Exploiting Linked Data For Building Web Applications

Michael Hausenblas

DERI, National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
michael.hausenblas@deri.org

Abstract. Semantic Web technologies are around now for a while, already. However, in the development of real-world Web applications these technologies have considerable little impact to date. With linked data this situation has changed dramatically in the past couple of months. This article shows how linked datasets can be exploited to build rich Web applications with little effort.

Many Web-developers nowadays use APIs such as offered by Google, Facebook, etc. to build and enhance their Web application. Due to the many ways these APIs are typically designed (proprietary XML formats, JSON, etc.) the development of applications based on Web 2.0 mashups usually is burdensome and does not scale well. Beside the effort to learn new interfaces over and over again, the so created data is locked in the respective platform. The Web of Data—also known as the Semantic Web—has promised for a long time to resolve these issues. To date, however, only partially solutions to real-world problems exist, many of them addressing rather toy datasets. With a recent initiative, the "Linked Data" community project, the situation has changed dramatically: based on simple Semantic Web technologies such as RDF[1] and URIs along with a set of so called "linked data" principles, a number of data sources such as Wikipedia have been made available on the Web of Data. Developers can now readily benefit from the linked datasets based on a common data model [1]. This article shows how to exploit the available linked datasets in order to build rich Web applications with little effort.

Example Usage of Linked Data

Before we tackle the technical challenges of linked data we have a look at some exemplary usages of linked datasets. For example, Faviki[2], a social bookmarking tool allows to tag Web-pages with "semantic tags" stemming from Wikipedia. Here, the main purpose of Web of Data technologies and data is providing unambiguous space for identifying concepts. The tool, shown in Fig. 1, uses URIs

[1] http://www.w3.org/TR/rdf-concepts/
[2] http://www.faviki.com/

from DBpedia (the interlinked version of Wikipedia in RDF) for tagging; in our example `http://dbpedia.org/resource/Internet` is used as a tag—anyone interested in this term can dereference this URI and is able to obtain further information about it.

Fig. 1. Screenshot of Faviki, a social bookmarking tool utilising Wikipedia terms for semantic tagging.

Further, DBpedia mobile [2], depicted in Fig. 2, is an interesting application for mobile environments. Basically it is a location-centric DBpedia client application for mobile devices, that is—based on the GPS signal of a mobile—able to render a map indicating nearby locations from the DBpedia dataset.

Fig. 2. Screen-shot of the DBpedia Mobile's map view.

The BBC Music beta site as shown in Fig. 3 is a Web-site in HTML primarily targeting human users. However, agents operating on the Web of Data consume RDF. In the following we will show how to use the "Swiss army knife" `curl` to obtain an RDF "view" on the data: by using content negotiation (that is, setting the accept-field in the HTTP header to RDF/XML), such as

```
curl -H "Accept: application/rdf+xml" http://www.bbc.co.uk/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287
```

the server will respond with an RDF/XML representation of the resource:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf      = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs     = "http://www.w3.org/2000/01/rdf-schema#"
         xmlns:foaf     = "http://xmlns.com/foaf/0.1/"
         xmlns:mo       = "http://purl.org/ontology/mo/"
         xmlns:owl      = "http://www.w3.org/2002/07/owl#">
<mo:SoloMusicArtist rdf:about="/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287#artist">
  <foaf:name>Madonna</foaf:name>
  <owl:sameAs rdf:resource="http://dbpedia.org/resource/Madonna_(singer)" />
  <mo:image rdf:resource="/music/images/artists/7col_in/79239441-bfd5-4981-a70c-55c3f15c1287.jpg" />
  <foaf:page rdf:resource="/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287.html" />
  <mo:musicbrainz rdf:resource="http://musicbrainz.org/artist/79239441-bfd5-4981-a70c-55c3f15c1287.html" />
  <mo:homepage rdf:resource="http://www.madonna.com/" />
  <mo:fanpage rdf:resource="http://www.mad-eyes.net" />
  <mo:fanpage rdf:resource="http://www.allaboutmadonna.com/" />
  ...
</mo:SoloMusicArtist>
</rdf:RDF>
```

We identify various vocabularies in this RDF graph. For example widely deployed ones such as FOAF (`foaf:`), but also specialised ones, e.g. the music ontology (`mo:`) are used to represent the information about the artist Madonna. Further, we find the interlinking to DBpedia (`http://dbpedia.org/resource/Madonna_(singer)`). One may now perform structured queries on top of this RDF representation using SPARQL. For example, to obtain the fan-pages for the artist Madonna, one may use the following query:

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?fanpage
FROM <http://triplr.org/rdf/www.bbc.co.uk/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287>
WHERE {
       ?artist a mo:SoloMusicArtist ;
               owl:sameAs <http://dbpedia.org/resource/Madonna_(singer)> ;
               mo:fanpage ?fanpage .
}
```

Sidebar 1: BBC Music beta site—Where is the RDF?

Only recently, BBC has announced[3] the release of their new Music beta site[4] built around the Musicbrainz[5] metadata and identifiers. Music metadata such as related artists are pulled from Musicbrainz, for those links pointing to Wikipedia, the introductory text for each artist's biography is fetched from there. An example artist's page—Madonna[6]—is depicted in (Fig. 3); see also the information in the sidebar 1.

[3] `http://www.bbc.co.uk/blogs/radiolabs/2008/07/music_beta_and_linked_data.shtml`

[4] `http://www.bbc.co.uk/music/beta/`

[5] `http://musicbrainz.org/`

[6] `http://www.bbc.co.uk/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287`

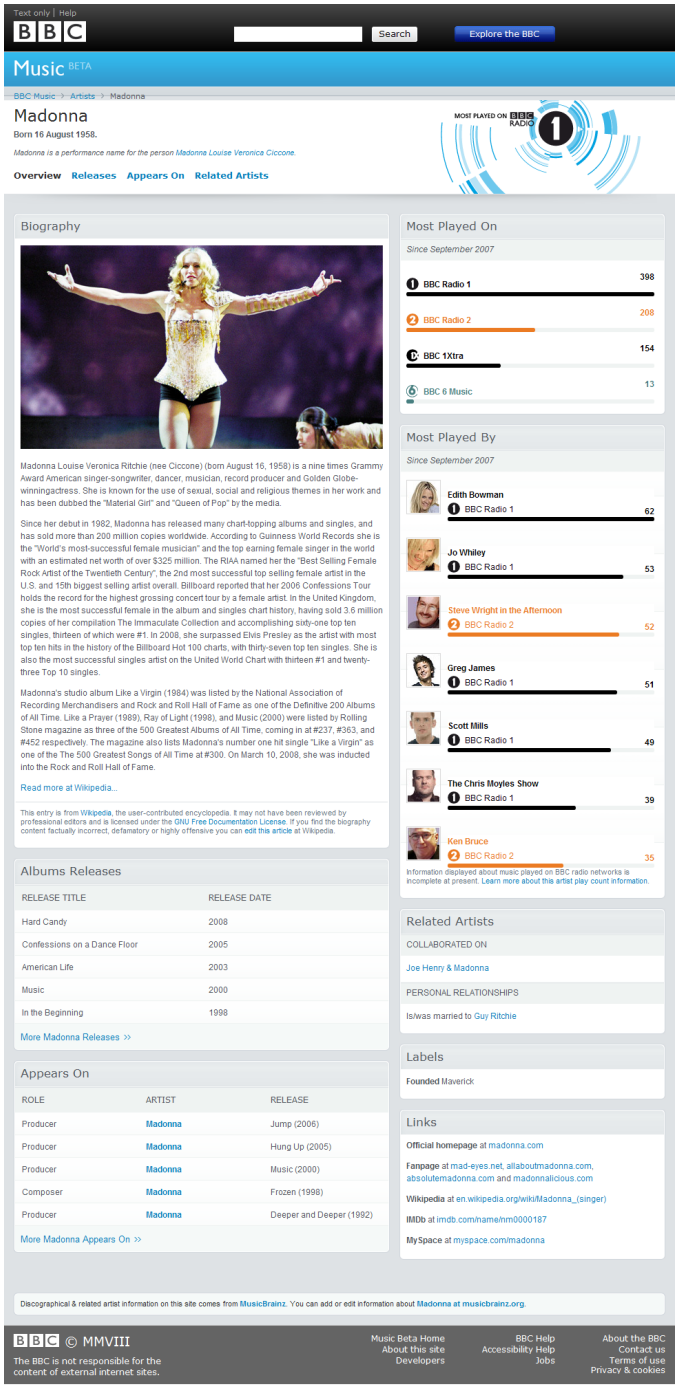

Fig. 3. Screen-shot of BBC Music BETA site showing artist's information pulled from Musicbrainz and Wikipedia.

But how has this been realised? What are the design principles of it? We now come back to the rather technical aspects of linked data(sets), the so called linked data principles and the technologies that enable their implementation.

Linked Data Principles

The basic idea of linked data has first been outlined by Sir Tim Berners-Lee in 2006. In his seminal design note[7] he described the four linked data principles as follows:

1. All items should be identified using *URIs*;
2. All URIs should be *dereferenceable*, that is, using HTTP URIs allows looking up the an item identified through the URI;
3. When looking up an URI—that is, an RDF property is interpreted as a hyperlink—it leads to more data, which is usually referred to as the follow-your-nose principle;
4. Links to other URIs should be included in order to enable the discovery of more data.

In contrast to the full-fledged Semantic Web vision, linked data is mainly about publishing structured data in RDF using URIs rather than focusing on the ontological level or inferencing. This simplification—just as the Web simplified the established academic approaches of Hypertext systems—lowers the entry barrier for data provider, hence fosters a wide-spread adoption [3–5].

The Linking Open Data (LOD) project, an open, collaborative effort carried out in the realm of the W3C SWEO[8] Community Projects initiative aimed at bootstrapping the Web of Data by publishing datasets in RDF on the Web and creating large numbers of links between these datasets. The project started out in early 2007 with a relatively modest number of datasets and participants and has grown since both in terms of depth, impact and contributors.

Currently, the project includes over 50 different datasets (Fig. 4, the LOD cloud, by courtesy of Richard Cyganiak[9]) with over two billion RDF triples and three million (semantic) links at the time of writing—representing a steadily growing, open implementation of the linked data principles.

When having a closer look at widely deployed vocabularies [6, 7] in the linked datasets one can group the "semantic link types" for example into:

– person-related links, e.g. `foaf:knows` from FOAF[10];
– spatial link types such as `foaf:based_near` or `geo:lat` of the basic geo (WGS84 lat/long) vocabulary[11];

[7] `http://www.w3.org/DesignIssues/LinkedData.html`
[8] `http://www.w3.org/2001/sw/sweo/`
[9] `http://richard.cyganiak.de/2007/10/lod/`
[10] `http://xmlns.com/foaf/0.1/`
[11] `http://www.w3.org/2003/01/geo/`

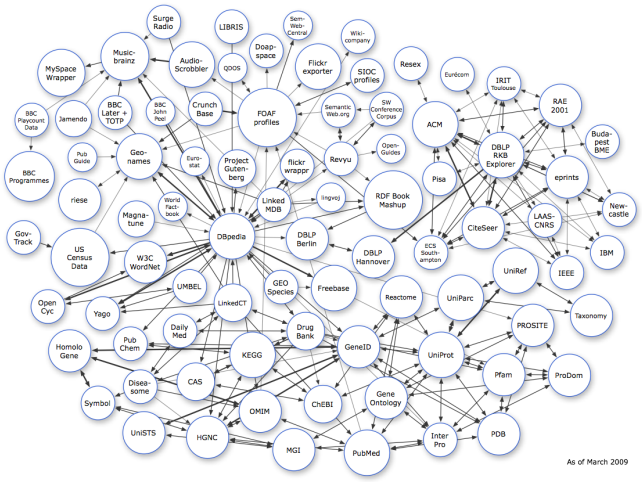

Fig. 4. The LOD cloud in early 2009: over 50 real-world datasets adhering to the linked data principles.

- temporal link types, for example Dublin Core's[12] `dc:created` property, and the `event:time` property of the Event Ontology[13];
- link types such as `dc:isPartOf` for representing structural semantics;
- others, such as `scovo:dimension` of the Statistical Core Vocabulary[14].

My First Linked Data Application

The linked data principles provide a framework, however do not tell about implementation details. Several phrases are kept deliberately rather generic, such as "leads to more data" or "... in order to enable the discovery of more data". This fact suggests that additional advises are needed to actually be able to use linked data in a practical setup. In the following we will describe the steps needed for exploiting linked datasets in an exemplary Web application. Imagine a historical inclined person running a Website, `http://example.org/cw/`, that deals with the topic "Cold War". Let us further assume the site is powered by a popular software such as Wordpress or Drupal. Several manually maintained sections exit about various aspects of the "Cold War": politicians, countries, conflicts, etc. The site maintainer has heard about linked data and has decided to utilise it to enrich his content. What are the necessary steps?

In order to exploit linked dataset properly, basically two steps are required: (i) prepare your own data, and (ii) select appropriate target datasets. We will elaborate on these steps in the following. For a more detailed description the

[12] `http://dublincore.org/documents/dcmi-terms/`
[13] `http://purl.org/NET/c4dm/event.owl`
[14] `http://purl.org/NET/scovo`

reader is referred to [8] where we report on building an interlinked version of the Eurostat statistical dataset.

Prepare Your Data

Typically, the data one is about to use is available in a non-RDF format such as relational data, spreadsheets, etc.—the actual format does not matter as long as it is structured data and the schema is known. One of the first things to make your data Web-of-Data-compliant is to mint, that is, to create, URIs ([9] contains more detailed advises how to achieve this). For example, somehow comparable to what DBpedia does, entities would be identified in the URI space `http://example.org/cw/resource/` (such as `http://example.org/cw/resource/conflict`), whereas an RDF representation would reside in `http://example.org/cw/rdf/` (e.g. `http://example.org/cw/rdf/conflict`) and a human-digestible version in the `http://example.org/cw/html/` space (e.g. `http://example.org/cw/html/conflict`). The ultimate guide about "How to Publish Linked Data on the Web"[15] basically explains the entire publishing process, incl. URI minting, vocabulary selection and deployment issues.

Now, the next challenge is to pick one or more existing vocabularies and extend them as needed for the own purpose. Based on the schema of your data and the selected vocabularies, the RDFising step is rather straight-forward. Experience shows that it is strongly advisable to reuse existing vocabularies and extend them if needed rather then reinventing the wheel for each kind of application. The maintainer of the "Cold War"-site has analysed the entities and the relations occurring in his content and has identified the need to represent people, geographical regions and events in a first iteration. This would, for example, mean to use FOAF for people descriptions, or the Event Ontology to state when and where a certain event, such as a conflict, has taken place. A finer grained description (for example regarding political systems or military aspects) would certainly be desirable, however, the maintainer has decided to start with a simple modelling and refine it in a second iteration; he might even invent a specialised vocabulary for his needs, later on.

The final step in preparing the data is to decide how to expose it (see sidebar 2). A range of options for the deployment of RDF data is available: RDF/XML stand-alone documents, XHTML+RDFa [10] (which basically allows the embedding of an RDF graph in (X)HTML using dedicated attributes) or SPARQL-endpoints (allows agents to query an RDF store via the SPARQL language), etc. as discussed in [8]. As our imaginary "Cold War"-site is based on a content management system, this step is rather straight-forward: URIs are typically minted based on system-specific rules with the possibility to create more legible URIs (for example Drupal allows for so called "clean URIs"[16]).

Again, the "Cold War"-site operator is in a comfortable position: for his system plug-ins exist allowing to expose the data with just a few configuration changes.

[15] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/`
[16] `http://drupal.org/node/15365`

For an out-of-the-box solution to expose relational data on the Web as RDF one may consider using mature frameworks such as the D2R server[a] or Triplify[b]. These tools allow a close-to-instant deployment based on simple configuration and mappings to RDF.
In the enterprise realm basically two option currently exist: the Talis platform[c] and OpenLink's Virtuoso[d], a middleware and database engine. We often use ARC2[e] for our projects; ARC2 is a freely available PHP library for RDF processing targeting at xAMP systems. A comprehensive list of appropriate Web of Data tools, frameworks and libraries is available as well[f].

[a] `http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/`
[b] `http://triplify.org/Overview`
[c] `http://www.talis.com/platform/`
[d] `http://virtuoso.openlinksw.com/`
[e] `http://arc.semsol.org/`
[f] `http://esw.w3.org/topic/SemanticWebTools`

Sidebar 2: Tools and Libraries For Exposing RDF Data

Discovery And Usage of Linked Data

To this end, the data has been made compliant with the Web of Data. We now will tackle the question how to find and select target linked datasets that can be utilised to enrich one's content. Given the current infrastructure, discovering linked datasets on the Web of Data can be challenging. In principle it is possible to learn about the content of a linked data set by applying the follow-your-nose principle (cf. for example [11]), that is, through step-wise inspecting its content. This is a laborious and expensive task. With semantic indexers such as Sindice [12] it is possible to get an idea what a dataset offers. Further, when a SPARQL-end point is advertised using the semantic sitemaps extension [13] one could query the dataset and learn about its internals. However, in terms of scalability, conciseness, as well as convenience the above mentioned may not be the final word.

We have recently started to address the discovery issues by proposing voiD, the "Vocabulary of Interlinked Datasets"[17]. In a nutshell, voiD introduces classes and properties to formally describe the content of a dataset and the interlinking with other datasets. Regarding interlinking, discriminators are defined to describe the type and quantity of links [7]; for example it can be stated that there are "120k links of type `foaf:depiction` from dataset A to dataset B".

However, as long as voiD or comparable technologies are not available or widely deployed, the exploration process is somehow limited. Our "Cold War"-site maintainer would likely inspect the LOD cloud (Fig. 4) or use a semantic indexer such as Sindice[18] to manually find and select worthwhile target datasets. For the "Cold War" site, the operator has picked two datasets: for people-related

[17] `http://semanticweb.org/wiki/VoiD`
[18] `http://sindice.com`

data he uses DBpedia and for geographical data he uses Geonames. On the one hand, this decision has enabled the seamless integration of data from the above mentioned datasets and on the other hand has literally plugged the "Cold War" site into the LOD cloud driving new agents (both humans and machines) to it. Typically, in order to consume RDF data, one would use SPARQL, the RDF equivalent to the relational query language SQL, see also [14] for an in-depth discussion. The complete setup may render as follows: the data provider exposes its data through standardised interfaces such as XHMTL+RDFa, or a SPARQL endpoint and the consumer chooses the best-fitting format for its purpose. A human using a browser will consume an HTML representations, whereas a machine agent, such as an indexer or a content syndicator will likely prefer an RDF serialisation.

Is This It?

To this end we have outlined the minimal steps needed to enhance a Web application by exploiting available linked datasets. To implement the "Cold War" site, one can use available technologies, for example building on widely used platforms such as Drupal[19].

Summarising, we note that with the above described approach, rather than having to learn a multitude of proprietary APIs, a developer learns once RDF (the data model), and, along with the knowledge about a manageable amount of widely deployed vocabularies[20] the only thing left is HTTP to be aware of. In a sense, linked data defines a simple, read-only REST-API with a high reusability factor. Regarding the latter issue, that is, turning the read-only Web of Data into a read-write Web of Data, we have only recently launched a community project called pushback[21], which defines an API and a vocabulary for so-called RDForms in order to update Web 2.0 data sources from the linked data space.

There are still unresolved issues regarding the management of URIs [15] and the creation and selection of vocabularies[22]. Additionally, several more steps may be required in the context of a commercial application: handling provenance and trust[23], addressing quality-of-service (reliability of data sources, etc.), and tackling performance and scalability issues.

Acknowledgements. The author would like to express his deepest gratitude to the Linking Open Data community, especially to Tom Heath for feedback on an early draft of this article and Richard Cyganiak for his never-ending support and willingness to discuss and explain issues around linked data.

[19] `http://drupal.org/project/rdfcck`

[20] Such as FOAF, Dublin Core or SIOC (`http://sioc-project.org/`, for describing blogs, mailing-lists, etc.).

[21] `http://esw.w3.org/topic/PushBackDataToLegacySources`

[22] Although the community has started to address this issue by holding regularly so called VoCamps, see `http://vocamp.org/`.

[23] `http://apassant.net/home/2008/11/msnws/trustprivacy.html`

References

1. T. Heath, "How Will We Interact with the Web of Data?," *IEEE Internet Computing*, vol. 12, no. 5, pp. 88–91, 2008.
2. C. Becker and C. Bizer, "DBpedia Mobile: A Location-Enabled Linked Data Browser," in *WWW 2008 Workshop: Linked Data on the Web (LDOW2008)*, (Beijing, China), 2008.
3. J. Zhao, G. Klyne, and D. Shotton, "Building a Semantic Web Image Repository for Biological Research Images," in *Proceedings of the 5th European Semantic Web Conference*, LNCS, 2008.
4. T. Heath and E. Motta, "Revyu.com: a Reviewing and Rating Site for the Web of Data," in *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pp. 895–902, 2007.
5. D. Ayers, "Evolving the Link," *IEEE Internet Computing*, vol. 11, no. 3, pp. 94–96, 2007.
6. L. Ding and T. Finin, "Characterizing the Semantic Web on the Web," in *5th International Semantic Web Conference, ISWC 2006*, pp. 242–257, 2006.
7. M. Hausenblas, W. Halb, Y. Raimond, and T. Heath, "What is the Size of the Semantic Web?," in *Proceedings of I-Semantics 2008*, (Graz, Austria), pp. 9–16, 2008.
8. W. Halb, Y. Raimond, and M. Hausenblas, "Building Linked Data For Both Humans and Machines," in *WWW 2008 Workshop: Linked Data on the Web (LDOW2008)*, (Beijing, China), 2008.
9. L. Sauermann and R. Cyganiak, "Cool URIs for the Semantic Web," W3C Interest Group Note 31 March 2008, W3C Semantic Web Education and Outreach Interest Group, 2008.
10. B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, "RDFa in XHTML: Syntax and Processing," W3C Recommendation 14 October 2008, W3C Semantic Web Deployment Working Group, 2008.
11. T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Prud'hommeaux, and mc schraefel, "Tabulator Redux: Browsing and Writing Linked Data," in *WWW 2008 Workshop: Linked Data on the Web (LDOW2008)*, (Beijing, China), 2008.
12. G. Tummarello, R. Delbru, and E. Oren, "Sindice.com: Weaving the Open Linked Data.," in *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pp. 552–565, 2007.
13. R. Cyganiak, R. Delbru, H. Stenzhorn, G. Tummarello, and S. Decker, "Exposing Large Datasets with Semantic Sitemaps," in *Proceedings of the 5th European Semantic Web Conference*, LNCS, 2008.
14. D. Lewis, "Building Semantic Web CRUD operations using PHP," Report, IBM developerWorks, 2008.
15. A. Jaffri, H. Glaser, and I. Millard, "Managing URI Synonymity to Enable Consistent Reference on the Semantic Web," in *IRSW2008 - Identity and Reference on the Semantic Web 2008*, (Tenerife, Spain), 2008.