How to Publish Linked Data on the Web

Tom Heath, Michael Hausenblas, Chris Bizer, Richard Cyganiak, Olaf Hartig

Half-day Tutorial at ISWC2008
27th October 2008, Karlsruhe, Germany
Objectives

- Introduce the concept of Linked Data
- Highlight why you would want to publish Linked Data on the Web
- Introduce the principles and best practices of publishing Linked Data on the Web
- Provide an in-depth understanding of the technical design decisions required when publishing Linked Data
- Demonstrate the consumption of Linked Data from the Web
- Look ahead to the future
- Answer your burning Linked Data publishing questions
Tutorial Schedule

- 09:00 – 09:10  Opening
- 09:10 – 09:40  Introduction: What and Why
- 09:40 – 10:30  Publishing Linked Data on the Web: How
- 10:30 – 11:00  Coffee Break
- 11:00 – 11:40  Publishing Linked Data on the Web: How
- 11:40 – 12:00  Consuming Linked Data from the Web
- 12:00 – 12:10  Conclusions and Outlook
- 12:10 – 12:30  Discussion and Linked Data Clinic
ISWC 2008, Tutorial on
How to Publish Linked Data on the Web

Introduction: What and Why

Christian Bizer
Freie Universität Berlin
Overview

1. From a Web of Documents to a Web of Data
   - Web APIs, Microformats, and Linked Data

2. Linked Data Deployment on the Web
   - What data is out there?

3. Applications
   - What is being done with the data?
The Classic Web

**Single global information space**

2. URLs as
   - globally unique IDs
   - retrieval mechanism

3. HTML as shared content format

4. Hyperlinks

**Shortcomings**

- Content is not well structured
- You can not ask expressive queries
- You can not process content within applications
What do we actually want?

Use the Web like a single global database.
Publish structured data directly on the Web.

Different Approaches

2. Web APIs
3. Microformats
4. Linked Data
Web APIs
Mashups

Positive
2. APIs expose structured data
3. APIs enable new applications

Negative
6. Proprietary interfaces
7. Mashups are based only on fixed set of sources
8. You can not set hyperlinks between data objects
Web APIs slice the Web into separate data silos
Microformats

- Embed structured data into HTML pages.
- hCard, hCalendar, hReview, XFN, ...

```html
<div class="vevent">
  <span class="summary">bdigital</span>
  <abbr class="dtstart" title="2008-05-20">May 20</abbr> -
  <abbr class="dtend" title="2007-05-22">22</abbr>
</div>
```

- Compatible with the idea of the Web as single information space.

- Shortcomings
  - Only a fixed set of microformats exist.
  - No way to connect data items.
Use Semantic Web technologies to
2. publish structured data on the Web,
3. set links between data from one data source
to data within other data sources.
Linked Data Principles

1. Use URIs as names for things.

2. Use HTTP URIs so that people can look up those names.

3. When someone looks up a URI, provide useful RDF information.

4. Include RDF statements that link to other URIs so that they can discover related things.

Tim Berners-Lee 2007

http://www.w3.org/DesignIssues/LinkedData.html
The RDF Data Model

```
pd:cygri  rdf:type  foaf:Person
         foaf:name  Richard Cyganiak
         foaf:based_near  dbpedia:Berlin
```
Data objects are identified with HTTP URIs

\[ \text{pd:cygri} = \text{http://richard.cyganiak.de/foaf.rdf#cygri} \]
\[ \text{dbpedia:Berlin} = \text{http://dbpedia.org/resource/Berlin} \]
Dereferencing URIs over the Web

- pd:cygri rdf:type foaf:Person
- foaf:name Richard Cyganiak
- foaf:based_near dbpedia:Berlin
- dp:population 3.405.259
- skos:subject dp:Cities_in_Germany
Dereferencing URIs over the Web

- `pd:cygri` rdf:type `foaf:Person`
  - `foaf:name` Richard Cyganiak
  - `foaf:based_near` `dbpedia:Berlin`
  - `dp:population` 3,405,259
  - `skos:subject` `dp:Cities_in_Germany`

- `dbpedia:Hamburg`
- `dbpedia:Muenchen`

- `skos:subject`
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>...</td>
<td>G2</td>
</tr>
<tr>
<td>type</td>
<td><a href="http://xmlns.com/foaf/0.1/Person">http://xmlns.com/foaf/0.1/Person</a></td>
<td>G1 G2 G3 G4</td>
</tr>
<tr>
<td>seeAlso</td>
<td><a href="http://richard.cyganiak.de/cygri.rdf">http://richard.cyganiak.de/cygri.rdf</a></td>
<td>G2</td>
</tr>
<tr>
<td>seeAlso</td>
<td><a href="http://richard.cyganiak.de/foaf.rdf">http://richard.cyganiak.de/foaf.rdf</a></td>
<td>G3</td>
</tr>
<tr>
<td>nearest airport</td>
<td>...</td>
<td>G1</td>
</tr>
<tr>
<td>phone</td>
<td>tel: +49-175-5630408</td>
<td>G1</td>
</tr>
<tr>
<td>sameAs</td>
<td>Richard Cyganiak</td>
<td>G1</td>
</tr>
<tr>
<td>based_near</td>
<td>...</td>
<td>G1</td>
</tr>
<tr>
<td>based_near</td>
<td>Berlin</td>
<td>G1</td>
</tr>
<tr>
<td>based_near</td>
<td><a href="http://sws.geonames.org/2950159/">http://sws.geonames.org/2950159/</a></td>
<td>G1</td>
</tr>
<tr>
<td>currentProject</td>
<td><a href="http://page.mi.fu-berlin.de/~cyganiak/foaf.rdf#StatCvs">http://page.mi.fu-berlin.de/~cyganiak/foaf.rdf#StatCvs</a></td>
<td>G3</td>
</tr>
<tr>
<td>currentProject</td>
<td><a href="http://www.wiwiss.fu-berlin.de/suhl/bizer#d2rq">http://www.wiwiss.fu-berlin.de/suhl/bizer#d2rq</a></td>
<td>G3 G4</td>
</tr>
<tr>
<td>depiction</td>
<td><img src="image-url" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>male</td>
<td>G1</td>
</tr>
</tbody>
</table>
### Berlin

**URL:** http://dbpedia.org/resource/city/Berlin

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>population</td>
<td>3398888</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td><a href="http://dbpedia.org/City">http://dbpedia.org/City</a></td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>Berlin is the capital city and one of the sixteen Federal States of Germany. It is the country's largest city in area and population, and the second most populous city in the European Union.</td>
<td>G2</td>
</tr>
<tr>
<td>comment</td>
<td>Berlin ist die deutsche Bundeshauptstadt und als Stadtstaat ein eigenständiges Land der Bundesrepublik Deutschland. Berlin ist die bevölkerungsreichste und flächengrößte Stadt Deutschlands und nach Einwohnern die zweitgrößte Stadt der EU.</td>
<td>G2</td>
</tr>
<tr>
<td>label</td>
<td>Berlin</td>
<td></td>
</tr>
<tr>
<td>sameAs</td>
<td><a href="http://sws.geonames.org/2950159/">http://sws.geonames.org/2950159/</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/Berlin">http://dbpedia.org/resource/category/Berlin</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/Capitals_in_Europe">http://dbpedia.org/resource/category/Capitals_in_Europe</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/Cities_in_Germany">http://dbpedia.org/resource/category/Cities_in_Germany</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/German_state_capitals">http://dbpedia.org/resource/category/German_state_capitals</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/Host_cities_of_the_Summer_Olympic_Games">http://dbpedia.org/resource/category/Host_cities_of_the_Summer_Olympic_Games</a></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td><a href="http://dbpedia.org/resource/category/States_of_Germany">http://dbpedia.org/resource/category/States_of_Germany</a></td>
<td></td>
</tr>
<tr>
<td>sourceURL</td>
<td><a href="http://dbpedia.org/resource/category/States_of_Germany">http://dbpedia.org/resource/category/States_of_Germany</a></td>
<td></td>
</tr>
<tr>
<td>depiction</td>
<td><img src="image" alt="Berlin" /></td>
<td></td>
</tr>
<tr>
<td>is birthplace of</td>
<td>Adolf von Baeyer</td>
<td></td>
</tr>
</tbody>
</table>
2. Linked Data Deployment on the Web

- Is this real?
Community effort to

- publish existing open license datasets as Linked Data on the Web
- interlink things between different data sources
As of September 2008
**Spotlight: Geonames**

- over 8 million geographical locations
- feature hierarchy
Spotlight: DBpedia

- extracts structured data from Wikipedia.
- covers over 2.2 million concepts from various domains.

<table>
<thead>
<tr>
<th>Calgary</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://example.com/downtown-calgary.jpg" alt="Downtown Calgary." /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Government</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Mayor</td>
<td>Dave Bronconnier</td>
</tr>
<tr>
<td>- Governing body</td>
<td>Calgary City Council</td>
</tr>
<tr>
<td>- Manager</td>
<td>Owen A. Tobeck</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- City</td>
<td>726.50 km² (280.5 sq mi)</td>
</tr>
<tr>
<td>- Metro</td>
<td>5,107.43 km² (1,972 sq mi)</td>
</tr>
<tr>
<td>Elevation</td>
<td>1,048 m (3,436.3 ft)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- City</td>
<td>988,193</td>
</tr>
<tr>
<td>- Density</td>
<td>1,360.2/km² (3,522.0/sq mi)</td>
</tr>
<tr>
<td>- Metro</td>
<td>1,079,310</td>
</tr>
<tr>
<td>- Population rank</td>
<td>3rd</td>
</tr>
<tr>
<td>- Metro rank</td>
<td>5th</td>
</tr>
</tbody>
</table>

http://en.wikipedia.org/wiki/Calgary

```html
<http://dbpedia.org/resource/Calgary>
dbpedia:native_name "Calgary";
dbpedia:altitude "1048";
dbpedia:population_city "988193";
dbpedia:populationMetro "1079310";
mayor_name
dbpedia:Dave_Bronconnier;
governing_body
dbpedia:Calgary_City_Council;
...
```
Example RDF Links

- **RDF links from DBpedia to other data sources**


- **RDF link from a FOAF profile to DBpedia**

Organizations publishing Linked Data

- **Universities and Research Institutes**
  - Massachusetts Institute of Technology (USA)
  - University of Southampton (UK)
  - Freie Universität Berlin (DE)
  - DERI (IRE)
  - KMi, Open University (UK)
  - University of London (UK)
  - Universität Hannover (DE)
  - University of Pennsylvania (USA)
  - Universität Leipzig (DE)
  - Universität Karlsruhe (DE)
  - Joanneum (AT)
  - University of Toronto (CA)

- **Companies**
  - BBC (UK)
  - OpenLink (UK)
  - Zitgist (USA)
  - Talis (UK)
  - Garlik (UK)
  - Mondeca (FR)
  - Cyc Foundation (USA)
The Bio2RDF Project

- **Goals**
  1. Make bioinformatics data available in RDF format on the Web.
  2. Promote the linked data vision within the bioinformatics community.
  3. Answer questions which were not possible or practical to ask before.

- **Participants**
  - Université Laval, Canada
  - Queer
The Bio2RDF Cloud

- 27 data sources
- 260 million records
- 2,7 billion RDF triples
The Linking Open Drug Data Effort

- W3C HCLSIG task started October 1\textsuperscript{st}, 2008
- Goal: Publish and interlink data sets about drugs and clinical trials.
3. Applications

What can I do with this?

Linked Data Browsers

Linked Data Mashups

Search Engines

Linked Data

Mashups

Search Engines

What can I do with this?
Linked Data Browsers

- Tabulator Browser (MIT, USA)
- Marbles (FU Berlin, DE)
- OpenLink RDF Browser (OpenLink, UK)
- Zitgist RDF Browser (Zitgist, USA)
- Disco Hyperdata Browser (FU Berlin, DE)
- Fenfire (DERI, Irland)
Linked Data Mashups

- Domain-specific applications using Linked Data from the Web
■ Website for rating everything
■ Uses Linked Data to augment ratings
DBtune Slashfacet

- Visualizes music-related Linked Data
- Uses LastFM, MySpace, and BBC data
DBpedia Mobile

- Geospatial entry point into the Web of Data
- Starts with DBpedia, Revyu and Flickr data
Web of Data Search Engines

- Falcons (IWS, China)
- Sindice (DERI, Ireland)
- MicroSearch (Yahoo, Spain)
- Watson (Open University, UK)
- SWSE (DERI, Ireland)
- Swoogle (UMBC, USA)
Falcons

Objects 1 - 10 of 8634 for your search Beijing (1.223 seconds)

Beijing
Types: Capital, City
Labels: 北京” || Рекин || Пекин” || 北京市” || Pequim || Pechino || Beijing || Pékin” || Peking || Pekín”
http://dbpedia.org/resource/Beijing - Described in 184 documents

Beijing
Types: Subject,
Labels: Beijing
http://ontoworld.org/wiki/Special:URIResolver/Beijing - Described in 11 documents

Beijing Guoan
Types: Club
Labels: Beijing Hyundai || 北京国安” || 北京国安足球俱乐部” || Beijing Guoan
http://dbpedia.org/resource/Beijing_Guoan - Described in 30 documents

Beijing
The declaration of this URI may be unauthorized.
Types: Capital City
Labels: Beijing
http://lonely.org/russia#Beijing - Described in 5 documents
Why publish Linked Data on the Web?

- **Linked Data builds on the classic architecture of the Web.**
  - Your data becomes part of a single global data space (the Web of data aka Semantic Web).
  - People can use various data browsers to explore your data.
  - Your data is crawled by Semantic Web search engines and is used by various applications.
  - People start setting links to your data, which might make more people find and use your data.

- **Linked Data is more generic than WebAPIs and Microformats.**
  - Builds on standards in contrast to proprietary Web APIs
  - Enables applications that work against an unbound set of data sources and incorporate new data sources as they become available on the Web.
Publishing Linked Data on the Web
Making a FOAF File into Linked Data
Making a FOAF File into Linked Data

The Forms

Personal

Some information about you, and how people can contact you.

Title (Mr, Mrs, Dr, etc)  
First Name  
Last Name (Family/Given)  
Nickname  
Your Email Address  
Homepage  
Your Picture  
Phone Number  

Work

Information about the place you work.

Work Homepage  
Page describing what you do at work  

School

Where did you go to school?

School Homepage  

People You Know

Tell FOAF-a-matic about some people you know. Click "Add Friend" to add space to add more people. If you friend already has a FOAF description, then place a link to it in the 'seeAlso' field.

Friend--Name  Email  See Also  
Friend--Name  Email  See Also  
Friend--Name  Email  See Also  

http://www.ldodds.com/foaf/foaf-a-matic
Making a FOAF File into Linked Data

```xml
<foaf:knows>
  <foaf:Person>
    <foaf:name>Michael Hausenblas</foaf:name>
    <foaf:mbox_sha1sum>636480acf3caa05e96e612e5e6da6090efd5c71f</foaf:mbox_sha1sum>
    <rdfs:seeAlso rdf:resource="http://www.sw-app.org/foaf/mic.rdf"/>
  </foaf:Person>
</foaf:knows>

<foaf:knows>
  <foaf:Person>
    <foaf:name>Chris Bizer</foaf:name>
    <foaf:mbox_sha1sum>50c02ff93e7d477ace450e3fbddd63d228fb23f3</foaf:mbox_sha1sum>
  </foaf:Person>
</foaf:knows>

<foaf:knows>
  <foaf:Person>
    <foaf:name>Richard Cyganiak</foaf:name>
    <foaf:mbox_sha1sum>39f3c9b7479a83c76596a7c92b61f76dee3f5343</foaf:mbox_sha1sum>
    <rdfs:seeAlso rdf:resource="http://richard.cyganiak.de/foaf.rdf"/>
  </foaf:Person>
</foaf:knows>
```
Making a FOAF File into Linked Data

• Adding URIs for People
Making a FOAF File into Linked Data

• Adding URIs for People

<foaf:knows>
  <foaf:Person rdf:about="http://sw-app.org/foaf/mic.rdf#me">
    <foaf:name>Michael Hausenblas</foaf:name>
    <foaf:mbox_sha1sum>636480acf3cca05e96e612e5e6da6090ef</foaf:mbox_sha1sum>
    <rdfs:seeAlso rdf:resource="http://sw-app.org/foaf/mic.rdf"/>
  </foaf:Person>
</foaf:knows>
Making a FOAF File into Linked Data

- Adding URIs for People

<foaf:knows>
  <foaf:Person rdf:about="http://semanticweb.org/id/Chris_Bizer">
    <foaf:name>Chris Bizer</foaf:name>
    <foaf:mbox_sha1sum>50c02ff93e7d477ace450e3fbddd63d228</foaf:mbox_sha1sum>
  </foaf:Person>
</foaf:knows>
Making a FOAF File into Linked Data

• Enriching Your Profile
<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:admin="http://webns.net/mvcb/">

    <foaf:PersonalProfileDocument rdf:about="#me"/>
    <foaf:maker rdf:resource="#me"/>
    <foaf:primaryTopic rdf:resource="#me"/>
    <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>

</foaf:PersonalProfileDocument>

<foaf:Person rdf:ID="me">
    <foaf:name>Tom Heath</foaf:name>
    <foaf:givenname>Tom</foaf:givenname>
    <foaf:family_name>Heath</foaf:family_name>
    <foaf:nick>cerealtom</foaf:nick>
    <foaf:mbox_sha1sum>1dd0dab71be578c153b8ed70bee284845439706</foaf:mbox_sha1sum>
    <foaf:workplaceHomepage rdf:resource="http://www.talis.com"/>

Making a FOAF File into Linked Data

- Adding Geodata
  - :me foaf:based_near <http://sws.geonames.org/123456>

- Adding Interests
  - :me foaf:topic_interest <http://dbpedia.org/resource/Semantic_Web>
  - :me foaf:topic_interest <http://dbpedia.org/resource/Whisky>

- Adding Your Other Identities
  - :me owl:sameAs <http://data.semanticweb.org/people/tom-heath>
  - :me owl:sameAs <http://kmi.open.ac.uk/people/tom/>
Publishing Linked Data - Process

1. Understand your Data
2. Publish it on the Web as RDF
3. Link it with other Data Sources
Understanding Your Data

- What are the key entities in the dataset?
- What properties do they have?
- How do they relate to other entities?
The Wiskii.com Scenario

- Online whisky shop: Wiskii.com
- New business venture
- For the whisky connoisseur
- Detailed background information from experts
- Contributions from customers
- Custom web app, relational backend
- Simultaneous publication in HTML and RDF
Understanding Your Data

• Things in the Wiskii.com database
  – Distilleries
  – Regions and Locations
  – Founders
  – Owners
  – Brands
  – Products
  – Photos
  – Reviews
  – Comments
  – Prices/Offer
How to Publish Linked Data
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Selecting Vocabularies

- To create RDF graph from our data
- Re-use if possible, it makes your data more valuable
- Create your own if re-use not possible
- Be aware of DC, FOAF, SKOS, SIOC
- Expect to mix & match
Falcons Concept Search

Review is a Class
- label: Review
- comment: A review of an work
- isDefinedBy: http://www.purl.org/stuff/rev
- From danja.tails.com
- http://www.purl.org/stuff/rev#Review
  - Described in 2963 documents

review is a Property
- label: review
- From dbpedia.openlinksw.com
- http://dbpedia.org/property/review
  - Described in 52 documents

hasReview is a ObjectProperty
- range: Review
- domain: Resource
- comment: Associates a work with a review
- From danja.tails.com
- http://www.purl.org/stuff/rev#hasReview
  - Described in 2990 documents

Review
- label: Review
- From dbpedia.openlinksw.com

Search in recommended vocabularies
- http://www.purl.org/stuff/rev# DBpedia Properties
- http://gaia.tdl.ucm.es/ontologies/restaurant.owl# Music
- http://www.w3.org/2006/03/h http://oaei.inrialpes.fr/2005/benchmarks/238/onto.rdf# YAGO Classification
- http://sw.deri.org/2006/07/location/loc# Classes and properties 1 - 10 of 1,527 for your search review (0.3 seconds) [ 1377 classes , 150 properties ]
SchemaWeb.info

SchemaWeb

Search SchemaWeb Directory

Search Results

Review Vocabulary
This document provides a vocabulary specification for the purpose of creating sharable reviews and ratings of blogs, CDs, books, software, anything identifiable. The vocabulary is suitable for inclusion in any RDF-compatible language: FOAF, RSS etc....

Reviews on the Web
A schema for identifying resources that are reviews of other resources.

Geographic Information Metadata
The metadata ontology was developed from International Standard Organization (ISO) Geographic Information - Metadata publications ISO 19115. This work is incomplete and continuing towards a more complete representation.

Platform for Privacy Preferences
The Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents will allow users to be informed of...

MeNow
The motivation for the MeNow schema is to be able to describe a variety of aspects of the current status of someone, either online or off, in a way that the data can be easily aggregated or retrieved. This schema allows the definition of a variety of...

GEM Element Set Qualifier Vocabulary v2.0
The GEM Element Set Qualifier Vocabulary is intended to facilitate more precise description of learning objects for purposes of discovery and retrieval than is possible with unqualified GEM. The Gateway to Educational Materials (http://www.geminfo.org)...

Band Description Vocabulary
This vocabulary is intended to describe relations between persons, any musical instrument they play and persons who played something together. For the scope of this document singing is equivalent to playing an instrument.

SPASE Ontology
The Space Physics Archive Search and Extract (SPASE) Consortium is an international working group developing a space physics data model for scientific data systems (http://www.spase-group.org). They are working on creating a full dictionary of space physics.
Talis Schema-Cache
Spotting good vocabularies

- Existing applications (!)
- Active community
- Good documentation
- Backed by reputable organizations
- Simple
- Few constraints or ontological assumptions
Creating your own

• Stick to what your app needs
• Publish at least an RDFS/OWL file
• Tools: Protégé, Neologism, OpenVocab, …
Linking to existing vocabularies

• rdfs:subClassOf
• rdfs:subPropertyOf
• owl:equivalentClass
• owl:equivalentProperty
• owl:inverseOf
Now we have an RDF graph (with blank nodes)
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Partitioning into “data pages”

- Put the graph online as RDF document(s)
- Huge graph = huge document?
- Hypertext principle: split into sections, interlink them
How to split

- Everything in one document?
- One document per entity?
- Should some entities be grouped together?
- Consider access time, ease of updates, ease of backend access, total # of requests to answer user question
If you already have HTML pages, use the same granularity for the data pages.
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
URIs for data pages

• To put each data page online as RDF doc
• Like web pages, but serve RDF
• E.g. http://wiskii.com/brand/talisker/about.rdf
• “Cool URIs” – stable, no implementation cruft
• http://wiskii.com:2020/demos/cgi-bin/resources.php?id=talisker&output=rdf
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
HTML Variants

• For compatibility with HTML browsers
• HTML rendering of each data page
• *Do we need to add something to the data?*
Content Negotiation

• “generic document” with RDF and HTML variants
• Clients express preferences for formats in `Accept` HTTP header
• Server decides which variant to serve
• Generic document: e.g. …/about
• Format-specific: e.g. …/about.rdf, …/about.html
HTTP Request/Response

GET /brand/talisker/about HTTP/1.0
Host: wiskii.com
Accept: application/rdf+xml

HTTP/1.0 200 OK
Content-Type: application/rdf+xml
Content-Location:
    http://wiskii.com/brand/talisker/about.rdf

<rdf:RDF xmlns:rdf=....
…or put HTML and RDF into one page with RDFa
What we have now

- The RDF graph is online
- In easily digestible chunks
- Chunks can be looked at as RDF or HTML
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Rules

• Permalinks
• Different URIs for different things
• Can be looked up
• URI ownership – don’t squat URI space
Don’t use these

- http://wiskii.com/brand/talisker/about.rdf
- http://wiskii.com/brand/talisker/about
- urn:x-wiskii:brand:talisker
Remember, generic document is at
http://wiskii.com/brand/talisker/about
Hash vs. slash

- http://wiskii.com/brand/talisker
  (with HTTP 303 redirect to .../about)

- http://wiskii.com/brand/talisker/about#it
  (#it is removed for lookup)

- Hash is quick and easy

- 303 is future-proof and less cluttered
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
To help clients understand each data page
Add some triples to about.rdf
dc:date, dc:publisher, dc:license
foaf:primaryTopic, foaf:topic
Link sugar

- Add a bit of information about other entities mentioned in the page
- To support rendering and navigation
- Clients need to make less HTTP requests
- rdfs:label, rdf:type, ...
- Redundancy is okay
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Semantic Sitemaps

• If you publish Linked Data and SPARQL endpoint or RDF dump

• Allows crawlers to find dumps and endpoints

• Add a line to robots.txt:
  Sitemap: sitemap.xml

• Add a file sitemap.xml
<urlset>
  <sc:dataset>
    <sc:datasetLabel>
      The Wiskii.com dataset
    </sc:datasetLabel>
    <sc:linkedDataPrefix>
      http://wiskii.com/
    </sc:linkedDataPrefix>
    <sc:dataDumpLocation>
      http://downloads.wiskii.com/dump.nt.gz
    </sc:dataDumpLocation>
    <sc:sparqlEndpointLocation>
      http://wiskii.com/sparql
    </sc:sparqlEndpointLocation>
    <changefreq>daily</changefreq>
  </sc:dataset>
</urlset>
Publishing Tools
Pubby

- When your data is already in RDF
- Java server in front of SPARQL store
D2R Server

- When your data is in a relational database
- Java server
- Mapping language for describing database-to-RDF mappings
- Provides SPARQL endpoint too
Triplify

- For LAMP applications
- Simple PHP script
- Specify some SQL queries and how the results should be rendered as RDF
Roll your own?

• Build normal HTML site
• Add content negotiation
• Add RDF version of all pages
Linked Data in 7 Easy Steps

1. Select vocabularies
2. Partition the RDF graph into “data pages”
3. Assign a URI to each data page
4. Create HTML variants of each data page
5. Assign a URI to each entity
6. Add page metadata and link sugar
7. Add a Semantic Sitemap
Revisiting the Principles

• Use URIs as names for things
• Use HTTP URIs
• Provide useful information in RDF
• Include RDF links to other URIs
Linking
Other Available Data Sets
• Popular Predicates for Linking
  
  – owl:sameAs
  – foaf:homepage
  – foaf:topic
  – foaf:based_near
  – foaf:maker/foaf:made
  – foaf:depiction
  
  – foaf:page
  – foaf:primaryTopic
  – rdfs:seeAlso
Link to other Data Sets

- Linking Opportunities for Wiskii.com Data
  - Distilleries
    - Link to their DBpedia entries
    - Link to their Parent Companies in DBpedia and WikiCompany
    - Link their Locations to Towns in Geonames/DBpedia
    - Link to Photos internally and externally (via FlickrWrapper)
  - Regions
    - Link to DBpedia and Geonames
  - Brands
    - Link to DBpedia entries? Caution!
    - Link to external Brand Homepages
  - Reviews
    - Link to Brands and Products internally
Link to other Data Sets

- Linking Opportunities for *Wiskii.com* Data
Link to other Data Sets

• Linking Algorithms
  – String Matching
  – Common Key Matching
    • e.g. ISBN, Musicbrainz IDs
  – Property-based Matching
    • Do these two things have the same label, type and coordinates
just as with Wikis, Tags, GWAPs, etc.: humans are good and willing to contribute high-quality content (semantic links, in our case)

certain use cases and/or resource types (e.g. multimedia assets with fine-grained spatio-temporal annotations) are good candidates for manual interlinking
CaMiCatzee [1], a concept demonstrator allowing the FOAF-based search for person depictions on flickr photos

foaf:depicts <http://saphira.blogr.com/#me>
Manual Linking

- quite new linking paradigm, not much experience/research available, yet
- issues
  - exposing link generation vs. hiding it
  - provenance, trust & privacy
  - motivation for end-user
The Semantic Web Client Library
Consuming Linked Data in Your Applications

http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/
Overview

- Introduction
- How does the library work?
- Using the command line tool
- Using the library in applications
Example

What's the interests of the people Tom knows?

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?i WHERE {
    ?p foaf:interest ?i
}
```

Answer:
- "travel"
- <http://tyne.shef.ac.uk/t-rex/>
- <http://www.3worlds.org/>
- ...
- <http://www.dcs.shef.ac.uk/%7Efabio/X-Media/>
- "new things"
- "climbing"

29 RDF documents retrieved
Main Features

- Enables to query the whole Web
  - SPARQL queries
  - find(SPO) queries

- Retrieves relevant RDF documents from the Web dynamically
  - dereferences HTTP URIs
  - follows rdfs:seeAlso links
  - follows alternate or meta links in HTML headers
  - queries Sindice

- Stores retrieved RDF documents as Named Graphs

- Supports GRDDL
Query Processing

- **Splitting SPARQL queries into triple patterns**

  ```sparql
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  SELECT DISTINCT ?i WHERE {
    ?p foaf:interest ?i
  }
  ```

- **Executing a directed-browsing algorithm for each triple pattern**
  - retrieves relevant RDF graphs iteratively
  - finds matching triples (i.e. solutions) in retrieved graphs
Directed-Browsing Algorithm

- **Step 1: Look up URIs in the triple pattern**
  - Look up of Tom's URI
    - GET http://kmi.open.ac.uk/people/tom/
      - Accept: application/rdf+xml;q=1, text/html;q=0.5
    - Response: 303 See Other (http://kmi.open.ac.uk/people/tom/html)
    - GET http://kmi.open.ac.uk/people/tom/html
      - Response: HTML document with
        - `<link rel="meta" type="application/rdf+xml" title="FOAF" href="/people/tom/rdf"/>
        - GET http://kmi.open.ac.uk/people/tom/rdf
      - Response: RDF document
  - Look up of http://xmlns.com/foaf/0.1/knows
    - similar procedure
Directed-Browsing Algorithm

Step 2: Follow rdfs:seeAlso links

FOR EACH triple (a, rdfs:seeAlso, b) in the local graph set where a is a URI in the current triple pattern DO

Look up b

- For 1 we have:
  ( <http://kmi.open.ac.uk/people/tom/> , rdfs:seeAlso , ?t )
  and
  ( foaf:knows , rdfs:seeAlso , ?k )
- Look up each URI that matches ?t or ?k
Directed-Browsing Algorithm

- **Step 3:** Match the triple pattern against all graphs in the local graph set

  - For 1 we get:

    ( <http://kmi.open.ac.uk/people/tom/> , foaf:knows , -11454bb1:11d1409ca3c:-7ff0 )
    ( <http://kmi.open.ac.uk/people/tom/> , foaf:knows , -11454bb1:11d1409ca3c:-7ff4 )
    ( <http://kmi.open.ac.uk/people/tom/> , foaf:knows , -11454bb1:11d1409ca3c:-7ff8 )
    ( <http://kmi.open.ac.uk/people/tom/> , foaf:knows , -11454bb1:11d1409ca3c:-7ff1 )
    ( <http://kmi.open.ac.uk/people/tom/> , foaf:knows , -11454bb1:11d1409ca3c:-7fee )

...
Directed-Browsing Algorithm

- Step 4: For each matching triple:
  
  e.g. (http://kmi.open.ac.uk/people/tom/ , foaf:knows , http://semanticweb.org/id/Richard_Cyganiak)

  1. Look up all new URIs in the triple
    - For http://semanticweb.org/id/Richard_Cyganiak
      we retrieve a new RDF document from:

  2. Follow rdfs:seeAlso links for all new URIs in the triple
    - For http://semanticweb.org/id/Richard_Cyganiak we find
      
      Tom's FOAF document

      ...
      http://semanticweb.org/id/Richard_Cyganiak
      rdfs:seeAlso http://richard.cyganiak.de/foaf.rdf
      ...

    - New document http://richard.cyganiak.de/foaf.rdf
Directed-Browsing Algorithm

- **Step 4**: For each matching triple ...
  - overall 21 new graphs in local graph set

- **Step 5**: Match the triple pattern against all newly retrieved graphs
  - nothing new for 1

- **Step 6**: Repeat steps 4 and 5 alternately until
  - no new matching triples in step 5,
  - maximum number of retrieval steps reached, or
  - timeout reached

- **Another query**:
  - triple pattern: ?p1 foaf:knows ?p2
  - seeded with Tom's FOAF document
  - after 1 min: 9812 matching triples, 372 graphs
Sindice Support

- **Sindice**
  - index of documents with structured data
  - provides an API to find documents
  - URI-based search: finds documents that mention the given URI

- URI look up triggers a query to the Sindice service

- More complete results:
  - triple pattern: ?prop rdfs:range foaf:Person
  - 1 matching triple without Sindice look up
  - 19 matching triples with Sindice look up

- Beware: number of discovered graphs may grow significantly
  - 2 graphs vs. 134 graphs
Implementation

- Implemented in Java
- Based on the Jena framework
- BSD license
- Part of the NG4J (Named Graphs API for Jena)
  - extends Jena with methods to parse, manipulate, query, and serialize sets of Named Graphs
- Multi-threaded for faster retrieval
Command Line Tool

- **Execute SPARQL or find(SPO) queries**
  
  ```
  ./bin/semwebquery -retrieveduris -sindice
  -find "ANY rdfs:range <http://xmlns.com/foaf/0.1/Person>"
  ```

- **Parameters (selection):**
  - `-find <Filename>` – executes a find(SPO) query (use ANY as wildcard)
  - `-sparql <Query>` – executes the given SPARQL query
  - `-sparqlfile <Filename>` – executes the SPARQL query in the specified file
  - `-load <URI>` – loads graph from the given URI into local cache before execution
  - `-sindice` – enables Sindice-based URI search during execution
  - `-maxsteps <Integer>` – sets maximum number of retrieval steps
  - `-timeout <Integer>` – sets timeout of the query in seconds
Using the Library

- **Main interface: SemanticWebClient class**
  - implements the NamedGraphSet interface defined by NG4J
  - methods (selection):
    - `read(url, lang)` – reads a Named Graph into the local graph set
    - `addRemoteGraph(uri)` – issues a URI look up
    - `find(pattern)` – executes find(SPO) query and returns iterator over all matching triples
    - `asJenaModel(nameOfDfltGraph)` – returns a jena model view on the local graph set
Using the Library

```java
import com.hp.hpl.jena.query.*;
import de.fuberlin.wiwiss.ng4j.semwebclient.SemanticWebClient;

SemanticWebClient semweb = new SemanticWebClient();

String queryString = ...  // Specify the query

// Execute the query and obtain the results
Query query = QueryFactory.create(queryString);
QueryExecution qe = QueryExecutionFactory.create(query, semweb.asJenaModel("default"));
ResultSet results = qe.execSelect();

// Consume the results
while (results.hasNext())
{
    QuerySolution s = results.nextSolution();
    ...
}
```
Using the Library

- **Methods of SemanticWebClient for custom control:**
  - reloadRemoteGraph(uri) – refresh local copy
  - clear() – clears the local graph set
  - requestDereferencing(uri, step, listener) – initiates URI look up
  - requestDereferencingWithSearch(uri, step, derefListener, searchListener)

- setConfig(option, value) – sets configuration option
  - CONFIG_MAXSTEPS
  - CONFIG_TIMEOUT
  - CONFIG_DEREF_CONNECT_TIMEOUT
  - CONFIG_DEREF_READ_TIMEOUT
  - CONFIG_ENABLE_SINDICE
  - ...

Provenance Information

- **SemWebTriple class**
  - provided by `find(pattern)` method of `SemanticWebClient`
  - method `getSource()` returns URI of the containing graph

- **Provenance graph**
  - always in the local graph set
  - contains statements about the source URL and retrieval time for each retrieved graph

- **Dereferenced URIs**
  - lists of successfully and unsuccessfully retrieved URIs
    - `successfullyDereferencedURIs()`
    - `unsuccessfullyDereferencedURIs()`
  - information about redirected URIs
    - `redirectedURIs()`
    - `getRedirectURI(uri)`
Conclusion

The Semantic Web Client Library
- enables queries over the whole Web
- dynamically retrieves RDF data during query execution
- is available from:
  http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/

Future work:
- smart caching and replacement of retrieved graphs
Conclusions and Outlook
Summary

- Linked Data is a generic approach for publishing structured data on the Web.
  - Builds on standards in contrast to proprietary Web APIs
- Linked Data builds on the classic architecture of the Web.
  - Links allow you to discover unexpected things
- The Web of Linked Data is growing rapidly.
- There is an increasing number of application prototypes that consume Linked Data from the Web.
Linked Data Prospects in 2009
Even More Data!

1. Conversion of further open license datasets into RDF
2. Wrappers around existing applications

• Growing number of tools available
  – D2R Server
  – Triplify
  – Pubby

• Growing number of wrappers for existing systems
  – Drupal
  – Wordpress
  – osCommerce
1. Increase the amount of links between datasets
2. Increase the quality of these links

- Today: Simple pattern- and graph-matching based techniques used for automated interlinking.

- There is lots of existing work in database and knowledge representation communities on identity resolution to be used.
Data Fusion

Users want an integrated view on all data that is available about an object!

Raises well known but hard problems:
- Schema mapping
- Inconsistency resolution
- Trust / information quality
In order to do anything serious with data from the Web, its license terms have to be clear.

- Need for
  - proper licensing vocabularies for dedicating data to the public domain
  - best practices on how to annotate data with licensing meta-data

- Can build on
  - Open Data Commons Public Domain Dedication & Licence (PDDL) (see LDOW2008 paper)
  - Creative Commons Licensing Framework
End user friendly views on the data
  - ordering and merging of properties
  - dealing with information overflow

More advanced data analysis features
  - aggregation, drill down
  - calculations, Web-Excel

Explanations about data provenance and trustworthiness

Interesting work happening around Freebase

Need for real tools, not only proof of concept prototypes!
IJSWIS Special Issue on Linked Data

- Special Issue of International Journal on Semantic Web and Information Systems
- Editor-in-Chief: Amit Sheth
- Guest Editors: Chris Bizer, Tom Heath, Martin Hepp
- Submission deadline in January 2009
Getting Involved

- Wiki Page
  - http://esw.w3.org/topic/SweolG/TaskForces/CommunityProjects/LinkingOpenData

- Mailing List
  - public-lod@w3.org
  - http://lists.w3.org/Archives/Public/public-lod/

- Participating in the project
  - Put your name on the Wiki page
  - Subscribe to the mailing list
  - Do something useful

- Tutorial: How to Publish Linked Data on the Web
  - http://linkeddata.org/docs/how-to-publish
Discussion and Linked Data Clinic