The RDF Book Mashup: From Web APIs to a Web of Data

Christian Bizer, Richard Cyganiak, and Tobias Gauß

Freie Universität Berlin
chris@bizer.de, richard@cyganiak.de, tobias.gauss@web.de

Abstract. The RDF Book Mashup demonstrates how Web 2.0 data sources like Amazon, Google and Yahoo can be integrated into the Semantic Web. Following the principles of linked data, the RDF Book Mashup makes information about books, their authors, reviews, and online bookstores available on the Semantic Web. This information can be used by RDF browsers and crawlers, and other publishers of Semantic Web data can link to it.

1 Introduction

Major web data sources like Google, Yahoo, Amazon and eBay have started to make their data accessible to third parties through web APIs. This has inspired many interesting "mashups" which combine data from multiple sources and present it through new user interfaces. Web APIs usually consist of REST or SOAP web services that deliver XML or JSON. Scripting languages play an important role in mashup development as they often serve as the "glue" that connects data sources and UI components.

The goal of the RDF Book Mashup is to show how data from web APIs can be integrated into the Semantic Web. This has two benefits. First, it adds useful data to the Semantic Web. Second, it addresses several limitations of traditional web APIs:

- They do not work with clients that have not been designed with the specific API in mind.
- Their content cannot be accessed by search engines and other generic web agents.
- Each mashup only allows access to data from a limited number of sources chosen by the developer.

In contrast, information on the Semantic Web can be used by by generic clients, including RDF browsers, RDF search engines, and web query agents.

2 Linked Data

Based on Tim Berners-Lee's *Linked Data* prinicples [1] [4], we identify a lowest common denominator for Semantic Web interoperability:

1. All items of interest, such as information resources, real-world objects, and vocabulary terms are identified by URI references [3].
2. URI references are dereferenceable; an application can look up a URI over the HTTP protocol and retrieve an RDF description of the identified item.
3. Descriptions are provided using the RDF/XML syntax.
4. Every RDF triple is conceived as a hyperlink that links to related information from the same or a different source and can be followed by Semantic Web agents.

These principles are sufficient to create a *Web of Data* in which anyone can publish information, link to existing information, follow links to related information, and consume and aggregate information without necessarily having to fully understand its schema nor to learn how to use a proprietary Web API.

3 How to Wrap Web APIs

The RDF Book Mashup applies these principles to web APIs that supply data about books. It assigns URIs to books, authors, reviews, online bookstores and purchase offers. When a URI is dereferenced, the mashup queries the Amazon API [1] for information about the book, and the Google Base data API[2] for purchase offers from different bookstores. An RDF description is assembled from the query results and returned to the client in RDF/XML syntax. The description uses a mix of RDFS vocabularies, including Dublin Core[3] and FOAF[4]. We introduced new vocabulary terms where we could not find existing terms in well-established vacabularies.

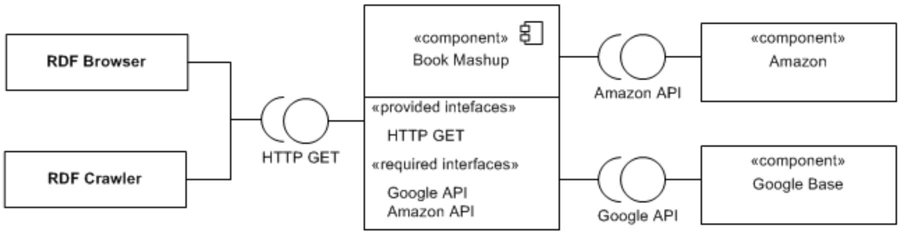

Fig. 1. Architecture of the RDF Book Mashup

Figure 1 gives an overview about the architecture of the RDF Book Mashup. Figure 2 shows the description of the book identified by this URI:

[1] http://aws.amazon.com/

[2] http://code.google.com/apis/base/

[3] http://dublincore.org/documents/dces/

[4] http://xmlns.com/foaf/0.1/

```
<http://www4.wiwiss.fu-berlin.de/bookmashup/books/006251587X>
```

Figure 3 shows the description of its author, who is identified by

```
<http://www4.wiwiss.fu-berlin.de/bookmashup/persons/Tim+Berners-Lee>
```

Both URIs are linked to each other by a `dc:creator` statement. The descriptions contain links to further URIs for reviews and purchase offers, which in turn can be dereferenced to yield RDF descriptions.

```
<scom:Book rdf:about="/bookmashup/books/006251587X">
   <dc:title>Weaving the Web</dc:title>
   <dc:creator rdf:resource="/bookmashup/persons/Tim+Berners-Lee"/>
   <dc:date>2000-11-01</dc:date>
   <dc:format>Paperback</dc:format>
   <dc:identifier rdf:resource="urn:ISBN:006251587X"/>
   <dc:publisher>Collins</dc:publisher>
   <foaf:thumbnail rdf:resource=
         "http://ec2.images-amazon.com/images/P/006251587X.jpg"/>
   <rev:hasReview rdf:resource=
         "/bookmashup/reviews/006251587X_EditorialReview1"/>
   <rev:hasReview rdf:resource=
         "/bookmashup/reviews/006251587X_EditorialReview2"/>
   <scom:hasOffer rdf:resource=
         "/bookmashup/offers/006251587XamazonOffer"/>
   <scom:hasOffer rdf:resource=
         "/bookmashup/offers/006251587XgoogleOffer10871997653977"/>
</scom:Book>
<rdf:Description rdf:about="/bookmashup/doc/books/006251587X">
   <dc:license rdf:resource="http://www.amazon.com/AWS-License"/>
   <dc:license rdf:resource="http://www.google.com/terms_of_service.html"/>
</rdf:Description>
```

Fig. 2. RDF description of a book (edited)

4 Implementation

The RDF Book Mashup is implemented in about 300 lines of PHP code. Whenever the script gets a lookup call for a URI, it decodes the ISBN number or author name from the URI and uses them to query the origin data sources. The resulting XML responses are parsed using PHP's SimpleXML functions and turned into an RDF model which is serialized to RDF/XML using the RDF API for PHP (RAP)[5].

[5] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/rdfapi/`

```
<foaf:Person rdf:about="/bookmashup/persons/Tim+Berners-Lee">
   <owl:sameAs rdf:resource=
"http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007"/>
   <foaf:name>Tim Berners-Lee</foaf:name>
</foaf:Person>
<scom:Book rdf:about="/bookmashup/books/006251587X">
   <dc:creator rdf:resource="/bookmashup/persons/Tim+Berners-Lee"/>
   <rdfs:label>Weaving the Web</rdfs:label>
</scom:Book>
<scom:Book rdf:about="/bookmashup/books/8432310409">
   <dc:creator rdf:resource="/bookmashup/persons/Tim+Berners-Lee"/>
   <rdfs:label>Tejiendo La Red - El Inventor del WWW Nos Descubre</rdfs:label>
</scom:Book>
```

Fig. 3. RDF description of an author (edited)

The services use different formats for queries and responses: The Amazon Web Service is a REST-based API that encodes requests in a URL string and answers simple XML. The Google Base data API uses a proprietary query language and delivers search results as Atom feeds. Both have in common that requests are encoded into a simple URL string, and responses are delivered as XML, which means that processing can be done easily with built-in PHP functions.

To comply with web architecture[6], the mashup uses 303 redirects to point clients from the URI for a book or person (e.g. `<books/006251587X>`) to an RDF document describing the item (e.g. `<doc/books/006251587X>`) [3].

The generated RDF contains metadata about the document itself, including links to the license terms under which the data is published.

5 RDF Links to and from other Data Sources

RDF is the obvious technology to interlink data from various data sources. In the following, we describe how the RDF Book Mashup is interlinked with various other Semantic Web data sources.

5.1 Outgoing RDF Links

Another publicly available source of bibliographic data is the DBLP database[7]. It contains journal articles and conference papers. In a previous project, we have published this database on the Semantic Web[8] using D2R Server [2]. The RDF Book Mashup automatically generates outgoing `owl:sameAs` links from

[6] http://norman.walsh.name/2005/06/19/httpRange-14

[7] http://dblp.uni-trier.de/

[8] http://www4.wiwiss.fu-berlin.de/dblp/

book authors to paper authors in the DBLP database. An RDF crawler that aggregates RDF statements from both sources can conclude that both URIs refer to the same person and thus allow queries over merged data from both sources.

The links are generated by asking the DBLP SPARQL endpoint for URIs identifying book authors. If the query for a `foaf:Person` with a specific name returns exactly one result, we assume (sometimes incorrectly) that they are the same person, and add the `owl:sameAs` link. An example can be seen in Figure 3.

5.2 Incoming RDF Links

To integrate the book data into the Semantic Web, we also need incoming links that enable Semantic Web clients to discover the data.

Another Semantic Web data source that contains information about books is DBpedia[9], a project that publishes RDF data extracted from Wikipedia. DBpedia uses the ISBN numbers that are found in Wikipedia articles to auto-generate links to the RDF Book Mashup. The DBpedia dataset currently contains about 9000 `owl:sameAs` links pointing at RDF Book Mashup URIs.

A second Semantic Web data source that auto-generates RDF links to RDF Book Mashup URIs is the Semantic Web server that publishes the DBLP bibliography. The server includes an RDF link to the RDF Book Mashup into the RDF description of each bibliography item that has a ISBN number.

The RDF data that is generated by the RDF Book Mashup can also be used to augment web pages. An example of a web page that uses Book Mashup data is the Semantic Web book list [10] which is maintained within the W3C ESW Wiki. This list contains around 40 manually-set links to RDF Book Mashup data describing books from the list.

A further source of incoming links are the FOAF profiles of book authors. Tim Berners-Lee and Ivan Herman are among the people who have such links in their profiles [11] [12]. The RDF Book Mashup's website has instructions that tell book authors how to set `dc:creator` links from their FOAF profile to their books.

6 Client Applications

Linked data can be consumed by a different kinds of clients on behalf of a user. Some scenarios are:

Browsing: Semantic Web browsers like Tabulator [4], Disco[13], and the OpenLink RDF Browser[14] can be used to explore and visualize Book Mashup data, and to surf into and out of the dataset.

[9] `http://dbpedia.org/`

[10] `http://esw.w3.org/topic/SwBooks`

[11] `http://www.w3.org/People/Berners-Lee/card`

[12] `http://www.ivan-herman.net/foaf.rdf`

[13] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/disco/`

[14] `http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html`

Searching and querying: RDF crawlers can harvest the data for search engines such as SWSE[15] and Swoogle[16]. Query agents like the Semantic Web Client Library[17] can crawl the data on-the-fly to answer queries.

Augmenting web pages: Web page authors can add `<link rel="alternate">` tags[18] to web pages about books. This allows web surfers to use enhanced browsers like Piggy Bank[19] to collect book descriptions from the Web.

7 Conclusion

We have shown how to integrate information from web APIs with the Semantic Web, making them part of the Web of Data, and making them accessible to generic Semantic Web clients like RDF browsers, RDF crawlers and query agents. This has been done by wrapping several APIs into an interface that applies the Linked Data principles. The wrapper is implemented as a fairly simple PHP script. We hope that this work will inspire others to integrate other interesting data sources into the growing Semantic Web.

References

1. Tim Berners-Lee. Linked data, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`.
2. Christian Bizer and Richard Cyganiak. D2r server - publishing releational databases on the semantic web. In *Poster at the 5th International Semantic Web Conference*, 2006.
3. Leo Sauermann, Richard Cyganiak, and Max Völkel. Cool URIs for the Semantic Web. Technical report, 2006.
4. Tim Berners-Lee et al. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006. `http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf`.

[15] `http://swse.deri.org/`

[16] `http://swoogle.umbc.edu/`

[17] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/semwebclient/`

[18] `http://www.w3.org/TR/rdf-syntax-grammar/\#section-rdf-in-HTML`

[19] `http://simile.mit.edu/piggy-bank/`