

Web Semantics in Practice

Web Architecture and Information Management [./] Spring 2009 — INFO 190-02 (CCN 42509)

Erik Wilde, UC Berkeley School of Information

2009-04-29



<http://creativecommons.org/licenses/by/3.0/>

This work is licensed under a [CC Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/) [http://creativecommons.org/licenses/by/3.0/]

Contents

• Abstract	2
• Using Semantics	3
• 1 Microformats	
◦ HTML "Microformats"	5
◦ vCard Microformat (hCard)	6
◦ Location Microformats	7
◦ Islands of Semantics	8
◦ Microformat Syntax	9
◦ Magic Names	10
◦ Microformats on the Web	11
• 2 Resource Description Framework (RDF)	
◦ Describing Resources	13
◦ RDF Graphs	14
◦ RDF is Simple and Complex	15
◦ RDF Schema Graph	16
◦ RDF in Attributes (RDFa)	17
• 3 More Languages	
◦ SPARQL	19
◦ Web Ontology Language (OWL)	20
• Semantics are Important and Hard	21

Abstract (2)

Web semantics are interesting to be able to know more about the *meaning* of Web content, not only its syntactic representation. *Microformats* and more formal approaches such as the *Resource Description Framework (RDF)*, *RDF in Attributes (RDFa)*, and the *Web Ontology Language (OWL)* can be used to describe Web content semantically. After looking at Semantic Web concepts such as *Microformats* and the *Resource Description Framework (RDF)*, we look into some practical issues of how to express semantics on the Web.

Using Semantics (3)

- Expressing semantics is not easy, why should anybody do it?
- 1. Findability of resources
 - search engines can do a better job if they understand more
 - machine-readable semantics are more reliable than text processing
- 2. Usability of resources
 - browsers/devices can pick up information and use it
 - more understanding allows more sophisticated applications
 - in particular, recombining things is easier when you understand them
- 3. Utility of resources
 - recombining things is easier when you understand them
 - understanding is much easier in a “closed user group”

Microformats

HTML "Microformats" (5)

```
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Erik Wilde</title>
  <link rel="next" href="cv" title="CV"/>
  <link rel="prev" href="publications" title="Publications"/>
  <link rel="glossary" href="http://dret.net/glossary"/>
  <link rel="stylesheet" type="text/css" href="http://dret.net/dretnet.css"/>
  <link rel="stylesheet" type="text/css" href="http://dret.net/netdret
/netdret.css"/>
  <link rel="meta" type="application/rdf+xml" title="FOAF"
href="http://dret.net/netdret/foaf.rdf"/>
  <link rel="icon" href="http://dret.net/favicon.ico" type="image/x-icon"/>
  <link rel="shortcut icon" href="http://dret.net/favicon.ico" type="image/x-
icon"/>
```

- HTML has some built-in "microformats"
 - support for attaching semantics to links
 - structured documents can be made explicit on the Web

vCard Microformat (hCard) (6)

```
<div class="vcard">
  <span class="fn">Erik wilde</span>
  <i class="email"><a href="mailto:dret@berkeley.edu" title="Send email
to dret@berkeley.edu">dret@berkeley.edu</a></i>
  <i class="tel"><a href="tel:+1-510-2061079" title="Call +1-510-2061079"
id="mobile">+1-510-2061079</a></i> /
  <i class="tel"><a href="tel:+1-510-6432253" title="Call +1-510-6432253"
id="tel">+1-510-6432253</a></i> /
```

Location Microformats

(7)

```
<p><em>Geolocation</em> (GPS coordinates), for example
<span class="geo">Grizzly Peak:
  <span class="latitude">37.882045</span>,
  <span class="longitude">-122.233698</span>
</span>
</p>
<hr />
<p><em>Address information</em> (requires geocoding), for example
<span class="adr">School of Information:
  <span class="extended-address">311 South Hall</span>,
  <span class="street-address">South Hall Road</span>,
  <span class="region">Berkeley</span>
  <span class="region">CA</span>
  <span class="postal-code">94720-4600</span>,
  <span class="country-name">USA</span>
</span>
</p>
```

Islands of Semantics

(8)

- Microformats solve very specific problems in a very specific way
 - encoding address information on a Web page
 - encoding a location of something represented by a Web resource
- Microformats can be compared to “tagging”
 - a very simple mechanism with a minimal barrier-to-entry
 - little flexibility in adapting the mechanism to slightly other uses
 - often underspecified and interpretation implementation-dependent
 - no unified rules across different platforms which makes processing hard
 - nice and easy to start with, but questionable for robust long-term solutions
- Currently there are about 10 reasonably popular microformats
 - [calendar entries](#), [addresses](#), [licenses](#), [outlines](#), [geolocation](#), [resumes](#), [social networking](#), ... [http://microformats.org/wiki/Main_Page]

Microformat Syntax (9)

- HTML has some underspecified and underused elements
 - dfn, code, samp, kbd, var, cite, abbr, acronym
 - they can be reused and augmented with additional information
- HTML allows non-HTML content in HTML pages
 - unknown elements and attributes must be ignored
- HTML allows class attributes to carry semantics
- HTML has a head which contains page metadata
 - for example, the link element specifies connections to other resources

Magic Names (10)

- A syntax defines where and how to embed information
 - what is embedded and how well is it defined semantically?
 - is there an underlying model for specifying dependencies?
 - how many assumptions does it take to implement a microformat?
- Names are never self-explanatory, they always represent concepts
 - nothing can remove the burden of defining a conceptual model
 - if this is not done, models evolve and there will be more than one
- “Microformats” and “tagging” share the same folklore
 - define simple things and good things will happen
 - this works by supporting a quickly growing ecosystem of diverging semantics
 - semantics are most useful when they are well-defined
 - loose semantics also have some utility

Microformats on the Web (11)

- Easy to embed for generated content
 - some of the very basic formats may even appear in browsers one day
 - combining well-designed URIs with document relationships is better than every site map
- Hard to rely on for applications that need dependable semantics
 - useful as a hint and as a starting point
 - microformats are not a good idea for complex information management tasks
- Use as foundation for representing common concepts
 - when formatting addresses, use [adr](http://microformats.org/wiki/adr) [http://microformats.org/wiki/adr] class names
 - for structured documents use [XOXO](http://microformats.org/wiki/xoxo) [http://microformats.org/wiki/xoxo]

Resource Description Framework (RDF)

Describing Resources (13)

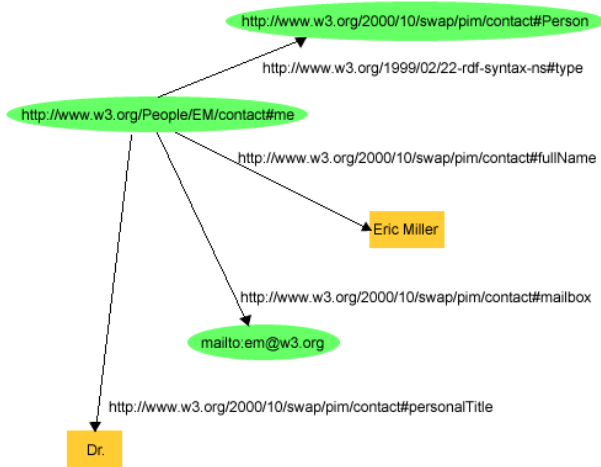
- RDF describes everything in *triples*
 - making a statement about a *resource* (identified by a [URI](#) [Web Foundations (URI and HTTP); Uniform Resource Identifier (URI) (1)])
 - describing a certain *property* of the resource
 - specifying a *value* for that property

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

[<http://www.w3.org/TR/REC-rdf-syntax/#intro>]

RDF Graphs

(14)



RDF is Simple and Complex

(15)

- **RDF's abstract model** [<http://www.w3.org/TR/rdf-concepts/>] is the idea of descriptive triples
 - the actual RDF model is rooted in *description logic*
 - RDF itself can only describe individuals (something identified by URI)
- **RDF/XML** [<http://www.w3.org/TR/rdf-syntax-grammar/>] is an XML syntax for encoding triples
 - the syntax allows a variety of ways to represent the same RDF statements
 - processing RDF/XML with XML tools is likely to fail
 - use RDF parsers to parse all variations of RDF/XML into an abstract RDF graph
- **RDF Schema** [<http://www.w3.org/TR/rdf-schema/>] supports the creation of *RDF vocabularies*
 - describe the *classes of things* that can be used in statements
 - describe the *properties* which can be used for each of these classes
 - describe the *allowed* values for the supported properties

RDF Schema Graph (16)



RDF in Attributes (RDFa) (17)

- Microformats can use any kind of markup design
 - this makes it hard to detect microformats when processing a Web page
 - combining microformats can become complicated and ill-designed
- RDFa defines only a syntax for embedding metadata into XHTML
 - the vocabulary must be described by some RDF schema language

```
<p>This document is licensed under a <a xmlns:cc="http://creativecommons.org/licenses/" rel="cc:license" href="http://creativecommons.org/licenses/by/nc-nd/3.0/">Creative Commons License</a>.</p>
```

- RDFa uses and extends HTML for embedding RDF
 - it uses HTML's `rel`, `rev`, `href`, and `src` attributes
 - it defines a number of [new attributes for HTML elements](http://www.w3.org/TR/rdfa-syntax/#s_metaAttributes)
 - it defines a [processing model](http://www.w3.org/TR/rdfa-syntax/#s_model) for deriving RDF triples from these attributes

More Languages

SPARQL

(19)

- RDF graphs can be large and hard to handle
 - querying RDF graphs using XML technologies is hard and slow
 - special data structures need special query languages
 - SPARQL is a query language for querying RDF graphs
- Using RDF without using SPARQL does not make a lot of sense
 - if the data is simple and restricted, why use RDF?
 - processing unrestricted RDF without a special language is very hard
- Semantic Web search engines can harvest the Web for RDF
 - the result is a huge graph of RDF describing all semantic Web resources
 - querying into this graph retrieves all formalized semantics on the Web

Web Ontology Language (OWL)

(20)

- RDF and RDF Schema are rather basic languages
- OWL adds more sophisticated features to RDF Schema
 - constructions of classes using existing ones
 - characterize relationships (e.g., whether they are transitive, symmetric, functional, etc.)
- Formal semantics are hard to write and compute
 - no property expressions or datatypes in RDF Schemas
 - not all set operators, restricted cardinality in *OWL Lite*
 - some restrictions, but a computational guarantee in *OWL DL*
 - full expressive power in *OWL Full* (but no computational guarantee)

Semantics are Important and Hard (21)

- Semantics must be captured somewhere
- Most semantic definitions are using prose and some formalism
- Completely formal semantics are hard to define and hard to use
- Semantic Web technologies may share the fate of AI