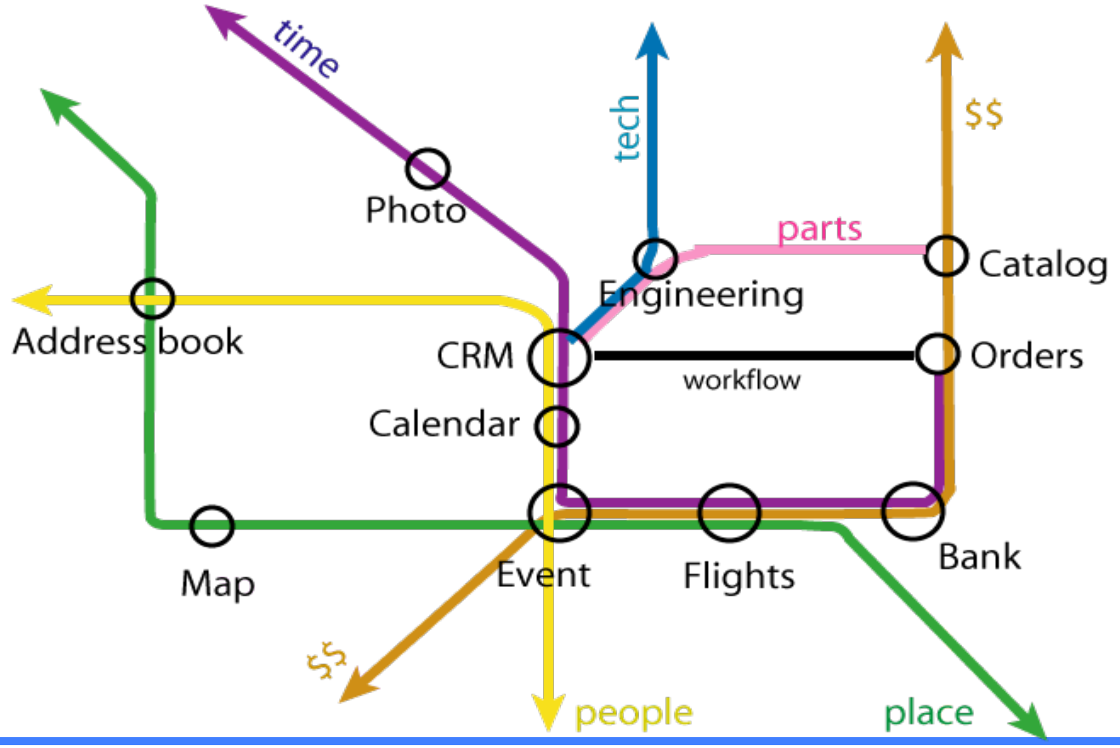
University of Dublin
Trinity College

# Introduction to Ontology

Owen.Conlan@scss.tcd.ie

# "Interconnected web of data"
## the need for a knowledge driven approach

Increasingly boundary between data in enterprise systems, personal devices and on web becoming blurred



# Metadata will solve all problems?

## Classes of Metadata

- Content independent – e.g. modification date of document
- Content dependent – e.g. size of the document
- Direct content-based – e.g. inverted tree index of a document
- Content Descriptive – e.g. textual annotations descibing an image
- Domain independent – e.g HTML document type definitions
- Domain specific – e.g. based on some domain like “University”

▪Relies on common understanding of syntax/terms

# EXIF Sub IFD

| Entry | Tag | Value | Meaning | Format | Components |
|---|---|---|---|---|---|
| Exposure Time | 829a | 1/350 | 0.00 | unsigned rational | 1 |
| F Number | 829d | 40/10 | 4.00 | unsigned rational | 1 |
| EXIF Version | 9000 | 30 32 31 30 | | undefined | 4 |
| Date Time Original | 9003 | 2001:06:09 15:17:32 | | ascii string | 20 |
| Date Time Digitised | 9004 | 2001:06:09 15:17:32 | | ascii string | 20 |
| Component Configuration | 9101 | 01 02 03 00 | | undefined | 4 |
| Compressed Bits/Pixel | 9102 | 3/1 | 3.00 | unsigned rational | 1 |
| Shutter Speed Value | 9201 | 553859/65536 | 8.45 | signed rational | 1 |
| Aperture Value | 9202 | 262144/65536 | 4.00 | unsigned rational | 1 |
| Exposure Bias Value | 9204 | 0/3 | 0.00 | signed rational | 1 |
| Max Aperture Value | 9205 | 194698/65536 | 2.97 | unsigned rational | 1 |
| Subject Distance | 9206 | 3750/1000 | 3.75 | unsigned rational | 1 |
| Metering Mode | 9207 | 2 | centre weighted average | unsigned short | 1 |
| Flash | 9209 | 0 | not used | unsigned short | 1 |
| Focal Length | 920a | 346/32 | 10.81 | unsigned rational | 1 |

# XML will solve all problems?

- XML is Syntax too
  - DTDs talk about element nesting
  - XML Schema schemas give you data types
  - need anything else? => *write comments!*
- Domain Semantics is complex:
  - implicit assumptions, hidden semantics
  - ⇒ sources seem unrelated to the non-expert
- Need Structure and Semantics beyond XML trees!
  - ⇒ employ richer OO models
  - ⇒ make domain semantics and "glue knowledge" ***explicit***
  - ⇒ use **semantic models** to fix terminology and conceptualization
  - ⇒ avoid ambiguities by using formal semantics
- XML DTDs and XML Schemas are sufficient for exchanging data between parties who have agreed to meaning of terms beforehand
- Semantic Models are critical for exchanging data between applications from diverse parties in order to ensure terms used are semantically equivalent

# Sharing Semantics

## Metadata

- Data describing the content and meaning of resources and services.
- But everyone must speak the same language…

## Terminologies

- Shared and common vocabularies
- For search engines, agents, curators, authors and users
- But everyone must mean the same thing…

## Semantic Models

- Shared and common understanding of a domain
- Essential for search, exchange and discovery

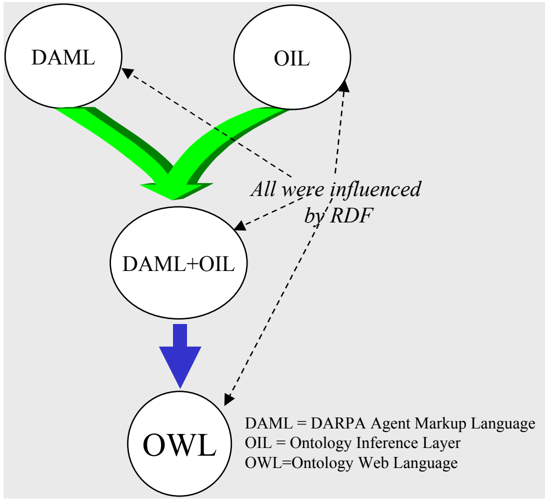
➔ Ontologies/Topic Maps aim at sharing meaning

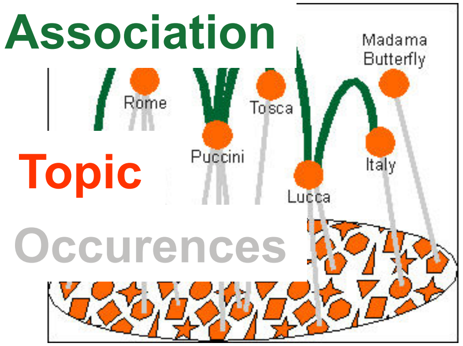


[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]

# Encoding Semantic Models

- XML based encoding methods becoming popular
  - Machine and human readable
  - Easily processed
- From the "Semantic Web" community
  - W3C standards (http://www.w3.org/2001/sw/)
    - RDF
    - RDFS
    - OWL
  - DERI- Digital Enterprise Research Institute
    - WSML
    - WSMO
- From ISO
  - Topic Maps
  - Conceptual Graphs
- Rules
  - SWRL
  - Beyond RuleML

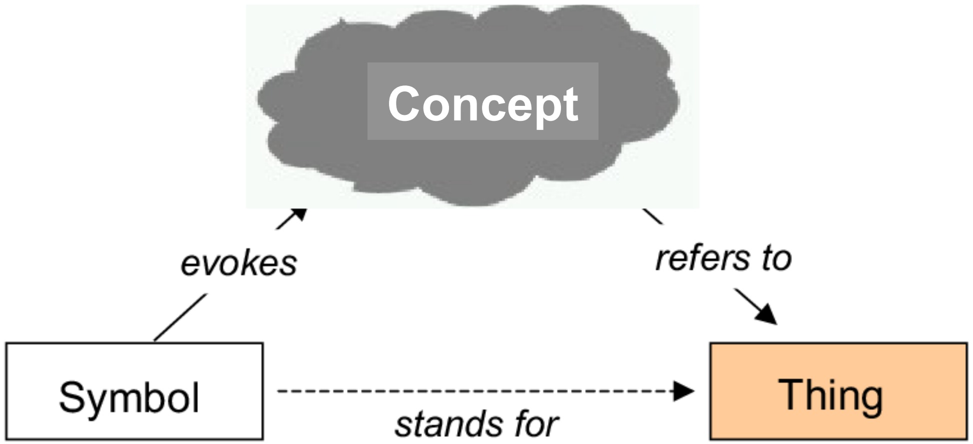


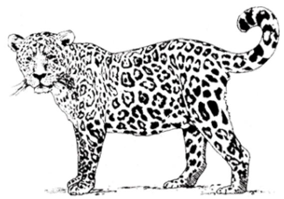



# The Meaning Triangle

- Humans require words (or at least symbols) to communicate efficiently. The mapping of words to things is indirect. We do it by creating *concepts* that refer to things.
- The relation between symbols and things has been described in the form of the *meaning triangle*:

*“Jaguar“*



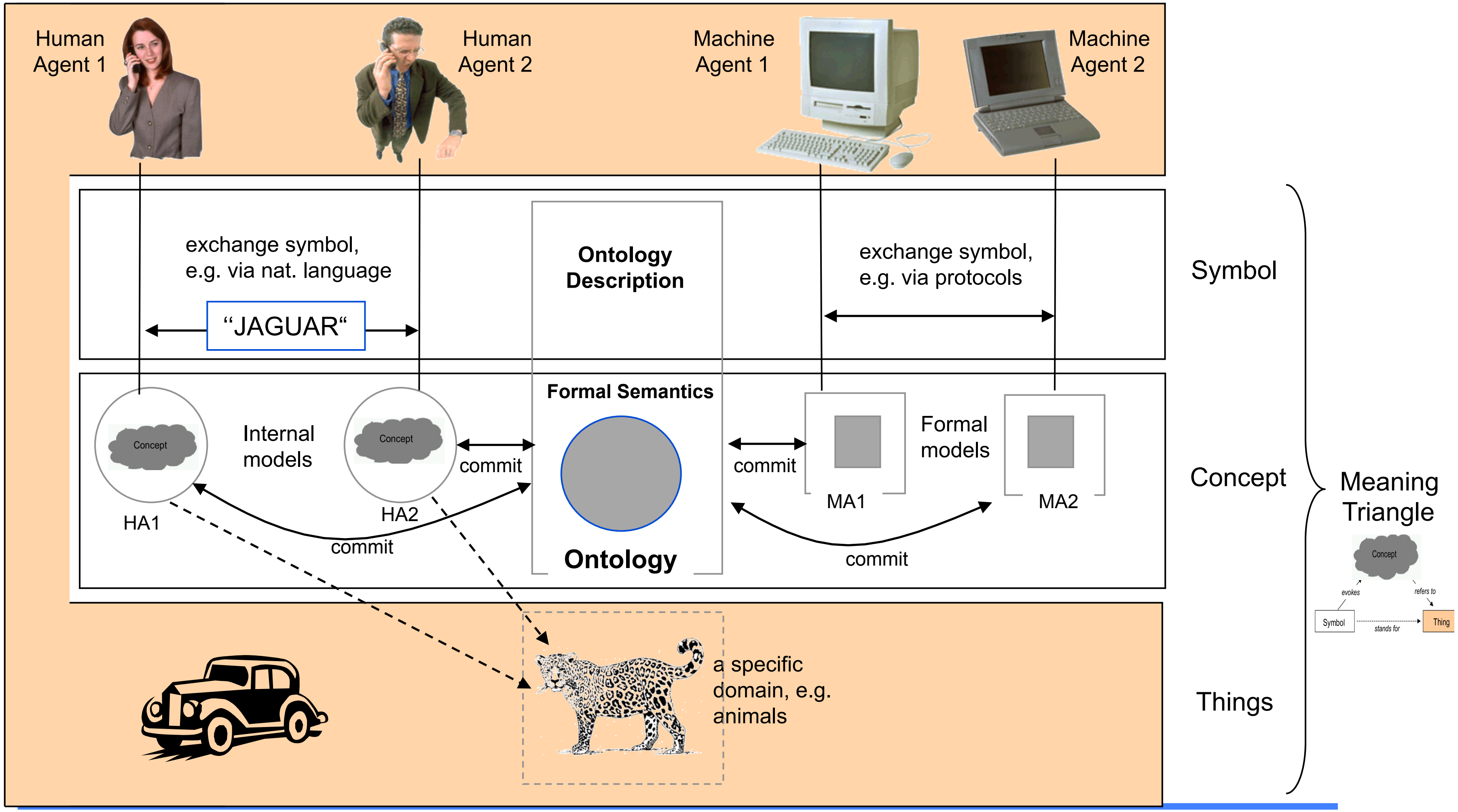
*Ogden, C. K. & Richards, I. A. 1923. "The Meaning of Meaning." 8th Ed. New York, Harcourt, Brace & World, Inc*

before: Frege, Peirce; see [Sowa 2000]



[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]

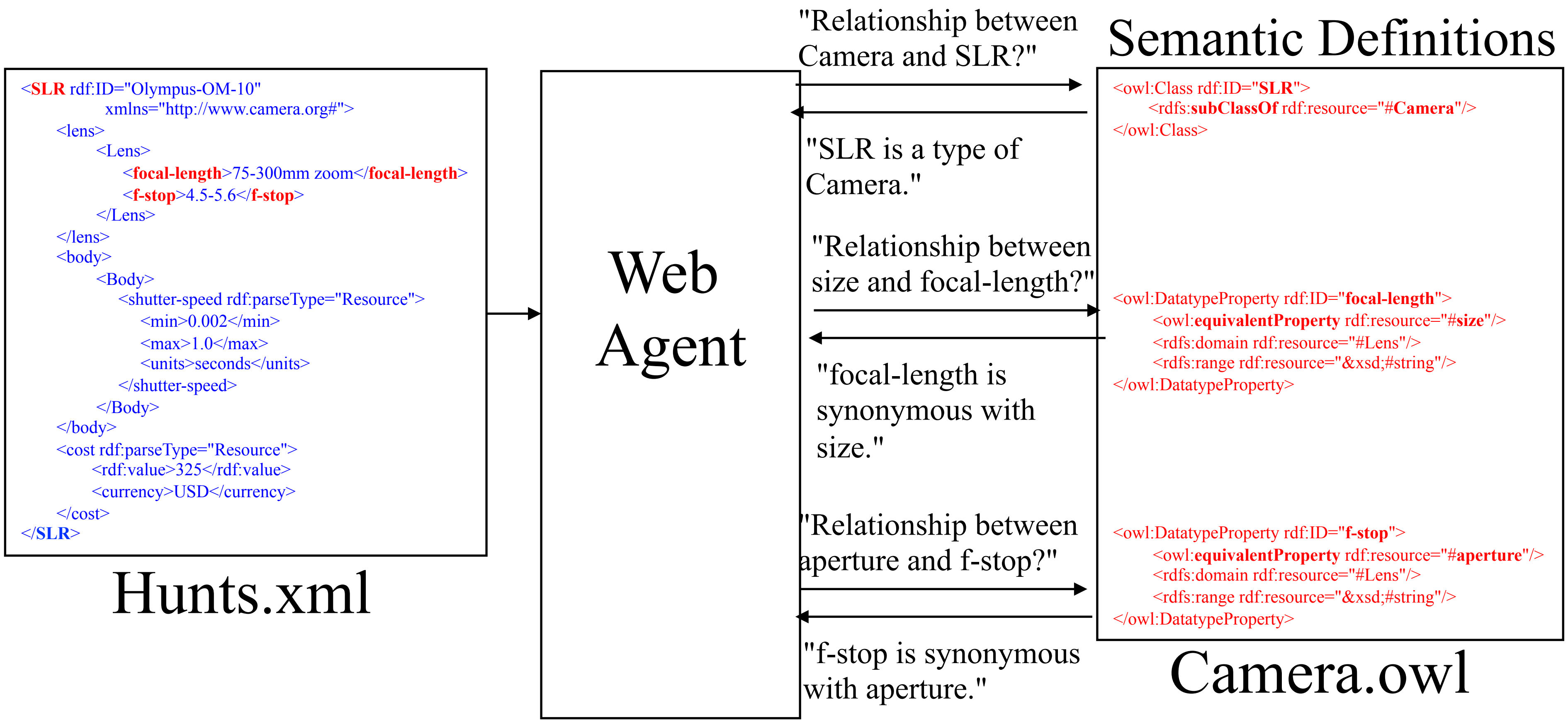
# Human and machine communication






[Maedche et al., 2002]

# Encoding Semantics (Example)






[DARPA 2003]

# Ontology: A definition

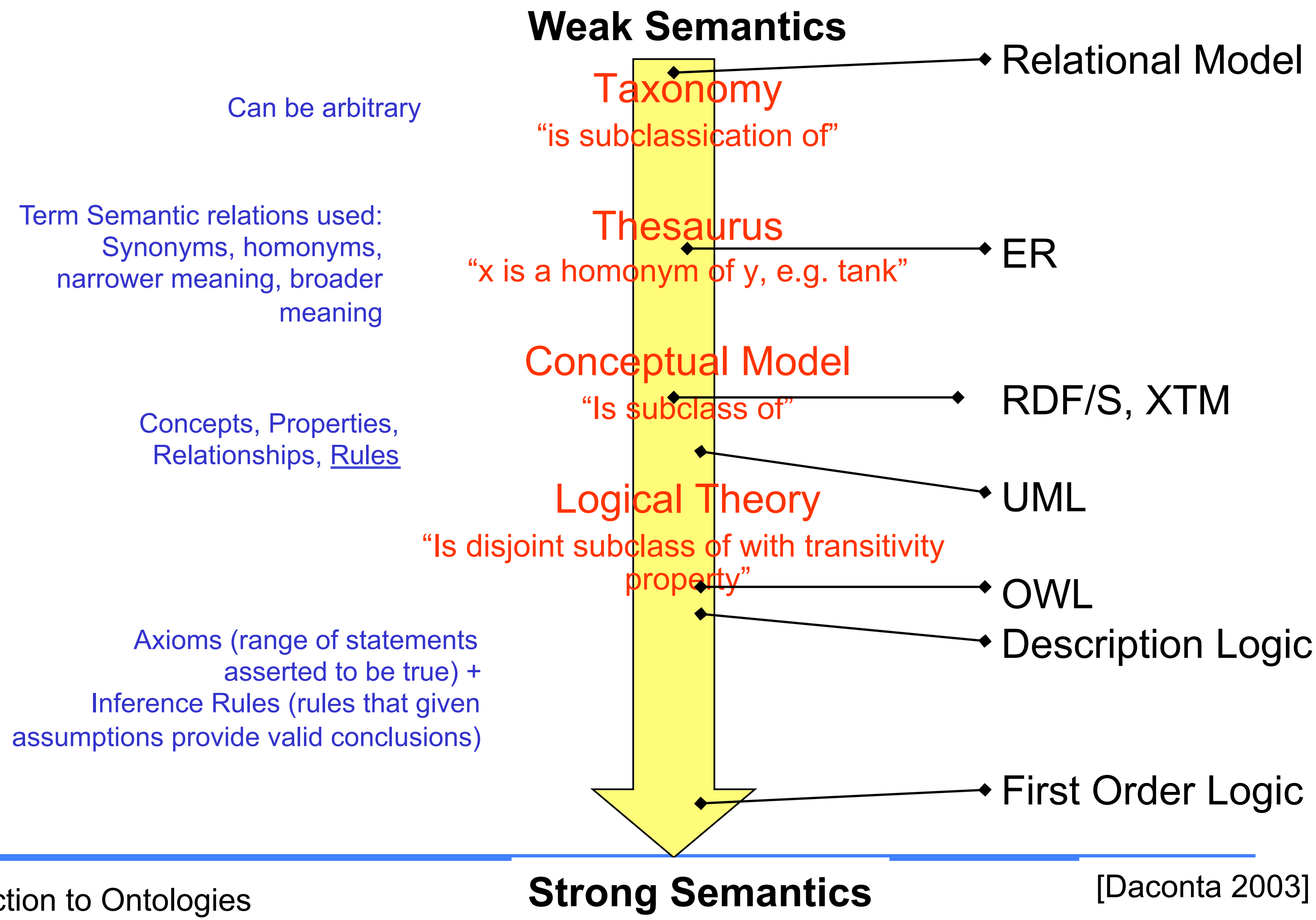
An "ontology" describes the common words, concepts and relationships between concepts used to describe and represent an area of knowledge.

An ontology can range from a

- Taxonomy (knowledge with minimal hierarchy or a parent/child structure) to a …
- Thesaurus (words and synonyms) to a ….
- Conceptual Model (with more complex knowledge) to a…
- Logical Theory (with very rich, complex, consistent and meaningful knowledge).

A well-formed ontology is one that is expressed in a well-defined syntax that has a well-defined machine interpretation consistent with the above ontology definition

# Ontology Spectrum






[Daconta 2003]

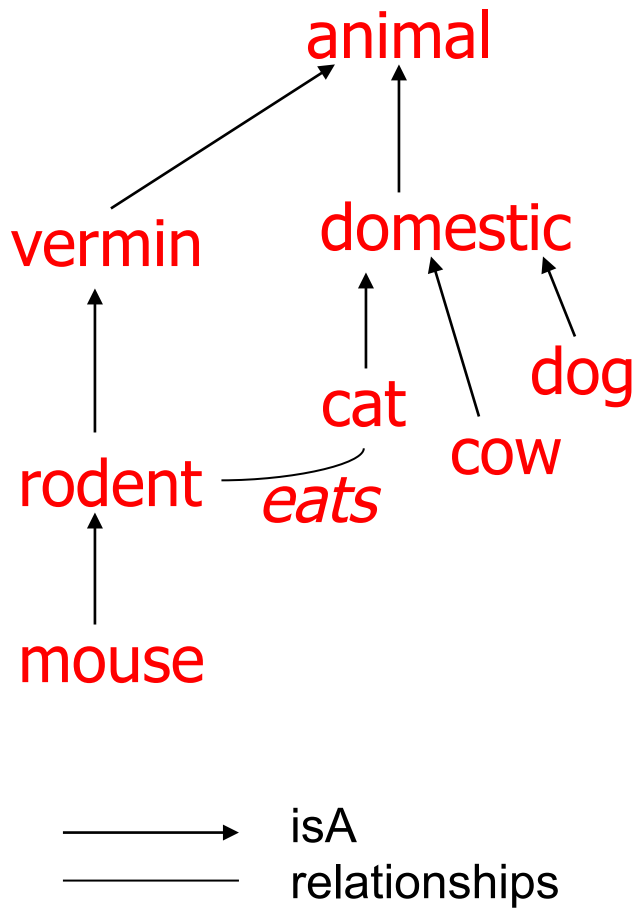
# Ontology Modeling
# An explicit description of a domain

*Concepts* (**class**, set, type, predicate)

- event, gene, gammaBurst, atrium, molecule, cat

*Properties* of concepts and *relationships* between them (**slot**)

- *Taxonomy*: generalisation ordering among concepts *isA*, *partOf*, *subProcess*
- *Relationship*, *Role* or *Attribute*: *functionOf*, *hasActivity* location, *eats*, size






[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]

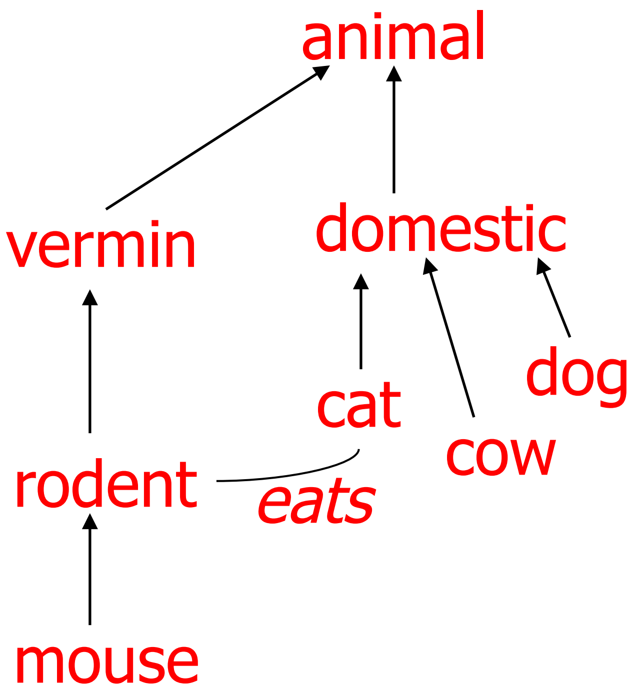
# An explicit description of a domain

*Constraints* or *axioms* on properties and concepts:

- value: integer
- domain: cat
- cardinality: at most 1
- range:  0 <= X <= 100
- oligonucleiotides < 20 base pairs
- cows are larger than dogs
- cats cannot eat only vegetation
- cats and dogs are disjoint

*Values* or *concrete domains*

- integer, strings
- 20, trypotoplan-synthetase






[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]

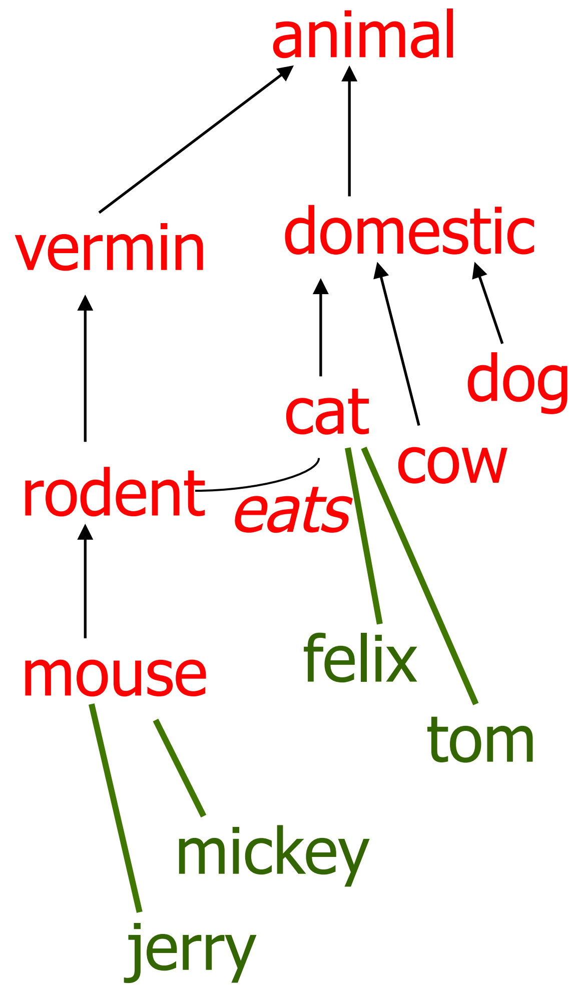
# An explicit description of a domain

*Individuals* or *Instances*

- sulphur, trpA Gene, **felix**

*Ontology* versus *Knowledge Base*

- An *ontology* = concepts+properties+axioms +values
- A *knowledge base* = ontology+instances






[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]

# Difference between Relational Schema and Ontology

Relational Schema primary purpose is to organise data within a database

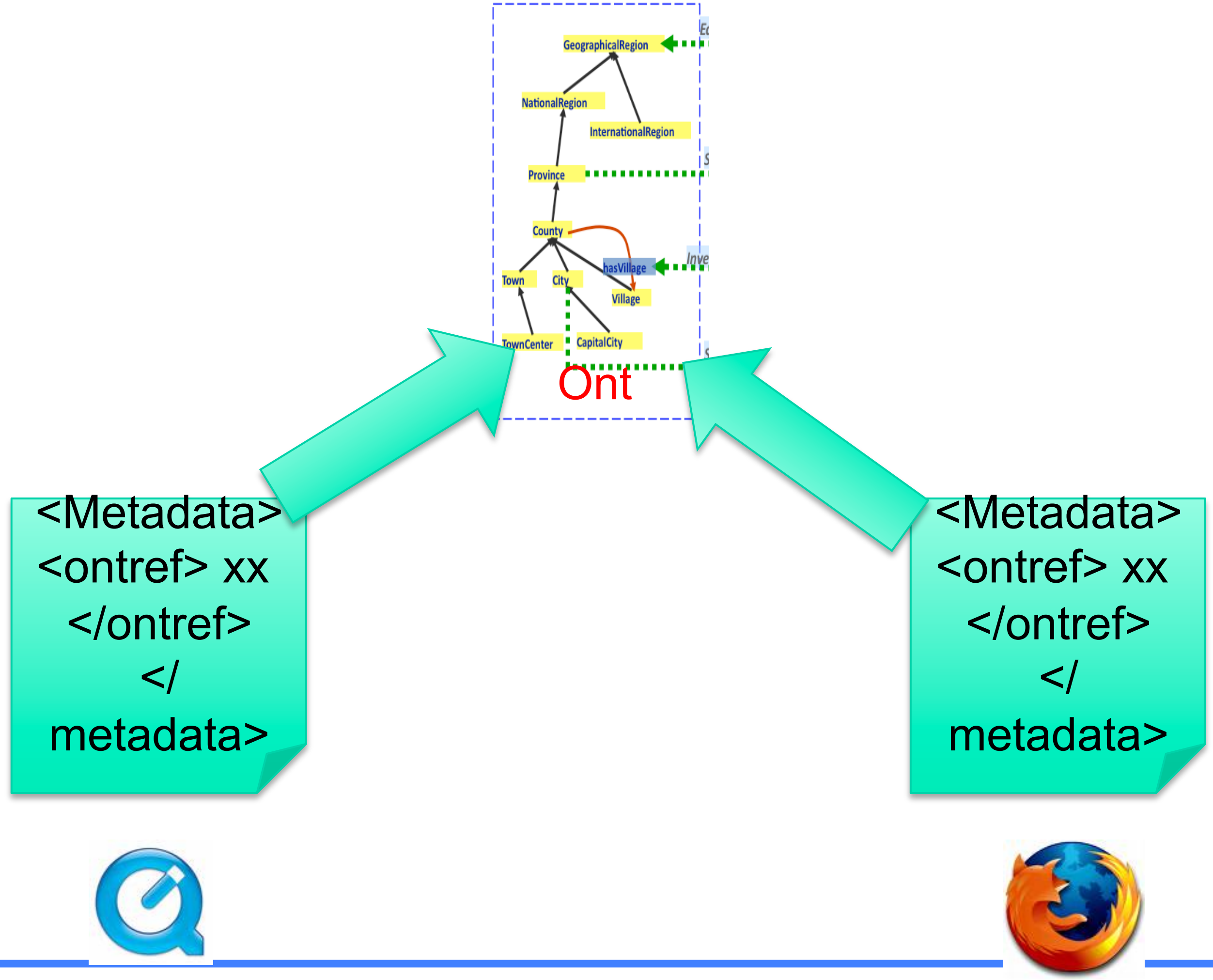
- As we have seen previously relationships between entities are implicit and interpretation is necessary by program/human who accesses
- If the program/human lacks knowledge of semantics of the data or database then becomes unusable

Ontology

- Relationships defined formally and interpretable by both human and program

# Problem Solved!... Right?

- Metadata + XML + Ontology

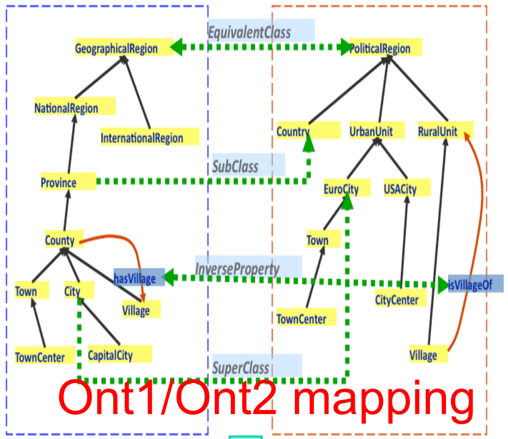


# … not quite…

- Ontologies are defined from a perspective
- “Whether it is a personal ontology, designed to support individual’s needs and preferences, or an ontology created by a particular company to reflect that company’s views on their industry, such ontologies will have biases and necessarily subjective features. …
- .. If that individual or company needs to exchange information and knowledge with other individuals or organisations, there will be a need to map between multiple ontologies either at inter-personal, intra-organisational or inter-organisational levels” [Hameed et al. 2004]

# Solution?

- Metadata + XML + Ontology + Ontology Mapping




<Metadata>
<ont1ref> xx
</ont1ref>
</metadata>

<Metadata>
<ont2ref> xx
</on2ref>
</metadata>

# Ontology Design

Methodology based on http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

University ontology based on COBRA ontology for “intelligent spaces”
from University of Maryland

# Ontology terminology that we will use

Class: Description of a **concept** in domain of discourse

- E.g. "Room" in the intelligent space domain

Slot/Property: Describes **attribute** of a concept or **relationship** of a concept

- E.g. longtitude and latitude of a room provides its coordinates
- E.g. isSpatiallySubsumedBy – points to instance of class "Building"

Facet/Restriction: Describes a **constraint** on a slot

- Slot cardinality e.g. one or many values allowed in slot
- Slot value type, e.g. String, Number, Boolean, Enummerated, Instance

# Fundamental Rules in Ontology Design

1) There is no one correct way to model a domain— there are always viable alternatives.
   - The best solution almost always depend on the application that you have in mind and the extensions that you anticipate.

2) Ontology development is necessarily an iterative process.

3) Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.
   - These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain

# Step 1: Determine Scope and Domain of Ontology

What is the domain that the ontology will cover?

- E.g representing an intelligent Universtity space

For what  we are going to use the ontology?

- E.g. applications that  support roaming students, lecturers and visitors are going to use the ontology

For what types of questions the information in the ontology should provide answers?

- E.g. to help roaming student print to the nearest printer etc.
- E.g. to help the university intelligent space applications to implement energy conservation policies etc.

Who will use and maintain the ontology?

- E.g. if it is maintained by someone in TCD for TCD users but needs also to be used by visiting users from UCD etc., then mappings need to be included

# Step 1: Competency Questions

One way to determine scope is to list out the questions the knowledge base based on the ontology should be able to answer

For example

- Which rooms have data projectors?
- Is ORI G.14 a conference room?
- Is there a conference room near restrooms?
- What is the best choice of room for a student tutorial?

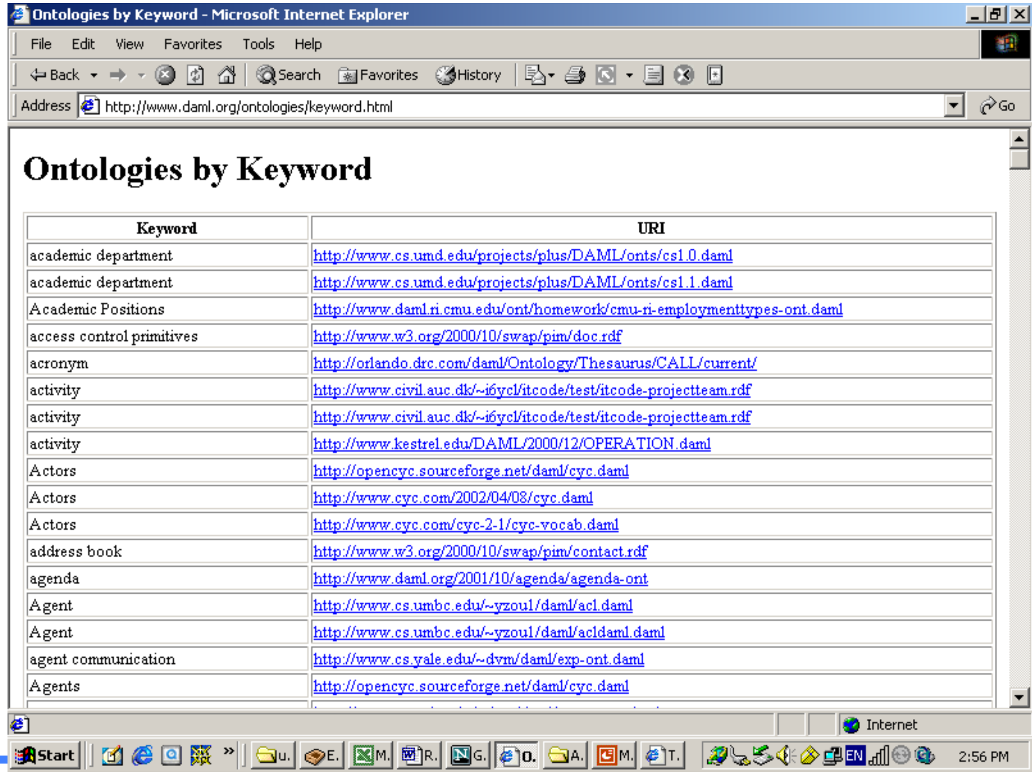
Judging from this list of questions, the ontology will include information on

- Various room types and characteristics
- Various event types and characteristics
- Classifications of rooms with respect to appropriateness for events
- Classifications of rooms with respect to location
- Classifications of rooms with respect to furniture/equipment

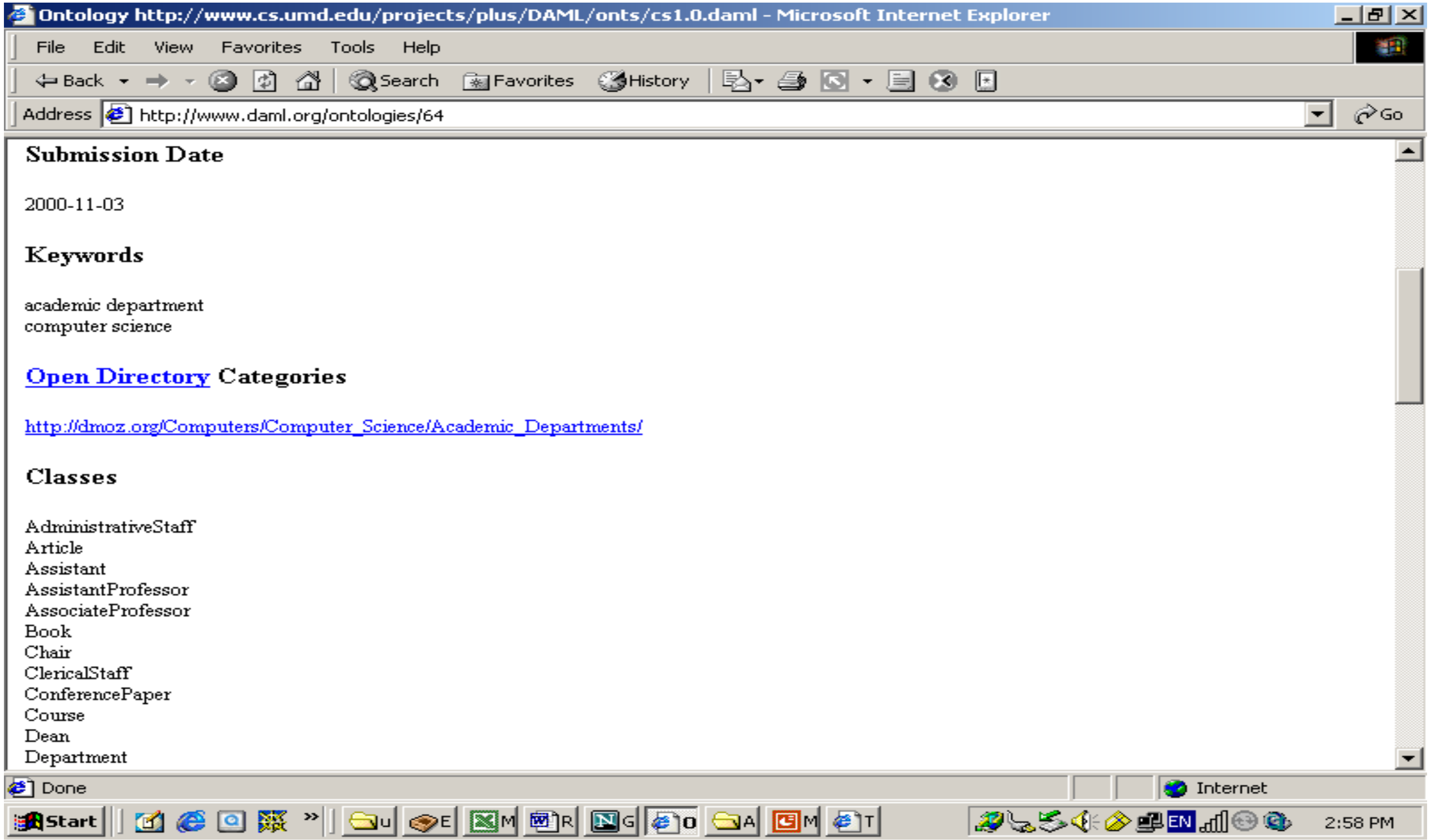
# Step 2: Consider reusing other ontologies

## Examples

- UNSPSC ontology which provides terminology for products and services (www.unspsc.org)
- DAML ontologies



# Example Ontology: CS Academic Dept



# SWOOGLE

http://swoogle.umbc.edu/

# Step 3: Enumerate important terms in the ontology

List of all terms we would like either to make statements about or to explain to a user.

- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?

For example, important a university intelligent space related terms will include

- Building, campus, restroom, room, hallway, stairs, chairs, projectors;
- different types of event, such as lecture, tutorial, seminar, meeting;
- and so on.

Get a comprehensive list of terms without worrying about:

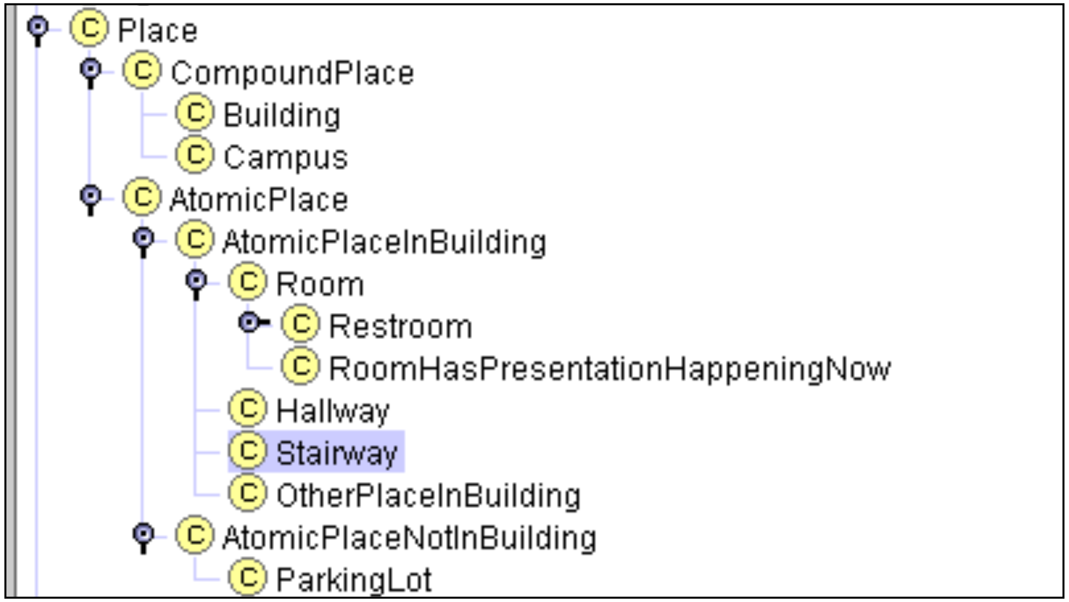
- overlap between concepts they represent,
- relations among the terms,
- or any properties that the concepts may have,
- or whether the concepts are classes or slots.

# Step 4: Define the classes and class hierarchy

From the list created in Step 3, we select the terms that describe objects having independent existence rather than terms that describe these objects.

These terms will be classes in the ontology and will become anchors in the class hierarchy

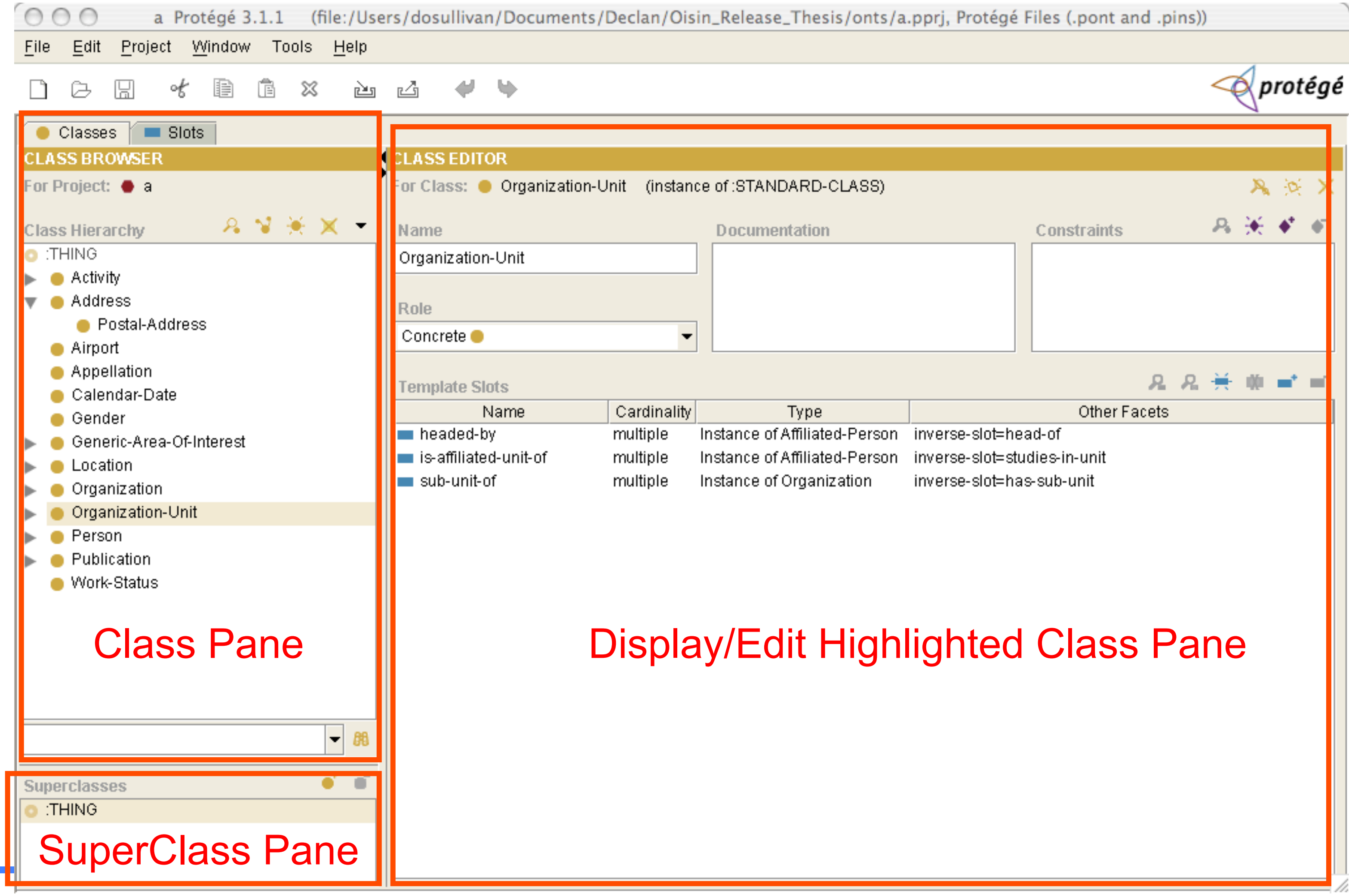
**Multiple Inheritance allowed**



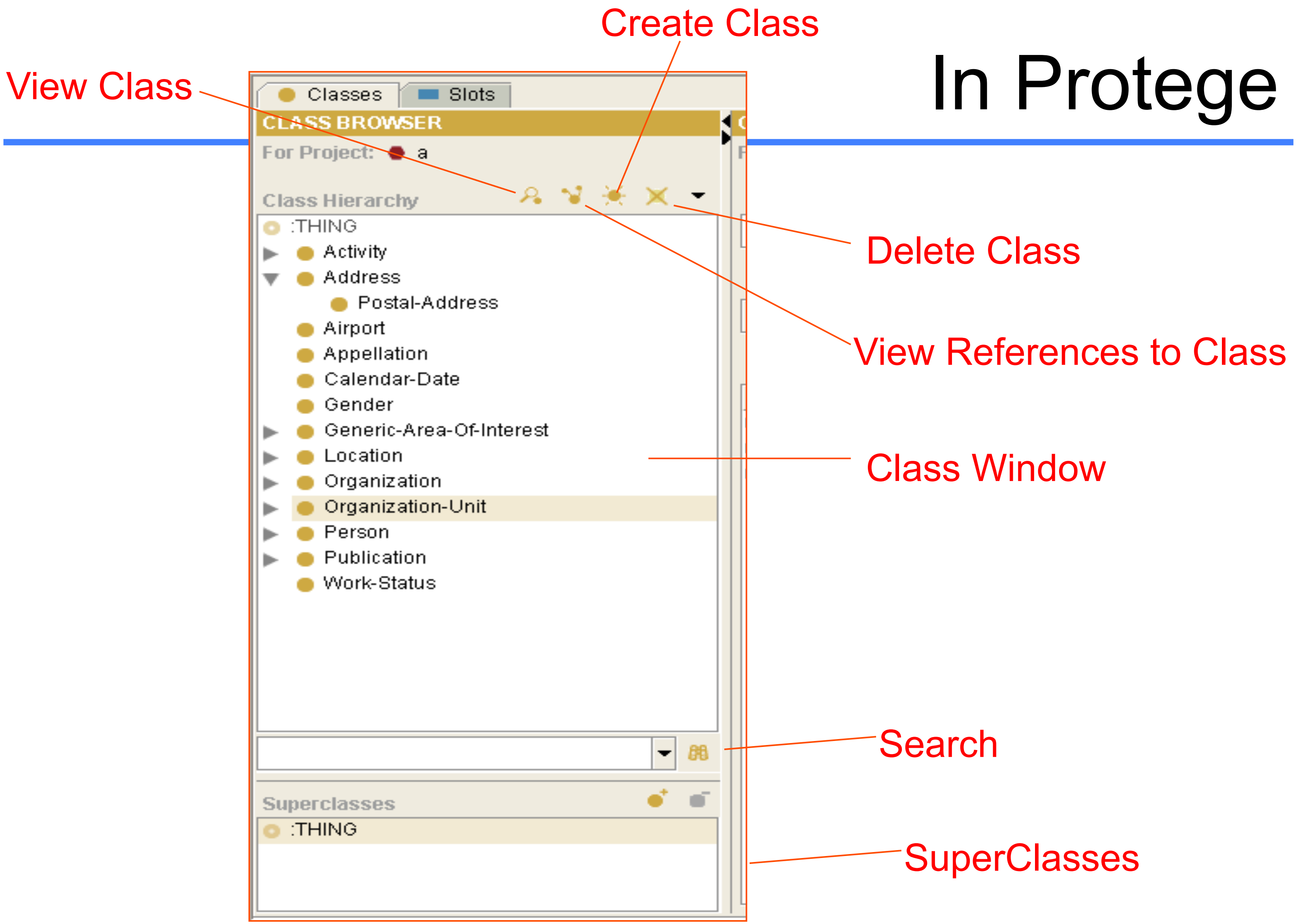
# More on Classes When to introduce a new class or not?

- The ontology should not contain all the possible information about the domain: you do not need to specialize (or generalize) more than you need for your application (at most one extra level each way).
- Subclasses of a class usually (1) have additional properties that the superclass does not have, or (2) restrictions different from those of the superclass, or (3) participate in different relationships than the superclasses
  - *E.g. a conference room may have a property "booked by" that room does not*
- Classes in terminological hierarchies do not have to introduce new properties
  - *For example, some ontologies include large reference hierarchies of common terms used in the domain*
- We should not create subclasses of a class for each additional restriction.
  - *For example, we could introduce Conference Room, Office Room etc. because this distinction is a natural one in the real world. We may not want to introduce classes for store room, secretary room, and so on. When defining a class hierarchy, our goal is to strike a balance between creating new classes useful for class organization and creating too many classes*

# In Protege


a Protégé 3.1.1 (file:/Users/dosullivan/Documents/Declan/Oisin_Release_Thesis/onts/a.pprj, Protégé Files (.pont and .pins))
File Edit Project Window Tools Help
protégé
Classes
Slots
CLASS BROWSER
For Project: a
Class Hierarchy
:THING
Activity
Address
Postal-Address
Airport
Appellation
Calendar-Date
Gender
Generic-Area-Of-Interest
Location
Organization
Organization-Unit
Person
Publication
Work-Status
Class Pane
Superclasses
:THING
SuperClass Pane
CLASS EDITOR
For Class: Organization-Unit (instance of :STANDARD-CLASS)
Name
Organization-Unit
Documentation
Constraints
Role
Concrete
Template Slots
Name
Cardinality
Type
Other Facets
headed-by
multiple
Instance of Affiliated-Person
inverse-slot=head-of
is-affiliated-unit-of
multiple
Instance of Affiliated-Person
inverse-slot=studies-in-unit
sub-unit-of
multiple
Instance of Organization
inverse-slot=has-sub-unit
Display/Edit Highlighted Class Pane

# In Protege


Create Class
View Class
Delete Class
View References to Class
Class Window
Search
SuperClasses
Classes
Slots
CLASS BROWSER
For Project: a
Class Hierarchy
:THING
Activity
Address
Postal-Address
Airport
Appellation
Calendar-Date
Gender
Generic-Area-Of-Interest
Location
Organization
Organization-Unit
Person
Publication
Work-Status
Superclasses
:THING

# Step 5: Define the properties of classes - slots

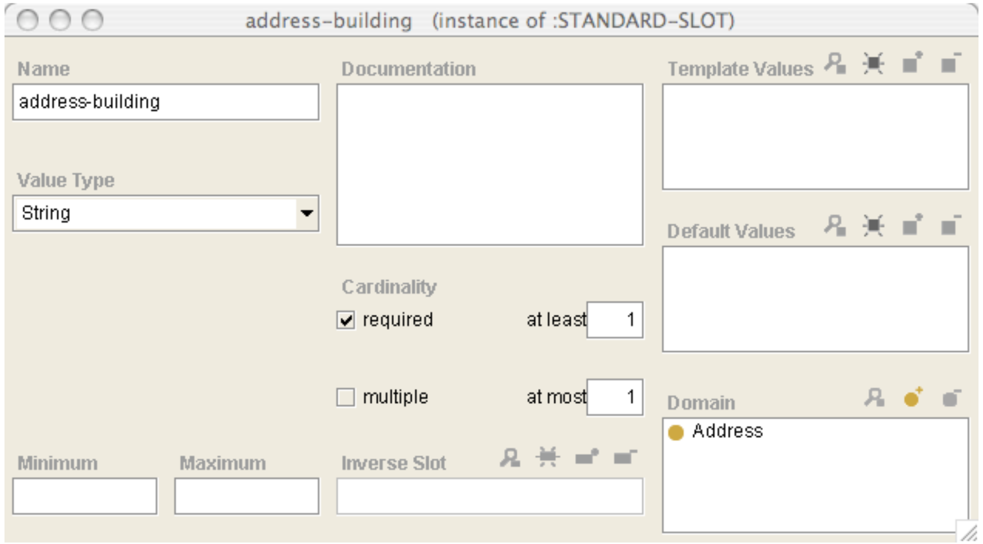
In general, there are several types of object properties that can become slots in an ontology:

- “intrinsic” properties such as the capacity of the room;
- “extrinsic” properties such as a room’s name;
- parts, if the object is structured; these can be both physical and abstract “parts” (e.g., the courses of a meal)
- relationships to other individuals; these are the relationships between individual members of the class and other items (e.g., spatiallySubsumedBy, representing a relationship between a room and a CompoundPlace (subclasses are Building and Campus))

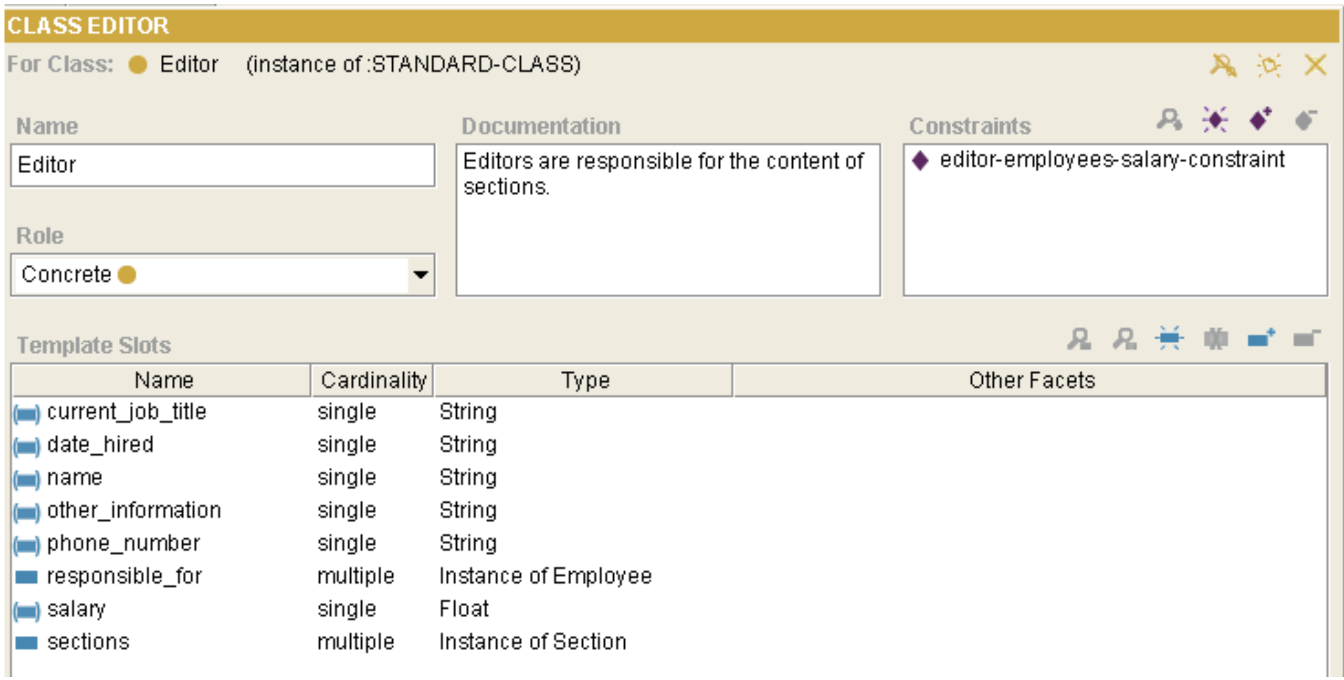
# Step 6: Define the Facets of each Slot

Facet: Describes a restriction on a slot

- Slot cardinality e.g. one or many values allowed in slot
- Slot value type, e.g. String, Integer, Boolean, Enummerated, Instance



# In Protege

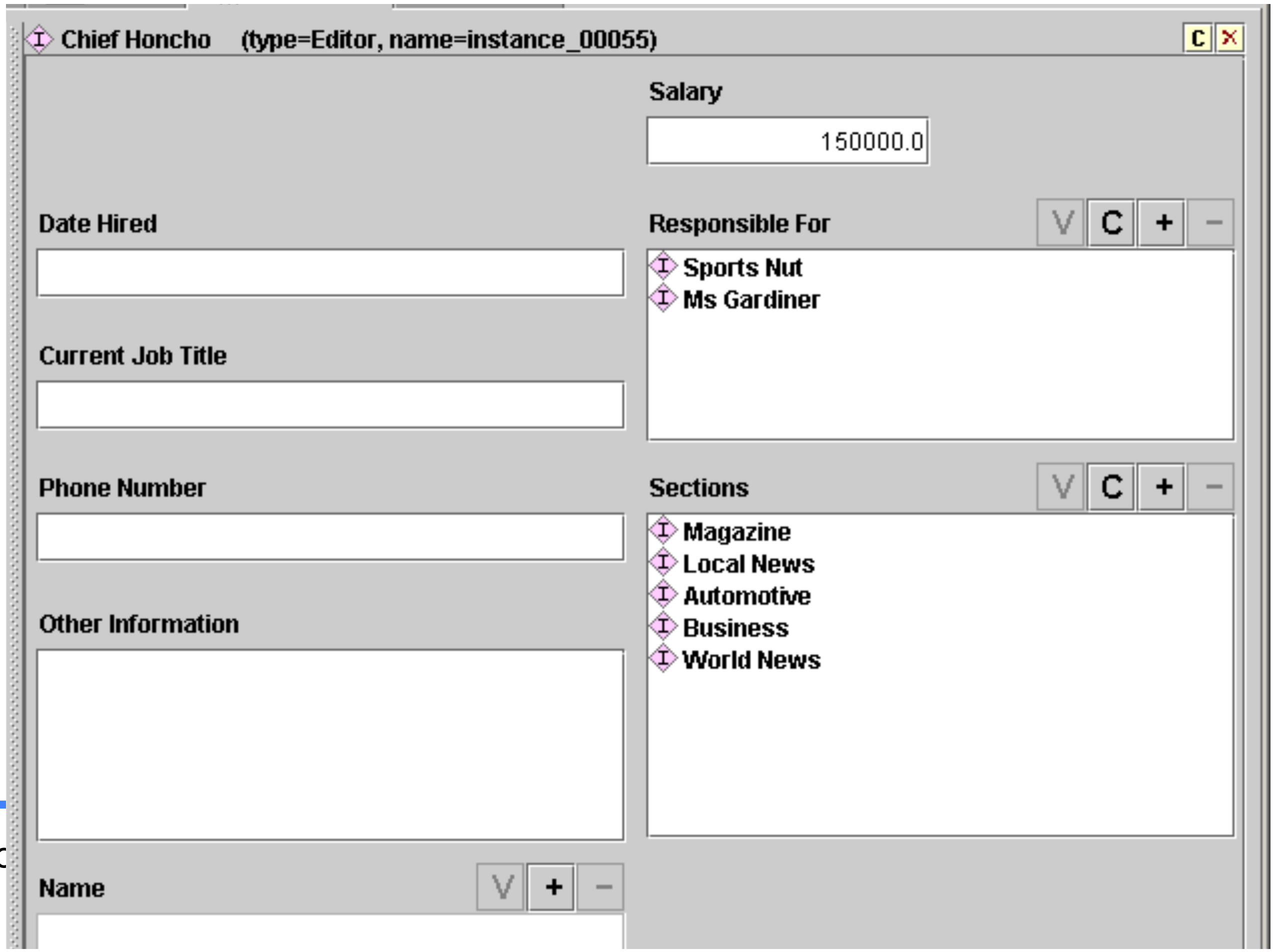



Blue rectangle means property defined on this class
Round brackets around rectangle means property inherited

# Step 7: (Optionally) Create Instances

Chief Honcho is an instance of the class Editor

This instance has the following slot values defined



# Seven steps to…

1. Determine scope and domain of ontology
2. Consider reusing existing ontologies
3. Ennumerate important terms in the ontology
4. Define the classes and class hierarchy
5. Define the properties of classes – slots
    - Properties, relationships
6. Define the facets of slots
    - Value type (String, Number, Boolean, Enummerated, Instance)
    - Value cardinality
7. Define the instances

# Questions?

Owen.Conlan@scss.tcd.ie

Many of these slides originally produced by Dr. Declan O'Sullivan