INFO/CS 4302
Web Information Systems

FT 2012
Week 10 / Lecture 18:
Publishing Structured Data on the Web

- Bernhard Haslhofer -
Plan for today...

• Recap

• Publishing Data – The Linked Data Way

• Embedding Structured Data into HTML

• Homework 8
PUBLISHING DATA – THE LINKED DATA WAY
What is Linked Data?

• A method to build a **Web of Data**
• Architectural style, set of standards
Dereferencing LD Resources

• Distinguish between non-information and information resource

• Sample non-information resource
  – http://dbpedia.org/resource/The_Shining_(film)

• Sample information resource
  – http://dbpedia.org/page/The_Shining_(film) - HTML
  – http://dbpedia.org/data/The_Shining_(film) - RDF
Dereferencing LD Resources

GET http://dbpedia.org/resource/The_Shining_(film)
Accept: application/rdf+xml

303 See Other
Location: http://dbpedia.org/data/The_Shining_(film)

GET http://dbpedia.org/data/The_Shining_(film)
Accept: application/rdf+xml

200 OK
...
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF ...
Publishing Large RDF Datasets

• Run a servlet that implements the 303 publishing approach
  – for non information resources
    • parse Accept Header field
    • Redirect (303 See Also) to corresponding information resource

• Generate RDF Serialization dynamically from underlying data storage
Publishing Vocabularies

• Hash-based URIs
  – e.g., http://example.com/vocab.rdf#ClassA
  – Suited to group the description of a moderate number of related terms into one RDF document
  – Agent can retrieve terms with a single request

• Slash-based URIs
  – e.g., http://example.com/vocab/ClassB
  – Suited to split terms in large vocabularies into one document per term
  – No need to download a massive document
Property: foaf:name

name - A name for some thing.

Status: testing

Domain: having this property implies being a Thing

The name of something is a simple textual string.

XML language tagging may be used to indicate the language of the name. For example:

```xml
<foaf:name xml:lang="en">Dan Brickley</foaf:name>
```

FOAF provides some other naming constructs. While foaf:name does not explicitly represent name substructure (family vs given etc.) it does provide a basic level of interoperability. See the issue tracker for status of work on this issue.

The name property, like all RDF properties with a range of rdfs:Literal, may be used with XMLLiteral datatypes values (multiple names are acceptable whether they are in the same language or not). XMLLiteral usage is not yet widely adopted. Feedback on this aspect of the FOAF design is particularly welcomed.

---

Property: foaf:nick

nickname - A short informal nickname characterising an agent (includes login identifiers, IRC and other chat nicknames).

Status: testing
EMBEDDING STRUCTURED DATA INTO HTML
Embedding Data in HTML

• The **Linked Data publishing recipes** separates
  – raw-data representations (RDF/XML, Turtle, N3)
  – human-readable representations (HTML)

• We can also embed structured data directly into HTML
  – RDFa
  – Microformats
  – Microdata
RDFa / Microformats / Microdata

• Mechanisms for embedding structured data in Web documents
• Define or use a set of attributes to augment presentation-oriented (HTML) documents with structured data
• User agents can extract triples from Web pages
RDFa Example

**XHTML**

```html
<div>
  <h2>The trouble with Bob</h2>
  <h3>Alice</h3>
</div>
```

**XHTML + RDFa**

```html
<div xmlns:dc=http://purl.org/dc/elements/1.1/>
  <h2 property="dc:title">The trouble with Bob</h2>
  <h3 property="dc:creator">Alice</h3>
</div>
```
RDFa Example

XHTML + RDFa

<div xmlns:dc="http://purl.org/dc/elements/1.1/">
  <div about="/alice/posts/trouble_with_bob">
    <h2 property="dc:title">The trouble with Bob</h2>
    <h3 property="dc:creator">Alice</h3>
  </div>
  <div about="/alice/posts/jos_barbecue">
    <h2 property="dc:title">Jo's Barbecue</h2>
    <h3 property="dc:creator">Eve</h3>
  </div>
</div>
RDFa Example

XHTML + RDFa

<div typeof="foaf:Person" xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <p property="foaf:name">Alice Birpemswick</p>
  <p>Email: <a rel="foaf:mbox" href="mailto:alice@example.com">alice@example.com</a></p>
  <p>Phone: <a rel="foaf:phone" href="tel:+1-617-555-7332">+1 617.555.7332</a></p>
</div>
RDFa Attributes

• **about** and **src**: the resource the metadata is about
• **rel** and **rev**: (reverse) relationship between resources
• **href** and **resource**: the partner resource
• **property**: a property for the content of an element
• **content**: override content of an element
• **datatype**: specify the datatype of text
• **typeof**: specifies the RDF type(s) or a subject
Real-world RDFa Examples

• OReilly.com: http://oreilly.com/catalog/9780596520694/

• NY Times: http://thecaucus.blogs.nytimes.com/2012/10/21/mondays-debate-puts-focus-on-foreign-policy-clashes/
RDFa 1.1 Distiller and Parser

Warning: This version implements RDFa 1.1 Core, including the handling of the Role Attribute. The distiller can also run in XHTML+RDFa 1.0 mode (if the incoming XHTML content uses the RDFa 1.0 DTD and/or sets the version attribute). The package available for download, although it may be slightly out of sync with the code running this service.

**Distill by URI**

**URI:**

http://thecaucusblogs.nytimes.com/2012/10/21/mondays-debate-puts-focus-

**Output Format:**

**Returned content:**

**Expand vocabularies:**

**Generate warnings for non RDFa 1.1 Lite usage:**

› More (non-standard) options
RDFa illustrated

http://www.minddevelopmentanddesign.com/blog/the-importance-of-rdfa-infographic/
Microformats

• A Microformat extends conventional HTML tags with semantic information
• Started by Technorati, Inc.; now community-driven (IRC, mailing list, blogs)
• Make use of the following (X)HTML attributes
  – class
  – rel
Microformats Example

**HTML**

```html
<div>
   <div>Joe Doe</div>
   <div>Jo</div>
   <div>The Example Company</div>
   <div>604-555-1234</div>
   <a href="http://example.com/">http://example.com/</a>
</div>
```

**XHTML + Microformats**

```html
<head profile="http://www.w3.org/2006/03/hcard">...
...
<div class="vcard">
   <div class="fn">Joe Doe</div>
   <div class="nickname">Jo</div>
   <div class="org">The Example Company</div>
   <div class="tel">604-555-1234</div>
   <a class="url" href="http://example.com/">http://example.com/</a>
</div>
```
Microformats Example

Property List

hCard properties (sub-properties in parentheses like this)

Required:

- **fn**
- **n** (family-name, given-name, additional-name, honorific-prefix, honorific-suffix)

Optional:

- **adr** (post-office-box, extended-address, street-address, locality, region, postal-code, country-name, type, value)
- **agent**
- **bday**
- **category**
- **class**
- **email** (type, value)
- **geo** (latitude, longitude)
- **key**
- **label**
- **logo**
- **maili**
- **nickname**
- **note**
- **org** (organization-name, organization-unit)
- **photo**
- **rev**
- **role**
Available Microformats

Specifications

The list of current, stable Microformats open standard specifications.

- hCalendar - hCalendar creator
- hCard - hCard creator
- rel-license
- rel-nofollow
- rel-tag
- VoteLinks
- XFN - XFN creator
- XMDP
- XOXO

*If you're tempted to try your hand at writing a microformat please read the process page first!*

Drafts

Drafts are newer microformats, for which the specifications haven't been completed yet. Drafts are somewhat mature in the development process (see exploratory-discussions for additional efforts that are not as far along in the process). The stability of these documents cannot be guaranteed, and implementers should be prepared to keep abreast of future developments and changes. Please watch the wiki pages for updates.

- adr - for marking up address information
- geo - for marking up WGS84 geographic coordinates (latitude; longitude)
- hAtom - syndicating episodic content (e.g. weblog postings)
- hAudio
Microformats vs. RDFa

Sample vCard Object in RDFa (see http://www.w3.org/Submission/vcard-rdf/)
Microformats vs. RDFa

```html
<div class="vcard">
  <span class="fn">Bernhard Haslhofer</span>
  <div class="adr">
    <div class="street-address">301 College Avenue</div>
    <span class="locality">Ithaca</span>
    <span class="postal-code">14850</span>
    <span class="country-name">United States</span>
  </div>
  <a class="email" href="mailto:bernhard.haslhofer@cornell.edu">
    bernhard.haslhofer@cornell.edu
  </a>
</div>

hCard sample created with http://microformats.org/code/hcard/creator
<table>
<thead>
<tr>
<th>Microformats</th>
<th>RDFa</th>
</tr>
</thead>
<tbody>
<tr>
<td>flat namespace</td>
<td>XML namespaces</td>
</tr>
<tr>
<td>support HTML4, XHTML 1.1, and HTML 5</td>
<td>support for XHTML 1., HTML 5</td>
</tr>
<tr>
<td>use latent HTML attributes</td>
<td>introduces new metadata attributes</td>
</tr>
<tr>
<td>vocabulary defined by one organization/community</td>
<td>open to any RDF-based vocabulary</td>
</tr>
</tbody>
</table>
Microdata (HTML5)

• A very young HTML 5 proposition that extends Microformats and addresses its shortcomings
• Items are created within an itemscope
• Ever item is assigned an arbitrary number of properties (itemprop) and relationships (itemref)
• Uses global identifiers for typing and naming items
Microdata Example

```html
<div itemscope itemtype="http://schema.org/Person">
  <span itemprop="name">Bernhard Haslhofer</span>,
  <span itemprop="nickname">behas</span>.
  <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress">301 College Avenue</span>
    <span itemprop="addressLocality">Ithaca</span>
    <span itemprop="addressCountry">United States</span>
  </div>
</div>
```
What are the pros / cons of embedding structured data directly into HTML?
## Web Data Publishing Approaches

<table>
<thead>
<tr>
<th></th>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linked Data</strong></td>
<td>Separates data / presentation markup</td>
<td>Technical complexity</td>
</tr>
<tr>
<td><strong>Conneg IR / NIR</strong></td>
<td>Web Architecture</td>
<td>Communication overhead</td>
</tr>
<tr>
<td><strong>RDFa / Microformats / Microdata</strong></td>
<td>Easy to implement / Search engine friendly</td>
<td>Mixes data / presentation markup / Becomes messy</td>
</tr>
</tbody>
</table>
Schema.org and the FB Open Graph Protocol
Rich snippets (microdata, microformats, and RDFa)

About rich snippets and structured data

Snippets—the few lines of text that appear under every search result—are designed to give users a sense for what’s on the page and why it’s relevant to their query.
What is Schema.org?

This site provides a collection of schemas, i.e., html tags, that webmasters can use to markup their pages in ways recognized by major search providers. Search engines including Bing, Google and Yahoo! rely on this markup to improve the display of search results, making it easier for people to find the right web pages.

Many sites are generated from structured data, which is often stored in databases. When this data is formatted into HTML, it becomes very difficult to recover the original structured data. Many applications, especially search engines, can benefit greatly from direct access to this structured data. On-page markup enables search engines to understand the information on web pages and provide richer search results in order to make it easier for users to find relevant information on the web. Markup can also enable new tools and applications that make use of the structure.

A shared markup vocabulary makes easier for webmasters to decide on a markup schema and get the maximum benefit for their efforts. So, in the spirit of sitemaps.org, Bing, Google and Yahoo! have come together to provide a shared collection of schemas that webmasters can use.

We invite you to get started!
<h1>Pirates of the Caribbean: On Stranger Tides (2011)</h1>
Jack Sparrow and Barbossa embark on a quest to find the elusive fountain of youth, only to discover that Blackbeard and his daughter are after it too.

Director: Rob Marshall
Writers: Ted Elliott, Terry Rossio, and 7 more credits
Stars: Johnny Depp, Penelope Cruz, Ian McShane
8/10 stars from 200 users. Reviews: 50.
schema.org / Microdata example

```html
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Pirates of the Carribean: On Stranger Tides (2011)</h1>
  <span itemprop="description">Jack Sparrow and Barbossa embark on a quest to find the elusive fountain of youth, only to discover that Blackbeard and his daughter are after it too.</span>
  Director:
  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Rob Marshall</span>
  </div>
  Writers:
  <div itemprop="author" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Ted Elliott</span>
  </div>
  <div itemprop="author" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Terry Rossio</span>
  </div>
  , and 7 more credits
  Stars:
  <div itemprop="actor" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Johnny Depp</span>,
  </div>
  <div itemprop="actor" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Penelope Cruz</span>,
  </div>
  <div itemprop="actor" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">Ian McShane</span>
  </div>
  <div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AverageRating">
    <span itemprop="ratingValue">8</span>/10 stars from 200 users.
  </div>
  Reviews: <span itemprop="reviewCount">50</span>.</div>
```
schema.org

• Defines
  – a number of types (e.g., person), organized in an inheritance hierarchy
  – a number of properties (e.g., name)

• Extension mechanisms to extend the schemas

• OWL representation:
  http://schema.org/docs/schemaorg/owl

• http://schema.rdfs.org/index.html
Graph API

At Facebook's core is the social graph; people and the connections they have to everything they care about. The Graph API presents a simple, consistent view of the Facebook social graph, uniformly representing objects in the graph (e.g., people, photos, events, and pages) and the connections between them (e.g., friend relationships, shared content, and photo tags).

Every object in the social graph has a unique ID. You can access the properties of an object by requesting \texttt{https://graph.facebook.com/ID}. For example, the official page for the Facebook Platform has id 19292868552, so you can fetch the object at \texttt{https://graph.facebook.com/19292868552}:

```
{
    "name": "Facebook Platform",
    "website": "http://developers.facebook.com",
    "username": "platform",
    "founded": "May 2007",
    "company_overview": "Facebook Platform enables anyone to build...",
    "mission": "To make the web more open and social.",
    "products": "Facebook Application Programming Interface (API)...",
    "likes": 449921,
    "id": 19292868552,
    "category": "Technology"
}
```

Alternatively, people and pages with usernames can be accessed using their username as an ID. Since "platform" is the username for the page above, \texttt{https://graph.facebook.com/platform} will return what you expect. All responses are JSON objects.

All objects in Facebook can be accessed in the same way:

- Users: \texttt{https://graph.facebook.com/btaylor} (Bret Taylor)
name: "Coca-Cola",
is_published: true,
website: "http://www.coca-cola.com",
username: "coca-cola",
founded: "1886",
description: "Created in 1886 in Atlanta, Georgia, by Dr. John S. Pemberton, Coca-Cola was first offered as a fountain beverage at Jacob's␣

Coca-Cola was patented in 1887, registered as a trademark in 1893 and by 1895 it was being sold in every state and territory in the United States. Coca-Cola might owe its origins to the United States, but its popularity has made it truly universal. Today, you can find Coca-Cola in countries around the world, from the Americas to Europe, Asia, Australia, and Africa. It has become one of the world's most recognized brands.

Coca-Cola Page House Rules: http://CokeURL.com/q28a",
about: "The Coca-Cola Facebook Page is a collection of your stories showing how people from around the world have helped make Coke into a special part of your family's life."

- location: {
    latitude: 19.2,
    longitude: -96.1333
},
checkins: 127,
talking_about_count: 1687218,
category: "Food/beverages",
id: "40796308305",
link: "http://www.facebook.com/coca-cola",
likes: 53165520,
- cover: {
    cover_id: "10151948639733396",
    offset_y: 0
}
The Coca-Cola Facebook Page is a collection of your stories showing how people from around the world have helped make Coke into what it is today.

Coca-Cola was patented in 1887, registered as a trademark in 1893 and by 1895 it was being sold in every state and territory in the United States. In 1899, The Coca-Cola Company began franchised bottling operations in the United States.

Coca-Cola might owe its origins to the United States, but its popularity has made it truly universal. Today, you can find Coca-Cola in virtually every part of the world.
Open Graph

At Facebook's core is the social graph: people and the connections they have to everything they care about. Historically, Facebook has managed this graph and has expanded it over time as we launch new products (ex: photos, places). In 2010, we introduced an early version of Open Graph, an extension of the social graph, via the Open Graph protocol, to include third-party websites and pages that people liked throughout the web. We are now extending the Open Graph to include arbitrary actions and objects created by third-party apps and enabling these apps to integrate deeply into the Facebook experience.
Introduction

The Open Graph protocol enables any web page to become a rich object in a social graph. For instance, this is used on Facebook to allow any web page to have the same functionality as any other object on Facebook.

While many different technologies and schemas exist and could be combined together, there isn’t a single technology which provides enough information to richly represent any web page within the social graph. The Open Graph protocol builds on these existing technologies and gives developers one thing to implement. Developer simplicity is a key goal of the Open Graph protocol which has informed many of the technical design decisions.

Basic Metadata

To turn your web pages into graph objects, you need to add basic metadata to your page. We’ve based the initial version of the protocol on RDFa which means that you’ll place additional <meta> tags in the <head> of your web
protocol builds on these existing technologies and gives developers one thing to implement. Developer simplicity is a key goal of the Open Graph protocol which has informed many of the technical design decisions.

---

**Basic Metadata**

To turn your web pages into graph objects, you need to add basic metadata to your page. We've based the initial version of the protocol on **RDFa** which means that you'll place additional `<meta>` tags in the `<head>` of your web page. The four required properties for every page are:

- **og:title** - The title of your object as it should appear within the graph, e.g., "The Rock".
- **og:type** - The type of your object, e.g., "video.movie". Depending on the type you specify, other properties may also be required.
- **og:image** - An image URL which should represent your object within the graph.
- **og:url** - The canonical URL of your object that will be used as its permanent ID in the graph, e.g., "http://www.imdb.com/title/tt0117500/".

As an example, the following is the Open Graph protocol markup for *The Rock on IMDB*:

```html
<html prefix="og: http://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="http://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="http://ia.media-imdb.com/images/rock.jpg" />
...
</head>
...
</html>
```

**Optional Metadata**

The following properties are optional for any object and are generally recommended:
Object Types
In order for your object to be represented within the graph, you need to specify its type. This is done using the `og:type` property:

```html
<meta property="og:type" content="website" />
```

When the community agrees on the schema for a type, it is added to the list of global types. All other objects in the type system are CURIES of the form

```html
<head prefix="my_namespace: http://example.com/ns#">
<meta property="og:type" content="my_namespace:my_type" />
```

The global types are grouped into verticals. Each vertical has its own namespace. The `og:type` values for a namespace are always prefixed with the namespace and then a period. This is to reduce confusion with user-defined namespaced types which always have colons in them.

Music

- Namespace URI: http://ogp.me/ns/music#

`og:type` values:

`music.song`

- `music:duration` - integer >=1 - The song's length in seconds.
- `music:album` - `music.album array` - The album this song is from.
- `music:album:disc` - integer >=1 - Which disc of the album this song is on.
- `music:album:track` - integer >=1 - Which track this song is.
- `music:musician` - profile array - The musician that made this song.
Readings

• Heath, Bizer: Linked Data: Evolving the Web into a Global Data Space (Chapter 5)
  http://linkeddatabook.com/editions/1.0/

• Web Data Commons.
  http://webdatacommons.org/

• Mika, Potter: Metadata Statistics for a Large Web Corpus

• Schema.org Tutorial: http://schema.org/docs/gs.html

• Open Graph Protocol:
  http://ogp.me/
HOMEWORK 8
http://www.infosci.cornell.edu/Courses/info4302/2012fa/homeworks.php#hw8