Semistructured Data Management
Part 3 - (Towards the) Semantic Web
Today's Question

1. What is the "Semantic Web"?

2. Semantic Annotation using RDF

3. Ontology Languages
1. The Vision of W3C: Semantic Web

- The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The mix of content on the Web has been shifting from exclusively human-oriented content to more and more data content. The Semantic Web brings to the Web the idea of having data defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications. For the Web to reach its full potential, it must evolve into a Semantic Web, providing a universally accessible platform that allows data to be shared and processed by automated tools as well as by people. The Semantic Web is an initiative of the World Wide Web Consortium (W3C), with the goal of extending the current Web to facilitate Web automation, universally accessible content, and the ‘Web of Trust’.

(http://www.w3.org/2001/sw/Activity)
Phase Zero: Textual Search (reminder)

- Web search engines (Google)

- Searching for data on "anglerfish"
  - Results will be precise

- This seems easy, but the same for "leech"
  - Organism leech
  - Authors: "Bleech", "Leechman", ...
  - Protein sequences: ...MNTSLEECHMPKG...

- Search for "257" ...

Actually Web search engines have made the Web already semantically interoperable, based on what can be supported on pure text processing (essentially text-based search - we will introduce technical details on that later in the part on information retrieval). The problem is that this method is simply not precise enough, in particular for automated processing. It relies on the human understanding of natural language.
The next step taken towards a more meaningful Web was the introduction of additional structure in web documents in the form of user-defined markup. This is a first step towards semantic interoperability. The markup provides in a machine-processable form the necessary interpretation of the data or content. Applications that are able to correctly interpret the "semantic" markup are thus enabled to interpret data correctly. The problem is that different applications use different markups, and that the meaning of the markup is encoded into the terms used for element names, into associated documentation or simply kept in the heads of developers. Again, as soon as different markup schemas are used the relationship between the different interpretations have to be established in the heads of the users or developers, thus semantic interoperation does not occur in an automated manner.
What Do You Think?

• How to overcome semantic heterogeneity?
Three Ways to Overcome Semantic Heterogeneity

1. **Standardization:** agree on common user-defined markup (schemas)
   - great if no pre-existing applications
   - great if power player enforces it

2. **Translation:** create mappings among different schemas
   - requires human interpretation and reasoning
   - mappings can be difficult, expensive to establish

3. **Annotation:** create relationships to agreed upon conceptualizations
   - requires human interpretation and reasoning
   - annotation can be difficult, expensive to establish
   - reasoning over the conceptualization can provide added value

There exist three principal possibilities to tackle the issue of automating semantic processing of the data on the Web:

1. Standardization avoids the problem of semantic heterogeneity at the level of schemas. This approach is used where there exists already (historically) a wide agreement on the structure of relevant information and their interpretation, such as in business. Terminology in the financial world, for example, is pretty standardized, and therefore it is not a major problem to come up with agreed upon formal specifications of the terminology. This is even more the case as there exist in business typically strong players that can enforce the standards.

2. Translation or Mapping between different schemas is the second possibility to establish interoperability. This is the approach that has been extensively studied for integration problems in relatively small and controlled domains, such as business and large organizations. The requirements are typically changing not too quickly, thus much effort can be invested into developing the necessary mappings in order to properly map data from one representation into another, or to map data from multiple representations into one common global representation.

3. The third possibility is slightly different from the second: instead of engineering mappings between heterogeneous schemas for each integration problem, one first agrees on a common conceptualization of the world (or the relevant aspects of the world for a large class of applications). This conceptualization is normally called an ontology. Thus ontologies are fairly application independent and since they are formalized they are machine-processable. Once such an ontology is in place existing information sources can relate the structural elements they use for expressing certain concepts (e.g. element names) to concepts from the the ontology. This then (ideally) allows other applications to properly interpret the contents of the information systems. In addition ontologies in the general case should include reasoning capabilities, which would allow not only to use hard-coded, or pre-canned knowledge (e.g. in form of annotations), but also to derive new knowledge from combining existing knowledge in different ways.
This simple example illustrates the point of how ontologies might help to increase semantic interoperability. Take our earlier example of biological databases. These typically use different schemas to model related facts. For example, Database 1 uses the term Organism to denote an organism, and database 2 uses the term Species to do the same. Two annotators, who share the same ontology, now inspect the document and each of them associates the elements with terms taken from the ontology. So Annotator 1 will decide that the element Organism corresponds to information related to an organism, whereas annotator 2 recognizes actually from the content that the element is related to information about a fish and annotates correspondingly (by establishing a is-instance-of relationship). From that point on the fact that both annotators used the same ontology and that reasoning is possible in this ontology comes into play. Since in the ontology a Fish is a subconcept of Organism (a fact represented formally by a ISA relationship in the ontology) an automated processing tool (e.g. for searching for information) might exploit this relationship and correctly identify for a request for information on Organisms in both databases the related elements.
Ontologies

- Ontologies are an explicit specification of a conceptualization of the real world (Gruber, 1993)
- Ideally
  - different information systems agree on the same ontology
  - relate their model/schema/data elements to the ontology
  - mapping can be constructed via the ontology
- Issues
  - ontology languages (e.g. RDF), mappings and engineering
- Problem: requires agreement on the conceptualization of the real world!

We recall what we have said in the introduction on ontologies:
In order to obtain a better handle on the problem of dealing with different interpretations of data, a possibility is to make the interpretation a formally described entity. This requires the use of some “proxy” for the real world that is formally describable. These “proxies” are called ontologies.
What are the important aspects on ontologies?

1. they require an agreement on the meaning of terms (symbols). In practice, there exists no other way to establish such an agreement than by extensively collecting (from humans) this information and represent it within a formal specification. This however is done and there exist extensive ontologies, some of them have been built up over many years (see the example on the next slide)

2. in order to store this information a representation mechanism (a model) is needed, and the model needs to be encoded into data. The choice of the model is an important issue, since it should be very expressive (we want to model the world!) and easy to use (everyone should be able to use the ontology) at the same time. The encoding is equally important, as it should be done in a standardized form. If this where not the case we would immediately loose the advantage of having a common conceptualization of the world at the abstract level, since we were not able to exchange it properly with others. The examples illustrate of how easy it would be to create confusion at the level of encoding.
Researchers in artificial intelligence first developed *ontologies* to facilitate knowledge sharing and reuse. Since the beginning of the 1990s, ontologies have become a popular research topic, and several AI research communities—including knowledge engineering, natural language processing, and knowledge representation—have investigated them.

More recently, the notion of an ontology is becoming widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management. Ontologies are becoming popular largely because of what they promise: a shared and common understanding that reaches across people and application systems.

One example of an ontology that has been developed as part of these efforts is WordNet. WordNet explains the meaning of English terms in a way as general and as precise as possible. The example gives the different meanings of the term "information" in English. WordNet can be freely accessed and downloaded over the Internet and has become very popular among researchers for that reason.
With respect to the modeling and encoding of ontologies to be used for the semantic Web, there exist a number of requirements, some of which follow from what we have discussed earlier:

1. Simplicity: the success of the Web was always founded on the principle of simplicity of concepts to encourage wide-spread use. Therefore complex models will not be successful. This is an important criterion, since some of the existing ontologies (one example is Cyc) are expressed in fairly complex knowledge representation models.

2. Exchangeability: Since the web is a communication environment, any kind of data that is processed must be easily exchangeable. This is what motivated the use of XML as a data representation format in the first place and should hold for metadata and ontology data as well.

3. Non-intrusive annotation: as the example on annotation we gave earlier demonstrated machine-processable knowledge required for the interpretation of data will be associated with the data typically a-posteriori. Also there exists no always a unique interpretation for the same data. Therefore any attempt to encode the knowledge required for interpretation directly into the data (as it would be the case if we use XML elements for annotation, to give an example) is not workable.

4. Domain-specific vocabularies: the model must provide a mechanism that allows to introduce vocabulary or terminology that is specific to a domain, in other words the possibility to specify schemas for different domains.

5. Modeling primitives: since any ontology model will be used in many different, and potentially very complex contexts (applications) they have to offer a sufficiently rich set of possibilities to model complex situations (e.g. complex structures or complex relationships). There exists a rich experience in modeling (e.g. from data modeling in databases, e.g. the entity-relationship model) and models for ontologies can draw from them.

6. Reasoning Capabilities: the example we discussed earlier already illustrate that also simple forms of reasoning within the ontology layer can make the interpretation of the data much more powerful (and thus the processing in the Semantic Web).

In the table we evaluate HTML, XML, RDF and OIL with respect to each of these aspects. RDF is the Resource Description Framework and is the first WWW standard proposed for the Semantic Web. OIL is the ontology interchange language, an extension of RDF proposed to enrich it with more reasoning capabilities and providing a well-defined semantics. We will introduce both models subsequently.

One way to interpret the introduction of ontology models to support automated processing of semantics on the Web is the following: it can be seen analogously to the step from HTML to XML, were the issues of structuring data where separated from the issues related to the presentation of data. With the Semantic Web an attempt is made to separate the issues related to structuring of data from the issues related to interpreting the data, or in other words separating meaning from structure.
This is the view the World Wide Web consortium develops on the Semantic Web:

Ontologies applied to the World Wide Web are creating the *Semantic Web*. When building the semantic web, several layers are required.

At the lowest level a generic mechanism for expressing machine readable semantics of data is required. Originally, the Web grew mainly around HTML, which provides a standard for structuring documents that browsers can translate in a canonical way to render those documents. On the one hand, HTML’s simplicity helped spur the Web’s fast growth; on the other, its simplicity seriously hampered more advanced Web applications in many domains and for many tasks. This led to XML, which lets developers define arbitrary domain- and task-specific extensions (even HTML appears as an XML application—XHTML).

The **Resource Description Framework (RDF)** is the foundation for processing metadata providing a simple data model and a standardized syntax for metadata. Basically, it provides the language for writing down factual statements.

The next layer, the schema layer introduces means to define vocabulary, structure and constraints for expressing meta data. This is provided by the **RDF Schema** specification.

The fourth layer, then is the **logical layer**. We need ways of representing logic in documents to allow for, for example, rules that represent the deduction of one type of document from a document of another type, the checking of a document against a set of rules of self-consistency; and the resolution of a query by conversion from terms unknown into terms known.
Summary

• Why is the possibility to markup content (as in XML) not sufficient in order to establish semantic interoperability?

• What are ontologies and what is their purpose?

• What advantage has the "annotation" approach to semantic interoperability as compared to the "translation" approach?

• What is the difference between an ontology model and the encoding of an ontology?

• What are reasons not to change documents in order to make them semantically interoperable?
References

• WebSite
  - RDF: http://www.w3.org/
  - OWL: http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.1

• Articles