Linked Data

Structured data on the Web

David Wood
Marsha Zaidman
Luke Ruth
WITH Michael Hausenblas

FOREWORD BY Tim Berners-Lee

SAMPLE CHAPTER

MANNING

Linked Data

by David Wood
Marsha Zaidman
Luke Ruth
and Michael Hausenblas

Chapter 3

brief contents

3 Consuming Linked Data

This chapter covers

- Thinking the Web way
- Finding Linked Data on the Web
- Retrieving Linked Data from web pages
- Combining Linked Data from multiple sources
- Displaying basic Linked Data in HTML

The World Wide Web that most of us envision is technically a subset better defined as the Web of Documents (the Classic Web). Another facet of the WWW is the Web of Data. You should think of the Semantic Web as a web of data that can be processed directly and indirectly by machines.

Just as web documents employ hyperlinks to connect to each other, linked datasets establish connections through the use of RDF links between data items in different datasets. Linked Data conforms to a set of principles for publishing structured data on the World Wide Web.

This chapter will facilitate your understanding of the Web of Data by demonstrating how you can be a consumer of its content. We'll demonstrate how Linked Data is distributed and utilized. We'll show how you can use tools to find embedded Linked Data. We'll illustrate how you can develop programs that retrieve Linked

Data from one source and use those results to retrieve additional data from a different source. We expect that you'll gain a better understanding of the Web of Data and how various companies are utilizing Linked Data to better serve their customers.

3.1 Thinking like the Web

Thinking like the Web is important because it enables you to make the best use of this resource. Thinking like the Web means recognizing that the Web provides a global information space that flourishes because embedded links establish relationships among published resources. These resources are stored on different servers in different physical locations. People and machines can traverse these hyperlinks and uncover new information. Search engines can index these links and infer relationships between the documents. By using unambiguous URIs, you facilitate the inferring of relationships.

In general, people search the Web of Documents and manually aggregate related information to fulfill their needs. The volume of hits and ambiguity of unstructured data complicate the task of assembling truly relevant information. For example, how would you know that the Marsha Zaidman in Facebook is the same as the Marsha Zaidman in Twitter? After all, names aren't unique identifiers. If both references to Marsha Zaidman use the same URI, then the identification is unique and the association is obvious.

Here's an example of unstructured data. Imagine an HTML document that contains the following text:

> *There is a person "Anakin," known as "Darth Vader" and "Anakin Skywalker." Anakin Skywalker has a wife named Padme Amidala.*

Here's similar information presented as structured data using the RDF Turtle format:

```
@base <http://rosemary.umw.edu/~marsha/starwars/foaf.ttl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rel: <http://purl.org/vocab/relationship>.
@prefix stars: <http://www.starwars.com/explore/encyclopedia/characters/> .
 <me> a foaf:Person;
        foaf:family_name "Skywalker";
        foaf:givenname "Anakin";
        foaf:nick "Darth Vader";
        rel:Spouse_Of <stars:padmeamidala/> .
```

Resolvable URIs eliminate ambiguity over references.

In this sample of structured data, the URIs are resolvable and hence eliminate ambiguity over the identity of Anakin Skywalker and Padme Amidala, his wife. The format of the structured data is predictable and lends itself to machine readability and may be used as input to other applications. Unlike unstructured data, these statements are in a predictable format, precise, and unambiguous. Historically, data on the Web was published as unstructured data in disparate, incompatible formats that impaired machine readability and automated aggregation of related data. Although multiple RDF formats are used in Linked Data, they're compatible because they share a common data model. Therefore, within the RDF formats, you can select one that works best for your particular context without sacrificing interoperability with other data.

In thinking like the Web, you're recognizing the distributed and interconnected highway of information. In using structured data, you're enabling machine readability and indexing of this data. In interlinking published data on the Web, you're enabling reuse of your information. In short, you're facilitating the sharing of information on the Web.

3.2 How to consume Linked Data

In this example of how to consume Linked Data, we'll raise a single question and show you how numerous data sources are linked. The ultimate benefit is that you'll learn about Linked Data resources and observe how this data is interlinked across the Web. When these links are traversed manually, you're employing the follow-your-nose method of discovery. Suppose you wanted to know: is President Barack Obama a *Star Wars* fan? Prior to reading this book, you'd likely try one of the popular search engines like Google, Yahoo!, or Bing to help you answer this question. But in this section, we're going to attempt to answer this question using Linked Data.

In general, you can start with any web resource that contains links and then follow your nose from one link to the next. One possible starting point is http://data.NYTimes.com, a portal to a Linked Open Data (LOD) site of *New York Times* sources. We selected this starting point because it's a useful and reliable resource. This page provides an interface that facilitates human browsing of individual records. Selecting "O" and then searching on "Obama, Barack" brings you to a unique URI (http://data.nytimes.com/47452218948077706853). Figure 3.1 *The New York Times* LOD for Barack Obama, is a sample of the HTML display of the subject "Obama, Barack." This page contains a link to a topics page, http://topics.nytimes.com/top/reference/timestopics/people/o/barack_obama/index.HTML.

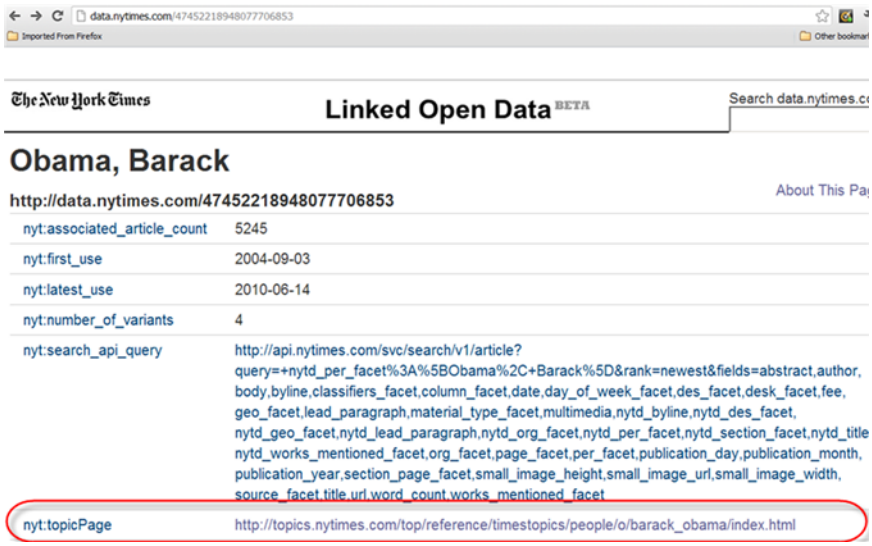

Figure 3.1 ***The New York Times* LOD for Barack Obama**

The topics page contains references and links to a broad variety of information about the subject and other related information. Although this source doesn't provide the answer to our question, it does provide links to other resources that could be useful. Because Obama was a resident of Chicago, perhaps the link to the *Chicago Tribune* would yield further information about Obama and his interest in *Star Wars*. Following these links takes you to the Barack Obama watch maintained by the *Chicago Tribune* (http://www.chicagotribune.com/topic/politics/government/barack-obama-PEPLT007408.topic).

From the *Chicago Tribune* page, you can search for articles related to *Star Wars* and follow the links to http://latimesblogs.latimes.com/washington/2009/09/obama-lightsaber-remix.HTML. This page contains links to http://latimesblogs.latimes.com/washington/2009/09/obi-wan-obama-white-house-olympics.HTML, shown in figure 3.2, and http://www.geeksofdoom.com/2009/09/17/greek-cred-president-obama-with-lightsaber/, shown in figure 3.3. Although we've not yet totally confirmed that Obama is an avid *Star Wars* fan, we've demonstrated the interlinked nature of the Web. We've demonstrated the unintentional reuse of information and benefitted from the unambiguous connections to the RDF data stored in *the New York Times* LOD site. Could you have predicted that you'd navigate from *the New York Times* LOD stores to the *Chicago Tribune*, the *Los Angeles Times*, and the Geeks of Doom website in your quest? Would a Google search have been easier? Maybe, but you wouldn't have been able to automate such a search as you'll learn to do using Linked Data.

Data and documents are distributed over many sites. You navigate from site to site following the links as you might follow the clues in a scavenger hunt. A user or a web spider can follow these links, amassing related data along the way, much like humans follow clues in a scavenger hunt. The next section shows additional examples.

Figure 3.2 **Obama using a light saber to defend Chicago as his choice for the Olympics**

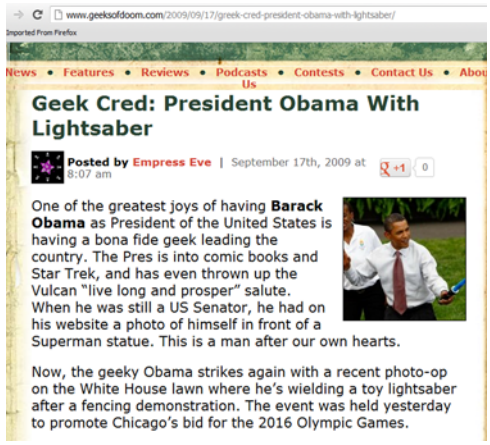

Figure 3.3 **Obama unveiled as a sci-fi fan**

3.3 Tools for finding distributed Linked Data

Let's assume that you need to find Linked Data for an application that you're developing. Sometimes you may already be aware of a previously published dataset that you can use. More often you need to find such data. Most likely this data is distributed across multiple sources. Numerous tools, such as Sindice, sameas.org, and the Data Hub, that facilitate finding Linked Data are available. You'll find other uses for these tools in chapter 8.

3.3.1 Sindice

One useful technology is Sindice (http://sindice.com/main/about), the Semantic Web Index. Sindice does for data what Google does for documents. The objective of Sindice is to provide multiple services: interactive data visualization and validation services, discovery and indexing of data, and search and query services. Sindice describes itself as a platform to build applications on top of semantic data. Sindice collects web data in many ways following existing web standards and updates its holdings frequently. A search using "Star Wars Episode I The Phantom Menace" resulted in identifying more than 2500 documents containing Linked Data that were distributed over 100 sets of Linked Data. Figure 3.4 is a screen shot of these results.

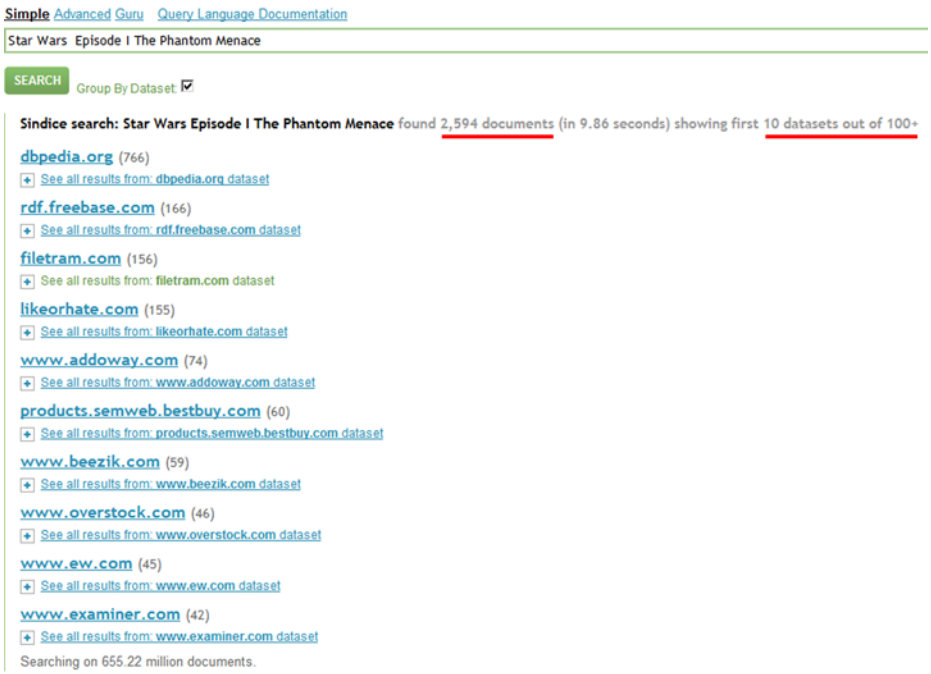

Figure 3.4 Sindice search outcomes for "Star Wars Episode I The Phantom Menace"

```
1.http://dbpedia.org/resource/Ep._1
■ ■ ■
57.http://dbpedia.org/resource/Star_Wars_Episode_I:_The_Phantom_Menace_(comic)
58.http://dbpedialite.org/things/50793#id
■ ■ ■
107.http://dbtropes.org/resource/Main/ThePhantomMenace
108.http://mpii.de/yago/resource/Invasion_of_Naboo
109.http://rdf.freebase.com/ns/m.0ddt_
110.http://rdf.freebase.com/ns/m.0d4dp5
111.http://rdf.freebase.com/ns/m.0f46bd
112.http://rdf.freebase.com/ns/en.star_wars_the_queens_gambit
113.http://rdf.freebase.com/ns/guid.9202a8c04000641f800000000006333f
114.http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000000c232a5
115.http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000000d2194c
116.http://rdf.freebase.com/ns/en.star_wars_episode_i_the_phantom_menace
117.http://sw.opencyc.org/concept/Mx4rvy4d7pwpEbGdrcN5Y29ycA
118.http://sw.opencyc.org/2008/06/10/concept/Mx4rvy4d7pwpEbGdrcN5Y29ycA
119.http://sw.opencyc.org/2009/04/07/concept/Mx4rvy4d7pwpEbGdrcN5Y29ycA
120.http://sw.opencyc.org/2008/06/10/concept/en/StarWarsThePhantomMenace_TheMovie
121.http://sw.opencyc.org/2009/04/07/concept/en/StarWarsThePhantomMenace_TheMovie
122.http://umbel.org/umbel/ne/wikipedia/Invasion_of_Naboo
```

Figure 3.5 **Equivalent URIs for dbpedia.org/resource/Star_Wars_Episode_I:_The_Phantom_Menace**

3.3.2 SameAs.org

Another avenue for discovery of Linked Data is SameAs.org (http://www.sameas.org). Its objective is to identify equivalent URIs to the Linked Data URI entered and provide an entry point to perform a Sindice search on a general search term.

One such search (see figure 3.5) performed using dbpedia.org/resource/Star_Wars_Episode_I:_The_Phantom_Menace yielded 122 equivalent URIs for http://dbpedia.org/resource/Star_Wars_Episode_I:_The_Phantom_Menace. In reviewing those lists of results, you can see that these 122 equivalents are contained in different datasets such as dbpedia.org/resource, dbpedialite.org/things, and rdf.freebase.com/ns.

3.3.3 Data Hub

The Data Hub (http://thedatahub.org) is a community-run catalog of useful sets of Linked Data on the Web. Here you can search and collect links from around the Web. The Data Hub is an openly editable open data catalog, in the style of Wikipedia. Most of the data indexed at the Data Hub is openly licensed, so if you find relevant data, you can use it. Unfortunately, the datasets recovered by our *Star Wars* example aren't relevant. Conducting our search using terms like "Star Wars Episode I" and "Star Wars The Phantom Menace" yielded no results. Trying a more general term yielded useful results. As you can see from figure 3.6, the Internet Movie Database (IMDb) is represented as one of the top three options. Obviously, you'll want to keep this resource in mind for future searches.

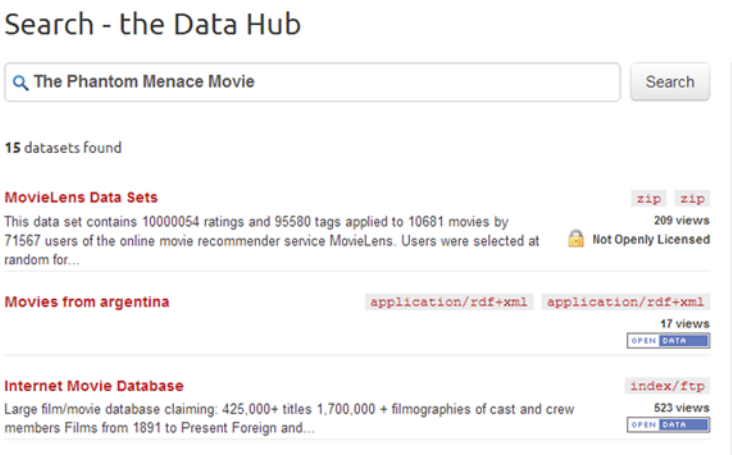

Figure 3.6 Results of searching for datasets on the Data Hub

3.4 Aggregating Linked Data

In section 3.1, we discussed manually finding Linked Data, and we examined tools to facilitate this process. In this section, we'll search published datasets that we're already aware of. You'll become familiar with extracting data from them and learn about the data they contain. We'll use these datasets later in the chapter to illustrate automating the extraction of data that we'll use in a sample application. For this example, we'll use *Star Wars*: *Episode I* as a sample movie and demonstrate how to find Linked Data about it. In particular, we'll be extracting data from two RDF databases, IMDb and ProductDB.

3.4.1 Aggregating some Linked Data from known datasets

Movies are well represented on the Web and in the Linked Open Data cloud. A good source for movie data is IMDb (http://www.imdb.com). Figure 3.7 is a screenshot of the IMDb listing for *Star Wars: Episode I*.

The IMDb URL associated with *Star Wars: Episode I* can be used as a search term in ProductDB (http://www.productdb.org). You can use this URL as a search key to obtain Linked Data about *Star Wars: Episode I* from ProductDB, an open source Linked Data database about products in general.

As its developer and maintainer Ian Davis says, "ProductDB aims to be the World's most comprehensive and open source of product data." His goal is "to create a page for every product in the world and to connect the underlying structured data together into one huge interlinked dataset." This data is compiled from open sources including ProductWiki (www.productwiki.com/), MusicBrainz (http://musicbrainz.org/), DBpedia (http://dbpedia.org/), Freebase (www.freebase.com/), and OpenLibrary (http://openlibrary.org/). It also is gathered by search engines' crawl sites that publish GoodRelations[1] RDFa or Open Graph protocol (http://opengraphprotocol.org/) data in

[1] GoodRelations, The Web Vocabulary for E-Commerce, http://www.heppnetz.de/projects/goodrelations/.

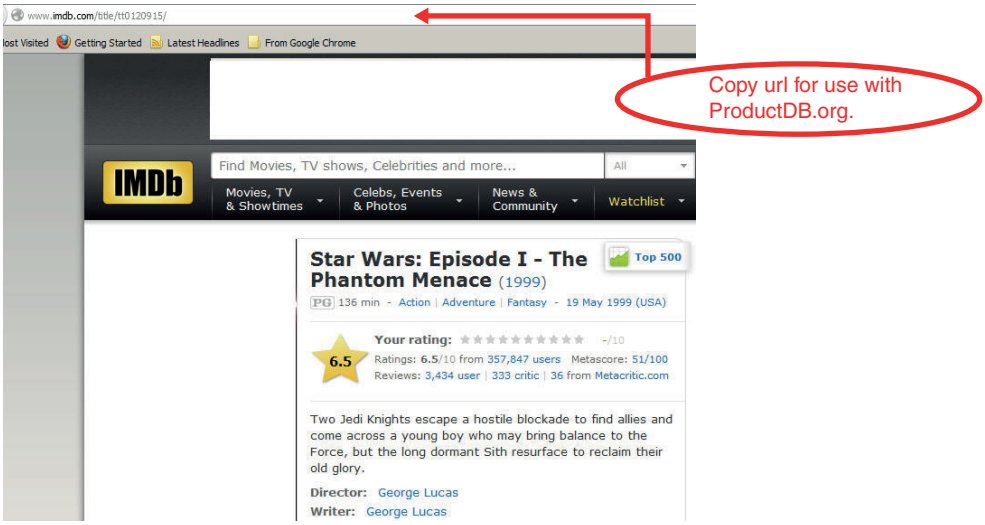

Figure 3.7 **IMDb entry for *Star Wars: Episode I – The Phantom Menace***

their pages; for example, BestBuy, IMDb, and Spotify, (http://www.spotify.com/). This aggregated data is gathered, combined, and analyzed to form linkages and correspondences between resources.

Here's the sequence of steps needed to manually obtain the ProductDB entry for *Star Wars: Episode I.*

Point your browser at http://www.productdb.org, as shown in figure 3.8.

ProductDB has numerous terms available via its pull-down menu that you can use to access the database, including an IMDb URL, as shown in figure 3.9.

We're using the selection IMDb URL. After you input a product code, enter the corresponding URL in the text box below. See figure 3.10 for the completed screen.

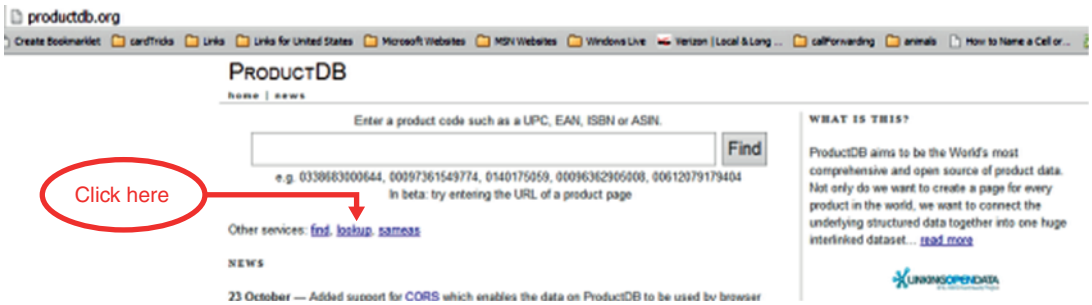

Figure 3.8 **Homepage for ProductDB**

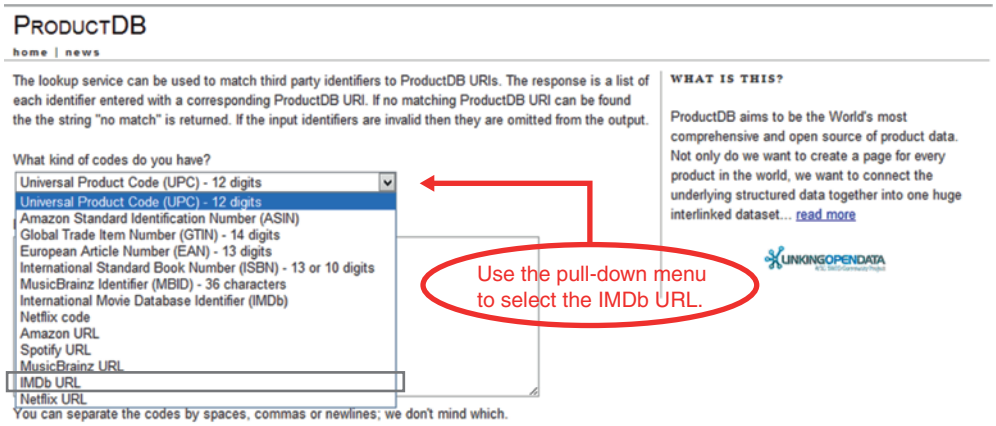

Figure 3.9 **Lookup page for ProductDB**

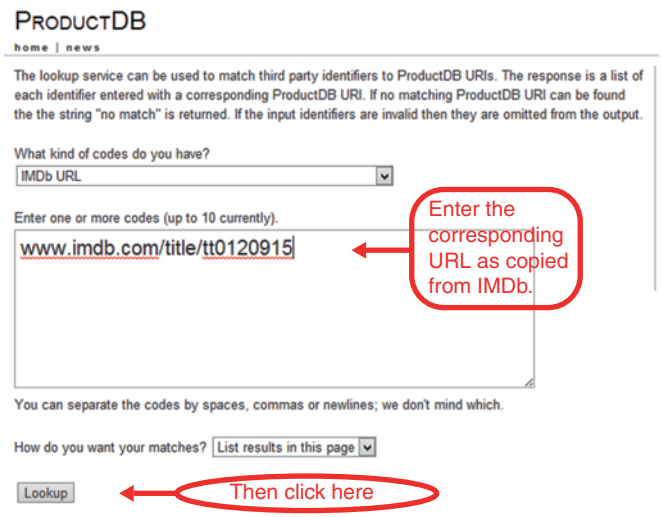

Figure 3.10 **ProductDB lookup page for *Star Wars: Episode I – The Phantom Menace***

ProductDB accesses its records and displays all matches, as shown in figure 3.11. In this case, there's a single match.

Click the result, and you'll obtain the product information page for the associated item. Note that ProductDB.org leads you to Netflix, Rotten Tomatoes, MOODb, Wikipedia, and other sites.

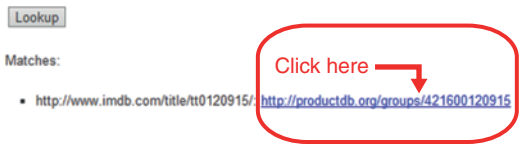

Figure 3.11 **ProductDB Matches (results) page for *Star Wars: Episode I – The Phantom Menace***

Figure 3.12 **ProductDB.org entry for *Star Wars: Episode I – The Phantom Menace***

You can obtain the raw data in Turtle format, as shown in figure 3.12, by clicking the Turtle link on the right-hand side. Those results are shown in listing 3.1.

Listing 3.1 ProductDB.org data for *Star Wars: Episode I* in Turtle format

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ns0: <http://dbpedialite.org/things/50793#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ns1: <http://www.rottentomatoes.com/m/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dct: <http://purl.org/dc/terms/> .
<http://productdb.org/groups/421600120915>
owl:sameAs   <http://data.linkedmdb.org/resource/film/69> ,
<http://rdf.freebase.com/ns/en.star_wars_episode_i_the_phantom_menace> ,
<http://dbpedia.org/resource/Star_Wars_Episode_I:_The_Phantom_Menace> ,

ns0:thing ;
foaf:isPrimaryTopicOf <http://www.imdb.com/title/tt0120915/> ,
<http://www.netflix.com/Movie/70003791> ,
ns1:star_wars_episode_i_the_phantom_menace ,
<http://en.wikipedia.org/wiki/Star_Wars_Episode_I:_The_Phantom_Menace> ,
<http://en.wikipedia.org/wiki/index.HTML?curid=50793> ,
<http://www.moodb.net/movie.asp?id=0000217> ;
rdfs:label "Star Wars: Episode I - The Phantom Menace" .
<http://productdb.org/gtin/00024543023913> dct:isVersionOf  <http://
     productdb.org/groups/421600120915> .
<http://productdb.org/gtin/00010232008374> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
<http://productdb.org/gtin/00391772364227> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
<http://productdb.org/gtin/00321337023526> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
```

Note the use of owl:sameAs to provide additional links.

Note the use of foaf:isPrimaryTopicOf to provide additional links.

```
<http://productdb.org/gtin/00039036007375> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
<http://productdb.org/gtin/00024543023937> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
<http://productdb.org/gtin/00712626010272> dct:isVersionOf <http://
     productdb.org/groups/421600120915> .
```

By focusing on the underlying RDFa data compiled by ProductDB, you should see the relationship between *Star Wars: Episode I—The Phantom Menace* and content available from other sources like Netflix (http://www.netflix.com/Movie/70003791), Rotten Tomatoes (http://www.rottentomatoes.com/m/star_wars_episode_i_the_phantom_menace), and MOODb (http://www.moodb.net/movie.asp?id=0000217), among others. This shows how following the data links establishes unanticipated connections. George Lucas would be unlikely to encourage viewers to seek out the description at Rotten Tomatoes. Here episode I is described as "Lucas needs to improve on the plot and character development, but there's plenty of eye candy to behold." Lucas may not approve of this information being tied to his movie, but such data is easily gatherable via Linked Data by referencing the embedded URIs.

3.4.2 Getting Linked Data and RDF from web pages using browser plug-ins

When viewing a web page, the underlying RDFa is generally hidden to the viewer but could be useful when extracted. As an example, the IMDb page contains RDFa, including an image file associated with the movie. One approach to discovering the presence of RDFa data is to install a browser plug-in. We've used a Firefox plug-in called RDFa Developer, but there are many others. The underlying RDFa as discovered by RDFa Developer is displayed in figure 3.13. But you can also automate the finding, extraction, and utilization of RDFa data. We'll demonstrate this technique later in the chapter.

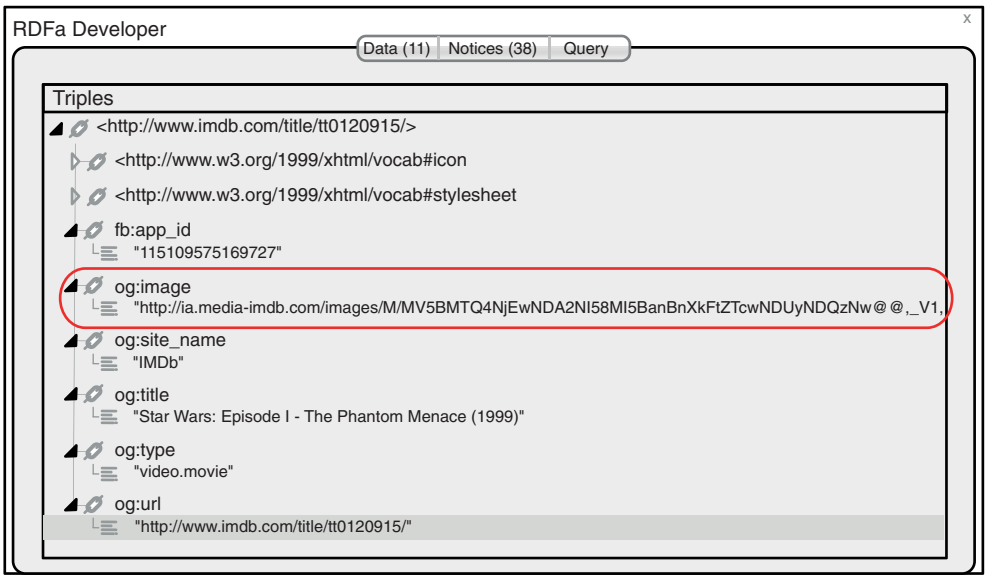

Figure 3.13 **IMDb entry for *Star Wars: Episode I – The Phantom Menace* with discovered RDFa displayed**

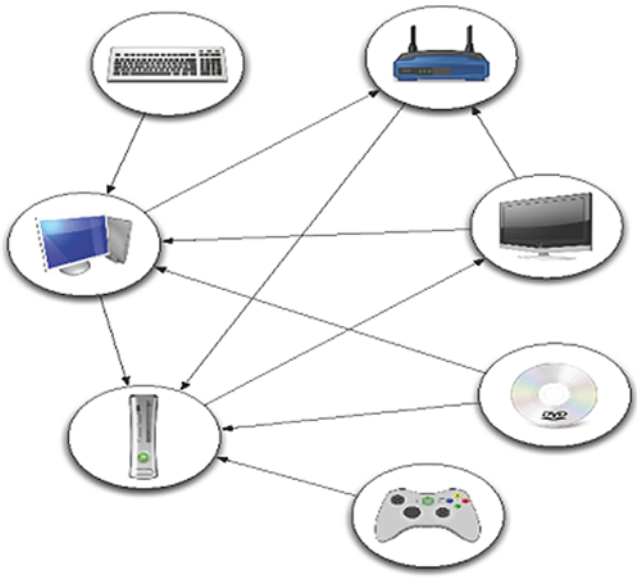

Figure 3.14 Illustration of linked relationships of common products

Jay Myers of Best Buy reports that as many as 100 different criteria could affect the purchase of one product.[2] He expects that the use of semantic data can enhance the Best Buy site, improve the visibility of more than 85% of the products, and help consumers identify more appropriate products. Figure 3.14 demonstrates the semantic relationships between products. Myers expects that "RDFa can ultimately create rich relationships between products, which will in turn 'create a deeper visibility to additional products' when a customer is shopping."[3]

Best Buy's site is a great source of RDFa; see figure 3.15. Myers, lead web development engineer, is a proponent of Linked Data. At the 2010 Semantic Technology Conference, he reported that Best Buy had a 30% increase in search traffic after incorporating RDFa data in its web pages. Myers also reported that the rank of the pages that incorporated Good Relations (a Semantic Web vocabulary) and RDFa rose significantly in Google search results. Myers intends to continue to explore other uses of Linked Data to help consumers discover more relevant products that better meet their needs.

Best Buy's RDFa is linked from ProductDB.org. In addition, you can use SameAs.org to find more data by following links from Best Buy's URIs to other datasets. SameAs.org has 122 related links for *Star Wars: Episode I* given the dbpedia.org URL we retrieved from ProductDB.org. You can use the results from SameAs.org, shown in figure 3.16, to manually discover additional related data.

---

[2] "Better Retailing through Linked Data. Opportunities, perspectives, and vision on Linked Data in retail," http://www.slideshare.net/jaymmyers/better-retailing-through-linked-data.

[3] Richard MacManus, interviewing Jay Myers, "How Best Buy Is Using the Semantic Web," June 30, 2010, http://www.readwriteWeb.com/archives/how_best_buy_is_using_the_semantic_Web.php.

RDFa Developer

Data (19) | Notices (15) | Query

Triples	Number of children
<http://www.bestbuy.com/site/Star+Wars%3A+Episode+I+-+The+Phant...?id=30545&skuid=4244785&s	9
<http://www.w3.org/1999/xhtml/vocab#stylesheet	11
fb:app_id "125188000891129"	1
og:description "Star Wars: Episode I - The Phantom Menace - Widescreen - DVD"	1
og:image "http://images.bestbuy.com:80/BestBuy_US/images/products/4244/4244785s.jpg"	1
og:site_name "Best Buy"	1
og:title "Star Wars: Episode I - The Phantom Menace - Widescreen - DVD"	1
og:type "product"	1
og:upc "024543023920"	1
og:url "http://www.bestbuy.com/site/Star+Wars%3A+Episode+I+-+The+Phant...?id=30545&skuid=4244785&st=star%	1

Figure 3.15 **Best Buy's RDFa for *Star Wars: Episode I***

You can use the outcome of SameAs.org to help identify a *canonical URL* for a given item. A canonical URL is the best URL among available choices. For example, you may consider www.example.com, example.com/, www.example.com/index.HTML, and example.com/home.asp interchangeable. However similar, each of these URLs may return different content. A canonical URL is the preferred URL and often refers to an item's homepage.

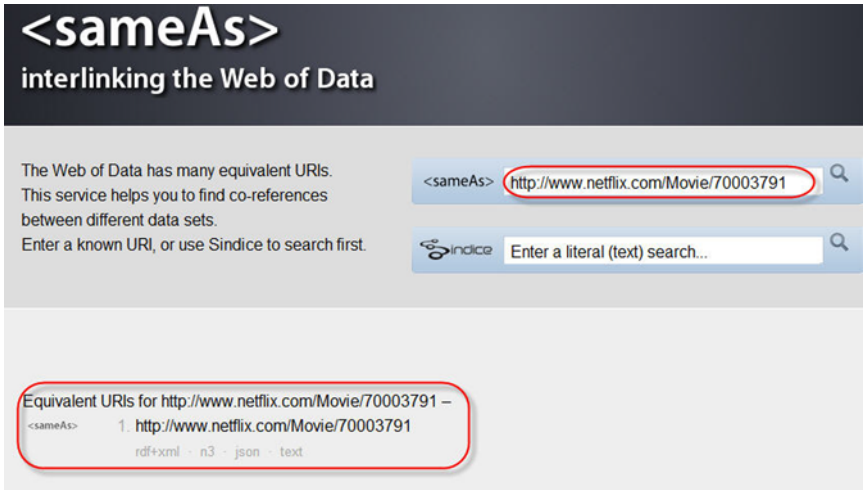

Figure 3.16 **SameAs.org results for *Star Wars: Episode I***

3.5 Crawling the Linked Data Web and aggregating data

In previous sections we've illustrated how to manually follow your nose and use tools to find data on the Web. As a developer, you may be interested in consuming existing data by combining extracted data and using it in another application. For such purposes, we'd prefer to automate the processing of the aggregation of data. In this section, we'll develop an application that will illustrate such automation. This application will show you how to use the Python scripting language, RDFLib, and html5lib to access the RDFa data available from Best Buy for a sample product, the Darth Vader Alarm Clock Radio. This application also accesses the data stored for the Darth Vader Alarm Clock Radio from the ProductDB database. All of this RDF data is gathered and displayed in a three-column table of subject, predicate, and object.

3.5.1 Using Python to crawl the Linked Data Web

We've chosen the Python scripting language because it supports gathering and using aggregated RDF information. The example script (listing 3.2) will gather RDFa Linked Data about the Darth Vader Alarm Clock Radio, shown in figure 3.17, from Best Buy web pages and use that data to obtain more linked information from ProductDB.

In the case of the Darth Vader Alarm Clock Radio, the RDFa Developer plug-in identified a triple that contained the UPC for this product. This UPC is used to search ProductDB. The output (output.html) contains the HTML for a web page that displays these discovered triples as a table of TTL statement components, as shown in figure 3.18.

NOTE The execution of this script requires that a Python interpreter be installed together with RDFlib and html5lib.

You can download a Python interpreter from Python.org (http://python.org). Follow the guidelines at http://wiki.python.org/moin/BeginnersGuide/Download to select the installation package to meet your needs.

Figure 3.17 Best Buy's Web page for Darth Vader Alarm Clock Radio

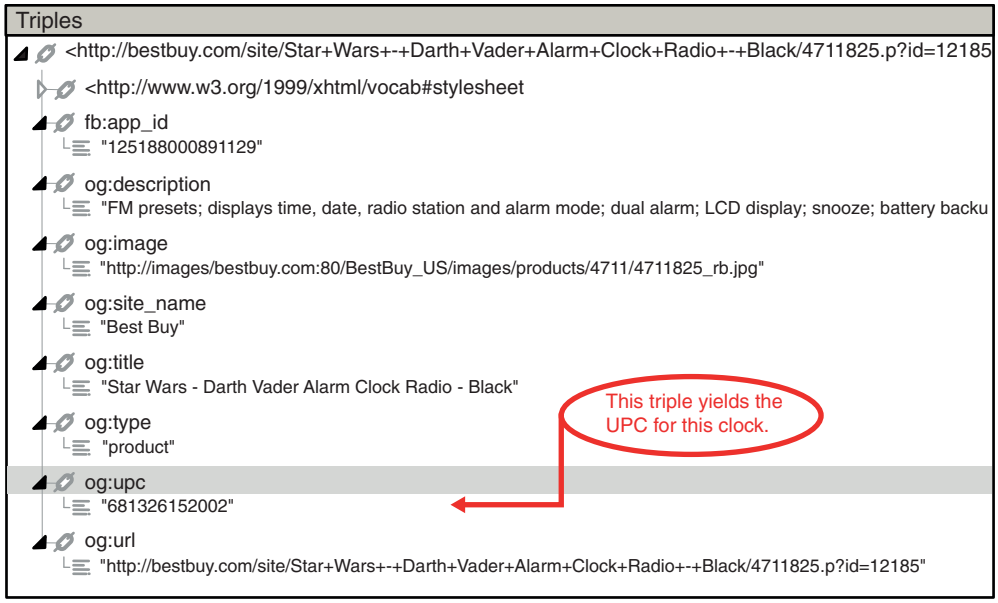

Figure 3.18 **RDFa data from Best Buy's web page**

To run these scripts successfully, the libraries RDFLib and html5lib *must* be installed. You can obtain RDFLib from https://github.com/RDFLib/rdflib. This library will allow you to successfully pull down the Turtle file from ProductDB. Best Buy utilizes HTML5, which requires html5lib. This library is available from http://code.google.com/p/html5lib/downloads/detail?name=html5lib-0.95.tar.gz&can=2&q=. These downloads should be stored in [PYTHON HOME]/lib. Installing these libraries should take no more than a few minutes and will allow you to run the Python script outlined in the following listing.

Listing 3.2 A Python script for aggregating RDF data for HTML display

```
#! /usr/bin/python

import rdflib
import html5lib

output = open("output.HTML", "w")

productDBGraph = rdflib.Graph()
productDBResult =
➥   productDBGraph.parse('http://productdb.org/gtin/00681326152002.ttl',
    format='turtle')

bestBuyGraph = rdflib.Graph()
bestBuyResult =
```

Establish the graph from ProductDB using RDFLib to parse it.

Establish the graph from Best Buy using RDFLib and html5lib to parse it.

```
➥   bestBuyGraph.parse('http://purl.org/net/BestBuyDarthVaderClock',
     format='rdfa')
print >>output, """<HTML>
<head>
    <title>Product Information</title>
</head>
<body>
<table border="1">"""

print >>output,
➥   "<tr><th>Subject</th><th>Predicate</th><th>Object</th></tr>"

for sub, pred, obj in productDBGraph:
    print >>output, "<tr><td>%s</td><td>%s</td><td>%s</td></tr>" % (sub,
➥   pred, obj)

for sub, pred, obj in bestBuyGraph:
    print >>output, "<tr><td>%s</td><td>%s</td><td>%s</td></tr>" % (sub,
➥   pred, obj)

print >>output, """</table>
</body></HTML>"""
```

A PURL was used here instead of a very long Best Buy URL.[4] This is a redirect so that the URL is more readable.

Begin printing HTML page that will contain RDF.

Print subject, predicate, and object of all triples pulled from both ProductDB and Best Buy and place them into an HTML table.

3.5.2 Creating HTML output from your aggregated RDF

In addition to aggregating RDF data from both Best Buy and ProductDB, the Python script in listing 3.2 creates an HTML output file named output.html. We could have retained the aggregated data in a file of triples, but we chose to display it in HTML format so that you could better appreciate the findings. This file should be opened in a browser. The first screen of the HTML file produced should contain the content shown in table 3.1. Each row in the table represents a set of Turtle triples.

Table 3.1 Representative sample output for listing 3.2

Subject	Predicate	Object
http://productdb.org/gtin/00681326152002	http://purl.org/goodrelations/v1#hasGTIN-14	00681326152002
http://productdb.org/gtin/00681326152002	http://purl.org/goodrelations/v1#hasManufacturer	http://productdb.org/brands/star-wars
http://productdb.org/gtin/00681326152002	http://schema.org/manufacturer	http://productdb.org/brands/star-wars
http://productdb.org/gtin/00681326152002	http://schema.org/image	http://images.bestbuy.com/BestBuy_US/images/products/4711/4711825_rc.jpg

[4] Best Buy URL referenced in listing 3.2, http://www.bestbuy.com/site/Star+Wars+-+Darth+Vader+Alarm+Clock+Radio+- +Black/4711825.p?id=1218515225401&skuId=4711825&st=Star%20Wars&cp=1&lp=7

Table 3.1 Representative sample output for listing 3.2 *(continued)*

Subject	Predicate	Object
http://productdb.org/gtin/00681326152002	http://schema.org/url	http://www.bestbuy.com/site/Star+Wars+-+Darth+Vader+Alarm+Clock+Radio+-+Black/4711825.p?id=1218515225401&skuId=4711825&cmp=RMX&ky=2nBwHqIqwH8HeGDnJH2Cia1DoDKws99jo
http://productdb.org/gtin/00681326152002	http://schema.org/productID	0681326152002
http://productdb.org/gtin/00681326152002	http://schema.org/name	Star Wars - Darth Vader Alarm Clock Radio – Black
http://productdb.org/gtin/00681326152002	http://open.vocab.org/terms/category	http://productdb.org/classifications/bestbuy/pcmcat263000050000
http://productdb.org/gtin/00681326152002	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Product
http://productdb.org/gtin/00681326152002	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://purl.org/goodrelations/v1#ProductOrServiceModel

This application aggregated data and demonstrated how Python can be used together with RDFLib and html5lib to extract RDF data from Best Buy and ProductDB. We then formatted the aggregated data as HTML so that you can examine the output in a browser. In chapters 6, 7, and 9, we'll illustrate how this aggregated data can be retained or piped and reused in other applications.

3.6 Summary

The purpose of this chapter was to expose you to the multiple facets of consuming Linked Data from the Web. To this end, we explored what it means to think the Web way. We explored finding Linked Data on the Web by manually following your nose and facilitating the process through the use of special tools. Finally, we illustrated how you could use Python, RDFLib, and html5lib to develop programs that retrieve Linked Data from one source and use those results to retrieve additional data from a different data source. In subsequent chapters, we'll emphasize techniques for developing and publishing your own Linked Data and enhanced search techniques for aggregating such data.

WEB DEVELOPMENT/DATABASES

Linked Data

Wood • Zaidman • Ruth • Hausenblas

The current Web is mostly a collection of linked documents useful for human consumption. The evolving Web includes data collections that may be identified and linked so that they can be consumed by automated processes. The W3C approach to this is Linked Data and it is already used by Google, Facebook, IBM, Oracle, and government agencies worldwide.

Linked Data presents practical techniques for using Linked Data on the Web via familiar tools like JavaScript and Python. You'll work step-by-step through examples of increasing complexity as you explore foundational concepts such as HTTP URIs, the Resource Description Framework (RDF), and the SPARQL query language. Then you'll use various Linked Data document formats to create powerful Web applications and mashups.

What's Inside

- Finding and consuming Linked Data
- Using Linked Data in your applications
- Building Linked Data applications using standard Web techniques

Written to be immediately useful to Web developers, this book requires no previous exposure to Linked Data or Semantic Web technologies.

David Wood is co-chair of the W3C's RDF Working Group. **Marsha Zaidman** served as CS chair at University of Mary Washington. **Luke Ruth** is a Linked Data developer on the Callimachus Project. **Michael Hausenblas** led the Linked Data Research Centre.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/LinkedData

Free eBook
SEE INSERT

"A friendly introduction to the use and publication of structured data on the WWW."
—From the Foreword by Tim Berners-Lee, Director of W3C

"A practical guide for integrating and publishing structured data on the Web."
—Cristofer Weber, NeoGrid

"Takes a complex academic subject and makes it clear and relevant."
—Mike Westaway, AstraZeneca

"Highly recommended to all explorers of the Semantic Web."
—Rob Crowther, author of *Hello! HTML5 & CSS3*

ISBN 13: 978-1-617290-39-8
ISBN 10: 1-617290-39-4
54999
9 781617 290398

MANNING $49.99 / Can $52.99 [INCLUDING eBOOK]