Three Ws of ontology

ASIYAH YU LIN

DEC. 15, 2015

Agenda

1. What is ontology, or ontologies?
2. Why do we need ontologies?
3. How do we build an ontology?
4. Importance of community driven consensus
5. Tools and learning resources.

What is ontology?

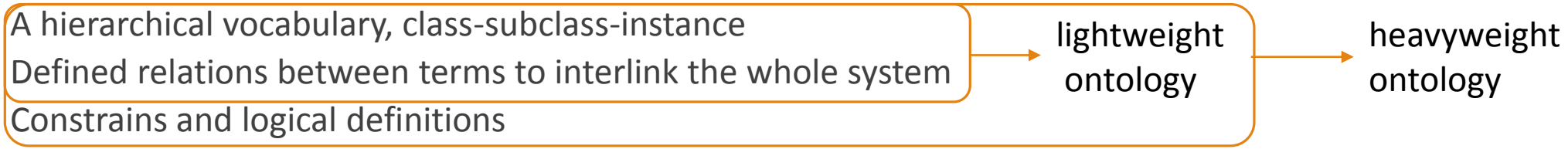
Originally from Philosophy, a systematic explanation of being: what is the essence of things and what is the existence of things.

Ontologies are widely used in Knowledge Engineering, Artificial Intelligence and Computer Science.

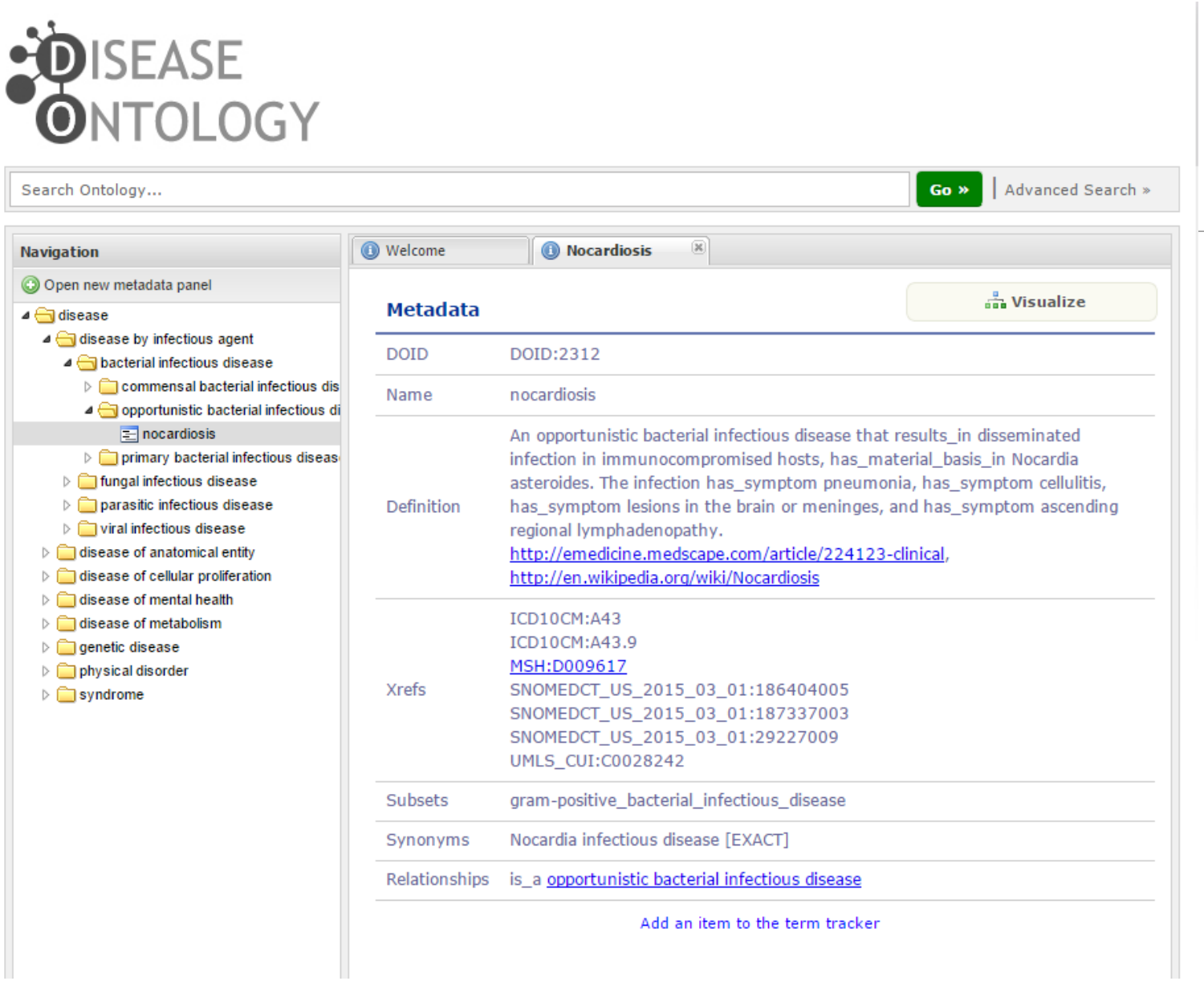
Ontology is a form of a knowledge base representation, which supports expert systems to solve different problems.

Ontologies are the structural frameworks for organizing terms hierarchically and defining relations between terms within a domain

- A hierarchical vocabulary, class-subclass-instance
- Defined relations between terms to interlink the whole system
- Constrains and logical definitions
- Explicit specification of a conceptualization (Tom Gruber,1993)

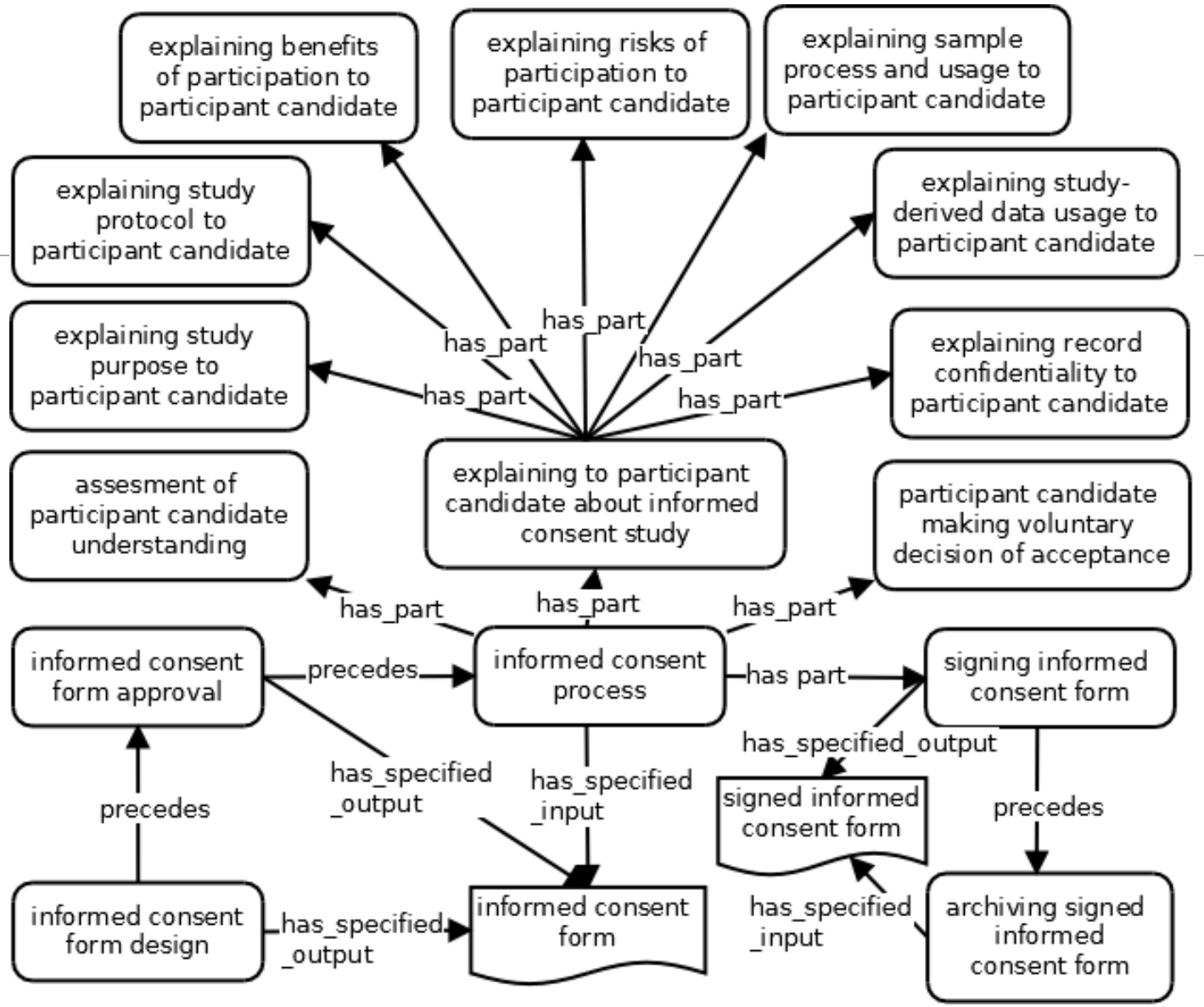


Example:



http://disease-ontology.org/

Example:



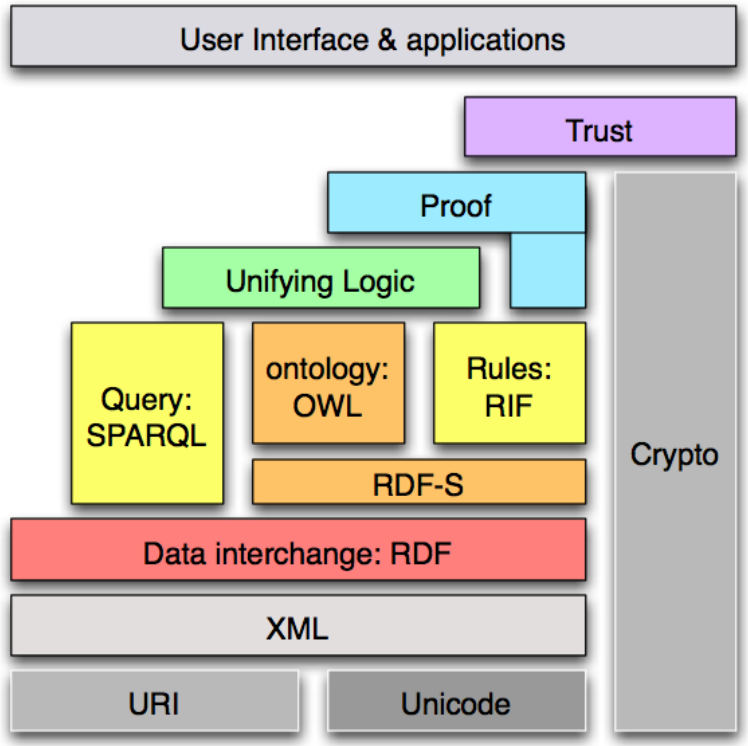
Lin Y, Harris MR, Manion FJ, Eisenhauer E, Zhao B, Shi W, Karnovsky A, He Y: Development of a BFO-based Informed Consent Ontology (ICO). The 5th International Conference on Biomedical Ontologies (ICBO): 2014; Houston, Texas, USA, October 8-9, 2014. CEUR Workshop Proceedings; 2013: Page 84-86. (http://ceur-ws.org/Vol-1327/icbo2014_paper_54.pdf)

Semantic Web layer cake

RDF: simple triples, graph-based queries, supports very large amount of data.

RDFS: allows to express the relationships between things, such as rdf:type or rdfs:subClassOf, which can be used to say things..

OWL: significantly more expressive language, strong axioms, inference capabilities, consistency verification, but can be rather slow



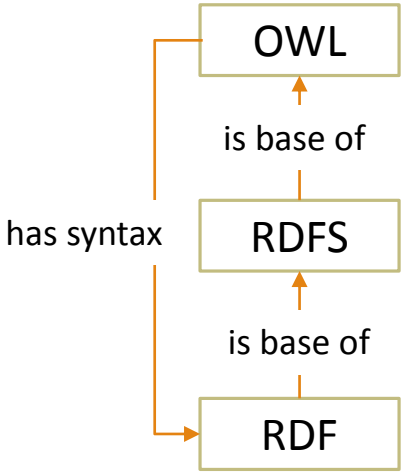
RDF, RDFS, and OWL

Resource Description Framework (**RDF**) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. (https://en.wikipedia.org/wiki/Resource_Description_Framework)

RDF Schema (**Resource Description Framework Schema**, variously abbreviated as RDFS, RDF(S), RDF-S, or RDF/S) is a set of classes with certain properties using the RDF extensible knowledge representation data model, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources. (https://en.wikipedia.org/wiki/RDF_Schema)

The **Web Ontology Language** (**OWL**) is a family of knowledge representation languages for authoring ontologies. (https://en.wikipedia.org/wiki/Web_Ontology_Language)

RDF, RDFS, and OWL



cat_dog_example:

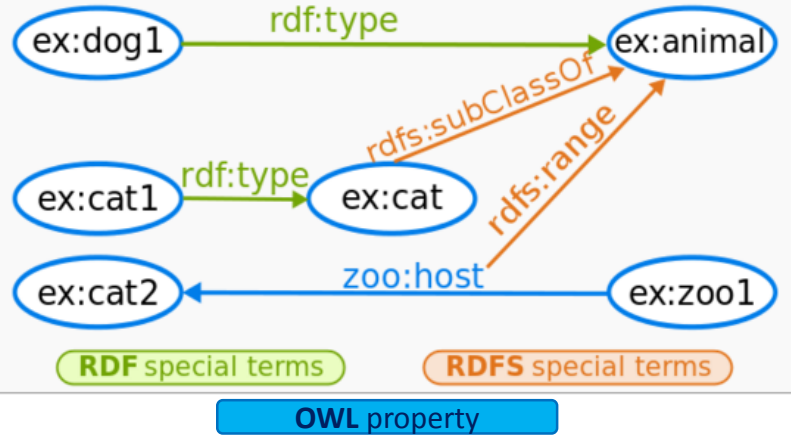
In English	The graph
• Dog1 is an animal • Cat1 is a cat • Cats are animals • Zoos host animals • Zoo1 hosts the Cat2	

Figure of table copied from: https://en.wikipedia.org/wiki/RDF_Schema. Added blue box to declare the OWL poperty.

OWL in turtle syntax

RDF/turtle

cat_dog_example:

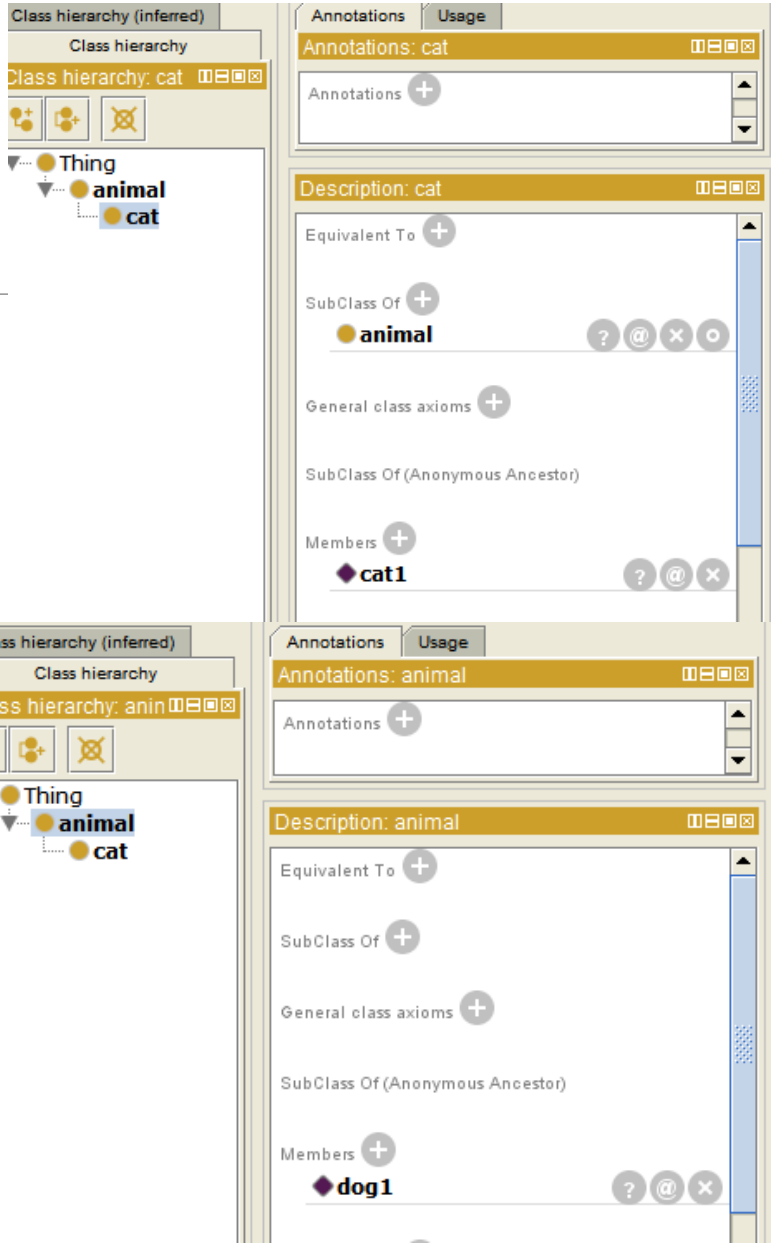
```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/> .
@prefix zoo:   <http://example.org/zoo/> .
ex:dog1    rdf:type      ex:animal .
ex:cat1    rdf:type      ex:cat .
ex:cat     rdfs:subClassOf  ex:animal .
zoo:host   rdfs:range       ex:animal .
ex:zoo1    zoo:host      ex:cat2 .
```

https://en.wikipedia.org/wiki/RDF_Schema.

OWL in XML/RDF syntax

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

<!DOCTYPE rdf:RDF [
    <!ENTITY ex "http://example.org/" >
    <!ENTITY zoo "http://example.org/zoo/" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.semanticweb.org/owl/owlapi/turtle#"
     xml:base="http://www.semanticweb.org/owl/owlapi/turtle"
     xmlns:ex="http://example.org/"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:zoo="&ex;zoo/">



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->


    <!-- http://example.org/zoo/host -->

    <owl:ObjectProperty rdf:about="&ex;zoo/host">
        <rdfs:range rdf:resource="&ex;animal"/>
    </owl:ObjectProperty>
```

```
    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Classes
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://example.org/animal -->
    <owl:Class rdf:about="&ex;animal"/>

    <!-- http://example.org/cat -->
    <owl:Class rdf:about="&ex;cat">
        <rdfs:subClassOf rdf:resource="&ex;animal"/>
    </owl:Class>

    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Individuals
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://example.org/cat1 -->
    <owl:NamedIndividual rdf:about="&ex;cat1">
        <rdf:type rdf:resource="&ex;cat"/>
    </owl:NamedIndividual>

    <!-- http://example.org/cat2 -->
    <owl:NamedIndividual rdf:about="&ex;cat2"/>

    <!-- http://example.org/dog1 -->
    <owl:NamedIndividual rdf:about="&ex;dog1">
        <rdf:type rdf:resource="&ex;animal"/>
    </owl:NamedIndividual>

    <!-- http://example.org/zoo1 -->
    <owl:NamedIndividual rdf:about="&ex;zoo1">
        <zoo:host rdf:resource="&ex;cat2"/>
    </owl:NamedIndividual>
</rdf:RDF>
```



Why do we need an ontology?

Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge. Ontology provides potential terms for describing our knowledge about the domain.

- provide a robust and coherent organization of knowledge
- integrate prior knowledge
- help manage complexity of the domain knowledge
- discover new knowledge.

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

B Chandrasekaran *et.al*. 1999 What Are Ontologies, and Why Do We Need Them?
NF Noy *et.al.* Ontology Development 101: A Guide to Creating Your First Ontology

How to build an ontology?

Step 1. Determine the domain and scope of the ontology

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Ask **competency questions**.

NF Noy *et.al.* Ontology Development 101: A Guide to Creating Your First Ontology

How to build an ontology?

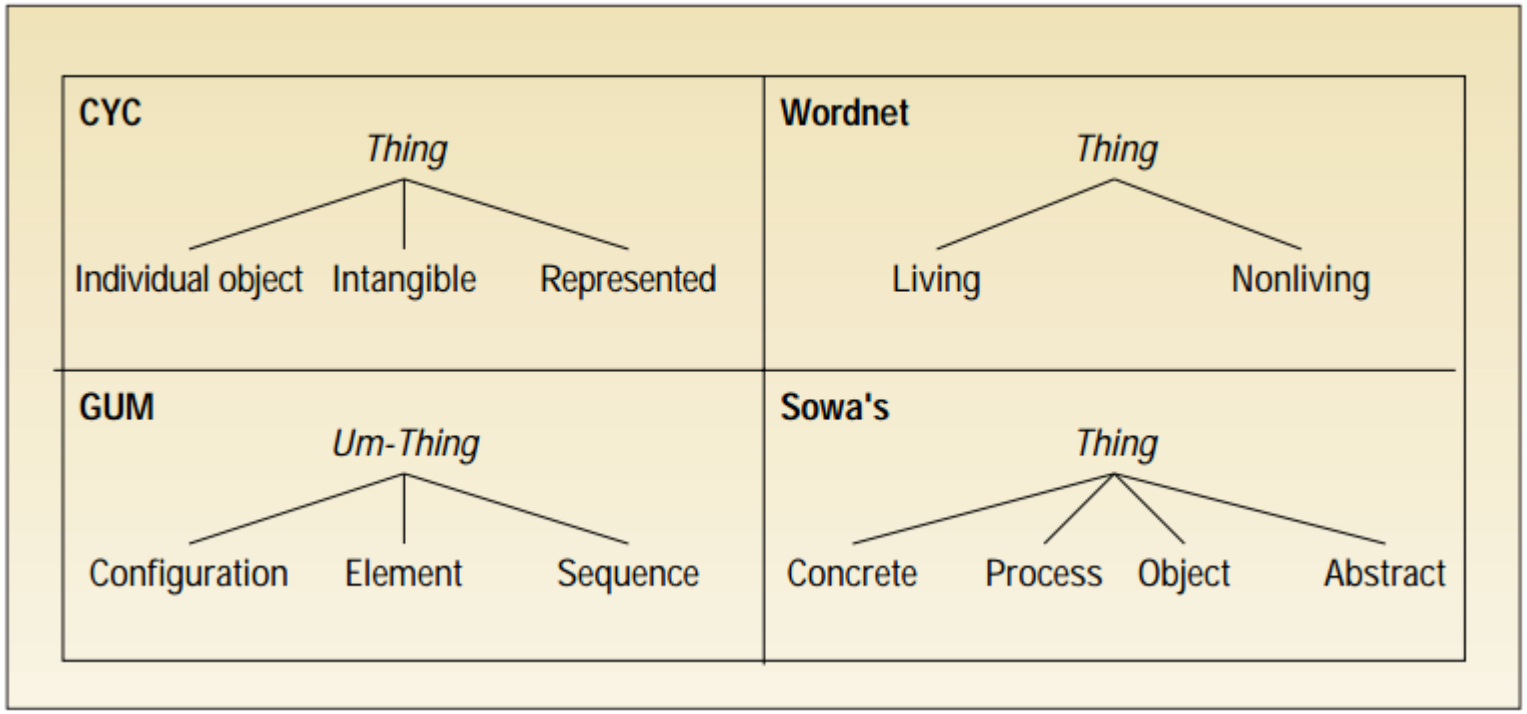
Step 2. Consider reusing existing ontologies

- BioPortal (http://bioportal.bioontology.org/)
- OBO fundry (http://www.obofoundry.org/)
- Ontobee (http://www.ontobee.org/)
- http://schema.org/
- http://dbpedia.org/ontology/
- Swoogle (http://swoogle.umbc.edu/)
-….

NF Noy *et.al.* Ontology Development 101: A Guide to Creating Your First Ontology

How to build an ontology?

Step 3. Consider reusing top level ontologies

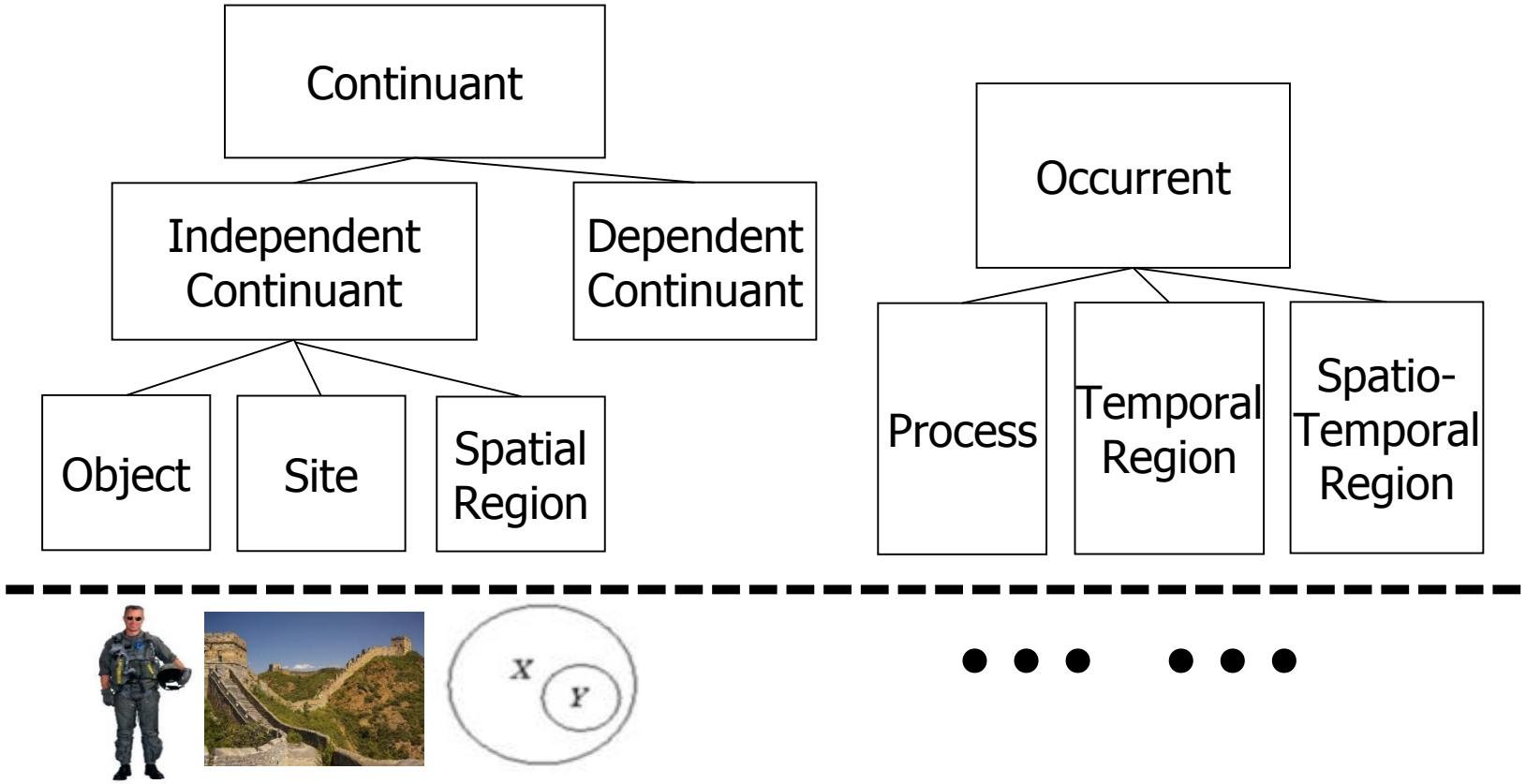


B Chandrasekaran *et.al*. 1999 What Are Ontologies, and Why Do We Need Them?

How to build an ontology?

Step 3. **Consider reusing top level ontologies**

BFO (Basic Formal Ontology)



Reference: Barry Smith's talk

How to build an ontology?

Step 4. Enumerate important terms in the ontology

- list out the key terms and linkage of the terms.
- brain storming
- concept map

NF Noy *et.al.* Ontology Development 101: A Guide to Creating Your First Ontology

How to build an ontology?

Step 4. Enumerate important terms in the ontology

- list out the key terms and relations of the terms.
- brain storming
- concept map

How to build an ontology?

Step 5. Building conceptual logo: define class hierarchy, instances, and relations

- define the class and class hierarchies
- determine the term is a class of an instance
- define the relations that linking classes, and/or instances

Logic as the clips for “Conceptual Lego”

hand

extremity

body

chronic

acute

abnormal

normal

protein

polysacharide

cell

expression

gene

Lung

infection

inflammation

mucus

bacterium

ischaemic

deletion

polymorphism

virus

Alan Rector: http://www.cs.man.ac.uk/~rector/tutorials/Medinfo-2007/Presentations/

Logic as the clips for “Conceptual Lego”

“***SNPolymorphism*** of ***CFTR Gene*** causing ***Defect in MembraneTransport*** of ***Chloride Ion*** causing ***Increase*** in ***Viscosity*** of ***Mucus*** in ***Cystic Fibrosis***…”

“***Hand*** which is ***anatomically normal***”

Alan Rector: http://www.cs.man.ac.uk/~rector/tutorials/Medinfo-2007/Presentations/

How to build an ontology?

Step 6. Evaluation, maintenance and evolution.

- check consistency, using reasoning tools.
- evaluated the ontology using use case scenarios, and SPARQL queries.
- maintenance, follow the best practice.
- make it reusable for others
- ontology evolves and extends

How to build an ontology?

Bottom-up strategy:

- List all the terms and then build up the hierarchies and relations

Top-down strategy:

- Choose the top-level ontology and build the subclasses of the top level terms

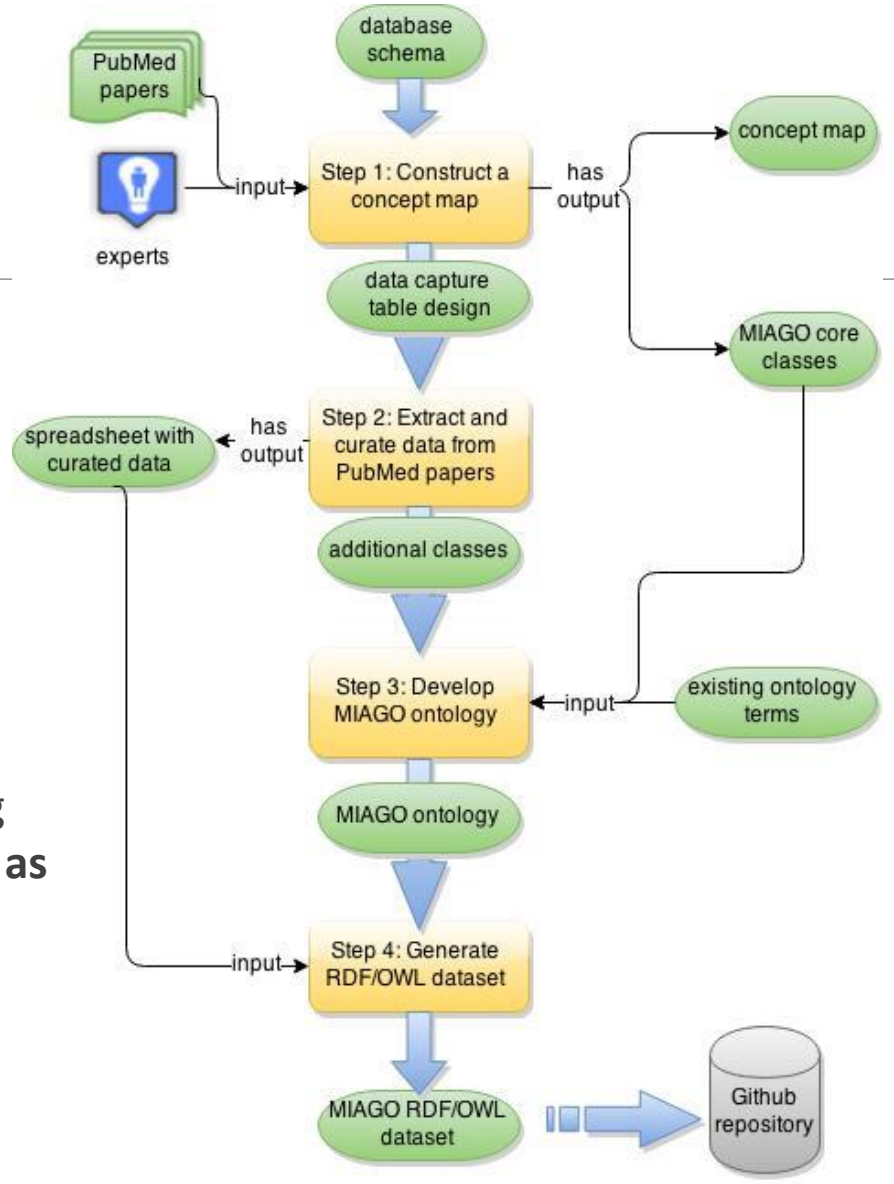
Hybrid middle out strategy:

- Most common
- Start up by both choosing top-level ontology, domain ontologies
- Enumerate the bottom level terms and extend the domain ontologies by adding subclasses.

Example:

Database schema -> concept map -> Develop ontology -> transform curated data from spreadsheet into RDF/OWL dataset

Note: The spreadsheet is designed according to the ontology, **header as the class and cell as individual data**. Sometimes the cell can be class as well.



Example: Informed Consent Ontology

- hybrid middle out strategy

- identify scope and driven application

- using BFO as top ontology, importing OBI (Ontology for Biomedical Investigation) and IAO(Information Artifact Ontology)-> initial development

- identify terms from informed consent template, mapping terms with other terminologies, such as NCI thesaurus.

- determine class hierarchy, term definitions

- analyze the informed consent work flow using concept maps: pre-informed consent processes, obtaining informed consent processes, and post-informed consent processes.

- adding terms, face to face meeting for term defections, mediawiki for term management.

- ongoing work…

Community Consensus!

- Ontology is representation of the domain knowledge.
- Your ontology has to reflect the community consensus in a domain.
- The key for ontology reuse and open resource.
- In the case of developing a vocabulary for a closed system, you may not need to have all domain experts' consensus, but at least, the team has to agree with your ontology representation.

Tools and resources

Tools:

- Protégé OWL editor
- OWL API
- OntoFox, Ontodog, Ontorat,… onto-animal serials tools
- TopBraid (Only work with RDF/XML format)
- Other tools

Learning Resources:

- Ontology Development 101: A Guide to Creating Your First Ontology
- OWL @ Manchester
- Ontolog Forum
- Barry Smith's site
- John Sowa's site
- Medical Ontology Research (Olivier Bodenreider)
- More …