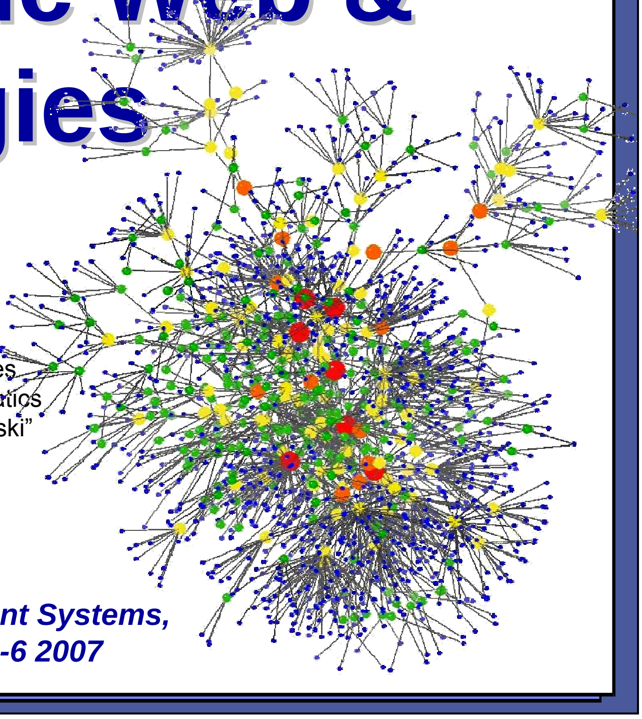
Semantic Web & Ontologies



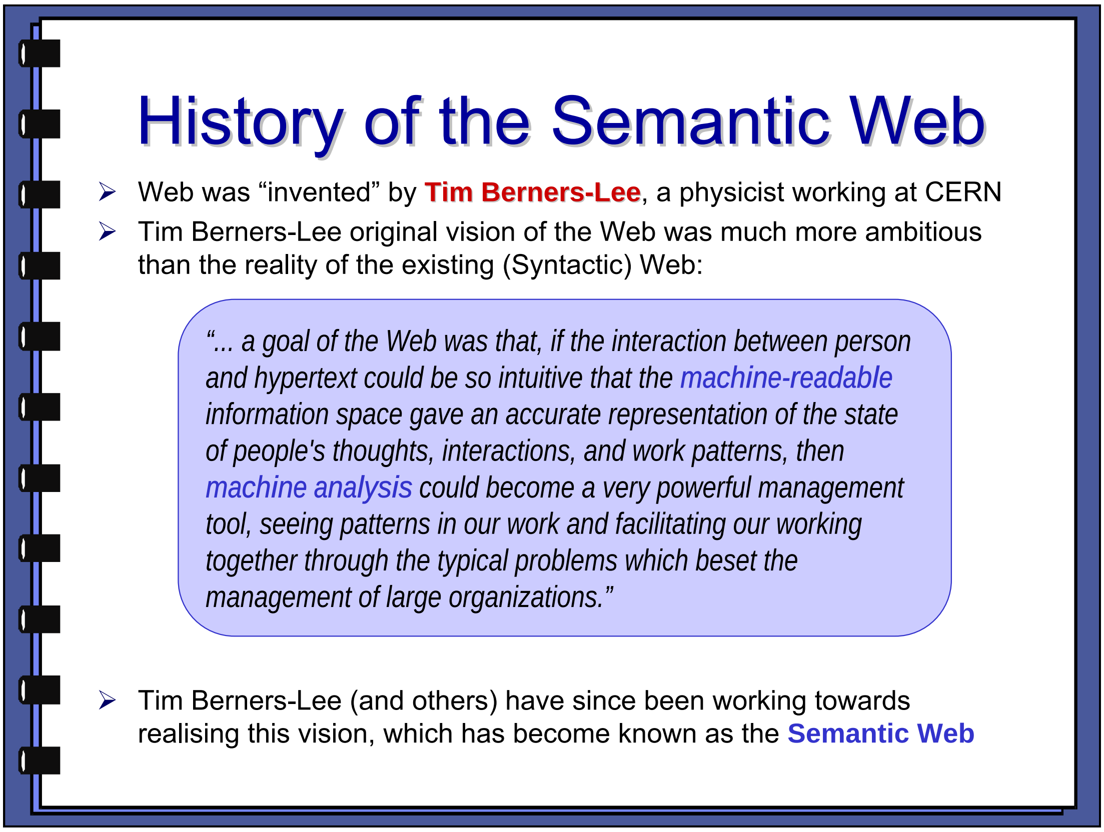
Svetla Boytcheva
Department of Computer Sciences
Faculty of Mathematics and Informatics
Sofia University “St. Kliment Ohridski”
Bulgaria

Summer School on Intelligent Systems, Nicosia, Cyprus, July 2-6 2007

Outline

- Semantic Web
  - History
  - Syntactic Web
  - Goals of The Semantic Web
  - Foundation
  - Features
- Ontologies
  - Basic Characteristics and Types
  - Languages for representing Ontologies
- Semantic Web & Ontologies Applications
- Future and Challanges

The Semantic Web



History of the Semantic Web

- Web was “invented” by **Tim Berners-Lee**, a physicist working at CERN
- Tim Berners-Lee original vision of the Web was much more ambitious than the reality of the existing (Syntactic) Web:

> *“... a goal of the Web was that, if the interaction between person and hypertext could be so intuitive that the machine-readable information space gave an accurate representation of the state of people's thoughts, interactions, and work patterns, then machine analysis could become a very powerful management tool, seeing patterns in our work and facilitating our working together through the typical problems which beset the management of large organizations.”*

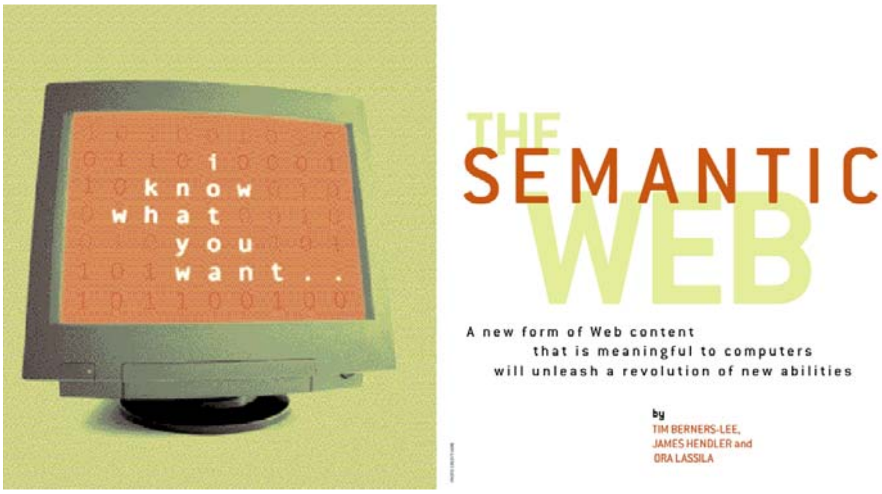
- Tim Berners-Lee (and others) have since been working towards realising this vision, which has become known as the **Semantic Web**

Semantic Web

> "… a plan for achieving a set of connected applications for data on the Web in such a way as to form a consistent logical web of data …"

> "… 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, Tim; Hendler, James and Lassila, Ora
"The Semantic Web", Scientific American, May 2001, p. 29-37.



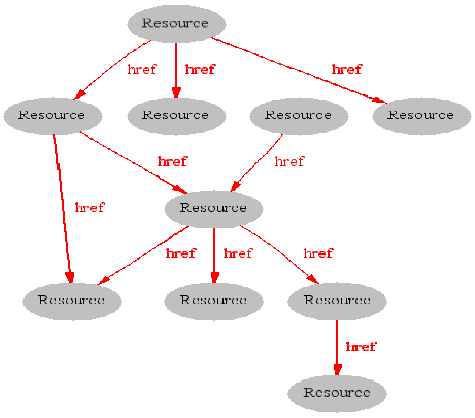
THE SEMANTIC WEB

A new form of Web content
that is meaningful to computers
will unleash a revolution of new abilities

by
TIM BERNERS-LEE,
JAMES HENDLER and
ORA LASSILA

- ➢ Realising the complete "vision" is too hard for now
- ➢ But we can make a start by adding semantic annotation to web resources

the *Syntactic* Web



[Hendler & Miller 02]

The Syntactic Web is…

- **A hypermedia, a digital library**
  - A library of documents called (web pages) interconnected by a hypermedia of links
- **A database, an application platform**
  - A common portal to applications accessible through web pages, and presenting their results as web pages
- **A platform for multimedia**
  - BBC Radio 4 anywhere in the world! Terminator 3 trailers!
- **A naming scheme**
  - Unique identity for those documents

[Goble 03]

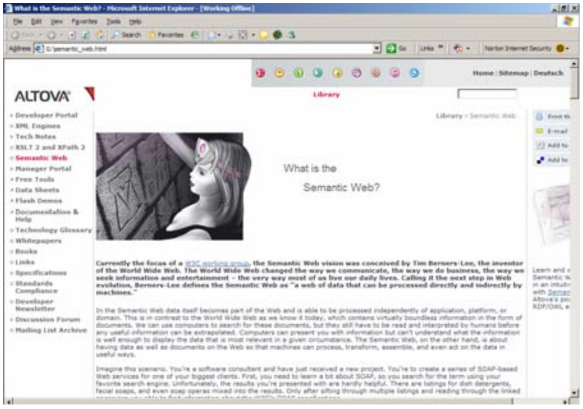
i.e. the Syntactic Web is…

- **A place where**
  - **computers do the presentation (easy) and**
  - **people do the linking and interpreting (hard).**
- ***Why not get computers to do more of the hard work?***

[Goble 03]

What is the Problem?

- Consider a typical web page:



Currently the focus of a W3C working group, the Semantic Web vision was conceived by Tim Berners-Lee, the inventor of the World Wide Web. The World Wide Web changed the way we communicate, the way we do business, the way we seek information and entertainment – the very way most of us live our daily lives. Calling it the next step in Web evolution, Berners-Lee defines the Semantic Web as "a web of data that can be processed directly and indirectly by machines."

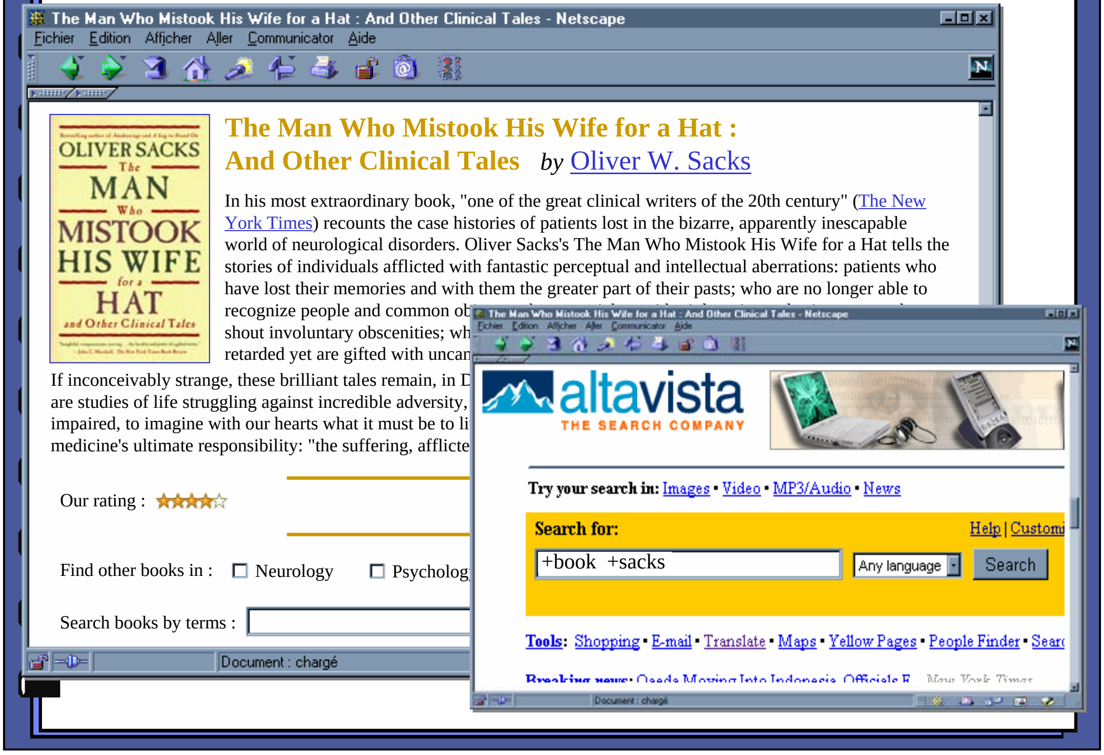
In the Semantic Web data itself becomes part of the Web and is able to be processed independently of application, platform, or domain. This is in contrast to the World Wide Web as we know it today, which contains virtually boundless information in the form of documents. We can use computers to search for these documents, but they still have to be read and interpreted by humans before any useful information can be extrapolated. Computers can present you with information but can't understand what the information is well enough to display the data that is most relevant in a given circumstance. The Semantic Web, on the other hand, is about having data as well as documents on the Web so that machines can process, transform, assemble, and even act on the data in useful ways.

Imagine this scenario. You're a software consultant and have just received a new project. You're to create a series of SOAP-based Web services for one of your biggest clients. First, you need to learn a bit about SOAP, so you search for the term using your favorite search engine. Unfortunately, the results you're presented with are hardly helpful. There are listings for dish detergents, facial soaps, and even soap operas mixed into the results. Only after sifting through multiple listings and reading through the linked

Learn and
Semantic W
in an intuit
with Seman
Altova's pr
RDF/OWL e

- Markup consists of:
  - rendering information (e.g., font size and colour)
  - Hyper-links to related content
- Semantic content is accessible to humans but not (easily) to computers…

Web to humans



The Man Who Mistook His Wife for a Hat : And Other Clinical Tales - Netscape

The Man Who Mistook His Wife for a Hat : And Other Clinical Tales *by* Oliver W. Sacks

In his most extraordinary book, "one of the great clinical writers of the 20th century" (The New York Times) recounts the case histories of patients lost in the bizarre, apparently inescapable world of neurological disorders. Oliver Sacks's The Man Who Mistook His Wife for a Hat tells the stories of individuals afflicted with fantastic perceptual and intellectual aberrations: patients who have lost their memories and with them the greater part of their pasts; who are no longer able to recognize people and common ob
shout involuntary obscenities; wh
retarded yet are gifted with uncan

If inconceivably strange, these brilliant tales remain, in D
are studies of life struggling against incredible adversity,
impaired, to imagine with our hearts what it must be to li
medicine's ultimate responsibility: "the suffering, afflicte

Our rating : ★★★★☆

Find other books in : □ Neurology □ Psycholog

Search books by terms :

Document : chargé

The Man Who Mistook His Wife for a Hat : And Other Clinical Tales - Netscape

Try your search in: Images • Video • MP3/Audio • News

Search for: Help | Customi

+book +sacks

Any language

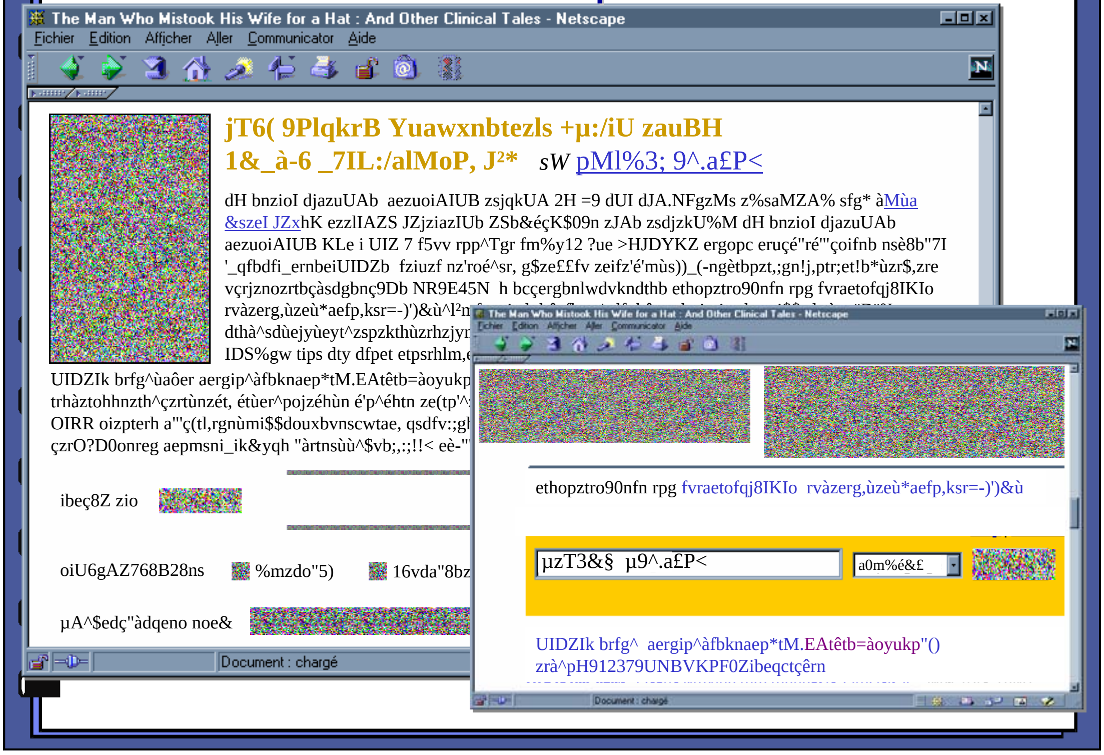
Search

Tools: Shopping • E-mail • Translate • Maps • Yellow Pages • People Finder • Searc

Breaking news: Qaeda Moving Into Indonesia, Officials F... New York Times

Document : chargé

Web to computers...



The Man Who Mistook His Wife for a Hat : And Other Clinical Tales - Netscape

jT6(9PlqkrB Yuawxnbtezls +µ:/iU zauBH 1&_à-6 _7IL:/alMoP, J²* *sW* pMl%3; 9^.a£P<

dH bnzioI djazuUAb aezuoiAIUB zsjqkUA 2H =9 dUI dJA.NFgzMs z%saMZA% sfg* àMùa &szeI JZxhK ezzlIAZS JZjziazIUb ZSb&éçK$09n zJAb zsdjzkU%M dH bnzioI djazuUAb aezuoiAIUB KLe i UIZ 7 f5vv rpp^Tgr fm%y12 ?ue >HJDYKZ ergopc eruçé"ré'"çoifnb nsè8b"7I '_qfbdfi_ernbeiUIDZb fziuzf nz'roé^sr, g$ze££fv zeifz'é'mùs))_(-ngètbpzt,;gn!j,ptr;et!b*ùzr$,zre vçrjznozrtbçàsdgbnç9Db NR9E45N h bcçergbnlwdvkndthb ethopztro90nfn rpg fvraetofqj8IKIo rvàzerg,ùzeù*aefp,ksr=-)')&ù^l²n
dthà^sdùejyùeyt^zspzkthùzrhzjyr
IDS%gw tips dty dfpet etpsrhlm,e

UIDZIk brfg^ùaôer aergip^àfbknaep*tM.EAtêtb=àoyukp
trhàztohhnzth^çzrtùnzét, étùer^pojzéhùn é'p^éhtn ze(tp'^
OIRR oizpterh a'"ç(tl,rgnùmi$$douxbvnscwtae, qsdfv:;gl
çzrO?D0onreg aepmsni_ik&yqh "àrtnsùù^$vb;,:;!!< eè-"

ibeç8Z zio ▨

oiU6gAZ768B28ns ▨ %mzdo"5) ▨ 16vda"8bz

µA^$edç"àdqeno noe& ▨

Document : chargé

The Man Who Mistook His Wife for a Hat : And Other Clinical Tales - Netscape

ethopztro90nfn rpg fvraetofqj8IKIo rvàzerg,ùzeù*aefp,ksr=-)')&ù

µzT3&§ µ9^.a£P<

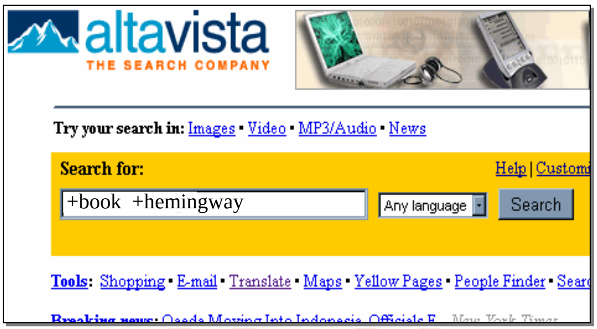
a0m%é&£

UIDZIk brfg^ aergip^àfbknaep*tM.EAtêtb=àoyukp"()
zrà^pH912379UNBVKPF0Zibeqctçêrn

Document : chargé

Example of a search on the Web

- "What are the books from Hemingway?"



Try your search in: Images • Video • MP3/Audio • News

Search for: +book +hemingway

Tools: Shopping • E-mail • Translate • Maps • Yellow Pages • People Finder • Sear

Breaking news: Qaeda Moving Into Indonesia, Officials F... New York Times

Noise ≠ Precision

Missed ≠ Recall

Nice pubs in Nice

The Old Book
12, R. Victor Hugo

The White Swan
3 Av. Hemingway

The Horseshoe

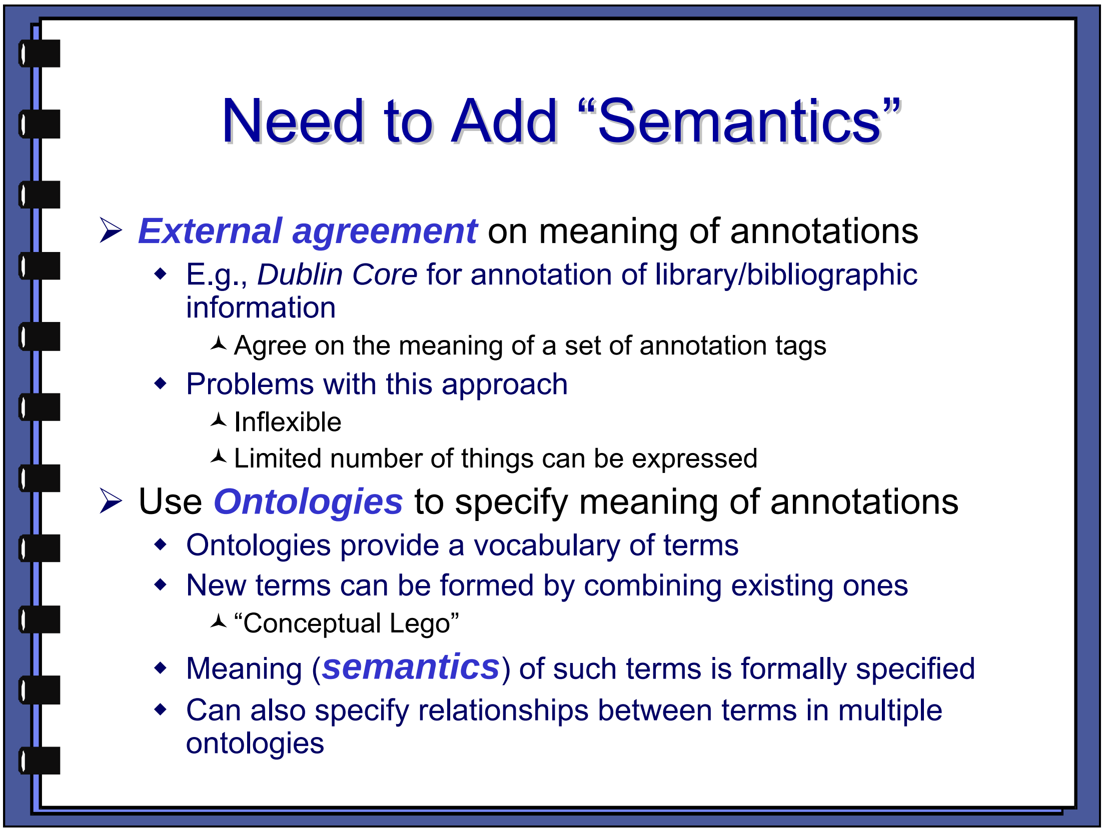
Summary of the novel

"The Old Man And The Sea"
by Ernest Hemingway

This new edition starts with a large historical introduction of the work

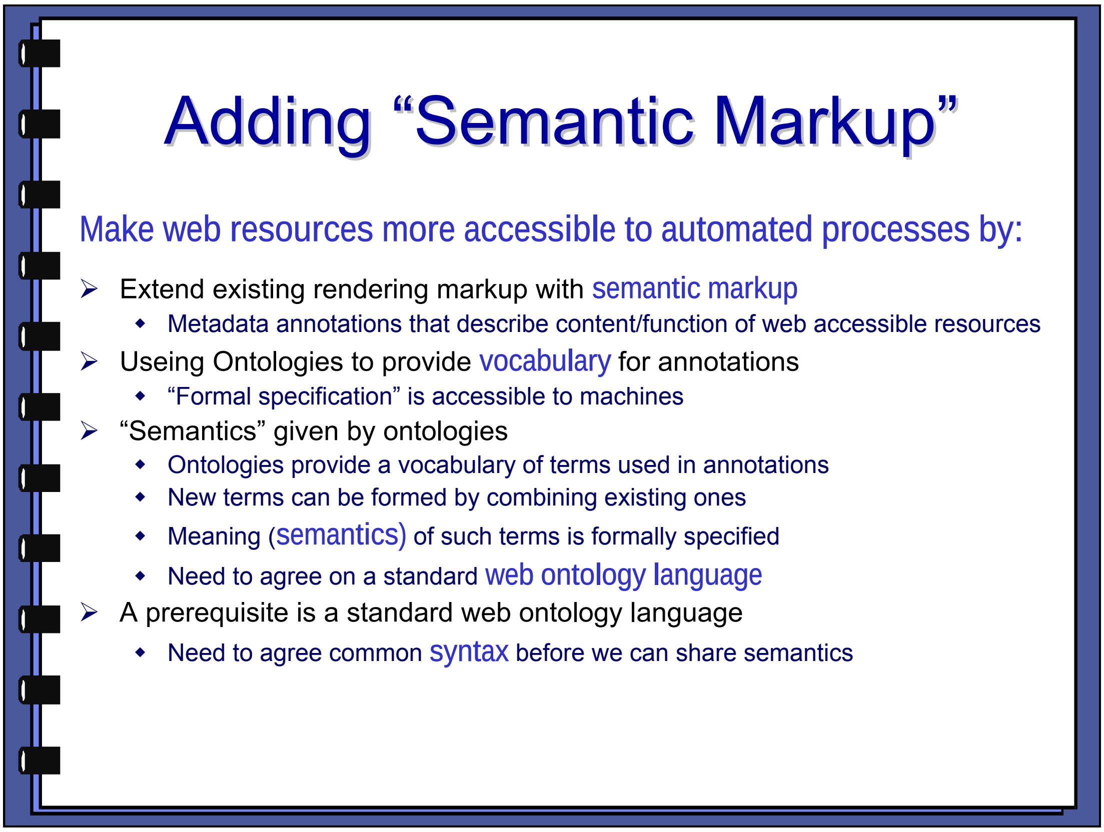
Impossible (?) using the Syntactic Web…

- Complex queries involving ***background knowledge***
  - Find information about "animals that use sonar but are not either bats or dolphins"
- Locating information in ***data repositories***
  - Travel enquiries
  - Prices of goods and services
  - Results of human genome experiments
- Finding and using "***web services***"
  - Visualise surface interactions between two proteins
- Delegating complex tasks to web "***agents***"
  - Book me a holiday next weekend somewhere warm, not too far away, and where they speak French or English



Need to Add "Semantics"

- ***External agreement*** on meaning of annotations
  - E.g., *Dublin Core* for annotation of library/bibliographic information
    - Agree on the meaning of a set of annotation tags
  - Problems with this approach
    - Inflexible
    - Limited number of things can be expressed
- Use ***Ontologies*** to specify meaning of annotations
  - Ontologies provide a vocabulary of terms
  - New terms can be formed by combining existing ones
    - "Conceptual Lego"
  - Meaning (***semantics***) of such terms is formally specified
  - Can also specify relationships between terms in multiple ontologies



Adding "Semantic Markup"

Make web resources more accessible to automated processes by:

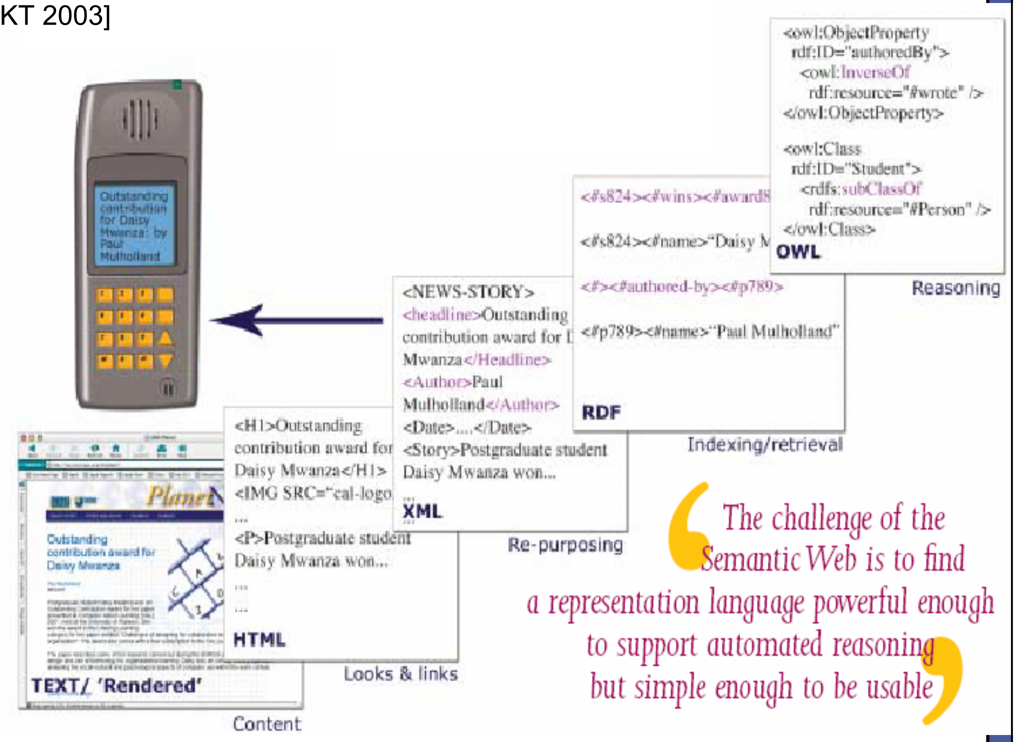
- Extend existing rendering markup with semantic markup
  - Metadata annotations that describe content/function of web accessible resources
- Useing Ontologies to provide vocabulary for annotations
  - "Formal specification" is accessible to machines
- "Semantics" given by ontologies
  - Ontologies provide a vocabulary of terms used in annotations
  - New terms can be formed by combining existing ones
  - Meaning (semantics) of such terms is formally specified
  - Need to agree on a standard web ontology language
- A prerequisite is a standard web ontology language
  - Need to agree common syntax before we can share semantics

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 co-operation.“

[Berners-Lee et al., 2001]

[AKT 2003]


Outstanding contribution for Daisy Mwanza: by Paul Mulholland
Outstanding contribution award for Daisy Mwanza
TEXT/ 'Rendered'
Content
HTML
Looks & links
XML
Re-purposing
RDF
Indexing/retrieval
OWL
Reasoning

```
<H1>Outstanding
contribution award for
Daisy Mwanza</H1>
<IMG SRC="cal-logo
...
<P>Postgraduate student
Daisy Mwanza won...
...
...
```

```
<NEWS-STORY>
<headline>Outstanding
contribution award for I
Mwanza</Headline>
<Author>Paul
Mulholland</Author>
<Date>....</Date>
<Story>Postgraduate student
Daisy Mwanza won...
...
```

```
<#s824><#wins><#award8
<#s824><#name>"Daisy M
<#><#authored-by><#p789>
<#p789><#name>"Paul Mulholland"
```

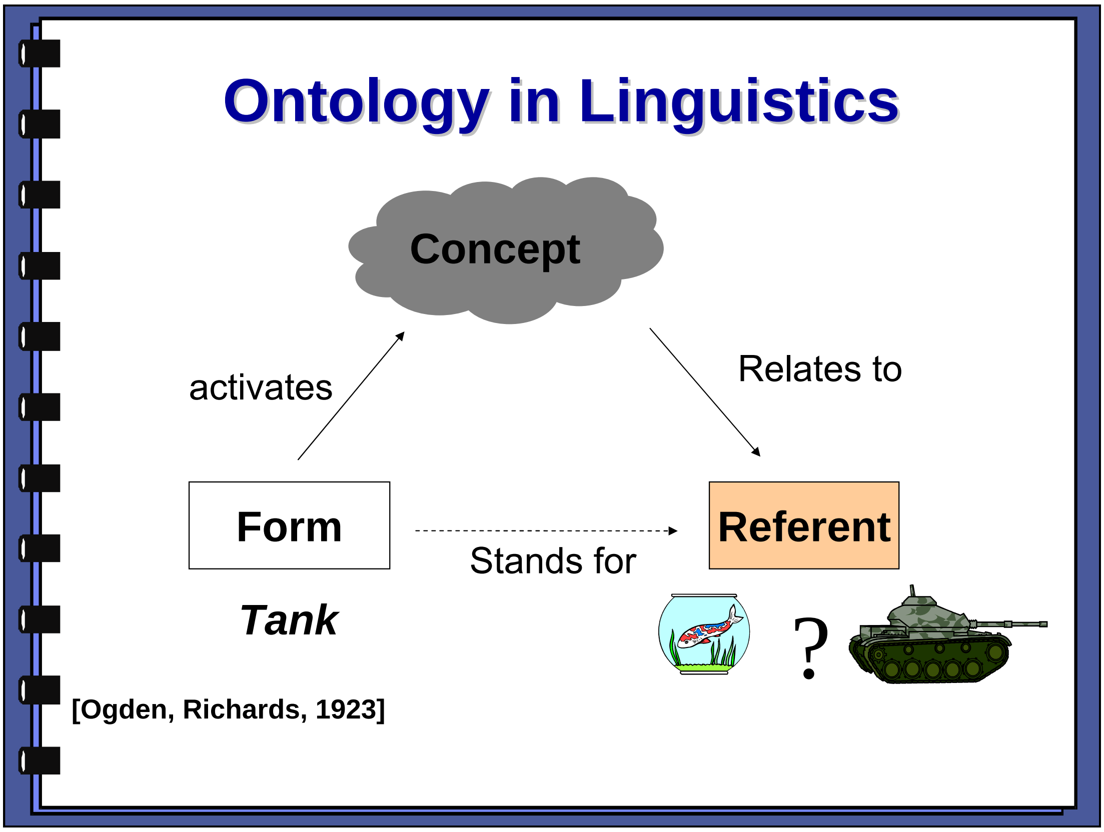
```
<owl:ObjectProperty
 rdf:ID="authoredBy">
  <owl:InverseOf
   rdf:resource="#wrote" />
</owl:ObjectProperty>

<owl:Class
 rdf:ID="Student">
  <rdfs:subClassOf
   rdf:resource="#Person" />
</owl:Class>
```

"The challenge of the Semantic Web is to find a representation language powerful enough to support automated reasoning but simple enough to be usable"

Ontologies

Ontology: Origins and History

- a philosophical discipline
- a branch of philosophy that deals with the nature and the organisation of reality
- Science of Being (Aristotle, Metaphysics, IV, 1)
- Tries to answer the questions:
  - *What characterizes being?*
  - *Eventually, what is being?*



Ontology in Linguistics

Concept

activates

Relates to

Form

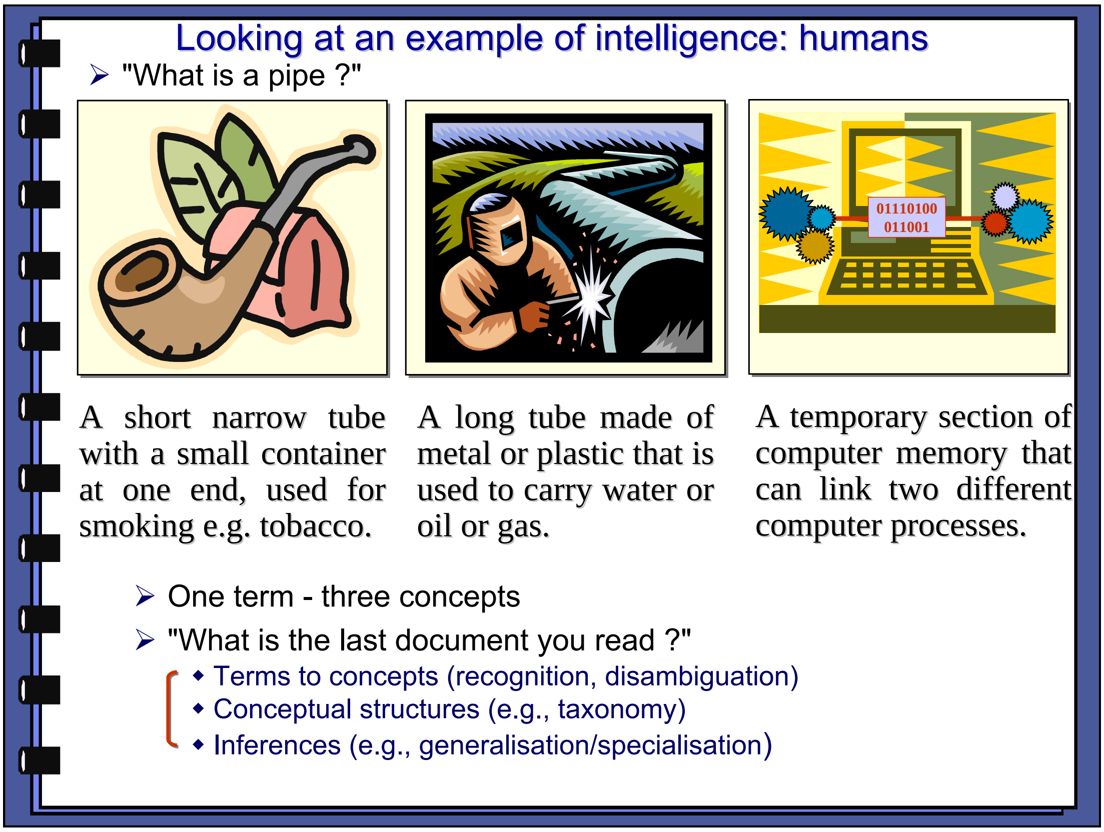
Stands for

Referent

Tank

?

[Ogden, Richards, 1923]



Looking at an example of intelligence: humans

- "What is a pipe ?"

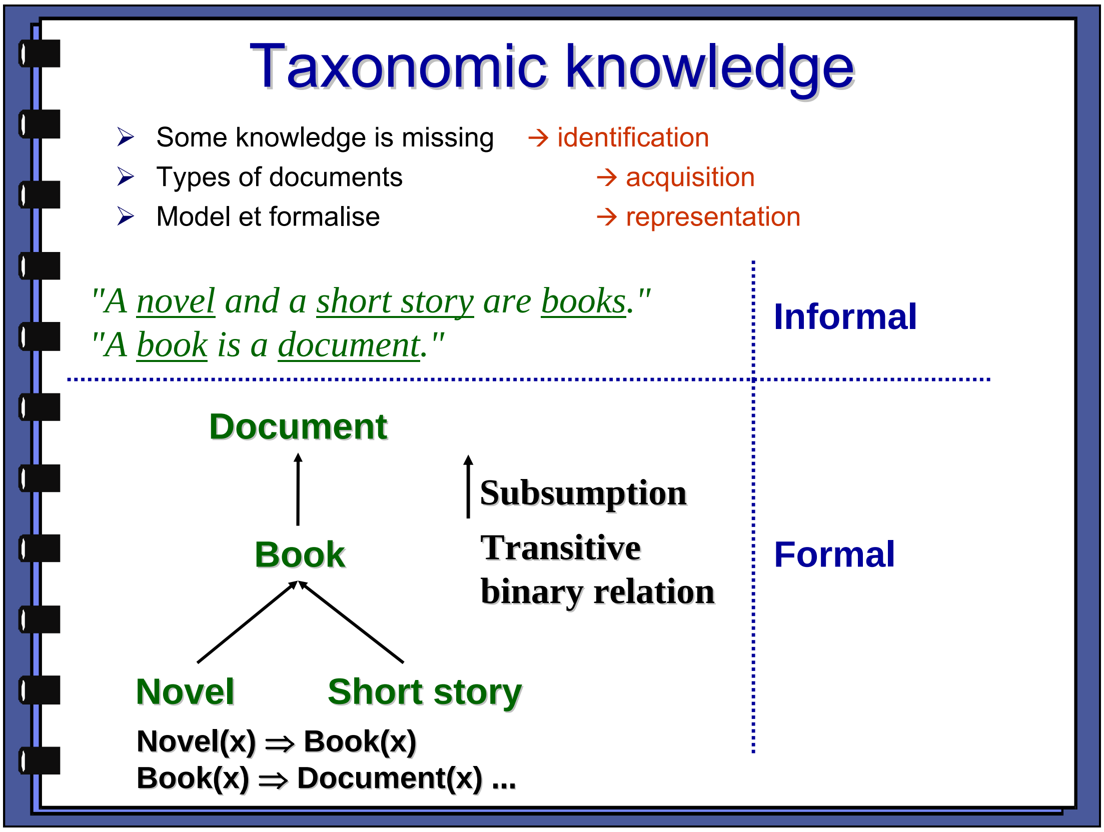
A short narrow tube with a small container at one end, used for smoking e.g. tobacco.

A long tube made of metal or plastic that is used to carry water or oil or gas.

A temporary section of computer memory that can link two different computer processes.

01110100
011001

- One term - three concepts
- "What is the last document you read ?"
  - Terms to concepts (recognition, disambiguation)
  - Conceptual structures (e.g., taxonomy)
  - Inferences (e.g., generalisation/specialisation)



Taxonomic knowledge

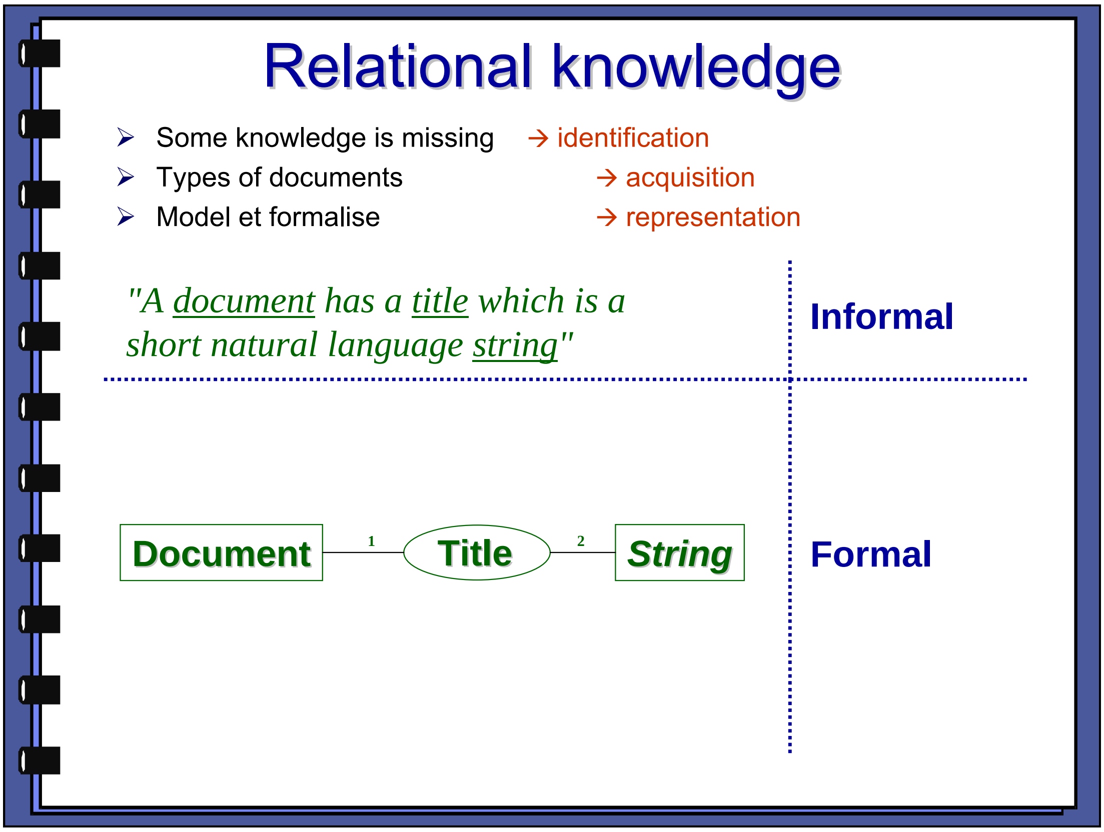
- ➢ Some knowledge is missing → identification
- ➢ Types of documents → acquisition
- ➢ Model et formalise → representation

"A <u>novel</u> and a <u>short story</u> are <u>books</u>."
"A <u>book</u> is a <u>document</u>."

Informal

Novel(x) $\Rightarrow$ Book(x)
Book(x) $\Rightarrow$ Document(x) ...

Formal



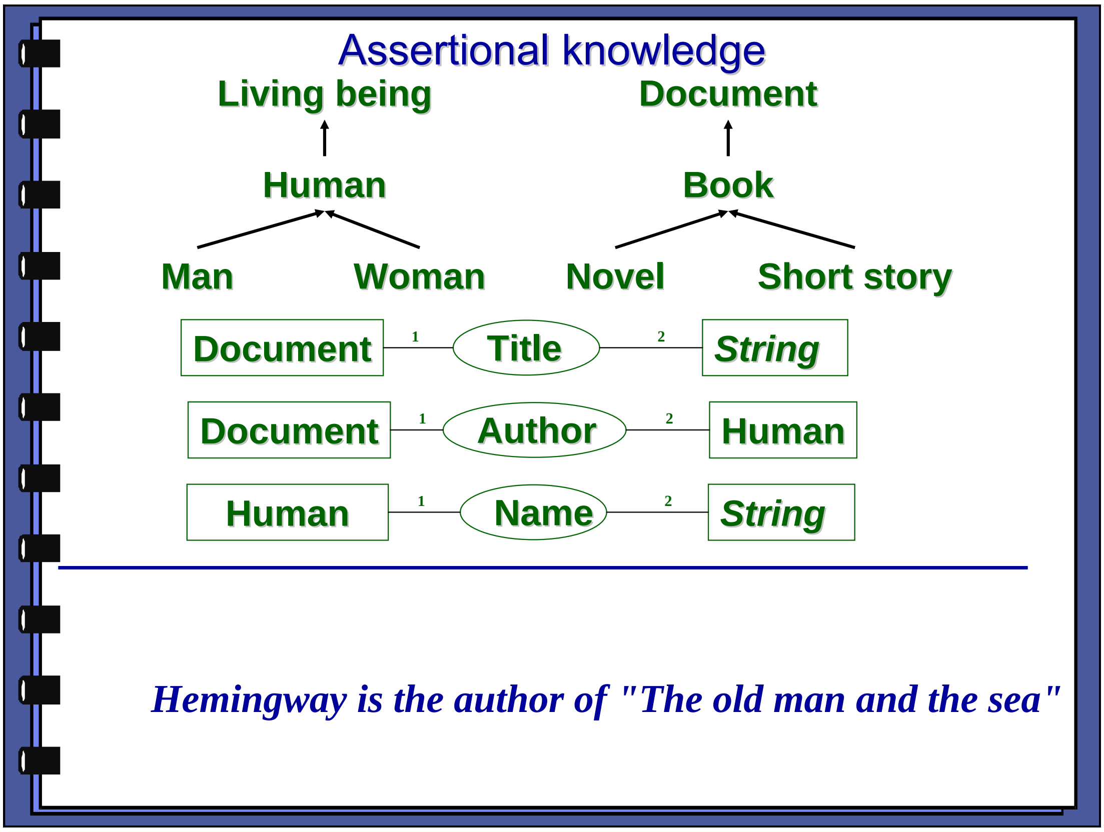
Relational knowledge

- ➢ Some knowledge is missing → identification
- ➢ Types of documents → acquisition
- ➢ Model et formalise → representation

"A <u>document</u> has a <u>title</u> which is a short natural language <u>string</u>"

Informal

Formal



Assertional knowledge

Living being

Document

Human

Book

Man

Woman

Novel

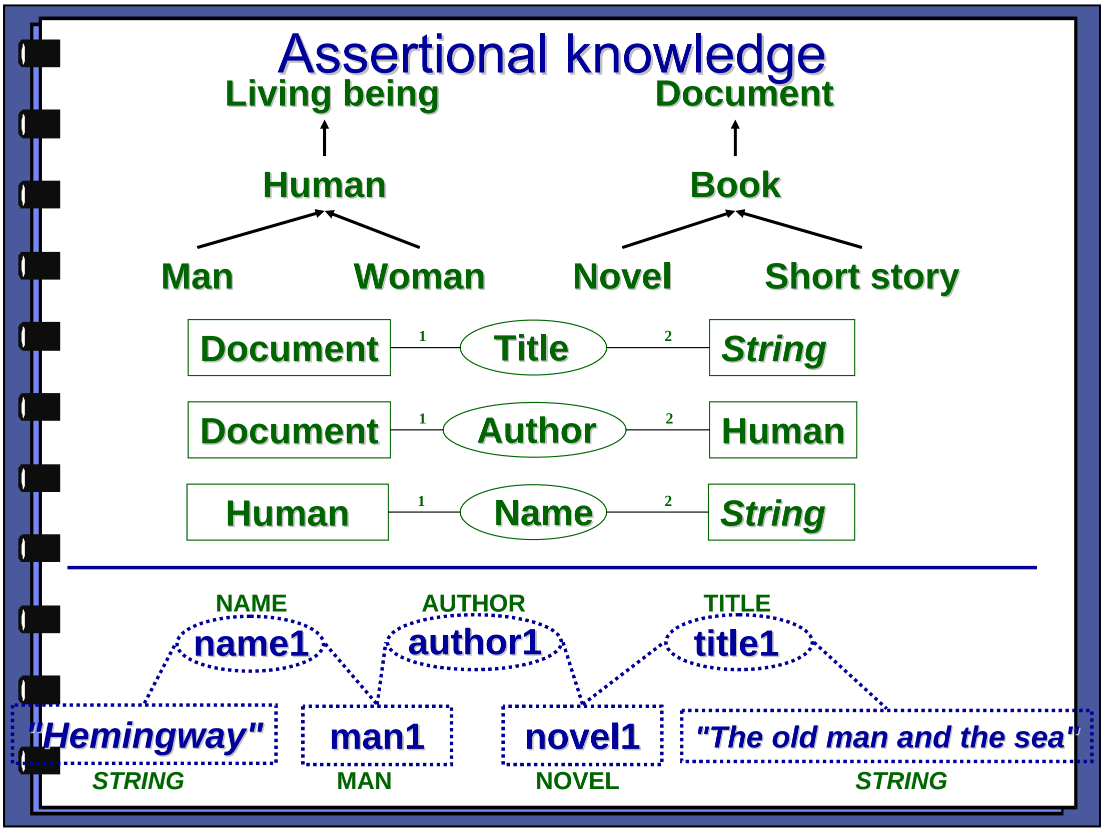
Short story

Document 1 Title 2 *String*

Document 1 Author 2 Human

Human 1 Name 2 *String*

Hemingway is the author of "The old man and the sea"



Assertional knowledge

Living being

Document

Human

Book

Man

Woman

Novel

Short story

Document 1 Title 2 *String*

Document 1 Author 2 Human

Human 1 Name 2 *String*

NAME name1

AUTHOR author1

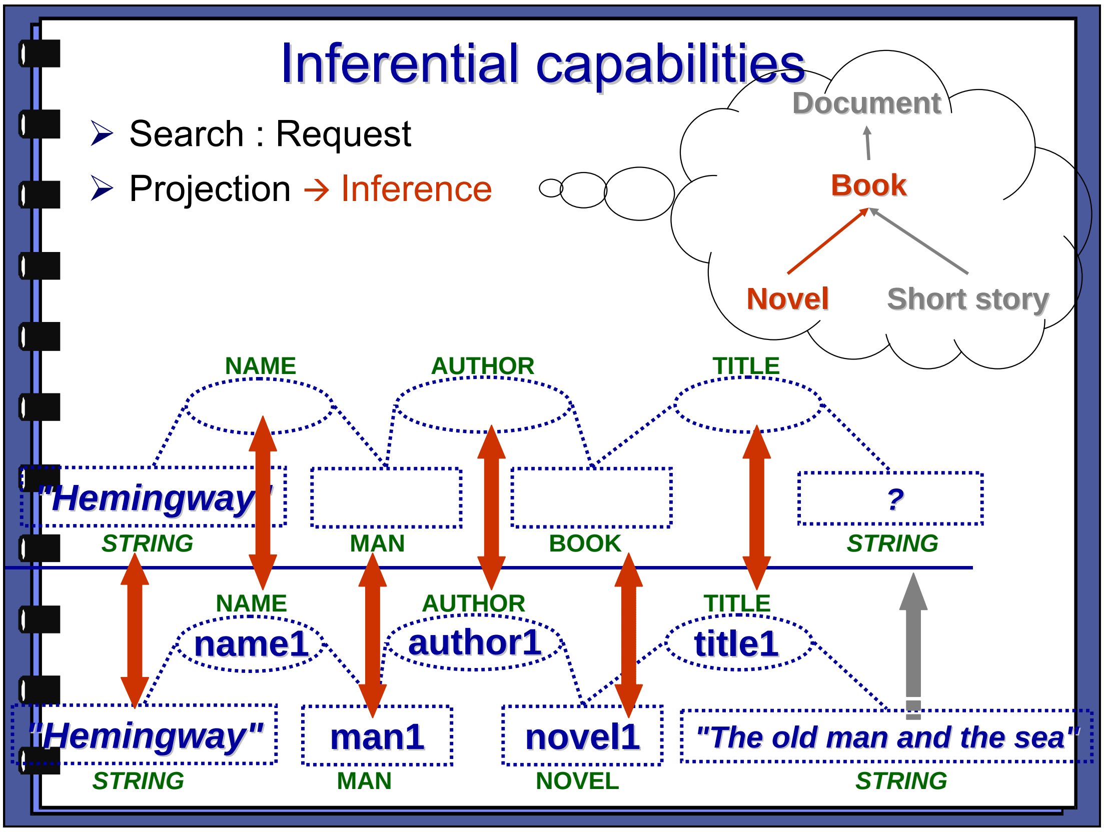
TITLE title1

"Hemingway" *STRING*

man1 MAN

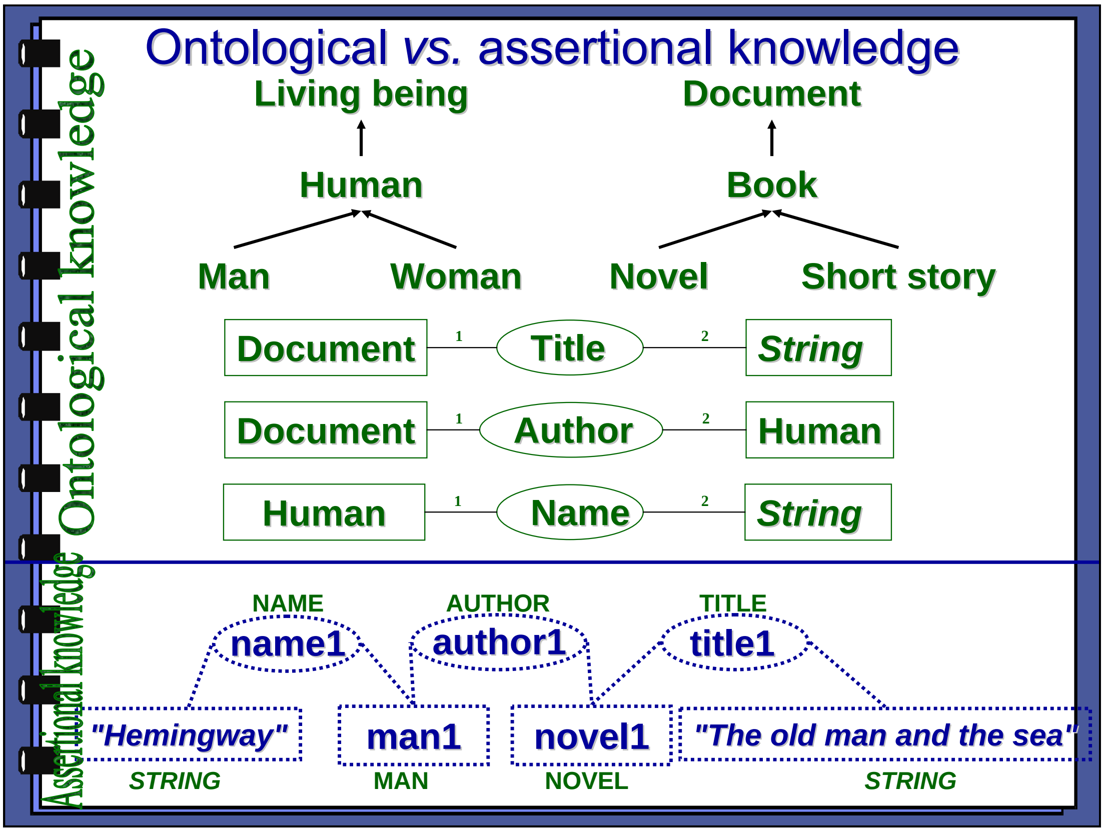
novel1 NOVEL

"The old man and the sea" *STRING*



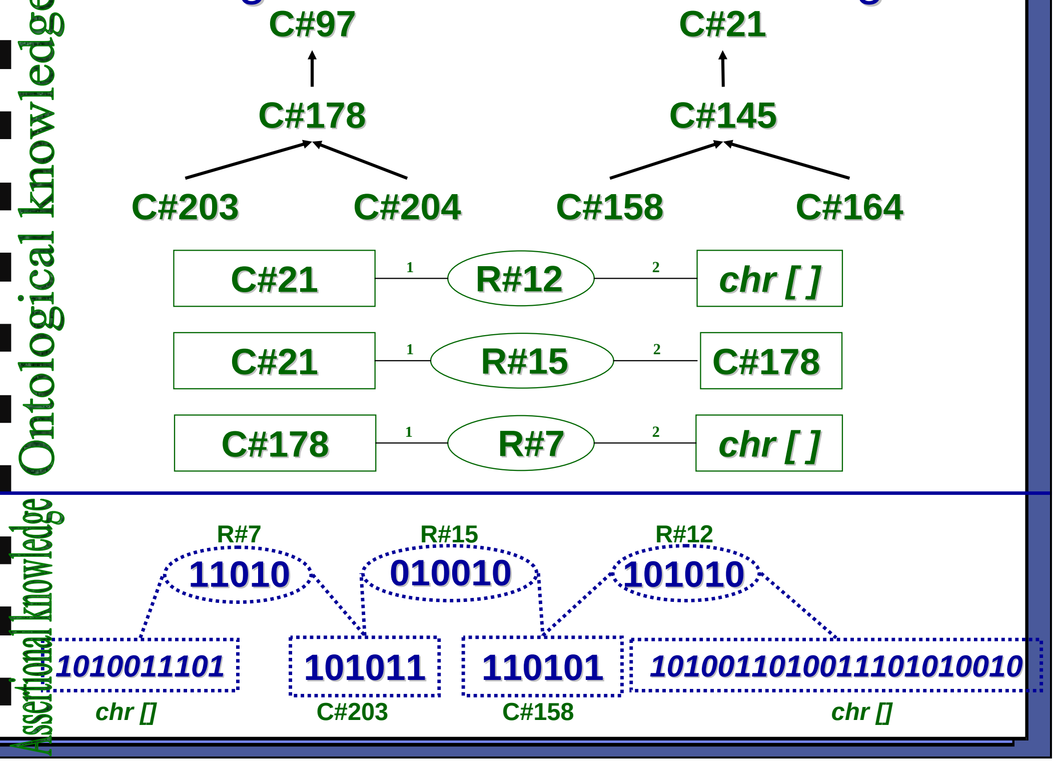
Inferential capabilities

- Search : Request
- Projection → Inference



Ontological *vs.* assertional knowledge

Ontological *vs.* assertional knowledge



Definitions

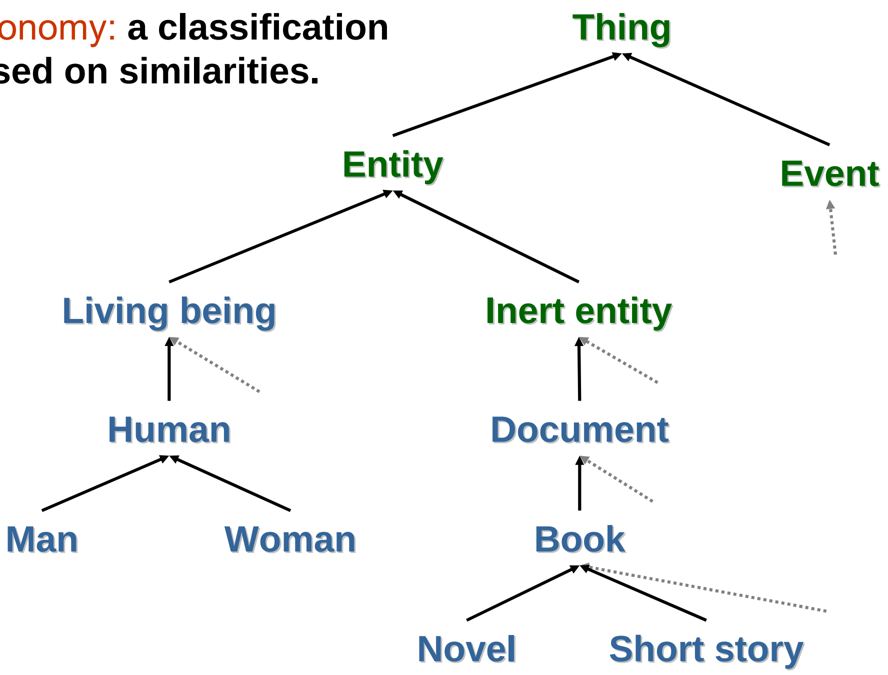
- ➢ conceptualisation: an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality [Guarino and Giaretta, 1995] || the action of building such a structure.
- ➢ Ontology: a branch of metaphysics which investigates the nature and essential properties and relations of all beings as such.
- ➢ ontology: a logical theory which gives an explicit, partial account of a conceptualisation [Guarino and Giaretta, 1995] [Gruber, 1993]; the aim of ontologies is to define which primitives, provided with their associated semantics, are necessary for knowledge representation in a given context. [Bachimont, 2000]
- ➢ formal ontology: the systematic, formal, axiomatic development of the logic of all forms and modes of being [Guarino and Giaretta, 1995].

Ontology in Computer Science

- An ontology is an engineering artifact:
  - It is constituted by a specific vocabulary used to describe a certain reality (domain), plus
  - a set of explicit assumptions regarding the intended meaning of the vocabulary.
- Thus, an ontology describes a formal specification of a certain domain:
  - Shared understanding of a domain of interest
  - Formal and machine manipulable model of a domain of interest (telecoms systems, gene structures, public services, ...)

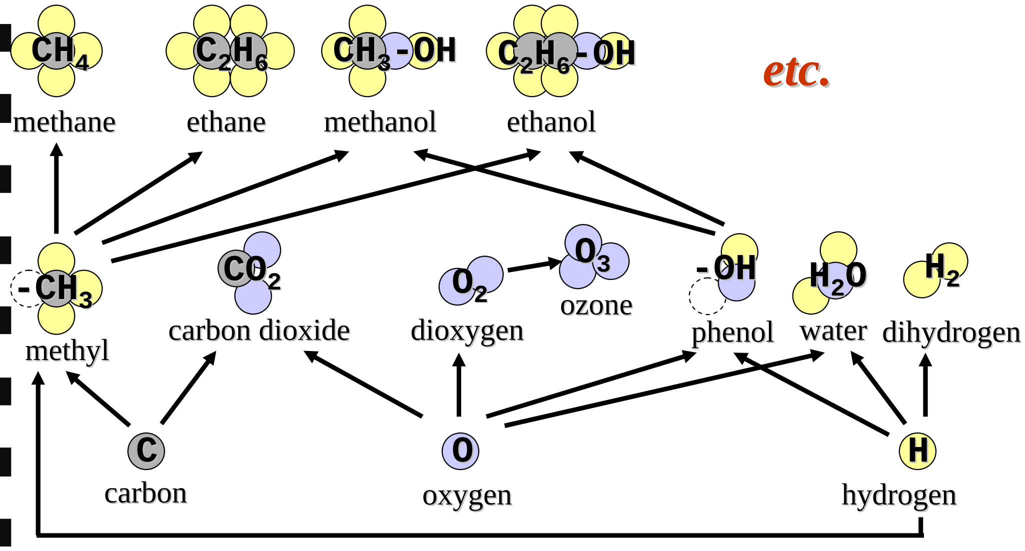
Ontology *vs.* taxonomy

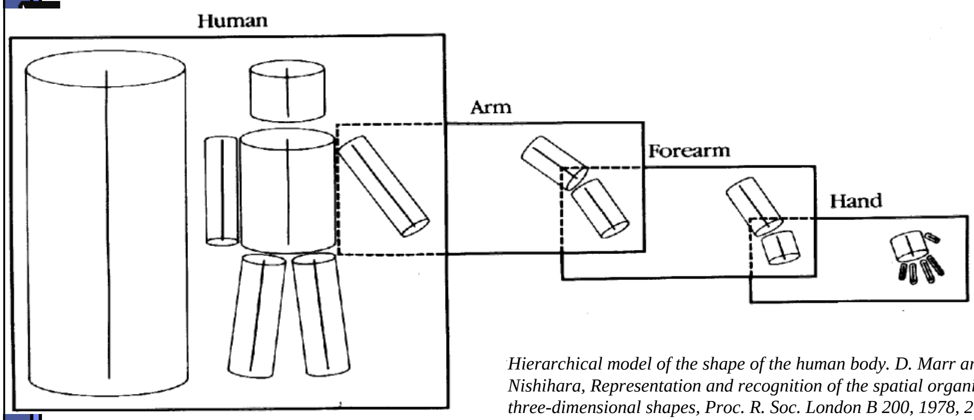
- taxonomy: **a classification based on similarities.**



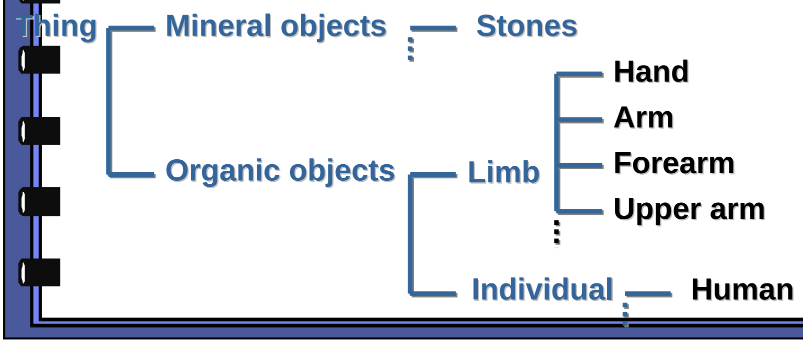
Partonomy example

- taxonomy: **a classification based on similarities.**
- partonomy: **a classification based on part-of relation.**



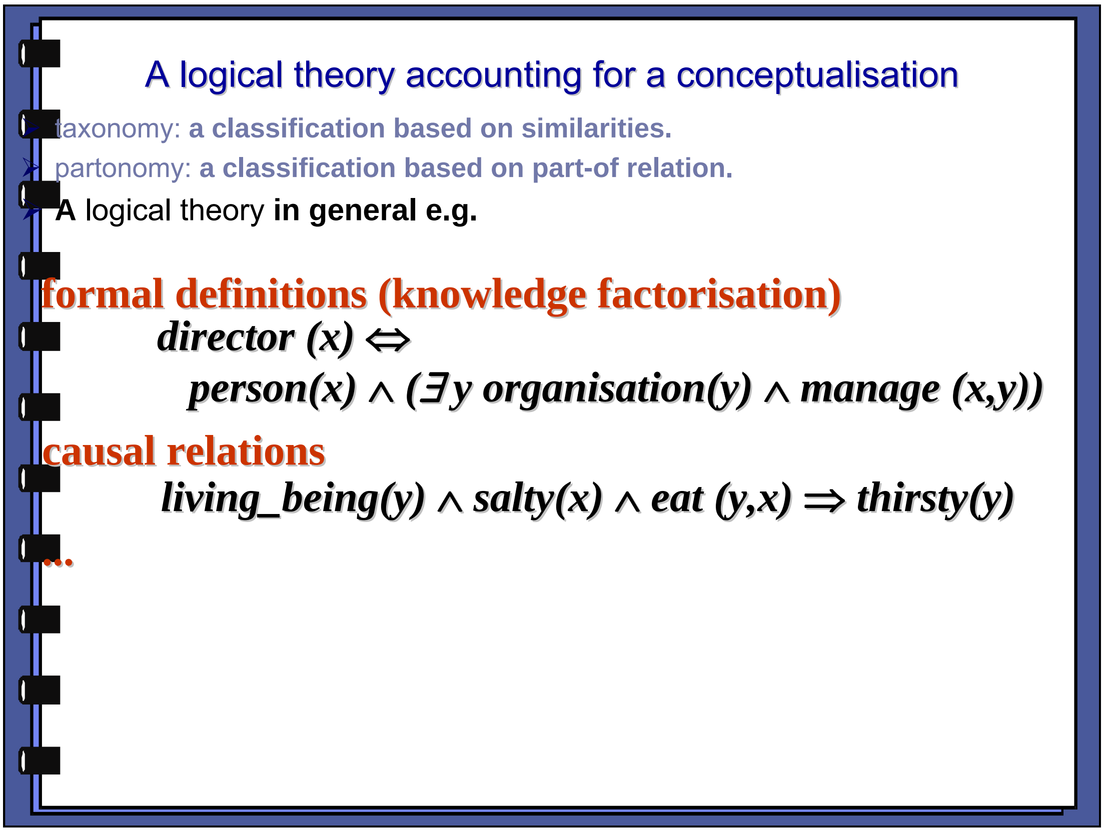


Hierarchical model of the shape of the human body. D. Marr and H.K. Nishihara, Representation and recognition of the spatial organization of three-dimensional shapes, Proc. R. Soc. London B 200, 1978, 269-294).



Thing
Mineral objects
Stones
Organic objects
Limb
Hand
Arm
Forearm
Upper arm
Individual
Human

Taxonomy Partonomy



A logical theory accounting for a conceptualisation

- taxonomy: **a classification based on similarities.**
- partonomy: **a classification based on part-of relation.**
- **A** logical theory **in general e.g.**

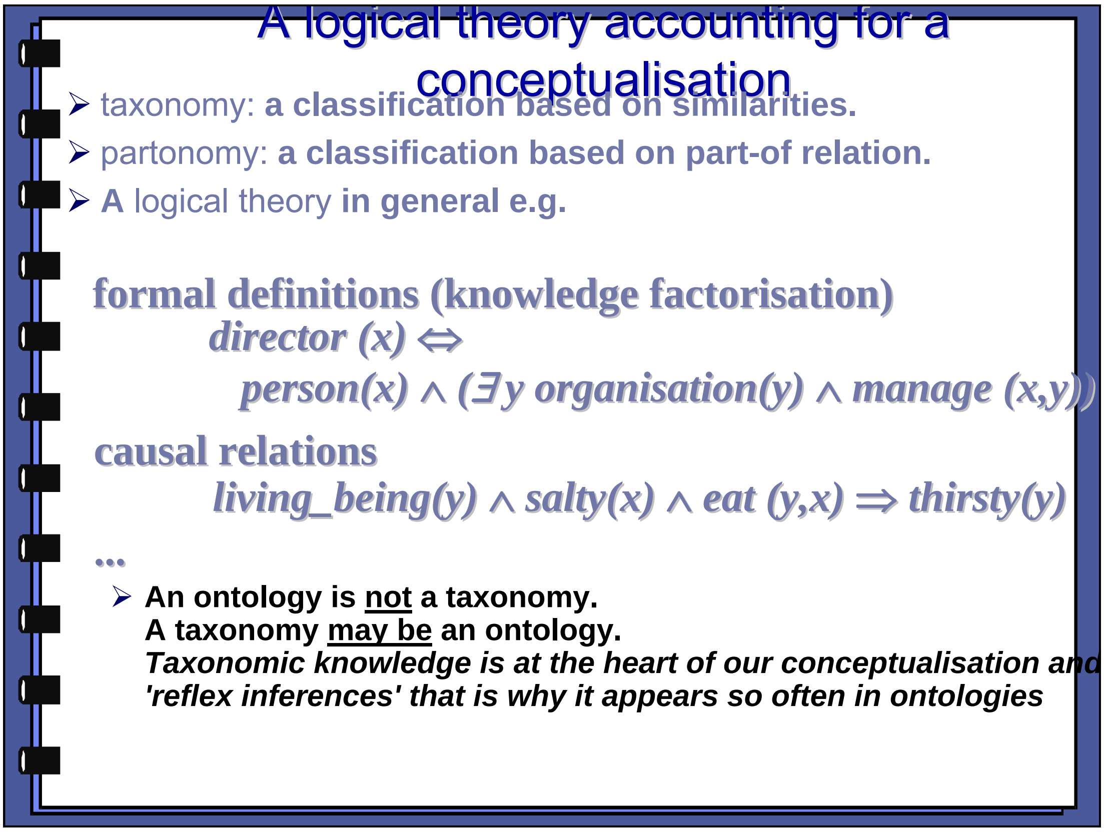
formal definitions (knowledge factorisation)

$$director(x) \Leftrightarrow$$
$$person(x) \wedge (\exists y\ organisation(y) \wedge manage(x,y))$$

causal relations

$$living\_being(y) \wedge salty(x) \wedge eat(y,x) \Rightarrow thirsty(y)$$

...



A logical theory accounting for a conceptualisation

- taxonomy: **a classification based on similarities.**
- partonomy: **a classification based on part-of relation.**
- **A** logical theory **in general e.g.**

formal definitions (knowledge factorisation)

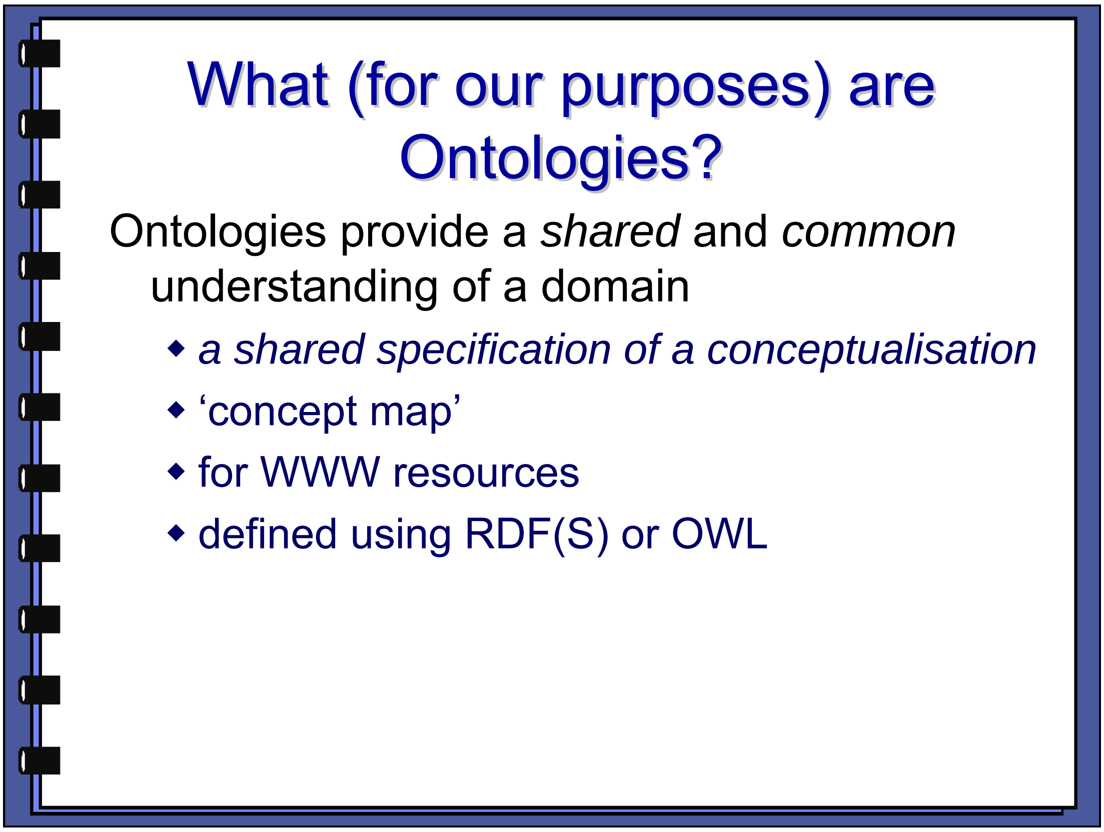
$$director(x) \Leftrightarrow$$
$$person(x) \wedge (\exists y\ organisation(y) \wedge manage(x,y))$$

causal relations

$$living\_being(y) \wedge salty(x) \wedge eat(y,x) \Rightarrow thirsty(y)$$

...

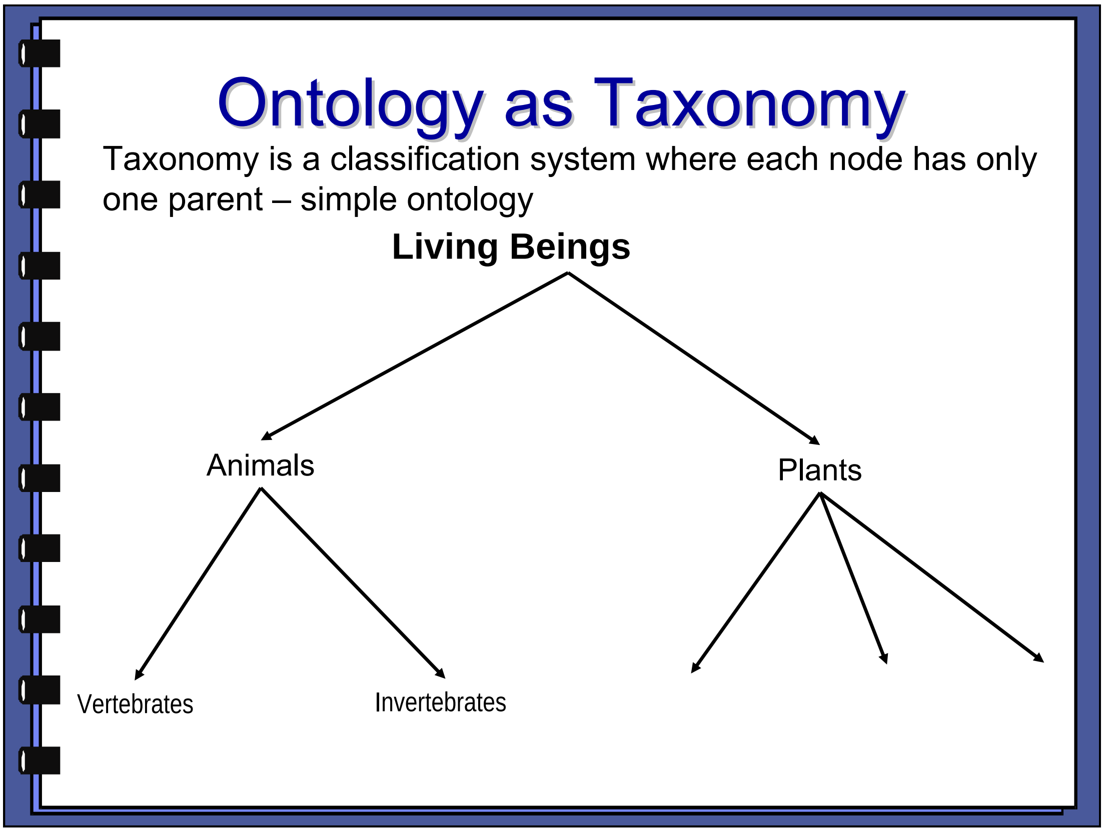
- **An ontology is <u>not</u> a taxonomy.**
  A taxonomy <u>may be</u> an ontology.
  Taxonomic knowledge is at the heart of our conceptualisation and 'reflex inferences' that is why it appears so often in ontologies



What (for our purposes) are Ontologies?

Ontologies provide a *shared* and *common* understanding of a domain

- *a shared specification of a conceptualisation*
- ‘concept map’
- for WWW resources
- defined using RDF(S) or OWL

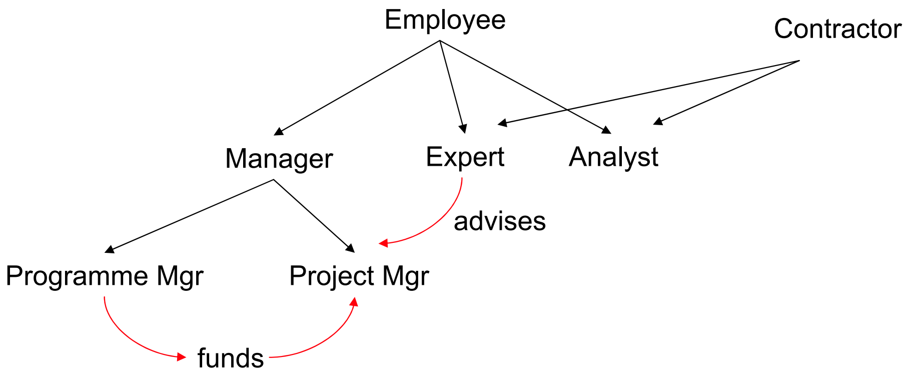


Ontology as Taxonomy

Taxonomy is a classification system where each node has only one parent – simple ontology

Ontology of People and their Roles

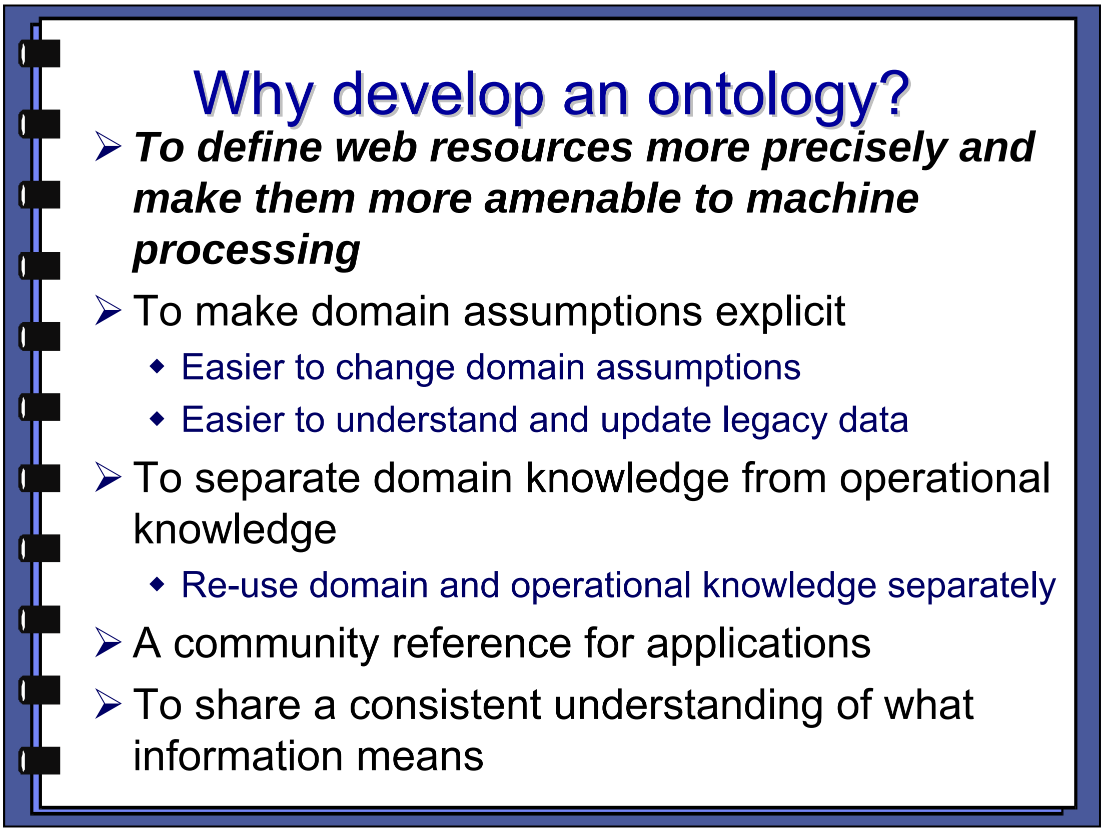
Typically, we want a richer ontology with more relationships between concepts:



Structure of an Ontology

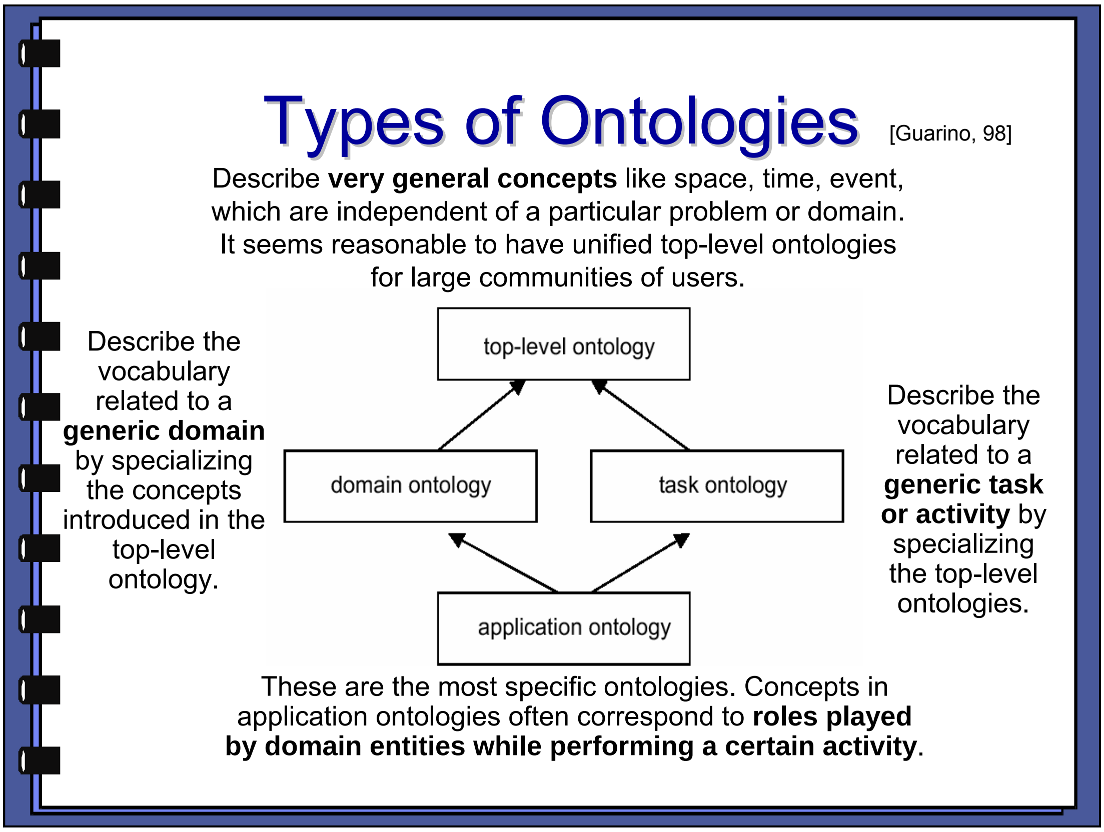
Ontologies typically have two distinct components:

- Names for important concepts and **relationships** 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
- Background knowledge/constraints on the domain
  - Adult_Elephants **weigh** at least 2,000 kg
  - No individual can be both a Herbivore and a Carnivore



Why develop an ontology?

- ***To define web resources more precisely and make them more amenable to machine processing***
- To make domain assumptions explicit
  - Easier to change domain assumptions
  - Easier to understand and update legacy data
- To separate domain knowledge from operational knowledge
  - Re-use domain and operational knowledge separately
- A community reference for applications
- To share a consistent understanding of what information means



Types of Ontologies

[Guarino, 98]

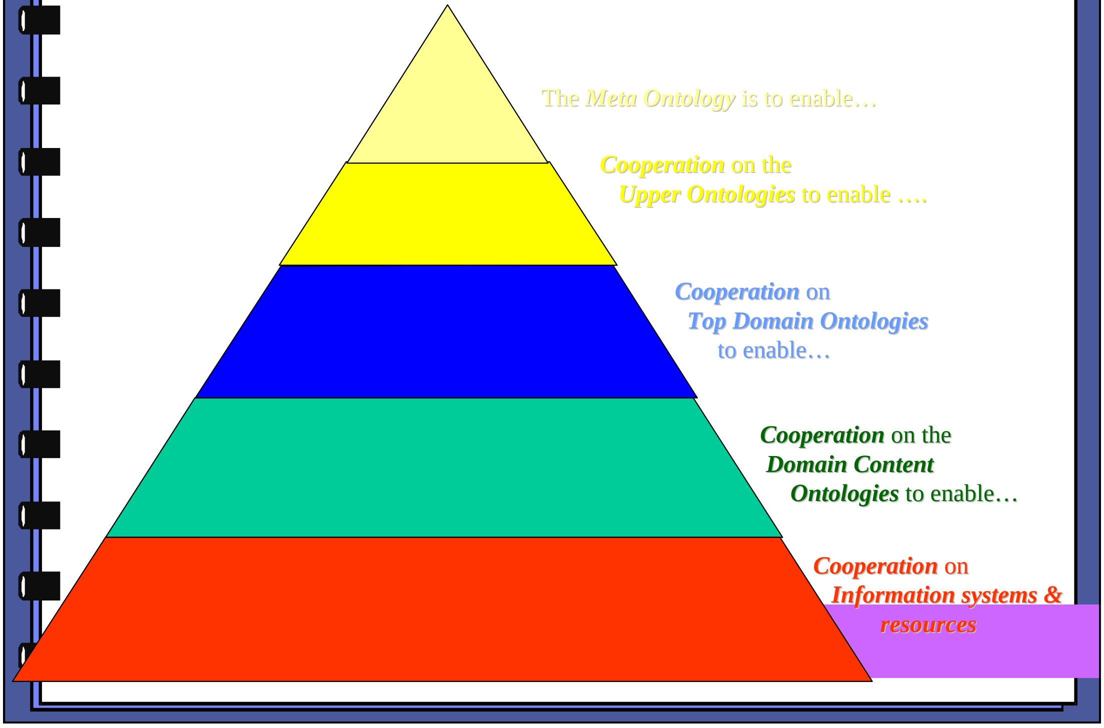
Describe **very general concepts** like space, time, event, which are independent of a particular problem or domain. It seems reasonable to have unified top-level ontologies for large communities of users.

Describe the vocabulary related to a **generic domain** by specializing the concepts introduced in the top-level ontology.

Describe the vocabulary related to a **generic task or activity** by specializing the top-level ontologies.

These are the most specific ontologies. Concepts in application ontologies often correspond to **roles played by domain entities while performing a certain activity**.

Ontology Layers: What's it for?



The *Meta Ontology* is to enable…

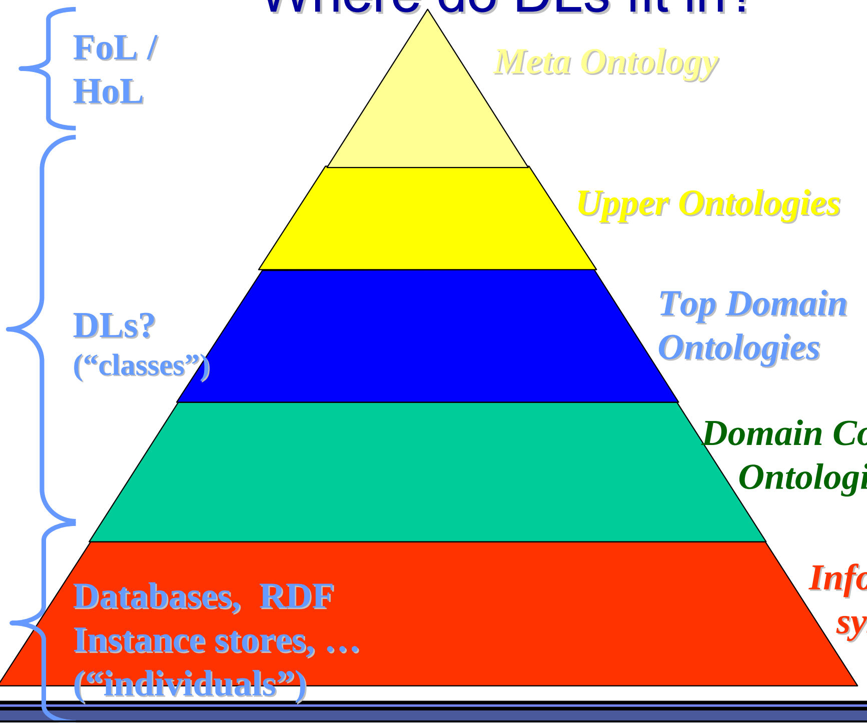
Cooperation on the ***Upper Ontologies*** to enable ….

Cooperation on ***Top Domain Ontologies*** to enable…

Cooperation on the ***Domain Content Ontologies*** to enable…

Cooperation on ***Information systems & resources***

Where do DLs fit in?



FoL / HoL

DLs? ("classes")

Databases, RDF Instance stores, … ("individuals")

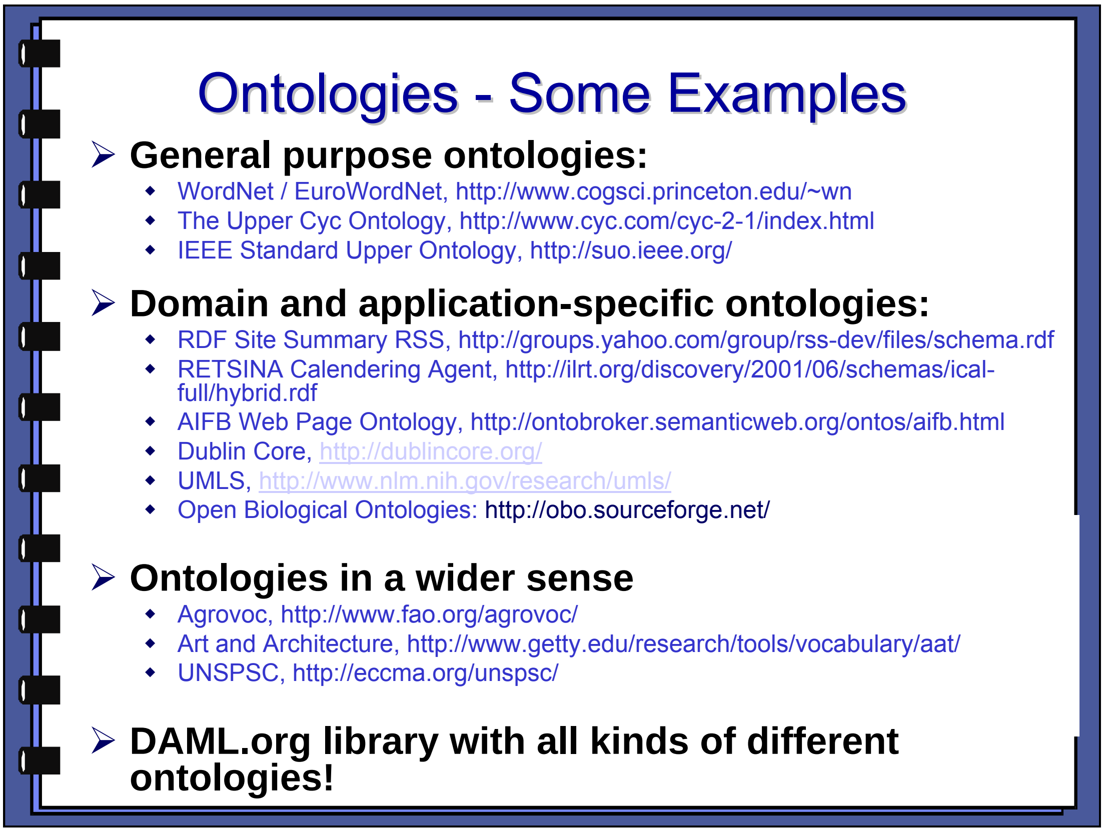
Meta Ontology

Upper Ontologies

Top Domain Ontologies

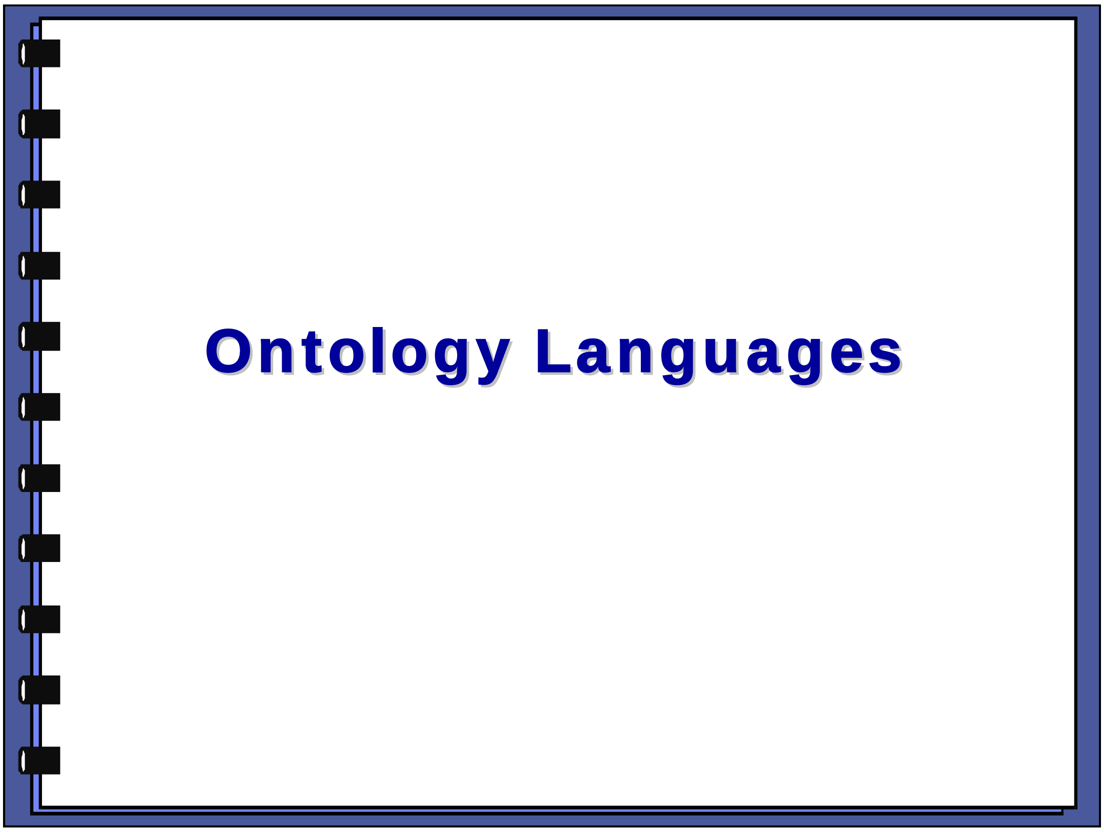
Domain Content Ontologies

Information systems & resources



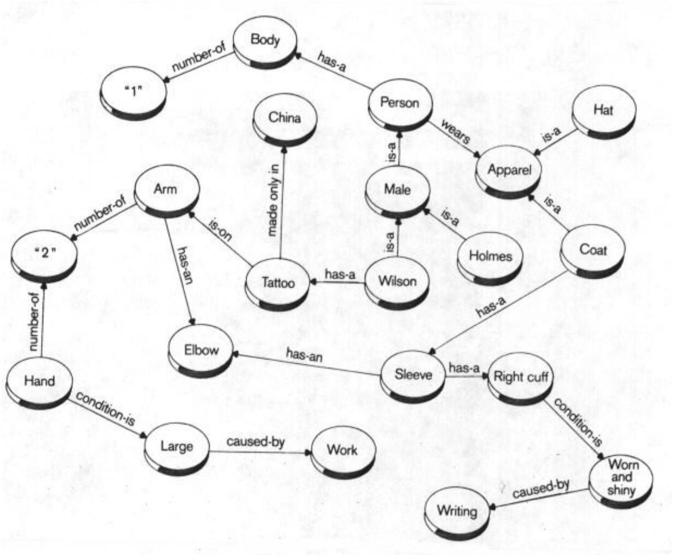
Ontologies - Some Examples

- **General purpose ontologies:**
  - WordNet / EuroWordNet, http://www.cogsci.princeton.edu/~wn
  - The Upper Cyc Ontology, http://www.cyc.com/cyc-2-1/index.html
  - IEEE Standard Upper Ontology, http://suo.ieee.org/
- **Domain and application-specific ontologies:**
  - RDF Site Summary RSS, http://groups.yahoo.com/group/rss-dev/files/schema.rdf
  - RETSINA Calendering Agent, http://ilrt.org/discovery/2001/06/schemas/ical-full/hybrid.rdf
  - AIFB Web Page Ontology, http://ontobroker.semanticweb.org/ontos/aifb.html
  - Dublin Core, http://dublincore.org/
  - UMLS, http://www.nlm.nih.gov/research/umls/
  - Open Biological Ontologies: http://obo.sourceforge.net/
- **Ontologies in a wider sense**
  - Agrovoc, http://www.fao.org/agrovoc/
  - Art and Architecture, http://www.getty.edu/research/tools/vocabulary/aat/
  - UNSPSC, http://eccma.org/unspsc/
- **DAML.org library with all kinds of different ontologies!**

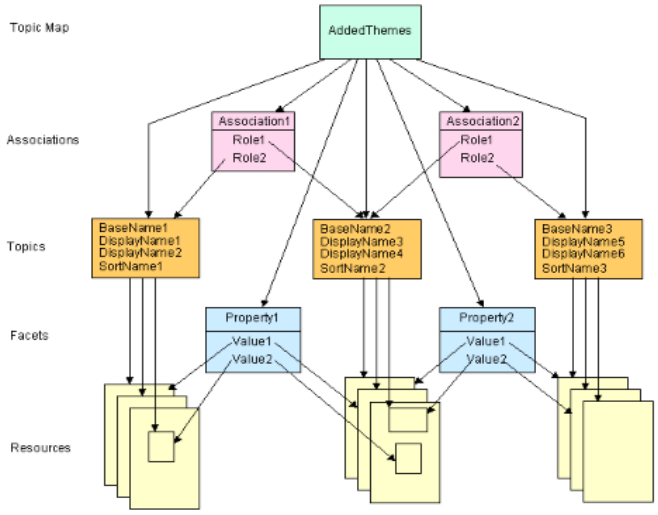


Ontology Languages

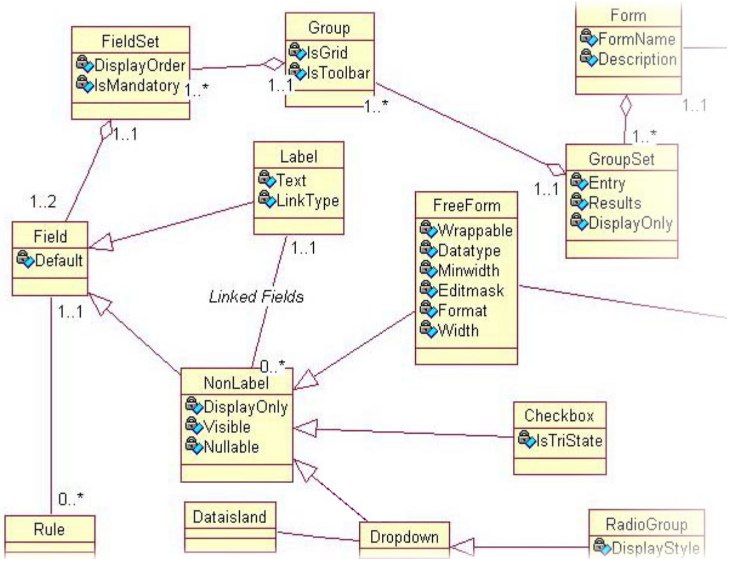
- Wide variety of languages for “Explicit Specification”
  - Graphical notations
    - Semantic networks



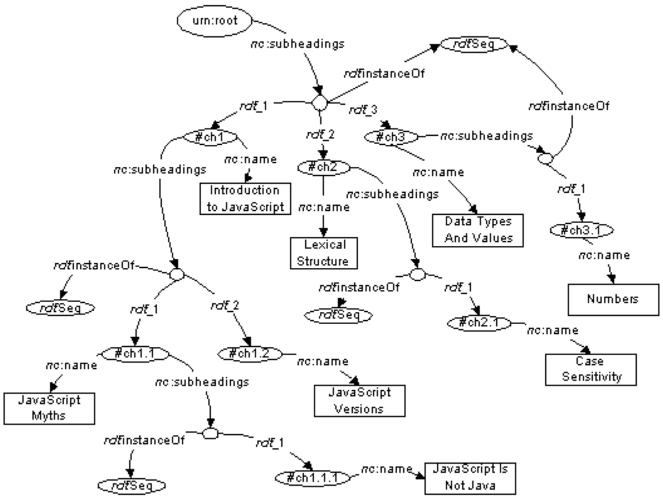
- Wide variety of languages for “Explicit Specification”
  - Graphical notations
    - Topic Maps



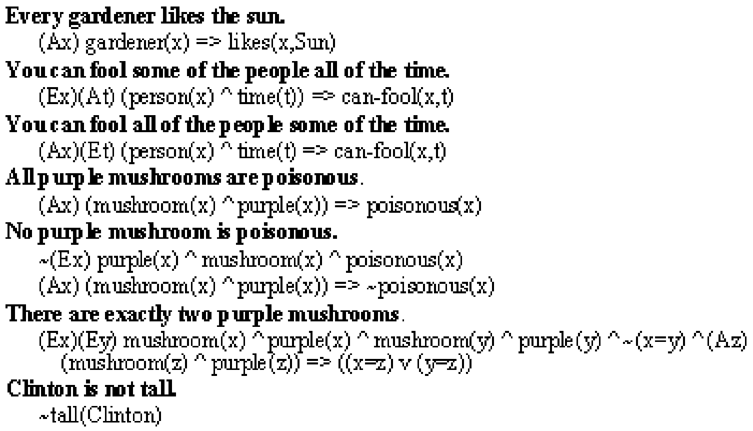
- Wide variety of languages for “Explicit Specification”
  - Graphical notations
    - UML



- Wide variety of languages for “Explicit Specification”
  - Graphical notations
    - RDF

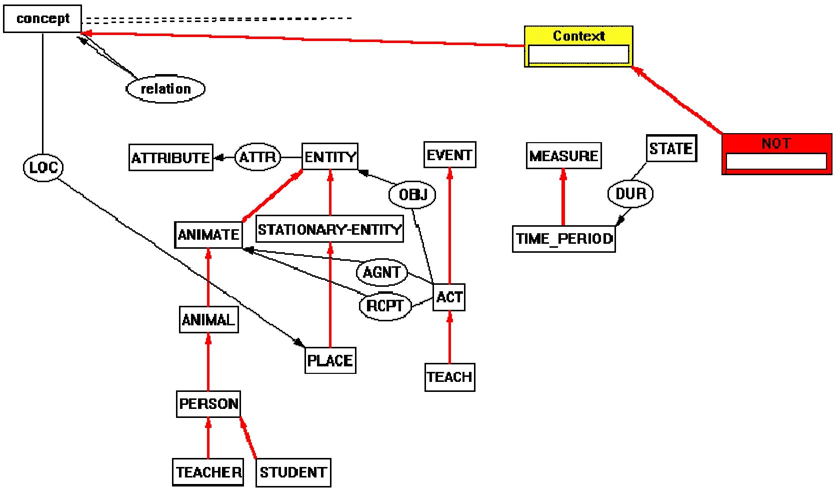


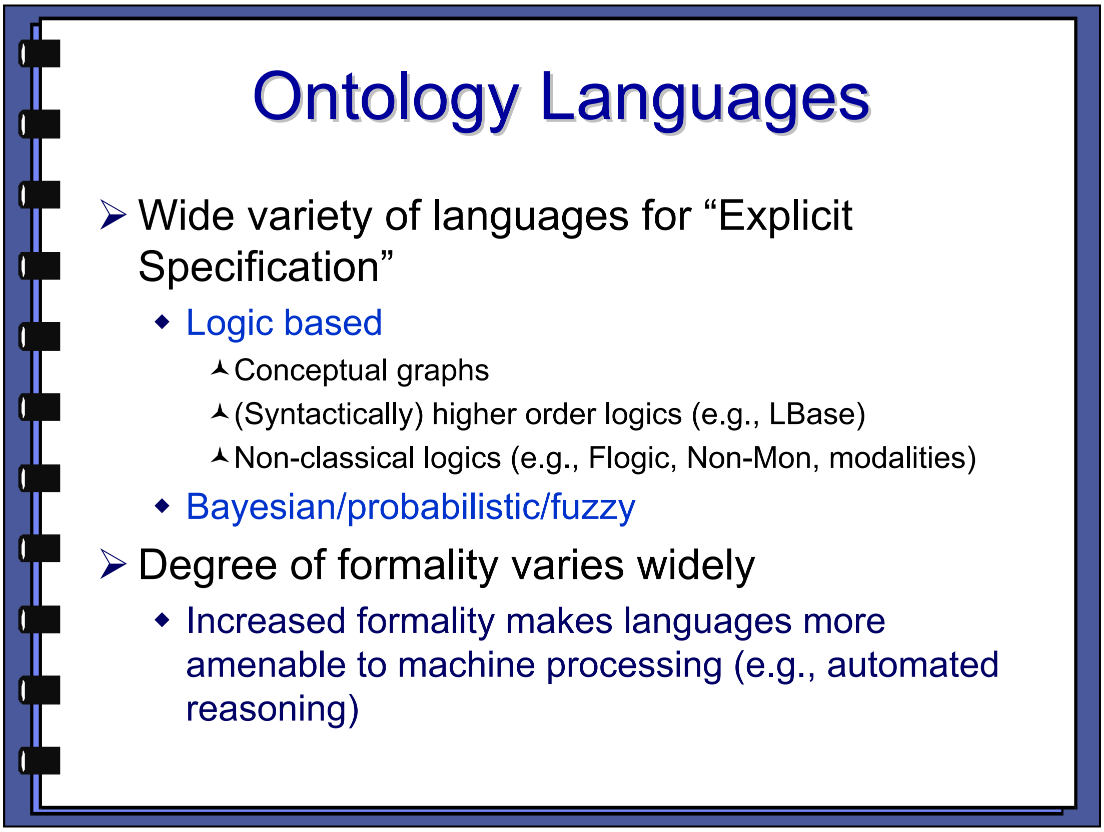
- Wide variety of languages for “Explicit Specification”
  - Logic based
    - Description Logics (e.g., OIL, DAML+OIL, OWL)
    - Rules (e.g., RuleML, LP/Prolog)
    - First Order Logic (e.g., KIF)



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)

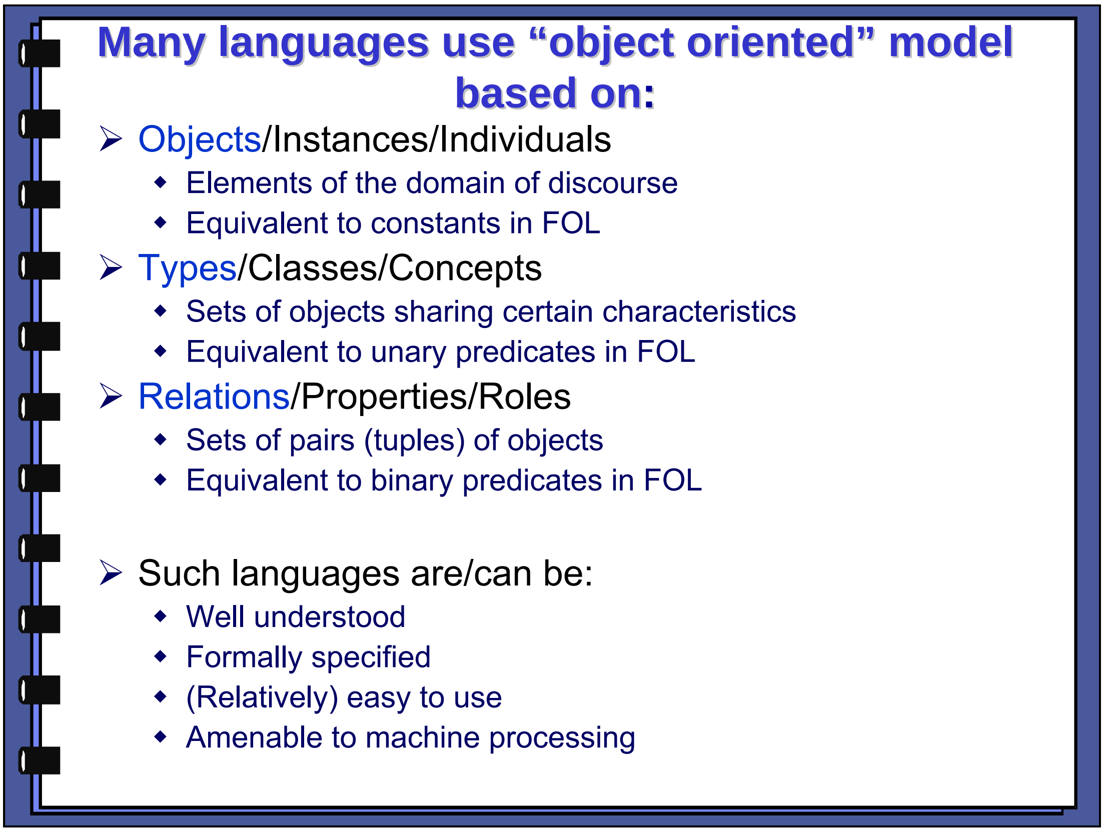
- Wide variety of languages for “Explicit Specification”
  - Logic based
    - Conceptual graphs





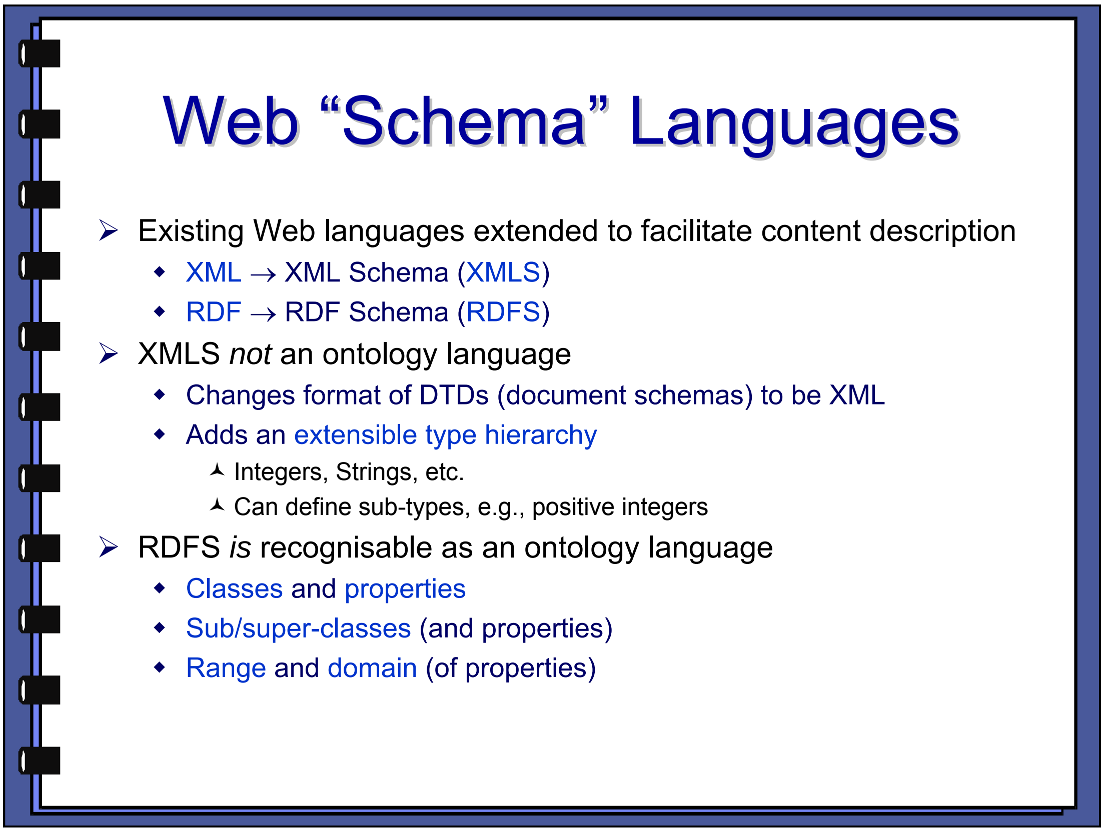
Ontology Languages

- Wide variety of languages for “Explicit Specification”
  - Logic based
    - Conceptual graphs
    - (Syntactically) higher order logics (e.g., LBase)
    - Non-classical logics (e.g., Flogic, Non-Mon, modalities)
  - Bayesian/probabilistic/fuzzy
- Degree of formality varies widely
  - Increased formality makes languages more amenable to machine processing (e.g., automated reasoning)



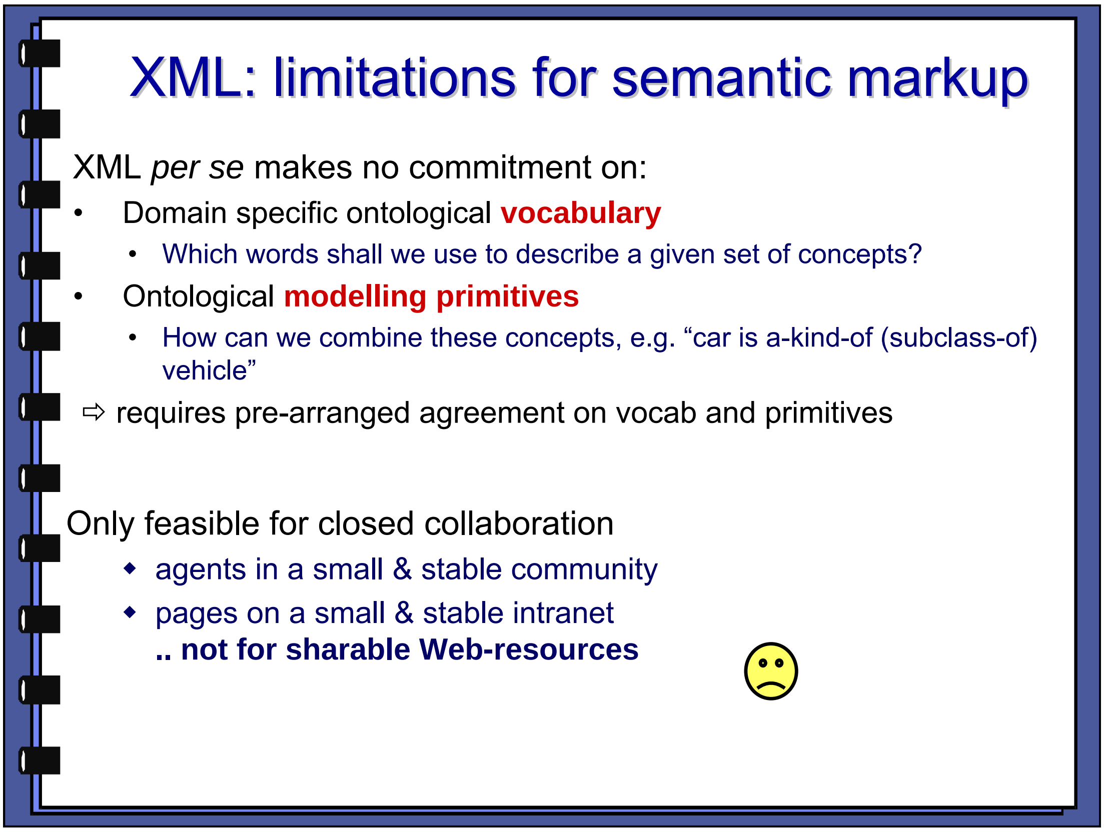
Many languages use “object oriented” model based on:

- Objects/Instances/Individuals
  - Elements of the domain of discourse
  - Equivalent to constants in FOL
- Types/Classes/Concepts
  - Sets of objects sharing certain characteristics
  - Equivalent to unary predicates in FOL
- Relations/Properties/Roles
  - Sets of pairs (tuples) of objects
  - Equivalent to binary predicates in FOL
- Such languages are/can be:
  - Well understood
  - Formally specified
  - (Relatively) easy to use
  - Amenable to machine processing



Web “Schema” Languages

- Existing Web languages extended to facilitate content description
  - XML → XML Schema (XMLS)
  - RDF → RDF Schema (RDFS)
- XMLS *not* an ontology language
  - Changes format of DTDs (document schemas) to be XML
  - Adds an extensible type hierarchy
    - Integers, Strings, etc.
    - Can define sub-types, e.g., positive integers
- RDFS *is* recognisable as an ontology language
  - Classes and properties
  - Sub/super-classes (and properties)
  - Range and domain (of properties)



XML: limitations for semantic markup

XML *per se* makes no commitment on:

- Domain specific ontological **vocabulary**
  - Which words shall we use to describe a given set of concepts?
- Ontological **modelling primitives**
  - How can we combine these concepts, e.g. “car is a-kind-of (subclass-of) vehicle”

⇨ requires pre-arranged agreement on vocab and primitives

Only feasible for closed collaboration

- agents in a small & stable community
- pages on a small & stable intranet

.. not for sharable Web-resources ☹

XML is a first step

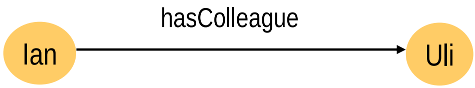
- Semantic markup
  - HTML ⇨ layout
  - XML ⇨ meaning
- Metadata
  - within documents, not across documents
  - *prescriptive*, not *descriptive*
  - No commitment on vocabulary and modelling primitives
- RDF is the next step

RDF and RDFS

- RDF stands for Resource Description Framework
- Standard of W3C
  - It is a W3C candidate recommendation (http://www.w3.org/RDF)
- RDF is graphical formalism (+ XML syntax + semantics)
  - for representing metadata
  - for describing the semantics of information in a machine- accessible way
- RDFS extends RDF with “schema vocabulary”, e.g.:
  - Class, Property
  - type, subClassOf, subPropertyOf
  - range, domain

The RDF Data Model

- Statements are <subject, predicate, object> triples:



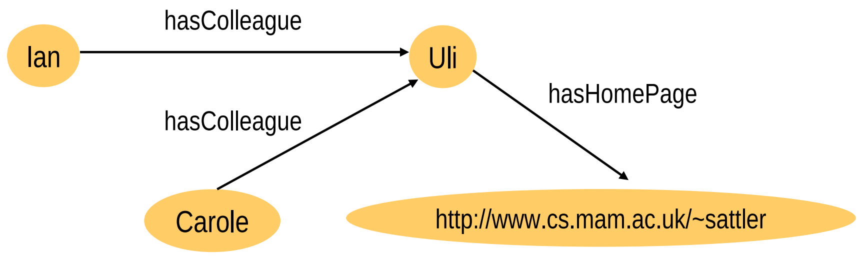
- **Can be represented using XML serialisation, e.g.:**
  `<Ian,hasColleague,Uli>`
- **Statements describe properties of resources**
- **A resource is a URI representing a (class of) object(s):**
  - **a document, a picture, a paragraph on the Web;**
  - **http://www.cs.man.ac.uk/index.html**
  - **a book in the library, a real person (?)**
  - **isbn://5031-4444-3333**
  - **…**
- **Properties themselves are also resources (URIs)**

URIs

- URI = Uniform Resource Identifier
- "The generic set of all names/addresses that are short strings that refer to resources“
- URIs may or may not be dereferencable
  - URLs (Uniform Resource Locators) are a particular type of URI, used for resources that can be accessed on the WWW (e.g., web pages)
- In RDF, URIs typically look like “normal” URLs, often with fragment identifiers to point at specific parts of a document:
  - http://www.somedomain.com/some/path/to/file#fragmentID

Linking Statements

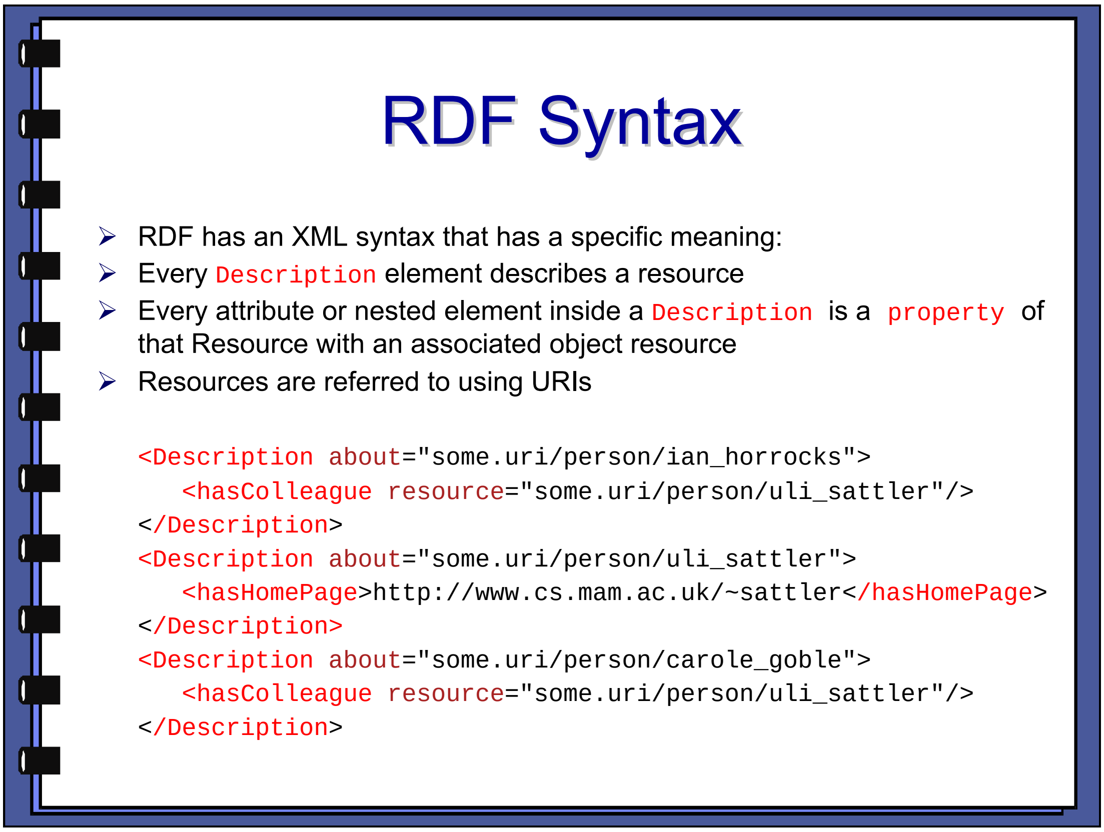
- The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph



- Note that the object of a triple can also be a “literal” (a string)

Problems with RDFS

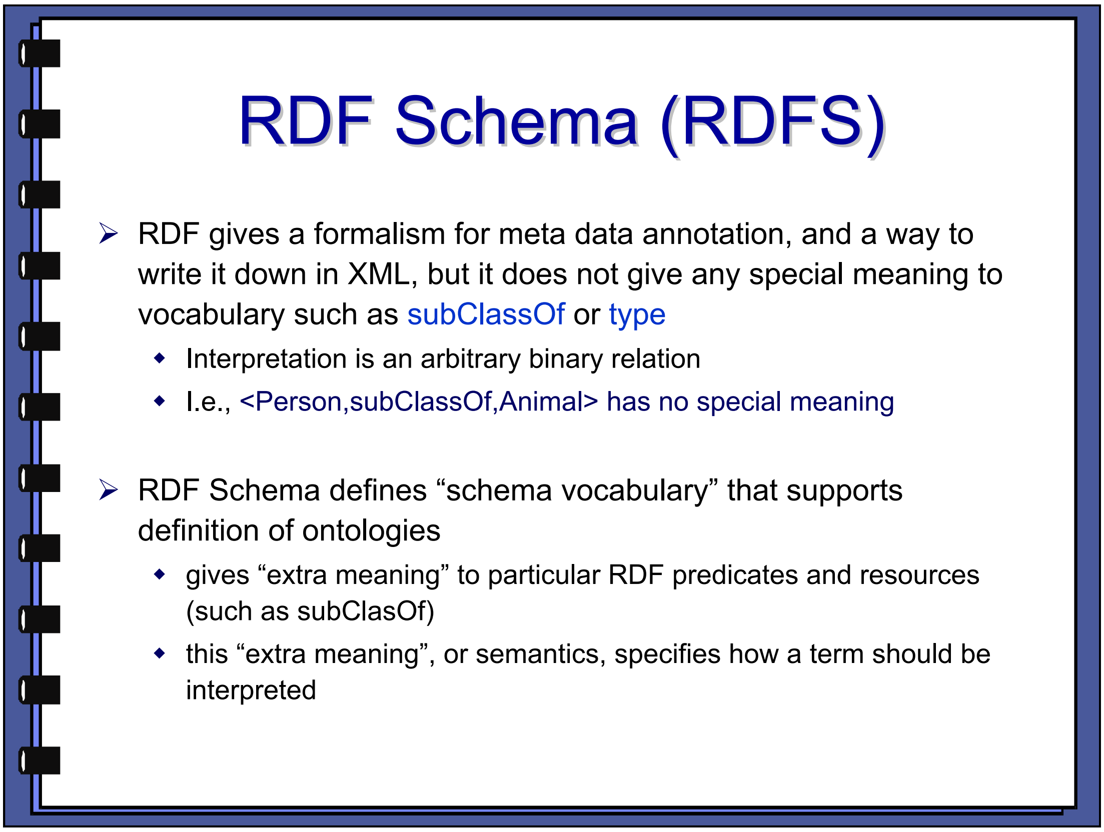
- RDFS too weak to describe resources in sufficient detail
  - No localised range and domain constraints
    - Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can’t say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No transitive, inverse or symmetrical properties
    - Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
  - …
- Difficult to provide reasoning support
  - No “native” reasoners for non-standard semantics
  - May be possible to reason via FO axiomatisation



RDF Syntax

- RDF has an XML syntax that has a specific meaning:
- Every Description element describes a resource
- Every attribute or nested element inside a Description is a property of that Resource with an associated object resource
- Resources are referred to using URIs

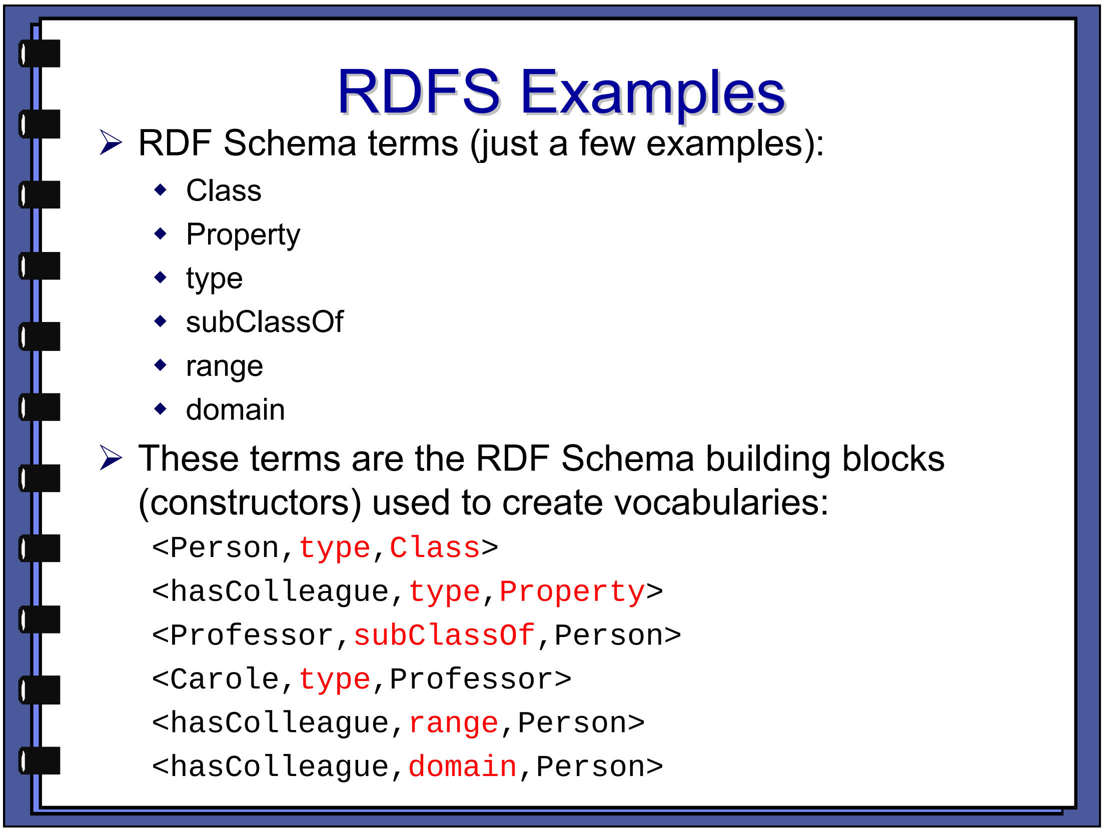
```
<Description about="some.uri/person/ian_horrocks">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
<Description about="some.uri/person/uli_sattler">
   <hasHomePage>http://www.cs.mam.ac.uk/~sattler</hasHomePage>
</Description>
<Description about="some.uri/person/carole_goble">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
```



RDF Schema (RDFS)

- RDF gives a formalism for meta data annotation, and a way to write it down in XML, but it does not give any special meaning to vocabulary such as subClassOf or type
  - Interpretation is an arbitrary binary relation
  - I.e., <Person,subClassOf,Animal> has no special meaning

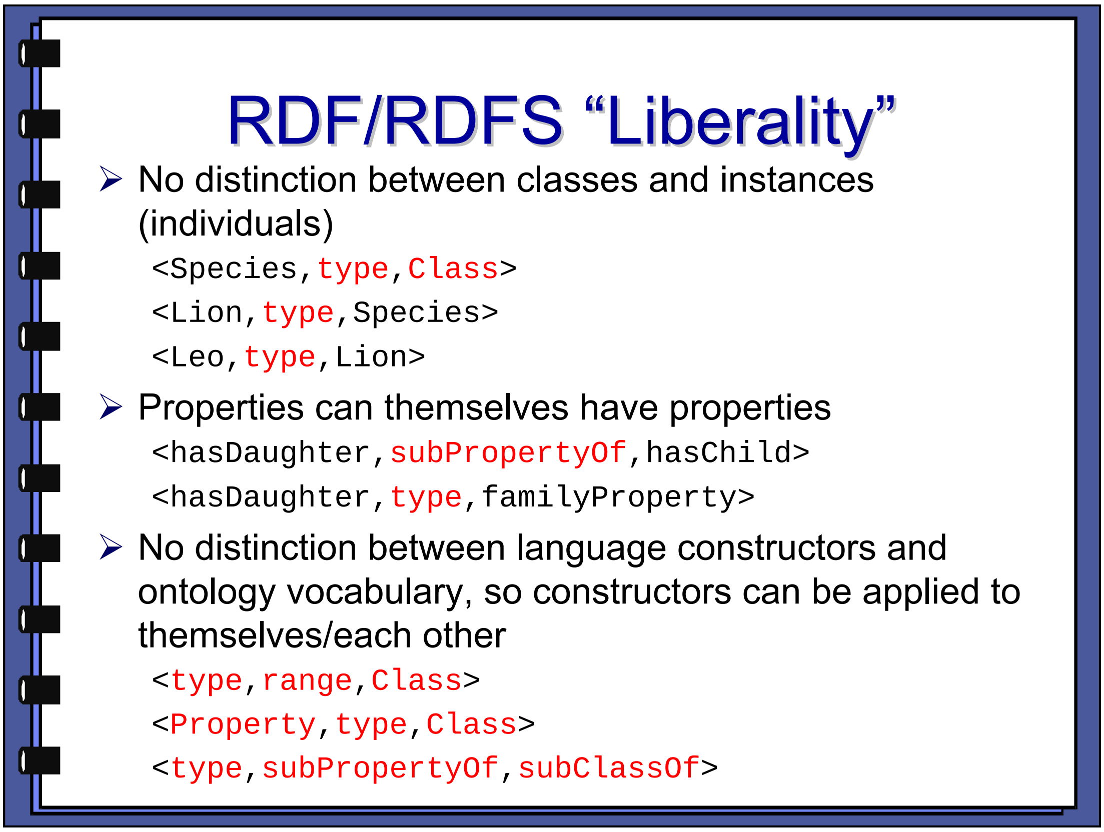
- RDF Schema defines “schema vocabulary” that supports definition of ontologies
  - gives “extra meaning” to particular RDF predicates and resources (such as subClasOf)
  - this “extra meaning”, or semantics, specifies how a term should be interpreted



RDFS Examples

- RDF Schema terms (just a few examples):
  - Class
  - Property
  - type
  - subClassOf
  - range
  - domain
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

```
<Person,type,Class>
<hasColleague,type,Property>
<Professor,subClassOf,Person>
<Carole,type,Professor>
<hasColleague,range,Person>
<hasColleague,domain,Person>
```



RDF/RDFS “Liberality”

- No distinction between classes and instances (individuals)

```
<Species,type,Class>
<Lion,type,Species>
<Leo,type,Lion>
```

- Properties can themselves have properties

```
<hasDaughter,subPropertyOf,hasChild>
<hasDaughter,type,familyProperty>
```

- No distinction between language constructors and ontology vocabulary, so constructors can be applied to themselves/each other

```
<type,range,Class>
<Property,type,Class>
<type,subPropertyOf,subClassOf>
```

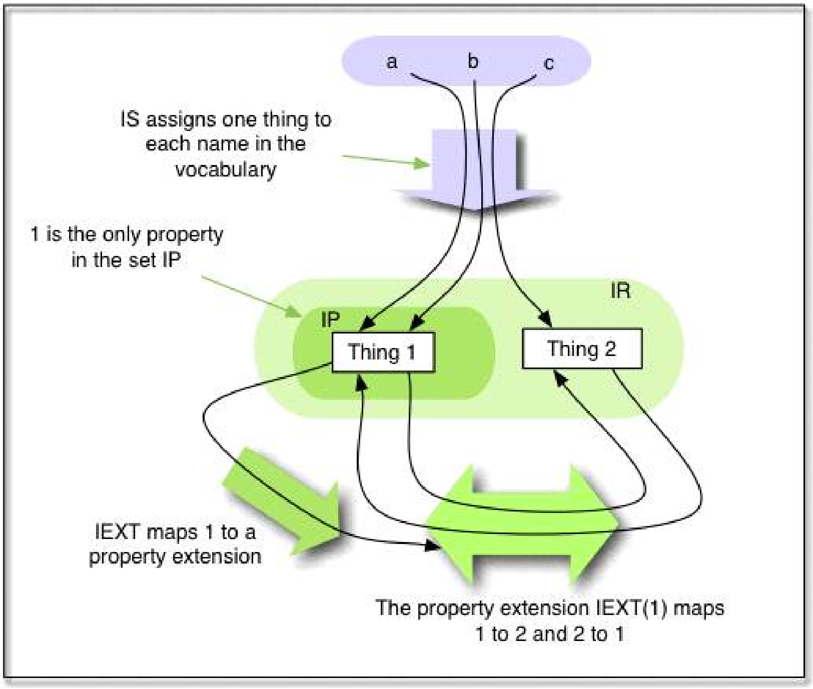
RDF/RDFS Semantics

- RDF has “Non-standard” semantics in order to deal with this
- Semantics given by RDF Model Theory (MT)

RDF Semantics

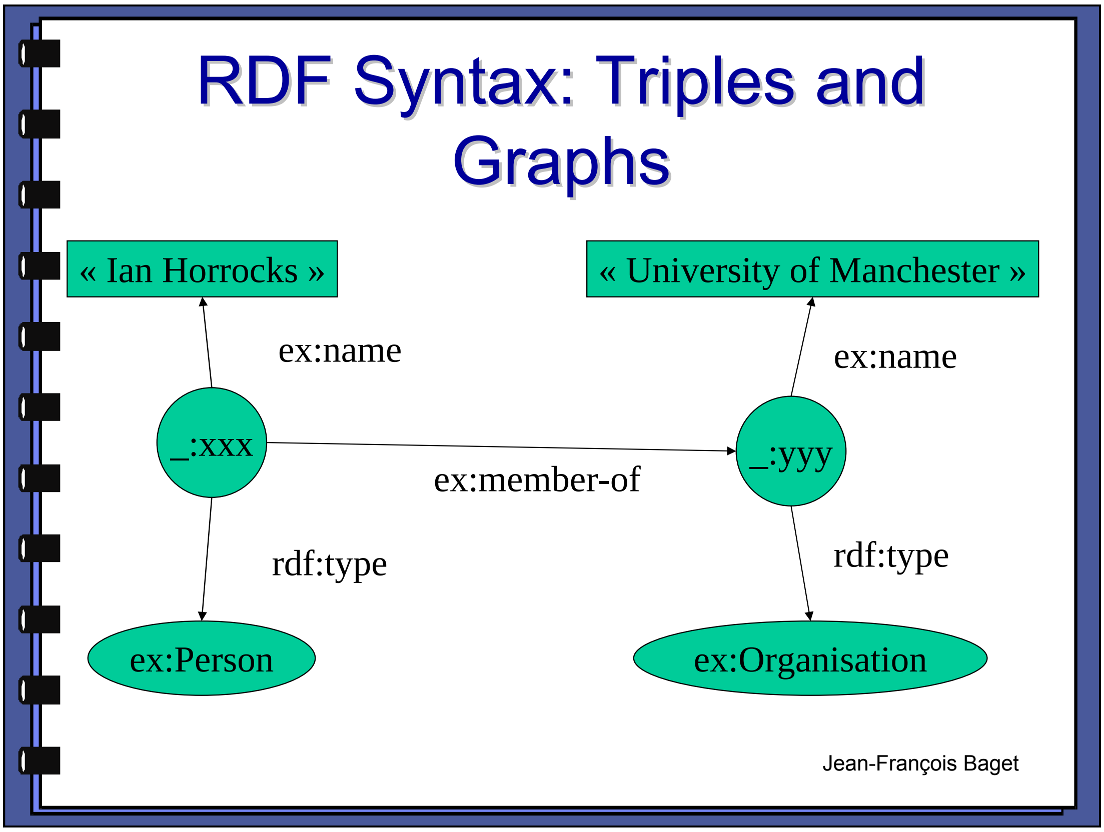
- In RDF MT, an interpretation **I** of a vocabulary V consists of:
  - IR, a non-empty set of resources (corresponds to ∆)
  - IS, a mapping from V into IR (corresponds to ¢^{I})
  - IP, a distinguished subset of IR (the properties)
    - A vocabulary element v 2 V is a property iff IS(v) 2 IP
  - IEXT, a mapping from IP into the powerset of IR£IR
    - I.e., property elements mapped to subsets of IR£IR
  - IL, a mapping from typed literals into IR

Example RDF Simple


a
b
c
IS assigns one thing to each name in the vocabulary
1 is the only property in the set IP
IR
IP
Thing 1
Thing 2
IEXT maps 1 to a property extension
The property extension IEXT(1) maps 1 to 2 and 2 to 1

Resource Description Framework (RDF)

- A standard of W3C
- Relationships *between* documents
- Consisting of triples or sentences:
  - <subject, property, verb>
  - <Tolkien, wrote, The Lord of the Rings>
- RDFS extends RDF with standard “ontology vocabulary”:
  - Class, Property
  - Type, subClassOf
  - domain, range



RDF Syntax: Triples and Graphs

« Ian Horrocks »

« University of Manchester »

ex:name

ex:name

_:xxx

ex:member-of

_:yyy

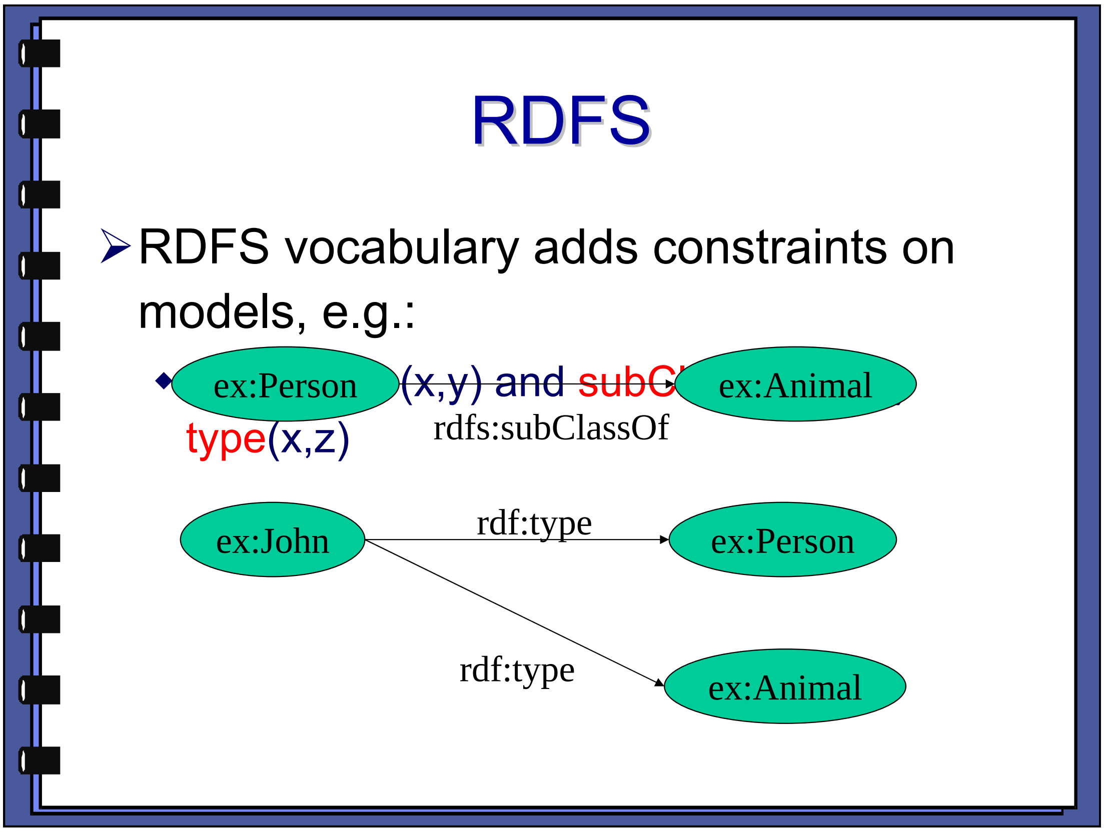
rdf:type

rdf:type

ex:Person

ex:Organisation

Jean-François Baget



RDFS

- RDFS vocabulary adds constraints on models, e.g.:
  - ex:Person ~~(x,y) and subC~~ ex:Animal type(x,z)

rdfs:subClassOf

ex:John

rdf:type

ex:Person

rdf:type

ex:Animal

RDFS Semantics

- RDFS simply adds semantic conditions and axiomatic triples that give meaning to schema vocabulary
- Class interpretation ICEXT simply induced by rdf:type, i.e.:
  - x is in ICEXT(y) if and only if <x,y> is in IEXT(IS(rdf:type))
- Other semantic conditions include:
  - If <x,y> is in IEXT(IS(rdfs:domain)) and <u,v> is in IEXT(x) then u is in ICEXT(y)
  - If <x,y> is in IEXT(IS(rdfs:subClassOf)) then x and y are in IC and ICEXT(x) is a subset of ICEXT(y)
  - IEXT(IS(rdfs:subClassOf)) is transitive and reflexive on IC
- Axiomatic triples include:
  - rdf:type rdfs:domain rdfs:Resource
  - rdfs:domain rdfs:domain rdf:Property

RDFS Interpretation Example

- If RDFS graph includes triples

```
<Species,type,Class>
<Lion,type,Species>
<Leo,type,Lion>
<Lion,subClassOf,Mamal>
<Mamal,subClassOf,Animal>
```

- Interpretation conditions imply existence of triples

```
<Lion,subClassOf,Animal>
<Leo,type,Mamal>
<Leo,type,Animal>
…
```

Problems with RDFS

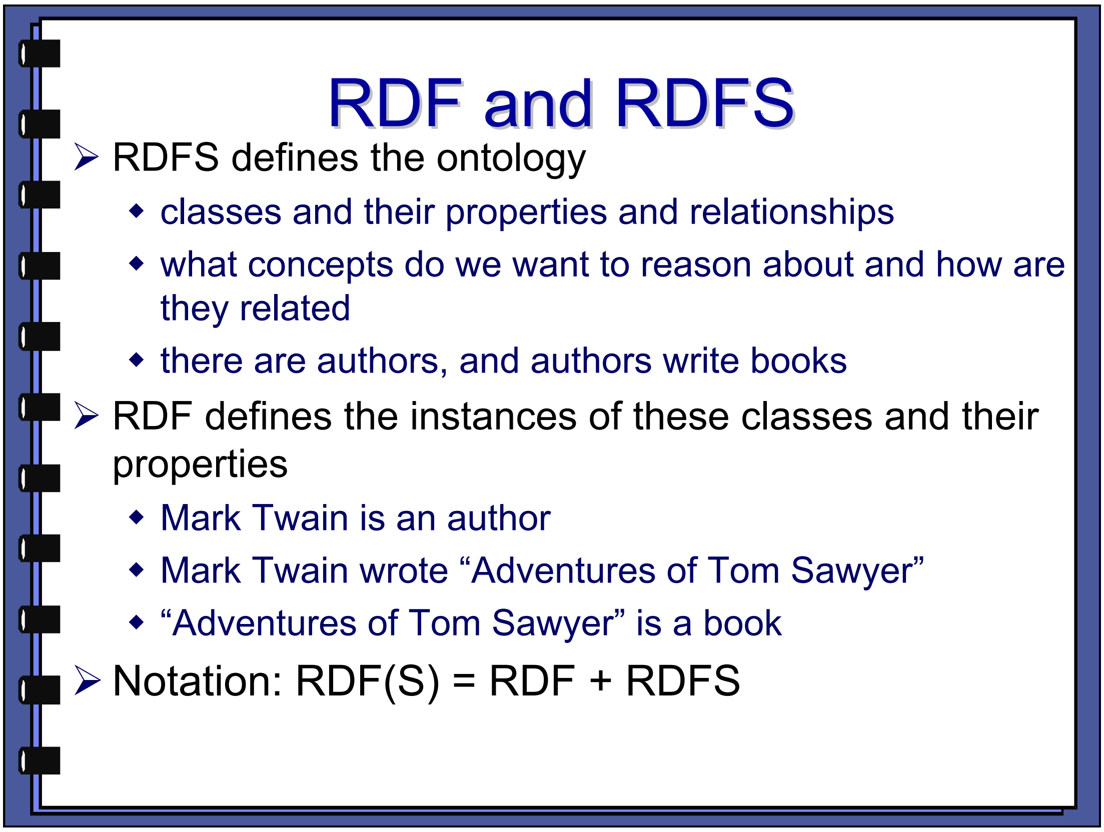
- RDFS too weak to describe resources in sufficient detail
  - No localised range and domain constraints
    - Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can’t say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No transitive, inverse or symmetrical properties
    - Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
  - …
- Difficult to provide reasoning support
  - No “native” reasoners for non-standard semantics
  - May be possible to reason via FO axiomatisation

An example

“Tolkein wrote ISBN00001047582”

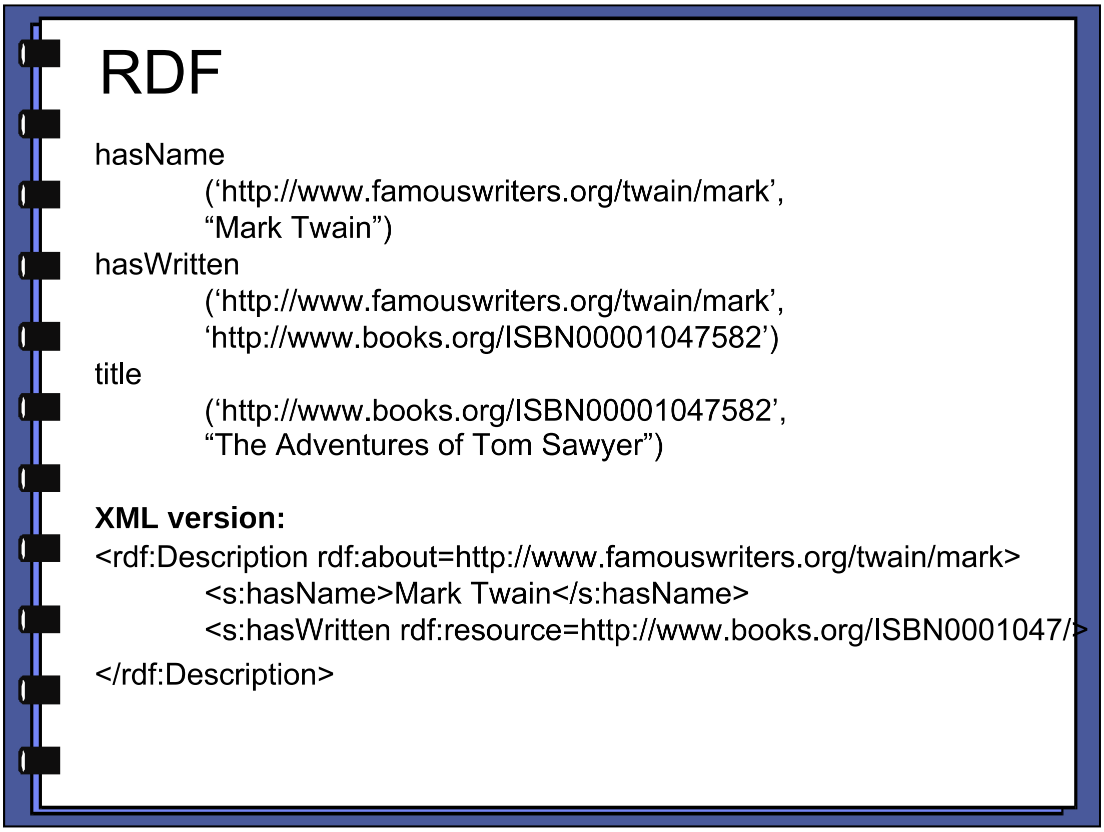
hasWritten
 (‘http://www.famouswriters.org/tolkein/’,

 http://www.books.org/ISBN00001047582’)



RDF and RDFS

- RDFS defines the ontology
  - classes and their properties and relationships
  - what concepts do we want to reason about and how are they related
  - there are authors, and authors write books
- RDF defines the instances of these classes and their properties
  - Mark Twain is an author
  - Mark Twain wrote “Adventures of Tom Sawyer”
  - “Adventures of Tom Sawyer” is a book
- Notation: RDF(S) = RDF + RDFS



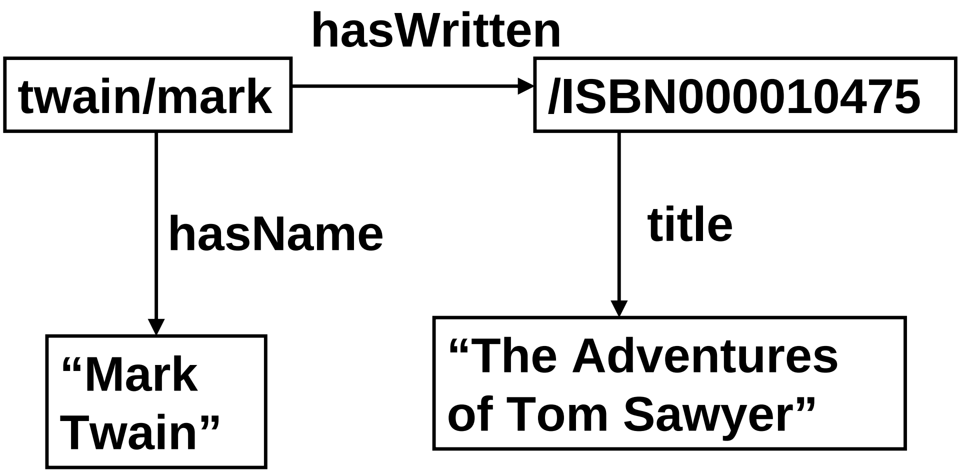
RDF

```
hasName
          (‘http://www.famouswriters.org/twain/mark’,
          “Mark Twain”)
hasWritten
          (‘http://www.famouswriters.org/twain/mark’,
          ‘http://www.books.org/ISBN00001047582’)
title
          (‘http://www.books.org/ISBN00001047582’,
          “The Adventures of Tom Sawyer”)
```

XML version:

```
<rdf:Description rdf:about=http://www.famouswriters.org/twain/mark>
          <s:hasName>Mark Twain</s:hasName>
          <s:hasWritten rdf:resource=http://www.books.org/ISBN0001047/>
</rdf:Description>
```

An example RDF data graph



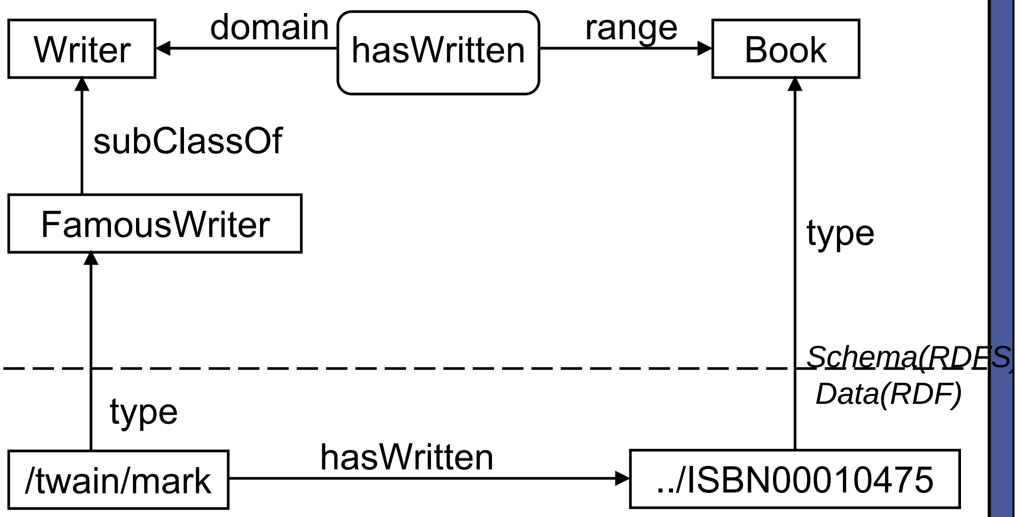
RDF(S) definitions

subclassof(FamousWriter, Writer)

type(‘http://www.books.org/ISBN00001047582’,
‘http://www.description.org/schema#Book’)

An example RDF Schema

Annotation of WWW resources and *semantic* links



Writer

domain

hasWritten

range

Book

subClassOf

FamousWriter

type

Schema(RDFS)

Data(RDF)

type

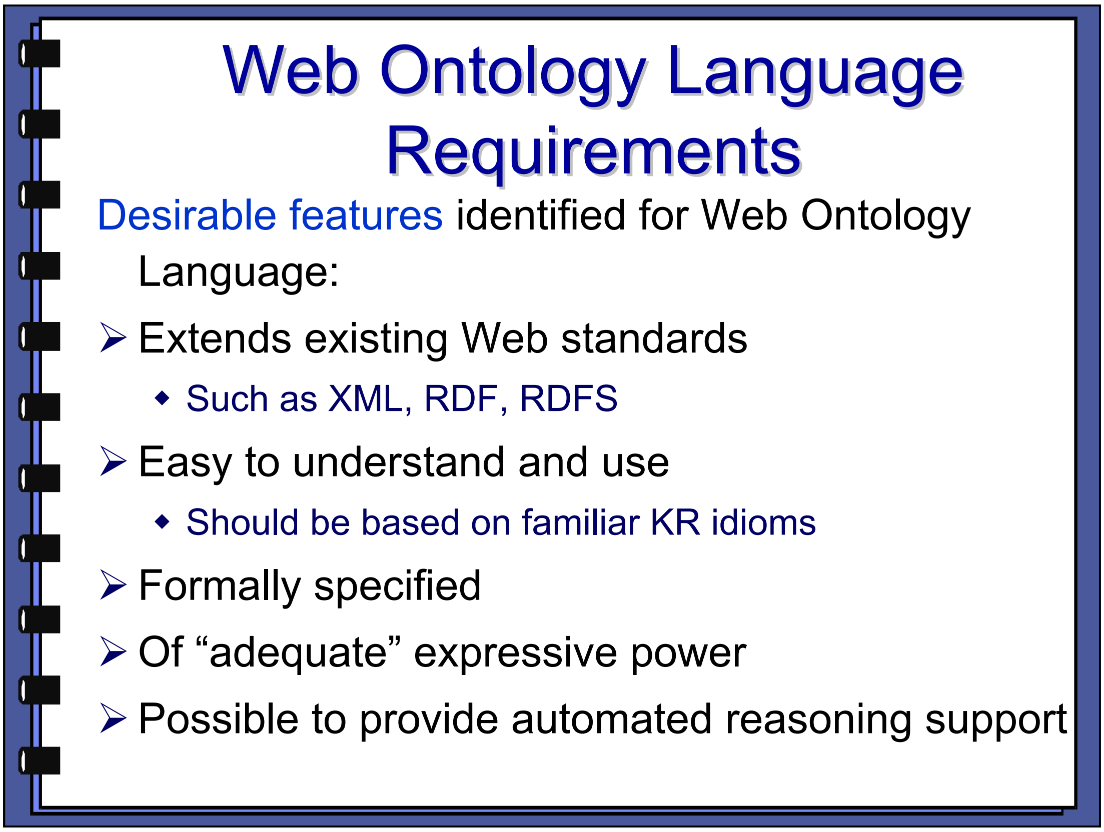
/twain/mark

hasWritten

../ISBN00010475

Conclusions about RDF(S)

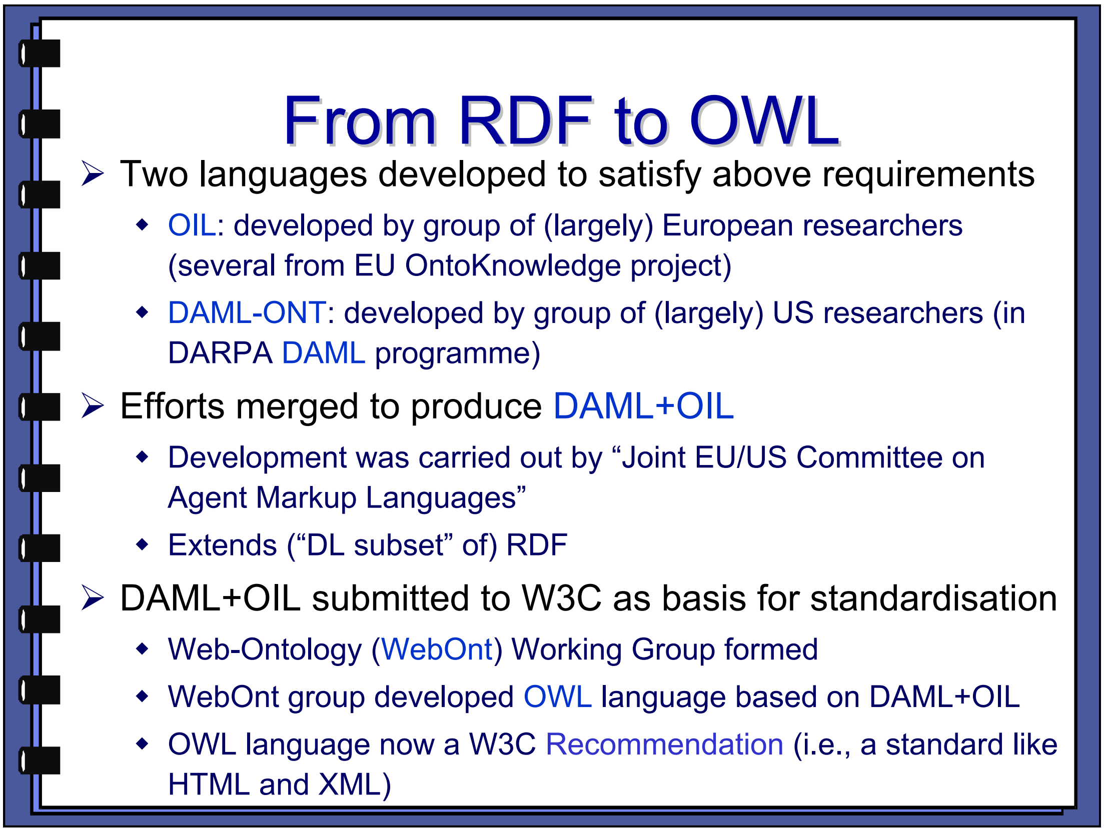
- Next step up from plain XML:
  - (small) **ontological commitment** to modeling primitives
  - possible to define **vocabulary**
- However:
  - **no** precisely described meaning
  - **no** inference model



Web Ontology Language Requirements

Desirable features identified for Web Ontology Language:

- Extends existing Web standards
  - Such as XML, RDF, RDFS
- Easy to understand and use
  - Should be based on familiar KR idioms
- Formally specified
- Of “adequate” expressive power
- Possible to provide automated reasoning support



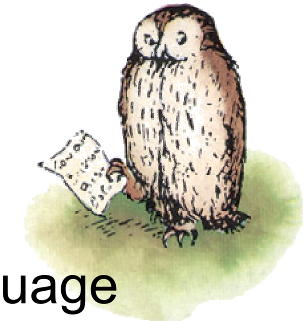
From RDF to OWL

- Two languages developed to satisfy above requirements
  - OIL: developed by group of (largely) European researchers (several from EU OntoKnowledge project)
  - DAML-ONT: developed by group of (largely) US researchers (in DARPA DAML programme)
- Efforts merged to produce DAML+OIL
  - Development was carried out by “Joint EU/US Committee on Agent Markup Languages”
  - Extends (“DL subset” of) RDF
- DAML+OIL submitted to W3C as basis for standardisation
  - Web-Ontology (WebOnt) Working Group formed
  - WebOnt group developed OWL language based on DAML+OIL
  - OWL language now a W3C Recommendation (i.e., a standard like HTML and XML)

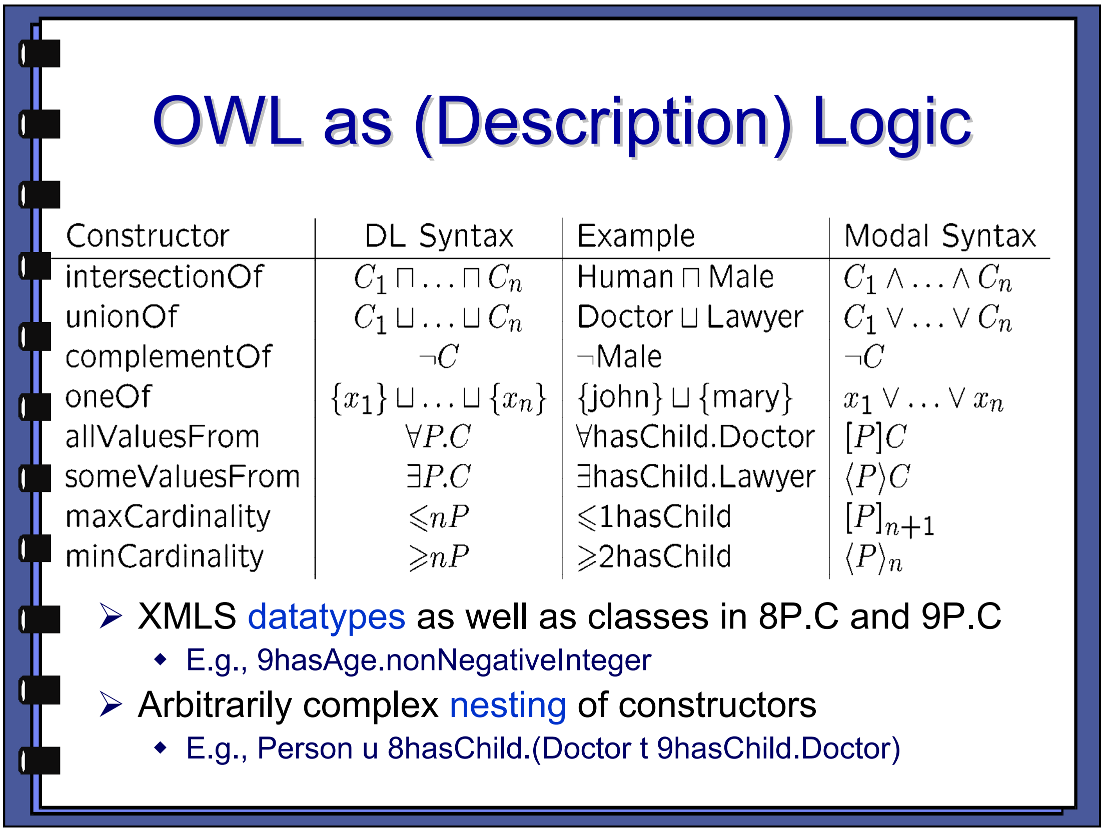
OWL Language

- OWL is based on Description Logics knowledge representation formalism
- OWL (DL) benefits from many years of DL research:
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)
- Three species of OWL
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to FOL fragment (¼ DAML+OIL)
  - OWL Lite is “easier to implement” subset of OWL DL

Why OWL?



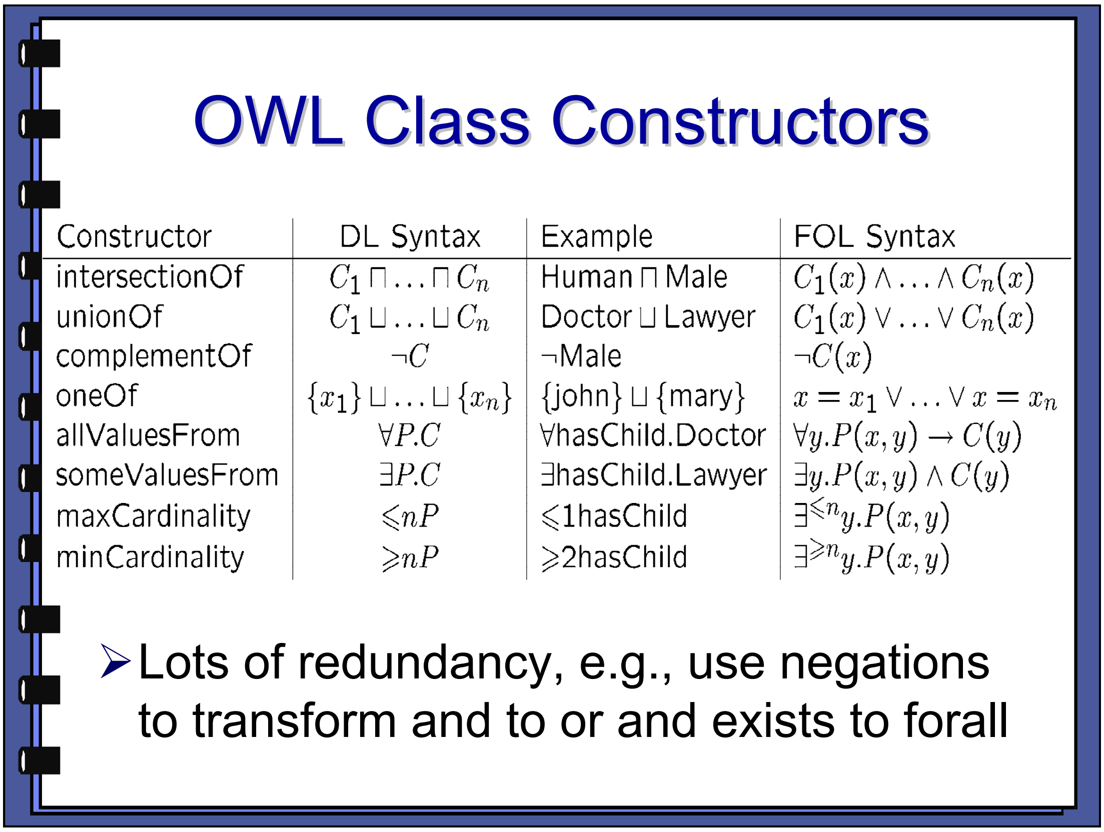
- OWL = Web Ontology Language
- Owl’s superior intelligence is known throughout the Hundred Acre Wood, as are his talents for Writing, Spelling, other Educated and Special tasks.
- *"My spelling is Wobbly. It's good spelling, but it Wobbles, and **the letters get in the wrong places**."*



OWL as (Description) Logic

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1 \wedge \ldots \wedge C_n$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1 \vee \ldots \vee C_n$
complementOf	$\neg C$	¬Male	$\neg C$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x_1 \vee \ldots \vee x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leqslant nP$	⩽1hasChild	$[P]_{n+1}$
minCardinality	$\geqslant nP$	⩾2hasChild	$\langle P \rangle_n$

- XMLS datatypes as well as classes in 8P.C and 9P.C
  - E.g., 9hasAge.nonNegativeInteger
- Arbitrarily complex nesting of constructors
  - E.g., Person u 8hasChild.(Doctor t 9hasChild.Doctor)



OWL Class Constructors

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	¬Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	⩽1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	⩾2hasChild	$\exists^{\geqslant n} y.P(x,y)$

- Lots of redundancy, e.g., use negations to transform and to or and exists to forall

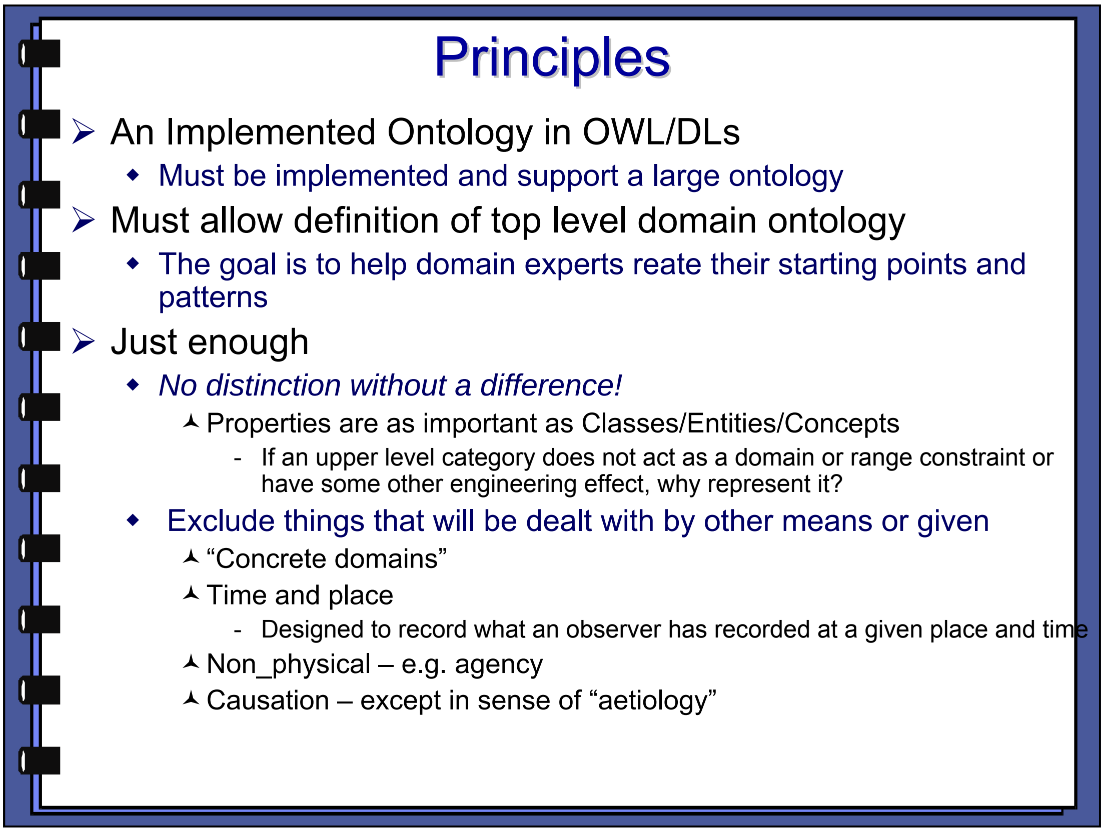
OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ ¬Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ ¬{peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$

- Axioms (mostly) reducible to inclusion (v)
  - C ´ D iff both C v D and D v C

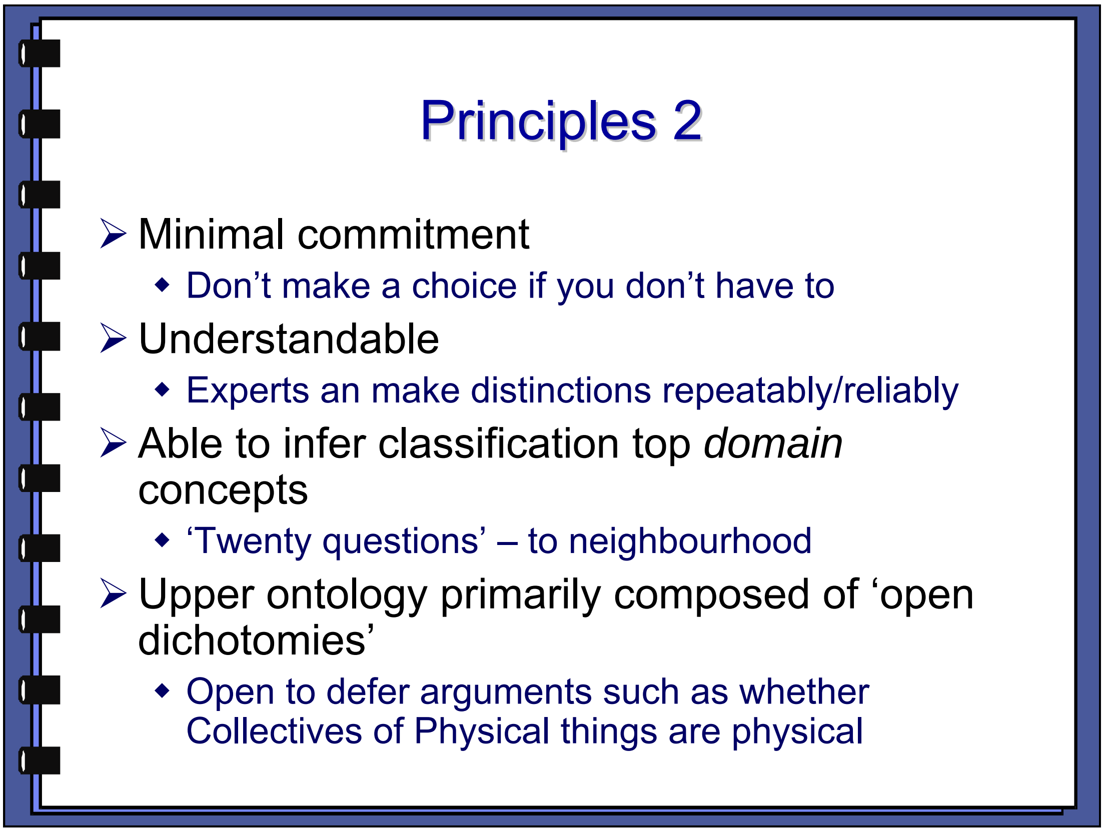
OWL

- DLs are family of object oriented KR formalisms related to frames and Semantic networks
  - Distinguished by formal semantics and inference services
- Semantic Web aims to make web resources accessible to automated processes
  - Ontologies will play key role by providing vocabulary for semantic markup
- OWL is a DL based ontology language designed for the Web
  - Exploits existing standards: XML, RDF(S)
  - Adds KR idioms from object oriented and frame systems
  - W3C recommendation and already widely adopted in e-Science
  - DL provides formal foundations and reasoning support



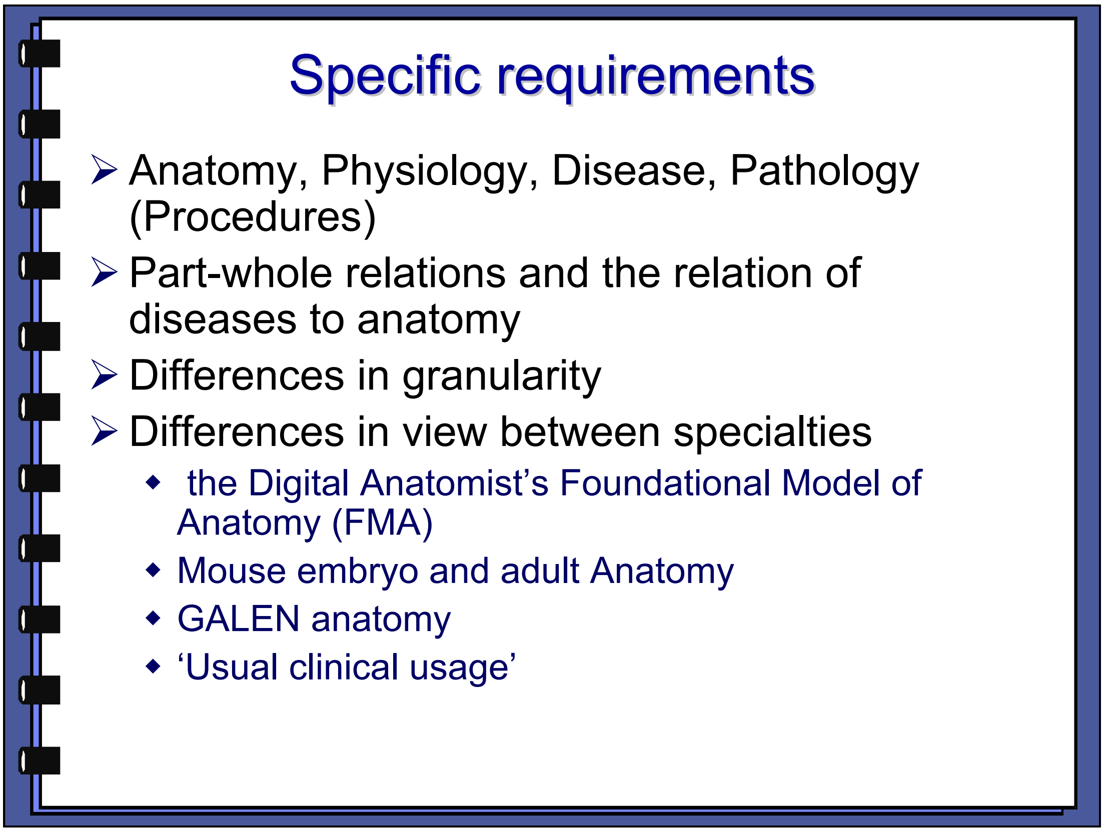
Principles

- An Implemented Ontology in OWL/DLs
  - Must be implemented and support a large ontology
- Must allow definition of top level domain ontology
  - The goal is to help domain experts reate their starting points and patterns
- Just enough
  - *No distinction without a difference!*
    - Properties are as important as Classes/Entities/Concepts
      - If an upper level category does not act as a domain or range constraint or have some other engineering effect, why represent it?
  - Exclude things that will be dealt with by other means or given
    - “Concrete domains”
    - Time and place
      - Designed to record what an observer has recorded at a given place and time
    - Non_physical – e.g. agency
    - Causation – except in sense of “aetiology”



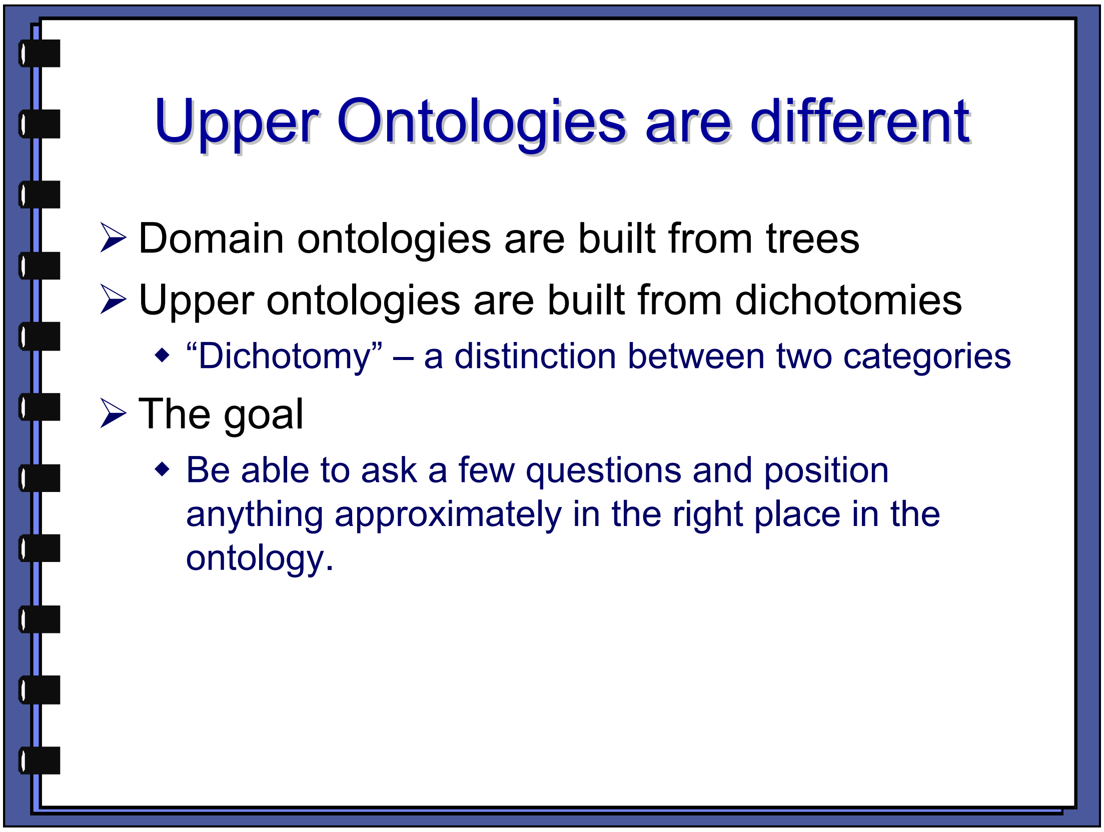
Principles 2

- Minimal commitment
  - Don’t make a choice if you don’t have to
- Understandable
  - Experts an make distinctions repeatably/reliably
- Able to infer classification top *domain* concepts
  - ‘Twenty questions’ – to neighbourhood
- Upper ontology primarily composed of ‘open dichotomies’
  - Open to defer arguments such as whether Collectives of Physical things are physical



Specific requirements

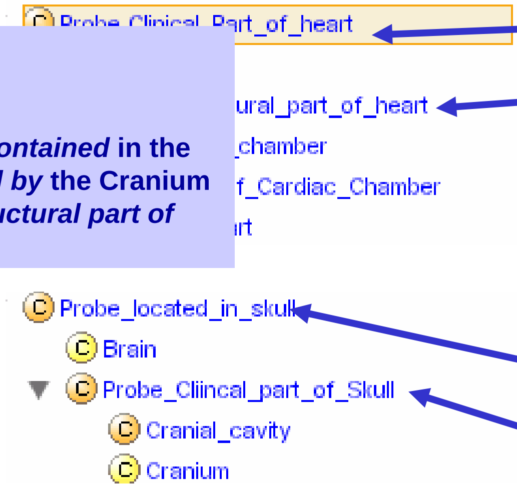
- Anatomy, Physiology, Disease, Pathology (Procedures)
- Part-whole relations and the relation of diseases to anatomy
- Differences in granularity
- Differences in view between specialties
  - the Digital Anatomist's Foundational Model of Anatomy (FMA)
  - Mouse embryo and adult Anatomy
  - GALEN anatomy
  - 'Usual clinical usage'



Upper Ontologies are different

- Domain ontologies are built from trees
- Upper ontologies are built from dichotomies
  - "Dichotomy" – a distinction between two categories
- The goal
  - Be able to ask a few questions and position anything approximately in the right place in the ontology.

Sufficient to support multiple “views”



Clinician’s view: **Pericardium is part of heart & Pericardiitis is a kind of Heart Disease**

Formally: **The Brain is *contained* in the Cavity *defined by* the Cranium which *is a structural part of* the skull.**

Anatomist’s view: **Pericardium is a distinct organ that develops separately from Heart**

Both views: **The Brain is located in the skul but not part of the skull**

OntoClean & Dolce
One Upper Ontology

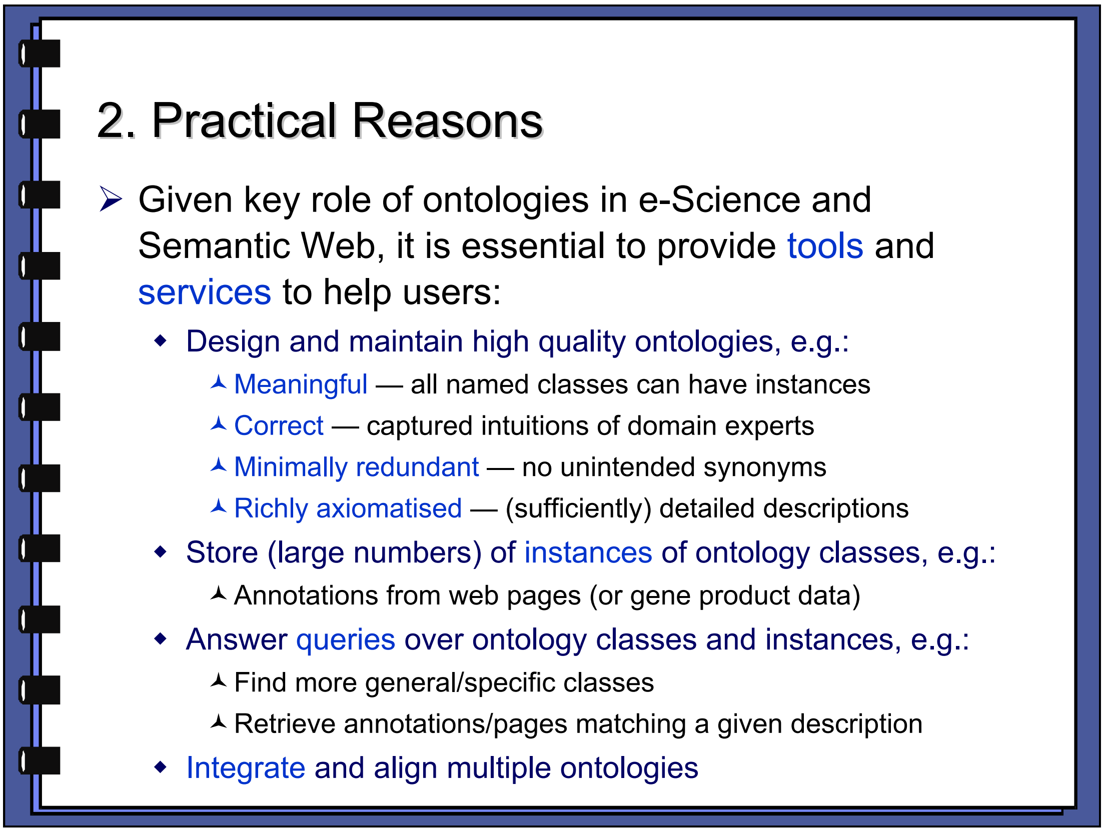
- Owl version Provided in the lab – see also URL
  - http://www.loa-cnr.it/DOLCE.html
  - Vocabulary:
    - “Predicate” – “Class”
      - i.e. a Class is equivalent to a one-place predicate
        - the Class ‘C’ is equivalent to the predicate C(x)
    - Sortal – “Self-standing entity”
      - To a good first approximation
    - “Amount of matter” - “Mass_entity”
  - OntoClean is a meta ontology & methodology for ontology building
    - An ontology about the properties of concepts
    - used to constrain
    - DOLCE is an upper ontology that conforms to Ontoclean

Reasoning with OWL

Why do we want/need to reason with OWL?

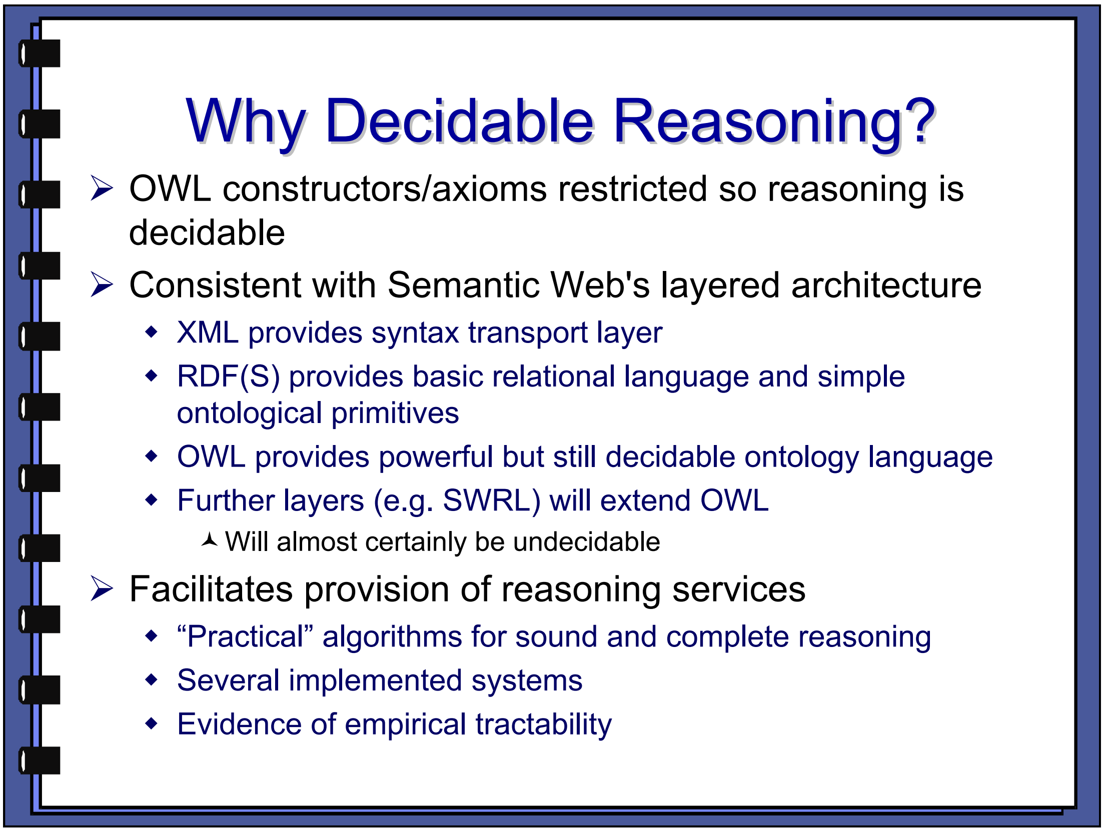
1. Philosophical Reasons

- Semantic Web aims at “machine understanding”
- Understanding closely related to reasoning
  - Recognising semantic similarity in spite of syntactic differences
  - Drawing conclusions that are not explicitly stated



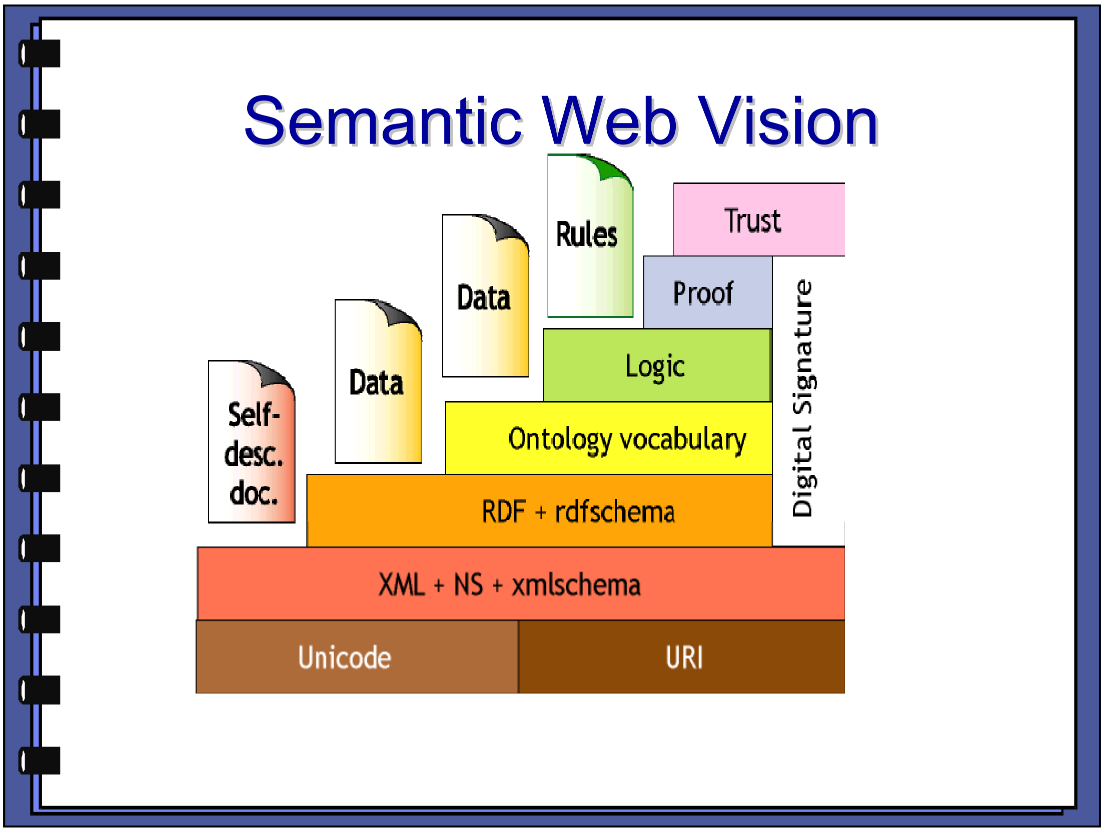
2. Practical Reasons

- Given key role of ontologies in e-Science and Semantic Web, it is essential to provide tools and services to help users:
  - Design and maintain high quality ontologies, e.g.:
    - Meaningful — all named classes can have instances
    - Correct — captured intuitions of domain experts
    - Minimally redundant — no unintended synonyms
    - Richly axiomatised — (sufficiently) detailed descriptions
  - Store (large numbers) of instances of ontology classes, e.g.:
    - Annotations from web pages (or gene product data)
  - Answer queries over ontology classes and instances, e.g.:
    - Find more general/specific classes
    - Retrieve annotations/pages matching a given description
  - Integrate and align multiple ontologies



Why Decidable Reasoning?

- OWL constructors/axioms restricted so reasoning is decidable
- Consistent with Semantic Web's layered architecture
  - XML provides syntax transport layer
  - RDF(S) provides basic relational language and simple ontological primitives
  - OWL provides powerful but still decidable ontology language
  - Further layers (e.g. SWRL) will extend OWL
    - Will almost certainly be undecidable
- Facilitates provision of reasoning services
  - “Practical” algorithms for sound and complete reasoning
  - Several implemented systems
  - Evidence of empirical tractability



Semantic Web Vision

Rules

Trust

Data

Proof

Data

Logic

Digital Signature

Self-desc. doc.

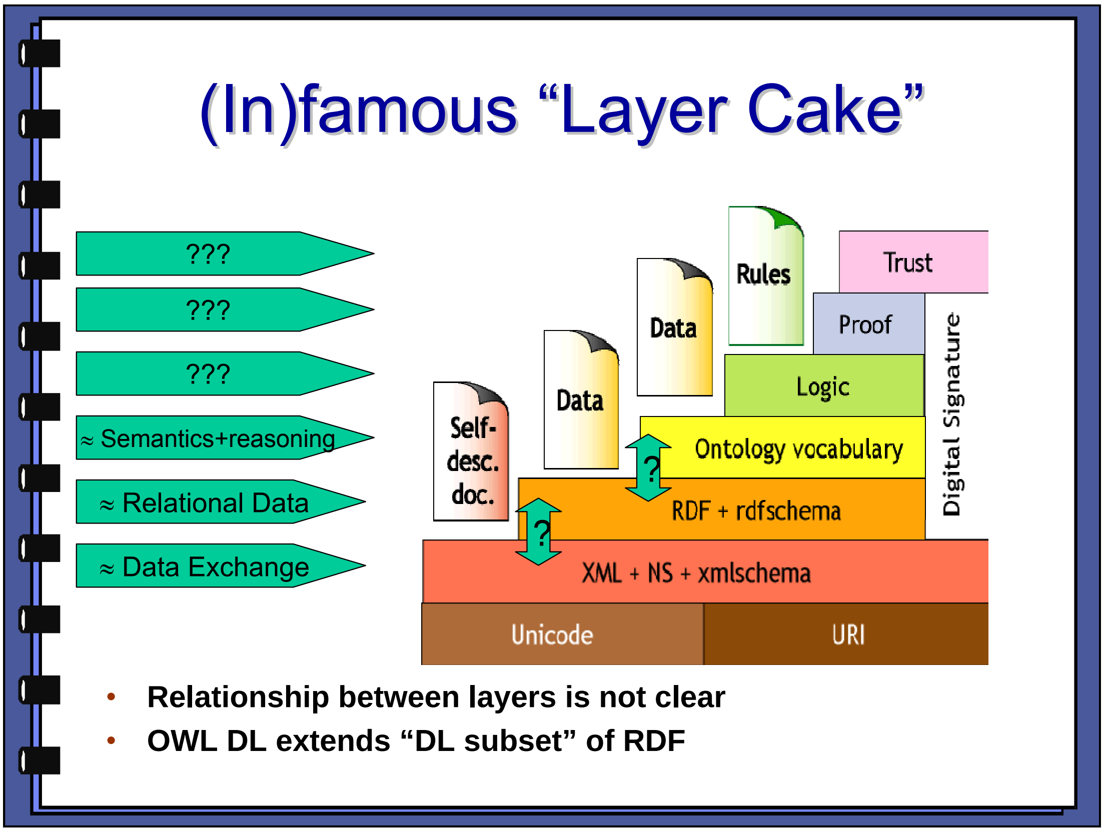
Ontology vocabulary

RDF + rdfschema

XML + NS + xmlschema

Unicode

URI



(In)famous “Layer Cake”

???

???

???

≈ Semantics+reasoning

≈ Relational Data

≈ Data Exchange

Rules

Trust

Data

Proof

Data

Logic

Digital Signature

Self-desc. doc.

Ontology vocabulary

?

RDF + rdfschema

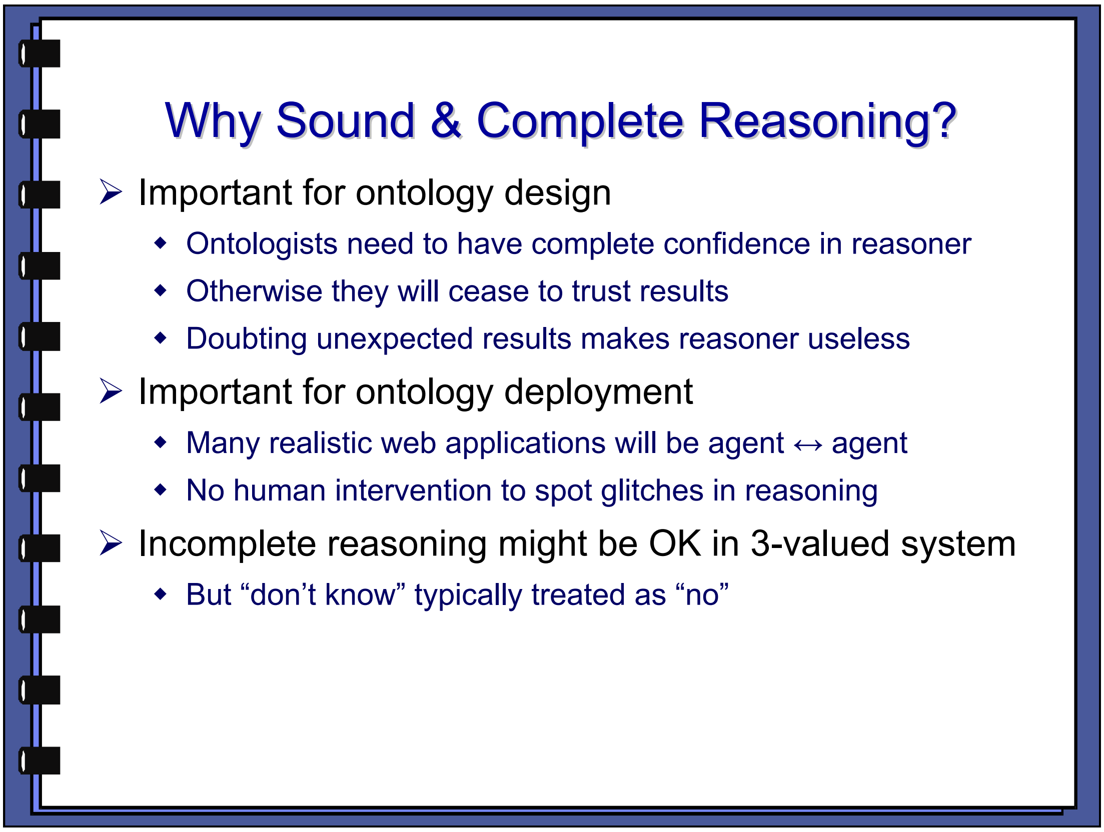
?

XML + NS + xmlschema

Unicode

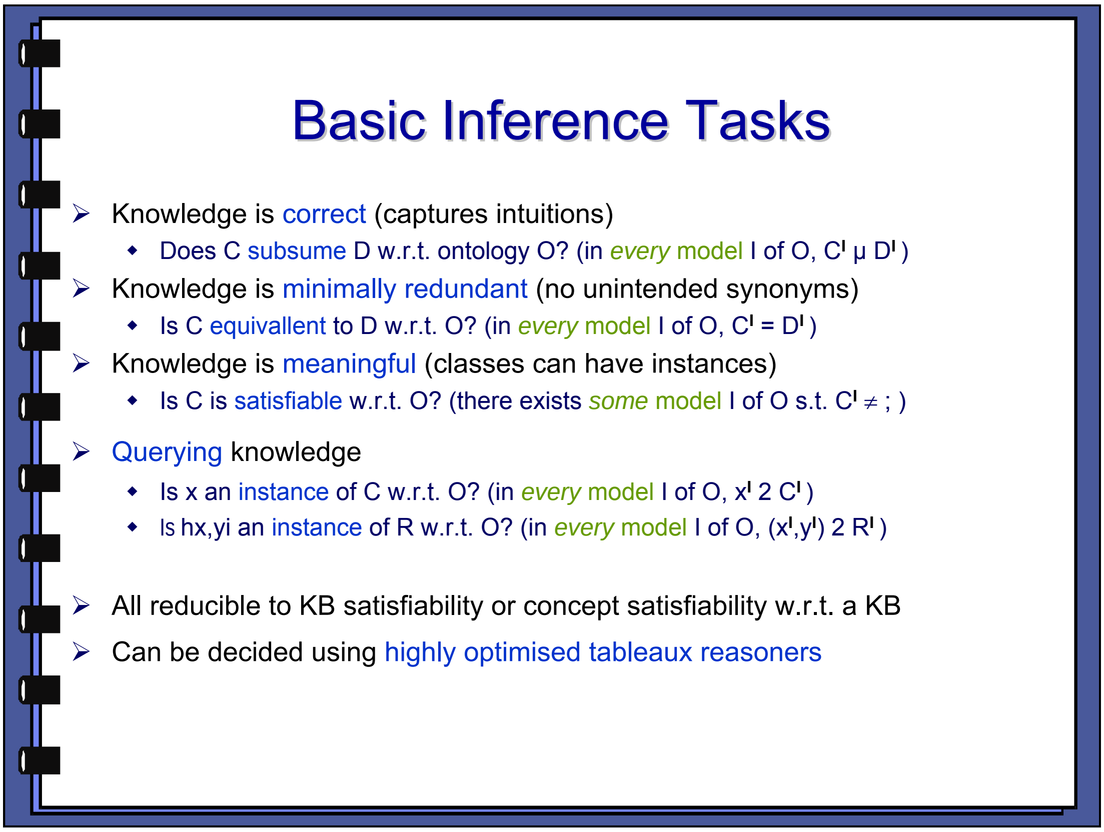
URI

- Relationship between layers is not clear
- OWL DL extends “DL subset” of RDF



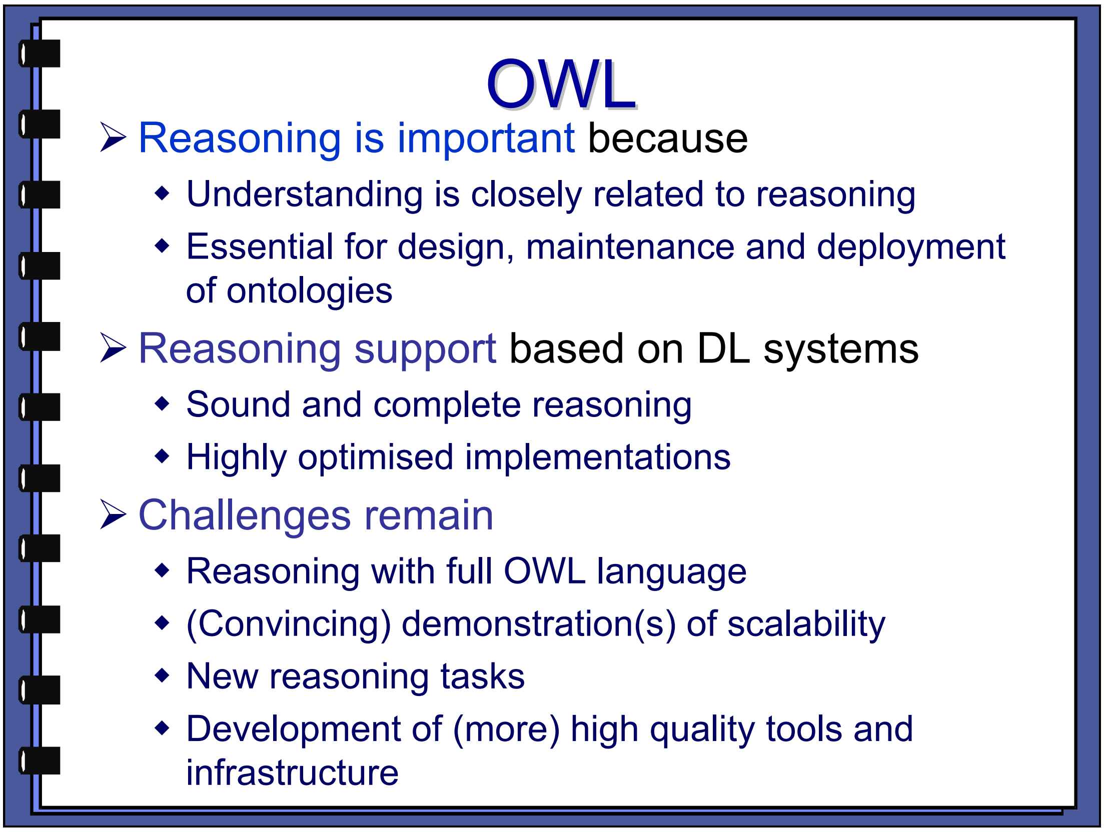
Why Sound & Complete Reasoning?

- Important for ontology design
  - Ontologists need to have complete confidence in reasoner
  - Otherwise they will cease to trust results
  - Doubting unexpected results makes reasoner useless
- Important for ontology deployment
  - Many realistic web applications will be agent ↔ agent
  - No human intervention to spot glitches in reasoning
- Incomplete reasoning might be OK in 3-valued system
  - But “don’t know” typically treated as “no”



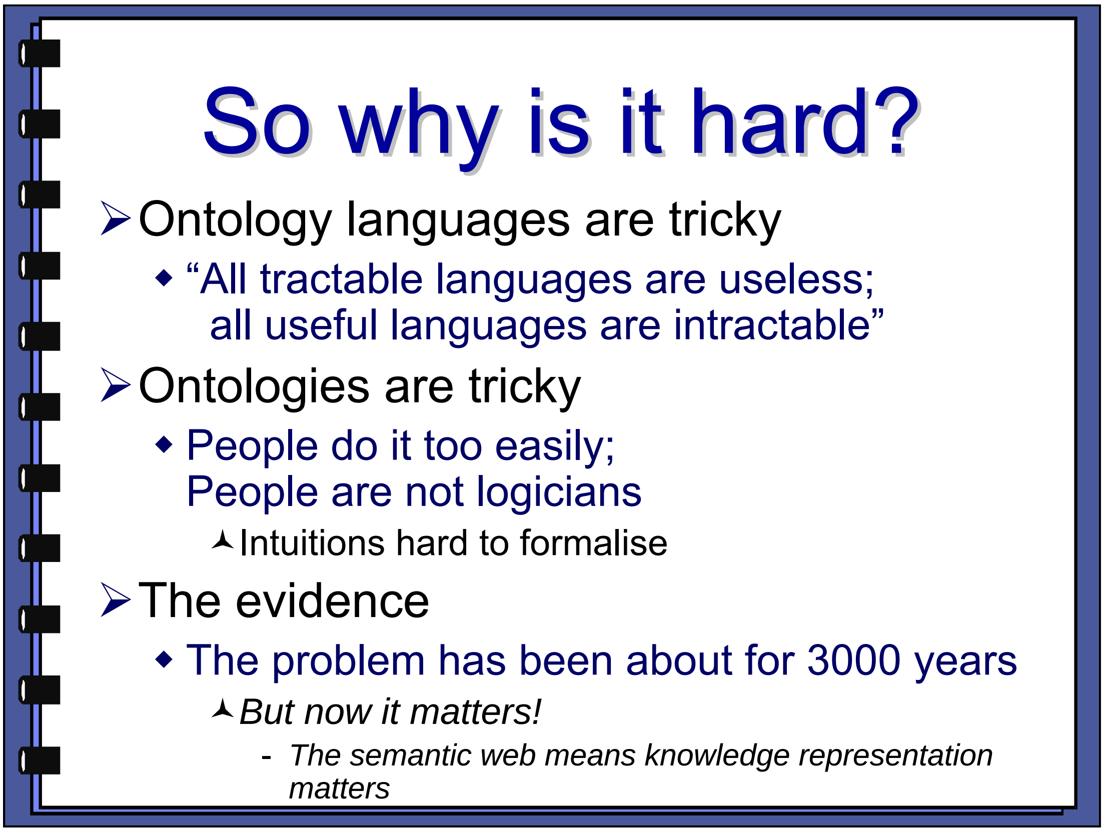
Basic Inference Tasks

- Knowledge is correct (captures intuitions)
  - Does C subsume D w.r.t. ontology O? (in *every* model I of O, C^I µ D^I)
- Knowledge is minimally redundant (no unintended synonyms)
  - Is C equivallent to D w.r.t. O? (in *every* model I of O, $C^I = D^I$)
- Knowledge is meaningful (classes can have instances)
  - Is C is satisfiable w.r.t. O? (there exists *some* model I of O s.t. $C^I \neq$;)
- Querying knowledge
  - Is x an instance of C w.r.t. O? (in *every* model I of O, x^I 2 C^I)
  - Is hx,yi an instance of R w.r.t. O? (in *every* model I of O, (x^I,y^I) 2 R^I)
- All reducible to KB satisfiability or concept satisfiability w.r.t. a KB
- Can be decided using highly optimised tableaux reasoners



OWL

- Reasoning is important because
  - Understanding is closely related to reasoning
  - Essential for design, maintenance and deployment of ontologies
- Reasoning support based on DL systems
  - Sound and complete reasoning
  - Highly optimised implementations
- Challenges remain
  - Reasoning with full OWL language
  - (Convincing) demonstration(s) of scalability
  - New reasoning tasks
  - Development of (more) high quality tools and infrastructure



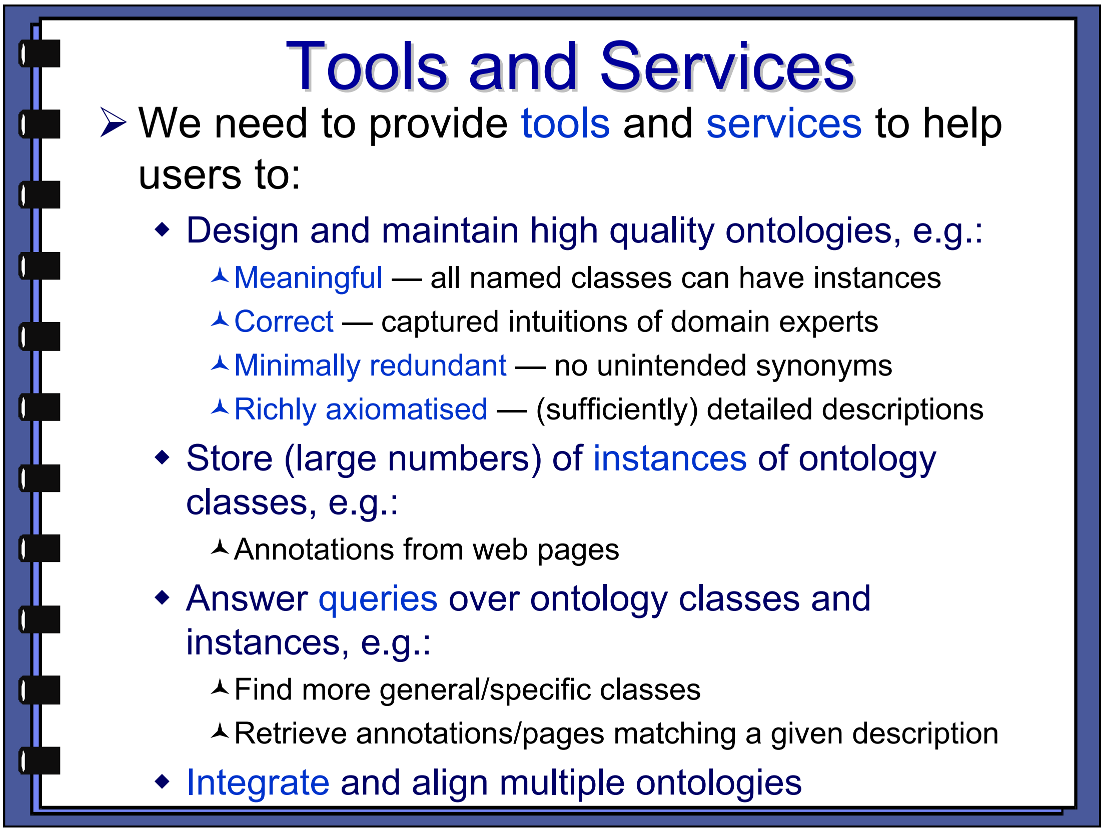
So why is it hard?

- Ontology languages are tricky
  - “All tractable languages are useless;
    all useful languages are intractable”
- Ontologies are tricky
  - People do it too easily;
    People are not logicians
    - Intuitions hard to formalise
- The evidence
  - The problem has been about for 3000 years
    - *But now it matters!*
      - *The semantic web means knowledge representation matters*

Semantic Web & Ontologies Applications

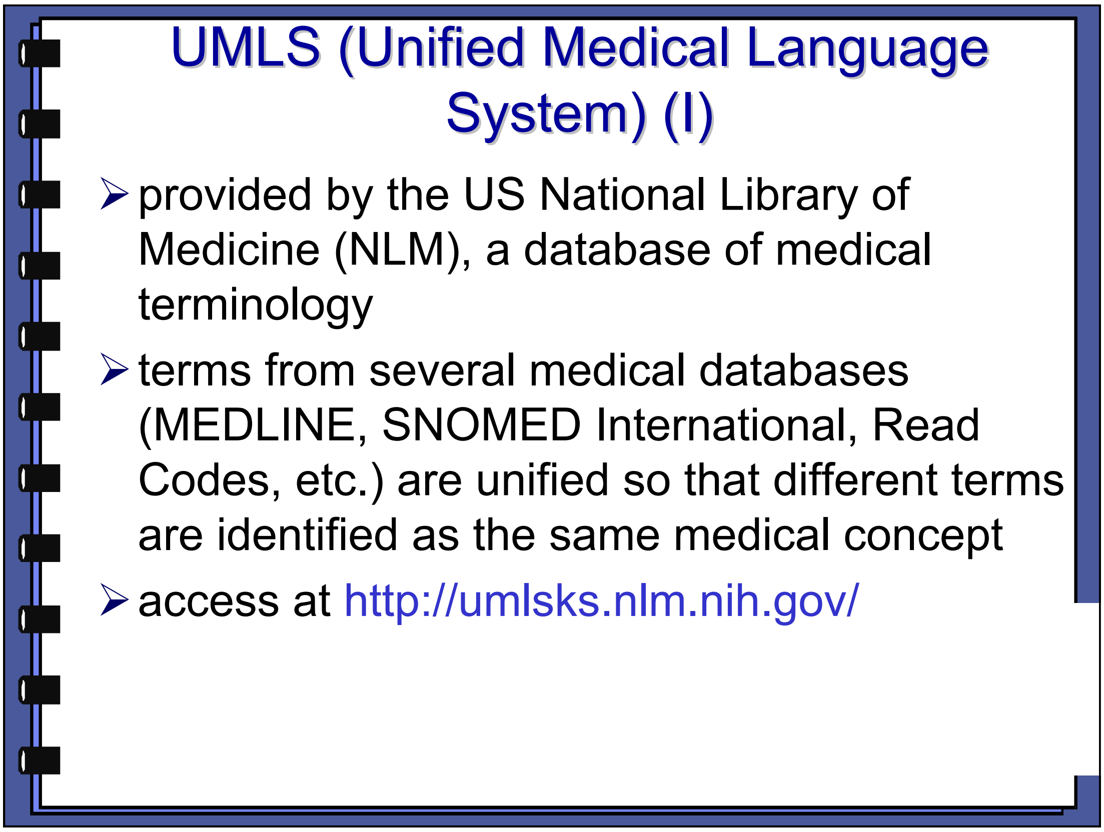
Where else are ontologies used?

- Bioinformatics
  - The Gene Ontology
  - The Protein Ontology (MGED)
- Medicine
  - “The terminology wars”
- Linguistics
- Database integration
- User interface design
- Fractal Indexing



Tools and Services

- We need to provide tools and services to help users to:
  - Design and maintain high quality ontologies, e.g.:
    - Meaningful — all named classes can have instances
    - Correct — captured intuitions of domain experts
    - Minimally redundant — no unintended synonyms
    - Richly axiomatised — (sufficiently) detailed descriptions
  - Store (large numbers) of instances of ontology classes, e.g.:
    - Annotations from web pages
  - Answer queries over ontology classes and instances, e.g.:
    - Find more general/specific classes
    - Retrieve annotations/pages matching a given description
  - Integrate and align multiple ontologies



UMLS (Unified Medical Language System) (I)

- provided by the US National Library of Medicine (NLM), a database of medical terminology
- terms from several medical databases (MEDLINE, SNOMED International, Read Codes, etc.) are unified so that different terms are identified as the same medical concept
- access at http://umlsks.nlm.nih.gov/

UMLS (Unified Medical Language System) (II)

- **UMLS Knowledge Sources:**
  - Metathesaurus provides the concordance of medical concepts:
    - 730,000 concepts
    - 1.5 million concept names in different source vocabularies

Open Biological Ontologies

- Various ontologies in the biological domain
- obo.sourceforge.net
- e.g. Gene Ontology (www.geneontology.org)

“Biologists currently waste a lot of time and effort in searching for all of the available information about each small area of research. This is hampered further by the wide variations in terminology that may be common usage at any given time, and that inhibit effective searching by computers as well as people. For example, if you were searching for new targets for antibiotics, you might want to find all the gene products that are involved in bacterial protein synthesis, and that have significantly different sequences or structures from those in humans. But if one database describes these molecules as being involved in 'translation', whereas another uses the phrase 'protein synthesis', it will be difficult for you — and even harder for a computer — to find functionally equivalent terms. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases.”

- Hundreds of classes

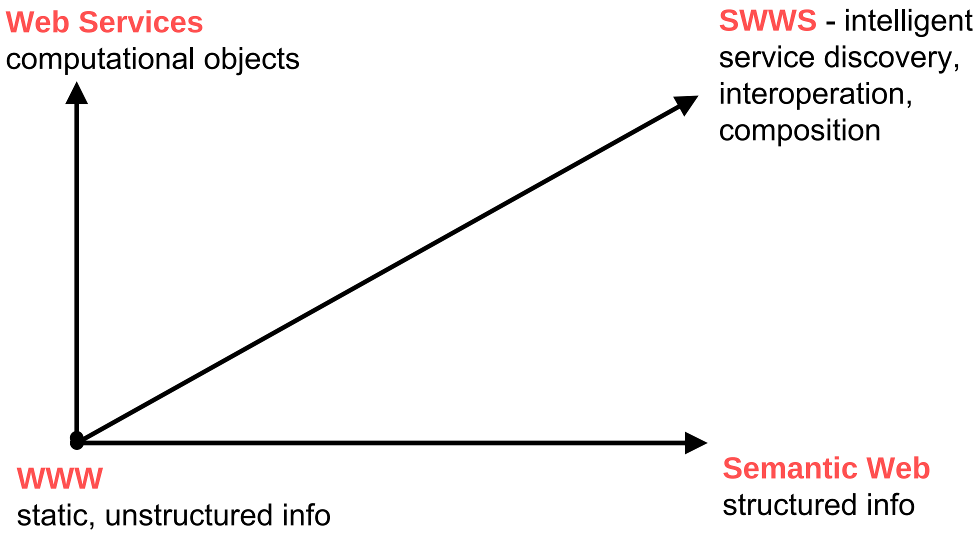
Semantic Web areas of application

- Semantic Web & Knowledge Management
  - SEKT (sekt.semanticweb.org)
- Semantic Web-enabled Web Services
  - SWWS (swws.semanticweb.org)
  - DIP (dip.semanticweb.org)

Semantic Web & KM

- Making WWW information machine processable
  - annotation via ontologies & metadata
  - offers prospect of enhanced knowledge management
    - “Rank all the documents containing the word Tolkien”
    - “Show me the non-fiction books written by Tolkien about philology before 1940”
    - Data integration
  - significant research & technology challenges are outstanding

Semantic Web-enabled Web Services



Web Services
computational objects

SWWS - intelligent service discovery, interoperation, composition

WWW
static, unstructured info

Semantic Web
structured info

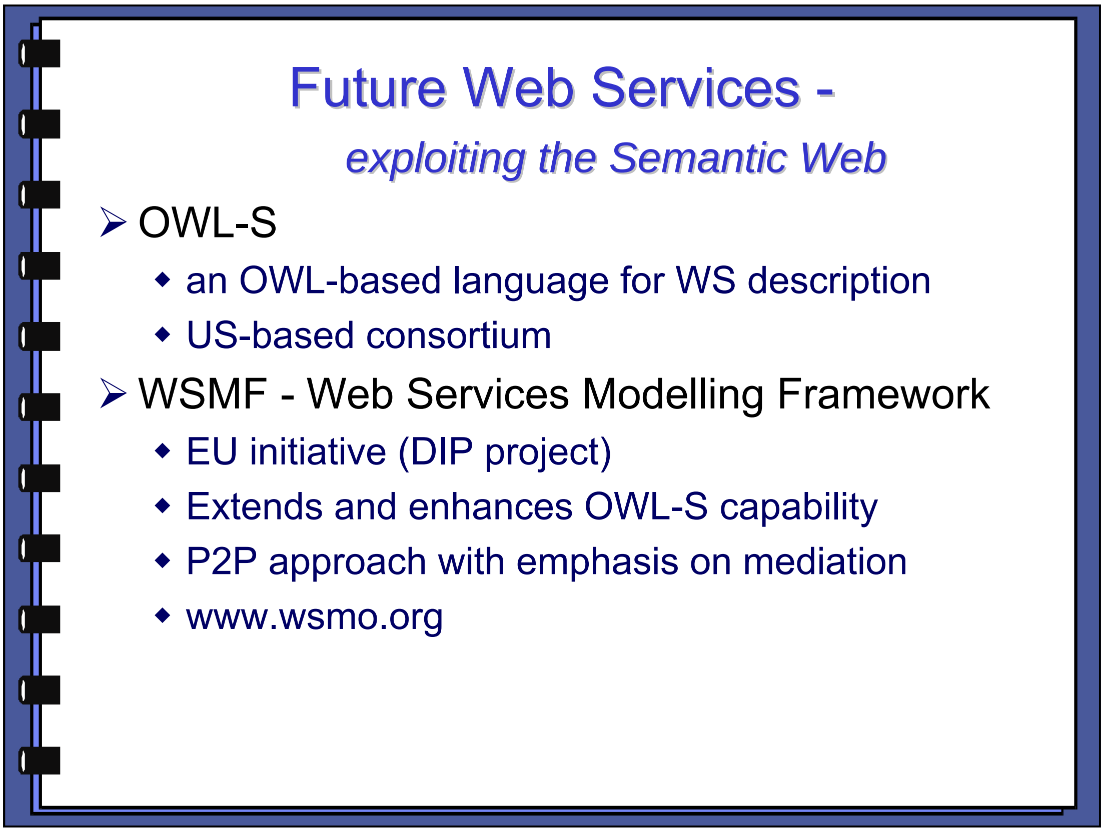
Current Web Services

- UDDI, WDSL, SOAP
  - Web Service discovery and description
  - No *semantic* (formal) description
  - Don’t support *automatic*
    - web service discovery
    - mediation
    - composition into complex services
    - negotiation

Semantic Web Services

- ➢ Automatic discovery
  Find a book selling service
- ➢ Automatic invocation
  Purchase the latest Delia Smith book
- ➢ Automatic composition and interoperation
  Purchase the cheapest latest Delia Smith book
- ➢ Automatic execution monitoring
  What is the status of my book order?

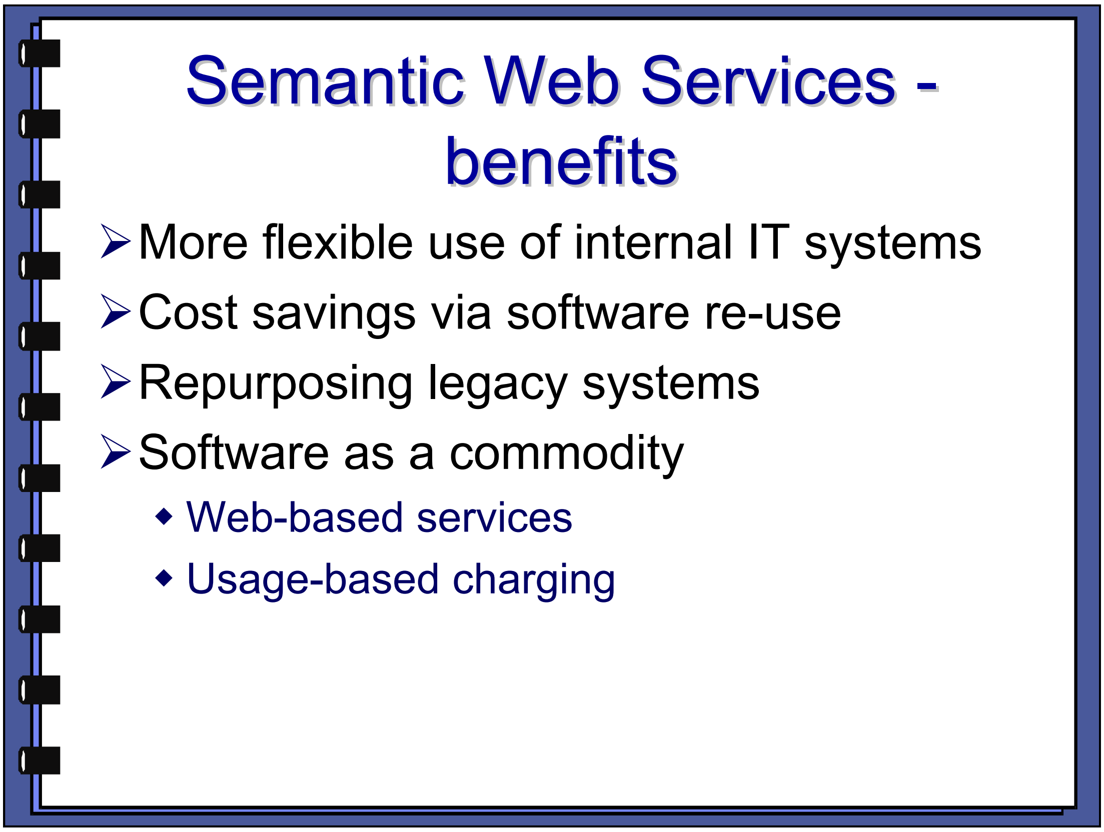
Future & Challenges



Future Web Services -

exploiting the Semantic Web

- OWL-S
  - an OWL-based language for WS description
  - US-based consortium
- WSMF - Web Services Modelling Framework
  - EU initiative (DIP project)
  - Extends and enhances OWL-S capability
  - P2P approach with emphasis on mediation
  - www.wsmo.org



Semantic Web Services - benefits

- More flexible use of internal IT systems
- Cost savings via software re-use
- Repurposing legacy systems
- Software as a commodity
  - Web-based services
  - Usage-based charging

Major research challenges

- Improve automation of ontology and metadata generation
- Research and develop techniques for ontology management and evolution
- Develop highly-scalable solutions
- Research sound inferencing despite inconsistent models
- Develop semantic knowledge access tools
- Develop methodology for deployment

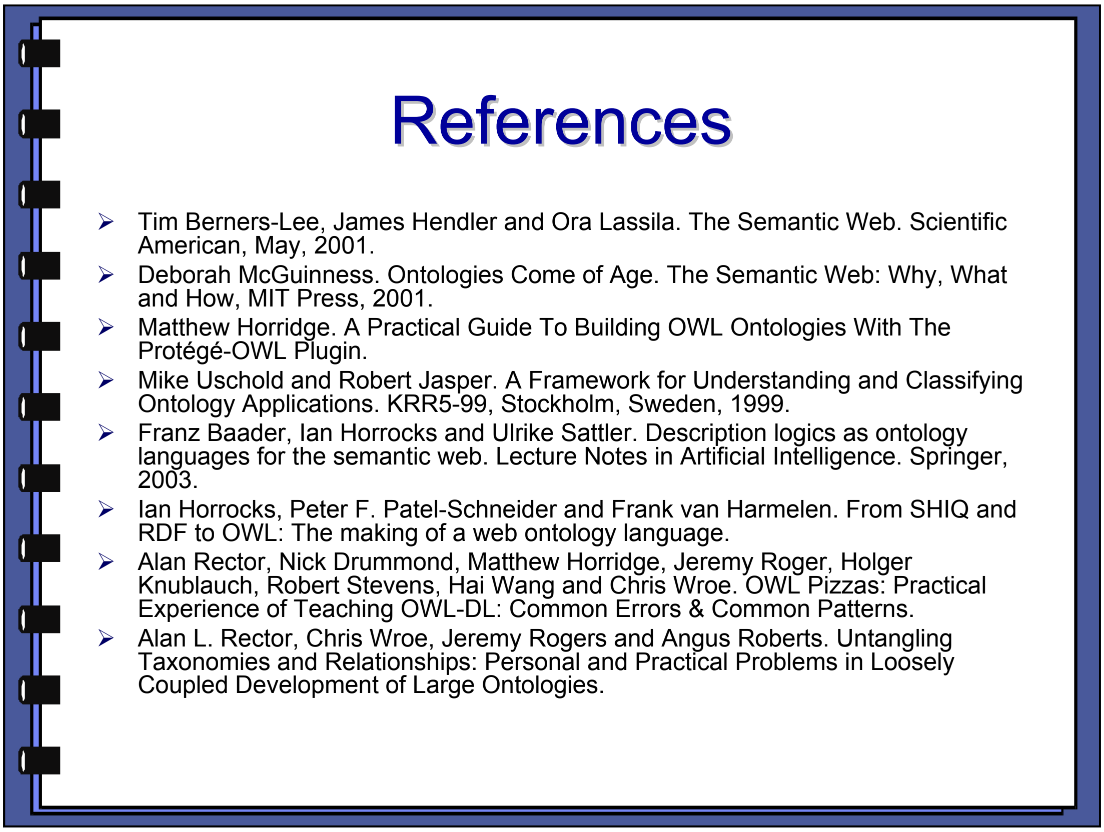
Summary

- The emergence of the Semantic Web
  - machine-processable information
  - Language stack: XML/RDF(S)/OWL
  - Ontologies
- Semantic Web for KM
  - next generation WWW-based KM tools (*inside*)
- Semantic Web for Web Services
  - automating Web Services processes (*buy/sellside*)

"… great implications for a huge range of industrial and social applications" *Gartner Group, Dec 2003*

Thank you for your attention!

Questions?



References

- Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web. Scientific American, May, 2001.
- Deborah McGuinness. Ontologies Come of Age. The Semantic Web: Why, What and How, MIT Press, 2001.
- Matthew Horridge. A Practical Guide To Building OWL Ontologies With The Protégé-OWL Plugin.
- Mike Uschold and Robert Jasper. A Framework for Understanding and Classifying Ontology Applications. KRR5-99, Stockholm, Sweden, 1999.
- Franz Baader, Ian Horrocks and Ulrike Sattler. Description logics as ontology languages for the semantic web. Lecture Notes in Artificial Intelligence. Springer, 2003.
- Ian Horrocks, Peter F. Patel-Schneider and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language.
- Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Roger, Holger Knublauch, Robert Stevens, Hai Wang and Chris Wroe. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns.
- Alan L. Rector, Chris Wroe, Jeremy Rogers and Angus Roberts. Untangling Taxonomies and Relationships: Personal and Practical Problems in Loosely Coupled Development of Large Ontologies.