Introduction to the Semantic Web

Before going deeper into our technologies, expertise and so on, we need to have a basic idea of the main concept behind them. This concept is used everywhere in our documentation - "semantic". For example, "Ontotext is a leading developer of core semantic technologies". So, what do we mean by semantic?

Introduction to the Semantic Web

The term “Semantic Web” refers to the W3C’s vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, OWL, and SKOS.

The first time people talked and wrote about semantic models for representing structured knowledge was in the early sixties. However the term Semantic Web was coined by Tim Berners Lee, the father of the World Wide Web and the director of the World Wide Web Consortium ("W3C"), which now oversees the development of proposed Semantic Web standards. "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. It extends the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are related to each other, enabling automated agents to access the Web more intelligently and perform tasks on behalf of users. Sir Tim Berners Lee defines the Semantic Web as "a web of data that can be processed directly and indirectly by machines."

Tim Berners-Lee on the next Web, 16:20 min. video talk

Short Introduction to the Semantic Web

- Intro Semantic Web and RDF(S) - A biased introduction (2003), pdf presentation by Atanas Kiryakov, CEO Ontotext AD
- "Semantic Search" book chapter, pdf, 33 pages, Jun 2006

Further Introduction to the Semantic Web

If you have like one day to dedicate to this part, please enjoy!

- Introduction to the Semantic Web Tutorial, video presentations by James A. Hendler, Rensselaer Polytechnic Institute, Rensselaer Polytechnic Institute, Sean Bechhofer, School of Mathematics, University of Manchester; Asunción Gómez-Pérez, Universidad Politecnica de Madrid; Aldo Gangemi, Institute of Cognitive Sciences and Technologies

- Realizing a Semantic Web Application, video presentation by Emanuele Della Valle, Politecnico di Milano
    - Slides

Before you start the real journey - get also equipped with:

- W3C Specifications - RDF(S), OWL, SPARQL, SA WSDL
- External Tutorials for RDF(S), OWL, Ontologies, etc.
- Semantic Web Specifications, W3C Recommendations and Notes

Other interesting stuff

- Semantic Web Design Issues Overview, personal notes by Tim Berners-Lee (TBL) explaining the thinking behind the specifications.
    - What the Semantic Web is not, TBL
    - Semantic Web Road map, TBL

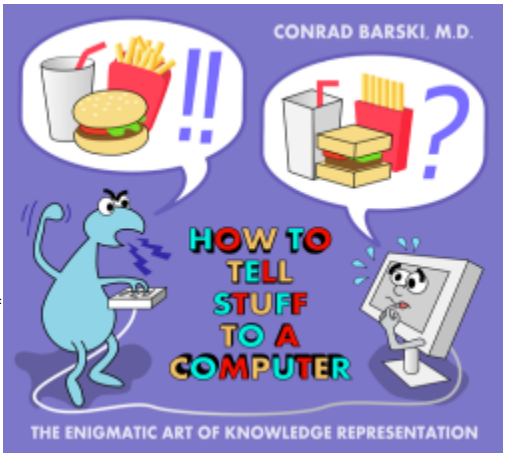
- The Semantic Web, an Interview with Tim Berners-Lee

- Artificial Intelligence and The Semantic Web, html presentation by Tim Berners-Lee, 40 slides, July 2006

- (pdf) The Future of the Web, special on-line issue No.2, Scientific American

Knowledge Representation - Ontologies

What is Knowledge Representation



Formal knowledge representation (KR) is about building models of the world, of a particular domain or a problem, which allow automatic reasoning and interpretation. Such formal models are called ontologies and can be used to provide formal semantics (i.e. machine-interpretable meaning) to any sort of information: databases, catalogues, documents, web pages, etc. The association of information with such formal models makes the information much more amenable to machine processing and interpretation.

Imagine, for instance, a typical database, populated with the information that John is a son of Mary. It will be able to "answer" just a couple of questions: "Who are the sons of Mary?" and "Whose son is John?" An ontology-based system could handle a much bigger range of questions, because it will be able infer that: John is a child of Mary (the more general relation); Mary is a woman; Mary is the mother of John (the inverse relation); Mary is a relative of John (a generalization of the inverse relationship); etc. Although this seems rather simple in the eyes of a human, the above facts would remain "invisible" to a typical database and to any other information system, because their models of the world are limited to data-structures of strings and numbers.

Unfortunately, building ontologies and defining the formal semantics of the data could be an extremely slow, expensive, and error-prone task. In order to automate the process and enable the spreading of ontology-based systems, a number of linguistic and statistical methods are put to use.

Ontologies are crucial for many natural language processing (NLP), knowledge discovery, and text mining tasks. They are the source of common sense required to support non-trivial analysis, and at the same time the periscope necessary to interpret, understand, and make use of the results. Ontologies also contribute significantly to NL generation tasks - it is impossible to generate a reasonable, redundancy-free text without a formal model of the domain, the context, and the reader. This is how KR and NLP, two of the most prominent AI disciplines, can live in synergy, supporting each other.

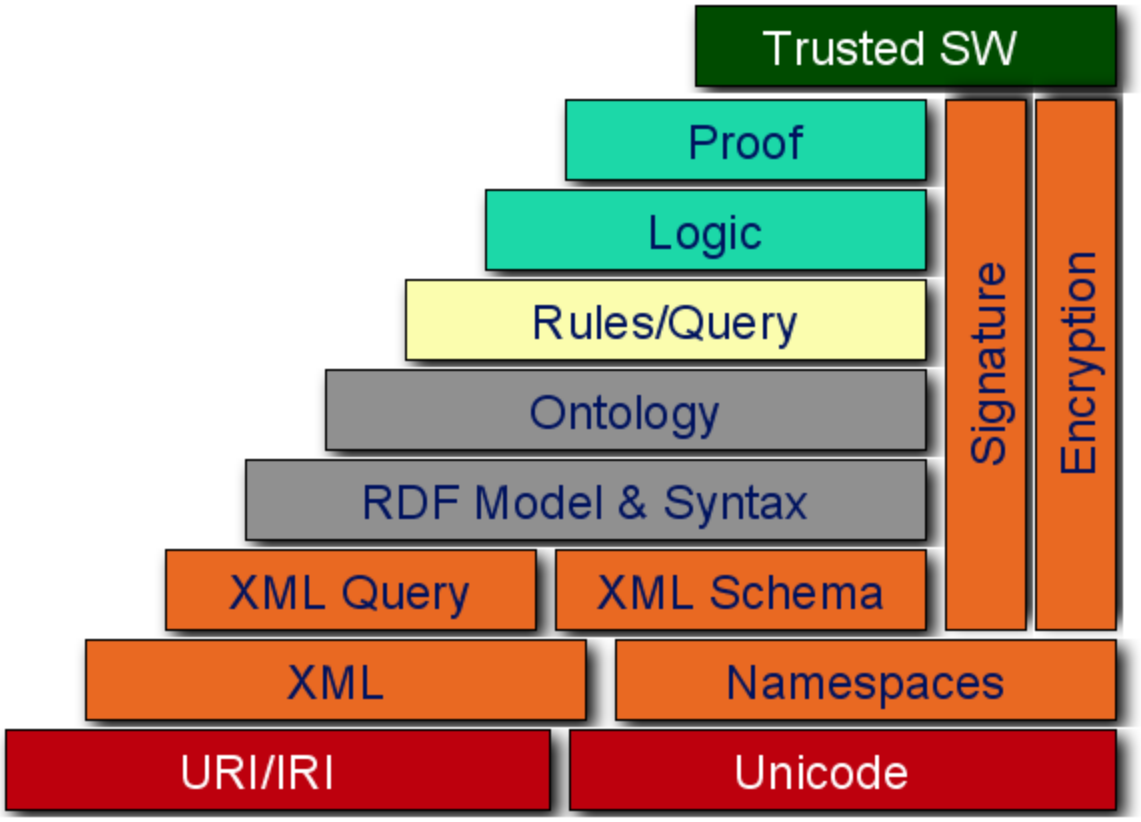
Ontologies can be considered as conceptual schemata, intended to represent knowledge in the most formal and re-usable way possible. Formal ontologies are represented in logical formalisms, such as OWL, that allow automatic inferencing over them and over the datasets aligned to them. An important role of ontologies is to serve as schemata or "intelligent" views over information resources. Thus, they can be used for indexing, querying and reference purposes over non-ontological datasets and systems, such as databases, document and catalogue management systems, etc. As ontology languages have formal semantics, ontologies allow a wider interpretation of data, i.e. inference of facts that are not explicitly stated. In this way, they can improve the interoperability and the efficiency of using arbitrary datasets.

Standards for KR

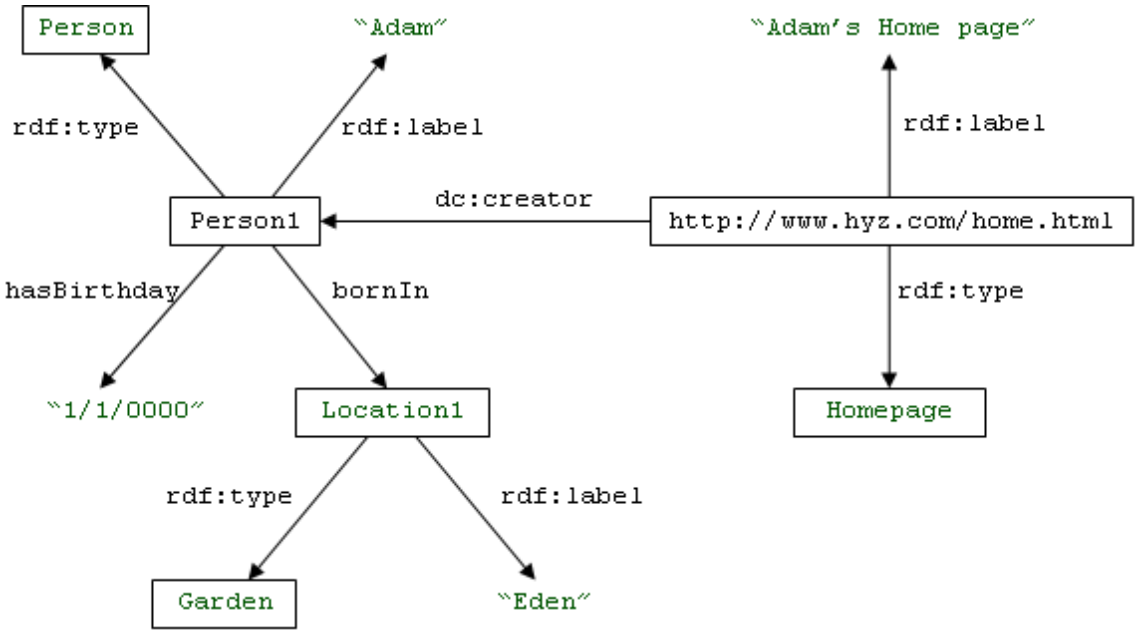
Under the W3C-driven community processes, a family of mark-up and KR standards were developed, as a basis for the Semantic Web. RDF, [KlyneandCarroll2004], is a metadata representation language, which serves as a basic data-model for the Semantic Web. It allows resources to be described through relationships to other resources and literals. The resources are defined through unified resource identifiers (URI), as in XML, e.g. URL. The notion of resource is virtually unrestricted. Anything can be considered as a

resource and described in RDF: from a web page or a picture published on the web to specific entities in the real world (e.g. people, organisations) or abstract notions (e.g. the number Pi and the musical genre Jazz). Literals, again as in XML, are any specific data values e.g. strings, dates, numbers, etc. The main modelling block in RDF is the statement - a triple <Subject, Predicate, Object>, where:

- Subject is the resource, which is being described;
- Predicate is a resource, which determines the type of the relationship;
- Object is a resource or a literal, which represents the "value" of the attribute.



A set of RDF triples can be seen as a graph, where resources and literals are nodes and each statement is represented by a labelled arc (the Predicate or relation), directed from the Subject to the Object. The so-called blank nodes can also appear in the graph, representing unique anonymous resources, used as auxiliary nodes. A sample graph, which describes a web page, created by a person called Adam, can be seen in the Figure below.



RDF Graph Describing Adam and His Home Page

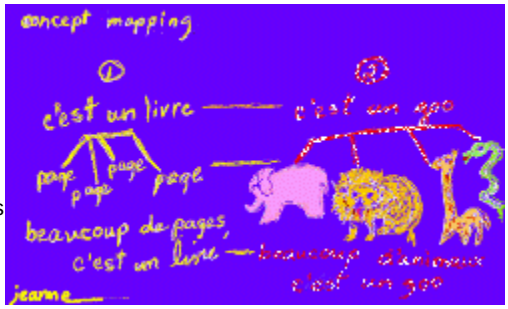
Resources can belong to (formally, be instances of) classes. This can be expressed as a statement through the `rdf:type` system property as follows: `<resource, rdf:type, class>`. Two of the system classes in RDF are `rdfs:Class` and `rdf:Property`. The instances of `rdf:Class` are resources that represent classes, i.e. such resources that can have other resources as instances. The instances of `rdf:Property` are resources that can be used as predicates (relations) in triple statements.

The most popular format for encoding RDF is its XML syntax, [Becket2004]. However, RDF can also be encoded in a variety of other syntaxes. The main difference between XML and RDF is that the underlying model of XML is a tree of nested elements, which is rather different from the graph of resources and literals in RDF.

RDFS, [BrickleyandGuha2000], is a schema language that allows new classes and properties to be defined. OWL, [Deanetal.2004], is an ontology language that extends RDF(S) with the means for more comprehensive ontology definitions. OWL has three dialects: OWL-Lite, OWL-DL, and OWL-Full. Owl-Lite is the least expressive but the most amenable to efficient reasoning. Conversely, OWL-Full provides maximal expressivity but is undecidable. OWL-DL can be seen as a decidable sub-language inspired by the so-called description logics. These dialects are nested in such a way that every OWL-Lite ontology is a legal OWL-DL ontology and every OWL-DL ontology is a legal OWL-Full ontology.

Semantic Databases and Reasoning

Semantic Databases



Semantic databases combine the characteristics of database management systems (DBMS) and inference engines. They provide storage, querying, and management of structured data. One major difference to DBMS is that semantic databases use ontologies as semantic schemata, which allows them to automatically reason about data. Another major difference is that they work with generic physical data models (e.g. graphs). This allows them to easily adopt updates and extensions in the schemata, i.e. in the data structure.

As a result, the semantic databases offer:

- **easy integration of multiple data-sources** - once the schemata of these sources is semantically aligned, the inference capabilities of the engine support the interlinking and combination of facts from different sources;
- **easy querying against rich or diverse data schemata** - inference is applied to match the semantics of the query to the semantics of the data, regardless the vocabulary and the data modelling patterns, used for encoding data;
- **great analytical power**
  - semantics is applied even with recursive inferences on multiple steps;
  - facts are uncovered (based on interlinking long-chains of evidences), the vast majority of which would not be spotted in DBMS;
- **efficient data interoperability** - importing RDF data from one store to another is based on the use of globally unique identifiers.

Benchmarking semantic databases is very complex. Typically the following tasks are benchmarked:

- **data loading** - including parsing, persistence, and indexing;
- **query evaluation** - including query preparation and optimisation and fetching;
- **data modification** - which may involve changes to the ontologies and the schemata;
- **inference** (not a first-level activity) - depending on the implementation, it can affect the performance of the other activities. In the current implementation of the data layer, inference is performed during loading and affects its performance.

However, the best way to evaluate a semantic database is to use a methodology that provides a complete picture of the performance with respect to the full "life cycle" of the data within the engine. We call this a full-cycle benchmarking. At a high-level this means publication of data for both loading and query evaluation performance in the framework of a single experiment or benchmark run. Full-cycle benchmarking also requires that load performance data (e.g. "5 billion triples of LUBM were loaded in 30 hours") is matched with query evaluation data (e.g. "... and the evaluation of the 14 queries took 1 hour on warm database.")

Reasoning Strategies

The two principle strategies for rule-based inference are:

- **Forward-chaining**: to start from the known facts (the explicit statements) and to perform inference in an inductive fashion. The goals of such reasoning can vary - to compute the inferred closure; to answer a particular query; to infer a particular sort of knowledge (e.g. the class taxonomy).
- **Backward-chaining**: to start from a particular fact or a query and to verify it or get all possible results, using deductive reasoning. In a nutshell, the reasoner decomposes (or transforms) the query (or the fact) into simpler (or alternative) facts, which are available in the knowledge database (KB) or can be proven through further recursive transformations.

Inferred closure: the extension of a KB (or a graph of RDF triples) with all implicit facts (triples) that could be inferred from it, based on the enforced semantics.

Materialisation: keeping an up-to-date inferred closure.

The outlined strategies have different advantages and drawbacks well-studied throughout the history of knowledge representation and expert systems. Hybrid strategies (involving partial forward- and backward-chaining) are also possible and in many contexts proven to be efficient.

References

- OWLIM v5.4 Documentation
- Measurable Targets for Scalable Reasoning, Atanas Kiryakov, 2007
- D2.6.3 A scalable repository for massive semantic annotation, Damyan Ognyanoff, Atanas Kiryakov, Rouslan Velkov, Milena Yankova, SEKT project, Jan 2007.
- OWLIM: balancing between scalable repository and light-weight reasoner, Atanas Kiryakov, Presented at the Developer's Track of WWW2006, Edinburgh, Scotland, UK, 23-26 May, 2006.
- OWLIM - a Pragmatic Semantic Repository for OWL, Atanas Kiryakov, Damyan Ognyanov, Dimitar Manov, In Proc. of Int. Workshop on

Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, 20 Nov, New York City, USA.
- On-To-Knowledge in a Nutshell, Dieter Fensel, Frank van Harmelen, Ying Ding, Michel Klein, Hans Akkermans, Jeen Broekstra, Arjohn Kampman, Jos van der Meer, York Sure, Rudi Studer, Uwe Krohn, John Davies, Robert Engels, Victor Iosif, Atanas Kiryakov, Thorsten Lau, Ulrich Reimer, Ian Horrocks, IEEE Computer, 2002.
- Tracking Changes in RDF(S) Repositories, Atanas Kiryakov and Damyan Ognyanov, In the Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management EKAW02, Sigüenza, Spain, 1-4 October 2002.
- BOR: a Pragmatic DAML+OIL Reasoner, Kiril Simov, Stanislav Jordanov, Deliverable 40, On-To-Knowledge project, June 2002.
- Tracking Changes in RDF(S) Repositories, Atanas Kiryakov and Damyan Ognyanov, In the Proceedings of Workshop on Knowledge Transformation for the Semantic Web, at the 15th European Conference on Artificial Intelligence, pages 27-25. Lyon, France, July 23, 2002.

Semantic Search in General

What is it?

Semantic Search is about finding information that is not based on the presence of text (keywords, phrases) but rather on the meaning of the words. The problem with keyword-based search engines is that, if the information is published by diverse sources, the same term may be used with different meaning and different terms may be used for concepts that have the same meaning.

Semantic Search engines try to bridge this gap by using semantics and thus offer the user more precise and relevant results.

Behind Semantic Search

Semantic Search takes advantage of conceptual models such as ontologies, knowledge bases, thesauri, etc.

These conceptual models work at the human conceptual level and at the same time they provide computer-usable definitions of the same concepts. By structuring the knowledge in a given domain, they offer common language that allows more efficient communication and problem-solving.

Semantic Search is Useful for

- handling complex and heterogeneous information resources
- retrieving documents based on a set of relationships that are external to these documents
- providing multiple search options for richer investigation
- targeting and sifting results more efficiently
- using authoritative information resources more effectively as guides to searching