Capítulo 3

Linked Data: Construindo um Espaço de Dados Global na Web

Regis Pires Magalhães, José Antônio F. de Macêdo, Vânia Maria Ponte Vidal

Departamento de Computação, Universidade Federal do Ceará, Fortaleza-CE, Brasil
{regispires, vvidal, jose.macedo}@lia.ufc.br

Abstract

Currently the Web is not only a global space of linked documents. It is also becoming a huge global data space consisting of billions of RDF triples from many different domains, which is called Web of Data. Linked Data defines a set of principles that form the basis for the dissemination and use of the data on the Web. Since 2007 datasets from various domains have been published according to these principles, generating a growing volume of data and hence a demand for their consumption. This chapter provides a conceptual and practical base related to Linked Data, where its foundations are presented and tools for publishing and consuming such data are discussed. The chapter also presents applications that benefit from using data published according to Linked Data principles. It also addresses the state of the art in the area and discuss limitations, open questions and challenges to be overcome in the research domain about Linked Data.

Resumo

A Web atual deixou de ser apenas um espaço global de documentos interligados e está se tornando também um enorme espaço global de dados vinculados constituído de bilhões de triplas RDF que cobrem os mais variados domínios, denominada Web de Dados. Linked Data define um conjunto de princípios que formam a base para a difusão e uso de dados na Web. Desde 2007 conjuntos de dados dos mais diversos domínios têm sido publicadas de acordo com estes princípios, gerando um volume crescente de dados e, consequentemente, uma demanda por seu consumo. Este capítulo provê uma base conceitual e prática relacionada à Linked Data, onde são apresentados os seus fundamentos

e discutidas ferramentas para publicação e consumo de tais dados necessários para o desenvolvimento de aplicações. Apresenta ainda aplicações que se beneficiam do uso de dados publicados de acordo com os princípios Linked Data, além de tratar do estado da arte na área e discutir limitações, questões em aberto e desafios a serem superados no domínio da pesquisa sobre Linked Data.

3.1. Introdução

A Web atual deixou de ser apenas um espaço global de documentos interligados e está se tornando um enorme espaço global de dados vinculados constituído de bilhões de triplas RDF que cobrem os mais variados domínios [Heath and Bizer 2011]. Esta nova Web, denominada Web de Dados, visa pavimentar o caminho para a Web Semântica funcional, onde haverá a disponibilidade de uma grande quantidade de dados vinculados em formato RDF. Sua implementação é baseada nos princípios *Linked Data* delineados pelo diretor geral do W3C, o pesquisador Tim Berners-Lee. De fato, *Linked Data* é um conjunto de melhores práticas para publicação e conexão de dados estruturados na Web.

Inúmeras iniciativas voltadas para fomentar a criação da Web de Dados surgiram nos últimos anos, como por exemplo, o projeto *Linking Open Data (LOD)*[1] que é um esforço comunitário iniciado em 2007 e suportado pelo W3C para identificar fontes de dados publicadas sob licenças abertas, convertê-las para RDF e publicá-las na Web usando os princípios de *Linked Data*. Em outubro de 2010, este projeto havia publicado 207 conjuntos de dados compostos de mais de 28 bilhões de triplas RDF e aproximadamente 395 milhões de *links* RDF englobando os mais variados domínios como informações geográficas, censo, pessoas, empresas, comunidades online, publicações científicas, filmes, músicas, livros, além de outros [Bizer et al. 2011]. A figura 3.1 mostra um diagrama de nuvem com as fontes de dados publicadas pelo projeto LOD e as interligações entre elas em setembro de 2010. O tamanho dos círculos corresponde ao número de triplas de cada fonte de dados. As setas indicam a existência de pelo menos 50 links entre duas fontes. A origem de uma seta indica a fonte que possui o link e a fonte referenciada é a fonte para a qual a seta está apontando. Setas bidirecionais representam fontes que se referenciam mutuamente. A espessura da seta corresponde ao número de links.

Outra importante iniciativa foi a criação do *Workshop Linked Data on the Web* (LDOW)[2] dentro da programação da *International World Wide Web Conference* (WWW2008), tendo, desde então, entre os membros de seu comitê organizador o pesquisador Tim Berners-Lee (W3C/MIT). No Brasil foi estabelecida no âmbito governamental a Infraestrutura Nacional de Dados Abertos (INDA)[3], uma importante iniciativa criada em 2011 para aplicar os princípios de *Linked Data* na publicação de dados governamentais abertos.

A Web de Dados cria inúmeras oportunidades para a integração semântica de dados, fomentando o desenvolvimento de novos tipos de aplicações e ferramentas. Muito esforço tem sido despendido pela comunidade para o desenvolvimento de navegadores, mecanismos de busca e outras ferramentas específicas para consumo de dados vincula-

[1] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[2] http://events.linkeddata.org/ldow2008/
[3] http://wiki.gtinda.ibge.gov.br/

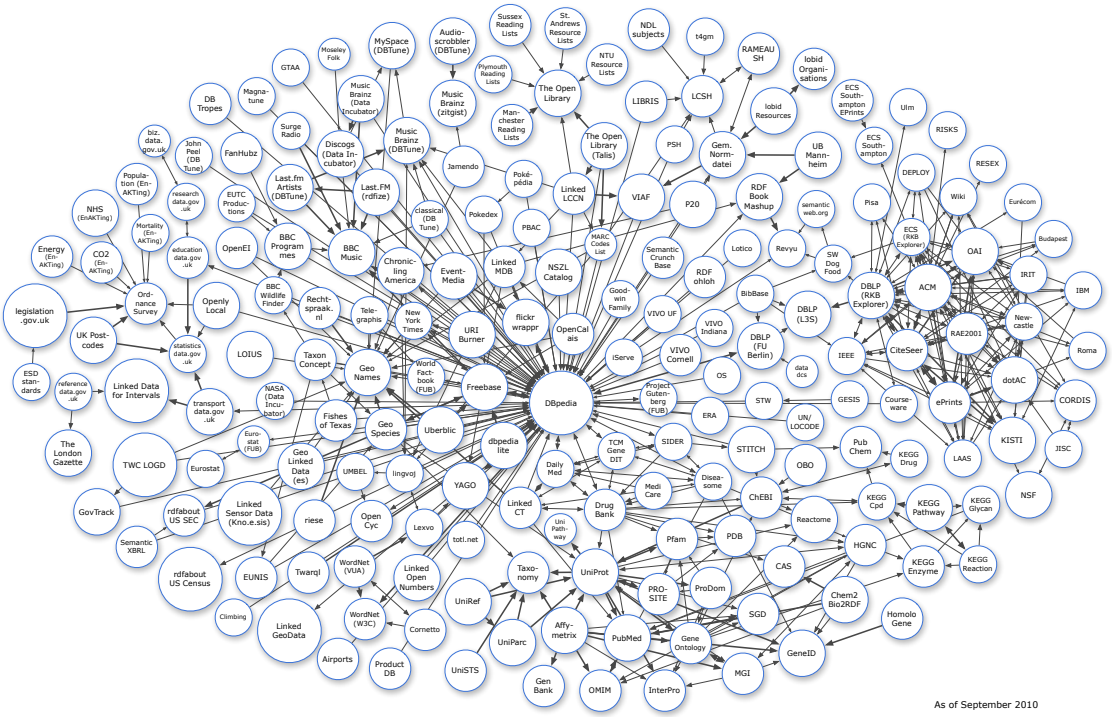

Figura 3.1. Diagrama de nuvem *Linking Open Data*, por Richard Cyganiak e Anja Jentzsch. http://lod-cloud.net/

dos. Além disso, muitas ferramentas para publicação de dados seguindo os princípios de *Linked Data* foram desenvolvidas e disponibilizadas, motivando a publicação de dados de forma aberta. Essas forças têm revelado desafios a serem superados para o uso efetivo da Web de Dados, o que tem aumentado o interesse de pesquisa nesta área.

Este capítulo apresenta uma fundamentação para publicar e consumir dados na Web de acordo com os princípios de *Linked Data*. A seção 3.2 inicia com a apresentação destes princípios e dos conceitos básicos relacionados a eles. As seções 3.3 e 3.4 abordam respectivamente a publicação e o consumo de dados seguindo esses princípios, inclusive apresentando as ferramentas e aplicações atualmente usadas para publicar e consumir *Linked Data*. A seção 3.5 apresenta as limitações das tecnologias existentes e os desafios e oportunidades de pesquisa sobre *Linked Data*. Por fim, a seção 3.6 faz um breve resumo do capítulo e estabelece algumas considerações finais.

3.2. Fundamentos de *Linked Data*

Linked Data é um conjunto de melhores práticas para publicação e consumo de dados estruturados na Web, permitindo estabelecer *links* entre itens de diferentes fontes de dados para formar um único espaço de dados global [Heath and Bizer 2011]. Os dados publicados na Web de acordo com essas melhores práticas podem ser processados por máquinas, possuem significado explicitamente definido e podem estar ligados a outras fontes de dados. [Bizer et al. 2009] resume *Linked Data* como o uso da Web para criar *links* tipados entre dados de diferentes fontes. As melhores práticas relacionadas à *Linked Data* foram inicialmente propostas por [Berners-Lee 2006] e ficaram conhecidas como os princípios

de *Linked Data* que são enumerados a seguir:

1. Usar URIs como nomes para coisas.
2. Usar URIs HTTP para que as pessoas possam procurar esses nomes.
3. Quando alguém procurar uma URI, prover informação útil, usando os padrões (RDF, SPARQL).
4. Incluir links para outras URIs, de modo que possam permitir a descoberta de mais coisas.

Esses princípios fornecem a base para a publicação e interligação de dados estruturados na Web. Posteriormente, eles foram estendidos por documentos originados a partir das experiências da comunidade de *Linked Data* [Bizer et al. 2007b, Sauermann and Cyganiak 2008], resultando em boas práticas de publicação e consumo de *Linked Data* que serão apresentadas ao longo deste capítulo.

Para facilitar o entendimento da Web de dados, podemos estabelecer um paralelo com a Web de documentos que já conhecemos. A Web de dados pode ser acessada a partir de navegadores RDF, assim como os navegadores HTML são usados para acessar a Web de documentos. Enquanto na Web de documentos usamos links HTML para navegar entre diferentes páginas, na Web de dados os links RDF são usados para acessar dados de outras fontes. Portanto, os links de hipertexto são capazes de conectar os documentos, assim como os links RDF interligam os dados.

Além disso, a Web de documentos está alicerçada em um pequeno conjunto de padrões: um mecanismo de identificação global e único (URIs - Uniform Resource Identifiers), um mecanismo de acesso universal (HTTP - Hypertext Transfer Protocol) e um formato de conteúdo amplamente usado (HTML - Hypertext Markup Language). De modo semelhante, a Web de dados também tem por base alguns padrões bem estabelecidos como: o mesmo mecanismo de identificação usado na Web de documentos (URIs), um modelo de dados comum (RDF) e uma linguagem de consulta para acesso aos dados (SPARQL). O modelo RDF [Manola and Miller 2004] é baseado na idéia de identificar os recursos da Web usando identificadores (chamados Uniform Resource Identifiers - URIs[4]), e descrever tais recursos em termo de propriedades, as quais podem apontar para outras URIs ou ser representadas por literais. Esses padrões serão abordados a seguir:

URIs – Uniform Resource Identifiers
URIs [Berners-Lee et al. 2005] são usadas no contexto de *Linked Data* para identificar objetos e conceitos, permitindo que eles sejam dereferenciados para obtenção de informações a seu respeito. Assim, uma URI dereferenciada resulta em uma descrição RDF do recurso identificado. Por exemplo, a URI *http://www.w3.org/People/Berners-Lee/card#i* identifica o pesquisador Tim Bernes-Lee.

RDF – Resource Description Framework
A utilização um modelo de dados comum – modelo RDF – torna possível a implementação de aplicações genéricas capazes de operar sobre o espaço de dados global

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

[Heath and Bizer 2011]. O modelo RDF [Manola and Miller 2004] é um modelo de dados descentralizado, baseado em grafo e extensível, possuindo um alto nível de expressividade e permitindo a interligação entre dados de diferentes fontes. Ele foi projetado para a representação integrada de informações originárias de múltiplas fontes. Os dados são descritos na forma de triplas com sujeito, predicado e objeto, onde o sujeito é uma URI, o objeto pode ser uma URI ou um literal e o predicado é uma URI que define como sujeito e predicado estão relacionados. Por exemplo a afirmação em português *'http://www.w3.org/People/Berners-Lee/card#i tem uma propriedade denominada creator cujo valor é Tim Bernes-Lee'* pode ser definida através de uma tripla RDF da seguinte forma:

```
Sujeito: http://www.w3.org/People/Berners-Lee/card#i
Predicado: 'creator'
Objeto: 'Tim Bernes-Lee'
```

Cada tripla faz parte da Web de Dados e pode ser usada como ponto de partida para explorar esse espaço de dados. Triplas de diferentes fontes podem ser facilmente combinadas para formar um único grafo. Além disso, é possível usar termos de diferentes vocabulários para representar os dados. O modelo RDF ainda permite a representação de dados em diferentes níveis de estruturação, sendo possível representar desde dados semi-estruturados a dados altamente estruturados.

RDF Links

No contexto de *Linked Data* os *RDF links* descrevem relacionamentos entre dois recursos [Heath and Bizer 2011]. Um *RDF link* consiste de três referências URI. As URIs referentes ao sujeito e objeto identificam os recursos relacionados. A URI referente ao predicado define o tipo de relacionamento entre os recursos. Um distincao util que pode ser feita é com relação à links internos e externos. *RDF links* internos conectam recursos dentro de uma unica fonte de dados Linked Data. Links externos conectam recursos qua sao servidos por diferentes fontes de dados Linked Data. No caso de links externos, as URIs referentes ao sujeito e predicado do link pertencem a diferentes namespaces. Links externos são cruciais para a Web dos Dados visto que eles permitem juntar as fontes de dados dispersas em um espaço global de dados.

A figura 3.2 apresenta dois exemplos de *RDF links*. O primeiro exemplo interliga o perfil FOAF do pesquisador Tim Berners-Lee localizado em um arquivo RDF ao recurso que o identifica na fonte de dados do DBLP. No segundo exemplo, o recurso que identifica Tim Berners-Lee na fonte DBpedia também é ligado ao recurso na fonte DBLP que o identifica. A propriedade *http://www.w3.org/2002/07/owl#sameAs* define que os recursos interligados representam a mesma entidade do mundo real.

O armazenamento de dados no modelo RDF pode ser realizado através de grafo em memória, arquivo texto ou banco de dados específico para armazenamento de triplas RDF, chamado de *RDF Triple Store*. O armazenamento de triplas em arquivo texto usa algum formato de serialização de RDF, como RDF/XML, Notation3 (N3), Turtle ou NTriples.

Protocolo e Linguagem SPARQL

Consultas à Web de Dados podem ser realizadas através da linguagem SPARQL

```
Sujeito: http://www.w3.org/People/Berners-Lee/card#i
Predicado: http://www.w3.org/2002/07/owl\#sameAs
Objeto: http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007
```

```
Sujeito: http://dbpedia.org/resource/Tim\_Berners-Lee
Predicado: http://www.w3.org/2002/07/owl\#sameAs
Objeto: http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007
```

Figura 3.2. Exemplos de links RDF

[Prud'hommeaux and Seaborne 2008], que é a linguagem de consulta padrão da Web Semântica para recuperação de informações contidas em grafos RDF. No entanto, SPARQL não é somente uma linguagem de consulta declarativa, mas também um protocolo [Clark et al. 2008] usado para enviar consultas e recuperar resultados através do protocolo HTTP.

Fontes *Linked Data* tipicamente fornecem um *SPARQL endpoint* que é um serviço Web com suporte ao protocolo SPARQL. Esse serviço possui uma URI específica para receber requisições HTTP com consultas SPARQL e retornar os resultados dessas consultas. Os resultados podem ter diferentes formatos. Consultas que usam os comandos SELECT e ASK geralmente são retornadas nos formatos XML, JSON ou texto plano. Já os resultados de consultas através dos comandos DESCRIBE ou CONSTRUCT normalmente usam os formatos RDF/XML, NTriples, Turtle ou N3. Caso o *endpoint* não receba os parâmetros exigidos pelo protocolo SPARQL, ele exibe uma página HTML interativa que permite ao usuário digitar e submeter uma consulta.

Além do uso dos padrões acima descritos, a Web de dados possui as seguintes características de acordo com [Bizer et al. 2009]:

1. É genérica e pode conter qualquer tipo de dado;
2. qualquer pessoa pode publicar dados;
3. não há restrições para seleção de vocabulários;
4. os dados são auto-descritos, de modo que ao dereferenciá-los é possível obter sua definição;
5. o uso de um mecanismo padrão de acesso aos dados (HTTP) e de um modelo de dados padrão (RDF) simplifica o acesso aos dados;
6. as aplicações que usam a Web de dados não ficam limitadas a um conjunto fixo de fontes de dados, podendo inclusive descobrir novas fontes de dados em tempo de execução ao seguir links RDF.

3.3. Publicação de *Linked Data*

Publicar *Linked Data* significa usar os princípios e melhores práticas de *Linked Data* para disponibilizar os dados na Web. Basicamente isso quer dizer que é preciso fornecer URIs

dereferenciáveis para cada entidade e criar links RDF para outras fontes de dados. Esses são os requisitos mínimos, mas além deles é frequente a disponibilização de *SPARQL endpoints* e de *dumps* dos dados.

A publicação de *Linked Data* normalmente envolve a disponibilização de uma **interface *Linked Data*** capaz de tratar requisições de URIs, dereferenciar URIs, tratar dos redirecionamentos 303 requeridos pela arquitetura Web e da negociação de conteúdo entre descrições de um mesmo recurso em diferentes formatos. Através da negociação de conteúdo pode-se retornar a representação mais adequada ao cliente. Assim, um usuário humano pode requisitar uma URI e obter uma representação HTML do recurso. Isso é possível porque os clientes HTTP enviam cabeçalhos indicando que tipo de representação eles preferem obter. Daí o servidor analisa o cabeçalho ao receber uma requisição e seleciona a resposta adequada. Vejamos um exemplo disso ao acessar a URI *http://dblp.l3s.de/d2r/resource/Marco_A._Casanova* de uma fonte DBLP. Se o cabeçalho usado na requisição especificar o conteúdo text/html, o servidor enviará para o cliente um redirecionamento para a URI *http://dblp.l3s.de/d2r/page/Marco_A._Casanova* que possui como conteúdo uma página HTML. Caso seja usado um cabeçalho especificando conteúdo application/rdf+xml, o cliente receberá como resposta um redirecionamento para a URI *http://dblp.l3s.de/d2r/data/Marco_A._Casanova* que contém o resultado do dereferenciamento da URI em formato RDF/XML. Pubby[5] é uma ferramenta para fornecer de forma simples uma interface *Linked Data* para fontes de dados RDF, como *SPARQL endpoints* e arquivos RDF estáticos.

Outro recurso frequentemente relacionado à publicação de *Linked Data* é o fornecimento de um ***SPARQL endpoint*** para possibilitar a realização de consultas SPARQL sobre uma fonte de dados. Alguns *endpoints* permitem inclusive atualizações através de *SPARQL Update* [Gearon et al. 2011]. O projeto Jena[6] disponibiliza duas implementações de *SPARQL endpoints*: o *Joseki*[7] e o *Fuseki*[8]. Ambos suportam o protocolo e a linguagem SPARQL; permitem a realização de consultas sobre arquivos RDF e *RDF Triple Stores*; e implementam SPARQL Update. A diferença principal entre eles é que o *Fuseki* é mais simples de configurar e usar, além de já possuir a *RDF Store Jena TDB*[9] embutida. No entanto o *Fuseki* é um projeto mais recente iniciado em 2011 que ainda possui limitações quanto ao gerenciamento de múltiplas fontes de dados e também quanto ao uso de mecanismos de segurança próprios.

3.3.1. Publicação de dados de fontes RDF como *Linked Data*

As fontes de dados que adotam o modelo RDF são normalmente armazenadas em arquivo RDF ou *RDF Store*. A seleção da forma de armazenamento mais adequada vai depender de fatores como o volume de dados utilizado e a frequência de atualização desses dados. Nesta subseção veremos como publicar arquivos RDF ou *RDF Stores* como *Linked Data*.

Publicação de arquivos RDF como *Linked Data*

[5]http://www4.wiwiss.fu-berlin.de/pubby/
[6]http://incubator.apache.org/jena/
[7]http://www.joseki.org/
[8]http://openjena.org/wiki/Fuseki
[9]http://openjena.org/TDB/

Criar arquivos RDF estáticos e disponibilizá-los através de um servidor Web é a forma mais simples de publicar *Linked Data*. Essa estratégia somente é viável quando o arquivo RDF é relativamente pequeno. Embora haja vários formatos para serialização de um arquivo RDF (RDF/XML, Turtle, Notation3, NTriples), o formato mais antigo e também mais usado para publicação de *Linked Data* é o RDF/XML. Assim, se o arquivo for escrito manualmente em um formato mais simples, compacto e legível para o ser humano como Turtle, deverá ser posteriormente convertido para RDF/XML. A figura 3.3 apresenta um trecho de um arquivo RDF no formato Turtle que possui informações sobre organizações e pesquisadores. Cada pesquisador possui um *RDF link* para o recurso correspondente na fonte de dados DBLP. Interessante observar que o RDF mostrado na figura adota a boa prática de somente usar vocabulários existentes. Nesse caso, fizemos uso de termos dos vocabulários FOAF e DC.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix :     <http://lia.ufc.br/~regispires/researchers.rdf#> .

# The <> (the empty URI) means "this document".
<>  a                   foaf:Document ;
    dc:title            "Researchers file" .

:ufc
    a                   foaf:Organization ;
    foaf:name           "Universidade Federal do Ceará" .

:vania
    a                   foaf:Person ;
    foaf:name           "Vânia Maria Ponte Vidal" ;
    foaf:Organization   :ufc ;
    rdfs:sameAs         <http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal> .
```

Figura 3.3. Arquivo RDF serializado no formato Turtle

A conversão de Turtle para RDF/XML pode ser realizada através de algum conversor online como *Babel*[10] ou *RDFConverter*[11]. A figura 3.4 apresenta o arquivo RDF/XML a ser publicado através de um servidor Web. Os recursos presentes no arquivo RDF podem ser acessados através do uso de *Hash URIs* [Sauermann and Cyganiak 2008] que possuem um identificador de fragmento adicionado ao nome do arquivo. Em nosso exemplo, o identificador de fragmento #vania é adicionado à URI do arquivo para que a URI *http://lia.ufc.br/ regispires/researchers.rdf#vania* possa ser usada para acessar o recurso correspondente. O nome *Hash URI* deve-se ao uso do símbolo # que é representado pela palavra *hash* na língua inglesa.

A disponibilização de um arquivo RDF/XML em um servidor Web possibilita o dereferenciamento de URIs e o uso de *RDF links*, mas ainda não resolve os redirecionamentos 303 e a negociação de conteúdo. Uma interface *Linked Data* para solucionar essas questões pode ser criada ou mesmo provida pelo serviço Pubby[12] através do uso da opção de configuração *loadRDF* que possibilita o carregamento de um arquivo RDF estático para a memória e o fornecimento de uma interface *Linked Data* para ele.

[10]http://simile.mit.edu/babel

[11]http://www.mindswap.org/2002/rdfconvert/

[12]http://www4.wiwiss.fu-berlin.de/pubby/

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns="http://lia.ufc.br/~regispires/researchers.rdf#">

<rdf:Description rdf:about="">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
    <dc:title>Researchers file</dc:title>
</rdf:Description>

<rdf:Description rdf:about="http://lia.ufc.br/~regispires/researchers.rdf#ufc">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
    <foaf:name>Universidade Federal do Ceará</foaf:name>
</rdf:Description>

<rdf:Description rdf:about="http://lia.ufc.br/~regispires/researchers.rdf#vania">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:name>Vânia Maria Ponte Vidal</foaf:name>
    <foaf:Organization rdf:resource="http://lia.ufc.br/~regispires/researchers.rdf#ufc"/>
    <rdfs:sameAs
rdf:resource="http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal"/>
</rdf:Description>
</rdf:RDF>
```

Figura 3.4. Arquivo RDF serializado no formato RDF/XML

Dados RDF também poder vir embutidos em documentos Web. RDFa [Adida and Birbeck 2008] permite a inclusão de dados RDF em um documento XHTML, aproximando a Web de Documentos da Web de Dados (Figura 3.5 apresenta o conteudo de um documento XHTML usando RDFa). Desse modo, textos e *links* legíveis ao ser humano também podem coexistir com dados relacionados que poderão ser facilmente processados pelas máquinas. As alterações ficam centralizadas em um único documento. Atualmente algumas ferramentas de publicação de conteúdo estão adicionando RDFa a seus documentos XHTML. Um exemplo disso é o Sistema de Gerenciamento de Conteúdo Drupal[13] a partir de sua versão 7.

Até aqui falamos de arquivos RDF com conteúdo estático. No entanto, os arquivos RDF ou RDFa podem ter seus conteúdos gerados e preenchidos dinamicamente através de uma aplicação Web.

Publicação de dados de *RDF Store* como *Linked Data*

A publicação de dados de uma *RDF Store* como *Linked Data* tipicamente envolve a disponibilização de uma interface *Linked Data* e de um *SPARQL endpoint* para acesso aos dados. O ideal seria que toda *RDF Store* já tivesse esses recursos integrados para que somente fosse necessário especificar que dados seriam publicados como *Linked Data*. Entretanto em muitos casos há necessidade de se instalar serviços adicionais para prover tais recursos.

Se a *RDF Store* não fornecer um *SPARQL endpoint* embutido, este serviço pode ser instalado para permitir a execução de consultas SPARQL sobre a *RDF Store*. Um servidor *Joseki* ou *Fuseki* pode ser usado como *SPARQL endpoint* de uma *RDF Store*.

[13] http://drupal.org/

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
 "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html
   xml:lang="en"
   version="XHTML+RDFa 1.0"
   xmlns="http://www.w3.org/1999/xhtml"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:rs="http://lia.ufc.br/~regispires/researchers.rdf#"
>
   <head>
     <title>Researchers</title>
     <meta http-equiv="content-type" content="text/html;charset=UTF-8" />
     <meta property="dc:title" content="Researchers file" />
     <link rel="rdf:type" href="http://xmlns.com/foaf/0.1/Document" />
   </head>
   <body>
     <h1>Organizations</h1>
     <ul>
       <li>
         <div about="http://lia.ufc.br/~regispires/researchers.rdf#ufc" typeof="foaf:Organization">
           <span property="foaf:name">Universidade Federal do Ceará</span>
         </div>
       </li>
     </ul>
     <h1>Researchers</h1>
     <ul>
       <li>
         <div about="http://lia.ufc.br/~regispires/researchers.rdf#vania" typeof="foaf:Person">
           <span property="foaf:name">Vânia Maria Ponte Vidal</span>
           <span rel="foaf:Organization" resource="http://lia.ufc.br/~regispires/researchers.rdf#ufc"></span>
           <span rel="rdfs:sameAs" resource="http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal"></span>
         </div>
       </li>
     </ul>
   </body>
</html>
```

Figura 3.5. Exemplo de RDFa

O *Joseki* provê suporte nativo para acesso às *RDF Stores Jena TDB*[14] e *Jena SDB*[15]. O *Fuseki* já inclui a *RDF Store Jena TDB*.

Caso a *RDF Store* não possua uma interface *Linked Data*, a aplicação Pubby pode ser usada para prover essa funcionalidade a partir de consultas realizadas sobre o *SPARQL endpoint*. Assim, ao receber uma requisição contendo uma URI a ser dereferenciada, o Pubby realiza uma consulta SPARQL usando o comando DESCRIBE para obtenção do resultado em formato RDF. Dependendo na negociação de conteúdo, esse resultado pode ser convertido para algum formato de serialização de RDF ou mesmo para a exibição de uma página HTML a ser exibida no navegador do usuário.

3.3.2. Publicação de dados de fontes não RDF como *Linked Data*

Como RDF é o modelo de dados comum usado em *Linked Data*, os dados publicados devem estar nesse modelo ou serem convertidos para ele. Caso os dados não adotem o modelo RDF, há duas abordagens possíveis para tratar essa heterogeneidade:

(i) Usar um processo de **conversão**, onde os dados não RDF são usados para gerar um arquivo RDF através de uma ferramenta específica[16]. Desse modo, através de conversores específicos é possível converter planilhas, arquivos CSV, arquivos XML e outros docu-

[14]http://openjena.org/TDB/

[15]http://openjena.org/SDB/

[16]http://www.w3.org/wiki/ConverterToRdf

mentos para o formato RDF. O projeto *RDFizer* [17] contém informações de ferramentas para conversão de vários formatos de dados para RDF, além de hospedar algumas dessas ferramentas. Após geração do arquivo em formato RDF, seus dados podem ser carregados em uma *RDF Store* e publicados conforme descrito na subseção 3.3.1. Uma vantagem dessa abordagem é a melhoria de desempenho que pode ser obtida ao usar formas de armazenamento especificamente otimizadas para realizar a persistência de triplas RDF. No entanto, o armazenamento das triplas requer espaço extra em relação aos dados originais. Além disso, a conversão demanda um certo tempo para ser realizada e os dados em RDF podem ficar desatualizados em relação aos dados originais.

(ii) Fornecer uma **visão RDF** para acesso a dados que não estão no modelo RDF através de um *RDF Wrapper*. A conversão dinâmica realizada pelo *RDF Wrapper* baseia-se em mapeamentos estabelecidos entre o modelo nativo e o modelo RDF, devendo haver um *Wrapper* específico para cada tipo de modelo. Um RDF Wrapper também pode prover uma visão RDF a dados que precisam ser acessados através de uma Web API. *RDF Book Mashup* [Bizer et al. 2007a] é uma aplicação *mashup* escrita em PHP que funciona como um *RDF Wrapper* usado para combinar dados obtidos a partir das APIs proprietárias *Amazon Web Service* e *Google Base*. Desse modo, informações sobre livros, autores, revisões e comparações de ofertas entre diferentes livrarias podem ser usados por clientes genéricos, incluindo navegadores e mecanismos de busca RDF. Essa abordagem tende a ter um desempenho inferior à abordagem anterior (i) devido às traduções dinâmicas entre os modelos que deve ser realizada a cada uso da visão RDF e também devido ao modelo original não estar otimizado especificamente para uso de triplas. No entanto, o uso de *RDF Wrappers* traz grandes vantagens, pois como o acesso ocorre sobre os dados originais, a visão RDF não requer espaço de armazenamento extra e não corre o risco de apresentar dados desatualizados.

A ampla difusão dos Bancos de Dados Relacionais motiva a necessidade de publicação dos dados no modelo relacional como *Linked Data*. Assim, seguindo a abordagem (ii), há *RDB-to-RDF Wrappers* que criam visões RDF a partir de mapeamentos entre as estruturas relacionais e os grafos RDF. A plataforma D2RQ[18] [Bizer and Seaborne 2004] fornece a infra-estrutura necessária para acessar bancos de dados relacionais como grafos RDF virtuais. Ela possui os seguintes componentes:

- **Linguagem de mapeamento D2RQ** é uma linguagem declarativa para descrever as correspondências entre o modelo relacional e o modelo RDF. Os mapeamentos escritos em D2RQ são documentos RDF.
- **Mecanismo D2RQ** é um plug-in para os *frameworks Jena* e *Sesame* que usa os mapeamentos escritos na linguagem D2RQ para converter chamadas às APIs desses *frameworks* em consultas SQL ao banco de dados para obtenção dos resultados.
- **Servidor D2R**[19] [Bizer and Cyganiak 2006] é um servidor HTTP que usa o mecanismo D2RQ para prover uma interface *Linked Data* e um *SPARQL endpoint* sobre o banco de dados relacional.

[17] http://simile.mit.edu/RDFizers/
[18] http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/
[19] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

Os componentes anteriormente descritos funcionam de forma integrada como pode ser observado na figura 3.6 que apresenta a arquitetura da plataforma D2RQ.

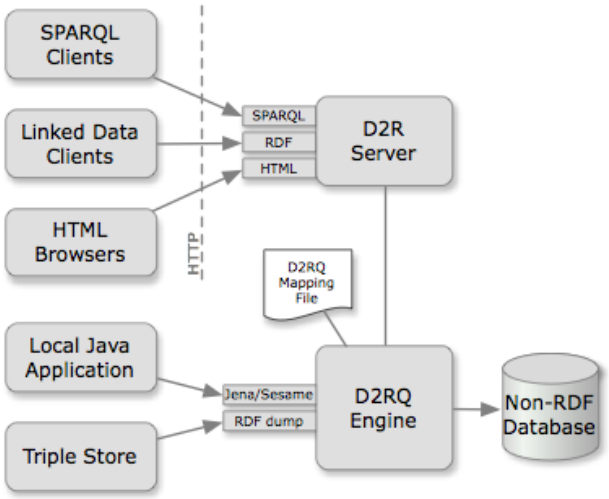

Figura 3.6. Arquitetura da plataforma D2RQ extraída de http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/

Além do D2R, duas outras ferramentas se destacam como *RDB-to-RDF Wrappers*: o Virtuoso RDF Views [Erling and Mikhailov 2006] e o Triplify[20] [Auer et al. 2009]. Este último é um pequeno *plugin* para aplicações Web que permite mapear os resultados de consultas SQL em RDF, JSON e *Linked Data*. Depois disso, os dados podem ser compartilhados e acessados na web de dados. *Triplify* consiste de poucos arquivos, totalizando menos de 500 linhas de código.

Um importante passo para a padronização desse tipo de solução para lidar com o modelo relacional foi a criação em 2009 do grupo de trabalho RDB2RDF[21] do W3C. Desde então, o grupo tem definido a linguagem padrão R2RML [Das et al. 2011] visando o mapeamento de dados e esquemas relacionais para RDF, que tende a substituir as soluções de mapeamento já existentes.

Processo de Triplificação de Modelo Relacional para RDF

O processo *StdTrip* [Salas et al. 2010] guia usuários durante os estágios de modelagem conceitual do processo RDB para RDF, que pode ser definido como uma tradução do modelo relacional para o modelo RDF. A maioria das ferramentas RDB para RDF atuais realizam essa tarefa através do mapeamento de tabelas para classes RDF e atributos para propriedades RDF, sem a preocupação de identificar possíveis correspondências com vocabulários padrões existentes. Em vez disso, essas ferramentas criam novos vocabulários. O processo *StdTrip* baseia-se nesse princípio e busca promover o reuso de padrões pela implementação de um processo guiado composto de seis passos. Os passos 1 a 6 são denominados respectivamente Conversão, Alinhamento, Seleção, Inclusão, Complementação e Saída, de acordo com as principais operações realizadas em cada um. Enquanto os passos 1, 2, 3 e 6 são obrigatórios, os passos 4 e 5 são opcionais. A seguir, uma descrição sucinta desses passos:

[20] http://triplify.org/

[21] http://www.w3.org/2001/sw/rdb2rdf/

1. **Conversão**. Este passo consiste em transformar a estrutura do banco de dados relacional em uma ontologia RDF. O projetista pode usar abordagens como *W-Ray* [Piccinini et al. 2010], onde ele manualmente define um conjunto de visões do banco de dados que capturam os dados que devem ser publicados e, então, especifica modelos *(templates)* que indicam como as triplas RDF devem ser geradas.

2. **Alinhamento**. Este passo usa a ferramenta de alinhamento de ontologias *K-match*[22] para obter correspondências entre a ontologia obtida no passo 1 e um conjunto de vocabulários padrões. O processo de alinhamento considera o esquema da ontologia previamente obtido como o esquema fonte que será recursivamente alinhado com cada ontologia representando vocabulários padrões (ontologias destino).

3. **Seleção**. Este passo apresenta ao usuário as listas de possibilidades a fim de que ele selecione os elementos dos vocabulários que melhor representam cada conceito no banco de dados. Desse modo, listas de possíveis correspondências são apresentadas para cada elemento do esquema (tabela ou atributo).

4. **Inclusão**. Se, para um dado elemento, o processo não dá um resultado (não há elemento nos vocabulários conhecidos que corresponda ao conceito no banco de dados) ou nenhuma das sugestões da lista é considerada adequada pelo usuário, *StdTrip* provê uma lista de possíveis correspondências de outros vocabulários. A escolha desses vocabulários é dependente do domínio e a busca baseada em palavras-chave é realizada através do mecanismo de busca *Watson* (ver seção 3.4).

5. **Complementação**. Se nada funciona, usuários são direcionados às melhores práticas de publicação de vocabulários RDF [Berrueta and Phipps 2008].

6. **Saída**. O processo resulta em dois artefatos: (1) um arquivo de configuração que serve como a parametrização de uma ferramenta RDB para RDF. (2) uma ontologia que contém os mapeamentos do esquema do banco de dados original para vocabulários RDF padrões.

3.3.3. Melhores práticas para publicação de *Linked Data*

A adoção das melhores práticas de publicação de *Linked Data* facilita a descoberta de informações relevantes para a integração de dados entre diferentes fontes. A seguir serão descritas algumas dessas práticas.

- **Selecionar URIs adequadas.** Deve-se evitar URIs contendo algum detalhe de implementação ou do ambiente em que estão publicadas. Como exemplo a evitar, consideremos o URI *http://lia.ufc.br:8080/ regispires/cgi-bin/resource.php?id=ufc* que possui detalhes da porta 8080 usada em seu ambiente de publicação e do script implementado em PHP necessário à sua execução.

  É frequente o uso de três URIs relacionadas a cada recurso: (i) um identificador para o recurso; (ii) um identificador para informações sobre o recurso para visualização através de navegadores HTML; (iii) um identificador para informações sobre

[22] K-match combina os resultados de algumas ferramentas de correspondência para a obtenção de um resultado mais preciso.

o recurso em formato RDF/XML. A figura 3.7 representa um exemplo de três URIs relacionadas à pesquisadora Vânia Vidal na fonte DBLP. A representação de um recurso através diferentes URIs permite que a interface Linked Data realize o dereferenciamento da URI de acordo com o tipo de conteudo requisitado no cabeçalho HTTP (i.e. Text/HTML, application/rdf+xml, etc).

```
http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal
http://dblp.l3s.de/d2r/page/V%C3%A2nia_Maria_Ponte_Vidal
http://dblp.l3s.de/d2r/data/V%C3%A2nia_Maria_Ponte_Vidal
```

Figura 3.7. Exemplos de URIs relacionadas a um mesmo recurso

A figura 3.8 apresenta dois exemplos de requisições HTTP referente à URI da pesquisadora Vânia Vidal na fonte DBLP. No exemplo referente ao item (a) a requisição define como tipo MIME dados no modelo RDF e recebe como resposta (através do redirecionamento 303 a URI referente aos dados da pesquisadora), no exemplo referente ao item (b) a requisição solicita os dados no formato HTML e recebe como resposta o redirecionamento para a URI referente à pagina HTML da pesquisadora.

```
(a)
$ curl -H "Accept: application/rdf+xml"
   http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal

303 See Other: For a description of this item,
see http://dblp.l3s.de/d2r/data/V%C3%A2nia_Maria_Ponte_Vidal

(b)
$ curl -H "Accept: text/html"
   http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal

303 See Other: For a description of this item,
see http://dblp.l3s.de/d2r/page/V%C3%A2nia_Maria_Ponte_Vidal
```

Figura 3.8. Exemplos de requisições HTTP com tipos MIME RDF e HMTL

- **Usar URIs dereferenciáveis** para que a descrição do recurso possa ser obtida da Web.
- **Utilizar URIs estáveis.** A alteração de URIs quebra links já estabelecidos, criando um problema para a localização de recursos. Para evitar esse tipo de alteração, recomenda-se um planejamento meticuloso das URIs que serão usadas e também que o responsável pela publicação detenha a propriedade do espaço de nomes.
- **Criar links para outras fontes de dados** de modo a permitir a navegação entre as fontes de dados. Os *links* podem ser criados de forma manual ou automatizada.
- **Publicação de Metadados.** Análise dos metadados facilita a seleção dos dados relevantes. Devem ser fornecidos metadados sobre proveniência e licenciamento dos dados. Também é recomendável a disponibilização de metadados sobre a fonte de dados.

- **Usar termos de vocabulários amplamente usados**. Embora não haja restrições para seleção de vocabulários, é considerada uma boa prática o reuso de termos de vocabulários RDF amplamente usados para facilitar o processamento de *Linked Data* pelas aplicações clientes [Bizer et al. 2007b]. Novos termos só devem ser definidos se não forem encontrados em vocabulários já existentes. A seguir apresentamos alguns vocabulários bastante difundidos: *Friend-of-a-Friend* (FOAF), *Semantically-Interlinked Online Communities* (SIOC), *Simple Knowledge Organization System* (SKOS), *Description of a Project* (DOAP), *Creative Commons* (CC) e *Dublin Core* (DC). Uma relação mais extensa desses vocabulários é mantida pelo projeto *Linking Open Data* no *ESW Wiki*[23].

- **Estabelecer relações entre os termos de vocabulários proprietários para termos de outros vocabulários.** Isso pode ser feito através do uso das propriedades *owl:equivalentClass*, *owl:equivalentProperty*, *rdfs:subClassOf*, *rdfs:subPropertyOf*. A figura 3.9 mostra que a classe Pessoa de um vocabulário local é equivalente à definição da classe *Person* no vocabulário da *DBpedia*.

```
<http://lia.ufc.br/Pessoa> owl:equivalentClass <http://dbpedia.org/ontology/Person> .
```

Figura 3.9. Relação de equivalência entre termo proprietário e termo da DBpedia

- **Explicitar formas de acesso adicional aos dados** como *SPARQL endpoints* e *RDF dumps*.

3.3.4. Validação dos dados publicados como *Linked Data*

Não basta simplesmente publicar dados RDF para que eles sejam considerados *Linked Data*. É necessário garantir que eles realmente estão de acordo com essas melhores práticas. Isso pode ser realizado através das ferramentas de validação que serão apresentadas a seguir.

O *W3C Validation Service*[24] permite a validação de um arquivo RDF antes de sua publicação. Documentos RDFa podem ser validados através do *W3C RDFa Distiller and Parser*[25].

Eyeball[26] é uma ferramenta criada pelo projeto Jena para a checagem de problemas comuns em modelos RDF. Ele fornece uma série de mensagens auto-explicativas sobre problemas existentes no modelo RDF analisado.

cURL é uma ferramenta de linha de comando bastante útil para checagem de tipos de conteúdos usados, redirecionamentos 303 e negociação de conteúdo decorrentes de acessos à URIs [Cyganiak 2007]. Algumas extensões de navegadores Web para modificação de cabeçalhos HTTP também podem ser usadas para realizar esses mesmos tipos de validações providas pelo *cURL*.

[23] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies
[24] http://www.w3.org/RDF/Validator/
[25] http://www.w3.org/2007/08/pyRdfa/
[26] http://jena.sourceforge.net/Eyeball/

Vapour [Berrueta et al. 2008] é um serviço online[27] de validação que a partir de uma URI, checa se os dados estão publicados de acordo com as melhores práticas definidas pelos documentos: Princípios de *Linked Data* [Berners-Lee 2006], *Best Practice Recipes* [Berrueta and Phipps 2008] e *Cool URIs* [Sauermann and Cyganiak 2008]. Ele fornece um relatório detalhado sobre os dereferenciamentos de URIs realizados por ele. Além de serviço online, Vapour[28] também é disponibilizado como software livre escrito em Python sob licença W3C.

Sindice Web Data Inspector[29] é uma ferramenta online para visualizar, inspecionar e validar o conteúdo de dados estruturados disponível em uma determinada URL ou através do preenchimento de uma caixa de texto. Esse conteúdo pode ser de arquivos RDF, páginas HTML com microformatos embutidos ou páginas XHTML contendo RDFa. No final do processo é apresentado um relatório completo com os resultados obtidos.

Os navegadores RDF abordados na seção 3.4 também são úteis para a detecção de problemas a partir da análise das visualizações de URIs dereferenciadas por eles.

3.4. Consumo de *Linked Data*

URIs, palavras-chave e consultas SPARQL são usados como ponto de partida para o consumo de *Linked Data*. Assim, todas as aplicações que consomem a Web de dados usam direta ou indiretamente pelo menos um desses itens.

Segundo [Heath and Bizer 2011] o consumo de *Linked Data* é realizado basicamente através de dois tipos de aplicações: **aplicações genéricas** que fazem uso de *Linked Data* de qualquer domínio e **aplicações de domínio específico** que são especificamente desenvolvidas para lidar com *Linked Data* relacionado a um determinado domínio.

3.4.1. Aplicações genéricas para consumo de *Linked Data*

Aplicações genéricas para consumo de *Linked Data* permitem o consumo de dados relacionados a múltiplos domínios distribuídos pelo amplo espaço de dados global. Ao percorrer os *RDF links* é possível explorar e descobrir novas informações na web de dados. A seguir serão abordados alguns tipos de aplicações genéricas normalmente usadas para acessar *Linked Data*.

Navegadores RDF
Navegadores RDF são aplicações executadas a partir dos navegadores Web convencionais que dereferenciam URIs e exibem uma visualização desse resultado, possibilitando a navegação aos dados de fontes relacionadas a partir dos *RDF links*.

LOD Browser Switch[30] é uma aplicação web que obtém detalhes a respeito de uma URI especificada pelo usuário a partir da seleção de um dos vários navegadores *Linked Data* disponibilizados pela aplicação. Assim é possível comparar a visualização de uma URI através de vários navegadores *Linked Data*.

[27] http://validator.linkeddata.org/vapour
[28] http://vapour.sourceforge.net/
[29] http://inspector.sindice.com/
[30] http://browse.semanticweb.org/

Explorator[31] [Araújo and Schwabe 2009, Araújo et al. 2009] é uma ferramenta desenvolvida pelo grupo TecWeb da PUC-Rio para exploração de dados RDF através de manipulação direta. Sua interface web possibilita a obtenção de conhecimentos e respostas a questões específicas sobre um domínio através de mecanismos de visualização, busca e exploração. A ferramenta é recomendada para a formulação de consultas complexas sobre um domínio desconhecido mesmo por usuários com pouco conhecimento sobre o modelo RDF.

Disco Hiperdata Browser[32] é uma aplicação Web usada como navegador simples para visualizar informações sobre um recurso em página HTML. Para iniciar a navegação, o usuário digita a URI do recurso em uma caixa de texto e pressiona o botão "Go". A partir daí, *Disco* recupera as informações sobre o recurso e as exibe em uma tabela contendo propriedades, valores e fontes. As abreviações G1, G2, ..., Gn referem-se a uma lista de todas as fontes que são mostradas em outra tabela. Os *links* exibidos permitem a navegação entre os recursos, de modo que ao selecionar um novo recurso, o navegador dinamicamente recupera informações sobre ele através do dereferenciamento de sua URI. À medida que a navegação é feita, *Disco* armazena os grafos RDF recuperados em um *cache* de sessão. Ao clicar sobre o link "Display all RDF graphs", uma nova janela é aberta contendo a lista dos grafos RDF recuperados e das URIs que não foram dereferenciadas com sucesso. *Disco* pode ser usado de forma online ou baixado para execução local. A figura 3.10 exibe informações obtidas a partir do dereferenciamento da URI *http://dblp.l3s.de/d2r/resource/authors/Marco_A._Casanova* pelo *Disco*.

Marbles [33] é uma aplicação web que funciona como navegador *Linked Data* para clientes XHTML. Pontos coloridos são usados para correlacionar a origem dos dados visualizados com uma lista de fontes de dados. Além de dereferenciar a URI, *Marbles* consulta os mecanismos de busca *Sindice* e *Falcons* por fontes que contenham informações sobre o recurso. Ele também obtém resultados de críticas relacionadas ao recurso a partir do serviço *Revyu*. Além disso, os predicados *owl:sameAs* e *rdfs:seeAlso* são seguidos para obtenção de informações adicionais sobre o recurso.

O navegador *Tabulator*[34] [Berners-Lee et al. 2006, Berners-Lee et al. 2007] possui os modos de exploração e análise dos dados. A utilização do modo de exploração inicia a partir da submissão de uma URI. Depois disso, *Tabulator* obtém informações sobre o recurso e as exibe como um grafo RDF em uma visão de árvore. A expansão de um nó permite a obtenção de mais informações sobre ele. Para passar ao modo de análise, o usuário pode selecionar predicados para definir um padrão e requisitar que o *Tabulator* encontre todos os exemplos daquele padrão. Para realizar a consulta, ele segue os links para encontrar o padrão no grafo RDF. Os resultados podem ser exibidos através das visões de tabela, mapa, calendário ou linha de tempo. Pode-se iniciar uma nova exploração pela seleção de um detalhe de uma das visões. Tabulator pode ser usado como um complemento do navegador Firefox ou como uma aplicação web que atualmente é compatível apenas com o Firefox. O principal objetivo do complemento do Firefox é explorar como dados vinculados poderiam ser visualizados em uma futura geração de navegadores Web.

[31] http://www.tecweb.inf.puc-rio.br/explorator
[32] http://www4.wiwiss.fu-berlin.de/rdf_browser/
[33] http://www5.wiwiss.fu-berlin.de/marbles/
[34] http://dig.csail.mit.edu/2005/ajar/ajaw/tab

Figura 3.10. Visualização de informações sobre recurso através do navegador Disco

Assim, ao abrir um documento, o usuário também pode visualizar os itens que o documento descreve. As propriedades desses itens são exibidas em uma tabela e *links* podem ser seguidos para carregar mais dados sobre outros itens.

LinkSailor[35] e *Graphite RDF Browser*[36] são navegadores simples e rápidos para obter detalhes sobre uma determinada URI após dereferenciá-la. *LinkSailor* exibe uma visualização adaptada aos tipos dos dados exibidos.

Mecanismos de Busca *Linked Data*

O acesso à Web de Dados pode ocorrer a partir de mecanismos de busca específicos capazes de realizar pesquisas que levam em consideração a semântica dos dados. Esses mecanismos de busca permitem localizar recursos de diferentes fontes normalmente através de

[35] http://linksailor.com/
[36] http://graphite.ecs.soton.ac.uk/browser/

palavras-chave. A consulta pode ser realizada pelo usuário através de uma interface web ou através de serviços web providos pelos mecanismos de busca. Mecanismos de busca *Linked Data* percorrem a Web de dados percorrendo os *links* entre as fontes de dados e fornecendo a possibilidade de consultas sobre os dados dessas fontes. Os resultados das buscas são URIs que podem ser dereferenciadas e visualizadas através dos navegadores RDF. Atualmente há vários mecanismos de busca *Linked Data*. A seguir apresentaremos alguns deles.

Sindice[37] [Oren et al. 2008] coleta dados estruturados na Web (RDF, RDFa e microformatos) e os indexa por URIs, propriedades funcionais inversas (IFPs) e palavras-chave, oferecendo uma interface Web para que os usuários possam fazer buscas a partir dos itens indexados. *Sindice* também fornece um *SPARQL endpoint* que permite a realização de consultas sobre todos os seus dados e uma API para permitir a utilização de seus serviços por desenvolvedores de aplicações.

Sig.ma[38] [Tummarello et al. 2010] busca dados estruturados a partir de uma palavra-chave e os exibe em uma única página, integrando os dados de múltiplas fontes. A visão criada pelo *Sig.ma* baseia-se em resultados fornecidos pelo *Sindice*. O usuário pode aprovar, rejeitar ou acrescentar fontes para estabelecer uma visão dos dados relevantes. Ao selecionar uma entidade da lista de resultados, uma nova visão é apresentada ao usuário. Um *link* permanente pode ser criado para futuros acessos ou compartilhamento dessa visão. As filtragens das fontes de dados realizadas pelos usuários coletivamente ajudam a classificar melhor a relevância das fontes e aperfeiçoar a qualidade dos resultados futuros. Além da interface web do usuário, *Sig.ma* ainda fornece uma API destinada aos desenvolvedores de aplicações. A figura 3.11 ilustra o resultado de uma consulta sobre a pesquisadora Vânia Vidal envolvendo dezesseis fontes, onde quatro delas foram rejeitadas.

VisiNav[39] pode ser usado para consultar e navegar na web de dados através de diferentes visões (tabela, grafo, mapa, linha de tempo) sobre os dados especificados. Os dados no VisiNav consistem de objetos que possuem atributos ou links para outros objetos. Inicialmente uma lista de objetos é obtida a partir de palavras-chave digitadas pelo usuário. Essa lista pode ser filtrada a partir do arrastar e soltar de objetos sobre as restrições já realizadas.

Watson[40] [d'Aquin et al. 2007] e *Swoogle*[41] [Ding et al. 2004] são mecanismos de busca mais voltados para a descoberta de informações sobre ontologias. Podem ser usados, por exemplo, para obter ontologias que possuem determinados conceitos e descobrir relacionamentos entre termos.

Outras aplicações genéricas

Informações adicionais sobre determinado recurso podem ser obtidas através da localização de objetos referenciados pelas propriedades `rdfs:seeAlso` e `owl:sameAs`.

[37] http://sindice.com/
[38] http://sig.ma/
[39] http://visinav.deri.org/
[40] http://watson.kmi.open.ac.uk/WatsonWUI/
[41] http://swoogle.umbc.edu/

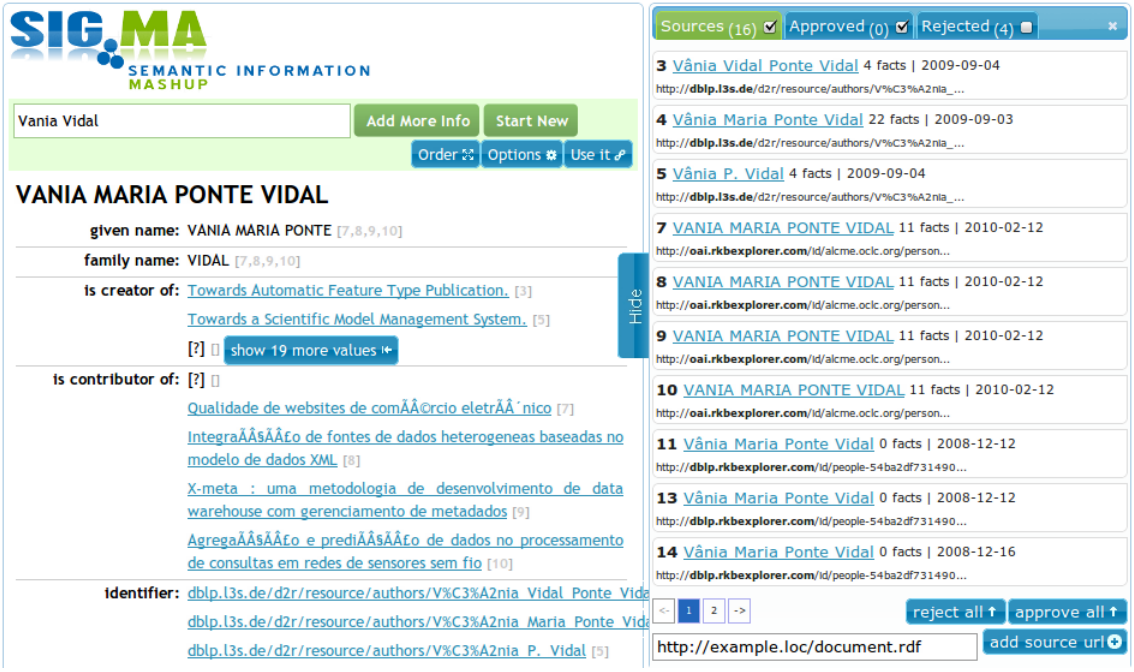

Figura 3.11. Visão criada pelo Sig.ma sobre a pesquisadora Vânia Vidal

Serviços online de coreferenciamento como o *sameAs*[42] são usados para encontrar URIs de diferentes fontes de dados que representam um mesmo conceito.

LDSpider[43] é um framework capaz de navegar pela web de dados seguindo *links* para obter dados de fontes *Linked Data* e os armazenar em uma *RDF Store* através de SPARQL Update ou como arquivo RDF.

3.4.2. Aplicações de domínio específico para consumo de *Linked Data*

Várias aplicações têm sido desenvolvidas para integrar *Linked Data* em domínios específicos. Essas aplicações são chamadas de *Linked Data Mashups*. A seguir descreveremos algumas delas.

Revyu[44] é uma aplicação web para crítica e classificação de qualquer item passível de avaliação. *Revyu* também disponibiliza uma API e um *SPARQL endpoint* para serem usados pelos desenvolvedores de aplicações.

DBpedia Mobile[45] [Becker and Bizer 2008] é uma aplicação cliente para dispositivos móveis consistindo de uma visão com um mapa e do navegador *Linked Data Marbles*. Baseado na localização geográfica de um dispositivo móvel, a aplicação exibe um mapa indicando localizações próximas a partir de dados extraídos das fontes *DBpedia*, *Revyu* e *Flickr*. O acesso ao *Flickr* é realizado através de um *Wrapper*. O usuário pode explorar informações sobre essas localizações e navegar em conjuntos de dados interligados. Também é possível a publicação de informações como *Linked Data*, de modo que

[42] http://sameas.org
[43] http://code.google.com/p/ldspider/
[44] http://revyu.com/
[45] http://beckr.org/DBpediaMobile/

possam ser usadas por outras aplicações.

Talis Aspire[46] é uma aplicação web voltada para que alunos e professores e possam encontrar os principais recursos educacionais em universidades do Reino Unido. O serviço é gratuito e provê ferramentas para criar e editar listas de leitura, além da produção e publicação de materiais educativos. Quando o usuário publica conteúdo, a aplicação cria triplas RDF em uma *RDF store*. Itens publicados são interligados de forma transparente a itens correspondentes de outras instituições.

BBC Programmes[47] e *BBC Music*[48] são projetos desenvolvidos pela *BBC Audio and Music Interactive*. A aplicação web *BBC Programmes* disponibiliza informações detalhadas sobre tipos, séries e episódios de todos os programas de TV e rádio transmitidos pela BBC. *BBC Music* fornece informações sobre artistas, vinculando-os aos programas da BBC. Assim é possível escolher um artista e obter todos os episódios de programas relacionados a ele. As aplicações mencionadas usam *Linked Data* como tecnologia de integração de dados, inclusive fazendo uso de vocabulários amplamente conhecidos como *DBpedia* e *MusicBrainz*.

Semantic Web Pipes (SWP) [Le-Phuoc et al. 2009] apresenta um estilo de arquitetura flexível para o desenvolvimento de *mashups* de dados usando as tecnologias da Web Semântica. *Pipes* são planos de consultas criados visualmente pelo desenvolvedor do *mashup* através da conexão de operações sobre os dados usando o modelo RDF. Depois de criado, um *Pipe* é salvo e disponibilizado para uso através de uma simples requisição HTTP a uma URI específica através de Serviços Web REST.

3.4.3. APIs para manipulação de *Linked Data*

A seguir descreveremos algumas APIs para manipulação de dados na web semântica que são usadas no desenvolvimento de aplicações de domínio genérico ou específico para consumo de *Linked Data*.

Sesame[49] e *Jena*[50] são *frameworks* de web semântica implementados em Java que fornecem APIs para manipulação de grafos RDF.

Sesame permite armazenamento, consulta e manipulação de dados RDF. Além disso, o *framework* é extensível e configurável em relação a formas de armazenamento (memória e *RDF store*), mecanismos de inferência, formatos de arquivo RDF e linguagens de consulta (SPARQL e SeRQL).

Jena foi desenvolvido no *HP Labs* entre 2000 e 2009. Atualmente faz parte do projeto *Apache* e suas principais características são: suporte a RDF, RDFa, RDFS, OWL e SPARQL; armazenamento de triplas RDF em memória, banco de dados relacional *(Jena SDB)* ou *RDF store (Jena TDB)*; processamento de consultas SPARQL *(Jena ARQ)*; disponibilização de *SPARQL endpoint* (*Joseki* ou *Fuseki*); disponibilização de mecanismos de inferência embutidos e interfaces para mecanismos de inferência externos.

[46] http://www.talisaspire.com/
[47] http://www.bbc.co.uk/programmes
[48] http://www.bbc.co.uk/music
[49] http://www.openrdf.org/
[50] http://incubator.apache.org/jena/

Named Graphs API for Jena (NG4J)[51] é uma extensão ao framework Jena para análise, manipulação e serialização de conjuntos de grafos nomeados representando os grafos como modelos ou grafos do Jena. NG4J permite o armazenamento de grafos em memória ou em banco de dados. Consultas SPARQL podem ser realizadas sobre os grafos nomeados.

O *Semantic Web Client Library* (SWClLib)[52] [Hartig et al. 2009] faz parte do NG4J e é capaz de representar a web de dados como um único grafo RDF. Ele recupera informações dereferenciando URIs, seguindo *links rdfs:seeAlso* e consultando o mecanismo de busca *Sindice*. O SWClLib considera todos os dados como um único conjunto global de grafos nomeados, sendo usado na implementação de vários navegadores *Linked Data*. Os grafos recuperados são mantidos em um *cache* local para melhorar o desempenho de buscas futuras.

ARQ2[53] é uma biblioteca escrita em PHP que contempla armazenamento de Triplas RDF, *SPARQL endpoint* e interface *Linked Data* em uma única ferramenta. As triplas RDF são armazenadas em um banco de dados MySQL. A infra-estrutura necessária para o funcionamento do ARQ2 é muito simples por requerer apenas um servidor Web com suporte a PHP e um banco de dados MySQL, sendo facilmente encontrada em qualquer serviço de hospedagem Web.

3.4.4. Abordagens para execução de consultas sobre múltiplas fontes de dados

Aplicações podem acessar *Linked Data* na web através de consultas a um *SPARQL endpoint* de um determinado conjunto de dados. Embora esse acesso possa prover dados valiosos para a aplicação, essa abordagem ignora o grande potencial da web de dados, pois não explora as possibilidades deste imenso espaço de dados que integra um grande número de conjuntos de dados interligados. Essas possibilidades podem ser alcançadas pela execução de consultas complexas e estruturadas sobre múltiplos conjuntos de dados. [Hartig and Langegger 2010] discutem diferentes abordagens para realizar essas consultas sobre a web de dados, classificando-as basicamente em dois tipos: tradicionais e inovadoras.

Abordagens Tradicionais
Data warehousing e federação de consultas são abordagens amplamente discutidas na literatura de banco de dados para realização de consultas sobre dados distribuídos em fontes autônomas. Consultas sobre a web de dados podem utilizar essas abordagens tradicionais que requerem o conhecimento prévio das fontes de dados relevantes e, portanto, limitam as fontes de dados que serão levadas em conta para obter as respostas de uma consulta. A seguir descreveremos a aplicação dessas abordagens sobre a web de dados.

Data warehousing usa uma base de dados centralizada que coleta e armazena os dados das fontes. No contexto de *Linked Data*, podem-se materializar dados das fontes relevantes em uma base centralizada para a execução de consultas sobre ela. Tal estratégia também pode ser usada em mecanismos de busca sobre a web de dados. Além disso, ela possui o melhor desempenho dentre as abordagens que serão aqui discutidas já que

[51] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/
[52] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/
[53] http://arc.semsol.org/

os dados podem ser acessados diretamente na base centralizada, sem a necessidade de comunicações adicionais através da rede. No entanto, em fontes de dados cujo volume de dados é muito grande, a materialização dos dados tende a requerer bastante tempo e espaço de armazenamento. Outro problema é que atualizações sobre as fontes não são imediatamente refletidas sobre o repositório central, podendo ocasionar consultas com resultados desatualizados em relação aos dados originais. Outra questão a ser considerada é que as consultas somente são realizadas sobre os dados materializados e não sobre toda a web de dados.

Federação de consultas baseia-se na distribuição do processamento de consultas para múltiplas fontes de dados autônomas. O objetivo é dar ao usuário acesso aos dados por meio de algum vocabulário padrão especificado em uma ontologia de domínio. Consultas podem ser formuladas baseadas na ontologia de domínio e um mediador transparentemente decompõe a consulta em subconsultas, direciona as subconsultas a múltiplos serviços de consulta distribuídos, e, finalmente, integra os resultados das subconsultas. Em mais detalhe, o processamento de uma consulta requer as seguintes tarefas: particionamento, adaptação, mapeamento, otimização e execução. A tarefa de adaptação consiste na modificação e extensão da consulta, por exemplo, através da inclusão de termos similares ou mais abrangentes, a partir de relacionamentos com outros vocabulários, expandindo assim o escopo do espaço de busca de forma a obter melhores resultados. A tarefa de mapeamento consiste na seleção conjuntos de *Linked Data* que têm potencial para retornar resultados para as expressões contidas na consulta. A tarefa de otimização avalia o custo de diferentes estratégias para processar a consulta, preparando um plano de execução para a consulta. Finalmente, a tarefa de execução implementa uma via de comunicação com os conjuntos de *Linked Data* e processa o plano de execução preparado pela tarefa de otimização, possivelmente adaptando-o dinamicamente. Uma vantagem da federação de consultas é que ela não requer tempo ou espaço adicional para materialização de dados. Por outro lado, a execução de consultas é mais lenta devido às transmissões de rede necessárias para realização das subconsultas sobre as fontes de dados. Além disso, as consultas não podem ser realizadas sobre toda a web de dados, mas somente sobre as fontes de dados registradas no mediador. *DARQ* [Quilitz and Leser 2008] é um mediador baseado no processador de consultas *Jena ARQ* capaz de realizar consultas distribuídas sobre a web dados. *SemWIQ* [Langegger 2010] é outro mediador que estende o *Jena ARQ* a fim de consultar a web de dados fazendo uso de estatísticas [Langegger and Woss 2009] para otimizar as consultas. [Vidal et al. 2011] apresentam um *framework* baseado em mediador de três níveis para integração de dados sobre *Linked Data*. Desafios relacionados à eficiência de consultas federadas e uma abordagem para otimização dessas consultas baseada em programação dinâmica foram tratados por [Görlitz and Staab 2011].

Abordagens Inovadoras

As abordagens inovadoras surgiram para eliminar a restrição imposta pelas abordagens tradicionais de limitarem as consultas sobre as fontes previamente conhecidas. Assim, elas permitem a descoberta das fontes durante a execução das consultas, podendo atuar sobre toda a web de dados. [Hartig and Langegger 2010] caracterizam duas abordagens inovadoras: descoberta ativa baseada em federação de consultas e consultas exploratórias (também conhecidas como *link traversal*).

Descoberta ativa baseada em federação de consultas é uma estratégia base-

ada na combinação de processamento de consultas federado com uma descoberta ativa de fontes de dados relevantes pode ser usada para possibilitar o uso de fontes de dados desconhecidas. Essa estratégia parece não ter sido implementada até o momento da publicação do presente capítulo, mas é uma estratégia que vale a pena ser objeto de investigações futuras, desde que pode combinar as vantagens da federação de consultas com a possibilidade de obter dados de fontes ainda desconhecidas pelo mediador.

Consultas exploratórias *(link traversal)*. No enfoque exploratório, proposto por [Hartig et al. 2009] dados são descobertos e recuperados em tempo de execução da consulta. Este enfoque é baseado na busca de URIs, onde uma consulta SPARQL é executada através de um processo interativo onde URIs são dereferenciadas de modo a recuperar suas descrições em RDF na Web e os resultados da consulta construídos a partir dos dados recuperados. Desse modo, consultas exploratórias seguem *RDF links* para obter mais informações sobre os dados já existentes. Através do uso de dados recuperados a partir das URIs usadas em uma consulta como ponto de partida, o processador de consultas avalia partes da consulta. Soluções intermediárias resultantes dessa avaliação parcial geralmente contêm URIs adicionais que possuem ligações para outros dados que por sua vez, podem prover novas soluções intermediárias para a consulta. Para determinar o resultado completo da consulta, o processador de consultas avalia as partes da consulta e dereferencia URIs. O conjunto de dados usado na consulta é continuamente ampliado com dados potencialmente relevantes da web, cuja descoberta é realizada a partir das URIs de soluções intermediárias que podem estar em espaços de nomes distintos. Este enfoque possui duas limitações: recupera apenas URIs e exige que a consulta seja executada a partir de uma URI somente, que faz o papel de padrão para a consulta. Finalmente, o fato do enfoque ser centralizado limita a otimização do processamento de consultas [Reddy and Kumar 2010]. SQUIN[54] [Hartig et al. 2009] é uma interface de consulta sobre *Linked Data* que implementa a abordagem de consultas exploratórias.

3.5. Limitações e Desafios

Esta seção aborda limitações das tecnologias atuais e apresenta desafios que ainda precisam ser superados para aperfeiçoar o consumo de *Linked Data*. As tecnologias atuais revelam deficiências como interfaces com o usuário ainda precárias; desempenho insatisfatório nas consultas sobre múltiplas fontes de dados; instabilidade no acesso a essas fontes; acesso a *links* quebrados e descoberta de fontes de dados relevantes. Além disso, faltam estratégias bem definidas para garantir a privacidade dos dados e tratar restrições sobre eles.

Desafios relacionados com a publicação de *Linked Data*
[Hartig and Langegger 2010] afirmam a necessidade de tornar mais transparente a integração de dados entre múltiplas fontes. Isso requer mapeamentos entre termos de diferentes vocabulários usados por fontes de dados com conteúdos similares. Além disso, pode ser necessário aplicar técnicas de fusão de dados para obter uma representação consistente de dados descritos diferentemente em fontes distintas, bem como, ajudar a resolver questões relacionadas a conflitos e qualidade dos dados. Muito ainda precisa ser feito também em relação à inferência e descoberta de conhecimento em dados provenientes de

[54]http://squin.sourceforge.net/

múltiplas fontes.

Permitir o mapeamento dos diversos vocabulários existentes, para que seja possível identificar e escolher dados de fontes diferentes sobre uma mesma entidade também é uma questão que requer maior aprofundamento.

Permitir a criação, edição e manutenção de *Linked Data* por vários usuários é um desafio. Outro desafio está relacionado à manutenção desses dados para evitar problemas de acesso a informações que não estejam mais disponíveis. A Web de Dados é dinâmica e deve permitir que aplicações possam fazer atualizações e utilizar técnicas avançadas para a detecção de inconsistências. A web de dados é alimentada com dados provenientes dos mais diversos domínios, causando problemas quanto à confiabilidade e qualidade daquilo que é disponibilizado.

As possibilidades criadas por esses dados integrados podem infringir os direitos de privacidade dos usuários. Proteger os direitos dos indivíduos se torna difícil, pois os dados estão em fontes descentralizadas e sob diversas jurisdições legais. Prover ferramentas para explicitar os direitos de cópia e reprodução sobre os dados é uma das lacunas no contexto de *Linked Data*.

Desafios relacionados com o consumo de *Linked Data*
Já existem várias aplicações funcionais e em desenvolvimento que permitem consultas complexas na Web de Dados, porém, ainda existem muitas oportunidades de pesquisa relacionadas à forma que os usuários poderão navegar por esses dados para tornar essa interação mais intuitiva, simples e objetiva.

Há algumas formas de consulta sobre múltiplas fontes *Linked Data*. Pode-se usar materialização dos dados em uma base centralizada, consultas federadas ou consulta exploratória *(link traversal)*. No entanto, ainda é necessário aperfeiçoar ou mesmo integrar esses tipos de acessos para tirar proveito das vantagens de cada um.

Determinar as informações mais relevantes, assim como detectar sua validade para melhorar a qualidade da informação, também são desafios que precisam ser superados através de algum *feedback* do usuário ou mesmo de forma automatizada.

Encontrar *SPARQL endpoints* relevantes normalmente é uma tarefa complexa. Para simplificá-la é possível obter a listagem de vários *endpoints* a partir do endereço *http://esw.w3.org/topic/SparqlEndpoints*. Além disso, a versão 1.1 do protocolo SPARQL prevê a existência do mecanismo **Descrição de Serviço** [Williams 2006] para a descoberta de informações sobre o *SPARQL endpoint*. ***VoiD*** *(Vocabulary of Interlinked Datasets)* [Alexander et al. 2011] é um vocabulário usado para definição de metadados sobre fontes de dados RDF. A partir dele fica muito mais fácil identificar fontes de dados relevantes. No entanto, muito ainda precisa ser realizado para reduzir ainda mais a complexidade da descoberta dessas fontes.

3.6. Conclusão

Utilizando os mecanismos de acesso padronizados disponibilizados como *Linked Data*, é possível ter acesso a fontes de dados ilimitadas em busca de um melhor aproveitamento do potencial da web e revolucionando a forma como os dados são publicados e consu-

midos. O volume de dados disponibilizados seguindo os princípios de *Linked Data* é enorme, cresce muito rapidamente e cobre os mais variados domínios. Várias aplicações apresentadas ao longo deste capítulo utilizam estes dados para publicação e consumo de *Linked Data*. No entanto, vários desafios ainda precisam ser superados para que seja possível aproveitar todo o potencial da Web de Dados, restando ainda muito mais a ser feito. Assim, interfaces mais interativas e de fácil uso para que o usuário possa consultar e navegar pela web de dados faz falta, assim como outras questões que precisam ser mais bem desenvolvidas, como desempenho de consultas e qualidade dos dados retornados.

Referências

[Adida and Birbeck 2008] Adida, B. and Birbeck, M. (2008). RDFa Primer — Bridging the Human and Data Webs. `http://www.w3.org/TR/xhtml-rdfa-primer/`.

[Alexander et al. 2011] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. (2011). Describing Linked Datasets with the VoID Vocabulary. `http://www.w3.org/2001/sw/interest/void/`.

[Araújo and Schwabe 2009] Araújo, S. F. C. and Schwabe, D. (2009). Explorator: a Tool for Exploring RDF Data Through Direct Manipulation. In *LDOW 2009: Linked Data on the Web*.

[Araújo et al. 2009] Araújo, S. F. C., Schwabe, D., and Barbosa, S. D. J. (2009). Experimenting with Explorator: a Direct Manipulation Generic RDF Browser and Querying Tool. In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009)*.

[Auer et al. 2009] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D. (2009). Triplify: Light-weight linked data publication from relational databases. In Quemada, J., León, G., Maarek, Y. S., and Nejdl, W., editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 621–630. ACM.

[Becker and Bizer 2008] Becker, C. and Bizer, C. (2008). DBpedia Mobile: A Location-Enabled Linked Data Browser. In *Linked Data on the Web (LDOW2008)*.

[Berners-Lee 2006] Berners-Lee, T. (2006). Linked Data - Design Issues. `http://www.w3.org/DesignIssues/LinkedData.html`.

[Berners-Lee et al. 2006] Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., and Sheets, D. (2006). Tabulator: Exploring and Analyzing Linked Data on the Semantic Web. In *In Procedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06*, page 06.

[Berners-Lee et al. 2005] Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform resource identifier (URI): Generic syntax. Internet Engineering Task Force RFC 3986, Internet Society (ISOC). Published online in January 2005 at `http://tools.ietf.org/html/rfc3986`.

[Berners-Lee et al. 2007] Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., D'ommeaux, P. E., and Schraefel, M. (2007). Tabulator redux: Writing into the semantic web. Technical report, School of Electronics and Computer Science, University of Southampton, Southampton, UK.

[Berrueta et al. 2008] Berrueta, D., Fernández, S., and Frade, I. (2008). Cooking HTTP content negotiation with Vapour. In *Proceedings of the 4th Worshop on Scripting for the Semantic Web 2008 (SFSW2008)*.

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

[Bizer and Cyganiak 2006] Bizer, C. and Cyganiak, R. (2006). D2R Server – Publishing Relational Databases on the Semantic Web. In *5th International Semantic Web Conference*.

[Bizer et al. 2007a] Bizer, C., Cyganiak, R., and Gaus, T. (2007a). The RDF Book Mashup: from Web APIs to a web of data. In *The 3rd Workshop on Scripting for the Semantic Web (SFSW 2007), Innsbruck, Austria*.

[Bizer et al. 2007b] Bizer, C., Cyganiak, R., and Heath, T. (2007b). How to Publish Linked Data on the Web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

[Bizer et al. 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.

[Bizer et al. 2011] Bizer, C., Jentzsch, A., and Cyganiak, R. (2011). State of the LOD Cloud. http://www4.wiwiss.fu-berlin.de/lodcloud/state/.

[Bizer and Seaborne 2004] Bizer, C. and Seaborne, A. (2004). D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. In *ISWC2004 (posters)*.

[Clark et al. 2008] Clark, K. G., Feigenbaum, L., and Torres, E. (2008). SPARQL Protocol for RDF. http://www.w3.org/TR/rdf-sparql-protocol/.

[Cyganiak 2007] Cyganiak, R. (2007). Debugging Semantic Web sites with cURL. http://richard.cyganiak.de/blog/2007/02/debugging-semantic-web-sites-with-curl/.

[d'Aquin et al. 2007] d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., and Motta, E. (2007). Characterizing knowledge on the semantic web with watson. In *Evaluation of Ontologies and Ontology-Based Tools: 5th International EON Workshop*.

[Das et al. 2011] Das, S., Sundara, S., and Cyganiak, R. (2011). R2RML: RDB to RDF Mapping Language. http://www.w3.org/TR/2011/WD-r2rml-20110324/.

[Ding et al. 2004] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 652–659, New York, NY, USA. ACM.

[Erling and Mikhailov 2006] Erling, O. and Mikhailov, I. (2006). Mapping Relational Data to RDF in Virtuoso. http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF.

[Gearon et al. 2011] Gearon, P., Passant, A., and Polleres, A. (2011). SPARQL 1.1 Update. http://www.w3.org/TR/sparql11-update/.

[Görlitz and Staab 2011] Görlitz, O. and Staab, S. (2011). Federated Data Management and Query Optimization for Linked Open Data. In Vakali, A. and Jain, L., editors, *New Directions in Web Data Management 1*, volume 331 of *Studies in Computational Intelligence*, pages 109–137. Springer Berlin / Heidelberg.

[Hartig et al. 2009] Hartig, O., Bizer, C., and Freytag, J.-C. (2009). Executing SPARQL Queries over the Web of Linked Data. In Bernstein, A., Karger, D., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., and Thirunarayan, K., editors, *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 293–309. Springer Berlin / Heidelberg.

[Hartig and Langegger 2010] Hartig, O. and Langegger, A. (2010). A Database Perspective on Consuming Linked Data on the Web. *Datenbank-Spektrum*, 14(2):1–10.

[Heath and Bizer 2011] Heath, T. and Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition.

[Langegger 2010] Langegger, A. (2010). *A Flexible Architecture for Virtual Information Integration based on Semantic Web Concepts*. PhD thesis, J. Kepler University Linz.

[Langegger and Woss 2009] Langegger, A. and Woss, W. (2009). Rdfstats - an extensible rdf statistics generator and library. In *Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application*, DEXA '09, pages 79–83, Washington, DC, USA. IEEE Computer Society.

[Le-Phuoc et al. 2009] Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., and Morbidoni, C. (2009). Rapid Prototyping of Semantic Mash-ups through Semantic Web Pipes. In *Proceedings of the 18th international conference on World wide web - WWW '09*, pages 581–590, New York, New York, USA. ACM Press.

[Manola and Miller 2004] Manola, F. and Miller, E. (2004). RDF Primer. `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/`.

[Oren et al. 2008] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., and Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. *Int. J. Metadata Semant. Ontologies*, 3:37–52.

[Piccinini et al. 2010] Piccinini, H., Lemos, M., Casanova, M. A., and Furtado, A. L. (2010). W-ray: a strategy to publish deep web geographic data. In *Proceedings of the 2010 international conference on Advances in conceptual modeling: applications and challenges*, ER'10, pages 2–11, Berlin, Heidelberg. Springer-Verlag.

[Prud'hommeaux and Seaborne 2008] Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. `http://www.w3.org/TR/rdf-sparql-query/`.

[Quilitz and Leser 2008] Quilitz, B. and Leser, U. (2008). Querying Distributed RDF Data Sources with SPARQL. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pages 524–538, Berlin, Heidelberg. Springer-Verlag.

[Reddy and Kumar 2010] Reddy, K. B. R. and Kumar, P. S. (2010). Optimizing SPARQL queries over the Web of Linked Data. In *Proceedings of the International Workshop on Semantic Data Management (SemData 2010), Singapore*.

[Salas et al. 2010] Salas, P. E., Breitman, K. K., Viterbo F., J., and Casanova, M. A. (2010). Interoperability by design using the stdtrip tool: an a priori approach. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, pages 43:1–43:3, New York, NY, USA. ACM.

[Sauermann and Cyganiak 2008] Sauermann, L. and Cyganiak, R. (2008). Cool URIs for the Semantic Web. http://www.w3.org/TR/cooluris/.

[Tummarello et al. 2010] Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., and Decker, S. (2010). Sig.ma: Live views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355 – 364. Semantic Web Challenge 2009; User Interaction in Semantic Web research.

[Vidal et al. 2011] Vidal, V. M. P., de Macêdo, J. A. F., Pinheiro, J. C., Casanova, M. A., and Porto, F. (2011). Query Processing in a Mediator Based Framework for Linked Data Integration. *IJBDCN*, 7(2):29–47.

[Williams 2006] Williams, G. T. (2006). SPARQL 1.1 Service Description. http://www.w3.org/TR/sparql11-service-description/.