
    rr:objectMap [ rr:column "currency" ];
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:isbn;
    rr:objectMap [ rr:column "isbn" ];
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:author;
    rr:objectMap [ rr:column "author" ];
  ];
  rr:predicateObjectMap [
    rr:predicate vocab:price;
    rr:objectMap [ rr:column "price" ];
  ];
  ].
```

Para criarmos a IRI usamos a IRI base “<http://www.amazon.com/book/>” concatenado com a chave primária da tabela book. A tabela lógica dá nome à classe da Ontologia. Em seguida, definimos para cada coluna que não é chave primária nem estrangeira um predicado e um objeto que representam um data property da classe Book.

<i>Tipo no BD</i>	<i>Elemento BD</i>	<i>Elemento RDF</i>	<i>Tipo RDF</i>
-------------------	--------------------	---------------------	-----------------

Tabela	book	a:Book	Subclasse
Coluna	book.title	a:title	Data Property
Coluna (FK)	book.publisher	a:publisher	Object Property
Tabela	publisher	a:Publ	Classe
Coluna	publisher.name	a:name	Data Property
Coluna	publisher.address	a:address	Data Property
Tabela	product	a:Product	Classe
Coluna	product.title	a:title	Data Property
Tabela	music	a:Music	Subclasse
Coluna (FK)	music.rec	a:rec	Object Property
Tabela	recorder	a:Recorder	Classe
Coluna	recorder.recname	a:recname	Data Property

Ebay

<i>Tipo no BD</i>	<i>Elemento BD</i>	<i>Elemento RDF</i>	<i>Tipo RDF</i>
Tabela	product	e:Product	Classe
Coluna	product.type	e:type	Data Property
Coluna	product.title	e:title	Data Property
Coluna	product.publisher	e:pub	Object Property
Coluna	product.publisher	e:Publ	Classe
Coluna	product.publisher	e:name	Data Property

(2) Tipos de mapeamentos identificados entre bases de dados e ontologias locais:

#1	$C_L \leq T$	Classe Local \leq Tabela
#2	$S_L \leq T$	Subclasse Local \leq Tabela
#3	$DP_L \leq C_O$	Data Property \leq Coluna
#4	$OP_L \leq C_O$	Object Property \leq Coluna
#5	$C_L \leq C_O$	Classe Local \leq Coluna
#6	$OP_L \leq C_{O, FK}$	Object Property \leq Coluna Foreign Key

(3) Regras para mapeamento da Ontologia Local para a de Domínio extraídas de [2]:

Amazon

#1	$S_D \leq S_L$	$s : \text{Book}(b) \leq a : \text{Book}(b)$
#2	$C_D \leq S_L$	$s : \text{Product}(b) \leq a : \text{Book}(b)$
#3	$S_D \leq C_L$	$s : \text{Music}(m) \leq a : \text{Music}(m)$

#4	$C_D \leq S_L$	$s : \text{Product}(m) \leq a : \text{Music}(m)$
#5	$C_D \leq C_L$	$s : \text{Publ}(p) \leq a : \text{Publ}(p)$
#6	$DP_D \leq DP_L$	$s : \text{title}(b; t) \leq a : \text{title}(b, t); a : \text{Book}(b)$
#7	$OP_D \leq OP_L$	$s : \text{pub}(b; p) \leq a : \text{publisher}(b, p); a : \text{Book}(b)$
#8	$DP_D \leq DP_L$	$s : \text{title}(m; t) \leq a : \text{title}(m, t); a : \text{Music}(m)$
#9	$DP_D \leq DP_L$	$s : \text{name}(p; n) \leq a : \text{name}(p, n); a : \text{Publ}(p)$
#10	$DP_D \leq DP_L$	$s : \text{address}(p; a) \leq a : \text{address}(p, a); a : \text{Publ}(p)$
#11	$DP_D \leq OP_L - DP_L$	$s : \text{recorder}(m; n) \leq a : \text{rec}(m, r); a : \text{recname}(r, n); a : \text{Recorder}(r)$

Ebay

#1	$S_D \leq C_{L,R}$	$s : \text{Book}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$
#2	$C_D \leq C_{L,R}$	$s : \text{Product}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$
#3	$S_D \leq C_{L,R}$	$s : \text{Music}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$
#4	$C_D \leq C_{L,R}$	$s : \text{Product}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$
#5	$DP_D \leq DP_{L,R}$	$s : \text{title}(p, t) \leq e : \text{title}(p, t), e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$
#6	$DP_D \leq DP_{L,R}$	$s : \text{title}(p, t) \leq e : \text{title}(p, t), e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$
#7	$C_D \leq DP_{L,R}$	$s : \text{Publ}(\text{fpubl}(n)) \leq e : \text{publisher}(b, n); e : \text{Product}(b);$ $e : \text{type}(b) = \text{'book'}$
#8	$DP_D \leq DP_{L,R}$	$s : \text{name}(\text{fpubl}(n), n) \leq e : \text{publisher}(b, n); e : \text{Product}(b);$ $e : \text{type}(b) = \text{'book'}$
#9	$OP_D \leq DP_{L,R}$	$s : \text{pub}(b, \text{fpubl}(n)) \leq e : \text{publisher}(b, n); e : \text{Product}(b);$ $e : \text{type}(b) = \text{'book'}$

(4) Tipos de Mapeamento entre Ontologias Locais e de Domínio identificados:

#1	$C_D \leq C_L$	Classe Domínio \leq Classe Local
#2	$C_D \leq S_L$	Classe Domínio \leq Subclasse Local
#3	$S_D \leq S_L$	Subclasse Domínio \leq Subclasse Local
#4	$DP_D \leq DP_L$	Data Property Domínio \leq Data Property Local
#5	$OP_D \leq OP_L$	Object Property Domínio \leq Object Property Local
#6	$DP_D \leq OP_L - DP_L$	Data Property Domínio \leq caminho de Object Property Local para Data Property Local
#7	$S_D \leq C_{L,R}$	Subclasse Domínio \leq Classe Local com Restrição
#8	$C_D \leq C_{L,R}$	Classe Domínio \leq Classe Local com Restrição
#9	$DP_D \leq DP_{L,R}$	Data Property Domínio \leq Data Property Local com Restrição

#10	$C_D \leq DP_{L,R}$	Classe Domínio \leq Data Property Local com Restrição
#11	$OP_D \leq DP_{L,R}$	Object Property Domínio \leq Data Property Local com Restrição

(5) Usando transição entre as regras para substituir os elementos das ontologias locais pelos objetos do banco de dados:

#	$O_L \leq BD$	$O_D \leq O_L$	$O_D \leq BD$
#1	$S_L \leq T$ $a : \text{Book}(b) \leq \text{book}$	$S_D \leq S_L$ $s : \text{Book}(b) \leq a : \text{Book}(b)$	$S_D \leq T$ $s : \text{Book}(b) \leq \text{book}$
#2	$S_L \leq T$ $a : \text{Book}(b) \leq \text{book}$	$C_D \leq S_L$ $s : \text{Product}(b) \leq a : \text{Book}(b)$	$C_D \leq T$ $s : \text{Product}(b) \leq \text{book}$
#3	$S_L \leq T$ $a : \text{Music}(m) \leq \text{music}$	$S_D \leq S_L$ $s : \text{Music}(m) \leq a : \text{Music}(m)$	$S_D \leq T$ $s : \text{Music}(m) \leq \text{music}$
#4	$S_L \leq T$ $a : \text{Music}(m) \leq \text{music}$	$C_D \leq S_L$ $s : \text{Product}(m) \leq a : \text{Music}(m)$	$C_D \leq T$ $s : \text{Product}(m) \leq \text{music}$
#5	$C_L \leq T$ $a : \text{Publ}(p) \leq \text{publisher}$	$C_D \leq C_L$ $s : \text{Publ}(p) \leq a : \text{Publ}(p)$	$C_D \leq T$ $s : \text{Publ}(p) \leq \text{publisher}$
#6	$DP_L \leq T.C_o$ $a : \text{title}(b, t); a : \text{Book}(b) \leq \text{book.title}$	$DP_D \leq DP_L$ $s : \text{title}(b; t) \leq a : \text{title}(b, t); a : \text{Book}(b)$	$DP_D \leq T.C_o$ $s : \text{title}(b; t) \leq \text{book.title}$
#7	$OP_L \leq T.C_{o,FK}$ $a : \text{publisher}(b, p); a : \text{Book}(b) \leq \text{book.publisher}$	$OP_D \leq OP_L$ $s : \text{pub}(b; p) \leq a : \text{publisher}(b, p); a : \text{Book}(b)$	$OP_D \leq T.C_{o,FK}$ $s : \text{pub}(b; p) \leq \text{book.publisher}$
#8	$DP_L \leq T.C_o$ $a : \text{title}(m, t); a : \text{Music}(m) \leq \text{music.title}$	$DP_D \leq DP_L$ $s : \text{title}(m; t) \leq a : \text{title}(m, t); a : \text{Music}(m)$	$DP_D \leq T.C_o$ $s : \text{title}(m; t) \leq \text{music.title}$
#9	$DP_L \leq T.C_o$ $a : \text{name}(p, n); a : \text{Publ}(p) \leq \text{publisher.name}$	$DP_D \leq DP_L$ $s : \text{name}(p; n) \leq a : \text{name}(p, n); a : \text{Publ}(p)$	$DP_D \leq T.C_o$ $s : \text{name}(p; n) \leq \text{publisher.name}$
#10	$DP_L \leq T.C_o$ $a : \text{address}(p, a); a : \text{Publ}(p) \leq \text{publisher.address}$	$DP_D \leq DP_L$ $s : \text{address}(p; a) \leq a : \text{address}(p, a); a : \text{Publ}(p)$	$DP_D \leq T.C_o$ $s : \text{address}(p; a) \leq \text{publisher.address}$
#11	$OP_L \leq T.C_{o,FK}; DP_L \leq T.C_o$ $a : \text{rec}(m, r) \leq \text{music.rec};$ $a : \text{recname}(r, n); a : \text{Recorder}(r) \leq \text{recorder.recname}$	$DP_D \leq OP_L; DP_L$ $s : \text{recorder}(m; n) \leq a : \text{rec}(m, r); a : \text{recname}(r, n);$ $a : \text{Recorder}(r)$	$DP_D \leq T.C_{o,FK}; T.C_o$ $s : \text{recorder}(m; n) \leq \text{music.rec};$ recorder.recname

Tabela : Transformação dos mapeamentos da base Amazon.

#	$O_L \leq BD$	$O_D \leq O_L$	$O_D \leq BD$
#1	$C_{L,R} \leq T_R$ $e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$ $\leq \text{product}, \text{product.type} = \text{'book'}$	$S_D \leq C_{L,R}$ $s : \text{Book}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$	$S_D \leq T_R$ $s : \text{Book}(p) \leq \text{product}, \text{product.type} = \text{'book'}$

#2	$C_{L,R} \leq T_R$ $e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$ $\leq \text{product}, \text{product.type} = \text{'book'}$	$C_D \leq C_{L,R}$ $s : \text{Product}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'book'}$	$C_D \leq T_R$ $s : \text{Product}(p) \leq$ $\text{product}, \text{product.type} = \text{'book'}$
#3	$C_{L,R} \leq T_R$ $e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$ $\leq \text{product}, \text{product.type} = \text{'music'}$	$S_D \leq C_{L,R}$ $s : \text{Music}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$	$S_D \leq T_R$ $s : \text{Music}(p) \leq$ $\text{product}, \text{product.type} = \text{'music'}$
#4	$C_{L,R} \leq T_R$ $e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$ $\leq \text{product}, \text{product.type} = \text{'music'}$	$C_D \leq C_{L,R}$ $s : \text{Product}(p) \leq e : \text{Product}(p), e : \text{type}(p) = \text{'music'}$	$C_D \leq T_R$ $s : \text{Product}(p) \leq$ $\text{product}, \text{product.type} = \text{'music'}$
#5	$DP_{L,R} \leq T.C_O$ $e : \text{title}(p, t), e : \text{Product}(p),$ $e : \text{type}(p) = \text{'book'} \leq$ product.title	$DP_D \leq DP_{L,R}$ $s : \text{title}(p, t) \leq e : \text{title}(p, t), e : \text{Product}(p),$ $e : \text{type}(p) = \text{'book'}$	$DP_D \leq T.C_O$ $s : \text{title}(p, t) \leq \text{product.title}$
#6	$DP_{L,R} \leq T.C_O$ $e : \text{title}(p, t), e : \text{Product}(p),$ $e : \text{type}(p) = \text{'music'} \leq$ product.title	$DP_D \leq DP_{L,R}$ $s : \text{title}(p, t) \leq e : \text{title}(p, t), e : \text{Product}(p),$ $e : \text{type}(p) = \text{'music'}$	$DP_D \leq T.C_O$ $s : \text{title}(p, t) \leq \text{product.title}$
#7	$DP_{L,R} \leq T.C_O$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$ \leq product.publisher	$C_D \leq DP_{L,R}$ $s : \text{Publ}(f_{\text{publ}}(n)) \leq$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$	$C_D \leq T.C_O$ $s : \text{Publ}(f_{\text{publ}}(n)) \leq$ product.publisher
#8	$DP_{L,R} \leq T.C_O$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$ \leq product.publisher	$DP_D \leq DP_{L,R}$ $s : \text{name}(f_{\text{publ}}(n), n) \leq$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$	$DP_D \leq T.C_O$ $s : \text{name}(f_{\text{publ}}(n), n) \leq$ product.publisher
#9	$DP_{L,R} \leq T.C_O$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$ \leq product.publisher	$OP_D \leq DP_{L,R}$ $s : \text{pub}(b, f_{\text{publ}}(n)) \leq$ $e : \text{publisher}(b, n);$ $e : \text{Product}(b); e : \text{type}(b) = \text{'book'}$	$OP_D \leq T.C_O$ $s : \text{pub}(b, f_{\text{publ}}(n)) \leq$ product.publisher

Tabela : Transformação dos mapeamentos da base Ebay.

(6) Tipos de mapeamentos identificados entre bases de dados e ontologias de domínio. Algumas adaptações foram feitas para gerar os mapeamentos D2R.

Por exemplo: $S_D \leq T$ foi convertido para $S_D \leq T.PK$

#1	$S_D \leq T.PK$	Subclasse Domínio \leq Tabela.PrimaryKey
----	-----------------	--

#2	$C_D \leq T$	Classe Domínio \leq Tabela
#3	$D P_D(C) \leq T.C_O$	Data Property Domínio (Classe Pai) \leq Tabela.Coluna
#4	$OP_D \leq T.C_{O, FK}; T.C_{O, PK}$	Object Property Domínio \leq Tabela.Coluna (FK); Tabela.Coluna (PK)
#5	$DP_D \leq T.C_O ;$ $T.C_{O, FK} ; T.C_{O, PK}$	Data Property Domínio \leq Tabela.Coluna; Tabela.Coluna (FK) ; Tabela.Coluna(PK)
#6	$S_D \leq T.PK_R$	Subclasse Domínio \leq Tabela.PrimaryKey com restrição
#7	$C_D \leq T_R$	Classe Domínio \leq Tabela com restrição
#8	$C_D \leq T.C_O$	Classe Domínio \leq Tabela.Coluna
#9	$OP_D \leq T.C_O$	Object Property Domínio \leq Tabela.Coluna

(7) Exemplos de geração de mapeamentos D2R para cada tipo definido em (6):

$S_D \leq T.PK$ $s:Book \leq book.title$	map:book a d2rq:ClassMap; d2rq:dataStorage map:database; d2rq:uriPattern "book/@@book.title urlify@"; d2rq:class $s:Book$; d2rq:classDefinitionLabel "book"; .
$C_D \leq T$	-
$DP_D(C) \leq T.C_O$ $s:title(map:book) \leq book.title$	map:book_title a d2rq:PropertyBridge; d2rq:belongsToClassMap map:book ; d2rq:property $s:title$; d2rq:column " book.title "; .
$OP_D \leq T.C_{O, FK}; T.C_{O, PK}$ $s:pub \leq book.publisher;$ $publisher.name$	map:book_pub a d2rq:PropertyBridge; d2rq:belongsToClassMap map:book; d2rq:property $s:pub$; d2rq:refersToClassMap map:publisher; d2rq:join " book.publisher => publisher.name "; .
$DP_D \leq$ $T.C_O ; T.C_{O, FK} ; T.C_{O, PK}$ $s:recorder \leq$ $recorder.recname;$ $music.rec;$ $recorder.recname$	map:music_rec a d2rq:PropertyBridge; d2rq:belongsToClassMap map:music; d2rq:property $s:recorder$; d2rq:column "recorder.recname"; d2rq:join " music.rec => recorder.recname "; .

$S_D \leq T.PK_R$ s:Book <= product.title; product.type='book'	map:book a d2rq:ClassMap; d2rq:dataStorage map:database; d2rq:uriPattern "book/@@ product.title urlify@"; d2rq:class s:Book ; d2rq:condition " product.type='book' "; .
$C_D \leq T_R$	-
$C_D \leq T.C_O$ s:Publ <= product.publisher	map:publisher a d2rq:ClassMap; d2rq:dataStorage map:database; d2rq:uriPattern "publisher/@@ product.publisher urlify@"; d2rq:class s:Publ ; d2rq:containsDuplicates "true"; .
$OP_D \leq T.C_O$ s:pub <= product.publisher	map:book_pub a d2rq:PropertyBridge; d2rq:belongsToClassMap map:book; d2rq:property s:pub ; d2rq:uriPattern "publisher/@@ product.publisher urlify@"; .

7. Trabalhos Futuros

Um vez que a solução abordada até aqui preocupa-se com a simples publicação dos dados relacionais em RDF de forma virtual. Como propostas futuras, pensamos em realizar testes para medir a performance dessas publicações e tentar definir uma estratégia de como publicar os dados ou até mesmo parte deles. Ou seja, queremos descobrir quais subconjuntos dos dados devem ser virtualizados e quais devem ser materializados. Dessa forma definiríamos uma solução híbrida que atenda a requisitos não funcionais de performance nas consultas aos Endpoints SPARQL.

8. Referências

1. Resource Description Framework (RDF) Model and Syntax Specification, W3C Proposed Recommendation 05 January 1999. Disponível em <http://www.w3.org/TR/PR-rdf-syntax/>. Acessado em 2012.
2. Tim Berners-Lee: Linked Data, Design Issues.
Disponível em: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Acessado em 2012.
3. A Direct Mapping of Relational Data to RDF, Marcelo Arenas, Eric Prud'hommeaux, Juan Sequeda, Editors. World Wide Web Consortium, 24 de Março de 2011. Disponível em <http://www.w3.org/TR/rdb-direct-mapping/>.
4. Percy E. Salas, Karin K. Breitman, Marco A. Casanova, José Viterbo: StdTrip: Promoting the Reuse of Standard Vocabularies in Open Government Data.

5. Sacramento, E., Vidal, V. M., Macêdo, J. A., Lóscio, B. F., Lopes, F. L. R., Casanova, M. A., and Lemos, F.: Towards automatic generation of application ontologies. In: Journal of Information and Data Management, Vol. V, No. N, Month 20YY, pp.1-16, 2010.
6. David Beckett, Tim Berners-Lee: Turtle - Terse RDF Triple Language. Disponível em <http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/> .
7. M. Dürst and M. Suignard: Internationalized Resource Identifiers (IRIs), Internet-Draft, June 2003, expires December 2003. This document is <http://www.w3.org/International/iri-edit/draft-duerst-iri-04>.
8. Hull, R. and Yoshikawa, M. Ilog: declarative creation and manipulation of object identifiers. In Proceedings of the 16th International Conference on Very Large Databases (VLDB). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 455#468, 1990.
9. Leme, L. A. P. P., Casanova, M. A., Breitman, K. K., and Furtado, A. L. Instance-based OWL schema matching. In Proceedings of the 11th International Conference on Enterprise Information Systems. Lecture Notes in Business Information Processing, vol. 24. pp. 14-26, 2009.
10. Das Souripriya, Sundara Seema, Cyganiak Richard (2011), R2RML: RDB to RDF Mapping Language. Disponível em <<http://www.w3.org/TR/r2rml/>>
11. Bizer, C., Cyganiak, R.: D2R server – publishing relational databases on the Semantic Web. Disponível em: <<http://www4.wiwi.fu-berlin.de/bizer/pub/Bizer-Cyganiak-D2R-Server-ISWC2006.pdf>>.
12. Jena - A Semantic Web Framework for Java. Disponível em: <<http://jena.sourceforge.net>>