Semantic Web

“The Semantic Web”
(Berners-Lee, Handerl, Lassila; Scientific American, May 2001)

Today’s Syntactic Web

- HTML (modest compliance with standards thanks to robust browsers)
- Hyperlinks (no data types; just annotated with text [sometimes merely “Click here!”]; often dangling references)
- Human eyeballs & common sense
- (Just barely?!?) suitable/scalable for
  - “trivial pursuit” information retrieval
    - What’s the capital of Botswana? Will it rain tomorrow? …
  - “mundane” transactions/services
    - Buying a book; Playing a game of chess; …

The Semantic Web

“I have a dream for the Web… and it has two parts.”

-- Tim Berners-Lee

- The first Web enables communication between people
  - The Web shows how computers and networks enable the information space while getting out of the way
- The new Web will bring computers into the action
  - Step 1 -- Describe: put data in machine-understandable form
    - RDF (based on XML)
    - Master list of terms used in a document (RDF schema)
    - Documents mix global standards and local agreed-upon terms (namespaces)
  - Step 2 -- Infer and reason: apply logic inference
    - Operate on partial understanding
    - Answering why
    - Heuristics

Automating people out of the loop

“The bane of my existence is doing things that I know the computer could do for me”

-- D Connolly; W3C guru

Why can’t my online calendar & bank account negotiate with my garage’s to arrange a mutually convenient time & price to repair my leaking tyre?

Science Fiction?

- 150 years ago, the telephone was outrageously sophisticated: “Do you seriously predict that every room in every building will have a small device that you type a few numbers into and you can talk to the person in any other room in any other building in the entire world??!!?”
- 30 years ago, email was outrageously sophisticated: “Do you seriously predict that every person will have a small device that you can type a person’s name into and you can send a private message to any other person in the world, that they can read even on the beach in Tahiti?????”
- 10 years ago, the Web was outrageously sophisticated: “Do you seriously predict that every person will have a device with which they can send their grocery list to the shop and in a few hours the groceries arrive??????”

The Challenge

- All the relevant data is (or soon will be) “on the Web”
- But in a form specialized to human vision/processing,
  - Not automated machine processing

How can my agent find/parse/extract garage’s free times?

Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?

Plus: dozens of constraints: How long will it take to get to the garage? Would I pay extra if they can collect the car? Can they repair the door lock too?
What/where is the Semantic Web?

- **Layered** on top of existing Web.
  - Just like HTTP is built on top of TCP
  - which is on top of IP,
  - which is on top of the data-link layer)

Layer 1: URI

- Everything is a “Resource” (people, books, the attribute “title” of an Amazon “book” object, Web pages, the concept “laziness”, …)
  - Every resource has a unique identifier
  - Called a Uniform Resource Identifier
  - eg, the URI of a Web Page is its URL
  - Eg, the URI of my email address is mailto:nick@ucd.ie
- Owner of object can pick any URI they want as long is it is unique.
  - Oftentimes has “URL-like” syntax but that is purely convention/arbitrary

Layer 2: XML

- “XML is the new ASCII” -- Tim Bray
- Common formatting standard for encoding data.
  - `<book><title>War & Peace</title>...</book>`
  - `<taxonomy id=amazon>`
    - `<concept superclass=thing>book</concept>`
    - `<attribute class=book>title</attribute>`
    - ...
  - `<ontology>`
    - `<match><source from=amazon><title>...</source>
    - `<dest ont=fredhanna>name</dest></match>`
  - Data
  - Meta-Data
  - Meta-Data

XML Namespaces

- An XML documents may use tags defined in more than one XML Schema document
- “Namespace” prefixes (xxx:yyy) are used to unambiguously point to the defining XML Schema document

XML Schema

- An XML Schema document is an XML document that defines a set of XML tags (and how they may be used)

XML:  Tools/Software

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Spy</td>
<td>By far, the most comprehensive editor. Handles XML files, DTD’s, XML files, as well as XSD (XML Schema). Unfortunately only a 30 day trial version. <a href="http://www.xmlspy.com/download.html">http://www.xmlspy.com/download.html</a></td>
</tr>
<tr>
<td>XML Pro</td>
<td>XML Pro is a top-notch XML editor but it doesn’t include as many features as XML Spy. Shareware. <a href="http://www.vervet.com/demo.html">http://www.vervet.com/demo.html</a></td>
</tr>
</tbody>
</table>

You can also validate your XML files by just opening them with IE5.0 or above. It checks if the XML file is well-formed or not, and also validates against a DTD (if specified on the DOCTYPE declaration).

Some nice & short Tutorials on XML/XSL/DTD/XML Schemas can be found at: [www.w3schools.com](http://www.w3schools.com)
Summary of the XML+ NS +XSD Layer
The Power of Simplicity

- Keeps principles of SGML
  - But its spec is thin enough to wave ☺
- "When I designed HTML, I chose to avoid giving it more power than it absolutely needed – a "principle of least power", which I have stuck to ever since. I could have used a language like Knuth’s Tex but..." -- TBL
- Saying "I’m using XML" is like saying "I’m using ASCII"
- Using XSD (XML Schema) makes a lot more sense

Layer 3: RDF

- All data/knowledge/facts/opinions/information is expressed on the Semantic Web as “Resource Description Framework” statements
- Very simple language for making assertions:
  - Triple: (value) (attribute) (object)
    - (nick@ucd.ie) (is email address of) (Nick Kushmerick)
    - (0140444173) (is ISBN number of) (War & Peace)
    - (field 5 of database A) (is a field of type) (postal code)

Everything is XML

- Remember
  (Nick’s Home Page) (is title of) (http://www.cs.ucd.ie/staff/nick)
  is actually encoded as some very ugly XML:

```xml
<?xml version="1.0"?>
.rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dc:"http://purl.org/dc/elements/1.1">  
  <rdf:Description about="http://www.cs.ucd.ie/staff/nick">
    <dc:title>Nick's Home Page</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Layer 4: Ontologies (RDF Schema)

- 3 common RDF attribute-sets for common tasks
- Eg “Dublin Core”
  - defines a few dozen standard attributes for asserting statements about documents: title, author, date, version, format, owner, …
- What if you want to define your own concepts/attributes --
  - RDF Schema = set of RDF tags for defining a new set of RDF tags (no, this isn’t circular)

RDF Schema for Dublin Core Ontology

```xml
<?xml version='1.0'?>
<!DOCTYPE rdf:RDF [  
<!ENTITY rdf:ns "http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
<!ENTITY dc:elements "http://purl.org/dc/elements/1.1"  
<!ENTITY dc:elements "http://purl.org/dc/elements/1.1"  
<!ENTITY dcterms "http://purl.org/dc/terms/"  
<!ENTITY foaf "http://xmlns.com/foaf/0.1/"  
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>
<!ENTITY rdf:Description rdf:about="http://purl.org/dc/things/1.1"/>]
</!ENTITY>
</rdf:RDF>
```

Layer 4½: Mapping Between Ontologies

- Taxonomy Crisis:
  - How can your agent know that my “title” is your “name”?!?
  - How can my agent know that some of your “address” objects are post-boxes, not physical addresses?!?
  - How can my agent know that many Asian first names correspond to Western surnames?
- Semantic Web Solution: Services for translating/mapping between “related” ontologies.
  - Suppose Amazon.com uses Dublin Core (“title”), while Fred Hanna uses it’s own document ontology (“name”).
  - So far … my agent is forced to choose an ontology, or must be carefully crafted to understand both languages
- A better solution:
  - Create independent entities (UniversalBookInfo.com) that map “title” ⇔ “name” etc
Layer 5: Logic

- **Ontologies also allow axioms**
  - “All people have brains”
- **Expressiveness:** Key challenge in formalizing axioms: want to be able to say anything you need to in a particular domain.
  - “All people have brains, except George Bush.”
- **But more expressive logics mean slower inference**
  - Intuitively, applying a rule such as “You can’t fool all of the people all of the time” could require checking everyone in the universe to determine if there exists even one foolish person.

Layer 6: Proofs

- **Proofs ⇒ Trust**
  - In the Semantic Web, a “proof” is a procedure that can be automatically followed in order to verify an assertion.
  - Believability is always relative to a set of resources that you trust
    - I own bank account #239489248234, because my Digital Signature XXXX matches the record on Web page http://bank.com/accounts, and you trust this page because you own bank.com

Motivating Proofs

I would like to buy this book; please send my company an invoice

- I am an employee of XYZ Corp (because it says so on this Web page, which is an XYZ Corp official document)

OK, book successfully ordered

Proof Verifier

Yes this proof is correct
No this proof is flawed

Sorry, we need a credit card!
Summary

- Distributed global information ecosystem enables wide variety of value-added information services
  - monitoring your online purchases;
  - finding entertainment in which you might be interested;
  - scheduling appointments; …
- But doing so requires computers to understand data
  - “Semantics” == meaning
- Global Database/Brain for All Humanity?
- Today’s Syntactic Web may well evolve….

ISWC 2005

- Academic Papers
  - Ontology checking & mapping
- Industry Papers
  - Applications to health care, web personalization, automotive