Structured Data on the Web

Introduction to Web Science

Steffen Staab
Agenda

- HTML Meta
- Microdata
- Microformats
- RDF
  - RDF-a

- Applications
  - Schema.org
  - Wikidata
  - Open data (especially Open government data)
The Semantic Tower of Babel

XML

XHTML, HTML5

Microformats

Microdata

RDFa

GRDDL

JSON
MICROFORMATS
A microformat (sometimes abbreviated μF) is a web-based approach to semantic markup

- re-use existing HTML/XHTML tags to convey metadata
- other attributes
  - in web pages and
  - in other contexts that support (X)HTML, such as RSS.
- Predefined classes
  - hcard, hcal, ...
- Example

```html
<ul class="vcard">
  <li class="fn">Joe Doe</li>
  <li class="org">The Example Company</li>
  <li class="tel">604-555-1234</li>
  <li><a class="url" href="http://example.com/">http://example.com/</a></li>
</ul>
```
Welcome to the microformats wiki!

microformats are extensions to HTML for marking up people, organizations, events, locations, blog posts, products, reviews, resumes, recipes etc. Sites use microformats to publish a standard API that is consumed and used by search engines, aggregators, and other tools. See what-are-microformats for more.

Introduction

Main article: introduction

This wiki is the central resource of the microformats community and provides microformats authoring guides, references, specifications, drafts, publishing patterns, research, brainstorming, and issue tracking.

Get Started

Main article: get-started

Get started with microformats:

- First, make sure your pages and web apps use semantically rich, descriptive, ‘POSH’ HTML.
- Mark-up your contact info with hCard (see authoring tips, examples) and link from your personal site to other profiles with rel-me.
- Questions? Read the frequently asked questions, ask on the IRC channel.
Trade-offs

+ maximal re-use of HTML tags

- lack of global identifiers for types
- lack of global identifiers for properties
- lack of global identifiers for objects
HTML MICRODATA
HTML Microdata

Working draft: http://dev.w3.org/html5/md/

- machine-readable data to be embedded in HTML documents
  - in an easy-to-write manner
  - an unambiguous parsing model.

- compatible with other data formats including RDF and JSON.
<section>
Hello, my name is John Doe, I am a graduate research assistant at the University of Dreams. My friends call me Johnny.
You can visit my homepage at 
</section>
Hello, my name is John Doe, I am a graduate research assistant at the University of Dreams. My friends call me Johnny. You can visit my homepage at www.JohnnyD.com. I live at 1234 Peach Drive, Warner Robins, Georgia.
Google rich snippet testing tool

---

**Item**

- **Type:** [http://data-vocabulary.org/Person](http://data-vocabulary.org/Person)
- **name:** John Doe
- **title:** graduate research assistant
- **affiliation:** University of Dreams
- **nickname:** Johnny
- **url:** [http://www.johnnyd.com/](http://www.johnnyd.com/)
- **address:** Item(1)

**Item 1**

- **Type:** [http://data-vocabulary.org/Address](http://data-vocabulary.org/Address)
- **street-address:** 1234 Peach Drive
- **locality:** Warner Robins
- **region:** Georgia
Trade-offs

+ URIs for Types
+ Microdata DOM API
+ standardized conversion to JSON

- no URIs for objects
- no URIs for properties
Java-script Object Notation

JSON
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
RDF
RDF

- RDF is a graph data format

- Different syntaxes to write down graphs
  - Turtle
  - XML/RDF
  - RDFa
RDF Model

- Resources (Subject, Object) connected by Predicates (relationships)

Subject \(\xrightarrow{\text{predicate}}\) Object


- **Resources**
  - A resource is a referenced entity (Class, Individual, Relationship, …)
  - Resources *must* have
    - URIs – Uniform Resource Identifiers or
    - IRIs - Internationalized Resource Identifiers
### RDF Model

#### Resource
- Resource is a referenced entity (Class, Object, Entity, Relationship, …)
- Resource **must** have:
  - URIs – Uniform Resource Identifiers *or*
  - IRIs - Internationalized Resource Identifiers

#### Property (relationship)
- Similar to association in UML or relationship in database
- Relationships between Resources and other Resources, or Resources to Literals
- Property is also a Resource (have URI)

#### Literal
- Simple (atomic) data type (e.g. String, int …)

#### Statements
- “Resource has Property with Value”
- Format: **Subject –[Property]→ Object**
- Resources and/or literals are included in statement
Statement example

- Statement
  - “Resource [http://west.uni-koblenz.de/#west](http://west.uni-koblenz.de/#west) has name Institute WeST”

- Structure
  - Resource (subject) [http://west.uni-koblenz.de/#west](http://west.uni-koblenz.de/#west)
  - Property (predicate) [http://west.uni-koblenz.de/#hasName](http://west.uni-koblenz.de/#hasName)
  - Value (object) “Institute WeST”

- Related Graph
  
  ![Related Graph Diagram]

  - [http://west.uni-koblenz.de/#west](http://west.uni-koblenz.de/#west)
  - [http://west.uni-koblenz.de/#hasName](http://west.uni-koblenz.de/#hasName)
  - Institute WeST
RDF represented as graphs

- **Nodes:**
  - Resources represented by URIs
  - Unnamed Resources (Blank Nodes)
  - Literals represented by Strings

- **Directed Edges:**
  - Represented by URIs

Diagram:
- Node 1: `http://west.uni-koblenz.de/#west`
- Node 2: `http://www.uni-koblenz.de/~staab/#me`
- Node 3: `http://west.uni-koblenz.de/preds/hasName`
- Node 4: `http://west.uni-koblenz.de/preds/hasEmployee`
- Node 5: Institute WeST
Example: Turtle notation

Turtle

<http://west.uni-koblenz.de/#west>
<http://west.uni-koblenz.de/preds/hasEmployee>
<http://www.uni-koblenz.de/~staab/#me>

Turtle with Namespaces

@prefix s <http://west.uni-koblenz.de/preds/>

<http://west.uni-koblenz.de/#west>
s:hasEmployee <http://www.uni-koblenz.de/~staab/#me>

http://west.uni-koblenz.de/ #WeST
http://www.uni-koblenz.de/~staab/#me
Institute WeST
s:hasName
s:hasEmployee
Example: Turtle notation (cont’d)

@prefix s <http://west.uni-koblenz.de/preds/> 
<http://west.uni-koblenz.de/#west> s:hasEmployee <http://www.uni-koblenz.de/~staab/#me> . 
<http://west.uni-koblenz.de/#west> s:hasEmployee <http://www.uni-koblenz.de/~sizov/#me> . 
<http://west.uni-koblenz.de/#west> s:hasEmployee <http://www.uni-koblenz.de/~groener/#me> . 
<http://west.uni-koblenz.de/#west> s:hasName “Institute WeST”

Shorter version

@prefix s <http://west.uni-koblenz.de/preds/> 
@prefix u <http://www.uni-koblenz.de/> 
<http://west.uni-koblenz.de/#west> s:hasEmployee u:~staab/#me; 
  s:hasEmployee u:~sizov/#me; 
  s:hasEmployee u:~groener/#me; 
  s:hasName “Institute WeST”.

Even shorter

@prefix s <http://west.uni-koblenz.de/preds/> 
@prefix u <http://www.uni-koblenz.de/> 
<http://west.uni-koblenz.de/#west> 
  s:hasEmployee u:~staab/#me, u:~sizov/#me, u:~janik/#me; 
  s:hasName “Institute WeST”.
RDFA
- RDFa 1.1 Primer
  - Rich Structured Data Markup for Web Documents
  - W3C Working Group Note 07 June 2012

- RDFa Core 1.1
  - Syntax and processing rules for embedding RDF through attributes
  - W3C Recommendation 07 June 2012
RDFa example

Example

```html
<html>
<head>
  ...
</head>
<body>
  ...
  <h2>The Trouble with Bob</h2>
  <p>Date: 2011-09-10</p>
  ...
</body>
```

Example

```html
<html>
<head>
  ...
</head>
<body>
  ...
  <h2 property="http://purl.org/dc/terms/title">The Trouble with Bob</h2>
  <p>Date: <span property="http://purl.org/dc/terms/created">2011-09-10</span></p>
  ...
</body>
```
Defining and mixing vocabularies

Example

```html
<html>
<head>
...
</head>
<body vocab="http://purl.org/dc/terms/">
...
<h2 property="title">The Trouble with Bob</h2>
<p>Date: <span property="created">2011-09-10</span></p>
...
<p vocab="http://creativecommons.org/ns#">All content on this site is licensed under
   <a property="license" href="http://creativecommons.org/licenses/by/3.0/">a Creative Commons License</a>. ©2011 Alice Birpemswick.</p>
</body>
</html>
```
Example

```xml
<body vocab="http://purl.org/dc/terms/">
  ...
  <div resource="/alice/posts/trouble_with_bob">
    <h2 property="title">The trouble with Bob</h2>
    <p>Date: <span property="created">2011-09-10</span></p>
    <h3 property="creator">Alice</h3>
    ...
  </div>
  ...
  <div resource="/alice/posts/jos_barbecue">
    <h2 property="title">Jo's Barbecue</h2>
    <p>Date: <span property="created">2011-09-14</span></p>
    <h3 property="creator">Eve</h3>
    ...
  </div>
  ...
</body>
```
More complex graph example

```
_:blanknode1
  TYPE foaf:Person
  foaf:phone
  foaf:mailbox <tel:+1-617-555-7332>

_:blanknode3
  foaf:knows
  foaf:mail <mailto:alice@example.com>
  foaf:name "Alice Birpemswick"

_:blanknode2
  foaf:name "Bob"
  foaf:homepage <http://example.com/bob>

_:blanknode4
  foaf:name "Manu"
  foaf:homepage <http://example.com/Manu>
```

```
_:blanknode1
  foaf:knows
  foaf:homepage <http://example.com/eve>
  foaf:name "Eve"

_:blanknode2
  foaf:homepage

_:blanknode4
  foaf:homepage

_:blanknode3
```
Tools für RDFa

- http://www.w3.org/2001/sw/wiki/RDFa
GRDDL
- Gleaning Resource Descriptions from Dialects of Languages (GRDDL)

- W3C Recommendation 11 September 2007

- Input: XML
- Output: RDF

- Examples:
  - microformat to RDF
COMPARISON OF STRUCTURED DATA APPROACHES
The Semantic Tower of Babel

XML

XHTML, HTML5

Microformats

Microdata

RDFa

GRDDL

JSON
## Comparison (in flux)

<table>
<thead>
<tr>
<th>Feature</th>
<th>RDFa 1.1</th>
<th>Microdata 1.0</th>
<th>Microformats 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Complexity</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Data Model</td>
<td>Graph</td>
<td>Tree</td>
<td>Tree</td>
</tr>
<tr>
<td>Item optionally identified by IRI</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Item type optionally specified by IRI</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Item properties specified by IRI</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multiple objects per page</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Overlapping objects</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Plain Text properties</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>IRI properties</td>
<td>Yes*</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Typed Literal properties</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>XML Literal properties</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Language tagging</td>
<td>Yes</td>
<td>Yes</td>
<td>Inconsistent</td>
</tr>
<tr>
<td>Override text and IRI content</td>
<td>Yes</td>
<td>No</td>
<td>Text only</td>
</tr>
<tr>
<td>Clear mapping to RDF</td>
<td>Yes</td>
<td>Problematic</td>
<td>No</td>
</tr>
<tr>
<td>Target Languages</td>
<td>8 (XHTML1, HTML4, HTML5, XHTML5, XML, SVG, ePub, OpenDocument)</td>
<td>2 (HTML5, XHTML5)</td>
<td>4 (XHTML1, HTML4, HTML5, XHTML5)</td>
</tr>
<tr>
<td>New Attributes</td>
<td>8 about, datatype, profile, prefix, property, resource, typeof, vocab</td>
<td>5 itemid, hemprop, itemsref, itemscope, hemtype</td>
<td>0</td>
</tr>
<tr>
<td>Re-used Attributes</td>
<td>5 content, href, rel, rev, src</td>
<td>5 content, src, href, data, datetime</td>
<td>4 class, title, rel, href</td>
</tr>
<tr>
<td>Multiple IRI types per object</td>
<td>Yes</td>
<td>RDF only</td>
<td>No</td>
</tr>
<tr>
<td>Multiple statements per element</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>“Locally scoped” vocabulary terms</td>
<td>Yes, via vocab</td>
<td>Yes, via itemscope</td>
<td>No</td>
</tr>
<tr>
<td>Item Chaining</td>
<td>Yes</td>
<td>Basic</td>
<td>No</td>
</tr>
<tr>
<td>Transclusion</td>
<td>No</td>
<td>Yes</td>
<td>Yes, via include pattern</td>
</tr>
<tr>
<td>Compact IRIs</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prefix rebinding</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Vocabulary Mashups</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>HTML5 time element support</td>
<td>Not yet</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Different attributes for different property types</td>
<td>Yes property for text, rel/rev for URLs, resource/content for overrides</td>
<td>Yes class for text and rel for URLs</td>
<td>Yes class for text and rel for URLs</td>
</tr>
<tr>
<td>Transform to JSON</td>
<td>Yes (RDFa API)</td>
<td>Yes (Parser and Microdata DOM API)</td>
<td>No</td>
</tr>
<tr>
<td>DOM API</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Unified Parser</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Conclusion

- Big common core (triples)
- Too many things to do almost the same

- Methods to do it the Web style
  - using URIs all over the place
  - mixing vocabularies

should be preferred