

Capítulo

4

Linked Data: da Web de Documentos para a Web de Dados

Danusa R. B. Cunha, Bernadette F. Lóscio, Damires Souza

Abstract

The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web with the purpose of creating a Web of Data. Recently, a significant number of datasets have been published adhering to the Linked Data principles and numerous efforts are underway to build applications for exploit this Web of Data. In this context, during this course, we present the fundamental concepts of Linked Data, as well as the theoretical basis of how to publish and to consume data available on the Web of Data. We also present some applications for visualization and consume of Linked Data and we discuss the main difficulties and challenges in this research area.

Resumo

O termo Linked Data refere-se a um conjunto de práticas para publicar e conectar dados estruturados na Web, com o intuito de criar uma “Web de Dados”. Recentemente, um grande volume de dados definidos de acordo com o padrão Linked Data tem sido publicado, com isso, muitos esforços estão sendo feitos para construir aplicações com o objetivo de facilitar a exploração de tais dados. Neste cenário, este minicurso apresenta os conceitos relacionados ao termo Linked Data, bem como provê aos participantes um embasamento prático de como publicar e consumir dados na Web de Dados. Além disso, são apresentadas as aplicações usadas para visualização e consumo dos dados Linked Data e, por fim, discute as principais dificuldades e desafios nessa área de pesquisa.

4.1 Introdução

A Internet contemporânea, nos moldes da *World Wide Web*, vive um constante processo de evolução e tem revolucionado a forma como criamos conteúdo e trocamos informações. A Web organiza as informações disponíveis na Internet por meio de hipertexto e torna a interação do usuário com a rede mundial mais amigável. Com isso, possibilita um ambiente de compartilhamento de documentos oriundos de diversas áreas do conhecimento. Entretanto, tais conteúdos geralmente seguem regras apenas sintáticas, com objetivos de apresentação, não permitindo que se consiga facilmente extrair semântica dos mesmos, sem que para isso seja feito um grande esforço de implementação. Considerando isso, a Web atual pode ser classificada como sintática e o processo de interpretação dos conteúdos disponibilizados fica geralmente a cargo dos usuários [Costa & Yamate 2009].

Em sua maior parte, os dados na Web ainda são organizados para serem lidos ou compreendidos por humanos e não por agentes de software. Para que um agente de software possa entender e interpretar um dado, é necessário processar a semântica envolvida naquele dado, num determinado contexto. Neste escopo, semântica diz respeito à atribuição de significado a elementos, dados ou expressões que precisam ser interpretados numa dada situação [Souza 2009]. No cenário da Web, isso representa atribuir significado aos dados interligando-os com outros conjuntos de dados ou outros domínios de conhecimento, conseguindo, assim, criar uma relação de significância entre os conteúdos publicados na Internet de modo que seja perceptível tanto pelo usuário quanto pelos agentes de software. Essa nova visão da Web vem sendo denominada de Web Semântica (*Semantic Web*) [Lee *et al.* 2001].

A Web Semântica é considerada uma extensão da Web atual cujo objetivo principal é facilitar a interpretação e integração dos dados na Web. Como parte do desenvolvimento da Web Semântica, surgiu o conceito de *Linked Data* (dados ligados) que pode ser definido como um conjunto de boas práticas para publicar e conectar conjuntos de dados estruturados na Web, com o intuito de criar uma “Web de Dados” [Bizer *et al.* 2009]. Estas práticas são fundamentadas em tecnologias Web, como HTTP (*Hypertext Transfer Protocol*) e URI (*Uniform Resource Identifier*), com o objetivo de permitir a leitura dos dados conectados semanticamente, de forma automática, por agentes de software. A Web de Dados cria inúmeras oportunidades para a integração semântica de dados, motivando o desenvolvimento de novos tipos de aplicações e ferramentas, como navegadores e motores de busca.

Este capítulo apresenta os conceitos básicos para publicação e consumo de dados na Web de acordo com os padrões de *Linked Data* e está organizado como segue. A seção 2 traça um paralelo entre a Web clássica e a Web de Dados. A Seção 3 apresenta uma visão geral sobre o tema *Linked Data*, introduzindo os princípios que regem esse conjunto de práticas. Além disso, são explicados os seguintes aspectos: (i) a nomeação de recursos através de URIs; (ii) como é possível utilizar URIs para prover navegação entre os recursos; (iii) como disponibilizar os dados através do modelo RDF, mostrando os benefícios oriundos desse modelo e (iv) como *links* entre os recursos são criados e identificados. A Seção 4 discute os aspectos básicos que devem ser considerados para a publicação de dados segundo os

princípios do *Linked Data*. Para tal, aborda questões de formatação e estruturação de dados e, por fim, apresenta um conjunto de padrões e diretrizes a serem seguidos com o intuito de tornar possível a publicação e interligação dos dados. A Seção 5 apresenta exemplos de aplicações existentes e, finalmente, a Seção 6 tece considerações sobre o tema exposto, apontando benefícios de seu uso e dificuldades ainda existentes.

4.2. Web de Documentos *versus* Web de Dados

Para um melhor entendimento sobre a Web de Dados, pode-se estabelecer um paralelo entre a Web de Documentos (a Web atual) e a Web de Dados. A primeira faz uso de navegadores HTML (*HyperText Markup Language*) para acessar dados na enquanto que na segunda os dados são acessados a partir de navegadores RDF. Na Web de Documentos *hiperlinks* são usados para navegar entre as páginas, enquanto que na Web de Dados os *links* RDF são usados para acessar dados de diversas fontes.

A Web de Documentos é baseada em um conjunto de padrões, incluindo: um mecanismo de identificação global e único, as URIs; um mecanismo de acesso universal, o HTTP e um formato padrão para representação de conteúdo, o HTML. De modo semelhante, a Web de Dados tem por base alguns padrões, como: o mesmo mecanismo de identificação e acesso universal usado na Web de documentos (as URIs e o HTTP); um modelo padrão para representação de dados, o RDF e uma linguagem de consulta para acesso aos dados, a linguagem SPARQL. A seguir esses padrões são apresentados.

URIs – Uniform Resource Identifiers

URI é uma cadeia de caracteres compacta usada para identificar ou denominar um recurso na Internet, onde um recurso pode ser um documento html, uma figura ou uma pessoa. O principal propósito desta identificação é permitir a interação com representações do recurso através de uma rede, tipicamente a Web, usando protocolos específicos. Uma URI pode ser classificado como um localizador URL (*Uniform Resource Locator*) ou um nome URN (*Uniform Resource Name*), ou ainda como ambos. O URN define a identidade de um item, enquanto que o URL nos dá um método para encontrá-lo. Tecnicamente URL e URN funcionam como IDs de recursos, no entanto, muitos conjuntos não podem ser categorizados como somente um ou outro, por que todas as URIs podem ser tratados como nomes, e alguns conjuntos integram aspectos de ambas ou de nenhuma das categorias.

No contexto *Linked Data*, URIs são usadas para identificar objetos e conceitos, permitindo que eles sejam dereferenciados para obtenção de informações a seu respeito. Assim, o dereferenciamento de uma URI 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.

HTTP – HyperText Transfer Protocol

HTTP é um protocolo de aplicação responsável pelo tratamento de pedidos e respostas entre cliente e servidor na Web. Ele surgiu da necessidade de distribuir informações pela Internet e, para que essa distribuição fosse possível, foi

necessário criar uma forma padronizada de comunicação entre os clientes e os servidores da Web. Com isso, o protocolo HTTP passou a ser utilizado para a comunicação entre computadores na Internet e a especificar como seriam realizadas as transações entre clientes e servidores, através do uso de regras básicas.

O HTTP utiliza o modelo cliente-servidor, como a maioria dos protocolos de rede, baseando-se no paradigma de requisição e resposta. Um programa requisitante (cliente) estabelece uma conexão com um outro programa receptor (servidor) e envia-lhe uma requisição, contendo a URI, a versão do protocolo, uma mensagem contendo os modificadores da requisição, informações sobre o cliente e, possivelmente, o conteúdo no corpo da mensagem. O servidor responde com uma linha de status (*status line*) incluindo sua versão de protocolo e com os códigos de erro informando se a operação foi bem sucedida ou não, seguido pelas informações do servidor, informações da entidade e possível conteúdo no corpo da mensagem. Após o envio da resposta pelo servidor, encerra-se a conexão estabelecida.

RDF – Resource Description Framework

A utilização de um modelo padrão para representação de dados, como o RDF, torna possível a implementação de aplicações genéricas capazes de operar sobre o espaço de dados global [Heath & Bizer 2011]. O modelo RDF é baseado no conceito de grafo, é extensível e possui um alto nível de expressividade, facilitando, dessa forma, a interligação entre dados de diferentes fontes.

Em RDF, um recurso pode estar relacionado com dados ou com outros recursos através das sentenças, as quais são estruturadas no formato sujeito + predicado + objeto, onde:

- **Sujeito:** Tem como valor o recurso sobre o qual se quer escrever uma sentença. Todo recurso deve ser capaz de ser identificado unicamente.
- **Predicado:** Especifica um relacionamento entre um sujeito e um objeto. O predicado é especificado por meio de propriedades, que são relações binárias geralmente nomeadas por um verbo e permitem relacionar um recurso a dados ou a outros recursos. Uma propriedade também é um recurso e, portanto, deve ter um identificador único.
- **Objeto:** Denomina o recurso ou dado que se relaciona com o sujeito. O valor de um objeto pode ser um recurso ou um literal, que pode ser um valor numérico ou uma cadeia de caracteres.

A Figura 4.1 mostra alguns exemplos de triplas RDF.

Sujeito	Predicado	Objeto
http://www.w3c.org/RDF/Validator/run/p91002043177	http://uni.org/uni/elements/1.1/nomeDocente	"Berna Farias"
http://www.w3c.org/RDF/Validator/run/ck120	http://uni.org/uni/elements/1.1/nomeDisciplina	"Banco de Dados I"
http://www.w3c.org/RDF/Validator/run/ck120	http://uni.org/uni/elements/1.1/EnsinadoPor	http://www.w3c.org/RDF/Validator/run/p91002043177

Figura 4.1. Triplas RDF.

As triplas representadas na Figura 4.1 contêm informações sobre dois recursos. A primeira tripla informa que o recurso “p91002043177” possui nome Berna Farias. A segunda tripla descreve um segundo recurso “CK120”, cujo nome é “Banco de Dados P”. A terceira tripla descreve um relacionamento entre os dois recursos criados através do predicado “EnsinadoPor”. 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 semiestruturados a dados altamente estruturados.

SPARQL

Assim como os sistemas de bancos de dados relacionais fazem uso do SQL para consultar registros nas suas bases de dados, SPARQL é a linguagem de consulta padrão recomendada pelo W3C para recuperação de informações contidas em grafos RDF. Apesar de existirem outras linguagens de consulta (SeRQL, RQL, etc..) mais antigas, maduras e com maior poder de expressividade que o SPARQL, estas ou foram projetadas para se trabalhar em um domínio específico ou são interpretadas apenas por algumas poucas ferramentas, o que acaba resultando em uma baixa interoperabilidade.

Semelhante ao SQL, o SPARQL possui uma estrutura Select-From-Where onde:

- **Select:** Especifica uma projeção sobre os dados como a ordem e a quantidade de atributos e/ou instâncias que serão retornados.
- **From:** Declara as fontes que serão consultadas. Esta cláusula é opcional. Quando não especificada, assumimos que a busca será feita em um documento RDF/RDFS particular.
- **Where:** Impões restrições na consulta. Os registros retornados pela consulta deverão satisfazer as restrições impostas por esta cláusula.

O resultado de uma consulta SPARQL pode ser encarado como um subgrafo resultante da execução da consulta sobre o grafo que representa o modelo. Considere por exemplo o grafo apresentado na Figura 4.2 extraído de [Allemang & Hendler 2008].

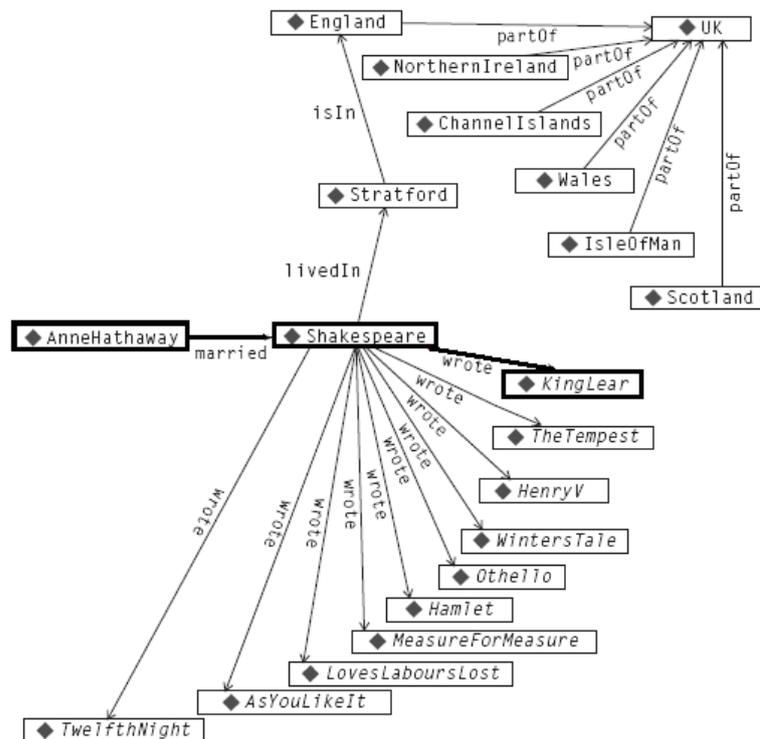


Figura 4.2. Representação das instâncias de um domínio

O grafo da Figura 4.2 representa a relação entre as instâncias de uma ontologia cujo domínio é focado na descrição e formalização de escritores. O subgrafo destacado em negrito pode ser encarado, por exemplo, como o resultado de uma consulta que retorna o escritor que escreveu o livro *King Lear* e é casado com *AnneHathaway*.

4.3. Princípios *Linked Data*

O termo *Linked Data* refere-se ao conjunto de melhores práticas para a publicação de dados estruturados na Web. Essas práticas foram introduzidas por Tim Berners-Lee em [Lee *et al* 2006] e resumem-se em quatro princípios básicos:

1. Usar URIs como nome para recursos.
2. Usar URIs HTTP para que as pessoas possam encontrar esses nomes.
3. Quando alguém procura por uma URI, garantir que informações úteis possam ser obtidas por meio dessa URI, as quais devem estar representadas no formato RDF.
4. Incluir *links* para outras URIs de forma que outros recursos possam ser descobertos.

O primeiro princípio defende o uso de referências URI para identificar, não apenas documentos Web e conteúdos digitais, mas também objetos do mundo real e conceitos abstratos.

O segundo princípio defende o uso de URIs HTTP para identificar os objetos e os conceitos abstratos definidos pelo princípio 1, possibilitando essas URIs serem dereferenciáveis sobre um protocolo HTTP. Neste contexto, dereferenciar é o processo de recuperar uma representação de um recurso identificado por uma URI, onde um recurso pode ter várias representações como documentos HTML, RDF, XML entre outros.

A fim de permitir que uma ampla gama de aplicações diferentes possam processar dados disponíveis na Web, é importante que exista um acordo sobre o formato padrão para disponibilização dos dados. O terceiro princípio *Linked Data* defende o uso de RDF como modelo para a publicação de dados estruturados na Web [Klyne *et al.* 2004]. Com o RDF, é possível descrever significado sobre recursos, habilitando agentes de software a explorar os dados de forma automática, muitas vezes, agregando, interpretando ou mesclando dados.

O quarto princípio diz respeito ao uso de hiperlinks para conectar não apenas os documentos da Web, mas qualquer tipo de recurso. Por exemplo, uma ligação pode ser fixada entre uma pessoa e um lugar, ou entre um local e uma empresa. Em contraste com a Web clássica onde os hiperlinks são em grande parte não tipados, hiperlinks que conectam os recursos em um contexto de *Linked Data* são capazes de descrever a relação entre eles. Hiperlinks no contexto de *Linked Data* são chamados de *links* RDF, a fim de distingui-los dos hiperlinks existentes na Web convencional [Heath & Bizer 2011].

O exemplo mais visível da adoção e aplicação dos princípios *Linked Data* tem sido o projeto *Linking Open Data*¹ fundado em janeiro de 2007 e apoiado pelo *W3C Semantic Web Education and Outreach Group*². O objetivo principal desse projeto é identificar conjuntos de dados disponíveis sob licenças abertas e convertê-los para RDF de acordo com os princípios *Linked Data*.

Os participantes nas fases iniciais do projeto foram os pesquisadores e desenvolvedores de laboratórios universitários e empresas de pequeno porte. Desde então, o projeto tem crescido consideravelmente, conseguindo um envolvimento significativo de grandes organizações como a BBC³. Este crescimento é possível graças à natureza aberta do projeto, onde qualquer um pode participar, sendo necessário apenas publicar um conjunto de dados de acordo com os princípios *Linked Data* e interligá-lo aos conjuntos de dados já existentes. Na Figura 4.3 podem-se observar os diversos conjuntos de dados publicados no contexto do projeto *Linking Open Data*.

¹ <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

² <http://www.w3.org/2001/sw/sweo/>

³ <http://www.bbc.co.uk/>

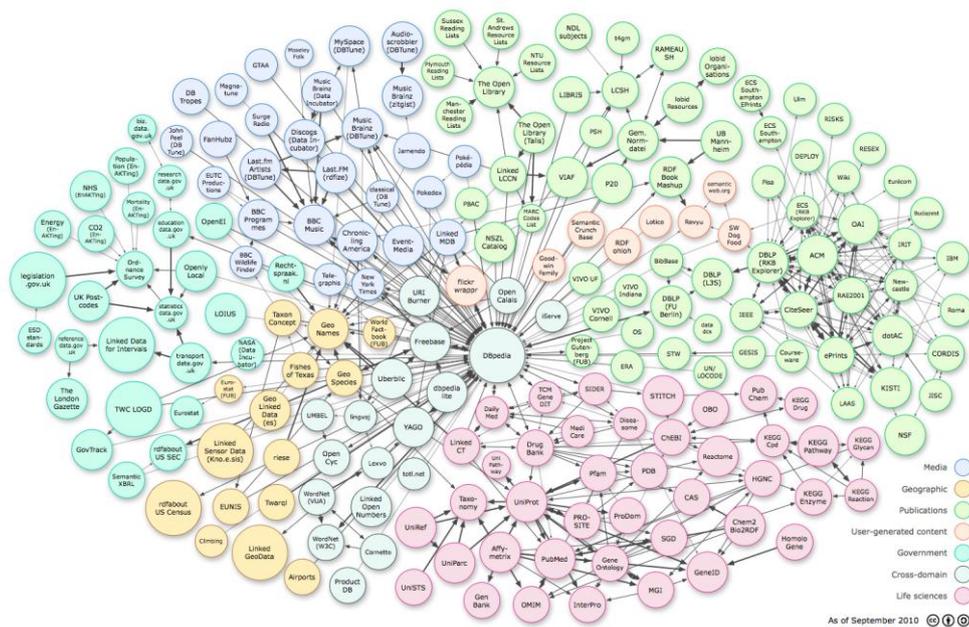


Figura 4.3. Visão geral de conjuntos de dados publicados e seus relacionamentos em Setembro de 2011.

No grafo da Figura 4.3, cada nó representa um conjunto de dados publicado seguindo os princípios *Linked Data*, os quais estão interligados com outros conjuntos de dados na nuvem. O tamanho de cada nó corresponde ao número de triplas RDF do conjunto de dados. As setas indicam a existência de pelo menos 50 ligações entre dois conjuntos, podendo ser unidirecionais, indicando que um certo conjunto contém triplas RDF de um outro conjunto, ou bidirecionais, indicando que ambos os conjuntos contêm triplas RDF um do outro.

O conteúdo da nuvem possui natureza diversa, incluindo dados sobre localizações geográficas, empresas, livros, publicações científicas, filmes, música, televisão e rádio, ensaios clínicos, comunidades online, dados estatísticos entre outros [Heath & Bizer 2011].

4.4. Diretrizes para Publicação de dados em *Linked Data*

De acordo com [Heath & Bizer 2011], a adoção das melhores práticas de publicação de dados ligados facilita a descoberta de informações relevantes para a integração de dados entre diferentes fontes. A seguir são apresentados alguns passos para publicar dados na Web de Dados, como definido em [Heath & Bizer 2011].

4.4.1. Criação de URIs adequadas

Como mencionado anteriormente, recursos são nomeados a partir de URIs. Dessa forma, ao publicar dados *Linked Data* é preciso fazer um esforço para criar boas URIs para identificação dos recursos. Para isso, devem ser feitas algumas considerações, incluindo:

- Usar URIs HTTP para tudo, tornando-as passíveis de serem dereferenciadas.
- Evitar URIs com detalhes de implementação ou do ambiente em que estão publicadas. Por exemplo, a URI <http://www.lia.ufc.br:8080/~danusarbc>

/index.php é um exemplo do que deve ser evitado, pois possui detalhes da porta 8080 usada em seu ambiente de publicação e do script implementado em PHP necessário para sua execução.

- Manter as URIs estáveis e persistentes, pois a alteração das URIs pode levar à quebra de *links* já estabelecidos, criando um problema para a localização de recursos. Para evitar esse tipo de alteração, recomenda-se um planejamento das URIs que serão usadas e também que o responsável pela publicação detenha a propriedade do espaço de nomes
- Muitas vezes será preciso usar algum tipo de chave primária dentro das URIs, para se certificar de que cada uma delas é única. Se possível, usar uma chave que é significativa dentro do domínio para o qual está sendo criada a URI. Por exemplo, quando se trata de livros, pode-se usar o ISBN como identificador único.

Essas são apenas algumas características desejáveis para serem consideradas quando se for criar uma URI. Maiores detalhes podem ser encontrados no tutorial *Cool URIs for the Semantic Web*⁴ disponibilizado pela W3C.

4.4.2. Usar URIs dereferenciáveis

Qualquer URI HTTP deve ser capaz de ser dereferenciada, o que significa que clientes HTTP podem procurar por uma URI usando um protocolo HTTP e recuperar uma descrição do recurso que é identificado pela URI.

A recuperação da representação mais adequada para o usuário é feita por meio da negociação de conteúdo. Assim, clientes HTTP enviam, por meio dos cabeçalhos HTTP (*Get, Head, Post, Put, Delete, Trace, Options, Connection*), o pedido para indicar qual tipo de conteúdo deseja obter. Após isso, ao receber uma requisição, o servidor analisa o cabeçalho e seleciona a resposta adequada. Há duas estratégias para fazer URIs serem dereferenciáveis: *303 URI* e *Hash URI*.

303 URI: 303 é um código de *status* de redirecionamento no qual o servidor pode dar a localização de um documento que contém informações sobre um recurso. Por exemplo, a Figura 4.4 mostra como ocorre o dereferenciamento quando um usuário deseja obter informações sobre a cidade de Berlin da fonte *DBpedia*⁵. Primeiramente, em 1, é especificado que o conteúdo desejado é no formato RDF/XML através do comando *Accept: text/html;q=0.5, application/rdf+xml*. Após isso, em 2, o servidor enviará para o cliente uma URI *http://dbpedia.org/data/Berlin*, mostrando a localização exata do conteúdo requerido e o código *303 See Other*, indicando que a requisição será redirecionada para essa URI. Com essa URI em mãos, em 3, o cliente reenvia sua solicitação e, por fim, em 4, recebe do servidor um arquivo RFD com o conteúdo conforme solicitado.

⁴ <http://www.w3.org/TR/cooluris/>

⁵ <http://dbpedia.org/>

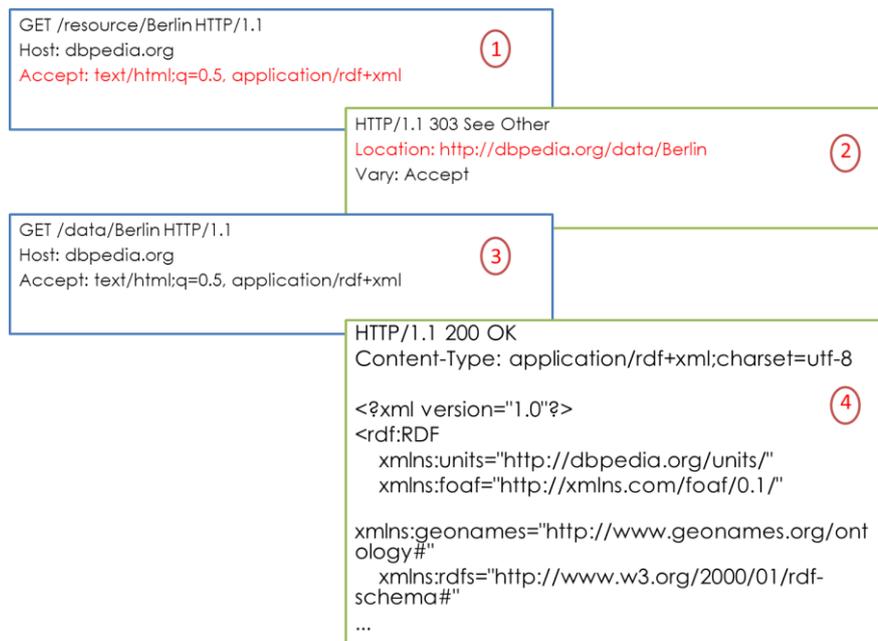


Figura 4.4. Dereferenciamento de URI utilizando a estratégia 303 URI.

Hash URI: Nessa estratégia, a URI contém um fragmento, uma parte especial que é separada do resto da URI pelo símbolo #. Quando um cliente deseja recuperar uma *hash* URI, ele remove tudo que vem após o símbolo # e envia o restante da URI para o servidor. Como resposta, o cliente recebe um documento completo com o conteúdo solicitado. Para um melhor entendimento, a Figura 4.5 mostra como ocorre o dereferenciamento utilizando *Hash URI* para o seguinte exemplo: suponha um arquivo que contém um vocabulário definido para uma empresa fictícia chamada *Small and Medium-sized Enterprises* representado pela seguinte URI: *http://biglynx.co.uk/vocab/sme*. As seguintes URIs foram criadas para identificar termos distintos que podem ser encontrados no documentos previamente criado: *http://biglynx.co.uk/vocab/sme#Small* *MediumEnterprise* e *http://biglynx.co.uk/vocab/sme#Team*. Para dereferenciar qualquer uma dessas duas URIs, o cliente removerá o fragmento especial (*#Team* por exemplo) e então enviará sua requisição para o servidor, especificando que o conteúdo desejado é no formato RDF/XML através do comando *Accept: application/rdf+xml* conforme visto em 1. Em seguida, o servidor retornará como resposta, em 2, um documento contendo todo o conteúdo expresso em *http://biglynx.co.uk/vocab/sme*.

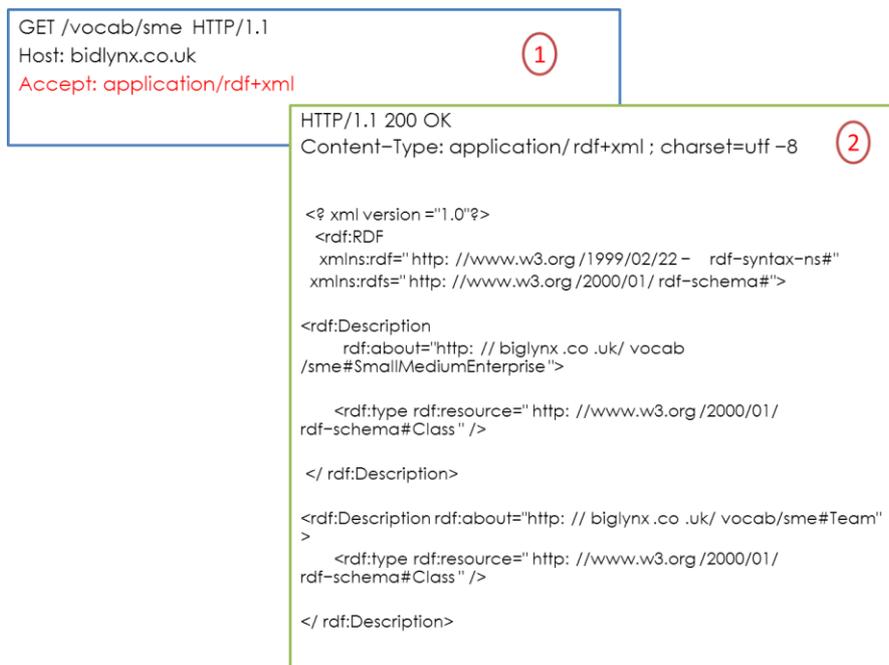


Figura 4.5. Dereferenciamento de URI utilizando a estratégia *Hash URI*.

4.4.3. Criação de *links* RDF

De modo a permitir a navegação entre as fontes de dados, devem ser criados *links* para outras fontes, seja de forma manual ou automatizada. *Links* no contexto de *Linked Data* fazem referência aos de *links* RDF. *Links* RDF podem ser classificados em dois tipos: *links* RDF internos e externos. O *link* RDF interno conecta recursos dentro de uma única fonte de dados *Linked Data*. *Links* externos conectam recursos os quais são provenientes de diferentes fontes de dados *Linked Data*. No caso de *links* externos, as URIs referentes ao sujeito e predicado do link pertencem a diferentes *namespaces* (os quais são URIs que referenciam o esquema do qual o elemento faz parte). *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 [Heath & Bizer 2011].

A Figura 4.6 apresenta dois exemplos de *links* RDF. O primeiro exemplo interliga o perfil *FOAF*⁶ do pesquisador Tim Berners-Lee localizado em um arquivo RDF ao recurso que o identifica na fonte de dados do *DBLP*⁷. No segundo exemplo, o recurso que identifica Tim Berners-Lee na fonte *DBpedia*⁸ 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.

Sujeito: <http://www.w3.org/People/Berners-Lee/card#i>

⁶ <http://www.foaf-project.org/>

⁷ <http://www.informatik.uni-trier.de/~ley/db/>

⁸ <http://dbpedia.org/About>

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 4.6. Exemplos de Links RDF.

Ao se criar *links* RDF é preciso estabelecer relações entre os termos dos vocabulários entre as fontes que estão sendo interligadas. 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 dados ligados pelas aplicações clientes. Novos termos só devem ser definidos se não forem encontrados em vocabulários já existentes. Alguns vocabulários bastante conhecidos são: *Friend-of-a-Friend* (FOAF), *Semantically-Interlinked Online Communities*⁹ (SIOC), *Simple Knowledge Organization System*¹⁰ (SKOS), *Description of a Project*¹¹ (DOAP), *Creative Commons*¹² (CC) e *Dublin Core*¹³ (DC). Propriedades como: *owl:equivalentClass*, *owl:equivalentProperty*, *rdfs:subClassOf*, *rdfs:subPropertyOf* podem ser usadas para estabelecer equivalências entre os termos dos vocabulários citados. Maiores detalhes sobre as linguagens OWL e RDFS podem ser encontrados em [Filho & Lóscio 2009].

4.4.4. Explicitar formas de acesso adicional aos dados

Para acessar os dados das fontes *Linked Data* é preciso realizar consultas SPARQL sobre as fontes. Para enviar consultas e recuperar resultados através da linguagem SPARQL é usado o protocolo SPARQL¹⁴. 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 em diferentes formatos como XML, JSON¹⁵, texto, RDF/XML, NTriples¹⁶, Turtle¹⁷ ou N3¹⁸ e HTML [Magalhães *et al* 2011]. O *framework* Jena¹⁹ disponibiliza duas implementações de SPARQL *endpoints*: o *Joseki*²⁰ e o *Fuseki*²¹. Ambos suportam o protocolo e a linguagem SPARQL, permitem a realização de consultas sobre arquivos RDF e RDF

⁹ <http://sioc-project.org/>

¹⁰ <http://www.w3.org/2009/08/skos-reference/skos.html>

¹¹ <http://trac.usefulinc.com/doap>

¹² <http://creativecommons.org/>

¹³ <http://dublincore.org/>

¹⁴ <http://www.w3.org/TR/rdf-sparql-query/>

¹⁵ <http://www.json.org/>

¹⁶ <http://www.w3.org/2001/sw/RDFCore/ntriples/>

¹⁷ <http://www.w3.org/TR/turtle/>

¹⁸ <http://www.w3.org/TeamSubmission/n3/>

¹⁹ <http://incubator.apache.org/jena/>

²⁰ <http://www.joseki.org/>

²¹ <http://openjena.org/wiki/Fuseki>

Triple Stores, e implementam *SPARQL Update*²² (linguagem para atualizar RDF store). 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*²³ 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 [Magalhães *et al* 2011].

4.4.5. Padrões para Publicar Dados *Linked Data*

Para a publicação de dados RDF podem ser usadas ferramentas específicas de conversão de planilhas, arquivos CSV, arquivos XML, dados relacionais e outros documentos para o formato RDF como, por exemplo, a ferramenta *ConvertToRDF*²⁴. Após a geração do arquivo em formato RDF, os dados podem ser carregados em um banco de dados que armazena as triplas RDF, chamado de *RDF Store*. A publicação de dados de uma *RDF Store*, seguindo os princípios *Linked Data*, tipicamente envolve a disponibilização de uma interface para acesso aos dados *Linked Data* e de um *SPARQL endpoint* para acesso aos dados. Uma vantagem de se converter dados para o formato RDF 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.

Um outra forma de acessar dados que não estão no modelo RDF consiste em fornecer uma visão RDF. Isso é feito através de um *RDF Wrapper*. Nesse caso, a conversão é realizada por um *RDF Wrapper* por meio de 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 para dados que precisam ser acessados através de uma Web API. Criar uma visão RDF tende a ter um desempenho inferior à conversão de dados para RDF devido às traduções dinâmicas entre os modelos que deve ser realizada a cada uso da visão RDF. No entanto, o uso de *RDF Wrappers* traz algumas 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.

Um exemplo comum de utilização de visões RDF é a publicação de dados armazenados em banco de dados relacionais. O *RDB-to-RDF*²⁵ *Wrappers* é uma solução que cria visões RDF a partir de mapeamentos entre as estruturas relacionais e os grafos RDF. A plataforma *D2RQ*²⁶ é um exemplo de *RDB-to-RDF Wrappers*, que fornece toda a infraestrutura necessária para acessar bancos de dados relacionais como grafos RDF virtuais. Esta plataforma 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.

²² <http://www.w3.org/Submission/SPARQL-Update/>

²³ <http://openjena.org/TDB/>

²⁴ <http://www.w3.org/wiki/ConverterToRdf>

²⁵ <http://www.w3.org/TR/r2rml/>

²⁶ <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/spec/>

- **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 D2R²⁷** 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 [Bizer & Cyganiak 2006].

Além do D2R, duas outras ferramentas destacam-se como RDB-to-RDF *Wrappers*: o Virtuoso *RDF Views* [Erling & Mikhailov 2006] e o *Triplify* [Auer *et al.* 2009]. Este último é um *plugin* para aplicações Web que permite mapear os resultados de consultas SQL em RDF e JSON. Depois disso, os dados podem ser compartilhados e acessados na Web de dados.

Após gerar os dados no modelo RDF, é necessário verificar se o resultado está de acordo com os princípios *Linked Data*. Essa verificação pode ser feita através de ferramentas de validação como, por exemplo, *Sindice Web Data Inspector*²⁸, *EyeBall*²⁹ e *W3C Validation Service*³⁰.

A próxima seção apresentará uma série de aplicações Web que vem sendo desenvolvidas para consumir os dados publicados nos padrões *Linked Data*.

4.5. Consumo de dados ligados

Nos últimos três anos, um número significativo de dados vem sendo disponibilizado de acordo com os princípios *Linked Data*. Como resultado, uma série de aplicações Web estão sendo desenvolvidas para explorar a Web de Dados. Segundo [Bizer *et al.* 2009], essas aplicações podem ser classificadas em três categorias: *browsers*, motores de buscas e aplicações para domínios específicos. Essa seção examinará cada uma dessas categorias.

4.5.1. *Browsers Linked Data*

Assim como os tradicionais *browsers* da Web clássica permitem aos usuários navegarem por páginas HTML, os *browsers Linked Data* permitem aos usuários navegar por fontes de dados seguindo os *links* expressos nas triplas RDF. Por meio destes *browsers* é possível percorrer os *links* RDF, explorando e descobrindo novas informações na Web. A seguir, serão apresentados alguns exemplos de *browsers* usados para acessar dados ligados.

*Tabulator*³¹ permite ao usuário percorrer a Web de Dados e expor parte dos dados de uma forma controlada, onde o usuário pode dividir a informação por tópicos. Os resultados das consultas podem ser analisados através de vários métodos, que vão desde a apresentação de dados de maneira convencional até a apresentação por meio de mapas. A utilização do modo de exploração inicia a partir da submissão de uma URI. Depois disso, o *Tabulator* obtém informações sobre o recurso e as

²⁷ <http://www4.wiwiw.fu-berlin.de/bizer/d2r-server/>

²⁸ <http://inspector.sindice.com/>

²⁹ <http://jena.sourceforge.net/Eyeball/>

³⁰ <http://www.w3.org/RDF/Validator/>

³¹ <http://www.w3.org/2005/ajar/tab>

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. Os resultados podem ser exibidos através das visões de tabela, mapas, calendários ou linhas de tempo. Pode-se iniciar uma nova exploração pela seleção de um detalhe de uma das visões. O *Tabulator* pode ser usado como um complemento do navegador Firefox ou como uma aplicação Web que atualmente é compatível apenas com o Firefox [Lee *et al.* 2006].

*Marbles*³² é um aplicativo do lado do servidor que formata o conteúdo da Web Semântica para clientes XHTML. Pontos coloridos são usados para correlacionar a origem dos dados apresentados com as fontes de dados de onde foram encontrados. Os dados são recuperados de múltiplas fontes e integrados em um único grafo que é mantido através das sessões do usuário. Além de dereferenciar a URI, *Marbles* consulta os mecanismos de busca *Sindice* e *Falcons* em busca de fontes que contenham informações sobre o recurso referenciado pela URI dada como entrada para a busca. Além disso, os predicados *owl:sameAs* e *rdfs:seeAlso* são seguidos para obtenção de informações adicionais sobre o recurso. A Figura 4.7 apresenta como o *Marbles* disponibiliza os dados para o usuário após uma consulta. Os pontos coloridos indicam as fontes de dados que forneceram os dados para o usuário. Dessa forma, o usuário pode clicar sobre as fontes e encontrar mais informações para a sua busca.

The screenshot shows the Marbles browser interface for the URI `http://www.w3.org/People/Berners-Lee/card#1`. The main heading is "Tim Berners-Lee". Below this, several RDF properties are listed, each with associated data and source indicators (colored dots):

- Person:** Includes a list of source icons.
- label:** "Tim Berners-Lee" with source icons.
- sameAs:** "Tim Berners-Lee (also at www1.wiss.fu-berlin.de)" with source icons.
- image:** A photograph of Tim Berners-Lee with source icons.
- Web links:** "[http://www.w3.org/People/Berners-Lee/](\"http://www.w3.org/People/Berners-Lee/\")" with source icons.
- name:** "Tim Berners-Lee", "Timothy Berners-Lee", and "Tim Berners Lee" with source icons.
- Given name:** "Timothy" with source icons.
- family name:** "Berners-Lee" with source icons.
- sha1sum of a personal mailbox URI name:** "9850f7c5a70db7407210cef6e4e6f5374a525c5c" with source icons.
- workplace homepage:** "[http://www.w3.org/](\"http://www.w3.org/\")" with source icons.
- uriAvatars:** "TimBL" with source icons.
- nickname:** "TimBL" and "timbl" with source icons.
- personal mailbox:** "mailto:timbl@w3.org" with source icons.
- seeAlso:** "Tim Berners-Lee's FOAF file" and "Tim Berners-Lee's FOAF file" with source icons.

Figura 4.7. Visualização de Informações sobre recursos do *Browser Marbles*.

³² <http://marbles.sourceforge.net/>

*Disco Hiperdata Browser*³³ é uma aplicação Web usada como navegador simples para visualizar informações sobre um recurso por meio de uma 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, o *browser* recupera as informações sobre o recurso e as exibe em uma tabela contendo propriedades, valores e as fontes das quais os dados foram recuperados. 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 executado localmente.

Além das aplicações descritas acima, destacam-se: *Dipper*³⁴, *Piggy Bank*³⁵, *URI Burner*³⁶, *LinkSailor*³⁷ e *Graphite RDF Browser*³⁸ como *browsers* simples e rápidos para obter detalhes sobre uma determinada URI após dereferenciá-la.

4.5.2. Motores de Busca *Linked Data*

Um grande número de motores de busca tem sido desenvolvido para rastrear dados no padrão *Linked Dados*. Esses mecanismos de busca permitem localizar recursos de diferentes fontes por meio de palavras-chave. A consulta pode ser realizada pelo usuário através de uma interface Web ou através de serviços Web oferecidos pelos mecanismos de busca. A seguir são apresentados alguns dos motores de busca mais utilizados.

Falcons permite tanto uma busca por palavras-chave, como oferece ao usuário a opção de buscar objetos, conceitos e documentos, possibilitando uma apresentação dos resultados um pouco diferente dos demais motores de busca. A busca por objeto é adequado para a busca por pessoas, lugares e outros itens considerados concretos, enquanto que a busca por conceito é orientada para a localização de classes e propriedades em ontologias publicadas na Web. O recurso de pesquisa por documento segue uma abordagem mais tradicional, onde os resultados apontam para documentos RDF que contêm os termos de pesquisa especificados [Cheg & Qu 2011].

*Sindice*³⁹ 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 que permite a utilização

³³ http://www4.wiwiwiss.fu-berlin.de/rdf_browser/

³⁴ <http://api.talis.com/stores/iand-dev1/items/dipper.html>

³⁵ <http://simile.mit.edu>

³⁶ <http://linkeddata.uriburner.com>

³⁷ <http://linksailor.com/>

³⁸ <http://graphite.ecs.soton.ac.uk/browser/>

³⁹ <http://sindice.com/>

de seus serviços por outras aplicações [Oren *et al* 2008]. Na Figura 4.8 pode-se observar como o motor de busca *Sindice* apresenta o resultado de uma consulta ao usuário.

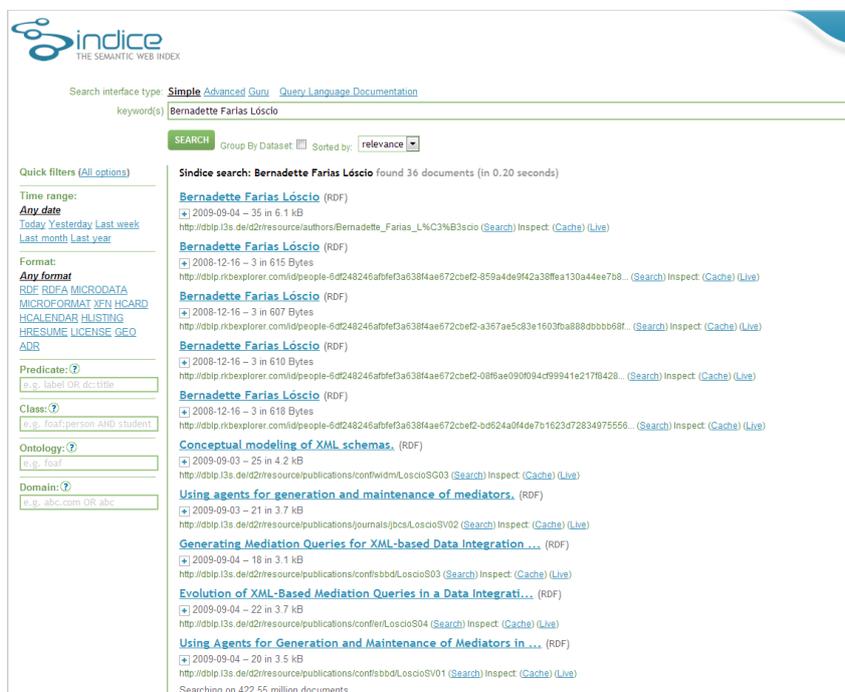


Figura 4.8. Resultado mostrado pelo Sindice sobre a pesquisadora Bernadette Farias Lóscio.

*Sig.ma*⁴⁰ busca dados estruturados a partir de uma palavra-chave e os exhibe 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. Os filtros aplicados nas fontes pelos usuários 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 4.9 ilustra o resultado de uma consulta sobre a pesquisadora Damires Yluska envolvendo três fontes nas quais ela é referenciada.

⁴⁰ <http://sig.ma/>

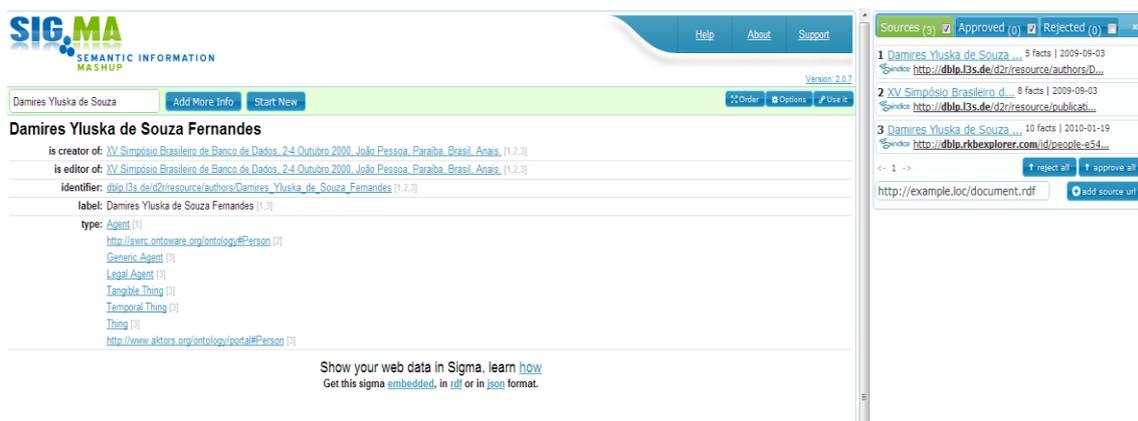


Figura 4.9. Visão criada pelo Sig.ma sobre a pesquisadora Damires Yluska.

*Watson*⁴¹ e *Swoogle*⁴² 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. Yahoo e Google também deram início ao uso de *Linked Data*. Yahoo provê acesso aos dados através da BOSS API⁴³, enquanto o Google usa os dados para a *Social Graph API*⁴⁴.

4.5.3. Aplicações para Domínios Específicos

Enquanto os *browsers Linked Data* e motores de busca descritos anteriormente fornecem funcionalidades muito genéricas, uma série de serviços, chamadas de *Linked Data Mashups*, foram desenvolvidos com o objetivo de prover funcionalidades mais específicas e de acordo com determinados domínios de dados. A seguir descreveremos algumas dessas aplicações [Lee *et al* 2006].

*Revyu*⁴⁵ 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*⁴⁶ 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 possam ser usadas por outras aplicações [Becker & Bizer 2008].

*Talis Aspire*⁴⁷ é uma aplicação Web voltada para que alunos e professores e possam encontrar os principais recursos educacionais em universidades do Reino

⁴¹ <http://watson.kmi.open.ac.uk/WatsonWUI/>

⁴² <http://swoogle.umbc.edu/>

⁴³ http://developer.yahoo.com/search/boss/boss_api_guide/

⁴⁴ <http://code.google.com/intl/pt-BR/apis/socialgraph/>

⁴⁵ <http://revyu.com/>

⁴⁶ <http://beckr.org/DBpediaMobile/>

⁴⁷ <http://www.talisaspire.com/>

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*⁴⁸ e *BBC Music*⁴⁹ 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*⁵⁰.

LinkedDataBr é um projeto brasileiro que visa construir uma infra-estrutura de suporte à criação de repositórios de dados governamentais públicos utilizando os padrões de *Linked Data*. A ideia consiste em utilizar e ligar dados governamentais disponíveis na Web, gerados a partir das iniciativas de *e-gov* e *open-government*, de forma a possibilitar a publicação padronizada e o acesso por parte dos cidadãos e organizações [Campos 2010].

4.6. Considerações Finais

O imenso emaranhado de documentos acessíveis na Web é composto de dados e informações de praticamente todas as áreas do conhecimento humano. Contudo, ainda é árdua a tarefa de prover meios eficientes que permitam aproveitar todo esse conteúdo, que pode ser composto tanto por dados estruturados, como os dados provenientes de bancos de dados relacionais, quanto por dados não estruturados, como textos e dados multimídia. No cenário da Web atual, o grande volume de dados e a falta de metadados dificultam o acesso à informação útil, específica e relevante.

Neste contexto, espera-se que o uso dos princípios do *Linked Data* possibilite a transformação de uma Web na qual os recursos são documentos HTML para uma Web de Dados, onde os dados estarão interligados através de metadados. O tema *Linked Data* traz novos desafios para o desenvolvimento de aplicações Web de uma maneira geral, bem como para o gerenciamento da grande nuvem de dados que vem se formando como resultado da crescente adoção dos princípios do *Linked Data*. Tendo em vista a relevância deste assunto para a comunidade de Computação e o grande potencial de pesquisa desta área, *Linked Data* tem sido o foco de diversas conferências internacionais, bem como o foco de estudo de diversos grupos de pesquisa. Dessa forma, torna-se de fundamental importância que este tema seja amplamente abordado e discutido por pesquisadores, alunos e profissionais da área de Computação.

⁴⁸ <http://www.bbc.co.uk/programmes>

⁴⁹ <http://www.bbc.co.uk/music>

⁵⁰ <http://musicbrainz.org/>

Referências

- [Allemang & Hendler 2008] Allemang, D., Hendler, D. (2008) *Semantic Web for the Working Ontologist*, 1st edition. Morgan Kaufmann publ., Amsterdam, Netherlands.
- [Auer et al. 2009] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumüller, 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 & Bizer 2008] Becker, C., Bizer, C. (2008) *DBpedia Mobile: A Location-Enabled Linked Data Browser*. In Linked Data on the Web (LDOW2008).
- [Bizer & Cyganiak 2006] Bizer, C., Cyganiak, R. (2006) *D2R Server – Publishing Relational Databases on the Semantic Web*. In 5th International Semantic Web Conference.
- [Bizer et al 2009] Bizer C., Heath T., Berners-Lee T. (2009) *Linked data - the story so far*. Int. J. Semantic Web Inf. Syst., 5(3):1–22, 2009.
- [Campos 2010] Campos M. L. (2010) *GT-LinkedDataBR – Exposição, compartilhamento e conexão de recursos de dados abertos na Web (Linked Open Data)*. Disponível em http://www.rnp.br/pd/gts2010-2011/gt_linkeddataabr.html
- [Cheng & Qu 2011] Cheng, G., Qu, Y. (2011) *Searching Linked Objects with Falcons: Approach, Implementation and Evaluation*. International Journal on Semantic Web and Information Systems, Special Issue on Linked Data.
- [Costa & Yamate 2009] Costa A., Yamate F. (2009) *Semantic Lattes: uma ferramenta de consulta baseada em ontologias*. Trabalho de Graduação em Engenharia de Computação - Escola Politécnica. IME/USP.
- [Erling & Mikhailov 2006] Erling, O., Mikhailov, I. (2006) *Mapping Relational Data to RDF in Virtuoso*. <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF>.
- [Filho & Lóscio 2009] Filho, F. W. B. H , Lóscio B. F. (2009) *Web Semântica: Conceitos e Tecnologias*. In Anais do ERCEMAPI (Escola Regional de Computação Ceará – Maranhão – Piauí).
- [Freitas 2003] Freitas, F. L. G. (2003) *Ontologias e a Web Semântica*. XXIII Congresso da Sociedade Brasileira de Computação. JAI. Campinas, São Paulo, Junho de 2003.
- [Gruber 1995] Gruber T. (1995) *Toward principles for the design of ontologies used for knowledge sharing*. 1995. International Journal Human-Computer Studies Vol. 43, Issues 5-6, November 1995, p.907-928.
- [Heath & Bizer 2011] Heath, T., Bizer, C. (2011) *Linked Data: Evolving the Web into a Global Data Space* (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool, 2011.

- [Klyne *et al* 2004] Klyne, G., Carroll, JJ., McBride., B. (2004) *Resource description framework (RDF): Concepts and abstract syntax*. Disponível em: <http://www.w3.org/TR/rdf-concepts/>
- [Lee *et al* 2006] Lee, B. 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 Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06), page 06.
- [Lee *et al* 2001] Lee, B. T., Hendler J., Lassilia O. (2001) *The semantic web*. Scientific American, 284(5):34–44, Mai 2001. <http://dx.doi.org/10.1038/scientificamerican0501-34> DOI: 10.1038/scientificamerican0501-34
- [Magalhães *et al* 2011] Magalhães, R. P., Macedo, J. A. F., Vidal, V. M. P. (2011) *Linked Data: Construindo um Espaço de Dados Global na Web*. In Anais do XXIV Simpósio Brasileiro de Banco de Dados. Outubro de 2011.
- [Oren *et al* 2008] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., and Tamma-relo, G. (2008) *Sindice.com: a document-oriented lookup index for open linked data*. Int. J. Metadata Semant. Ontologies, 3:37–52.
- [Souza 2009] Souza D. (2009) *Using Semantics to Enhance Query Reformulation in Dynamic Distributed Environments*. PhD Thesis, Federal University of Pernambuco (UFPE), Recife, PE, Brazil.