RDFa

Acknowledgement

inspired by

"RDFa - Bridging the Web of Documents and the Web of Data"

Michael Hausenblas, JOANNEUM RESEARCH
Ivan Herman, W3C
Ben Adida, Creative Commons / Harvard University

AND

http://www.w3.org/TR/xhtml-rdfa-primer/

RDFa

- RDFa (Resource Description Framework in Attributes)
  - provides a set of markup attributes to semantically annotate (X)HTML or HTML5 documents.
  - Latest version RDFa 1.1
- Goals of RDFa
  - Bridging the "Web of Documents" and the "Web of Data" closer.

The Web of Documents View

- Metadata should be added to an (X)HTML page without too much hassle
- (X)HTML page should remain valid under some well defined constraints (DTD, Schema, etc.)
- The additional data should be accessible, if necessary, in a DOM tree for further processing
- "Don't repeat yourself" (DRY) - the (meta)data should make use of the ("visible") XHTML content as much as possible
- "Mashup" of various data should be possible

The Web of Documents View (cont.)

- The additional data should possibly **rely on externally defined vocabularies** (simple or complex)
- Many different types of vocabularies should be **usable** in the same document
- There should be an easy way to **transform** the data into an RDF repository to combine it with other data on the Web of Data

The Web of Documents View - Existing Solution (I)

- (X)HTML has the <meta> and <link> elemets to add metadata to content, but
  - They can only be used in the header (i.e., no way of DRY for the content)
  - Only a limited number of terms for the attribute name.

The Web of Documents View - Existing Solution (II)

- Microformat
  - **class** attribute: class name.
  - **rel** attribute: relationship with the target address.

```
<div>
   I had a meeting with
   <a href="http://example/people/john" rel="met"> John Smith </a>
   in London.
    <span class="geo">
     <span class="latitude">51.5072</span>,
     <span class="longitude">0.1275</span>
   </span>
</div>
```

A list of existing classes and rel-values:
http://microformats.org/wiki/existing-classes
http://microformats.org/wiki/existing-rel-values

The Web of Documents View - Existing Solution (III)

- Microformats (re)use existing (X)HTML attributes to encode some data
- Through a suitable XSLT transformation the (X)HTML can be converted into RDF, but..
  - a separate approval process has to define each vocabulary and a separate XSLT has to be developed again and again
  - becomes complex if more than one vocabulary is to be used on one page
  - here might be clashes with the goals of the XHTML attributes
- The RDF data must be written in a separate serialization format (RDF/XML, Turtle, etc.)
- Violated the DRY principle

The Web of Data View

- It is not easy to document and describe RDF content (e.g., vocabulary definitions, ontologies, etc.)
- The only way is to produce the vocabulary in one syntax (RDF/XML, Turtle, ontology syntaxes) and provide a separate description in, say, XHTML
- Best would be better to be able to **combine the two**

What is RDFa

- From the Web of Documents' point-of-view: a set of new (X)HTML attributes to express metadata within (X)HTML

- From the Web of Data point-of-view: a serialization format for RDF (such as RDF/XML, etc.), where the RDF triples are "embedded" into (X)HTML

The Basics of RDFa: An Example

- Markup text to make it machine readable

http://example.com/book/harry_potter

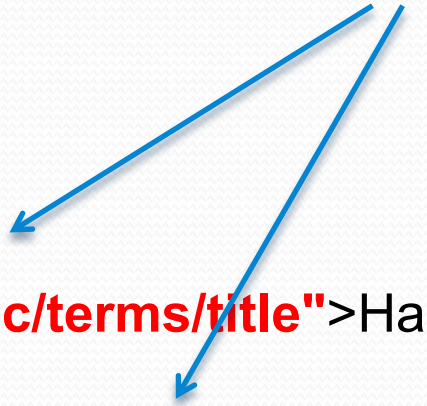
```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2>Harry potter and the chamber of secrets</h2>
  <p>November 3, 2002</p>
  ...
</body>
```

Hints on web pages

- Markup HTML text to make it machine readable using Dublin Core vocabulary

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2 property="http://purl.org/dc/terms/title">Harry potter and the
chamber of secrets</h2>
  <p>Date: <span property="http://purl.org/dc/terms/
created">November 3, 2002</span></p>
  ...
</body>
```

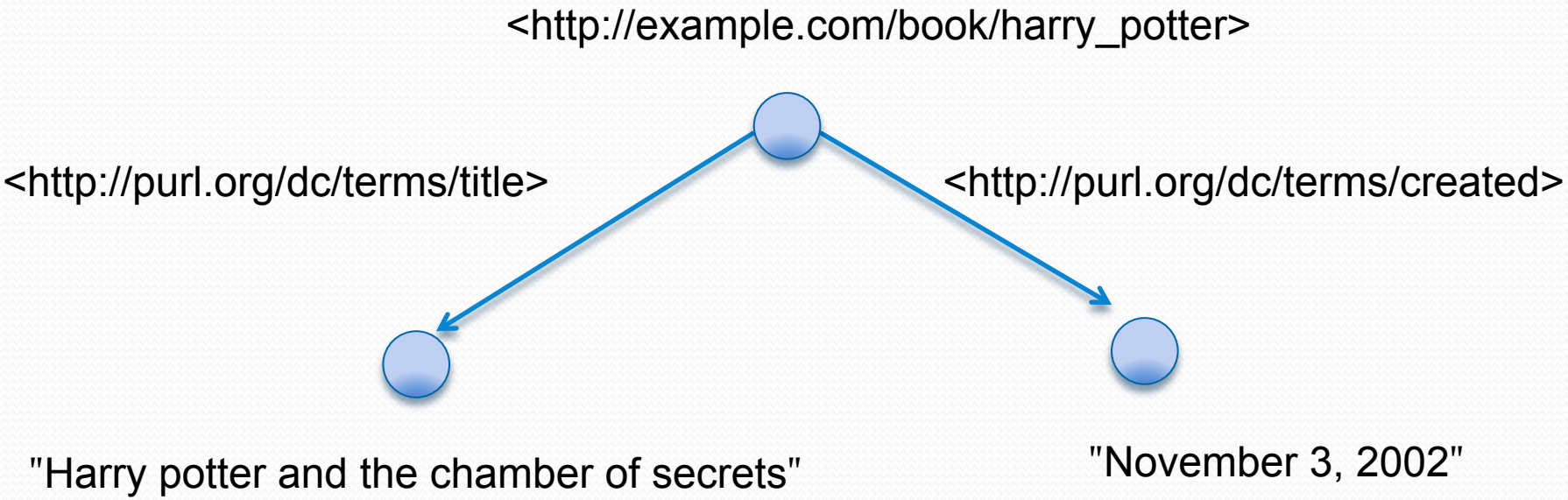
RDF properties defined in Dublin Core



The tag is used to group inline-elements in a document. It provides no visual change by itself.

Hints on web pages (2)

- Markup HTML text to make it machine readable using Dublin Core vocabulary



Using URLs as identifiers, RDFa provides a solid way of disambiguating vocabulary terms.

Links with Flavor

- Mark up the hyperlinks in a machine-readable way, to express the type of link being described

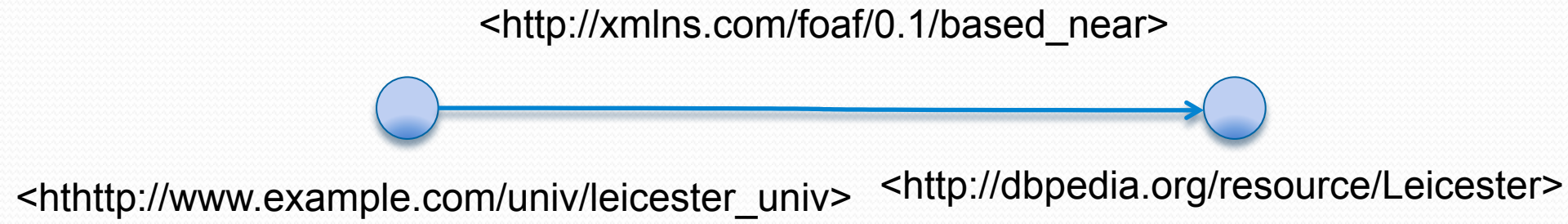
http://www.example.com/univ/leicester_univ

```
<p> Leicester University is located in
  <a property="http://xmlns.com/foaf/0.1/based_near"
        href="http://dbpedia.org/resource/Leicester">
    Leicester</a>
</p>
```

Links with Flavor

- RDFa lets the publisher add a "flavor", i.e., a label, to an existing clickable link that processors can understand.

```
<p> Leicester University is located in
  <a property="http://xmlns.com/foaf/0.1/based_near"
        href="http://dbpedia.org/resource/Leicester">
    Leicester</a>
</p>
```



Setting a Default Vocabulary

- Sometimes, a webpage will predominantly use a single vocabulary
- RDFa introduces the `vocab` attribute to let the author declare a single vocabulary for a chunk of HTML.

```
…
<body vocab="http://purl.org/dc/terms/">
...
  <h2 property="title">Harry potter and the chamber of secrets</h2>
  <p>Date: <span property="created">November 3, 2002</span></p>
  …
</body>
…
```

Describing Multiple Items per Page

- RDFa provides **resource**, an attribute for specifying the different entries.

```
<body vocab="http://purl.org/dc/terms/">
  ...
  <div resource="/library/book/semantic_web_primer">
    <h2 property="title">Semantic Web Primer</h2>
    ...
  </div>
  ...
  <div resource="/library/book/programming_the_semantic_web">
    <h2 property="title">Programming the Semantic Web</h2>
    ...
  </div>
  ...
</body>
```

* tag is a container unit that encapsulates other page elements and divides the HTML document into sections*

RDFa Example: Original HTML source code

```
<div>
  <ul>
    <li>
      <a href="http://example.com/bob/">Bob</a>
    </li>
    <li>
      <a href="http://example.com/eve/">Eve</a>
    </li>
    <li>
      <a href="http://example.com/manu/">Manu</a>
    </li>
  </ul>
</div>
```

Using *typeof* attribute

To describe rdf:type relations

```
<div vocab="http://xmlns.com/foaf/0.1/">
  <ul>
    <li typeof="Person">
      <a href="http://example.com/bob/">Bob</a>
    </li>
    <li typeof="Person">
      <a href="http://example.com/eve/">Eve</a>
    </li>
    <li typeof="Person">
      <a href="http://example.com/manu/">Manu</a>
    </li>
  </ul>
</div>
```

Adding relevant properties

```
<div vocab="http://xmlns.com/foaf/0.1/">
  <ul>
    <li typeof="Person">
      <a property="homepage" href="http://example.com/bob/">Bob</a>
    </li>
    <li typeof="Person">
      <a property="homepage" href="http://example.com/eve/">Eve</a>
    </li>
    <li typeof="Person">
      <a property="homepage" href="http://example.com/manu/">Manu</a>
    </li>
  </ul>
</div>
```

Adding relevant properties (cont.)

```
<div vocab="http://xmlns.com/foaf/0.1/">
  <ul>
     <li typeof="Person">
       <a property="homepage" href="http://example.com/bob/">
          <span property="name">Bob</span></a>
     </li>
     <li typeof="Person">
       <a property="homepage" href="http://example.com/eve/">
          <span property="name">Eve</span></a>
     </li>
     <li typeof="Person">
       <a property="homepage" href="http://example.com/manu/">
         <span property="name">Manu</span></a>
     </li>
   </ul>
</div>
```

Annotating Contact Information

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

Adding Internal References

```
<div vocab="http://xmlns.com/foaf/0.1/" typeof="Person">
  <p> <span property="name">Alice Birpemswick</span>,
   Email: <a property="mbox" href="mailto:alice@example.com">
   alice@example.com</a>,
   Phone: <a property="phone" href="tel:+1-617-555-7332">
   +1 617.555.7332</a></p>
  <ul>
    <li property="knows" typeof="Person">
      <a property="homepage" href="http://example.com/bob/">
      <span property="name">Bob</span></a>
    </li>
    <li property="knows" typeof="Person">
      <a property="homepage" href="http://example.com/eve/">
      <span property="name">Eve</span></a>
    </li>
    <li property="knows" typeof="Person">
      <a property="homepage" href="http://example.com/manu/">
      <span property="name">Manu</span></a>
    </li>
  </ul>
</div>
```

rel vs. *property*

- Using prefixes, the vocabulary elements can be abbreviated.
- **property** must be linked to a target or some textual content while **rel** never considers the textual content of an element if no clear target has been specified for a link via, e.g., a resource or an href attribute, it will go “down” and find one or more targets in the hierarchy and use those.
  - e.g. the knows attribute on the ul element does not include any obvious target; however, the processor finds those in the individual li elements and will use those. This pattern is typical for the usage of rel.

Setting multiple properties using *rel* attribute

```
<div vocab="http://xmlns.com/foaf/0.1/" typeof="Person">
  <p> <span property="name">Alice Birpemswick</span>,
   Email: <a property="mbox" href="mailto:alice@example.com">
   alice@example.com</a>,
   Phone: <a property="phone" href="tel:+1-617-555-7332">
   +1 617.555.7332</a></p>
  <ul rel="knows">
    <li typeof="Person">
      <a property="homepage" href="http://example.com/bob/">
      <span property="name">Bob</span></a>
    </li>
    <li typeof="Person">
      <a property="homepage" href="http://example.com/eve/">
      <span property="name">Eve</span></a>
    </li>
    <li typeof="Person">
      <a property="homepage" href="http://example.com/manu/">
      <span property="name">Manu</span></a>
    </li>
  </ul>
</div>
```

Using *resource*

- RDFa provides resource, an attribute for specifying the "context"
- The value of resource could be absolute URI or relative URI

```
...
 <div resource="http://example.com/people/alice"
      typeof="http://www.cs.le.ac.uk/rdf#Person">
      <h2 property="http://www.cs.le.ac.uk/rdf#name">
       Alice
     </h2>
      ...
</div>…
```

Multiple Vocabularies Using Prefix

- Using prefixes, the vocabulary elements can be abbreviated.

```
...
<body prefix="foaf: http://xmlns.com/foaf/0.1/ uol: http://www.cs.le.ac.uk/rdf">
  <div resource="http://example.com/people/alice" typeof="uol:Person">
    <h2 property="foaf:name">Alice</h2>
    ...
    <h3 property="uol:hasFriend" resource="#johnsmith">John Smith</h3>
    <div property="uol:hasWelcomeMessage">
      <p>Hello, welcome to my homepage</p>
    </div>
   ...
  </div>
 </body>
...
```

RDFa Resources

- **RDFa** Tools
  - http://rdfa.info/tools/
- A realtime RDFa Editor
  - http://rdfa.info/play/
- A Complete RDFa Example
  - https://www.w3.org/2010/02/rdfa/sources/rdfa-primer/alice-example.html

Microdata

Microdata

- Microdata is an attempt to provide a simpler way of annotating HTML elements with machine-readable tags than the similar approaches of using RDFa and Microformats.
  - HTML5 draft specification includes Microdata (W3C HTML Working Group later decided to publish HTML Microdata as a WG Note in 2013)

Microdata Attributes

- Microdata introduces five simple global attributes (available for any element to use) which give context for machines about your data. These five new attributes are:
  - **Itemscope**
  - **itemprop**
  - **Itemtype**
  - **itemid**
  - **itemref**
- Most developers will only ever use itemscope, itemtype and itemprop.

Microdata Attributes

- At a high level, microdata consists of a group of name-value pairs called **items**, and each name-value pair is a **property**. Items and properties are represented by regular elements.

Microdata Attributes

- **itemscope**: creating the items
- **itemprop**: adding a property to an item (used on one of the item's descendants).
- **itemtype**: specifying the type for an item
- **itemid**: associated an item with a global identifier
- **itemref**: adding a property to items that are not descendants of the element.

Itemscope

- Two items, each of which has the property "name" and "person-id"

```
<div itemscope>
 …

</div>

<div itemscope>
 ….
</div>
```

Itemprop

- Using **itemscope** and **itemprop**
- Two items, each of which has the property "name" and "person-id"

```
<div itemscope>
 <p>My name is <span itemprop="name">Elizabeth</span>.</p>

</div>

<div itemscope>
 <p>My name is <span itemprop="name">Daniel</span>.</p>
</div>
```

Itemprop

- When a string value is in some machine-readable format unsuitable for human consumption, it is expressed using the value attribute of the data element

```
<div itemscope>
 <p>My name is <span itemprop="name">Elizabeth</span></p>
 <data itemprop="person-id" value=STU001">
      Some private data.
</data>
</div>
<div itemscope>
 <p>My name is <span itemprop="name">Daniel</span>.</p>
 <data itemprop="person-id" value=STU002">
      Some private data.
</data>
</div>
```

itemtype

- Itemtype can be used to give each item a type.

- The type for an item is given as the value of an itemtype attribute on the same element as the itemscope attribute

```
<div itemscope
            itemtype="http://www.cs.le.ac.uk/rdf#Person">
 <h1 itemprop="name">John Smith</h1>
 <p itemprop="description"></p>
 <img itemprop="img" src="selfie.jpeg" alt="" title="John
Smith taking selfie">
</div>
```

itemid

- Itemid can be used to associate an item with a global identifier
- For example, books can be identified by their ISBN number.

```
<div itemscope
     id="johnsmith"
     itemtype=http://www.cs.le.ac.uk/rdf#Person >
 <h1 itemprop="name">John Smith</h1>
 <p itemprop="description"></p>
 <img itemprop="img" src="selfie.jpeg" alt="" title="John
Smith taking selfie">
</div>
```

itemref

- **itemref**: can be used to associate the item with the properties that are not descendants of the element with the itemscope attribute

```
<div itemscope id="johnsmith" itemref="a b"></div>
<p id="a">Name: <span itemprop="name">John Smith</span>
</p>
<div id="b" itemprop="location" itemscope itemref="c"></div>
<div id="c">
 <p>Address: <span itemprop="street">University Road,
Leicester</span>
</p>
 <p>Postcode: <span itemprop="postcode">LE1 7RH</span>
</p>
</div>
```

A Complete Example

```
<section itemscope itemtype="http://xmlns.com/foaf/0.1/Person">
    Hello, my name is
    <span itemprop="name">John Smith</span>,
    I am a
    <span itemprop="title">Professor</span>
    at the
    <span itemprop="affiliation">University of Leicester</span>.
    My homepage is
    <a href="http://www.cs.le.ac.uk/people/johnsmith" itemprop="homepage">
        http://www.cs.le.ac.uk/people/johnsmith</a>.
    <section itemprop="place" itemscope
            itemtype="http://example.com/Location">
        My address is
        <span itemprop="street">University Road, Leicester</span>,
        <span itemprop="postcode">LE1 7RH</span>,
    </section>
</section>
```

Microdata References

- https://www.w3.org/TR/microdata/
- http://schema.org/docs/gs.html