Semantic Web Applications (part 2) & Trends in the Web
WIS (2II35) lecture
Semantic Web Challenge

• New technologies are only viable for mass adoption if a critical mass of applications exist for it.
• The Semantic Web Challenge aims to find new innovative applications that are based on Semantic Web Technology
• CHIP and iFanzy (presented in the previous class) were top applications in last year’s challenge
• Check it out: it is a good source for seeing the state of the art and get inspiration.
Semantic Web Challenge - Examples

- Paggr
  - build Widgets over annotated Web pages using SPARQL
- DBpediaMobile

- Revyu
  - Review anything on the Web
- Semaploorer
  - interactively explore and visualize large semantically heterogeneous distributed semantic data sets in real-time
The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data. [Tim Berners-Lee, www.w3.org/DesignIssues/LinkedData.html]

Unlike the web of hypertext (links are relationship anchors in hypertext documents written in HTML), for data links between arbitrary things are described by RDF: the URIs identify any kind of object or concept. But, the same expectations apply to make the web grow. We can see these as additional rules for Semantic Data so that we can build a Web of Data, i.e. as an addition to the existing Web of hypertext documents:

- Four principles of Linked Data
  - Use URIs as names for things
  - Use HTTP URIs so that people can look up those names
  - When someone looks up a URI, provide useful information
  - Include links to other URIs, so that they can discover more things
Linked Data: publishing open datasets as RDF & setting RDF links btw items from different sources
Linked Data - Advantages

• Separation of Concern: In current Web pages the semantics are not separated from presentation
  • Using linked data the same content URI can be rendered in different ways (e.g. localization, device dependency, etc)
• ‘Mashups’ of data from multiple sources
  • such as in maps, timelines, etc.
• Define ‘views’ using (SPARQL) queries
• Reuse!
  • No longer create complete domain schemas, but only the stuff that is interesting for you
Combining Datasets

- **IMDB-Cinema-Restaurant scenario**
  - Find a Cinema that shows a film by Guillermo del Toro for which there is a French Restaurant within one kilometer

- **Airline-Hotel-Car Rental scenario**
  - Find the trip to South America for two weeks in July that includes flight, a hotel room for the whole period and a rental car for the whole period with the lowest combined price
“Legacy” Data

- A lot of useful data already exist on the Web in non-RDF form
- RDF is a flexible data model: most other data models can be converted into RDF
- Many RDF-wrappers exist
  - Babel, ConverterToRdf, GRDDL, RDFizers, Triplr, etc
  - Also for relational databases
- Many sources are becoming available
  - DBpedia: Linked data version of Wikipedia
  - US Census: RDF version of the 2000 US census data
  - LinkedMDB: RDF version of IMDB
Why convert to RDF?

- Query information on Webpages
  - i.e. beyond Google keyword matches
- For Example on DBPedia, you can now query for:
  - Give me all Sitcoms that are set in NYC?
  - All tennis players from Moscow?
  - All films by Quentin Tarentino?
  - All German musicians that were born in Berlin in the 19th century?
  - All soccer players with tricot number 11, playing for a club having a stadium with over 40,000 seats and is born in a country with over 10 million inhabitants?
Browsing Linked Data

- RDF Visualization is still a research issue
  - However, specialized visualization exist (like map and timeline visualizations)
  - A mechanism is needed to combine RDF with stylesheet for presentation purposes
- Some General purpose browsers exist
  - Tabulator Browser
  - DISCO Hyperdata Browser
  - OpenLink RDF Browser
  - Rhodonite RDF-editor and browser
RDFa: Integration of RDF in Web pages

- **Vision:** Close the Chasm Between Human and Data Webs

- **Enhancing current Web (XHTML) documents with embedded semantics**
  - Explain the semantics of pieces of content (e.g., dates)

- **Provides a set of attributes to carry metadata in XML tags**

- **One-to-one mapping with RDF**
RDFa

- **about**
  - a URI specifying the resource the metadata is about
- **rel and rev**
  - specifying a relationship or reverse-relationship with another resource
- **href, src and resource**
  - specifying the partner resource
- **property**
  - specifying a property for the content of an element
- **content**
  - overrides the content of the element when using the property attribute
- **datatype**
  - specifies the datatype of text specified for use with the property attribute
- **typeof**
  - specifies the RDF type(s) of the subject
RDFa Example

```html
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
  <h2>The Trouble with Bob</h2>
  <h3>Alice</h3>
</div>
```

```
:http://example.com/alice/posts/42

http://purl.org/dc/elements/1.1/title

"The Trouble with Bob"

http://purl.org/dc/elements/1.1/creator

"Alice"
```
RDFa Example

```xml
<div xmlns:dc="http://purl.org/dc/elements/1.1/"/>
  <h2 property="dc:title">The Trouble with Bob</h2>
  <h3 property="dc:creator">Alice</h3>
  <em property="dc:date" datatype="xsd:date" content="20080421">April 21st, 2008</em>
</div>
```

http://example.com/alice/posts/42

http://purl.org/dc/elements/1.1/title

http://purl.org/dc/elements/1.1/creator

http://purl.org/dc/elements/1.1/date

"The Trouble with Bob"   "20080421"^^xsd:date   "Alice"

http://ben.adida.net/presentations/www2008-rdfa/##(30)
RDFa Example

```html
<div about="/alice/posts/trouble_with_bob"...>
  <h2 property="dc:title">The Trouble with Bob</h2>
  <h3 property="dc:creator">Alice</h3>
</div>

...  

<div about="/alice/posts/jos_barbecue"...>
  <h2 property="dc:title">Jo's Barbecue</h2>
  <h3 property="dc:creator">Eve</h3>
</div>
```

http://ben.adida.net/presentations/www2008-rdfa/#(33)
• The semantic web index: “Over 10 billion pieces of reusable information can already be found across 100 million web pages which embed RDF and Microformats.”

• http://sindice.com/map
• Example: processing today’s business news; for each article I want to extract all of the companies mentioned – but only if the article also mentions a merger or acquisition; I am only interested in companies whose headquarters (or those of their subsidiaries) are located in New York State; do all of that and give me a widget for my news site titled “Merger Activity for NY Consulting Companies”; and oh, by the way, this isn’t a research project – I want you to do it real time for the 10,000 pieces of news I process every day.

• [www.opencalais.com]
How to do that automatically? Some code like:

- For each Article
- Submit to Calais, get response
- If MergerAcquisition exists then
  - For each Company
    - Retrieve Calais Company URI, extract DBpedia link
    - Send Linked Data inquiry to DBpedia, get response
    - If CompanyIndustry contains “Consulting”
      - If CompanyHeadquarters = “New York”
        - Put them on the list
    - For each subsidiary
      - Send Linked Data query to Dbpedia, get result
      - If CompanyHeadquarters = “New York”
        - Put them on the list
- (lots of endif’s)
- Print the list
Calais (Thomson Reuters)

- Pretty straightforward example. How about companies in the news with at least one subsidiary doing business in an area that the CIA Factbook considers dangerous? Or books released by authors who attended Harvard who live in Ohio? Or ... .

- So, TR’s vision: “The combination of semantic data extraction (generic extraction, tags, keywords won’t do the trick) + de-referenceable URIs (entity identifiers you and your programs can retrieve) + the Linked Data Cloud = amazing stuff.”
Machines will never „understand“ content:

“The computer doesn't truly "understand" any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user.”

The semantic web is an engineering solution that will allow the data to be meshed and available for use by machines globally.
Web Science

• Web Science = the science of decentralised information systems.

• Web Science is required both as a way to understand the Web, and as a way to focus its development on key communicational and representational requirements.

Web science, diverse aspects

- Central *engineering* issues, such as the development of the Semantic Web, Web services and P2P.

- *Analytic* approaches to discover the Web’s topology, or its graph-like structures.

- Since the Web as a technology is essentially *socially* embedded, various issues and requirements for Web use and governance are relevant.
Understanding

• The science of decentralised information structures is essential for understanding how informal and unplanned informational links between people, agents, databases, organisations and other actors and resources can meet the informational needs of important drivers such as e-science and e-government.

• How an essentially decentralised system can have such performance designed into it is the key question of Web Science.
Physical science is an analytic discipline that aims to find laws that generate or explain observed phenomena; computer science is predominantly synthetic (formalisms and algorithms are created in order to support particular desired behaviour).

Web science has to be a merging of these two paradigms; the Web needs to be studied and understood, and it needs to be engineered.

- At micro scale, the Web is an infrastructure of artificial languages and protocols; piece of engineering.
- But the linking philosophy of the Web, and its use in communication, result in emergent properties at the macro scale (some of which desirable, and therefore to be engineered in, others undesirable, and if possible to be engineered out).
Managing trust:

• The activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability of potential transactions involving risk, and that also allow players and system owners to increase and correctly represent the reliability of themselves and their systems.
Computing trust:

• The known trust relationships on the web are only a small fraction of the potential pairings. Furthermore, the number of pages, services, and users on the web is so large, that it is difficult to estimate how much trust there is between entities. For example, a user cannot possibly know how much to trust every other user and every page on the web. Instead, trust must be calculated from other available data. Depending on the context, the methods for doing that will vary.
Applications using trust:

• Managing and computing trust are interesting problems, but ultimately they exist to provide trust information that can be used. Building applications that take advantage of trust information and improve their functionality because of it requires an understanding of how trust relates to the system’s goals and how to integrate it. Doing this effectively is a challenge in all domains.
Web 2.0

• In the opening lecture we discussed the new generations of the Web.
• Let us consider now the trend of the new Web and new applications in which terms are social, collaborative, sharing and creative are prominent.
• Clip by Michael Wesch.
WIS 2.0

- These are IS of a different kind.
  - World-wide
  - Open
  - Evolving
  - SOCIAL

- Their engineering asks for more.
  - Content & structure
  - Users
WIS 2.0

• Leads to new research questions.
  • Data/information extraction & transformation
  • Mapping & aligning (linking)
  • Mash-up & engineering
  • User modeling & interoperability
  • Scalability & performance
Final Assignment

- Topic 1: Rule languages
- Topic 2: Data mapping
- Topic 3: Linked data javascript visualization library
- Topic 4: OWL Time
- Topic 5: iFanzy on mobile
- Topic 6: iFanzy and social network website
- Topic 7: Group recommendation
- Topic 8: RDFa
- Topic 9: OpenID, FOAF and SIOC
- Topic 10: Interaction design
- Topic 11: Trust
Final assignment

• Have a look at the topics.
• Choose a top3, and email that to g.j.houben@tue.nl by November 28, noon, in an email from your tue address with subject “WIS final topic: [x] - [y] - [z]” with x, y, and z the numbers of the favorite topics.
• The teacher will then assign all students their final topic and notify them of the precise assignment.
• It is strongly recommended before starting to write (the paper) to send an outline to the teacher for feedback (at least then 😊).
• The results of the assignments should be handed in ultimately by January 25, 2009.
• After handing in the results, a meeting will be scheduled to discuss the result and for grading.
Continuation

• And now we continue on an individual basis 😊

• Let us learn and have fun!