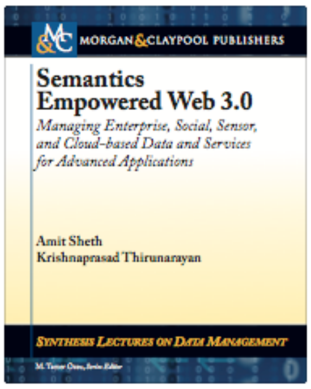
Lecture notes for INFO116

Lecture 2: Models of Semantics

Material from Ch2. P.18 – 27



Dr. Csaba Veres,
Institute for Information and Media Science,
The University of Bergen

How to represent semantics so that it is accurate and useful, in real world applications.

We want semantics to do something for us, and we need to think about precise problems it can help us solve (rather than the high level descriptions like *integration* and *abstraction*).

An initial classification of broad *types* of semantics is between *prescriptive* and *descriptive* semantics.

- *Prescriptive* semantics involve the use of special logical languages to

encode knowledge. The language provides the tools for representation and reasoning, but the content must be entered from scratch. That is, it allows us to add new information to existing resources in a precise and formal way.
- *Descriptive* semantics tries to capture contemporary usages of terms, to eventually promote consensus. Typical approaches provide some notation which captures the existing use of terms. This kind of semantics describes existing information in resources, in a more precise way.

Within each approach we find different levels of complexity and expressive power.

Prescriptive semantics.

- first-order logic
- modal logics
- Relational algebra/calculus [Ramakrishnan and Gehrke, 2002]
- RDF (Resource Description Framework)
- description logics [Baader, 2003]
- OWL [Hitzler et al., 2009b]

In each case, the meaning of symbols representing concepts and relationships is captured unambiguously through the sentences in the language of logic, and sound inferences can be mechanically derived. On the other hand, the "completeness" of inferences depends on how accurately a specification mirrors or can mirror the real-world concepts and relationships. In any case, these approaches clearly specify, in purely symbolic terms, expected answers to queries and inferences. They differ in the kinds of things they can express.

Logic is the formal mathematical study of the methods, structure, and validity of mathematical deduction and proof.

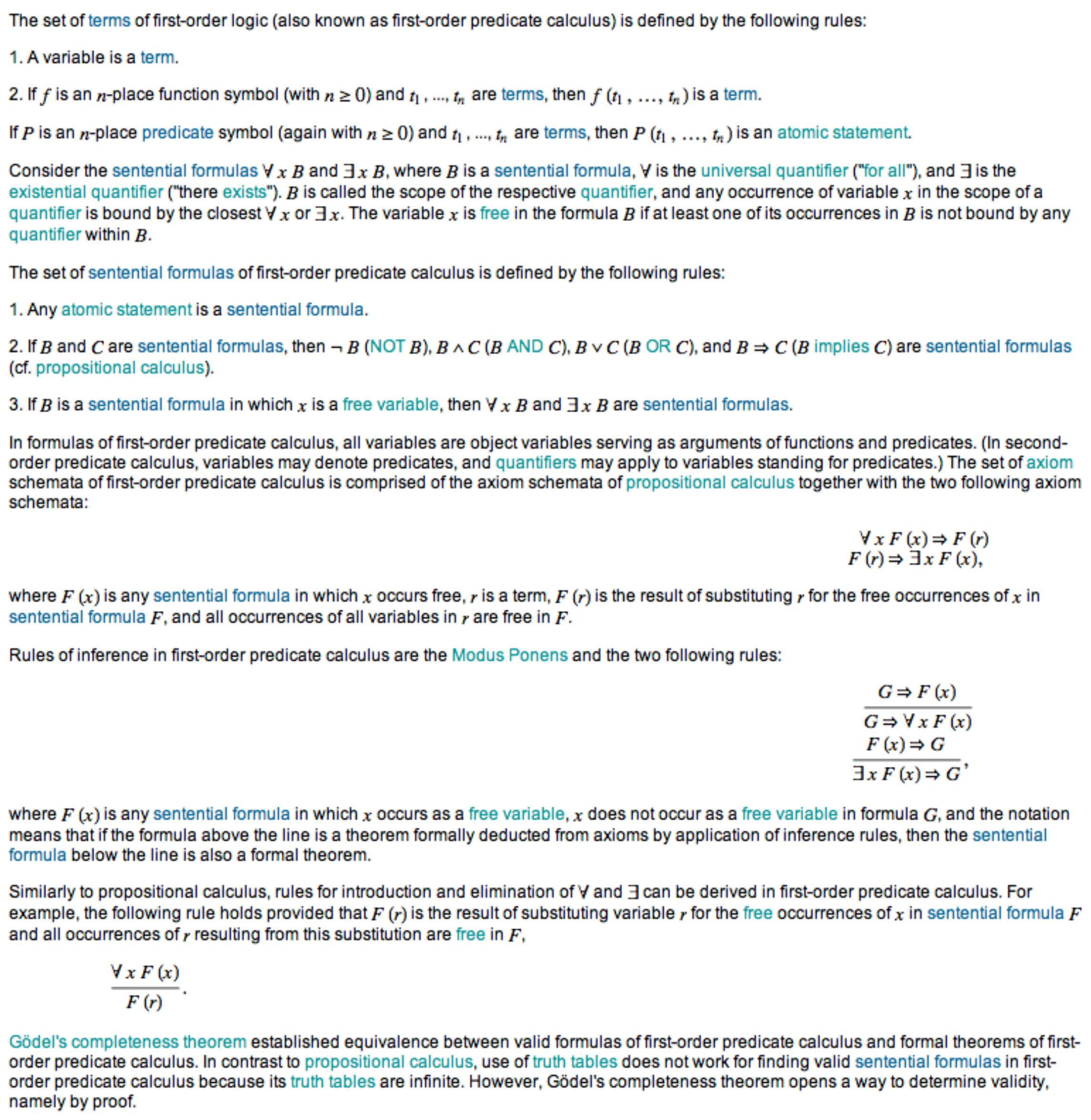
In Hilbert's day, formal logic sought to devise a complete, consistent formulation of mathematics such that propositions could be formally stated and proved using a small number of symbols with well-defined meanings. The difficulty of formal logic was demonstrated in the monumental Principia Mathematica (1925) of Whitehead and Russell's, in which hundreds of pages of symbols were required before the statement could be deduced.

The foundations of this program were obliterated in the mid 1930s when Gödel unexpectedly proved a result now known as Gödel's incompleteness theorem. This theorem not only showed Hilbert's goal to be impossible, but also proved to be only the first in a series of deep and counterintuitive statements about rigor and provability in mathematics.

A very simple form of logic is the study of "truth tables" and digital logic circuits in which one or more outputs depend on a combination of circuit elements (AND, OR, NAND, NOR, NOT, XOR, etc.; "gates") and the input values. In such a circuit, values at each point can take on values of only true (1) or false (0). de Morgan's duality law is a useful principle for the analysis and simplification of such circuits.

A generalization of this simple type of logic in which possible values are true, false, and "undecided" is called three-valued logic. A further generalization called fuzzy logic treats "truth" as a continuous quantity ranging from 0 to 1.

- FOL (First Order Logic) and Modal Logics. First-order logic deals with objects, functions, and relationships. It provides syntax for specifying functions and relationships through terms and well-formed formulae involving Boolean operators (and, or, not, implies, etc.) and quantifiers (universal and existential).



The set of terms of first-order logic (also known as first-order predicate calculus) is defined by the following rules:

1. A variable is a term.

2. If f is an n-place function symbol (with $n \geq 0$) and $t_1, ..., t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.

If P is an n-place predicate symbol (again with $n \geq 0$) and $t_1, ..., t_n$ are terms, then $P(t_1, \ldots, t_n)$ is an atomic statement.

Consider the sentential formulas $\forall x\, B$ and $\exists x\, B$, where B is a sentential formula, $\forall$ is the universal quantifier ("for all"), and $\exists$ is the existential quantifier ("there exists"). B is called the scope of the respective quantifier, and any occurrence of variable x in the scope of a quantifier is bound by the closest $\forall x$ or $\exists x$. The variable x is free in the formula B if at least one of its occurrences in B is not bound by any quantifier within B.

The set of sentential formulas of first-order predicate calculus is defined by the following rules:

1. Any atomic statement is a sentential formula.

2. If B and C are sentential formulas, then $\neg B$ (NOT B), $B \wedge C$ (B AND C), $B \vee C$ (B OR C), and $B \Rightarrow C$ (B implies C) are sentential formulas (cf. propositional calculus).

3. If B is a sentential formula in which x is a free variable, then $\forall x\, B$ and $\exists x\, B$ are sentential formulas.

In formulas of first-order predicate calculus, all variables are object variables serving as arguments of functions and predicates. (In second-order predicate calculus, variables may denote predicates, and quantifiers may apply to variables standing for predicates.) The set of axiom schemata of first-order predicate calculus is comprised of the axiom schemata of propositional calculus together with the two following axiom schemata:

$$\forall x\, F(x) \Rightarrow F(r) \quad (1)$$
$$F(r) \Rightarrow \exists x\, F(x), \quad (2)$$

where $F(x)$ is any sentential formula in which x occurs free, r is a term, $F(r)$ is the result of substituting r for the free occurrences of x in sentential formula F, and all occurrences of all variables in r are free in F.

Rules of inference in first-order predicate calculus are the Modus Ponens and the two following rules:

$$\frac{G \Rightarrow F(x)}{G \Rightarrow \forall x\, F(x)} \quad (3)$$
$$\frac{F(x) \Rightarrow G}{\exists x\, F(x) \Rightarrow G}, \quad (4)$$

where $F(x)$ is any sentential formula in which x occurs as a free variable, x does not occur as a free variable in formula G, and the notation means that if the formula above the line is a theorem formally deducted from axioms by application of inference rules, then the sentential formula below the line is also a formal theorem.

Similarly to propositional calculus, rules for introduction and elimination of $\forall$ and $\exists$ can be derived in first-order predicate calculus. For example, the following rule holds provided that $F(r)$ is the result of substituting variable r for the free occurrences of x in sentential formula F and all occurrences of r resulting from this substitution are free in F,

$$\frac{\forall x\, F(x)}{F(r)}. \quad (5)$$

Gödel's completeness theorem established equivalence between valid formulas of first-order predicate calculus and formal theorems of first-order predicate calculus. In contrast to propositional calculus, use of truth tables does not work for finding valid sentential formulas in first-order predicate calculus because its truth tables are infinite. However, Gödel's completeness theorem opens a way to determine validity, namely by proof.

(Sakharov, Alex. "First-Order Logic." From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/First-OrderLogic.html)

- Modal Logic. A modal is an expression (like ‘necessarily’ or ‘possibly’) that is used to qualify the truth of a judgement. Modal logic is, strictly speaking, the study of the deductive behavior of the expressions ‘it is necessary that’ and ‘it is possible that’. However, the term ‘modal logic’ may be used more broadly for a family of related systems. These include logics for belief, for tense and other temporal expressions, for the deontic (moral) expressions such as 'it is obligatory that' and 'it is permitted that', and many others. An understanding of modal logic is particularly valuable in the formal analysis of philosophical argument, where expressions from the modal family are both common and confusing. Modal logic also has important applications in computer science. (http://plato.stanford.edu/entries/logic-modal/)

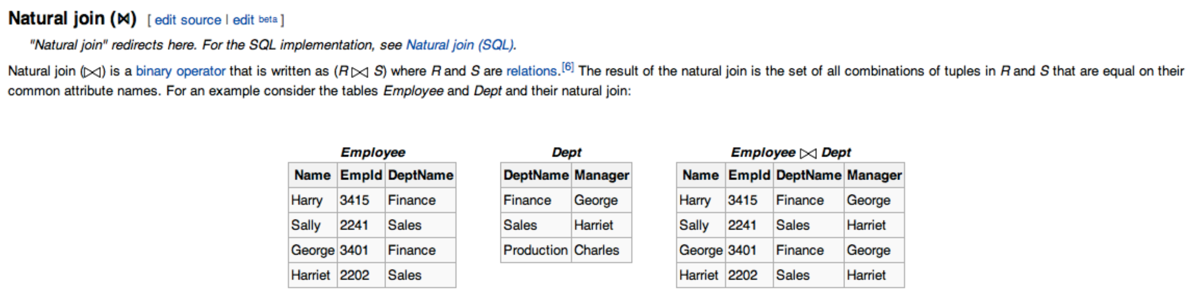
Logic	**Symbols**	**Expressions Symbolized**
Modal Logic	□	It is necessary that ..
	◊	It is possible that ..
Deontic Logic	*O*	It is obligatory that ..
	P	It is permitted that ..
	F	It is forbidden that ..
Temporal Logic	*G*	It will always be the case that …
	F	It will be the case that …
	H	It has always been the case that …
	P	It was the case that …
Doxastic Logic	*Bx*	x believes that …

Axiom Name	Axiom
(D)	□A→◊A
(M)	□A→A
(4)	□A→□□A
(B)	A→□◊A
(5)	◊A→□◊A
(CD)	◊A→□A
(C4)	□□A→□A
(C)	◊□A → □◊A

- Relational Algebra/Calculus. Relational algebra deals with finitary relations and five primitive closure operators: the selection, the projection, the Cartesian product, the set union, and the set difference. Codd [1970] proposed such an algebra as a basis for database query languages, and showed that it is essentially equivalent in expressive power to relational calculus (a limited fragment of first-order logic). Furthermore, relation queries can be computed efficiently.

Basis for *relational database*. Much less expressive than the more powerful logics, but computationally tractable and fast. No general purpose reasoning, just operations permitted by relational algebra. Good for data retrieval.

Implemented in SQL.
SELECT *
FROM employee NATURAL JOIN department;


Natural join (⋈) [edit source | edit beta]

"Natural join" redirects here. For the SQL implementation, see Natural join (SQL).

Natural join (⋈) is a binary operator that is written as (*R* ⋈ *S*) where *R* and *S* are relations.[6] The result of the natural join is the set of all combinations of tuples in *R* and *S* that are equal on their common attribute names. For an example consider the tables *Employee* and *Dept* and their natural join:

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Manager
Finance	George
Sales	Harriet
Production	Charles

Employee ⋈ Dept

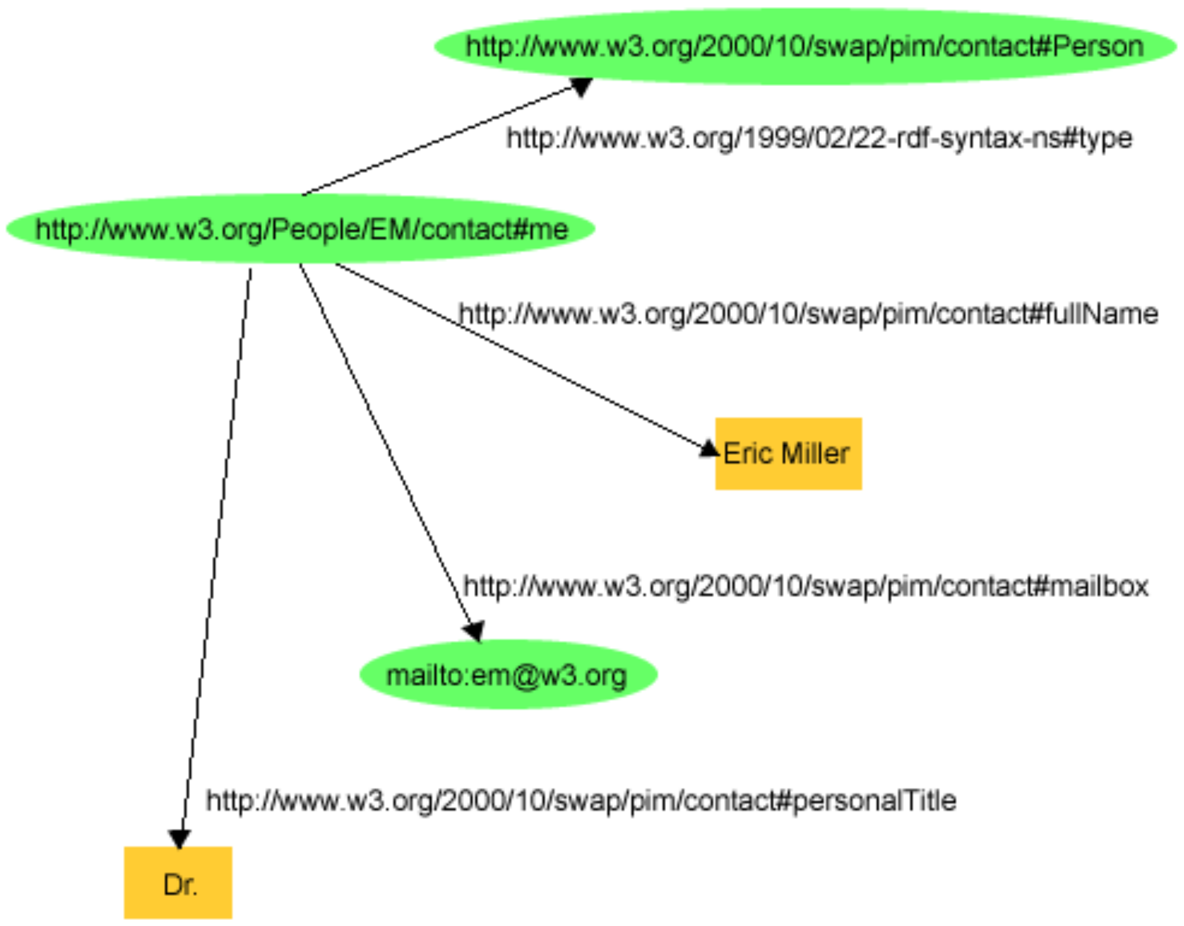
Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet

- RDF. The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a **metadata data model**. It has come to be used as a general method for conceptual description or **modeling** of information that is implemented in web resources, using a variety of syntax notations and data **serialization** formats.

The RDF data model is similar to classic conceptual modeling approaches such as entity–relationship or class diagrams, as it is based upon the idea of making statements about resources (in particular web resources) in the form of subject-predicate-object expressions. These expressions are known as **triples** in RDF terminology. The *subject* denotes the resource, and the *predicate*

denotes traits or aspects of the resource and expresses a relationship between the subject and the *object*. For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue". RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.
This mechanism for describing resources is a major component in the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.
A collection of RDF statements intrinsically represents a labeled, directed multi-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models. However, in practice, RDF data is often persisted in relational database or native representations also called Triplestores, or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple. As RDFS and OWL demonstrate, one can build additional ontology languages upon RDF.

Example: RDF Description of a person named Eric Miller (http://www.w3.org/TR/rdf-primer/)
The following example is taken from the W3C website describing a resource with statements "there is a Person identified by http://www.w3.org/People/EM/contact#me, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr.".


http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
Eric Miller
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:em@w3.org
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Dr.

An RDF Graph Describing Eric Miller

The resource "http://www.w3.org/People/EM/contact#me" is the subject.
The objects are:
"Eric Miller" (with a predicate "whose name is"),
em@w3.org (with a predicate "whose email address is"), and
"Dr." (with a predicate "whose title is").
The subject is a URI.
The predicates also have URIs. For example, the URI for each predicate:
"whose name is" is http://www.w3.org/2000/10/swap/pim/contact#fullName,
"whose email address is" is http://www.w3.org/2000/10/swap/pim/contact#mailbox,
"whose title is" is http://www.w3.org/2000/10/swap/pim/contact#personalTitle.
In addition, the subject has a type (with URI http://www.w3.org/1999/02/22-rdf-syntax-ns#type), which is person (with URI http://www.w3.org/2000/10/

swap/pim/contact#Person).
Therefore, the following "subject, predicate, object" RDF triples can be expressed:
http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#fullName, "Eric Miller"
http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#mailbox, em@w3.org
http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#personalTitle, "Dr."
http://www.w3.org/People/EM/contact#me, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/10/swap/pim/contact#Person

RDF can be stored in text files, relational databases, or native *triplestores*. In addition, it can be embedded into HTML via RDFa.

Originally, the emphasis was on using RDF for describing metadata for Web resources, but later its wider applicability has been realized. Specifically, RDF statements can be used to define class (unary relation) instances, property (binary relation) instances, collections, and properties about RDF statements through reification. These are also called ABox or assertional knowledge.
RDFS specifies information about standard properties (type, domain, range, subclass, sub- property, etc.) used in the vocabulary of RDF. These properties are used to define other vocabulary terms such as by specifying class and property inheritance hierarchies. These are also called TBox or terminological knowledge. For example, the RDFS assertion that "PassengerVehicle is a MotorVe- hicle" can be encoded in XML as

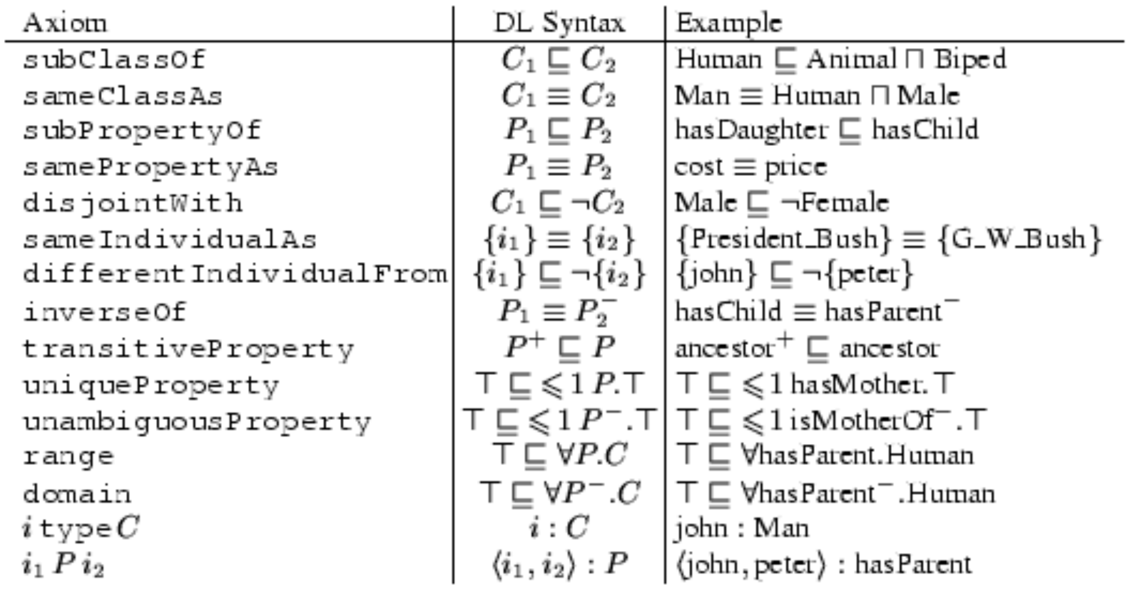
```
<rdf:Description ID="PassengerVehicle">
   <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
```

In other words, RDF can represent more complex statements including Description Logic (DL)

- Description Logics (DLs) and Web Ontology Language (OWL)

Description logics (DLs) are a family of restricted FOL languages that are more expressive than propositional logic but are more tractable than first-order logic. Specifically, theoremhood in DLs is decidable. OWL is a family of knowledge representation languages for modeling ontologies that is a culmination of two earlier concurrent efforts called the DARPA Agent Markup Language (DAML) Program and Ontology Inference/Interchange Layer (OIL). OWL comes in three flavors, called species, to provide different expressivity/computational efficiency trade- offs: OWL-Full, OWL-DLs, and OWL-Lite. OWL-DLs can be further refined on the basis of the constraints and the closure characteristics of the concepts (classes) and the roles (properties) they support. OWL-DLs (Web Ontology Language-DL) can be viewed as machine-processable standardization of DLs for interoperability and scalability that promotes reuse of data and reasoning infrastructure over the Web. OWL-DLs overlap with RDF/RDFS in that both can use RDF formatted data and share common classes and predicates. However, OWL-DLs also simplify and regularize RDF formal semantics by abolishing reification, and enrich RDFS by providing more expressive primitives (e.g., subsumption, inverse, transitivity, and cardinality constraints) to facilitate axiomatization. OWL2 further includes constructs for expressiveness and syntactic sugars for convenience by defining three different profiles: OWL 2 EL, OWL 2 QL, and OWL 2 RL with useful computational properties for scalability and interoperability.

Using strict logic to structure knowledge, as we have discussed with first order logic, can make our knowledge database slow and can lead to unavoidable infinite loops in our knowledge storage and retrieval useful information. However, we can avoid some of this intractability in our software by restricting the power of our language to be less than that of first order logic. Earlier, we had done this by restricting ourselves only to *and or* and *not*. In this we're going to do this by laying out our logic data so that we only use logic commands that are useful for discussing worlds that are organized into objects. Although this ends up limiting the descriptive power of our knowledge structures, we hope that it does so in a way that is easier for a software program to handle. **Also, we hope that much real-world information tends to be easily described with objects, so that our language remain expressive in all the ways that are important.** (http://lisperati.com/tellstuff/dl.html)



Axiom	DL Syntax	Example
`subClassOf`	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
`sameClassAs`	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
`subPropertyOf`	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
`samePropertyAs`	$P_1 \equiv P_2$	cost $\equiv$ price
`disjointWith`	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
`sameIndividualAs`	$\{i_1\} \equiv \{i_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
`differentIndividualFrom`	$\{i_1\} \sqsubseteq \neg\{i_2\}$	{john} $\sqsubseteq$ $\neg${peter}
`inverseOf`	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
`transitiveProperty`	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
`uniqueProperty`	$\top \sqsubseteq \leqslant 1 P.\top$	$\top \sqsubseteq \leqslant 1$ hasMother.$\top$
`unambiguousProperty`	$\top \sqsubseteq \leqslant 1 P^-.\top$	$\top \sqsubseteq \leqslant 1$ isMotherOf$^-$.$\top$
`range`	$\top \sqsubseteq \forall P.C$	$\top \sqsubseteq \forall$hasParent.Human
`domain`	$\top \sqsubseteq \forall P^-.C$	$\top \sqsubseteq \forall$hasParent$^-$.Human
i `type` C	$i : C$	john : Man
i_1 P i_2	$\langle i_1, i_2 \rangle : P$	$\langle$john, peter$\rangle$: hasParent

Descriptive semantics.

- XML
- Entity-Relationship Model (ERM) and Unified Modelling Language (UML)

- ControlledVocabulary,Thesaurus,andTaxonomy
- Folksonomy

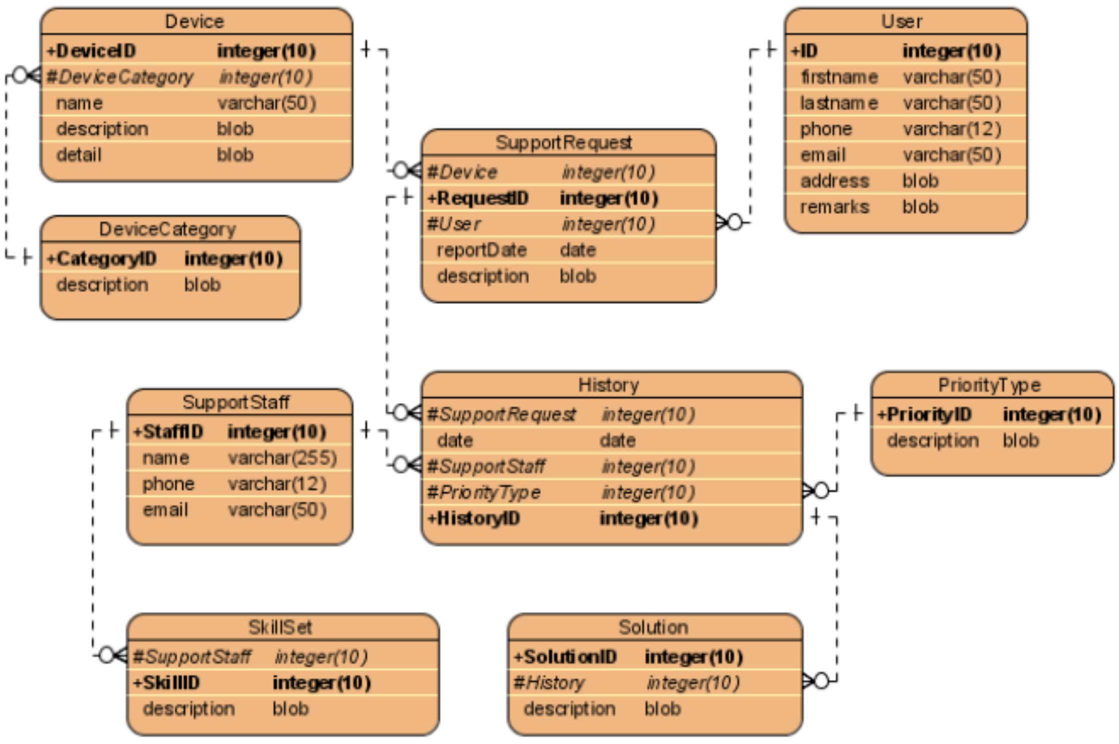
- XML

Extensible Markup Language (XML) is a set of syntactic rules for encoding documents and data in a machine-readable form. It is a metalanguage for designing custom markup languages for different sorts of documents and data. Hypertext Markup Language (HTML) is an instance of XML. Even though originally HTML was intended to make the logical structure and content of a semi-structured document explicit, it evolved into a markup language for presentation. XML Schemas are analogous to context-free grammars in that they specify valid nested structures.

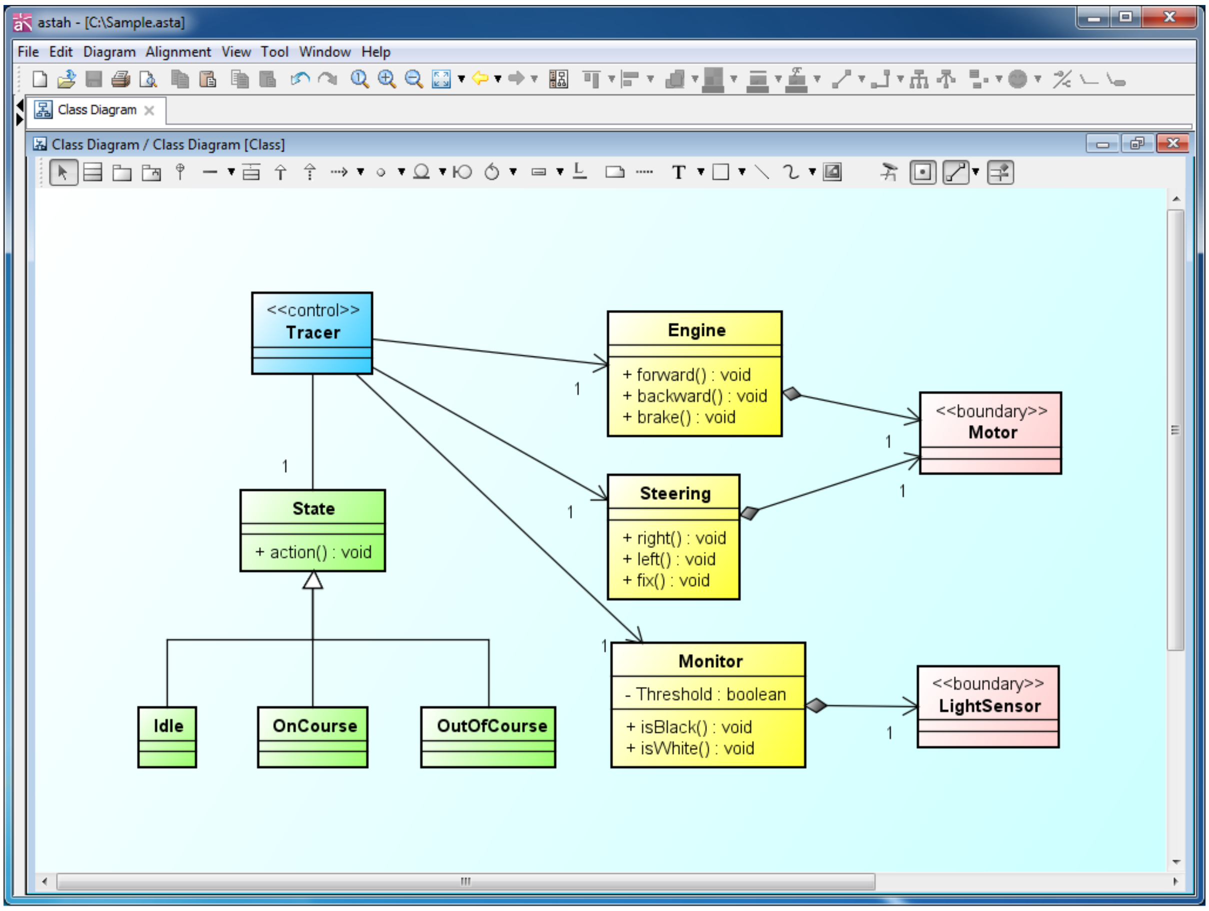
XML can describe any document, but the semantics are implicit: what is the meaning of each section?

- Entity-Relationship Model (ERM) and Unified Modelling Language (UML)

Entity-Relationship Model (ERM) is used to design a conceptual scheme or semantic data model of an information system. An entity represents an object (e.g., book, person). A relationship captures how two or more entities are related to one another (e.g., authorOf, causes). Entities and relationships can both have attributes (e.g., age, or totalCost [associated with an order for an item from a supplier]). Entities have primary keys, which are uniquely identifying attributes (e.g., studentId, SSN). Variants of ERMs provide additional notation for capturing weak relationship semantics such as cardinality constraints, participation constraints, generalization, and specialization.



The Unified Modeling Language (UML) is used to specify, visualize, modify, construct, and document an object-oriented software system (in contrast with databases or information systems), especially the structure and the behavior (including interaction) of system components, to varying degrees of detail.



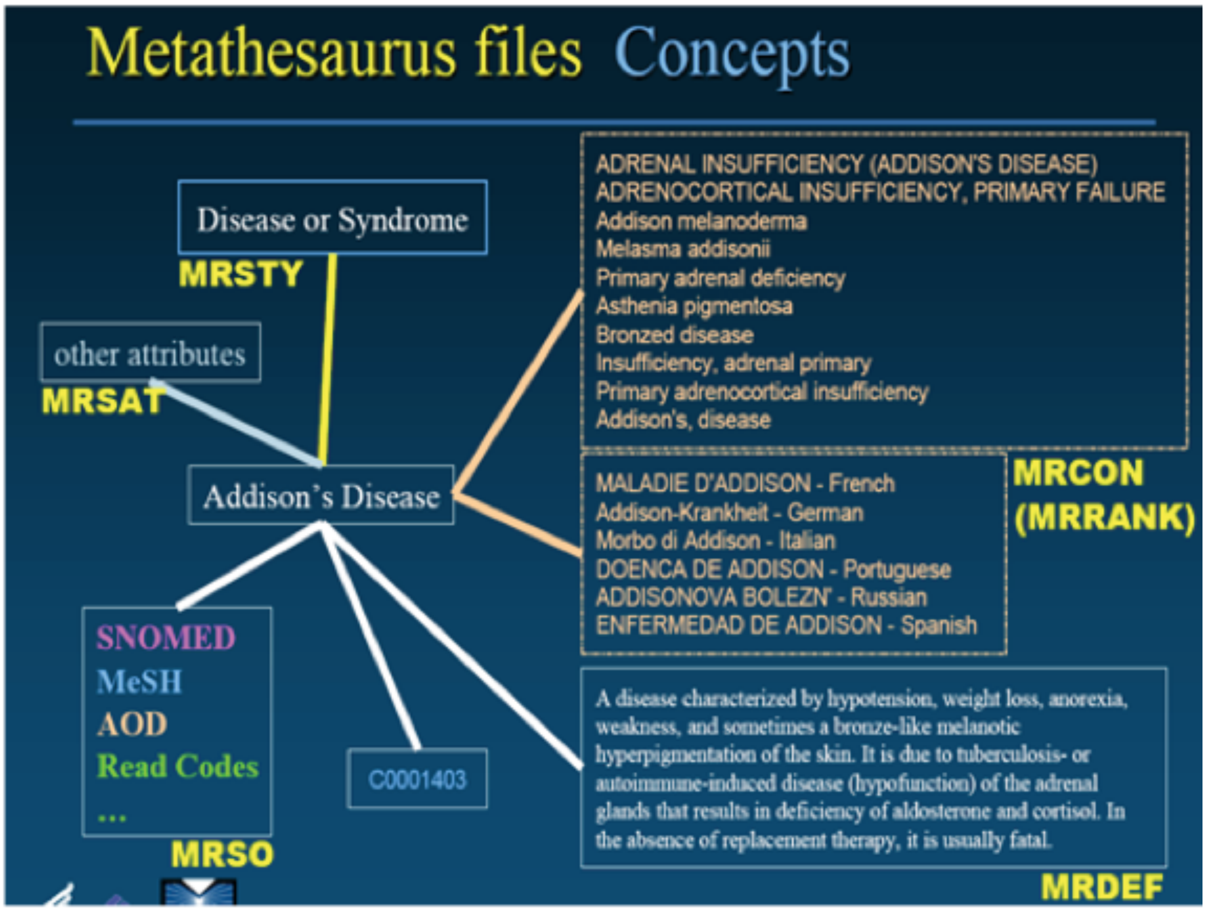
- ControlledVocabulary,Thesaurus,andTaxonomy

Controlled vocabularies provide a way to organize knowledge by grouping together terms corresponding to a concept, and mandating the use of preferred/normalized terms to refer to the concept. For instance, controlled vocabularies are used for indexing and cataloging books in a library. They are used for formalizing materials and process specifications by defining terms for describing composition, processing, treatment, and testing of alloys, and packaging for shipment. Controlled vocabularies facilitate capturing and communicating semantics of a document through preferred terms, that is, enable determining semantic equivalence through syntactic matching of mapped preferred terms. A thesaurus is similar to a controlled vocabulary in that it associates a term with a group of terms that are synonymous (share the same denotation) or very similar.

WORDNET [Fellbaum, 1998] groups nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms (synsets), and these sets are further organized using semantic relations such as broadening/generalization, narrowing/ specialization, is-part-of, includes, and entailment. Synsets can be used in equational reasoning while ISA-relationships can be used in inheritance reasoning.

Taxonomy refers to hierarchical classification and nomenclature. Historically, Carl Linnaeus is credited with developing the first well-developed taxonomy to classify organisms. It used a binomial nomenclature system consisting of a generic name followed by a specific epithet to stand for a class of organism (e.g., Homo Sapiens for humans, Solanum Lycopersicum for tomatoes). The relationship between neighboring concepts in a taxonomy does not have a fixed interpretation, that is, it can represent generalization/specialization, whole-part, or some other association (e.g., Flynn's taxonomy classifies computer architectures as: SISD, SIMD, MISD, and MIMD, where S=Single, I=Instruction, M=Multiple, and D=Data.) Cloud taxonomy classifies services as: IaaS, PaaS, SaaS, and DaaS, where I=Infrastructure, P=Platform, S=Software, D=Data Analytics, and aaS=as a Service. ACM categories and subject descriptors provide taxonomy to classify a technical paper based on its content.

The Unified Medical Language System (UMLS) (http://www.nlm.nih.gov/ research/ umls/) is a widely used resource that integrates various controlled vocabularies using a biomedical thesaurus and taxonomy. It consists of a metathesaurus of medical concepts (obtained from established vocabularies such as SNOMED, ICD-9-CM, and MeSH) and inter-concept relationships to enable information exchange between different clinical databases and systems. That is, UMLS sup- ports conversion of terms from one controlled medical vocabulary to another. UMLS also consists of a semantic network, which specifies categories (e.g., Disease or Syndrome, Biologic Function, Antibiotic) to which medical concepts (e.g., Adrenal Gland Disease, Adrenal Disorder, Surrenale Maladies [French]) defined in the Metathesaurus can belong, and semantic relationships (e.g., functionally related to, affected-by, temporally related to, follows, etc.) that can be assigned between these concepts. UMLS contains 135 semantic types and 54 semantic network relationships.


Metathesaurus files Concepts
Disease or Syndrome
MRSTY
other attributes
MRSAT
Addison's Disease
ADRENAL INSUFFICIENCY (ADDISON'S DISEASE)
ADRENOCORTICAL INSUFFICIENCY, PRIMARY FAILURE
Addison melanoderma
Melasma addisonii
Primary adrenal deficiency
Asthenia pigmentosa
Bronzed disease
Insufficiency, adrenal primary
Primary adrenocortical insufficiency
Addison's, disease
MALADIE D'ADDISON - French
Addison-Krankheit - German
Morbo di Addison - Italian
DOENCA DE ADDISON - Portuguese
ADDISONOVA BOLEZN' - Russian
ENFERMEDAD DE ADDISON - Spanish
MRCON
(MRRANK)
SNOMED
MeSH
AOD
Read Codes
...
MRSO
C0001403
MRDEF

A disease characterized by hypotension, weight loss, anorexia, weakness, and sometimes a bronze-like melanotic hyperpigmentation of the skin. It is due to tuberculosis- or autoimmune-induced disease (hypofunction) of the adrenal glands that results in deficiency of aldosterone and cortisol. In the absence of replacement therapy, it is usually fatal.

- Folksonomy

Folksonomy is a taxonomy created by folks (ordinary users). It consists of freely selectable keywords or tags, which can be liberally attached to any information resource [Gruber, 2007]. The motivation for assigning tags to Web resources or entities is to succinctly summarize the nature or content or context of the information resource, such as photos, videos, and web pages. The tags provide multiple perspectives on a resource that facilitates sharing, organizing, and retrieving the resource (especially the multimedia kind). The tagging process is also referred to as social tagging, collaborative indexing, and social bookmarking. A folksonomy is very easy to create and use (compared to more formal approaches to semantics). However, it is prone to user subjectivity bias. Because the tag search is based on syntactic matching, its success relies on how well the tags reflect consensus between the categorization vocabulary used by the creators and the retrieval vocabulary used by the consumers. Thus, the search effectiveness is not guaranteed.

Folksonomy grew out of a bottom-up process of people tagging Web content they created or encountered (using applications such as social bookmarking [Del.icio.us] and social photo- sharing [Flickr]). In contrast, typical taxonomies are developed in a top-down manner. Whereas taxonomies constrain the local choices at the leaves by global categorizations in the branches, folksonomies as practised do not impose such global consistency requirements. So, as Tom Gruber2 observed, taxonomy is really an embodiment of top-down categorization as a way of finding and organizing information, while folksonomy is really the observation that we now have an entirely new source of data for finding and organizing information: user participation.

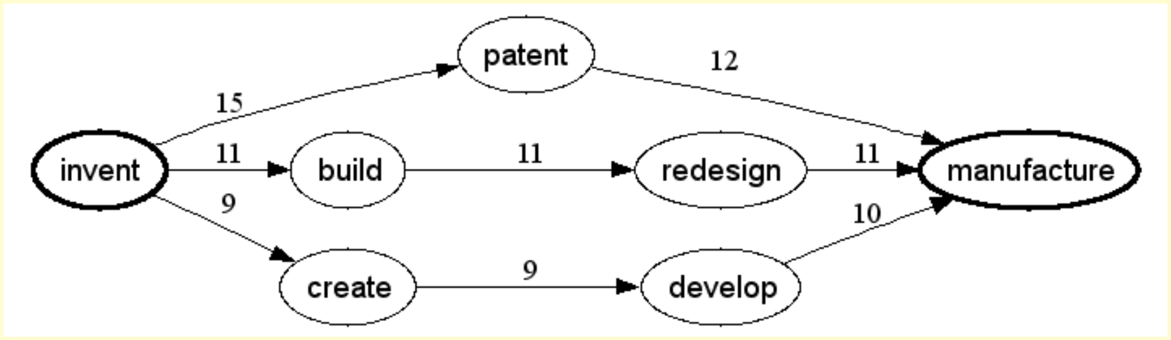
(http://quicktips.vn/dev/dinh-nghia-va-ap-dung-tag-clouds-trong-seo--art-26-quicktips.html)

General Methods for Creation and Publication of Semantic Models

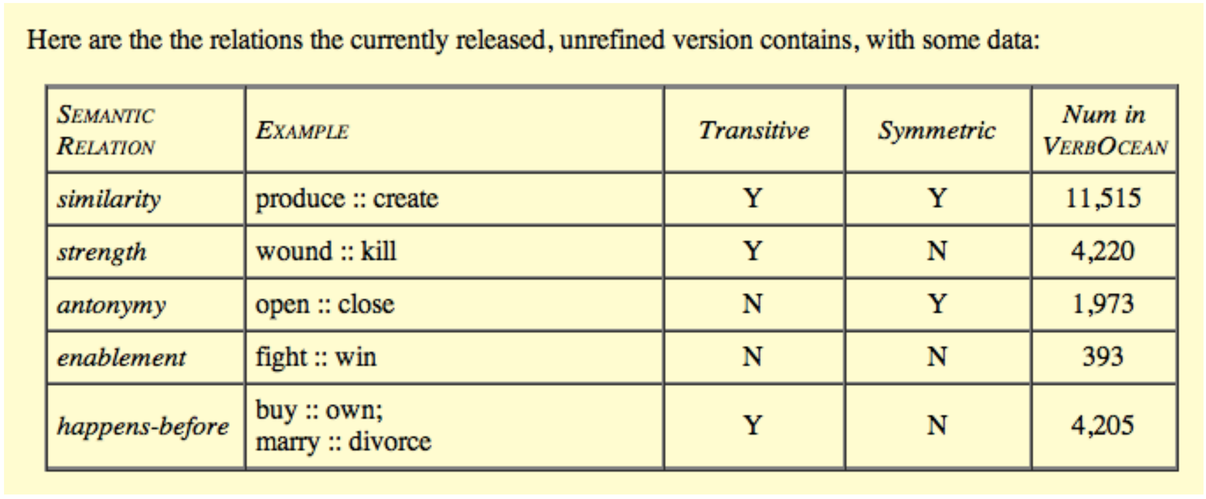
The way in which semantic models are constructed can depend on the particular kind of semantics. Folksonomy, for example, is by definition created as a collaborative social exercise. Descriptive semantic models can be constructed manually, semi-automatically, or automatically.

For instance, UMLS is an example of a manually curated semantic model. On the other hand VerbOcean is a broad-coverage semantic network of verbs semi-automatically mined from the Web documents [Chklovski and Pantel, 2004]. Fine-grained semantic relations between verbs such as similarity, antonymy, and temporal happens before relations between pairs of verbs are determined using lexico-syntactic patterns over the Web.

http://demo.patrickpantel.com/demos/verbocean/



Example: VerbOcean's temporal precedence chains (the "happens-before" relation) between *invent* and *manufacture* shown with edge weights



Here are the the relations the currently released, unrefined version contains, with some data:

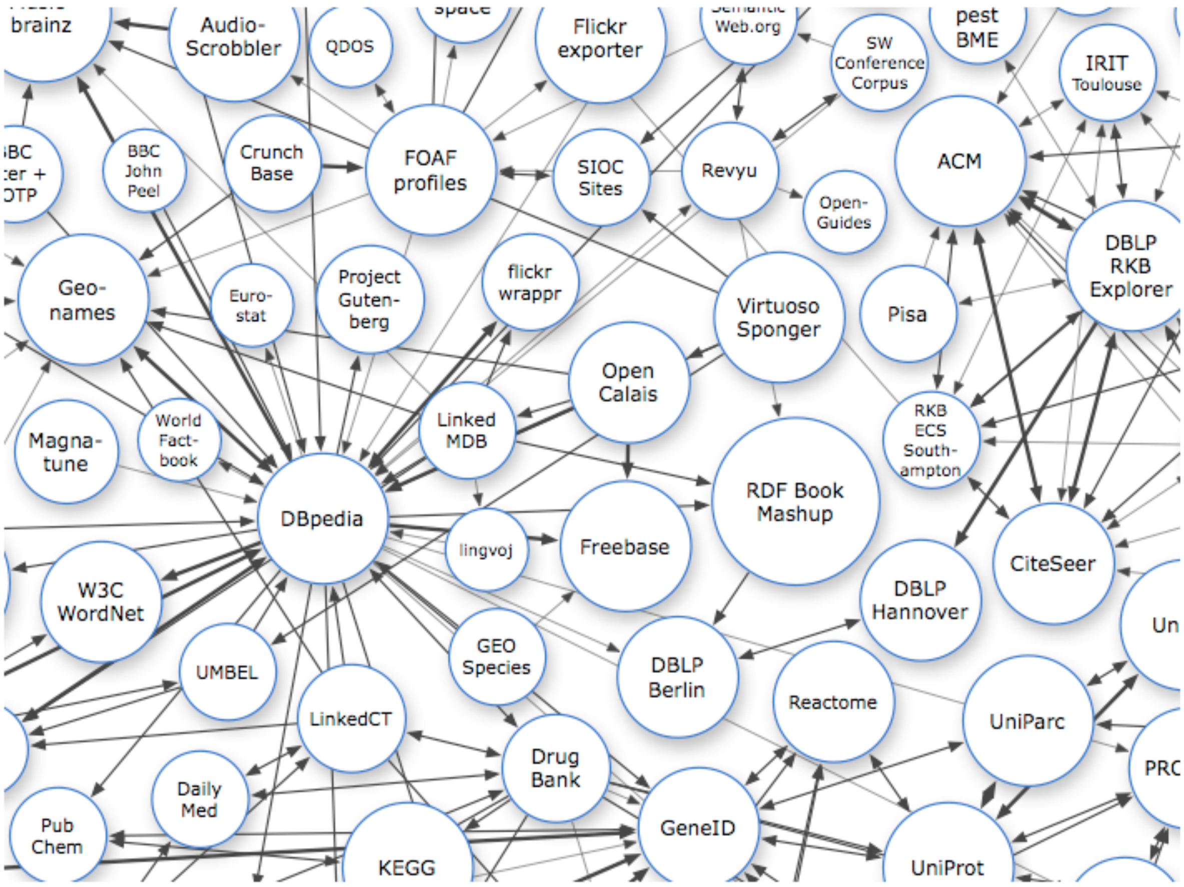
Semantic Relation	*Example*	*Transitive*	*Symmetric*	*Num in VerbOcean*
similarity	produce :: create	Y	Y	11,515
strength	wound :: kill	Y	N	4,220
antonymy	open :: close	N	Y	1,973
enablement	fight :: win	N	N	393
happens-before	buy :: own; marry :: divorce	Y	N	4,205

http://www.patrickpantel.com/download/papers/2004/emnlp04.pdf

Linked Open Data, simplified publishing on the web

LOD (http://linkeddata.org/) provide an approach to publishing structured data (which includes ontologies and instance data) in a form that is conducive to their consumption. Berners Lee [2006] outlined four design principles of LOD:

1. Use URIs to identify things.
2. Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.
3. Provide useful information about the thing when the URI is dereferenced, using standard formats such as RDF/XML.
4. Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.


brainz
Audio-Scrobbler
QDOS
space
Flickr exporter
Semantic Web.org
SW Conference Corpus
pest BME
IRIT Toulouse
BBC John Peel
Crunch Base
FOAF profiles
SIOC Sites
Revyu
Open-Guides
ACM
DBLP RKB Explorer
Geo-names
Euro-stat
Project Guten-berg
flickr wrappr
Virtuoso Sponger
Pisa
Open Calais
Magna-tune
World Fact-book
Linked MDB
RDF Book Mashup
RKB ECS South-ampton
DBpedia
lingvoj
Freebase
W3C WordNet
DBLP Hannover
CiteSeer
Un
UMBEL
GEO Species
DBLP Berlin
Reactome
UniParc
LinkedCT
Drug Bank
PRO
Daily Med
Pub Chem
KEGG
GeneID
UniProt