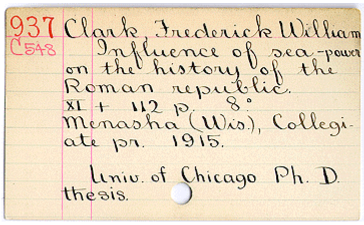
Outline

Why the Semantic Web?
What is the Semantic Web?
Semantic Web Layers
Semantic Web Technologies
Issues

Outline

Why the Semantic Web?
What is the Semantic Web?
Semantic Web Layers
Semantic Web Technologies
Issues

Why Library Card Catalogs?


937 C548 Clark, Frederick William
Influence of sea-power on the history of the Roman republic.
xi + 112 p. 8°
Menasha (Wis.), Collegiate pr. 1915.
Univ. of Chicago Ph. D. thesis.

Source: http://www.library.upenn.edu/exhibits/pennhistory/library/cards/cards.samples.html

Source: http://content.cdlib.org/ark:/13030/kt1489q7c9/?brand=calisphere

Source: http://workgroups.cwrl.utexas.edu/visual/files/TRINITY-COLLEGE-LIBRARY-DUB.jpg

Catalogs as Metadata

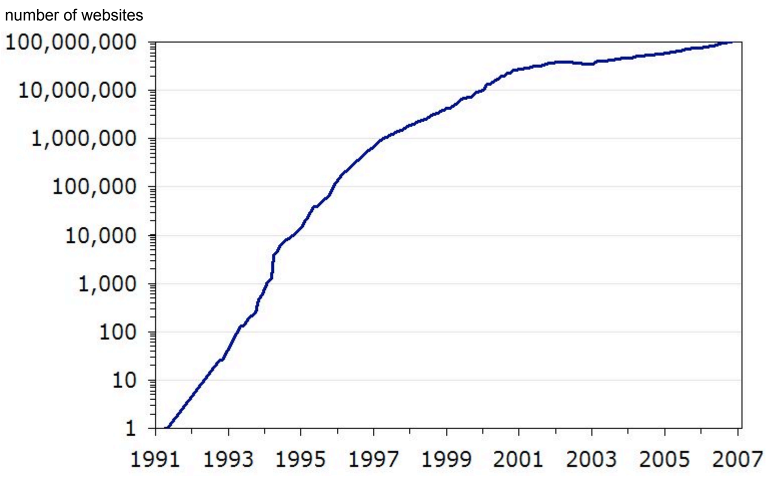
Library card catalogs describe books and other information resources in the library.

They facilitate the search and management of the resources in the library.

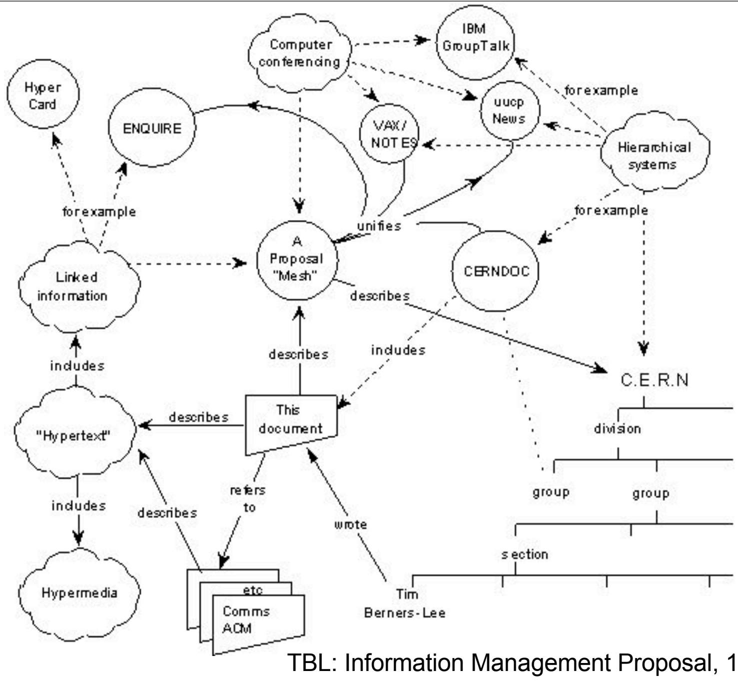
They are an example of metadata.

Metadata is ***structured data about other data***.

Tremendous Growth of the Web



Source: http://www.useit.com/alertbox/web-growth.html



TBL: Information Management Proposal, 1989

Outline

Why the Semantic Web?

What is the Semantic Web?

Semantic Web Layers

Semantic Web Technologies

Semantic Web, Web 2.0 and Semantic Computing

Issues

Semantic Web: Definition

"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."

Berners-Lee, Hendler and Lassila, Scientific America, May,2001

• The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

http://www.w3.org/2001/sw/

Definition

The semantic web is a web whose resources are annotated or described by formal, machine-processable, web-accessible, ontology-based metadata.

These metadata are normally expressed in an XML-based knowledge representation language, e.g., RDF(S), OWL.

Semantic Web and Metadata

A machine's understanding of information will help it better process the information.

Simple tagging of information wouldn't do very much to help a machine to understand information, particularly the information generated by various groups of people on the web.

Tags with well-defined words taken from common vocabularies would help.

These tags - - … - - > **metadata**

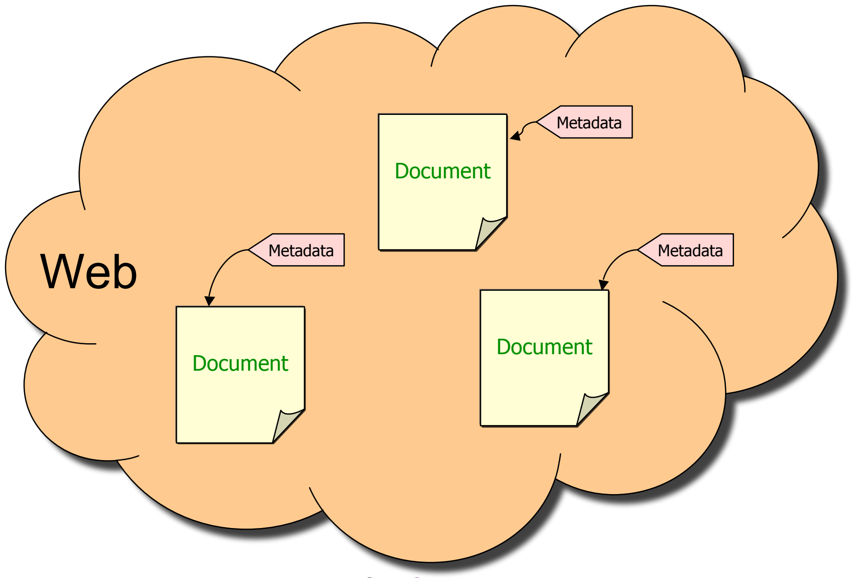
Theses common vocabularies - - … - -> **ontologies**.

Current Web vs Semantic Web

The current contains lots of information for humans to consume.

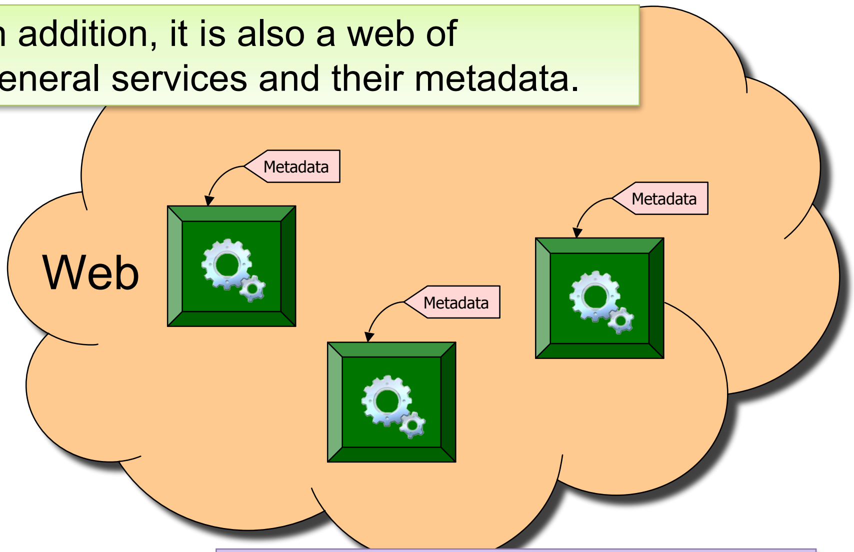
The current web is for humans.

The semantic web is for **both humans to consume and for machine to process for human consumption**.



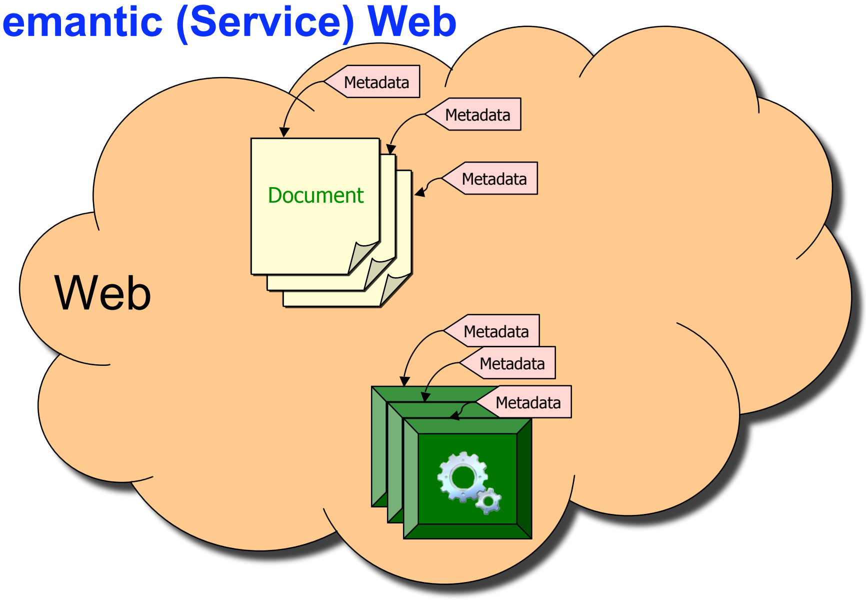
A semantic web is a web of **information providing** services in which information is annotated with metadata.

In addition, it is also a web of general services and their metadata.



Such a **service with its metadata** is called a **semantic web service**.

Semantic (Service) Web



Semantic Web, Semantic Service Web and Semantic Web Services

A **semantic web** is a web of information providing services.

A **semantic service web** is a web of general services and their metadata.

A service with associated metadata is called a **semantic web service.**

Metadata

Metadata can be about anything (any resource or service), used for any application and by any user.

There will be lots of metadata.

It must be machine-processable and shareable.

Metadata is expensive.

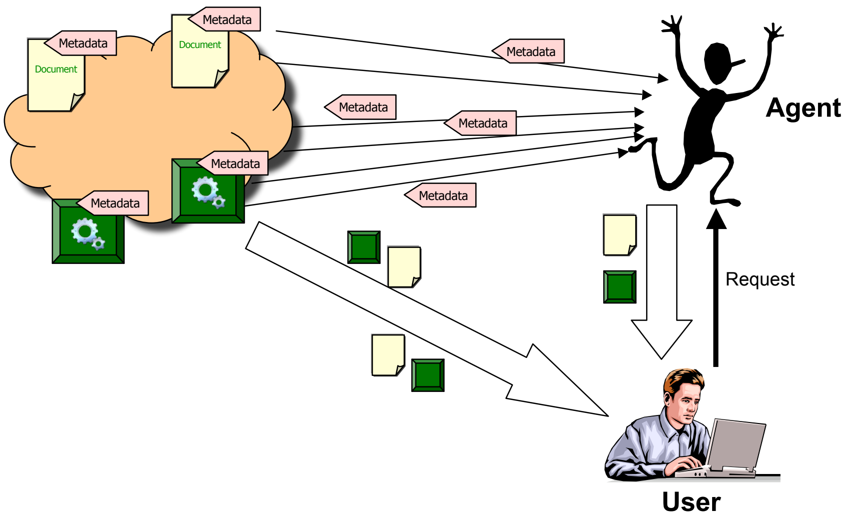
Metadata may contain constraints and rules.

– authoring date must precede publication date

Therefore,

– its elements and their values must follow some standards,
– it must be reusable,
– its languages should be formal, universal, and able to express constraints and rules.

Metadata and Semantic Web



From Book Catalogues to Metadata

Book Catalogue

530.1 Black holes and baby universes
H392b and other essays
1993 Hawking, S.W. (Stephen W.)
New York, N.Y. :Bantam Books,
c. 1993 ix, 182 p. ; 24 cm

1. Hawking, SW 2. Cosmology
2. Science—Philosophy

QC16.H33A3 1993 530.1 93-8269
Library of Congress AACR2 MARC

Metadata

Element	Value
Title	:string
Creator	:string
Publisher	:string
Date	:date
Subject	category list

Standardized, sharable, reusable

ONTOLOGIES*

**An formal, explicit specification of a shared conceptualization of a domain of interest .*

Ontology

A heavily overloaded term
with several different meanings
in different disciplines:

- Philosophy
- Linguistics
- Computer Science)

Ontology - Philosophy

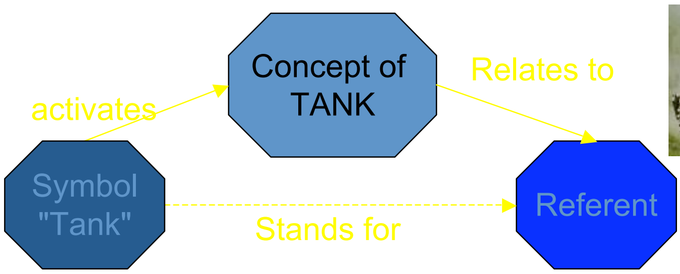
Ontology deals with the nature and organisation of reality (Aristotele)

Tries to answer the questions:

- What characterizes being?
- What is being?

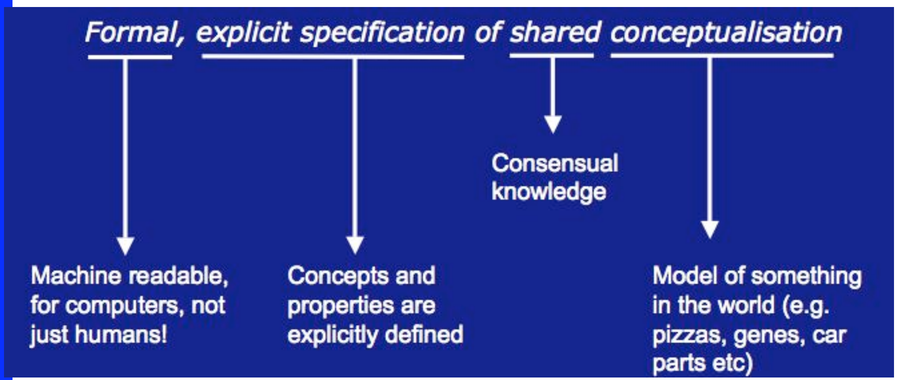
Ontology - Linguistics

a concept, is the mediator that relates the symbol to its object



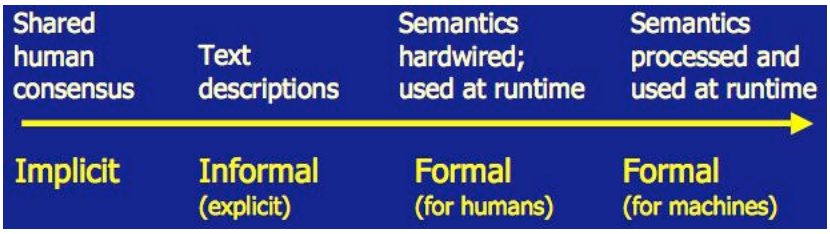
"Semiotic triangle"

Ontology - Computer Science



Rudi Studer(98)

A Semantic continuum



Further to the right:

- Less ambiguity
- Better inter-operation
- More robust
- More difficult

Structure of an Ontology

Ontologies typically have two distinct components:

Names for important concepts in the domain

- Elephant is a concept whose members are a kind of animal
- Herbivore is a concept whose members are exactly those animals who eat only plants or parts of plants
- Carnivore is a concept whose members are exactly those animals who eat other animals
- Adult_Elephant is a concept whose members are exactly those elephants whose age is greater than 20 years

Background knowledge/constraints on the domain

- Adult_Elephants weight at least 2,000 kg
- All Elephants are either African_Elephants or Indian_Elephants
- No individual can be both a Herbivore and a Carnivore

Ontologies

An ontology is a systematic structuring of externalized perceived knowledge or conceptualization.

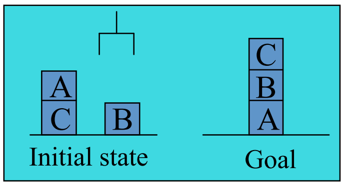
An ontology is a taxonomy (a vocabulary with structure) plus a set of constraints, relationships, and rules.

Ontologies can be used to design and create metadata elements as well as their values.

The same domain could be perceived or conceptualized differently by different people.

Sharing of ontologies is important.

What is a conceptualization?



Conceptualization 1

Objects: Relations:

Objects	Relations
block A	on(X, Y)
block B	above(X, Y)
block C	clear(X)
table A	handEmpty
hand A	

Conceptualization 2

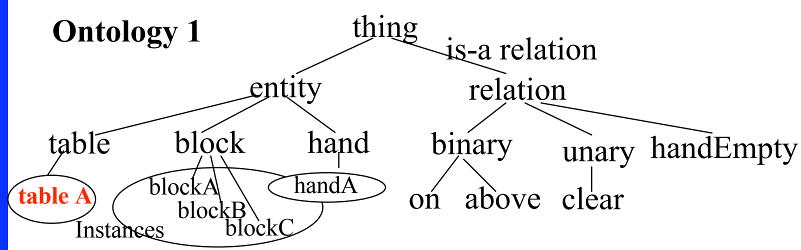
Objects: Relations:

Objects	Relations
block A	on(X, Y)
block B	above(X, Y)
block C	**onTable(X)**
hand A	clear(X)
	handEmpty

[source: Mizoguchi]

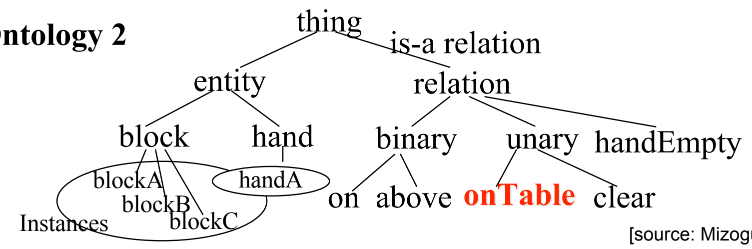
An ontology of the block world

Ontology 1



Axiom: **above(X,Z):-on(X,Y), on(Y,Z).**

Ontology 2

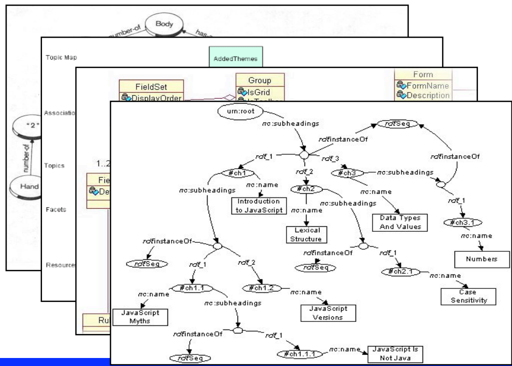


[source: Mizoguchi]

Ontology Languages

There are a wide variety of languages for “Explicit Specification”

- Graphical Notations
  - Semantic Networks
  - Topic Maps
  - UML
  - RDF



Ontology Languages

There are a wide variety of languages for “Explicit Specification”

- Graphical Notations
  - Semantic Networks
  - Topic Maps
  - UML
  - RDF
- Logic Based
  - Description Logics
  - Rules
  - First Order Logic
  - Conceptual Graphs

Every gardener likes the sun.
(Ax) gardener(x) => likes(x,Sun)
You can fool some of the people all of the time.
(Ex)(At) (person(x) ^ time(t)) => can-fool(x,t)
You can fool all of the people some of the time.
(Ax)(Et) (person(x) ^ time(t) => can-fool(x,t)
All purple mushrooms are poisonous.
(Ax) (mushroom(x) ^ purple(x)) => poisonous(x)
No purple mushroom is poisonous.
~(Ex) purple(x) ^ mushroom(x) ^ poisonous(x)
(Ax) (mushroom(x) ^ purple(x)) => ~poisonous(x)
There are exactly two purple mushrooms.
(Ex)(Ey) mushroom(x) ^ purple(x) ^ mushroom(y) ^ purple(y) ^ ~(x=y) ^ (Az) (mushroom(z) ^ purple(z)) => ((x=z) v (y=z))
Clinton is not tall.
~tall(Clinton)

Requirements for Ontology Languages

Ontology languages allow users to write explicit, formal conceptualizations of domain models

The main requirements are:

- a well-defined syntax
- efficient reasoning support
- a formal semantics
- sufficient expressive power
- convenience of expression

Tradeoff between Expressive Power and Efficient Reasoning Support

The richer the language is, the more inefficient the reasoning support becomes

Sometimes it crosses the border of *noncomputability*

We need a compromise:

- A language supported by reasonably efficient reasoners
- A language that can express large classes of ontologies and knowledge.

Reasoning About Knowledge in Ontology Languages

Class membership

- If x is an instance of a class C, and C is a subclass of D, then we can infer that x is an instance of D

Equivalence of classes

- If class A is equivalent to class B, and class B is equivalent to class C, then A is equivalent to C, too

Reasoning About Knowledge in Ontology Languages (2)

Consistency

- X instance of classes A and B, but A and B are disjoint
- This is an indication of an error in the ontology

Classification

- Certain property-value pairs are a sufficient condition for membership in a class A; if an individual x satisfies such conditions, we can conclude that x must be an instance of A

Uses for Reasoning

Reasoning support is important for

- checking the consistency of the ontology and the knowledge
- checking for unintended relationships between classes
- automatically classifying instances in classes

Checks like the preceding ones are valuable for

- designing large ontologies, where multiple authors are involved
- integrating and sharing ontologies from various sources

Outline

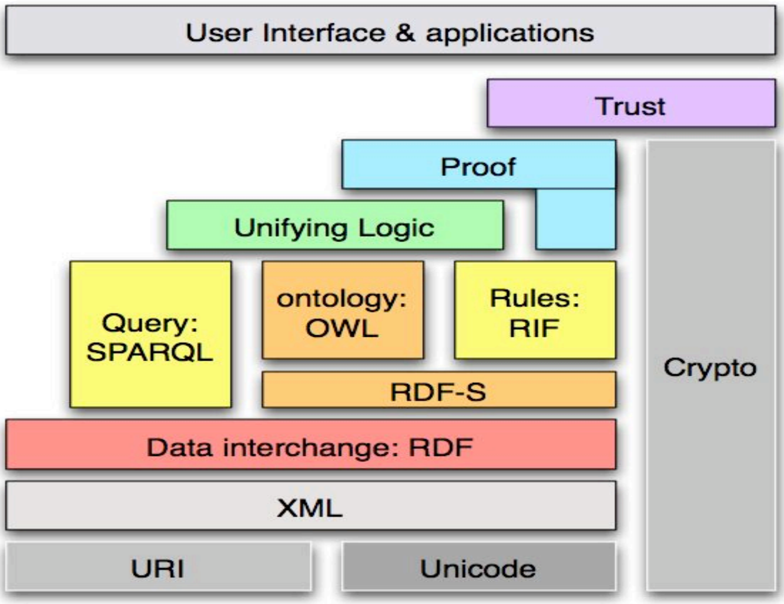
Why the Semantic Web?

What is the Semantic Web?

Semantic Web Layers

Semantic Web Technologies

Issues



http://www.w3.org/2007/03/layerCake.png

Logic, Proof, Trust

Logic

- Jack is an engineer of Scandinavian Airline (SAS).
- Engineers are permanent employees.
- All permanent employees of SAS will get 50% discount for all Radison hotels.
- **THEREFORE**, Jack will get 50% discount for all Radison hotels.

Proof

- SAS' employee document lists Jack as an engineer.
- SAS' employment classification document asserts that engineers are permanent employees.
- Radison's sales procedure asserts that all permanent employees of SAS will get 50% discount for all SAS air tickets.

Trust

- SAS's employee list and classification document are signed by a private key that Radison trusts to make such assertions.
- Radison's sales procedure is trusted.

W3C SW Working Groups

GRDDL Working Group

Semantic Web Education and Outreach (SWEO) Interest Group

Semantic Web Deployment Working Group

RDF Data Access Working Group

Rules Working Group

Semantic Web Health Care and Life Sciences Interest Group

Semantic Web Interest Group

Outline

Why the Semantic Web?

What is the Semantic Web?

Semantic Web Layers

Semantic Web Technologies

Issues

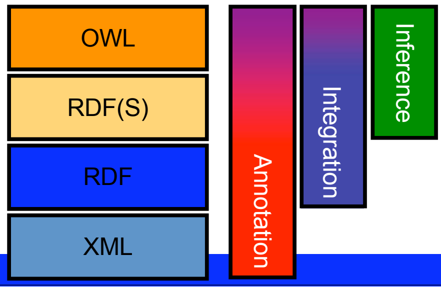
Languages

Work on Semantic Web has concentrated on the definition of a collection or “stack” of languages.

- These languages are then used to support the representation and use of metadata.

Basic machinery to represent the extra semantic information needed for the Semantic Web

- XML
- RDF
- RDF(S)
- OWL
- …



RDF **(Resource Description Framework)**

A W3C’s recommended framework for **describing and modeling Web resources** in terms of named properties and values.

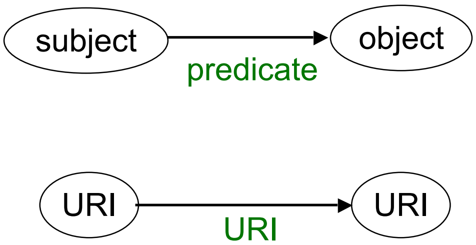
An **RDF statement** is a triple (p, s, o):

- a resource *s* has a property *p* with the value *o*,
- with a graphical notation as: [S] – P –> [O].

Its XML serialisation is also available.

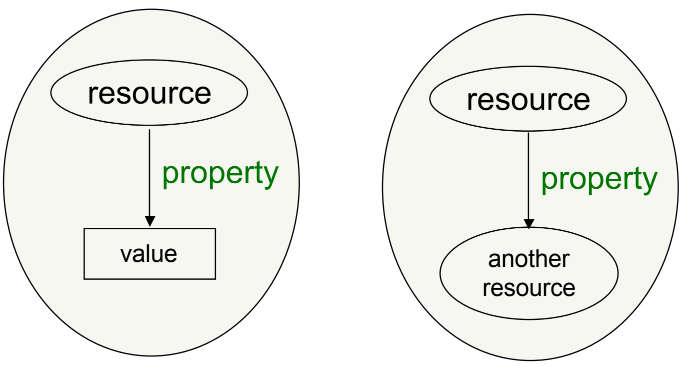
Combining with XML, RDF is **both syntactic and semantic interoperability**.

RDF Statements



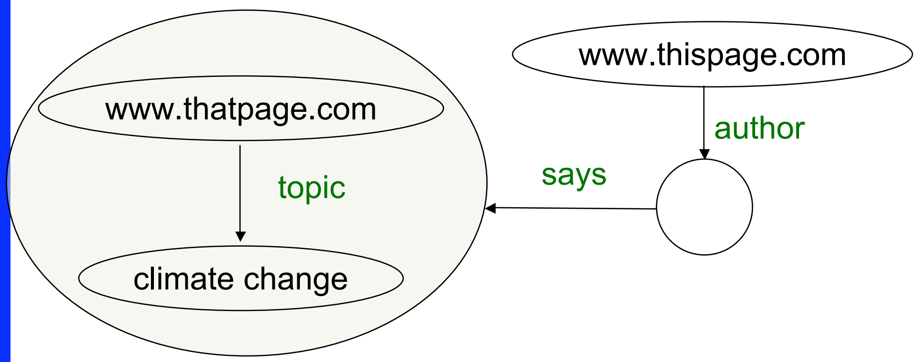
Each of the subject, predicate and object is identified by a unique URI.

RDF Statements

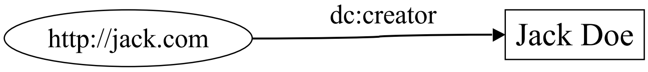


More RDF: Higher Order Statements

“the author of www.thispage.com says: ‘the topic of www.thatpage.com is climate change’ “



RDF Syntaxes and Dialects




```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
 <rdf:Description rdf:about="http://jack.com">
  <dc:creator>Jack Doe</dc:creator>
 </rdf:Description>
</rdf:RDF>
```

RDF/XML

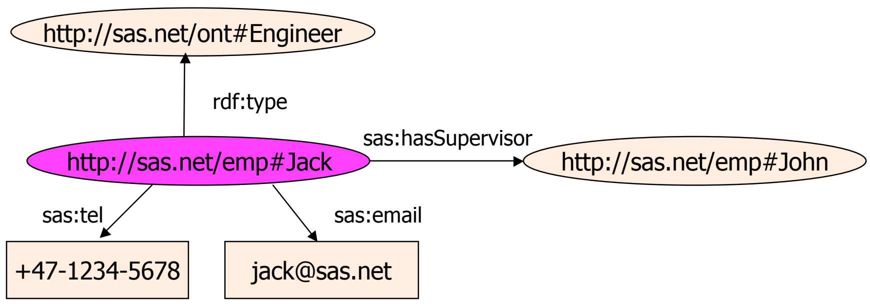
```
@prefix dc:  <http://purl.org/dc/elements/1.1/> .

<http://jack.com>  dc:creator  “Jack Doe" .
```

Notation3

```
<http://jack.com> <http://purl.org/dc/elements/1.1/creator> “Jack Doe"
.
```

N-Triples

RDF Example




```
<sas:Engineer rdf:about="http://sas.net/emp#Jack">
    <sas:tel>+47-1234-5678</sas:tel>
    <sas:email>jack@sas.net</sas:email>
    <sas:hasSupervisor rdf:resource="http://sas.net/emp#John"/>
</sas:Engineer>
```


RDF Schema

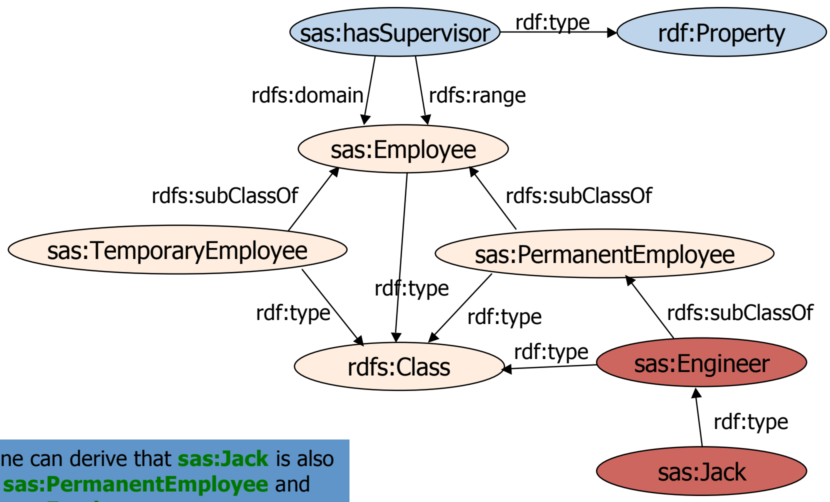
aka. RDF Vocabulary Description Language.

For defining an appropriate RDF vocabulary (classes, properties and constraints) for each specific domain.

Comprises **very limited predefined primitives**: subClassOf, subPropertyOf, domain and range.

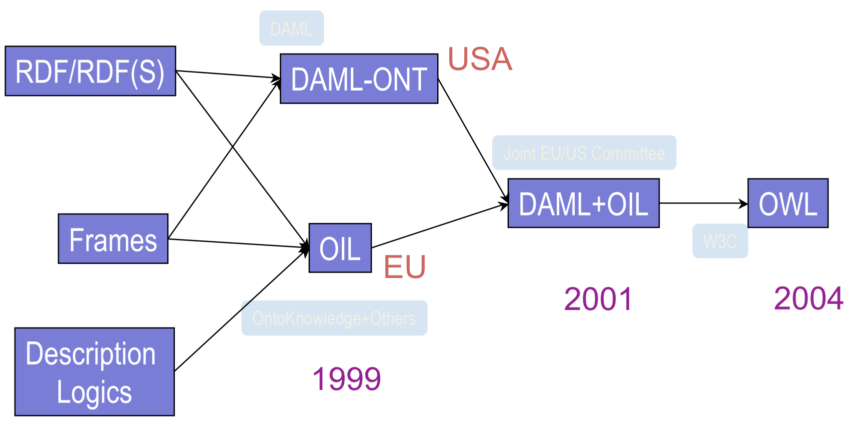
Cannot assert that particular properties are equivalent, transitive, reverse of one another, etc.

RDF Schema Example



One can derive that **sas:Jack** is also a **sas:PermanentEmployee** and a **sas:Employee**.

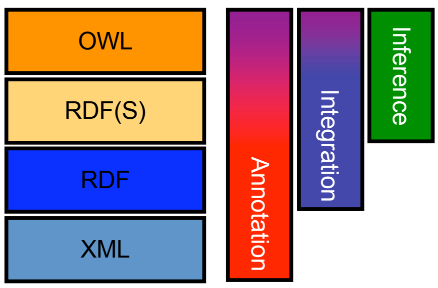
OWL: Web Ontology Language



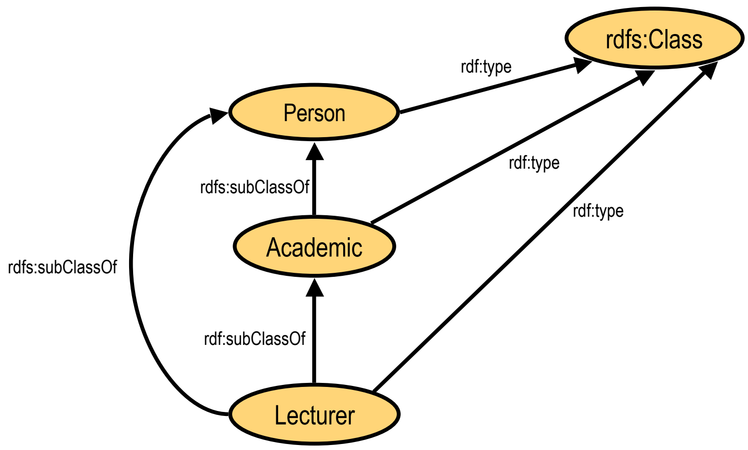
A Printer Ontology – HP Products

```
<owl:Class rdf:ID="hpProduct">
    <owl:intersectionOf>
        <owl:Class rdf:about="#product"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#manufactured-by"/>
          <owl:hasValue>
               <xsd:string rdf:value="Hewlett Packard"/>
          </owl:hasValue>
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
```

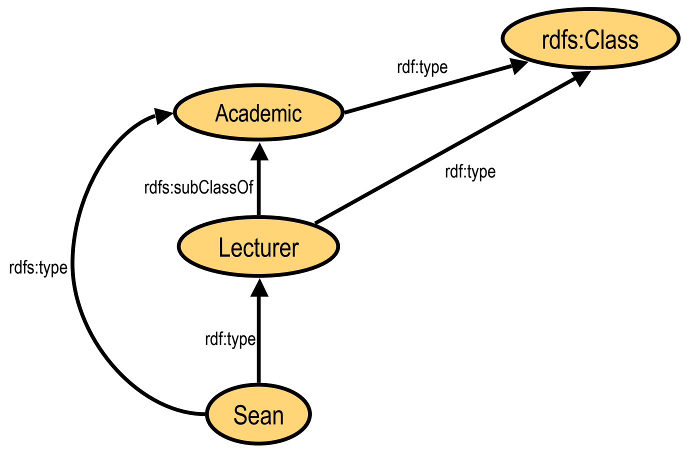
Was’nt RDF(S) enough?



RDF(S) Inference



RDF(S) Inference



Limitations of the Expressive Power of RDF Schema

Local scope of properties

- **rdfs:range** defines the range of a property (e.g. eats) for all classes
- In RDF Schema we cannot declare range restrictions that apply to some classes only
- E.g. we cannot say that cows eat only plants, while other animals may eat meat, too

Limitations of the Expressive Power of RDF Schema (2)

Disjointness of classes

- Sometimes we wish to say that classes are disjoint (e.g. **male** and **female**)

Boolean combinations of classes

- Sometimes we wish to build new classes by combining other classes using union, intersection, and complement
- E.g. **person** is the disjoint union of the classes **male** and **female**

Limitations of the Expressive Power of RDF Schema (3)

Cardinality restrictions

- E.g. a person has exactly two parents, a course is taught by at least one lecturer

Special characteristics of properties

- Transitive property (like “greater than”)
- Unique property (like “is mother of”)
- A property is the inverse of another property (like “eats” and “is eaten by”)

Combining OWL with RDF Schema

Ideally, OWL would extend RDF Schema

- Consistent with the layered architecture of the Semantic Web

But simply extending RDF Schema would work against obtaining expressive power and efficient reasoning

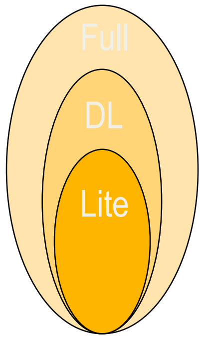
- Combining RDF Schema with logic leads to uncontrollable computational properties

Three Species of OWL

W3C'sWeb Ontology Working Group defined OWL as three different sublanguages:

- OWL Full
- OWL DL
- OWL Lite

Each sublanguage geared toward fulfilling different aspects of requirements



OWL Full – (FOP)

It uses all the OWL languages primitives

It allows the combination of these primitives in arbitrary ways with RDF and RDF Schema

OWL Full is fully upward-compatible with RDF, both syntactically and semantically

OWL Full is so powerful that it is undecidable

- No complete (or efficient) reasoning support

OWL DL

OWL DL (Description Logic) is a sublanguage of OWL Full that restricts application of the constructs from OWL and RDF

- Application of OWL's constructs' to each other is disallowed
- Therefore it corresponds to a well studied description logic

OWL DL permits efficient reasoning support

But we lose full compatibility with RDF:

- Not every RDF document is a legal OWL DL document.
- Every legal OWL DL document is a legal RDF document.

Aside: Description Logics

A family of logic based Knowledge Representation formalisms

- Descendants of semantic networks and KL-ONE
- Describe domain in terms of concepts (classes), roles (relationships) and individuals

Distinguished by:

- Formal semantics (typically model theoretic)
  - Decidable fragments of FOL
- Provision of inference services
  - Sound and complete decision procedures for key problems
  - Implemented systems (highly optimised)

OWL Lite

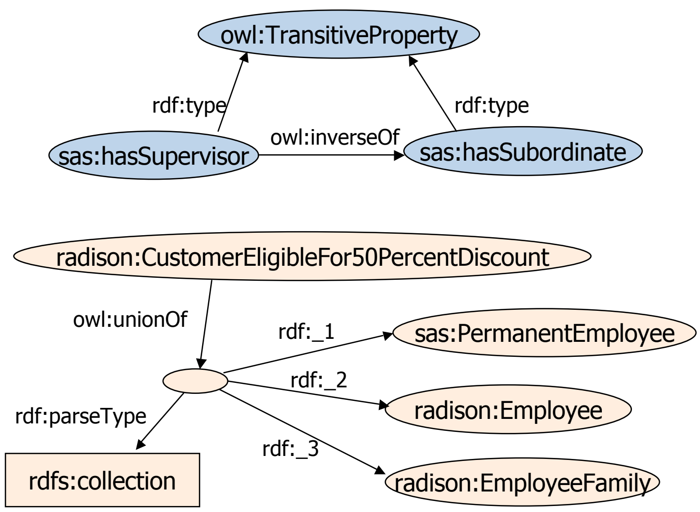
An even further restriction limits OWL DL to a subset of the language constructs
- E.g., OWL Lite excludes enumerated classes, disjointness statements, and arbitrary cardinality.

The advantage of this is a language that is easier to
- grasp, for users
- implement, for tool builders

The disadvantage is restricted expressivity

practically not used

OWL Example



RDFa – Embedding RDF in XHTML

Current web pages written in HTML contain significant **inherent structured data**

RDFa is a syntax that expresses this structured data using a set of elements and attributes that **embed RDF in HTML**

Application: e.g., an event on a web page can be **directly imported** into a user's desktop calendar

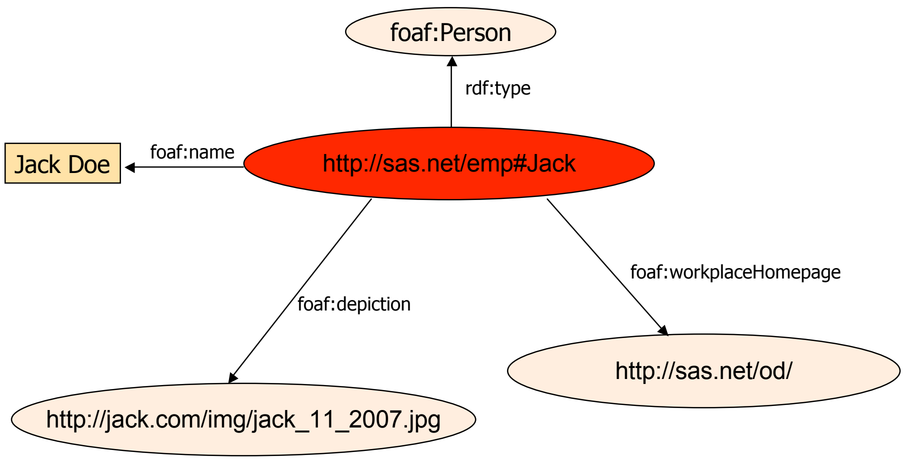
Important goal: achieve RDF embedding without repeating existing HTML content when that content *is* the structured data

Source: http://www.w3.org/TR/xhtml-rdfa-primer

RDFa Example

```
<div href="foaf:Person" rel="rdf:type"
   about=" http://sas.net/emp#Jack ">

  My name is
  <span property="foaf:name">Jack Doe</span>
  and
  I work at <a href="http://sas.net/od/"
  rel="foaf:workplaceHomepage">Operation Department</a>
  as a pilot.

  <div>
    <img src="http://jack.com/img/jack_11_2007.jpg"
      rel="foaf:depiction" alt="Picture of John"/>
  </div>

</div>
```

RDFa Example



Microformats

Microformats are used to enrich the semantics of web documents

Principles

- Something that web developers can use immediately. A "Semantic Web" now.
- No change to existing set of HTML tags

Community-driven standard semantic Labels

Result: Bottom-up, grassroot semantics of whole web

Microformat Example

```
<html>
    <head> … </head>
    <body>
      …
        <div class="vcard">
            <div class="fn n">
                <span class="given-name">Jack</span>
                <span class="family-name:>Doe</span>
            </div>
            <div class="org">Scandinavian Airlines System</div>
            <div class="adr">
                <span class="street-address">2061 Gardermoen</span>
                <span class="locality">Oslo</span>
                <span class="postal-code">NO0166</span>
            </div>
            <div class="tel">
                <span class="type">work</span>
                <span class="value">+47-1234-5678</span>
            </div>
        </div>
      …
    </body>
</html>
```


GRDDL

A framework to extract RDF data from XML documents using, e.g., XSLT

Gleaning Resource Descriptions from Dialects of Languages (GRDDL)

– http://www.w3.org/TR/grddl/

GRDDL Primer

– http://www.w3.org/TR/grddl-primer/

GRDDL Example

book.html

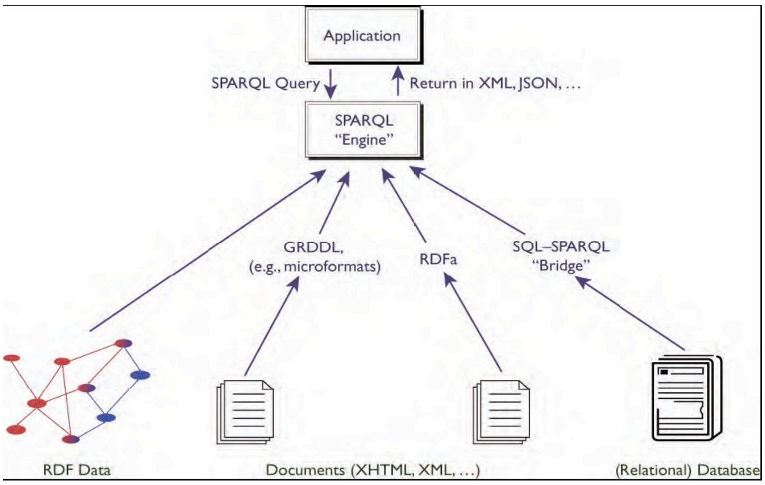
```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head profile="http://www.w3.org/2003/g/data-view">
    <title>Lonely Universe</title>
    <link rel="transformation"  href="http://www.w3.org/2000/06/dc-extract/dc-extract.xsl" />
    <meta name="DC.Creator"  content="Jack Doe" />
    <meta name="DC.Subject"  content="Hotel; Guest House; Beach; Mountain; Food" />...
  </head>...
</html>
```

transformation

RDF

```
<rdf:Description rdf:about="">
<dc:creator>Jack Doe </dc:creator>
<dc:subject>Hotel; Guest House; Beach; Mountain; Food </dc:subject>
</rdf:Description>
```


SPARQL-based Integration



Source: http://www.net.intap.or.jp/INTAP/s-web/data/18-semanticweb-report.pdf

Issues: Metadata

There will be lots of metadata.
How to create metadata?

- metadata schema design
- manual
  - text/ontology editors
  - online forms
  - (semantic) wikis
- (semi)-automated
  - GRDDL
  - information extraction
  - database/spreadsheet conversion
- embedding metadata in data files
- relationships with existing ontologies

Issues: Metadata

How to manage metadata?

- storage
- constraints and consistency
- updating and version control
- querying and retrieval
- trust and privacy

How to share and reuse metadata?

- Application profiles and their schemas
- Metadata integration

Issues: Ontologies

How to create ontologies?
- ontology design
- ontology engineering
- manual
- (semi)-automated
  - ontology learning

How to manage ontologies?
- storage
- constraints representation and computation
- updating and version control
- querying and retrieval
  - Ranking

How to deal with many ontologies?
- ontology mapping/aligning
- ontology integration

Issues: Representation and Computation

How to represent SW data which is large, inconsistent and heterogeneous?

How to represent constraints, rules and regulations?

How to perform computation?
- Is logical inference appropriate?
  - completeness?
  - soundness?
- any other more generic, powerful computational mechanisms?