10 – The Semantic Web and RDF

In the previous episodes...

A (video) summary:

Michael Wesch: "Web2.0... The Machine is Us/ing Us"

http://www.youtube.com/watch?v=6gmP4nk0EOE

Web Semantics

- Describing syntax ("classic" HTML) → Describing data (XML)
- Describing data (XML) → Describing *knowledge* (???)

"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined *meaning*, better enabling computers and people to work in cooperation"

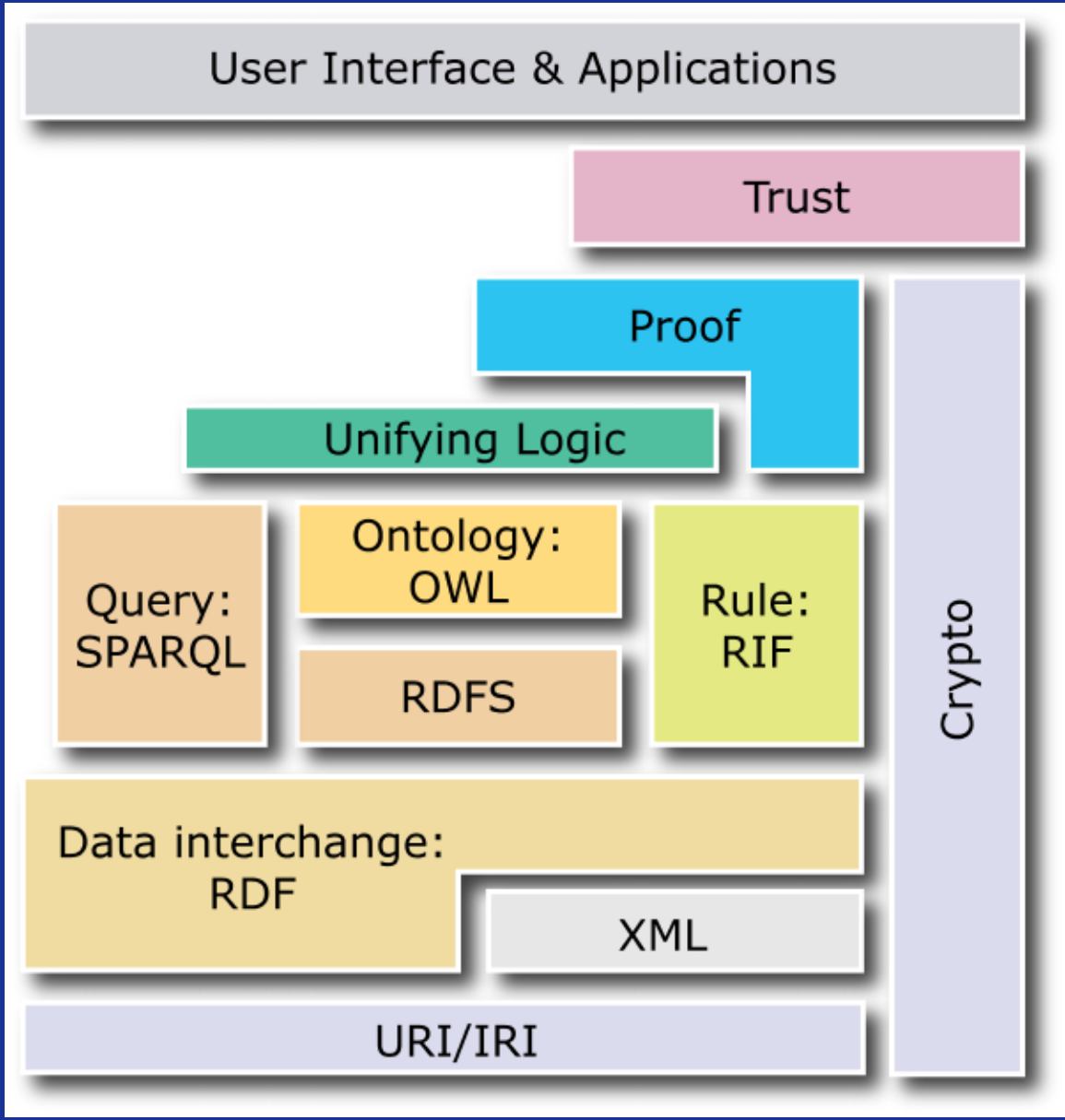
Tim Berners-Lee
The Semantic Web
Scientific American, 2001

The Semantic Web

According to Jim Hendler, two main directions:

- Web
  - Web-based apps with little semantics
  - Emphasis on *linking data* using URIs
  - Standards: RDF (Resource Description Framework) and SPARQL
- Semantic
  - Models to represent knowledge in an expressive way
  - *Inference of new knowledge* by using reasoners
  - Standards: OWL (Web Ontology Language)

The Semantic Web Layer Cake



RDF at a glance

Table 3-1 Tabular Data about Elizabethan Literature and Music

ID	Title	Author	Medium	Year
1	*As You Like It*	Shakespeare	Play	1599
2	*Hamlet*	Shakespeare	Play	1604
3	*Othello*	Shakespeare	Play	1603
4	"Sonnet 78"	Shakespeare	Poem	1609
5	*Astrophil and Stella*	Sir Phillip Sidney	Poem	1590
6	*Edward II*	Christopher Marlowe	Play	1592
7	*Hero and Leander*	Christopher Marlowe	Poem	1593
8	*Greensleeves*	Henry VIII Rex	Song	1525

ROW

1	*As You Like It*	Shakespeare	Play	1599

4	Sonnet 78	Shakespeare	Poem	1609
6	Edward II	Christopher Marlowe	Play	1592

Needs common schema—which column is which?

7	Hero and Leander	Christopher Marlowe	Poem	1593
3	Othello	Shakespeare	Play	1603

COLUMN

Year	Medium
1599	Play
1604	Play
1603	Play
1609	Poem
1590	Poem
1592	Play
1593	Poem
1525	Song

Needs to reference entities—which thing are we talking about?

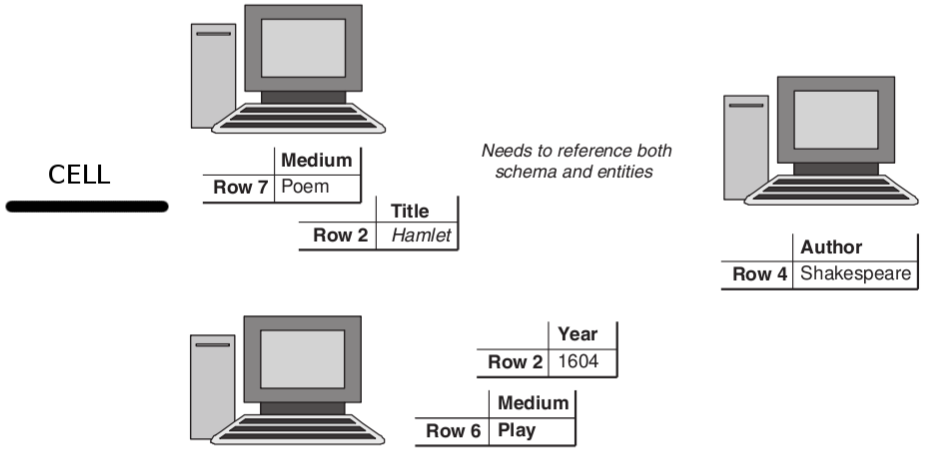
Author
Shakespeare
Shakespeare
Shakespeare
Shakespeare
Sir Phillip Sidney
Christopher Marlowe
Christopher Marlowe
Henry VIII Rex

Title
As You Like It
Hamlet
Othello
"Sonnet 78"
Astrophil and Stella
Edward II
Hero and Leander
Greensleeves

RDF at a glance

Table 3-1 Tabular Data about Elizabethan Literature and Music

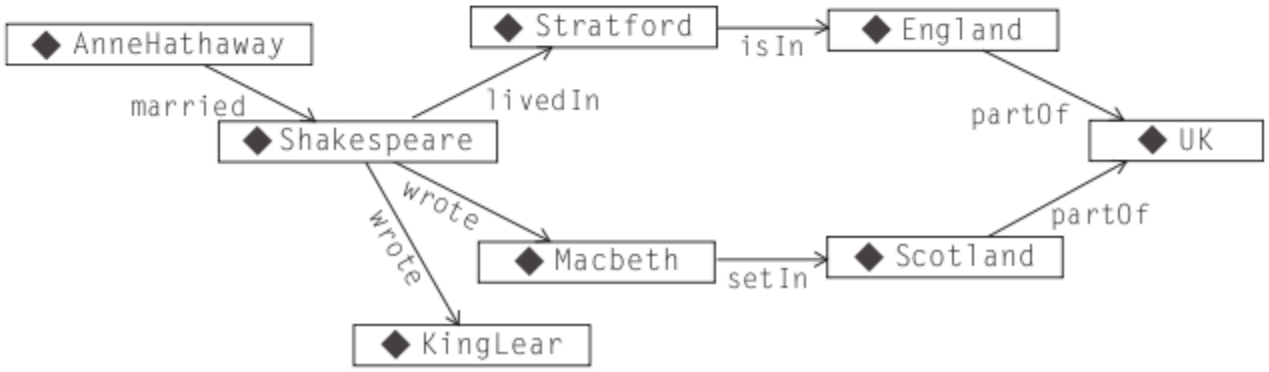
ID	Title	Author	Medium	Year
1	*As You Like It*	Shakespeare	Play	1599
2	*Hamlet*	Shakespeare	Play	1604
3	*Othello*	Shakespeare	Play	1603
4	"Sonnet 78"	Shakespeare	Poem	1609
5	*Astrophil and Stella*	Sir Phillip Sidney	Poem	1590
6	*Edward II*	Christopher Marlowe	Play	1592
7	*Hero and Leander*	Christopher Marlowe	Poem	1593
8	*Greensleeves*	Henry VIII Rex	Song	1525



RDF at a glance

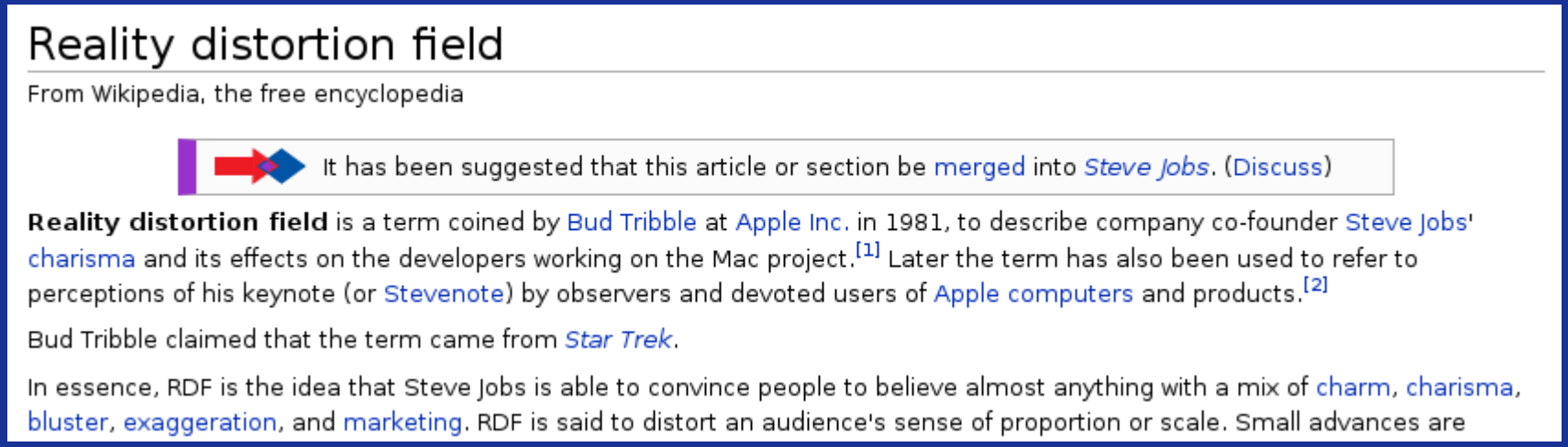
Table 3-3 Sample Triples

Subject	Predicate	Object
Shakespeare	Wrote	King Lear
Shakespeare	Wrote	Macbeth
Anne Hathaway	Married	Shakespeare
Shakespeare	Lived In	Stratford
Stratford	Is in	England
Macbeth	Set in	Scotland
England	Part of	The UK
Scotland	Part of	The UK



RDF

- Not this one



Reality distortion field

From Wikipedia, the free encyclopedia

It has been suggested that this article or section be merged into *Steve Jobs*. (Discuss)

Reality distortion field is a term coined by Bud Tribble at Apple Inc. in 1981, to describe company co-founder Steve Jobs' charisma and its effects on the developers working on the Mac project.[1] Later the term has also been used to refer to perceptions of his keynote (or Stevenote) by observers and devoted users of Apple computers and products.[2]

Bud Tribble claimed that the term came from *Star Trek*.

In essence, RDF is the idea that Steve Jobs is able to convince people to believe almost anything with a mix of charm, charisma, bluster, exaggeration, and marketing. RDF is said to distort an audience's sense of proportion or scale. Small advances are

- Resource Description Framework
  - W3C recommendation (2004)
  - a *Semantic Web specification* together with OWL
- Graph data model
- Abstract syntax based on the concept of *triple*
- Serialization in different text-based formats (including XML)

RDF Triples

- RDF is based on the following ideas:
  - things being described have *properties* which have *values*
  - and resources can be described by making *statements* that specify those properties and values
- These statements are called *triples*:
  - the *Subject* is the resource the statement is about
  - the *Predicate* identifies the property or the characteristic
  - the *Object* identifies the value of the property
- Example:

http://www.example.org/index.html Subject
(has a) **creator** Predicate
(whose value is) **John Smith** Object

Literals, Resources, URIs

- Each element in a triple can belong to two different types:
  - Resource
    - http://www.whatever.com/index.html#me
    - dc:creator
  - Literal
    - Plain: “666”, “English”, “April, 8 2009”
    - Typed: "27"^^xsd:integer, "2009-04-08"^^xsd:date
- Subjects and predicates can only be resources, while objects can be resources or literals
- Resources are identified by Uniform Resource Identifiers (URIs)
  - URLs are a particular kind of URI
  - URI reference = URI + *fragment identifier*
    - *i.e. http://www.example.org/index.html#section2*

Namespaces and prefixes

- As in any XML document, all elements have to belong to a given namespace
  - NOTE: in the XML serialization, properties can become either elements or attributes
- Example (RDF header + CD description):

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

</rdf:RDF>
```

Anonymous nodes

- Look at this address:

```
Students:123456                                  (subj)
students:address                                 (pred)
"765 San Antonio Ave, Palo Alto, CA 94304".      (obj)
```

- What if we want to be able to access the single elements of the address?

```
(subj)                (pred)                 (obj)
students:123456       students:address       studaddrid:654321 .
studaddrid:654321     students:street        "765 San Antonio Ave" .
studaddrid:654321     students:city          "Palo Alto" .
studaddrid:654321     students:state         "CA" .
studaddrid:654321     students:zip           "94304" .
```

- studaddrid:654321 is a *universal* identifier, but we won't need it again in other documents
  - solution is to use a *local*, *anonymous* node which does not need an identifier

```
studaddrid:654321 => _:anon123
```

RDF vs XML

RDF and XML models are fundamentally different

- RDF has a very simple model which consists of *labeled arcs*
- Any specific group of RDF declarations forms a *graph* that can be serialized in XML
- XML data model is a *labeled tree*, which is less flexible for describing metadata

RDF vs XML

Resources used in RDF and XML Schema are different

- In RDF, nodes do not necessarily appear inside the document itself, but could be any resource which has a URI (typically *external*)
- RDF is a language *for metadata*
- The nodes an XML Schema refers to are *internal* to the XML document, in a specific location within the structure of a document

RDF vs XML

The semantics of RDF and XML schemas are different

- RDF schemas have an interpretation which is primarily *semantic*
- RDF is used to build (*model*) *knowledge*, where tree-based representation structures are not sufficient
- XML schemas have an interpretation which is primarily *syntactic*
- XML schemas are used to model *documents*

RDF vs XML

- An example:

- How would you render it in XML?

```
<author>
      <uri>book</uri>
      <name>Bob</name>
</author>
```

```
<document href="book">
    <author>Bob</author>
</document>
```

```
<document>
    <details>
      <uri>href="book"</uri>
      <author>
          <name>Bob</name>
      </author>
      </details>
</document>
```

RDF vs XML

- Meaning is not hardcoded in tag names. What you have here:

```
<document>
    <author>
     <uri>href="book"</uri>
     <details>
         <name>Bob</name>
     </details>
     </author>
</document>
```

is interpreted by a machine as a meaningless text:

```
<v>
    <x>
     <y> a="ppppp"</y>
     <z>
         <w>qqqqq</w>
     </z>
    </x>
</v>
```

... what are the relations between elements, now?

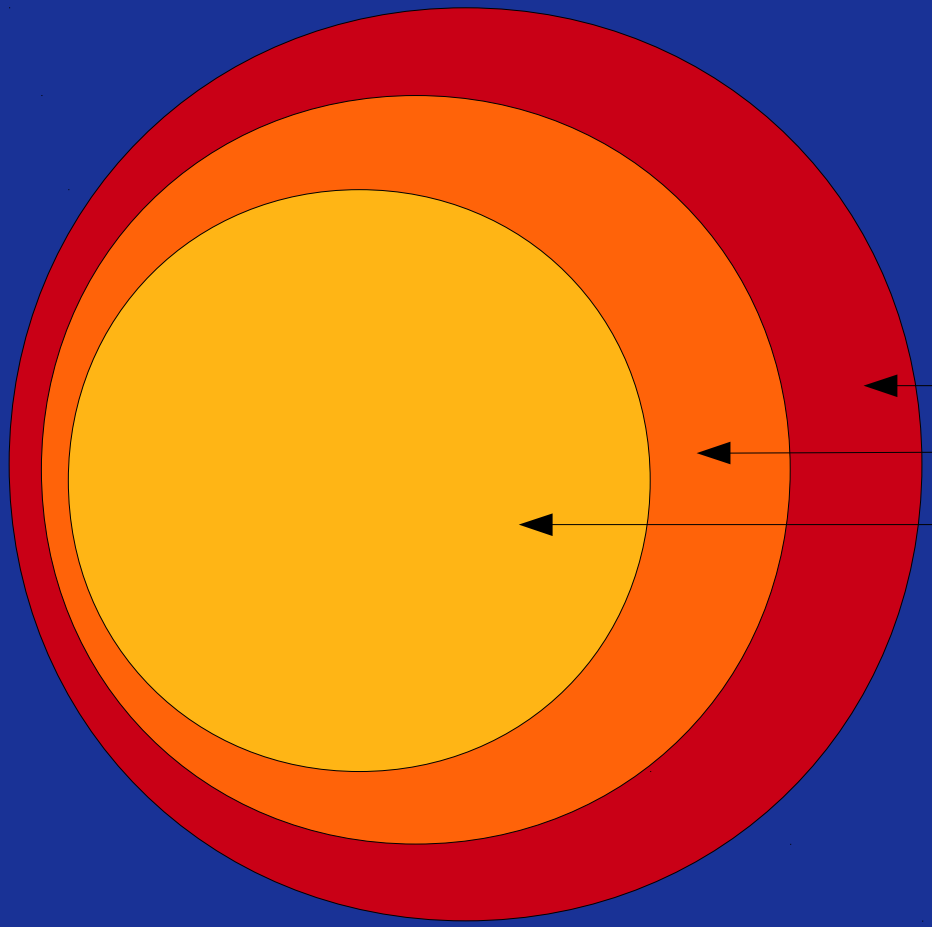
RDF serializations

- RDF/XML

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description
      rdf:about="http://en.wikipedia.org/wiki/Lugano">
    <dc:title>Lugano</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

RDF serializations



- Notation3
- Turtle
- N-Triples

A Notation3 (N3) example:

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
<http://en.wikipedia.org/wiki/Lugano> dc:title "Lugano";
                                      dc:publisher "Wikipedia".
```

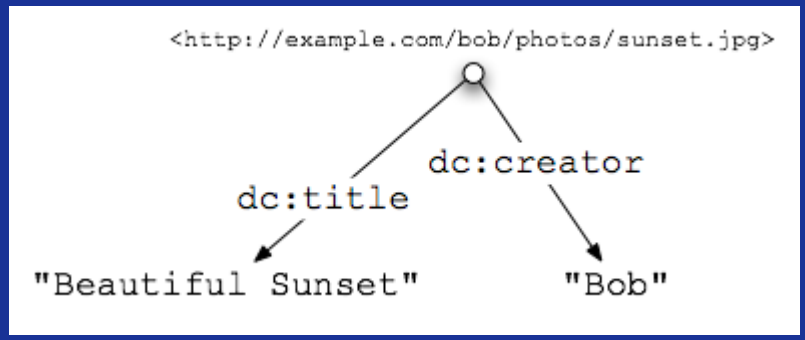
RDF serializations

- RDFa

```
<div about="/alice/posts/trouble_with_bob">
  <h2 property="dc:title">The trouble with Bob</h2>

   The trouble with Bob is that he takes much better photos
   than I do:

  <div about="http://example.com/bob/photos/sunset.jpg">
    <img src="http://example.com/bob/photos/sunset.jpg" />
    <span property="dc:title">Beautiful Sunset</span>
    by <span property="dc:creator">Bob</span>.
  </div>
</div>
```



References

- Some Web references:
  - Why RDF is different from XML: http://www.w3.org/DesignIssues/RDF-XML
  - RDF Primer: http://www.w3.org/TR/REC-rdf-syntax
  - Dean Allemang, Jim Hendler: "Semantic Web for the Working Ontologist". http://workingontologist.org
  - FOAF: http://www.foaf-project.org
  - Dublin Core: http://dublincore.org
- Tools:
  - W3C RDF Validator: http://www.w3.org/RDF/Validator
  - Morla RDF editor: http://www.morlardf.net
  - FOAF-o-matic: http://www.foaf-o-matic.org