Overview of this lecture

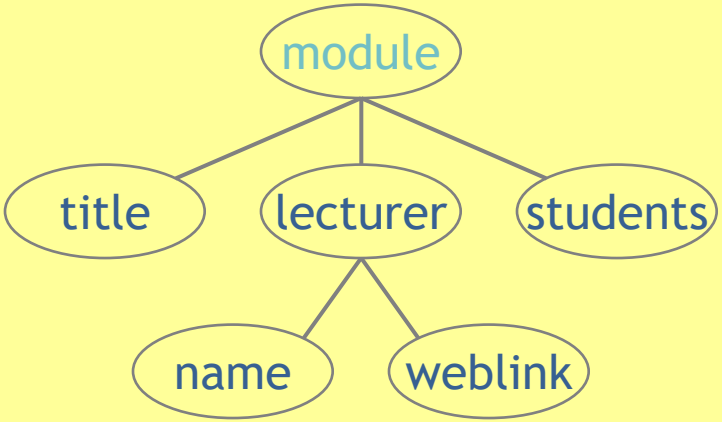
- Introduction to Ontology
- Ontology Components
- Type of Ontology
- Ontology Languages

XML: Document = labeled tree

- node = label + contents

```
<module date="...">
      <title>...</title>
      <lecturer>
            <name>...</name>

      <weblink>...</weblink>
      </lecturer>
      <students>...</students>
</module>
```



- DTD (Document Type Definition)

describe the grammar and structure of permissible XML trees

Introduction(cont.)
Giving Semantics to Annotations

- **Agree on language** used to define meanings
    - E.g., an ontology language
  - Flexible and extensible
    - New terms can be formed by combining existing ones
  - Meaning (semantics) of such terms is formally specified

Introduction
Giving Semantics to Annotations

- External agreement on meaning of annotations
  - **Agree on meaning** of a set of annotation tags
    - E.g., Dublin Core

Term Name: creator	
URI:	http://purl.org/dc/elements/1.1/creator
Label:	Creator
Definition:	An entity primarily responsible for making the resource.

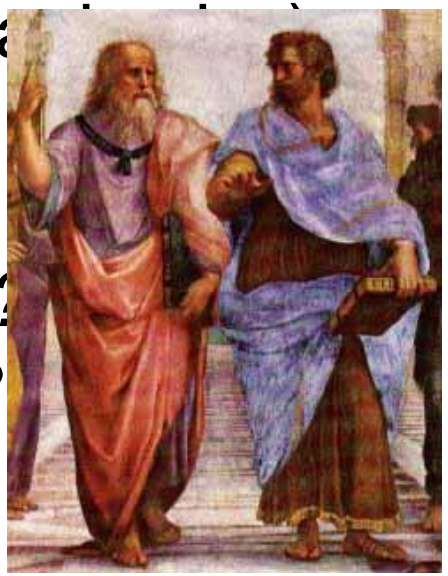
  - Limited flexibility and extensibility
  - Limited number of things can be expressed

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, Meta
Tries to answer the questions:

What characterizes being?
Eventually, what is being?



The origin of ontology

- The word itself comes from two Greek words:
- :ONTO: = existence or being real
- : Logia: = Science or study

What is an ontology?

- *An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary.*

Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; Swartout, W.R. *Enabling Technology for Knowledge Sharing*. AI Magazine. Winter 1991. 36-56

- *An ontology is an explicit specification of a conceptualization.*

Gruber, T. *A translation Approach to portable ontology specifications.* Knowledge Acquisition. Vol. 5. 1993. 199-220

What is an ontology (ii)?

- *An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base.*

B. Swartout; R. Patil; k. Knight; T. Russ. *Toward Distributed Use of Large-Scale Ontologies* **Ontological Engineering.** AAAI-97 Spring Symposium Series. 1997. 138-148

- *An ontology provides the means for describing explicitly the conceptualization behind the knowledge represented in a knowledge base*

A. Bernaras;I. Laresgoiti; J. Correra. *Building and Reusing Ontologies for Electrical Network Applications* **ECAI96. 12th European conference on Artificial Intelligence.** Ed. John Wiley & Sons, Ltd. 298-302

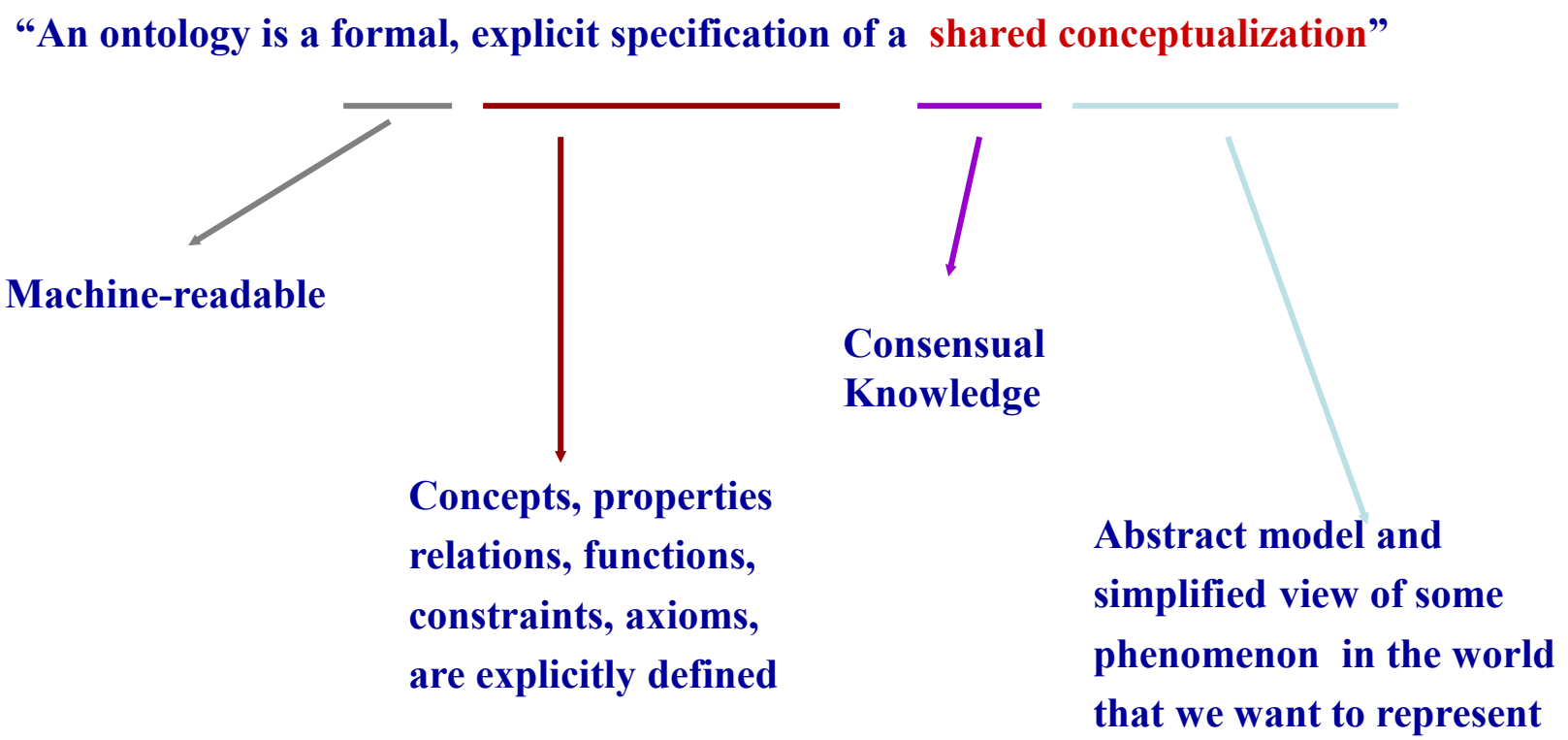
Ontology in Computer Science

- Essentially: a way of encoding domain knowledge

- An ontology consists of:
  - A **vocabulary** used to describe some domain
  - An **explicit specification** of the **intended meaning** of the vocabulary.

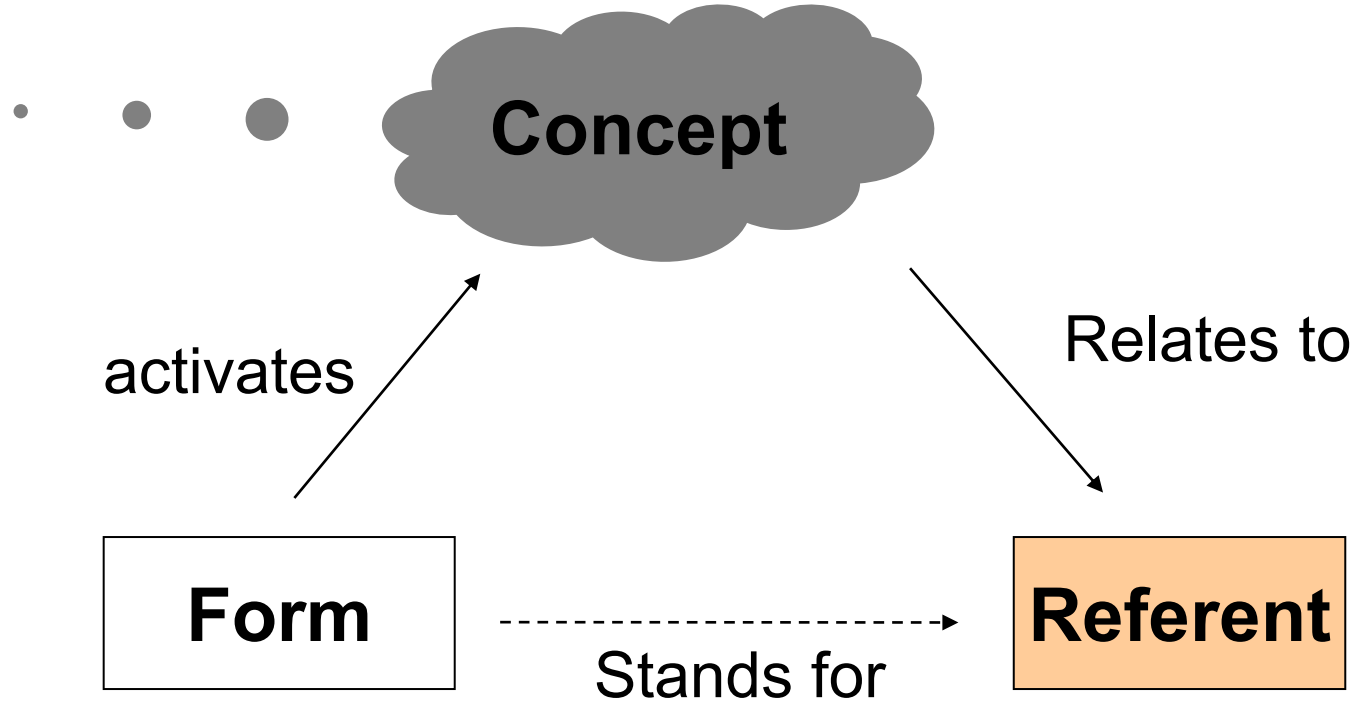
- Ideally, an ontology should:
  - Capture a **shared understanding** of a domain of interest
  - Provide a **formal** and **machine understandable** model
  - An ontology describes the things we want to talk about, including both objects and relationships
- "An explicit specification of a conceptualisation" [Gruber93]

Definitions of Ontologies

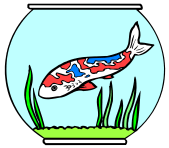


Studer, Benjamins, Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*. 25 (1998) 161-197

Ontology in Linguistics



"Tank"
[Ogden, Richards, 1923]

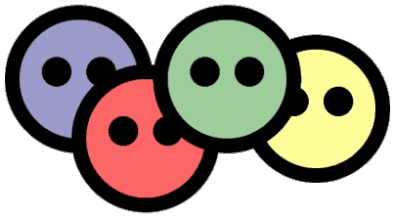


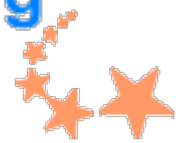
?

Context is important

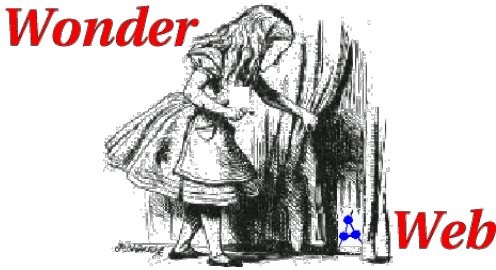
Examples

OpenCyc.org





OpenGALEN



What is an ontology (ii)?

- An ontology is an explicit description of a domain:
  - concepts
  - properties and attributes of concepts
  - Relationships
  - constraints on properties and attributes
  - Individuals
- An ontology defines
  - a common vocabulary
  - a shared understanding

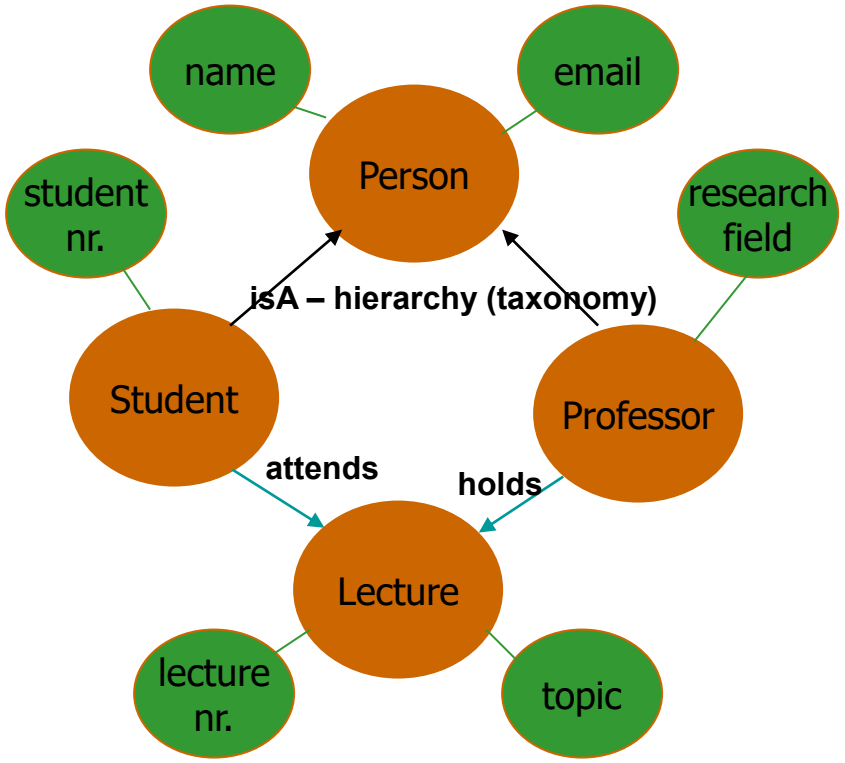
Ontology Example

Concept
conceptual entity of the domain

Attribute
property of a concept

Relation
relationship between concepts or properties

Axiom (constraints)
coherent description between Concepts / Properties / Relations via logical expressions



holds(Professor, Lecture) ⇒ Lecture.topic ∈ Professor.researchField

Ontology Examples

- Taxonomies on the Web
  - Yahoo! Categories
  - Google Directory, Google scholar , Google Recipes
- Catalogs for on-line shopping
  - Amazon.com product catalog
- Domain-specific standard terminology
  - SNOMED Clinical Terms – terminology for clinical medicine
  - UNSPSC - terminology for products and services

Domain standard terminology

- **Unified Medical Language System (UMLS)**
  - UMLS integrates and distributes key terminology, classification and coding standards, and associated resources
  - Promote creation of more effective and interoperable biomedical information systems and services, including electronic health records
- **(Medical Subject Headings) Mesh Largest Medical Library**

- **Broad general ontologies**
  - Wordnet

Ontology Examples

- Taxonomies on the Web
  - Google Directory
- Catalogs for on-line shopping
  - Amazon.com product catalog
- Domain-specific standard terminology
  - Unified Medical Language System (UMLS) and MeSH
- Broad general ontologies
  - Cyc

Web Images Video News Maps **more »**

Preferences

Search Directory

The web organized by topic into categories.

Arts
Movies, Music, Television,...

Business
Industries, Finance, Jobs,...

Computers
Hardware, Internet, Software,...

Games
Board, Roleplaying, Video,...

Health
Alternative, Fitness, Medicine,...

Home
Consumers, Homeowners, Family,...

Kids and Teens
Computers, Entertainment, School,...

News
Media, Newspapers, Current Events,...

Recreation
Food, Outdoors, Travel,...

Reference
Education, Libraries, Maps,...

Regional
Asia, Europe, North America,...

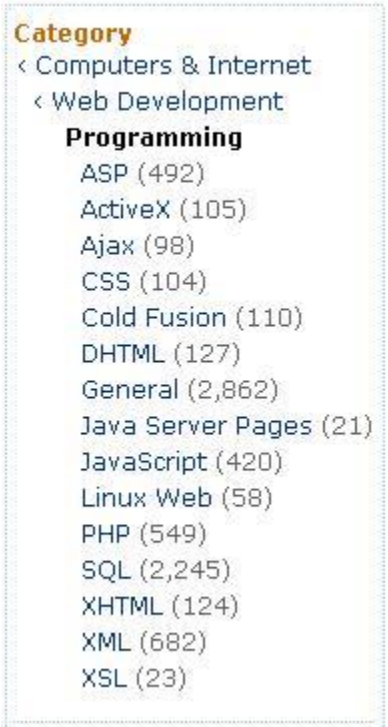
Science
Biology, Psychology, Physics,...

Shopping
Autos, Clothing, Gifts,...

Society
Issues, People, Religion,...

Sports
Basketball, Football, Soccer,...

World
Deutsch, Español, Français, Italiano, Japanese, Korean, Nederlands, Polska, Svenska, ...



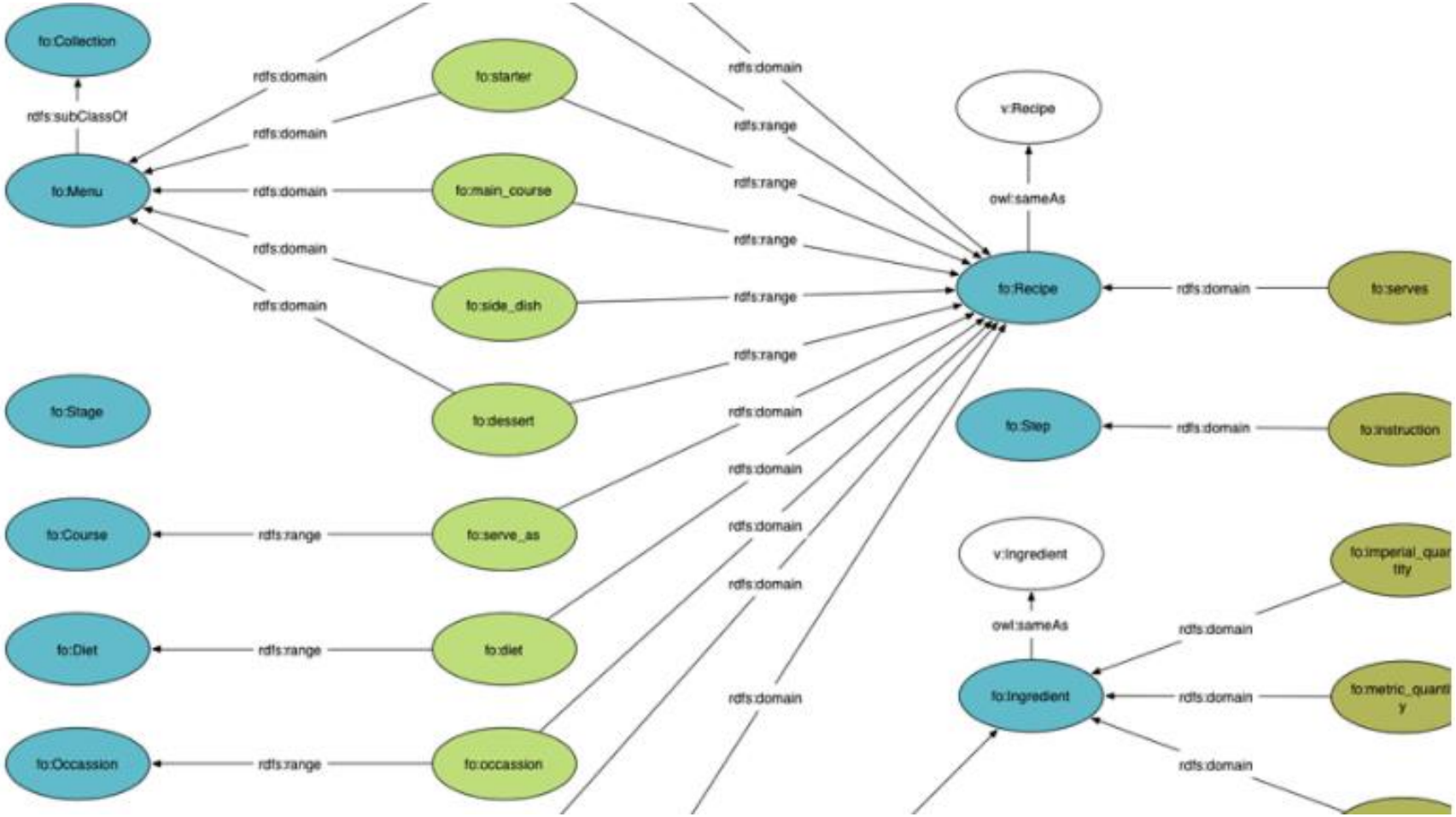
MeSH Tree Structures - 2008

Return to Entry Page

1. Anatomy [A]
2. Organisms [B]
   - Animals [B01] +
   - Algae [B02] +
   - Bacteria [B03] +
   - Viruses [B04] +
   - Fungi [B05] +
   - Plants [B06] +
   - Archaea [B07] +
   - Mesomycetozoea [B08] +
3. Diseases [C]
4. Chemicals and Drugs [D]
5. Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
6. Psychiatry and Psychology [F]
7. Biological Sciences [G]
8. Natural Sciences [H]
9. Anthropology, Education, Sociology and Social Phenomena [I]
10. Technology, Industry, Agriculture [J]
11. Humanities [K]
12. Information Science [L]
13. Named Groups [M]
14. Health Care [N]
15. Publication Characteristics [V]
16. Geographicals [Z]

isa	[associated_with] (continued)
associated_with	[functionally_related_to] (continued)
physically_related_to	performs
part_of	carries_out
consists_of	exhibits
contains	practices
connected_to	occurs_in
interconnects	process_of
branch_of	uses
tributary_of	manifestation_of
ingredient_of	indicates
spatially_related_to	result_of
location_of	temporally_related_to
adjacent_to	co occurs_with
surrounds	precedes
traverses	conceptually_related_to
functionally_related_to	evaluation_of
affects	degree_of
manages	analyzes
treats	assesses_effect_of
disrupts	measurement_of
complicates	measures
interacts_with	diagnoses
prevents	property_of
brings_about	derivative_of
produces	developmental_form_of
causes	method_of
	conceptual_part_of
	issue_in

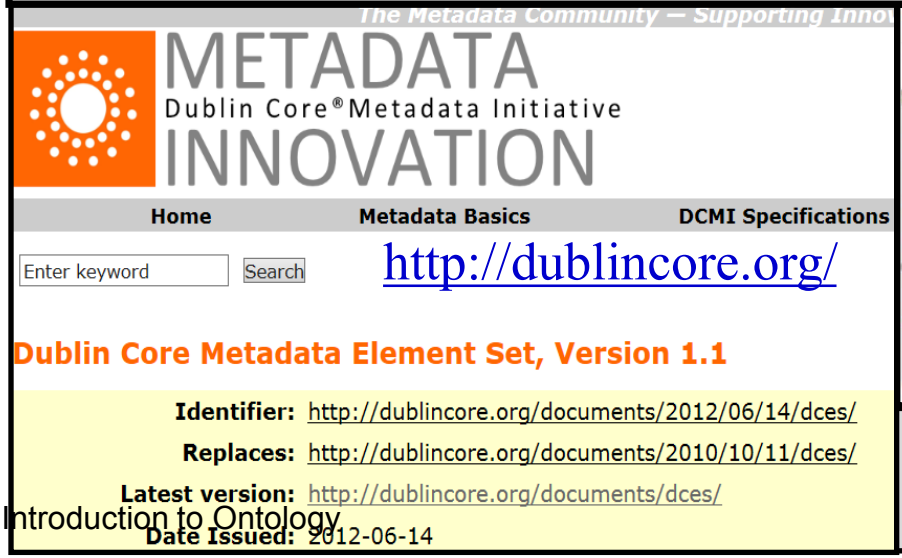
Ontology Example: Food

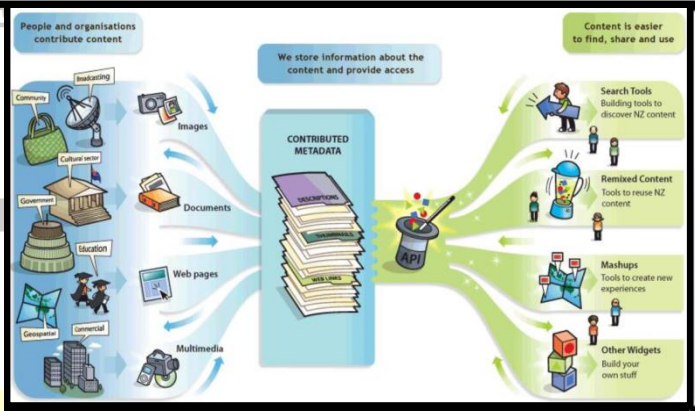


Dublin Core

- A set of ***fifteen basic properties*** for describing generalised Web resources;
- ISO Standard 15836-2003 (February 2003) and ANSI/NISO Z39.85-2012 (February 2013) : http://www.niso.org/apps/group_public/download.php/10256/Z39-85-2012_dublin_core.pdf

The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models.





- Title
- Creator
- Subject
- Description
- Publisher
- Contributor
- Date
- Type
- Format
- Identifier
- Source
- Language
- Relation
- Coverage
- Rights

Namespace:
@prefix dc: http://purl.org/dc/elements/1.1/

Dublin Core (15 basic properties)

	DC Element Name	Definition
1.	Title	A name given to the resource.
2.	Creator	An entity primarily responsible for making the resource.
3.	Subject	The topic of the resource.
4.	Description	An account of the resource.
5.	Publisher	An entity responsible for making the resource available.
6.	Contributor	An entity responsible for making contributions to the resource.
7.	Date	A point or period of time associated with an event in the lifecycle of the resource.
8.	Type	The nature or genre of the resource.
9.	Format	The file format, physical medium, or dimensions of the resource.
10.	Identifier	An unambiguous reference to the resource within a given context.
11.	Source	A related resource from which the described resource is derived.
12.	Language	A language of the resource.
13.	Relation	A related resource.
14.	Coverage	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
15.	Rights	Information about rights held in and over the resource.

Dublin Core Example (RDF/XML)

```
<?xml version="1.0"?>

  <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://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/">
      <dc:title> Guidance on expressing the Dublin Core within the RDF </dc:title>
      <dc:creator> Eric Miller </dc:creator>
      <dc:creator> Paul Miller </dc:creator>
      <dc:creator> Dan Brickley </dc:creator>
      <dc:subject> Dublin Core; RDF; XML </dc:subject>
      <dc:publisher> Dublin Core Metadata Initiative </dc:publisher>
      <dc:contributor> Dublin Core Data Model Working Group </dc:contributor>
      <dc:date> 1999-07-01 </dc:date>
      <dc:format> text/html </dc:format>
      <dc:language> en </dc:language>
    </rdf:Description>

  </rdf:RDF>
```

Property	Description
author	Sub- class-of Person Creator of the recipe.
cookTime	Sub- class-of Duration The time it takes to actually cook the dish in
datePublished	Sub- class-of Date The date the recipe was published
description	Sub- class-of Text A short summary describing the dish.
keywords	Sub- class-of Text Other terms for your recipe such as the season (“summer”), the holiday
nutrition.calories	Sub- class-of Energy The number of calories in each serving.
prepTime	Sub- class-of Duration The length of time it takes to prepare the dish
recipeCategory	The type of meal or course your recipe is about. For example: "dinner", "entree", or "dessert, snack".
recipeCuisine	The region associated with your recipe. For example, "French", Mediterranean", or "American".
recipeIngredient	

Why Develop an Ontology?

- To share **common understanding** of the structure of information
  - among people
  - among software agents
- To enable **reuse** of domain knowledge
  - to avoid “re-inventing the wheel”
  - to introduce standards to allow interoperability

What to Reuse?

- Ontology libraries
  - Protégé ontology library (protege.stanford.edu/ontologies.html)
  - DAML ontology library (www.daml.org/ontologies)
  - Ontolingua ontology library (www.ksl.stanford.edu/software/ontolingua/)
- Upper ontologies
  - IEEE Standard Upper Ontology (suo.ieee.org)
  - Cyc (www.cyc.com)

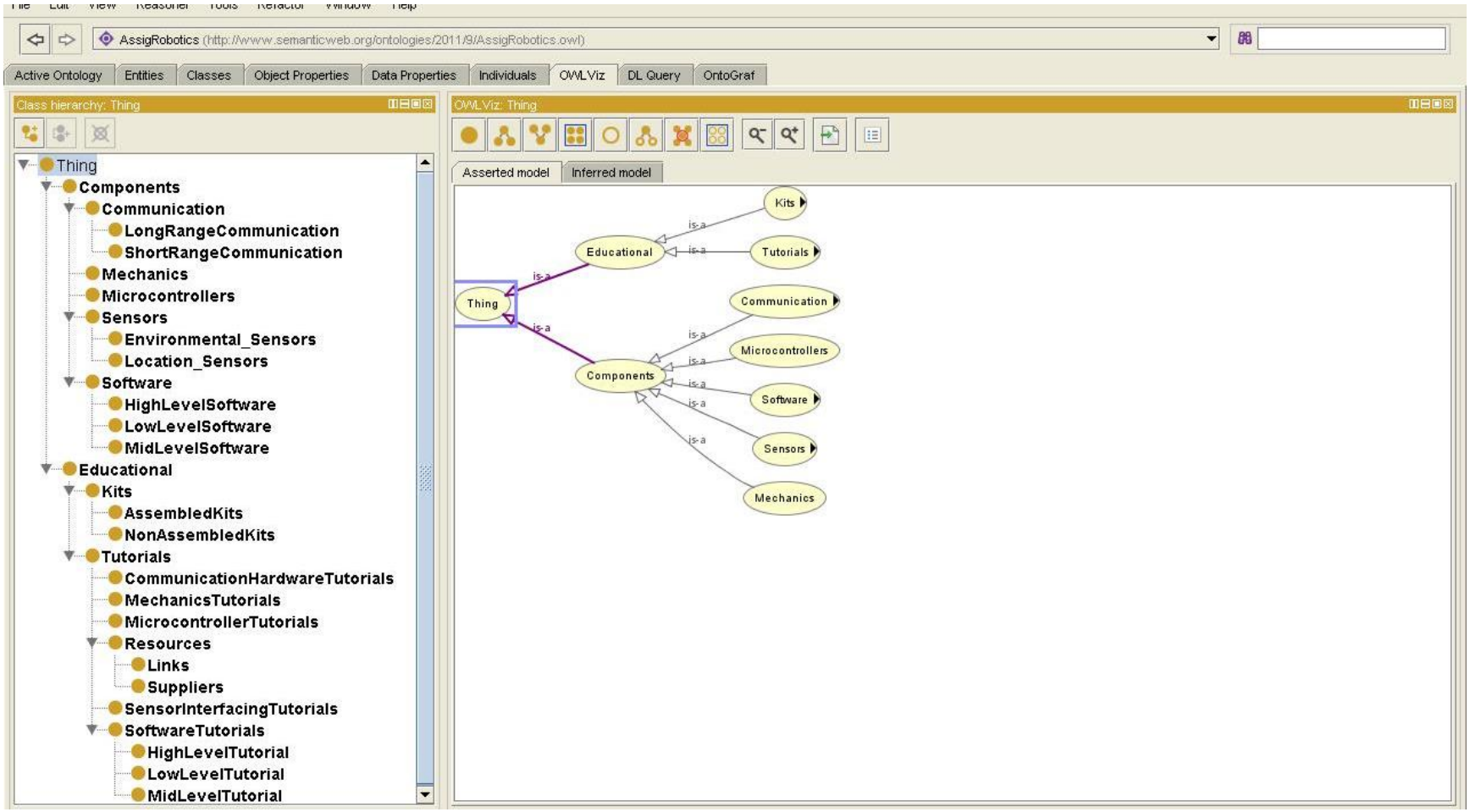
More Reasons

- To make domain assumptions **explicit**
  - easier to change domain assumptions (consider a genetics knowledge base)
- To **separate** domain knowledge from the operational knowledge
  - re-use domain and operational knowledge separately

Ontology components

- Concepts: set of entities within a domain.
- Individuals :**instances** or **objects**
  - concrete examples on concepts in a domain.
- Relations: interactions between concepts or concepts' properties (attributes, slots).
- Axioms: explicit rules to constrain the use of concepts.

Protégé



Ontology Languages

Ontology Languages for the Web

- Semantic Web effort led to development of "resource description" language(s)
  - E.g., **RDF**, and later RDF Schema (**RDFS**)
- RDFS is recognisable as an ontology language
  - **Classes** and **properties**
  - **Sub/super-classes** (and properties)
  - **Range** and **domain** (of properties)
- Differences between ontology languages
  - Expressiveness
  - Computational complexity of reasoning

Ontology Languages for the Web(cont.)

- **RDF:**
  - is a W3C standard, which provides tool to describe Web resources
  - provides interoperability between applications that exchange machine-understandable information

- **RDF Schema:**
  - is a W3C standard which defines vocabulary for RDF
  - organizes this vocabulary in a typed hierarchy
  - capable to explicitly declare semantic relations between vocabulary terms

RDF Statement

- ***Subject*** of an RDF statement is a resource
- ***Predicate*** of an RDF statement is a property of a resource
- ***Object*** of an RDF statement is the value of a property of a resource

Making RDF Statements is like filling a questionnaire form

http://www.bla-bla.fi/#VaganTerziyan

Subject

Predicate

Object

NAME: Vagan Terziyan

Address: _____

Telephone Number: (home) _____

Telephone Number: (cell) _____

Age: _____ Date of Birth: _____ Social Security No: _____

EMPLOYER: _____

Address: _____

Telephone Number: (work)_____

Occupation: _____ Worked there how long?_____

Immediate Supervisor: _____

SPOUSE'S NAME: _____

Address: _____

Telephone Number: (home) _____

Making RDF Statements is like filling a questionnaire form

http://www.bla-bla.fi/#VaganTerziyan

Subject

Predicate

Object

NAME: _____

Address: _____

Telephone Number: (home) _____

Telephone Number: (cell) _____

Age: 33 Date of Birth: _____ Social Security No: _____

EMPLOYER: _____

Address: _____

Telephone Number: (work)_____

Occupation: _____ Worked there how long?_____

Immediate Supervisor: _____

SPOUSE'S NAME: _____

Address: _____

Telephone Number: (home) _____

Making RDF Statements is like filling a questionnaire form

http://www.bla-bla.fi/#VaganTerziyan

Subject

Predicate

Object

NAME: _____

Address: _____

Telephone Number: (home) _____

Telephone Number: (cell) _____

Age: _____ Date of Birth: _____ Social Security No: _____

EMPLOYER: _____ http://dbpedia.org/page/University_of_Jyvaskyla

Address: _____

Telephone Number: (work) _____

Occupation: _____ Worked there how long? _____

Immediate Supervisor: _____

SPOUSE'S NAME: _____

Address: _____

Telephone Number: (home) _____

Making RDF Statements is like filling a questionnaire form

http://www.bla-bla.fi/#VaganTerziyan

Subject

Predicate

Object

NAME: _____

Address: _____

Telephone Number: (home) _____

Telephone Number: (cell) _____

Age: _____ Date of Birth: _____ Social Security No: _____

EMPLOYER: _____

Address: _____

Telephone Number: (work)_____

Occupation: _____ Worked there how long?_____

Immediate Supervisor: _____ http://www.bla-bla.fi/#PekkaNeittaanmaki

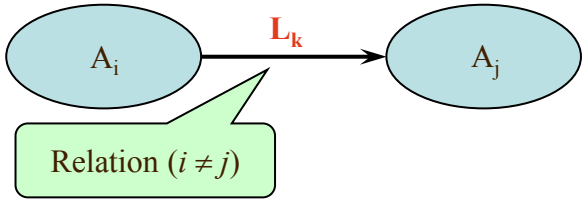
SPOUSE'S NAME: _____

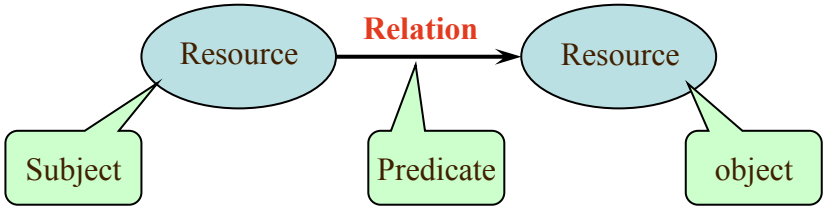
Address: _____

Telephone Number: (home) _____

Semantic Relation as RDF statement

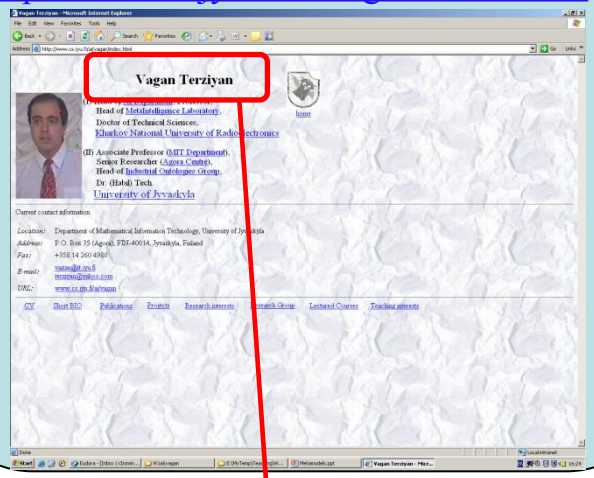
(so called "object property")





Personal web page of Professor X

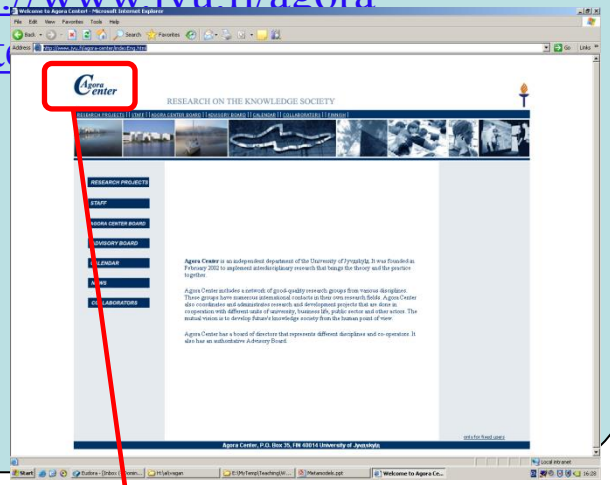
http://www.cs.jyu.fi/ai/vagan/index.html



refers_to

Web page of Agora Center

http://www.jyu.fi/agora-cente



URI of Professor X.

http://www.cs.jyu.fi/ai/vagan/#vagan

employed_by

URI of Agora Center

http://www.jyu.fi/agora-center/#AC

Dereferenceable URI ("Hash vs. Slash")

Previous Example in RDF

- RDF Schema terms:
  - Class
  - Property (range-domain)
  - type
  - subClassOf

- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

```
<Person,type,Class>
<hasColleague,type,Property>
<Professor,subClassOf,Person>
<Abeer,type,Professor>
<hasColleague,range,Person>  (What can have this property )
<hasColleague,domain,Person>(What can be the value of property)
```

From RDFS to OWL

- Two other languages
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed by group of (largely) US researchers
- Efforts merged to produce **DAML+OIL**
  - Development carried out by "Joint EU/US Committee on Agent Markup Languages"
- DAML+OIL submitted to as basis for standardisation
  - Web-Ontology (**WebOnt**) Working Group formed
  - WebOnt developed **OWL** language based on DAML+OIL
  - OWL now a W3C **recommendation** (i.e., a standard)
- OIL, DAML+OIL and OWL based on **Description Logics**
  - OWL is effectively a "Web-friendly" syntax