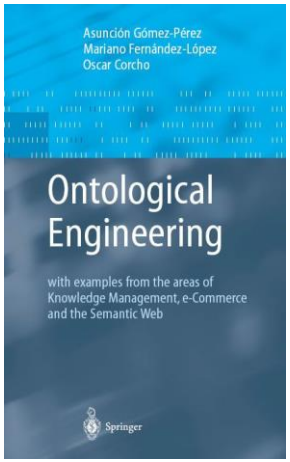
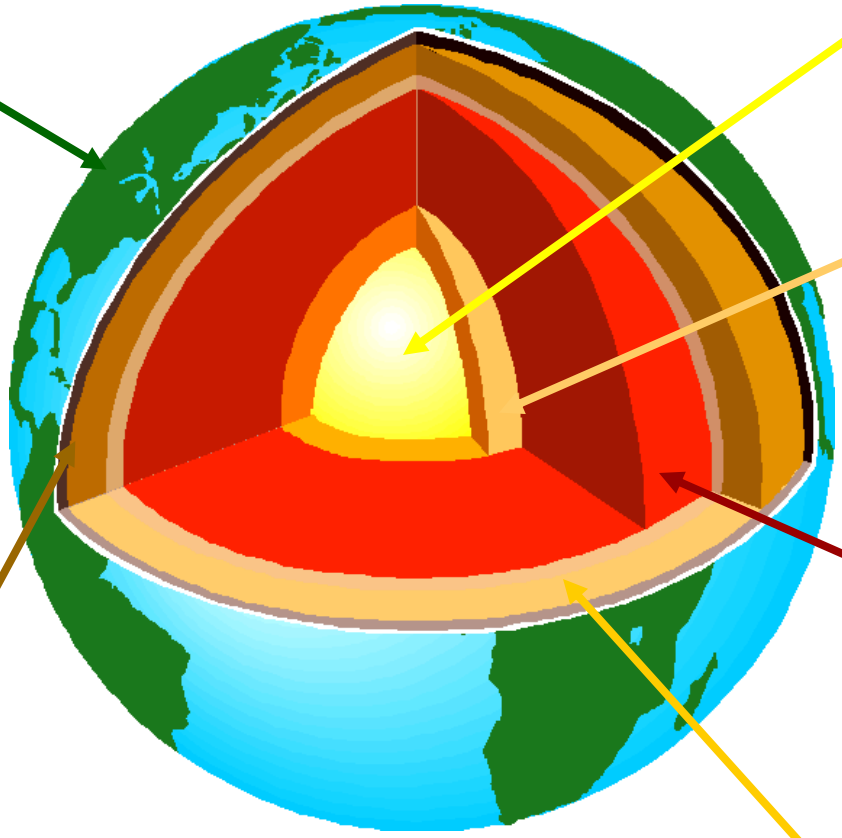
Main Components

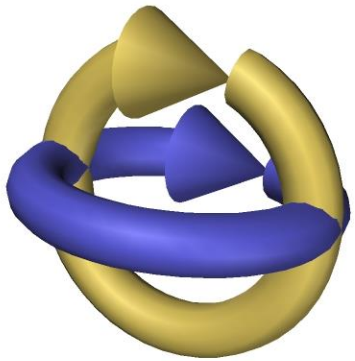
Applications

Build Ontologies





Methodologies and Methods



Technological Support

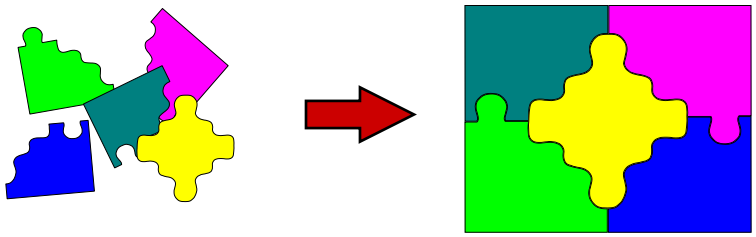
Reasoners

Languajes

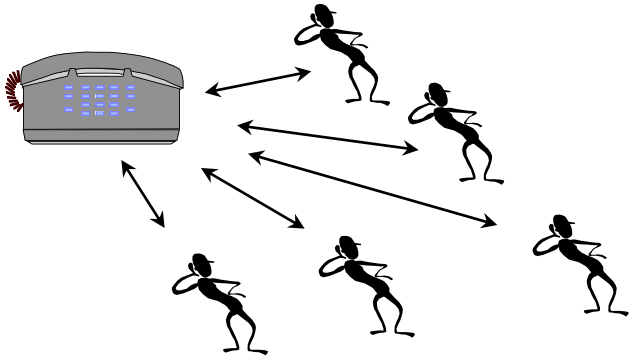
Table of Content

Reuse and Sharing

Reuse means to build new applications assembling components already built



Sharing is when different applications use the some resources



Advantages:

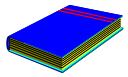
- Less money
- Less time
- Less resources

Areas:

- Software
- Knowledge
- Communications
- Interfaces
- ---

The knowledge Sharing Initiative

"Building new Knowledge Based Systems today usually entails constructing new knowledge bases from scratch. It could instead be done by assembling reusable components. System developers would then only need to worry about creating the specialized knowledge and reasoners new to the specific task of their systems. This new system would interoperate with existing systems, using them to perform some of its reasoning. In this way, declarative knowledge, problem-solving techniques, and reasoning services could all be shared between systems. This approach would facilitate building bigger and better systems cheaply. The infraestructure to support such sharing and reuse would lead to greater ubiquity of these systems, potentially transforming the knowledge industry ..."



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.

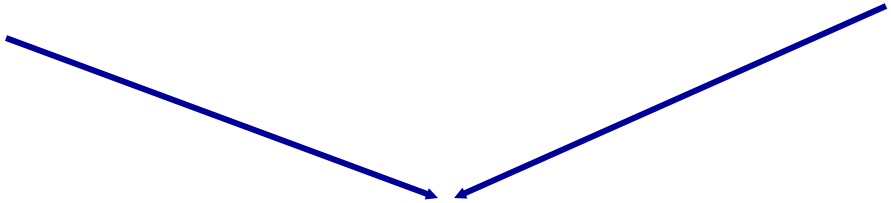
Reusable Knowledge Components

Ontologies

Describe domain knowledge in a generic way and provide agreed understanding of a domain

Problem Solving Methods

Describe the reasoning process of a KBS in an implementation and domain-independent manner

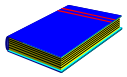


Interaction Problem

Representing Knowledge for the purpose of solving some problem

is strongly affected by the nature of the problem

and the inference strategy to be applied to the problem [Bylander et al., 88



Bylander Chandrasekaran, B. **Generic Tasks in knowledge-based reasoning.: the right level of abstraction for knowledge acquisition.** In B.R. Gaines and J. H. Boose, EDs *Knowledge Acquisition for Knowledge Based systems*, 65-77, London: Academic Press 1988.

Ontological Engineering

It refers to the set of activities that concern

the ontology development process,

the ontology life cycle,

the methods and methodologies for building ontologies,

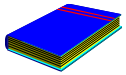
and the tool suites

and languages that support them.

Table of Content

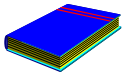
Definitions of Ontologies (I)

1. "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.

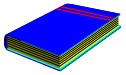
2. "An ontology is an explicit specification of a conceptualization"



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

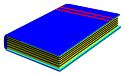
Definitions of Ontologies (II)

3. 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.

4. 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.

Definitions of Ontologies (III)

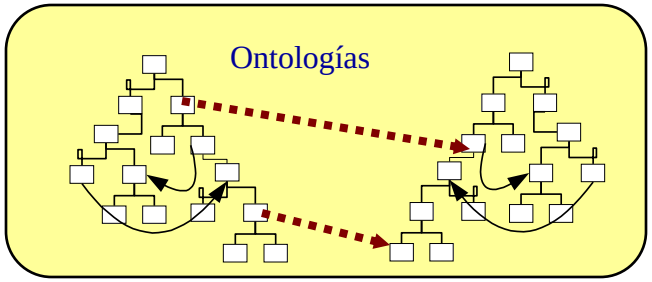
5. "An ontology is a formal, explicit specification of a shared conceptualization"

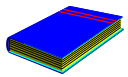
Machine-readable

Concepts, properties relations, functions, constraints, axioms, are explicitly defined

Consensual Knowledge

Abstract model and simplified view of some phenomenon in the world that we want to represent





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

Definitions of Ontologies (IV)

Lightweight Ontologies :

- **Include Concepts with properties and Taxonomies**
- **Do not include Axioms and constraints.**

Heavyweight Ontologies :

- **Include all the components**
- **Excellent!! If they have a lot of axioms.**

Table of Content

Components of an Ontology

Concepts are organized in **taxonomies**

Relations R: $C_1 \times C_2 \times ... \times C_{n-1} \times C_n$

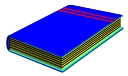
Subclass-of: Concept 1 x Concept2
Connected to: Component1 x Component2

Functions F: $C_1 \times C_2 \times ... \times C_{n-1} \rightarrow C_n$

Mother-of: Person --> Women
Price of a used car: Model x Year x Kilometers --> Price

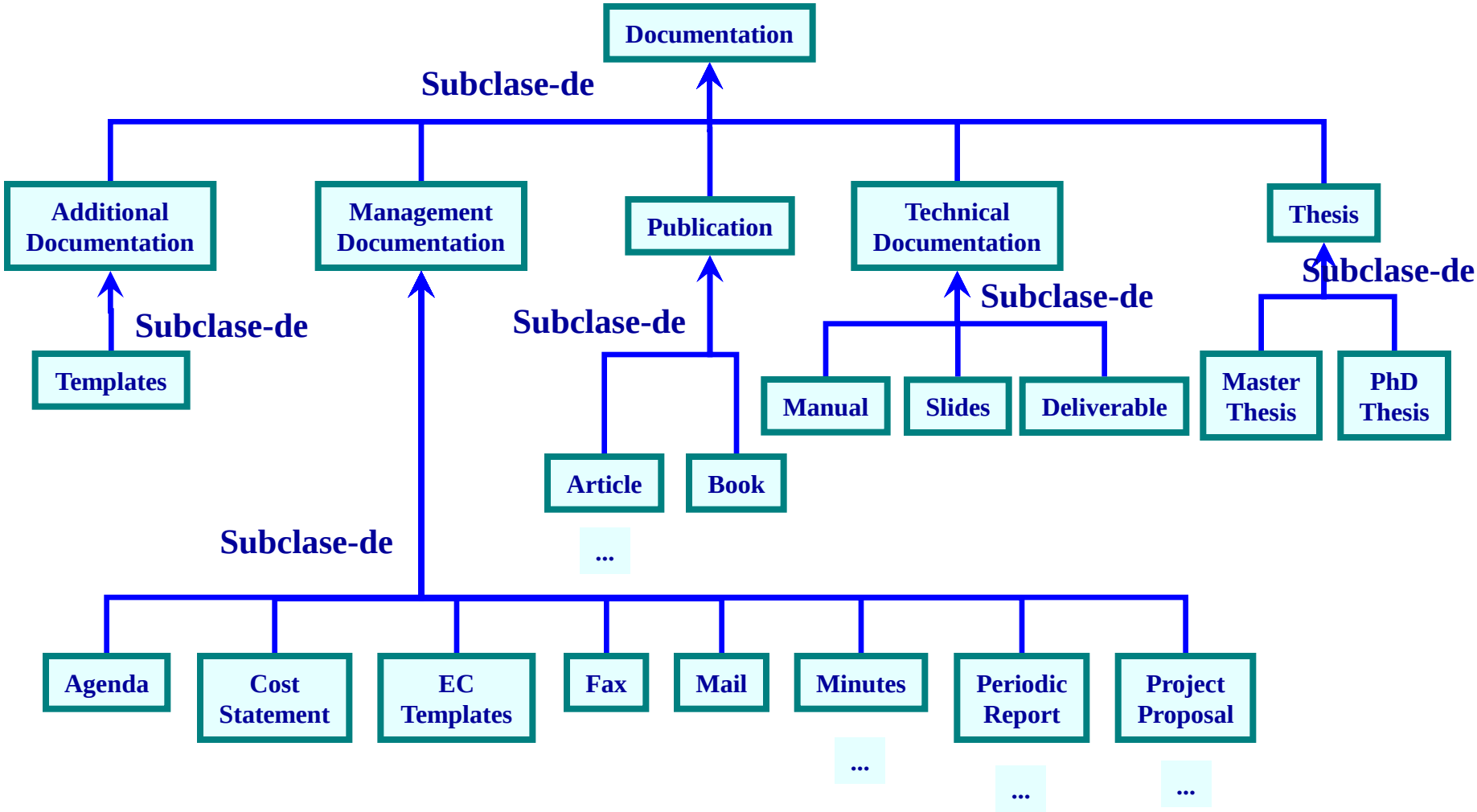
Instances Elements

Axioms Sentences which are always true



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

Documentation Taxonomy



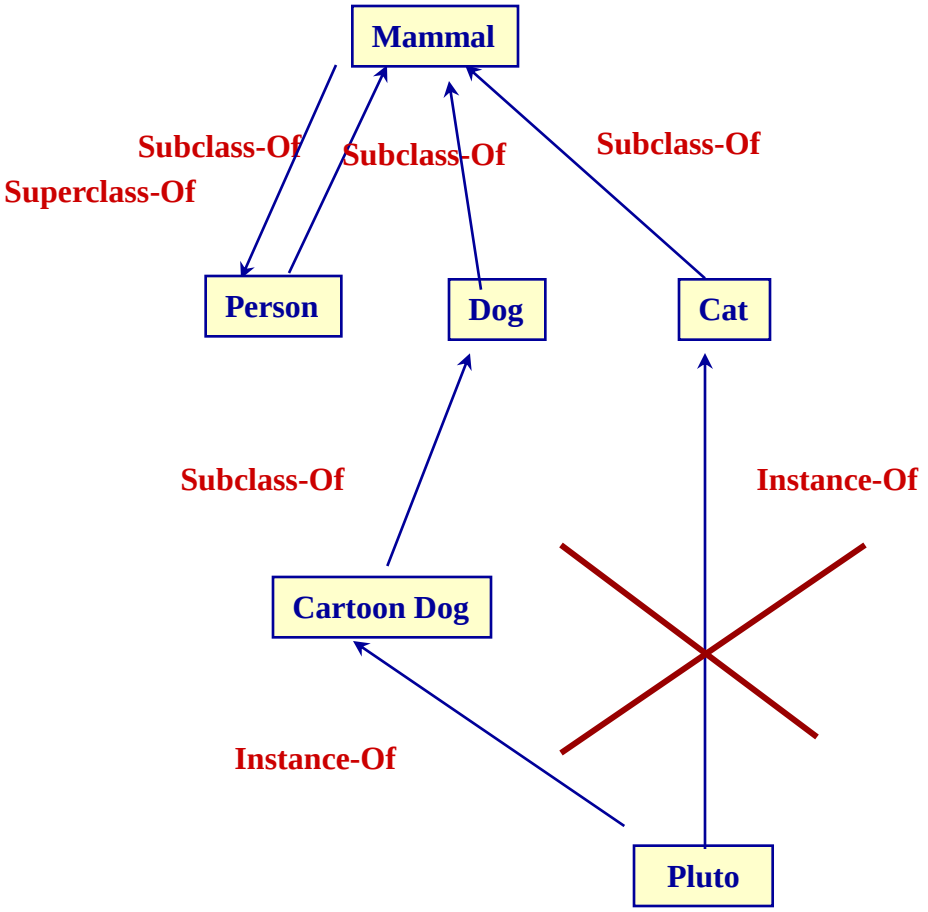
Modelling disjoint knowledge

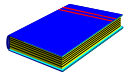
class-Partition: a set of disjoint classes

Disjoint: Defines the set of classes in the partition as subclasses of the parent class. This classification does not necessarily to be complete.

Exhaustive-Disjoint: Defines the set of classes in the partition as subclasses of the parent class. This classification is complete.

Why disjoint knowledge is important (I)



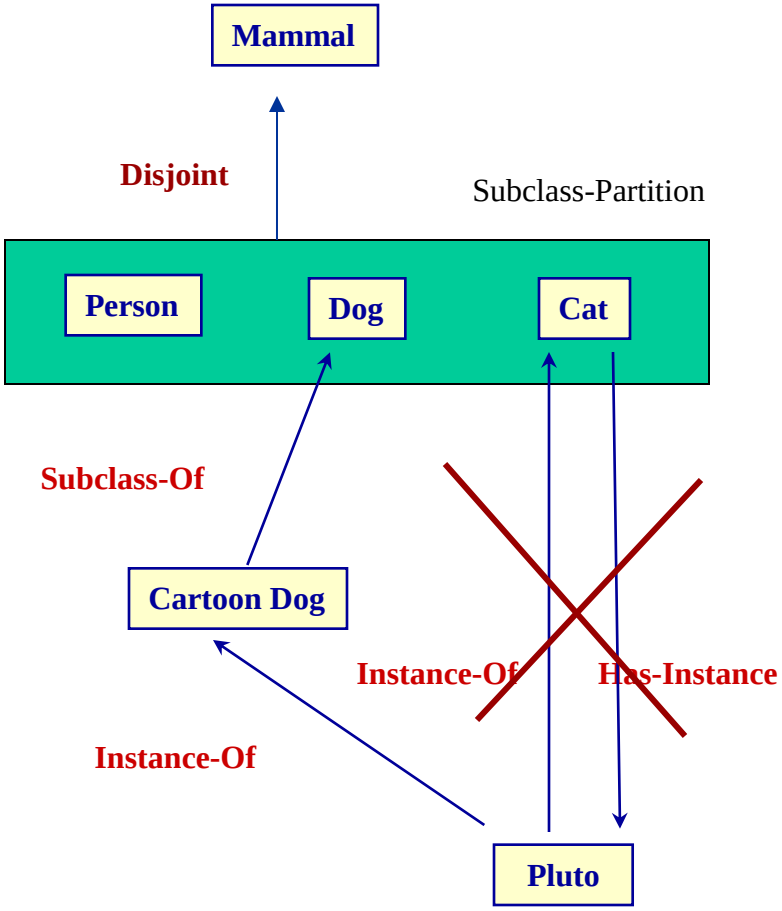


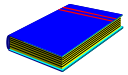
A. Gómez-Pérez. *Evaluation of Ontologies.*
International Journal of Intelligent Systems.
Vol. 16, Nº3. March 2001. PP391-410

Pluto could be an instance of cat and dog

Semantic Error

Why disjoint knowledge is important (II)

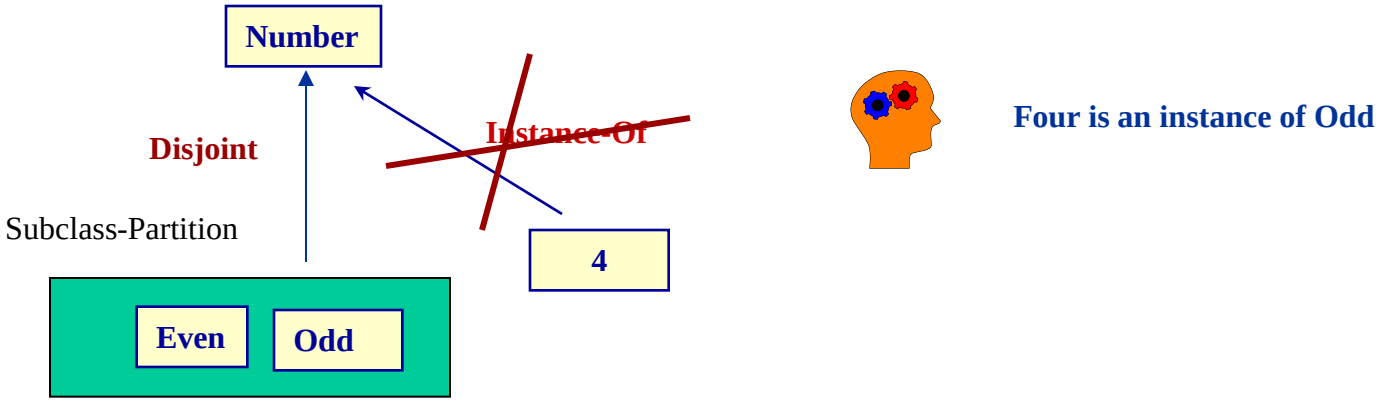


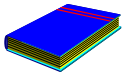


A. Gómez-Pérez. *Evaluation of Ontologies.*
International Journal of Intelligent Systems.
Vol. 16, Nº3. March 2001. PP391-410

Pluto can not be simultaneously a class of Cat and Dog because they are disjoint

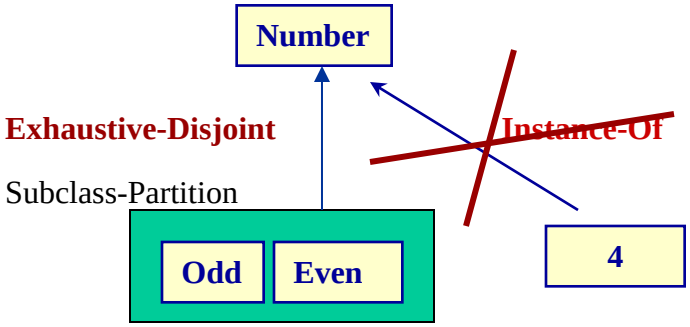
Why disjoint knowledge is important (III)





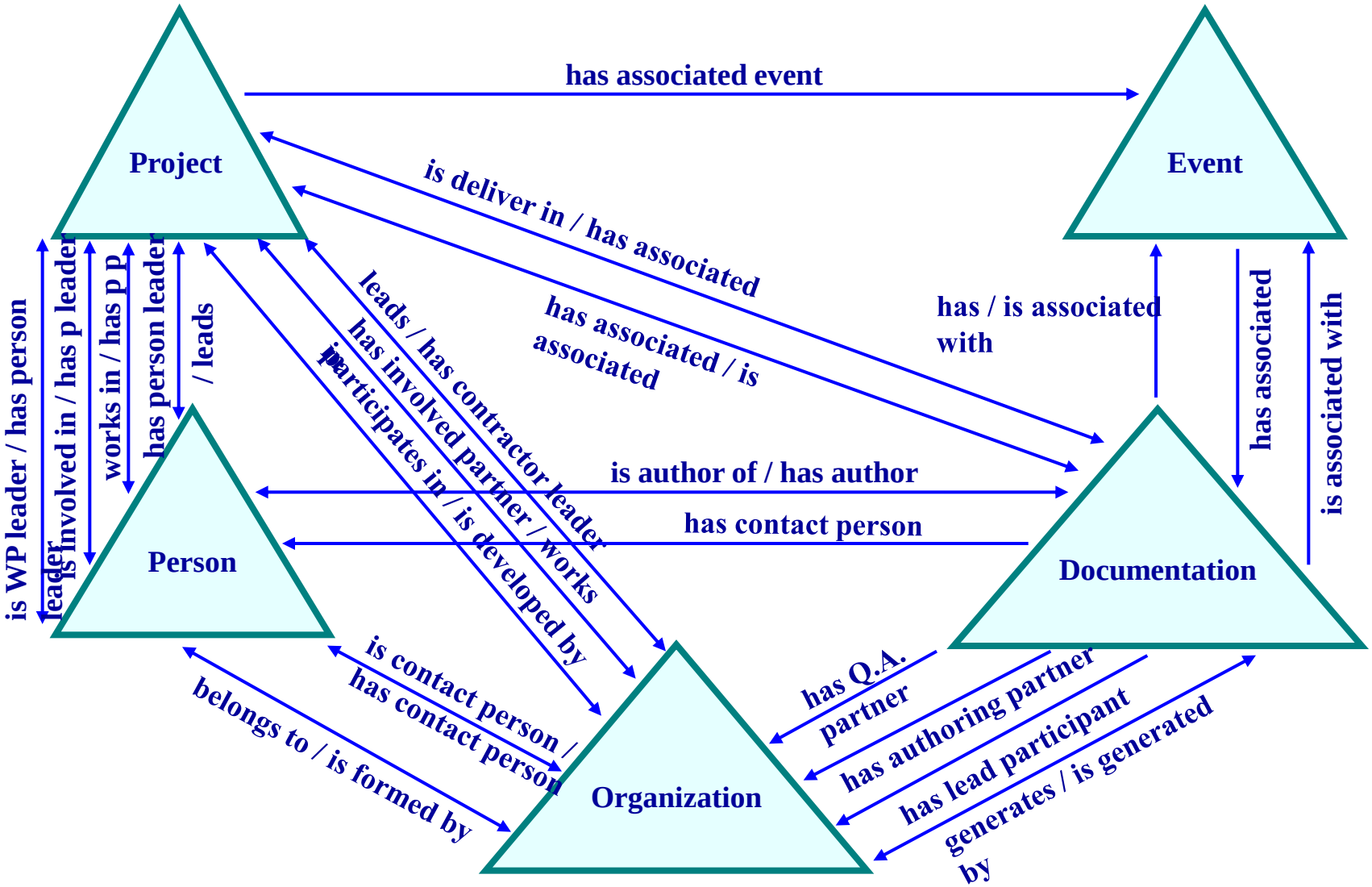
A. Gómez-Pérez. *Evaluation of Ontologies.* **International Journal of Intelligent Systems.** Vol. 16, Nº3. March 2001. PP391-410

Why disjoint knowledge is important (IV)

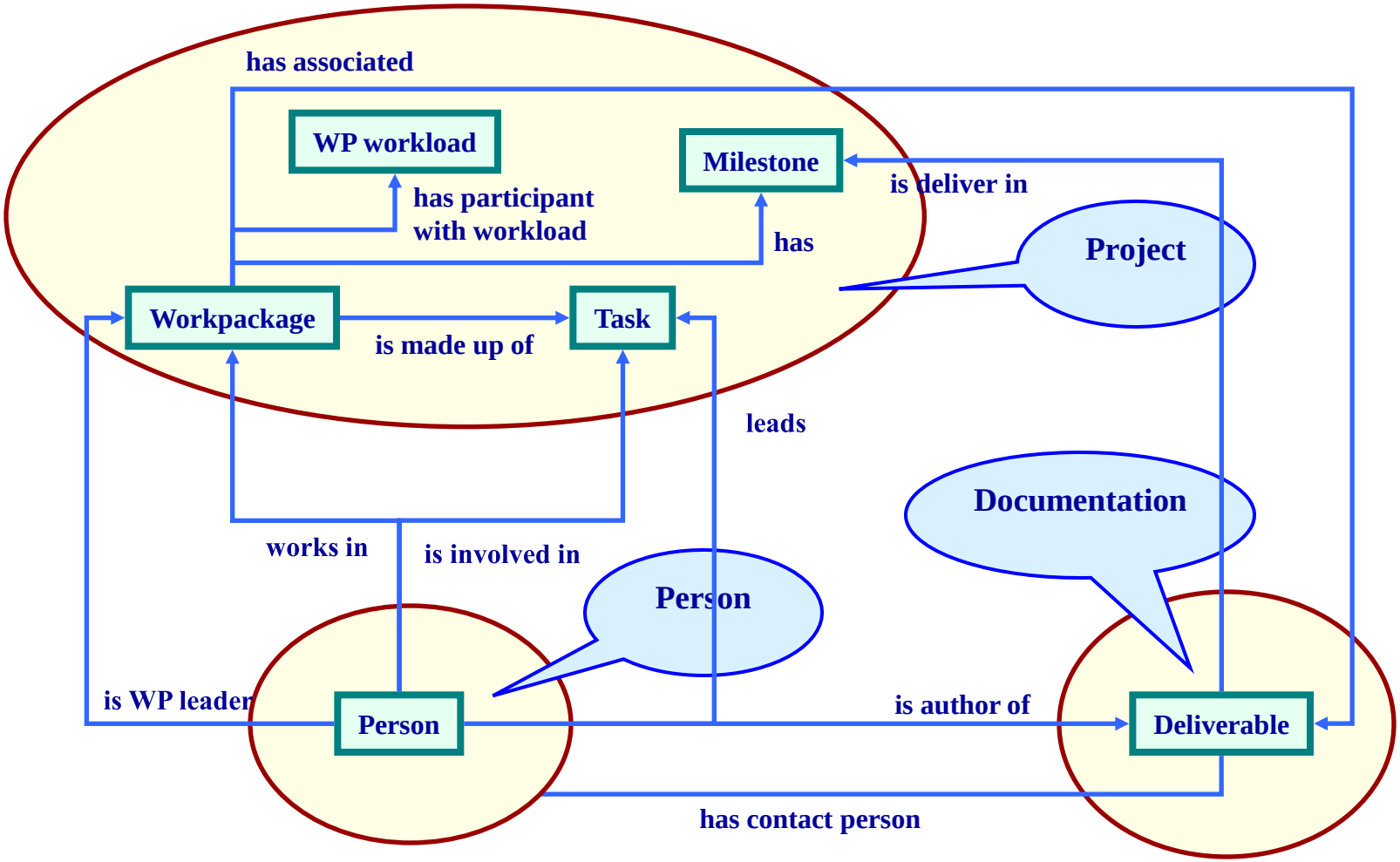


Four is an instance of something in the partition

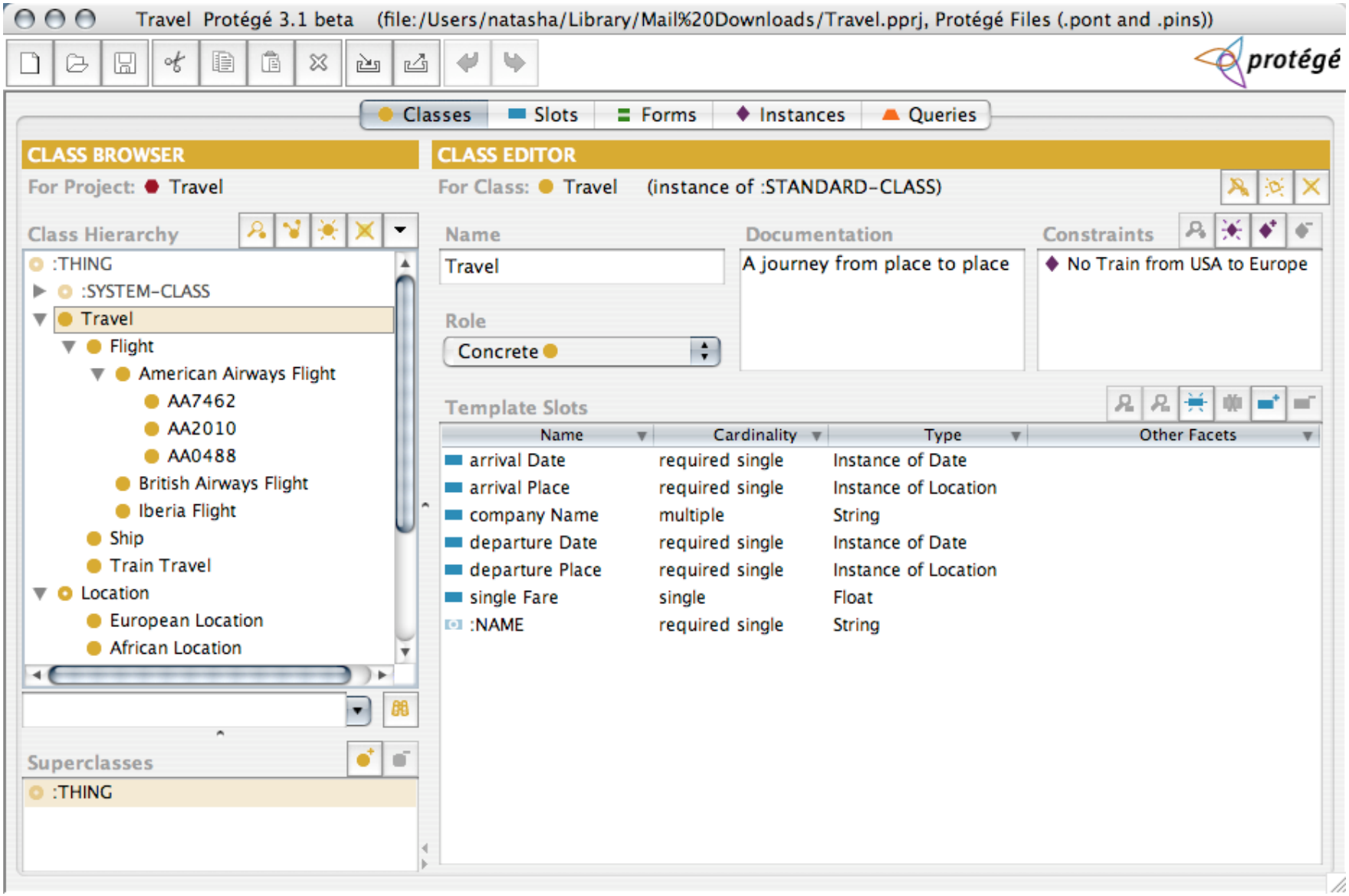
Relations between concepts



Relationships between Person, Project and Documentation



Properties


Travel Protégé 3.1 beta (file:/Users/natasha/Library/Mail%20Downloads/Travel.pprj, Protégé Files (.pont and .pins))
protégé
Classes
Slots
Forms
Instances
Queries
CLASS BROWSER
For Project: Travel
Class Hierarchy
:THING
:SYSTEM-CLASS
Travel
Flight
American Airways Flight
AA7462
AA2010
AA0488
British Airways Flight
Iberia Flight
Ship
Train Travel
Location
European Location
African Location
Superclasses
:THING
CLASS EDITOR
For Class: Travel (instance of :STANDARD-CLASS)
Name
Travel
Documentation
A journey from place to place
Constraints
No Train from USA to Europe
Role
Concrete
Template Slots
Name
Cardinality
Type
Other Facets
arrival Date
required single
Instance of Date
arrival Place
required single
Instance of Location
company Name
multiple
String
departure Date
required single
Instance of Date
departure Place
required single
Instance of Location
single Fare
single
Float
:NAME
required single
String

Example of axioms

```
(define-axiom No-Train-from-USA-to-Europe
   "It is not possible to travel from the USA to Europe by train"
:= (forall (?travel)
    (forall (?city1)
     (forall (?city2)
       (=> (and (Travel ?travel)
                (arrivalPlace ?travel ?city1)
                (departurePlace ?travel ?city2)
                (EuropeanLocation ?city1)
                (USALocation ?city2))
           (not (TrainTravel ?travel)))))))


(define-axiom No-Train-between-USA-and-Europe
   "It is not possible to travel by train between the USA and Europe"
:= (forall (?travel)
    (forall (?city1)
     (forall (?city2)
      (=> (and (Travel ?travel)
               (arrivalPlace ?travel ?city1)
               (departurePlace ?travel ?city2)
               (or (and (EuropeanLocation ?city1)
                        (USALocation ?city2))
                   (and (EuropeanLocation ?city2)
                        (USALocation ?city1))))
          (not (TrainTravel ?travel)))))))
```

Key aspects of Ontological Engineering

- Ontologies
  - Single versus network of ontologies?
  - Are ontologies built from scratch or reusing knowledge-aware resources?
  - Are mappings used for solving conceptual mistmaches?
- Instances
  - Where are the data/instances?
    - Instances are in the ontology
    - Instances are in RDF files independently of the ontology
    - Data are kept in the original sources
  - Are instances distributed or centralized?
  - Have instances a very high rate of changes?
  - Heterogeneous provenance of instances
  - Degrees of data quality
  - Permissions

Where are the instances?

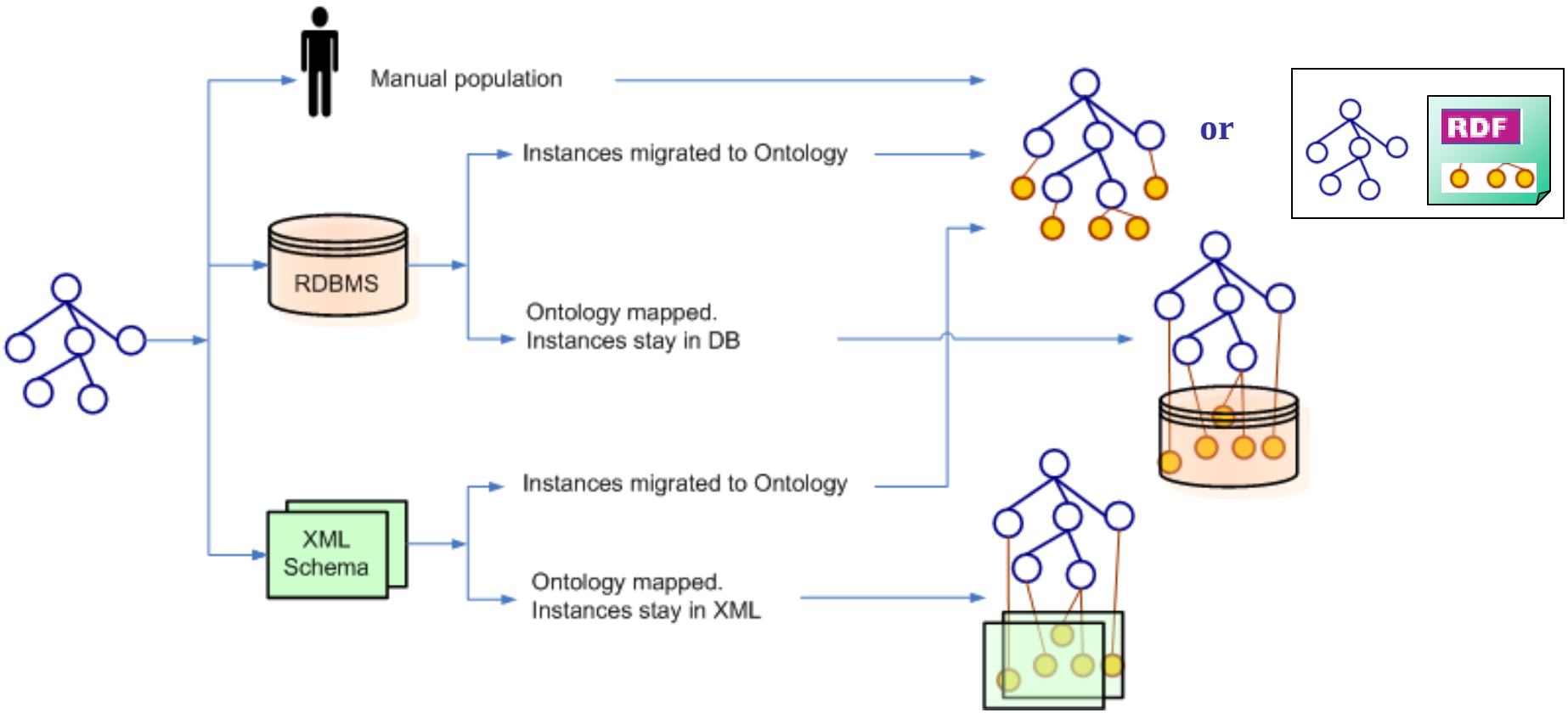
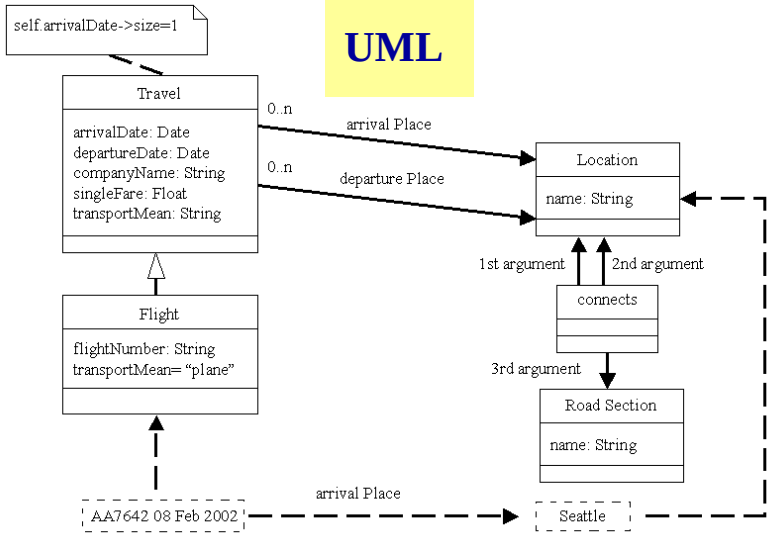


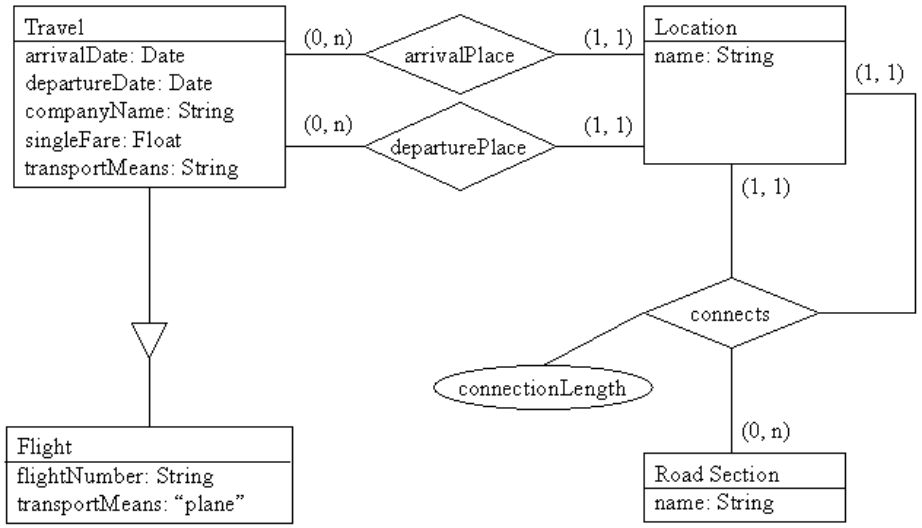
Table of Content

Approaches for building ontologies

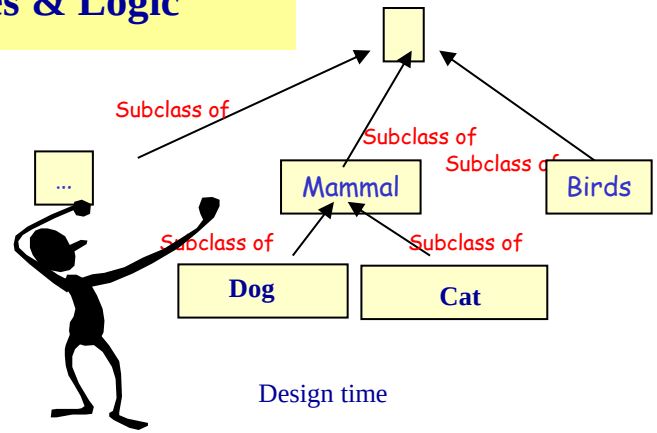
UML



E/R Model

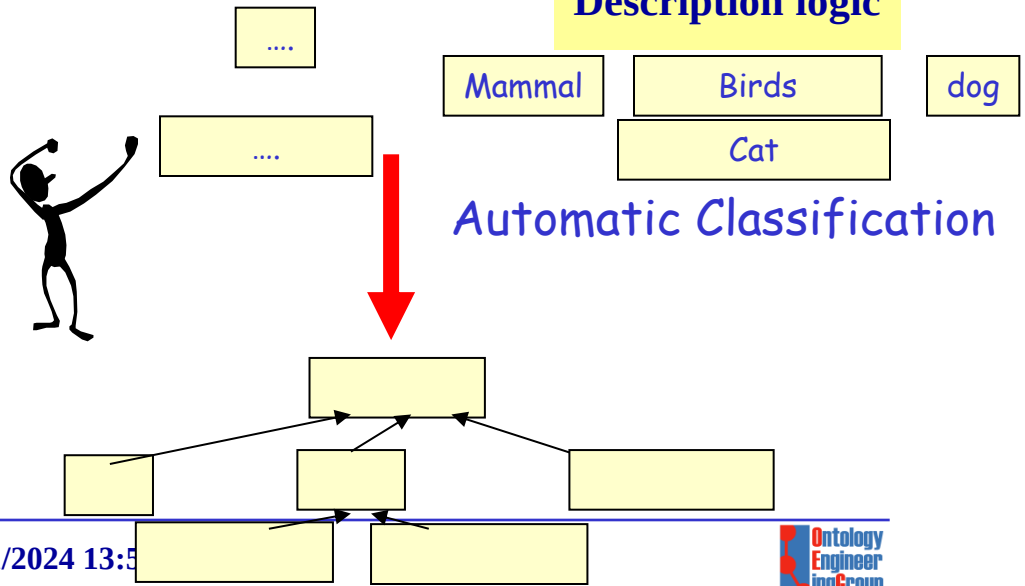


Frames & Logic



Design time

Description logic



Using Frames and First Order Logic for Modeling Ontologies

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Template-Facet-Value Cardinality
           arrivalDate Travel 1)
      (Template-Facet-Value Cardinality
           departureDate Travel 1)
      (Template-Facet-Value Maximum-Cardinality
           singleFare Travel 1))
:def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date)
       (singleFare ?travel Number)
       (companyName ?travel String)))
```

```
(define-instance AA7462-Feb-08-2002 (AA7462)
:def ((singleFare AA7462-Feb-08-2002 300)
      (departureDate AA7462-Feb-08-2002 Feb8-2002)
      (arrivalPlace AA7462-Feb-08-2002 Seattle)))
```

```
(define-function Pays (?room ?discount) :-> ?finalPrice
   "Price of the room after applying the discount"
:def (and (Room ?room) (Number ?discount)
          (Number ?finalPrice)
          (Price ?room ?price))
:lambda-body
          (- ?price (/ (* ?price ?discount) 100)))
```

```
(define-relation connects (?edge ?source ?target)
   "This relation links a source and a target by an edge.
The source and destination are considered as spatial
points. The relation has the following properties: symmetry
and irreflexivity."
:def (and (SpatialPoint ?source)
          (SpatialPoint ?target)
          (Edge ?edge))
:axiom-def
((=> (connects ?edge ?source ?target)
     (connects ?edge ?target ?source)) ;symmetry
 (=> (connects ?edge ?source ?target)
     (not (or (part-of ?source ?target) ;irreflexivity
              (part-of ?target ?source))))))
```

Using Description Logics for Modeling Ontologies

```
(defconcept Travel
   "A journey from place to place"
:is-primitive
  (:and
    (:all arrivalDate Date)(:exactly 1 arrivalDate)
    (:all departureDate Date)(:exactly 1
departureDate)
    (:all companyName String)
    (:all singleFare Number)(:at-most singleFare 1)))
```

```
(defrelation Pays
:is
 (:function (?room ?Discount)
  (- (Price ?room) (/(*(Price ?room) ?Discount) 100)))
:domains (Room Number)
:range Number)
```

```
(defrelation connects
  "A road connects two different cities"
:arity 3
:domains (Location Location)
:range RoadSection
:predicate
  ((?city1 ?city2 ?road)
   (:not (part-of ?city1 ?city2))
   (:not (part-of ?city2 ?city1))
   (:or (:and (start ?road ?city1)(end ?road ?city2))
        (:and (start ?road ?city2)(end ?road ?city1)))))
```

```
(tellm (AA7462 AA7462-08-Feb-2002)
      (singleFare AA7462-08-Feb-2002 300)
      (departureDate AA7462-08-Feb-2002 Feb8-2002)
      (arrivalPlace AA7462-08-Feb-2002 Seattle))
```

Different Approaches to Build Ontologies

- The formalism and the language limit the kind of knowledge that can be represented

- All the aforementioned formalisms allow representing: classes, organized in class taxonomies, attributes, and binary relations

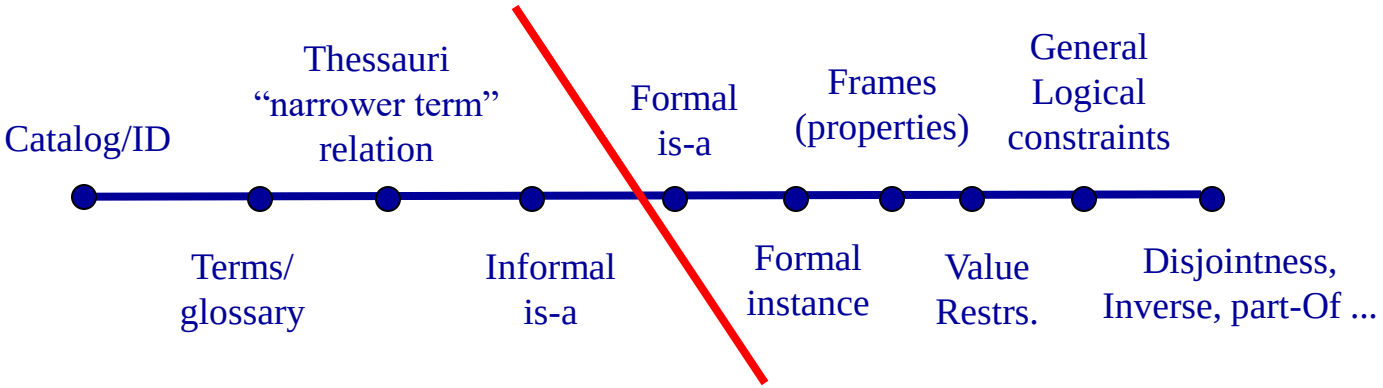
- Only AI formalisms are specially prepared to model formal axioms either as independent components in the ontology or embedded in other components

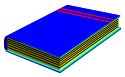
- A domain model is not necessarily an ontology only because it is written in Ontolingua or OWL, for the same reasons that we cannot say that a program is a knowledge-based system because it is written in Prolog

- Although some languages are more appropriate than others to represent ontologies, a model is an ontology only if it is agreed and machine readable

Table of Content

Types of Ontologies
Lassila and McGuiness Classification



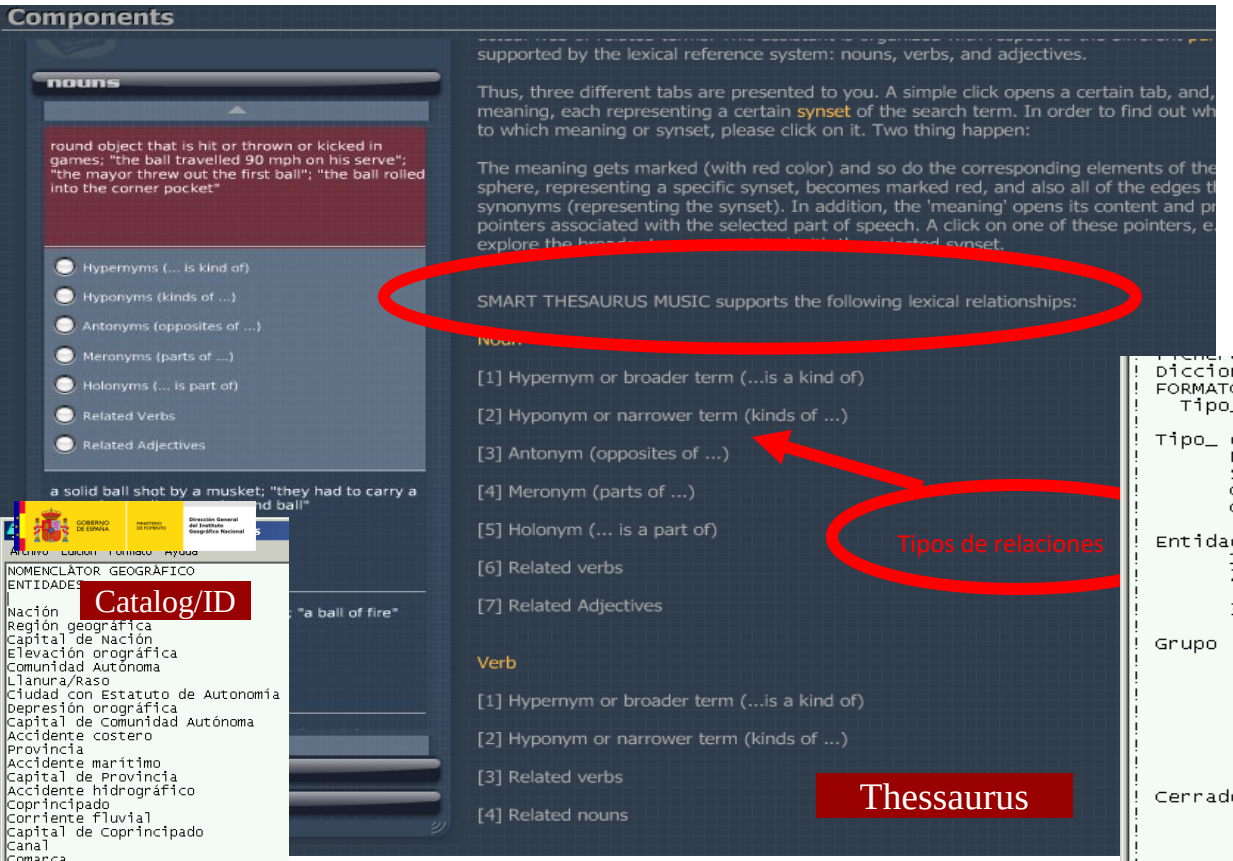


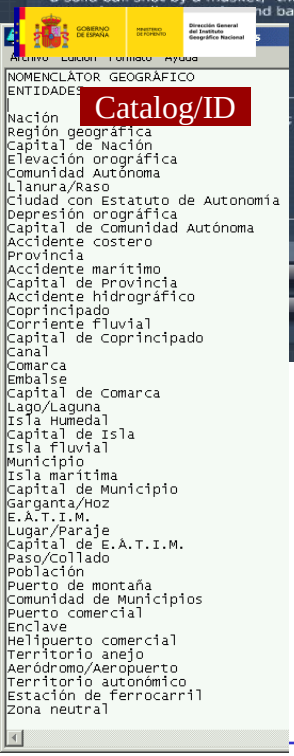
Lassila O, McGuiness D. The Role of Frame-Based Representation on the Semantic Web. Technical Report. Knowledge Systems Laboratory. Stanford University. KSL-01-02. 2001.

Catalog/ID Glossary Thessaurus Informal is-a

Informal is-a

Id	Category Name	Parent
20000	Water area	1
21000	Environmental area	20000
22000	Fishing Statistical area	20000
24020	Jurisdiction area	20000
21001	Inland/marine	21000
21002	Ocean	21000
21003	North/South/Equatorial	21000
21004	Sub Ocean	21000
21005	Large Marine ecosystem	21000



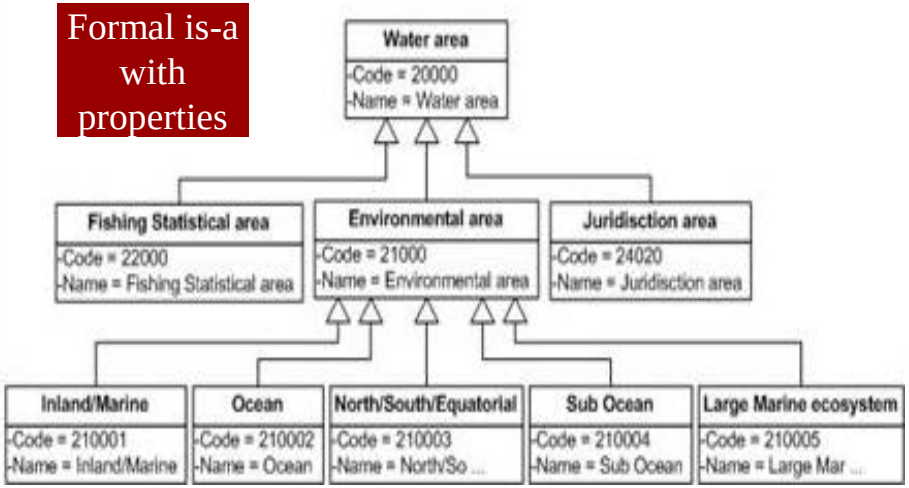



```
! Diccionario de conversión DGN -> EDM.
! FORMATO:
!   Tipo_dgn Entidad Tipo_istram Grupo Código_bcn Cerrado Trato
!
! Tipo_ dgn...NNSCCCGG                          Codigo_bcn...TTGGSS
!       NN   : Nivel elemento                          TT : Tema
!       S    : Estilo linea dgn                        GG : Grupo
!       CCC  : Color  linea dgn                        SS : Subgrupo
!       GG   : Grosor linea dgn
!
! Entidad                                       Tipo_istram....???
!       104 : polilinea
!       203 : célula se convierte a símbolo
!        -1 : célula se explota en sus componentes
!       304 : rótulo
!
! Grupo
!         0 : sin determinar
!         1 : carreteras
!         2 : hidrografía
!         3 : conducciones
!         4 : administrativo
!
!         En textos el grupo corresponde a la fuente Microstatic
!
! Cerrado
!        en lineas                                      en textos
!                1 : perimetral                                 n : altu
!                0 : entidad lineal abierta
!               -1 : cultivo perimetral
!               -2 : cultivo linea abierta
!
! Trato
!   I: Intocable  A: Altimetría  N: No tratar  T: Textos Asociad
!   S: Textos Sueltos  C: Cultivo  F: Solo salida !: Tratar norm
!                                  TTGGSS
02000900    104      1    0    090101    1    !I   Marco de hoja
02300902    104      2    0    100200    0    !    Base Geodésica de M
06003900    104      3    0    025102    0    !    Acantilado
06006900    104      4    0    025302    0    !    Costa rocosa no aca
06009900    104      5    2    037402    1    !    Playa fluvial de gu
06012900    104      6    0    025501    1    !    Lavas. Contorno
06015900    104      7    0    058303    0    !I   Dique de hormigón >
06018900    104      8    0    058304    0    !I   Dique de hormigón <
07013400    104      9    0    058302    0    !I   Dique de tierra
07016400    104     10    0    055401    1    !    Vertedero. Contorno
11003003    104     11    1    062202    0    !    Autopista. Enlace
11012000    104     12    0    056091    1    !I   Patio. Contorno
13003300    104     13    1    060101    0    !    Autopista. Eje
13303300    104     14    1    060131    0    !    Autopista en Contru
14002401    104     15    1    066901    1    !I   Puesto de S.O.S.
14003301    104     16    1    067901    1    !I   Peaje
15003003    104     17    1    062204    0    !    Autovía. Enlace
15003004    104     18    1    060701    0    !    Autovía
```

Informal is-a

Formal is-a | Formal instance | Frames (properties) | Value Restrs. | General Logical constraints | Disjointness, Inverse, part-Of ...

Formal is-a with properties




```
(define-relation connects (?edge ?source ?target)
  "This relation links a source and a target by an edge. The source and
destination are considered as spatial points. The relation has the
following properties: symmetry and irreflexivity."
:def (and (SpatialPoint ?source)
          (SpatialPoint ?target)
          (Edge ?edge))
:axiom-def
((=> (connects ?edge ?source ?target)
     (connects ?edge ?target ?source)) ;symmetry
 (=> (connects ?edge ?source ?target)
     (not (or (part-of ?source ?target) ;irreflexivity
              (part-of ?target ?source))))))
```

General Logical constraints

```
(define-class Travel (?travel)
  "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Template-Facet-Value Cardinality
           arrivalDate Travel 1)
      (Template-Facet-Value Cardinality
           departureDate Travel 1)
      (Template-Facet-Value Maximum-Cardinality
           singleFare Travel 1))
:def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date)
       (singleFare ?travel Number)
       (companyName ?travel String)))
```

Value Restrs.

```
(define-class AmericanAirlinesFlight (?X)
:def (Flight ?X)
:axiom-def
  (Disjoint-Decomposition AmericanAirlinesFlight
          (Setof AA7462 AA2010 AA0488)))

(define-class Location (?X)
:axiom-def
  (Partition Location
      (Setof EuropeanLocation NorthAmericanLocation
          SouthAmericanLocation AsianLocation
          AfricanLocation AustralianLocation
          AntarcticLocation)))
```

Disjointness

Types of Ontologies

Content Ontologies

Domain O.

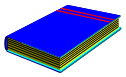
Scalpel, scanner
anesthetize, give birth

Task O.

goal, schedule
to assign, to classify

General/Common O.

Things, Events, Time, Space
Causality, Behavior, Function



Mizoguchi, R. Vanwelkenhuysen, J.; Ikeda, M.
Task Ontology for Reuse of Problem Solving Knowledge.
Towards Very Large Knowledge Bases:
Knowledge Building & Knowledge Sharing.
IOS Press. 1995. 46-59.

Issue of the Conceptualization

Representation O.

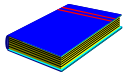
- Conceptualization of KR formalisms

Generic O.

- Reusable across D.

Domain O.

- Reusable



Application O.

- Non reusable
- Usable

Van Heist, G.; Schreiber, T.; Wielinga, B.
Using Explicit Ontologies in KBS
International Journal of Human-Computer Studies.
Vol. 46. (2/3). 183-292. 1997

Table of Content

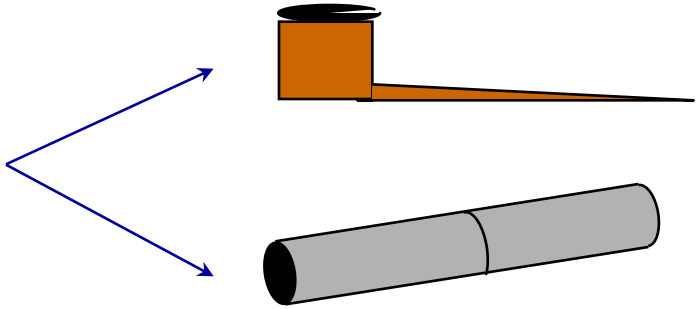
Ontological Commitments

Agreements to use the vocabulary in a coherent and consistent manner (Gruber)

Connection between the ontology vocabulary and the meaning of the terms of such vocabulary

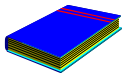
An agent commits (conforms) to an ontology if it "acts" consistently with the definitions

Example: What is a pipe?



9 definitions of the term flight from wordnet

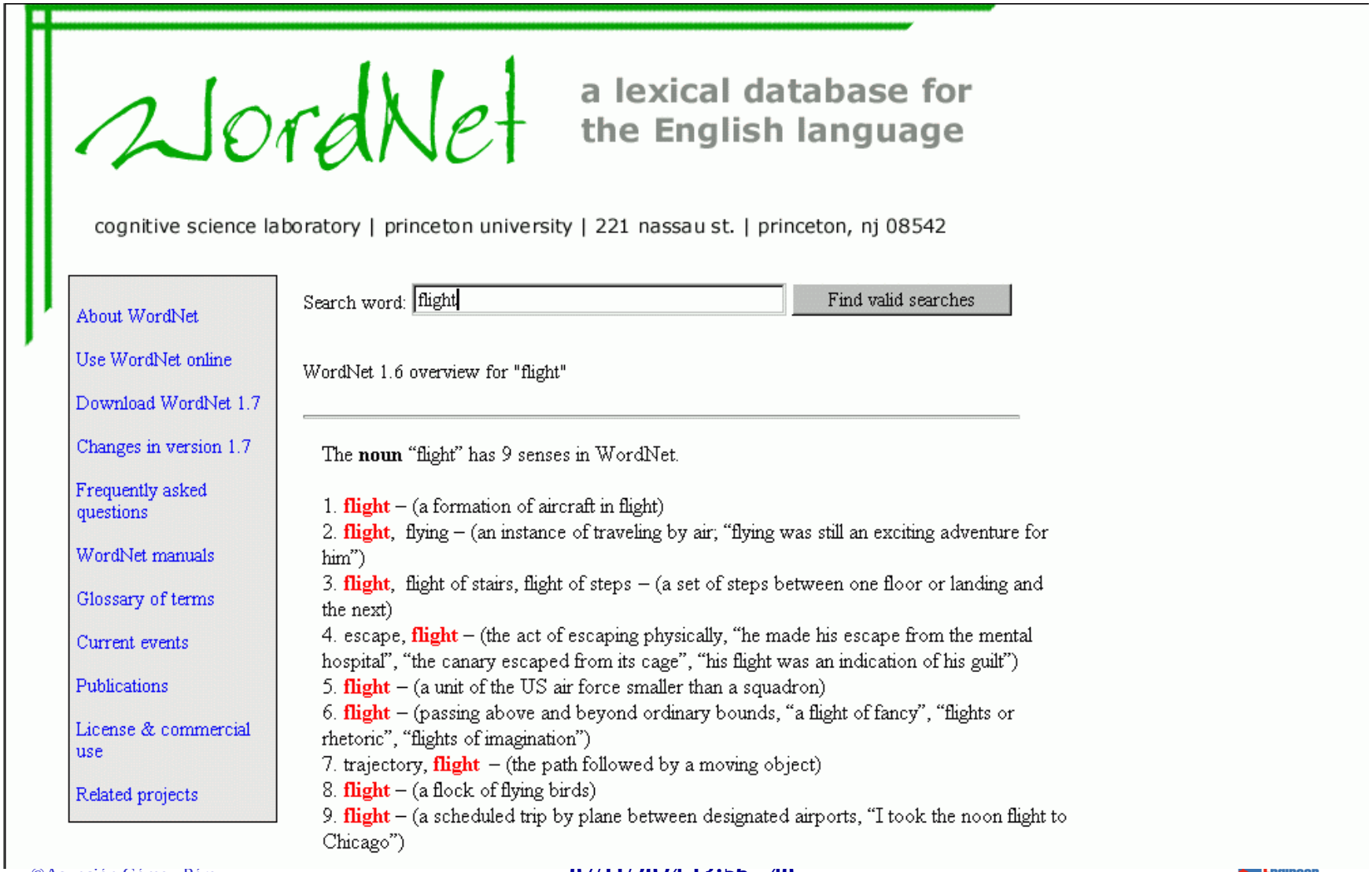
Identification of the ontological commitment



• Gruber, T.; Olsen, G. *An Ontology for Engineering Mathematics.* **Fourth International Conference on Principles of Knowledge Representation and Reasoning**. Ed by Doyle and Torasso. Morgan Kaufmann. 1994. Also as KSL-94-18.

• Guarino, N.; Carrara, M.; Giaretta, P. *Formalizing Ontological Commitments.* **12th National Conference on Artificial Intelligence. AAAI-**94. 1994. 560-567

Ontological Commitments



A formation of aircraft in flight

A scheduled trip by plane between designated airports

Vuelo	Salida		Llegada	
Iberia IB 646	Santiago (SCQ), Santiago, España	06:30	Barajas (MAD), Madrid, España	07:35
Iberia IB 6741	Barajas (MAD), Madrid, España	12:10	Eldorado (BOG), Bogotá, Colombia	16:46
Aces VX 7290	Eldorado (BOG), Bogotá, Colombia	18:00	Rafael Nunez (CTG), Cartagena, Colombia	19:18
Iberia IB 646	Santiago (SCQ), Santiago, España	06:30	Barajas (MAD), Madrid, España	07:35
Iberia IB 6741	Barajas (MAD), Madrid, España	12:10	Eldorado (BOG), Bogotá, Colombia	16:46
Aces VX 7290	Eldorado (BOG), Bogotá, Colombia	18:00	Rafael Nunez (CTG), Cartagena, Colombia	19:18

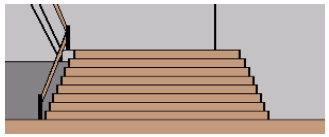
An instance of traveling by air

A flock of flying birds

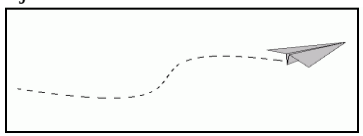
OC_1 OC_2 OC_3 OC_4 OC_5 OC_6 OC_7 OC_8 OC_9

flight

A set or steps between one floor or landing for him



The path followed by a moving object



The act of escaping physically

A unit of the US air force smaller than a squadron

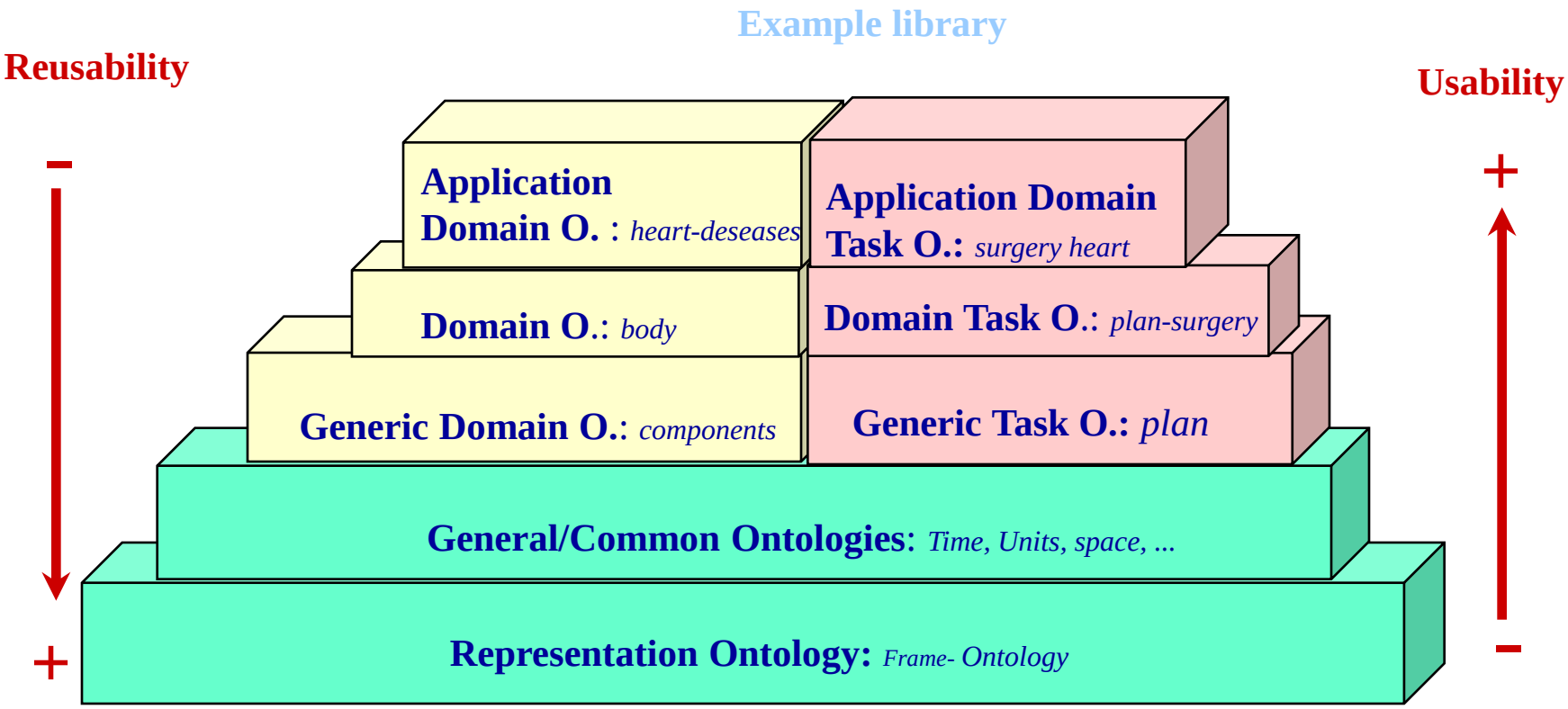
Passing above and beyond ordinary bounds

(**define-class** Flight (?X)
"A journey by plane"
:axiom-def
(**and** (**Subclass-Of** Flight Travel)
(**Template-Facet-Value** Cardinality
flightNumber Flight 1))
:class-slots ((transportMeans "plane")))

flight

Table of Content

Libraries of Ontologies



http://delicias.dia.fi.upm.es/mirror-server/ont-serv.html

Modular approach for ontology construction

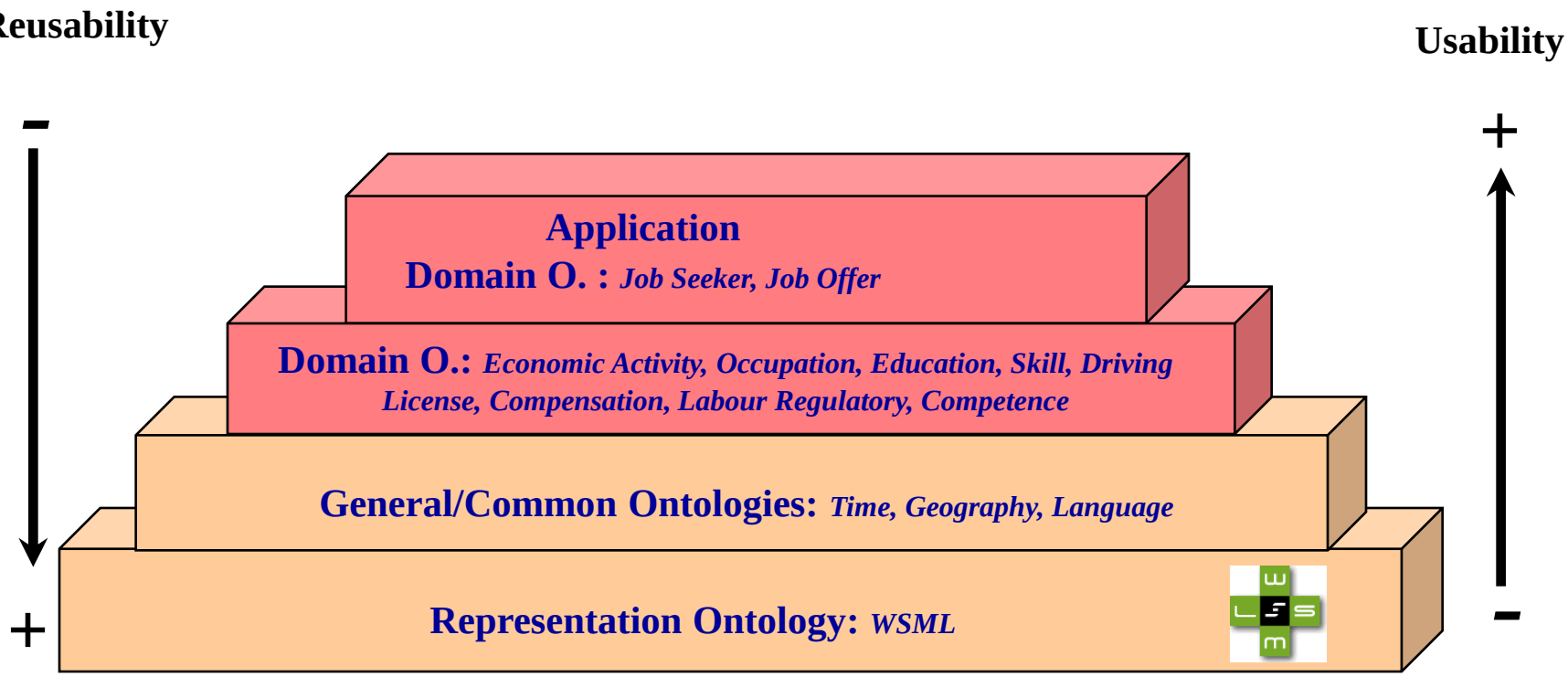
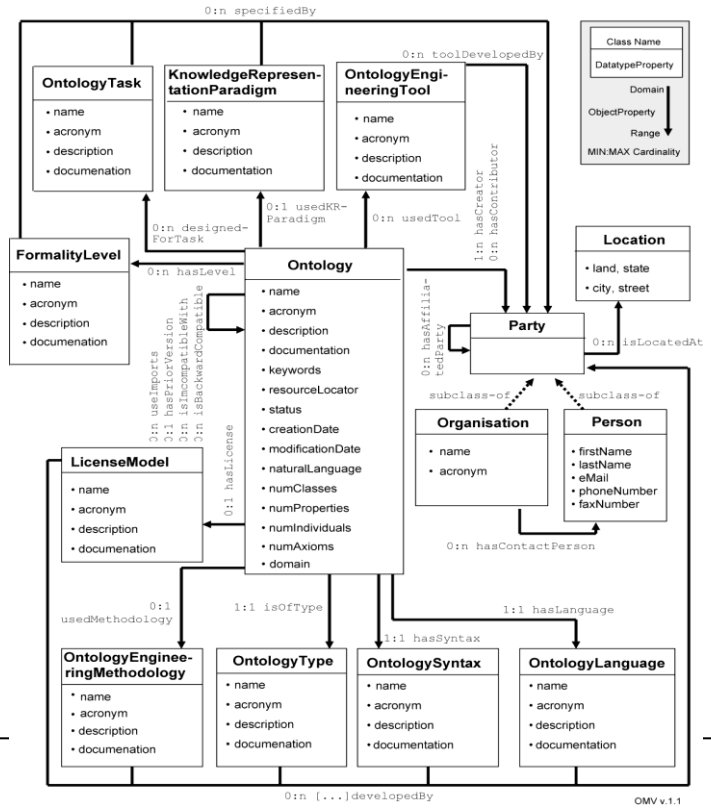


Table of Content

Searching Ontologies

- OMV: Ontology Metadata Vocabulary



- Ontology registries

CYSTER

Onthology.org

knowledge zone one stop shop for ontologies

Swoogle semantic web search 2006

watson exploring the semantic web

Searching ontologies: Obtain the set of candidate ontologies using Oyster

O. Searching

O. Assessment

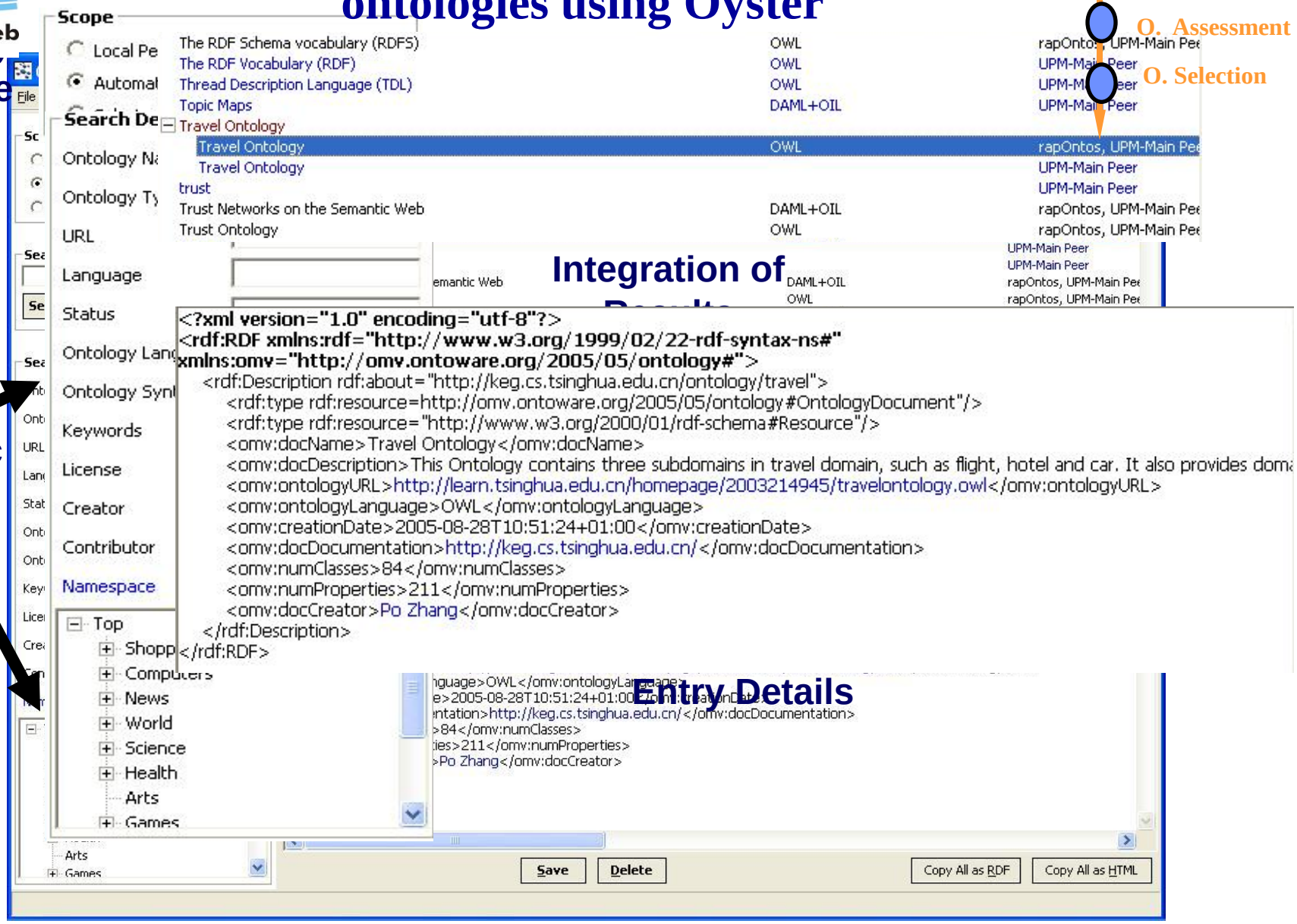
O. Selection

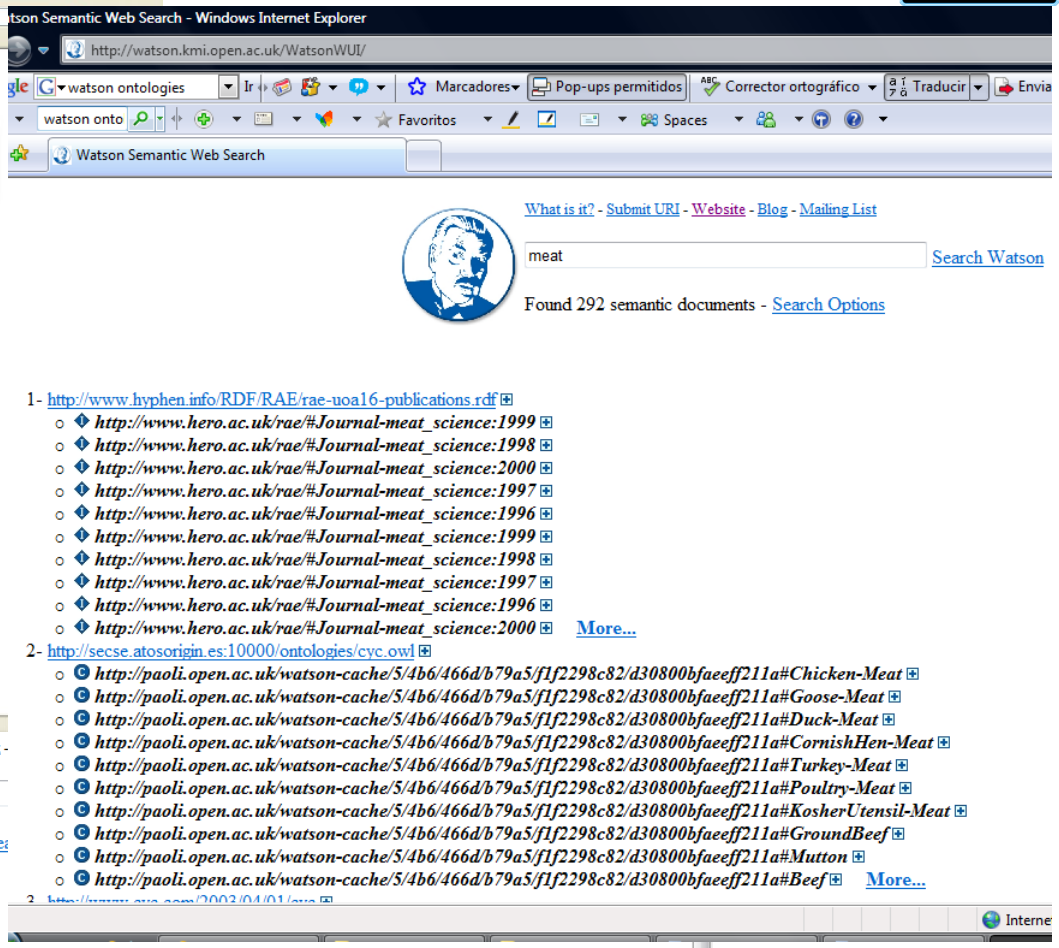
Scope

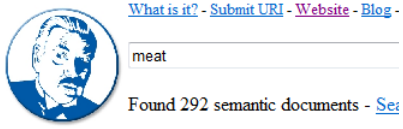
Semantic Search

Integration of Results

Entry Details







11- http://neuroscientific.net/bio-zen.owl
- http://www.loa-cnr.it/ontologies/DOLCE-Lite#amount-of-matter

12- http://www.hyphen.info/RDF/RAE/rae-uoa44-publications.rdf
- http://www.hero.ac.uk/rae/#H-0153:UoA-44:Id-219109:PubNum-4

13- http://ontologyportal.org/translations/SUMO.owl.txt
- http://kmi-web05.open.ac.uk:81/cache/0/339/c2ff/21d76/1013cd189c/557c6d296bdc6957c#Meat
- http://kmi-web05.open.ac.uk:81/cache/0/339/c2ff/21d76/1013cd189c/557c6d296bdc6957c#Beverage

14- http://reliant.teknowledge.com/DAML/SUMO.daml
- http://kmi-web05.open.ac.uk:81/cache/1/f5d/857a/f0cae/8b96c6b3ac/1e19b0c5cf7f2a849#Meat
- http://kmi-web05.open.ac.uk:81/cache/1/f5d/857a/f0cae/8b96c6b3ac/1e19b0c5cf7f2a849#Beverage

15- http://reliant.teknowledge.com/DAML/SUMO.owl
- http://kmi-web05.open.ac.uk:81/cache/c/7a4/11d2/d58c5/c6f902b40b/2bdd54d158be86601#Meat
- http://kmi-web05.open.ac.uk:81/cache/c/7a4/11d2/d58c5/c6f902b40b/2bdd54d158be86601#Beverage

16- http://www.ling.helsinki.fi/kit/2004k/ctl310semw/OWL/wine.daml.rdf
- http://potato.cs.man.ac.uk/ontologies/booze#MEAT
- http://potato.cs.man.ac.uk/ontologies/booze#RED-MEAT
  - *Label:* RED-MEAT
  - *Comment:*
- http://potato.cs.man.ac.uk/ontologies/booze#LIGHT-MEAT-FOWL

Watson plug-in in the NeOn Toolkit

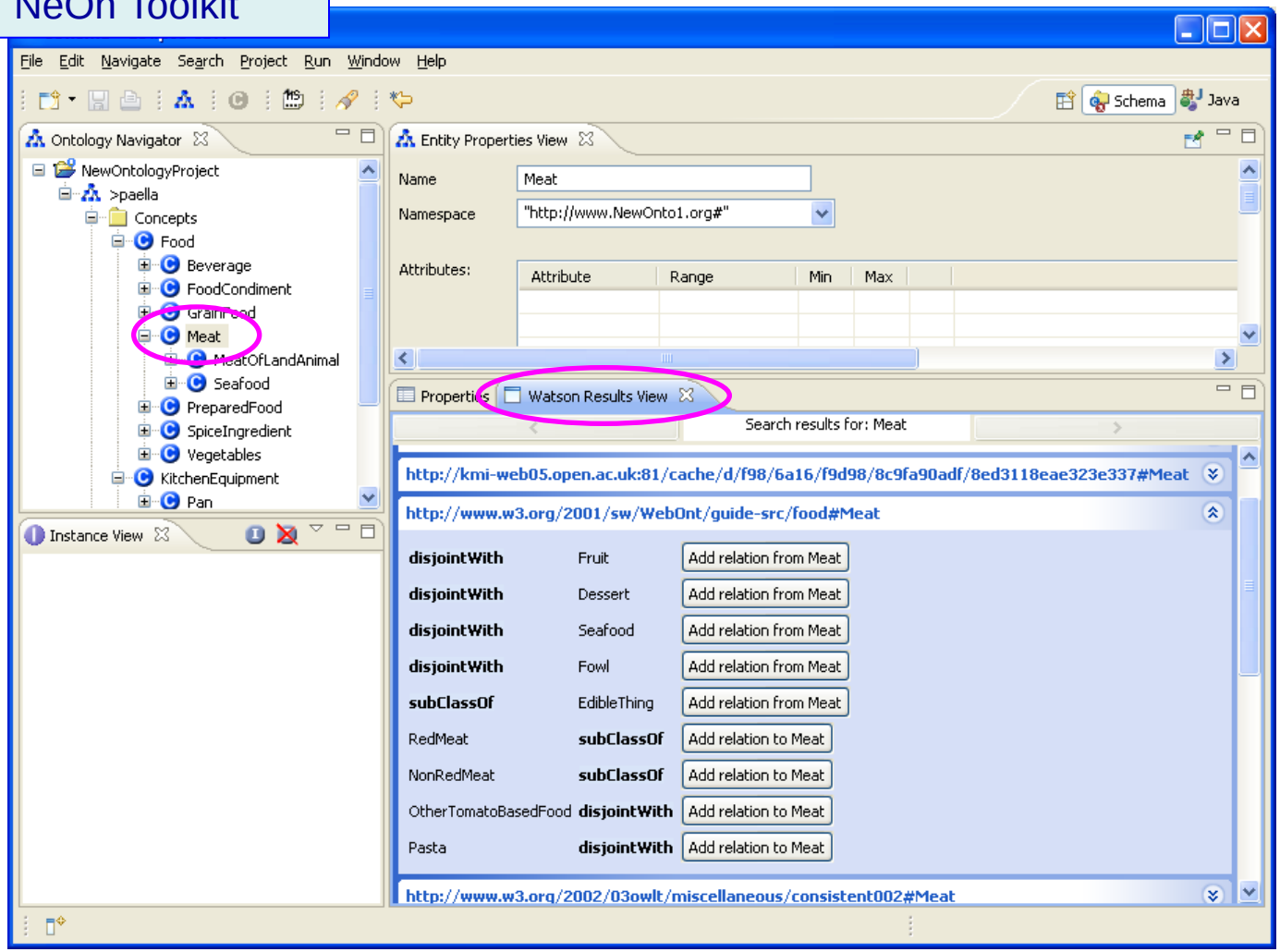


Table of Content

Knowledge Representation Ontologies

Knowledge Representation (KR) ontologies capture the representation primitives used to formalize knowledge under a given KR paradigm.

•The Frame Ontology and the OKBC Ontology
(http://ontolingua.stanford.edu**)**

•Gruber TR (1993a) *A translation approach to portable ontology specification*. Knowledge Acquisition 5(2):199–220

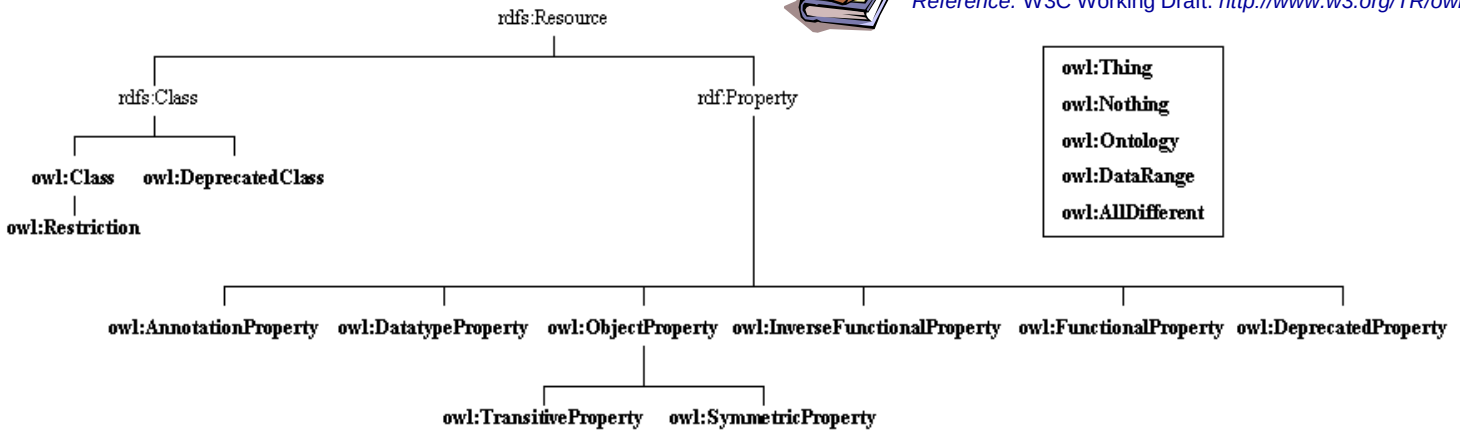
•Chaudhri VK, Farquhar A, Fikes R, Karp PD, Rice JP (1998) *Open Knowledge Base Connectivity 2.0.3*. Technical Report. *http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf*

•RDF and RDF Schema knowledge representation ontologies
(http://www.w3.org/1999/02/22-rdf-syntax-ns
http://www.w3.org/2000/01/rdf-schema**)**

Lassila O, Swick R (1999) *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation. *http://www.w3.org/TR/REC-rdf-syntax/*

•OWL knowledge representation ontology
(http://www.w3.org/2002/07/owl**)**

Dean M, Schreiber G (2003) *OWL Web Ontology Language Reference*. W3C Working Draft. *http://www.w3.org/TR/owl-ref/*



RDF and RDF Schema knowledge representation ontologies

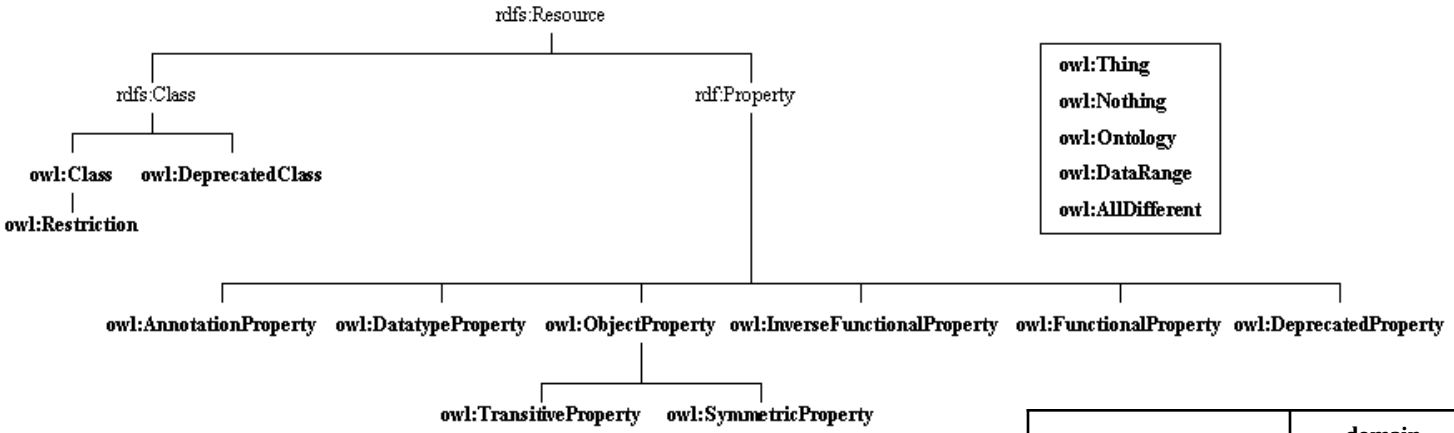


Table: Property descriptions of the RDF(S) KR ontology.

Property name	domain	range
rdf:type	rdfs:Resource	rdfs:Class
rdf:subject	rdf:Statement	rdfs:Resource
rdf:predicate	rdf:Statement	rdf:Property
rdf:object	rdf:Statement	rdfs:Resource
rdf:value	rdfs:Resource	rdfs:Resource
rdf:first	rdf:List	rdfs:Resource
rdf:rest	rdf:List	rdf:List
rdfs:subClassOf	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	rdf:Property	rdf:Property
rdfs:comment	rdfs:Resource	rdfs:Literal
rdfs:label	rdfs:Resource	rdfs:Literal
rdfs:seeAlso	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	rdfs:Resource	rdfs:Resource
rdfs:member	rdfs:Resource	rdfs:Resource
rdfs:domain	rdf:Property	rdfs:Class
rdfs:range	rdf:Property	rdfs:Class

Class hierarchy (23 classes defined):

- Binary-Relation
  - Antisymmetric-Relation
    - Asymmetric-Relation
    - Partial-Order-Relation
      - Total-Order-Relation
  - Irreflexive-Relation
    - Asymmetric-Relation
  - Many-To-Many-Relation
  - Many-To-One-Relation
  - One-To-Many-Relation
  - Reflexive-Relation
    - Equivalence-Relation
    - Partial-Order-Relation ...
  - Symmetric-Relation
    - Equivalence-Relation
  - Transitive-Relation
    - Equivalence-Relation
    - Partial-Order-Relation ...
  - Weak-Transitive-Relation
- Class
  - Root_Class
- Class-Partition
- Function
  - Many-To-One-Relation
- Individual-Thing
- Named-Axiom
- One-To-One-Relation
- Relation
- Unary-Relation

31 relations defined:

- Alias
- Composition-Of
- Default-Facet-Value
- Default-Slot-Value
- Default-Template-Facet-Value
- Default-Template-Slot-Value
- Disjoint-Decomposition
- Documentation
- Domain-Of
- Exhaustive-Decomposition
- Has-Author
- Has-Instance
- Has-Source
- Has-Subdefinition
- Has-Subrelation
- Inherited-Facet-Value
- Inherited-Slot-Value
- Nth-Argument-Name
- Nth-Domain
- Nth-Domain-Subclass-Of
- Obsolete-Same-Values
- Obsolete-Value-Type
- Onto
- Partition
- Range-Of
- Range-Subclass-Of
- Related-Axioms
- Single-Valued-Slot
- Slot-Documentation
- Subrelation-Of
- Total-On

13 functions defined:

- All-Instances
- All-Values
- Arity
- Compose
- Domain-Name
- Exact-Domain
- Exact-Range
- Function-Arity
- Obsolete-Slot-Cardinality
- Projection
- Range-Name
- Relation-Universe
- Subdefinition-Of

Definition of the relation SUBCLASS-OF in the Frame Ontology

(define-relation Subclass-Of (?child-class ?parent-class)

"Class C is a subclass of parent class P if and only if every instance of C is also an instance of P. A class may have multiple superclasses and subclasses. Subclass-of is transitive: if (subclass-of C1 C2) and (subclass-of C2 C3) then (subclass-of C1 C3). Object-centered systems sometimes distinguish between a subclass-of relationship that is asserted and one that is inferred. For example, (subclass-of C1 C3) might be inferred from asserting (subclass-of C1 C2) and (subclass-of C2 C3)..."

:iff-def
(and (Class ?parent-class)
(Class ?child-class)
(forall (?instance)
(=> (Instance-Of ?instance ?child-class)
(Instance-Of ?instance ?parent-class))))

:axiom-constraints
(Transitive-Relation Subclass-Of)
:issues
((:see-also direct-subclass-of)
(:see-also "In CycL, subclass-of is called #%allGenls because it is a slot from a collection to all of its generalizations (superclasses)."
"In the KL-ONE literature, subclass relationships are also called subsumption relationships and ISA is sometimes used for subclass-of.")
("Why is it called Subclass-of instead of subclass or superclass?"
"Because the latter are ambiguous about the order of their arguments. We are following the naming convention that a binary relationship is read as an English sentence `Domain-element Relation-name Range-value'. Thus, `person subclass-of animal' rather than `person superclass animal'.")))

http://www-ksl.stanford.edu

Top-level Ontologies

Top-level Ontologies or Upper-level Ontologies describe very general concepts and provide general notions under which all root terms in existing ontologies should be linked.

•Top-level ontologies of universals and particulars (http://webode.dia.fi.upm.es/)

•Guarino N, Welty C (2000) *A Formal Ontology of Properties*. In: Dieng R, Corby O (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 97–112

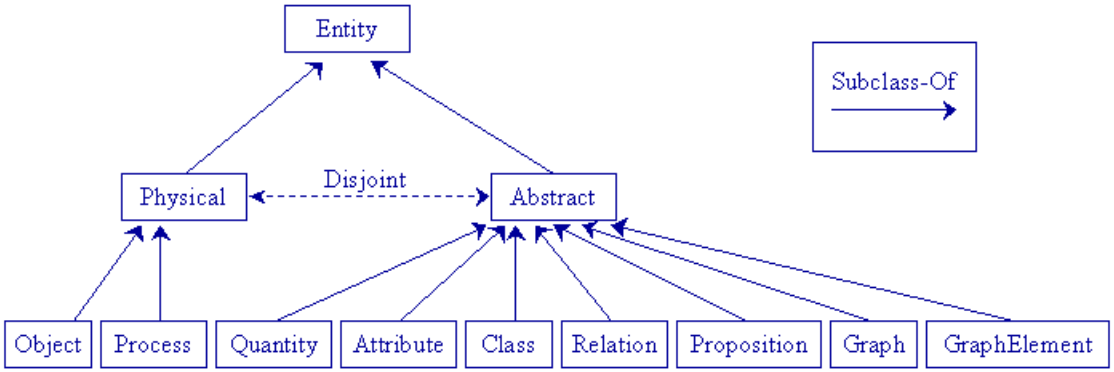
•Gangemi A, Guarino N, Oltramari A (2001) *Conceptual analysis of lexical taxonomies: the case of Wordnet top-level*. In: Smith B, Welty C (eds) International Conference on Formal Ontology in Information Systems (FOIS'01). Ogunquit, Maine. ACM Press, New York, pp 3–15

•Sowa's top-level ontology (http://www.jfsowa.com/ontology/toplevel.htm)

Sowa JF (1999) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, California

•Cyc's upper ontology (http://www.cyc.com/cyc-2-1/cover.html)

Lenat DB, Guha RV (1990) *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Boston, Massachusetts

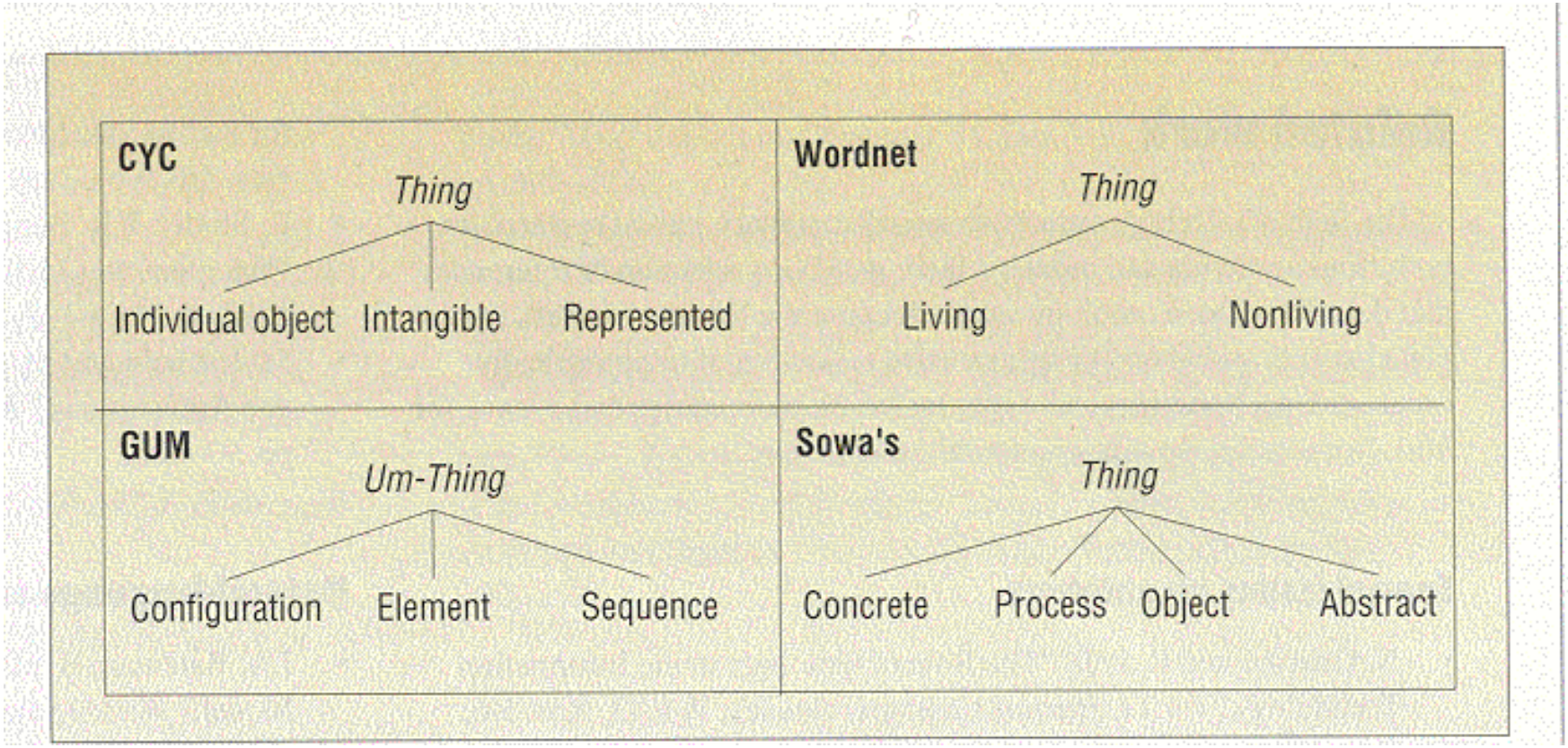


•The Standard Upper Ontology (SUO) (http://suo.ieee.org/)

Pease RA, Niles I (2002) *IEEE Standard Upper Ontology: A Progress Report*. The Knowledge Engineering Review 17(1):65–70

One Unique Top-Level Ontology?

Various proposals



Linguistic Ontologies

•WordNet (http://www.hum.uva.nl/~ewn/gwa.htm)

•Miller GA (1995) *WordNet: a lexical database for English*. Communications of the ACM 38(11):39–41

•Miller GA, Beckwith R, Fellbaum C, Gross D, Miller K (1990) *Introduction to WordNet: An on-line lexical database*. International Journal of Lexicography 3(4):235–244

•EuroWordNet (http://www.hum.uva.nl/~ewn/)

•Vossen P (ed) (1999) *EuroWordNet General Document. Version 3*. http://www.hum.uva.nl/ewn/

•Vossen P (ed) (1998) *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Dordrecht, The Netherlands

•The Generalized Upper Model
(http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html)

Bateman JA, Fabris G, Magnini B (1995) *The Generalized Upper Model Knowledge Base: Organization and Use*. In: Mars N (ed) Second International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KBKS '95). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, pp 60–72

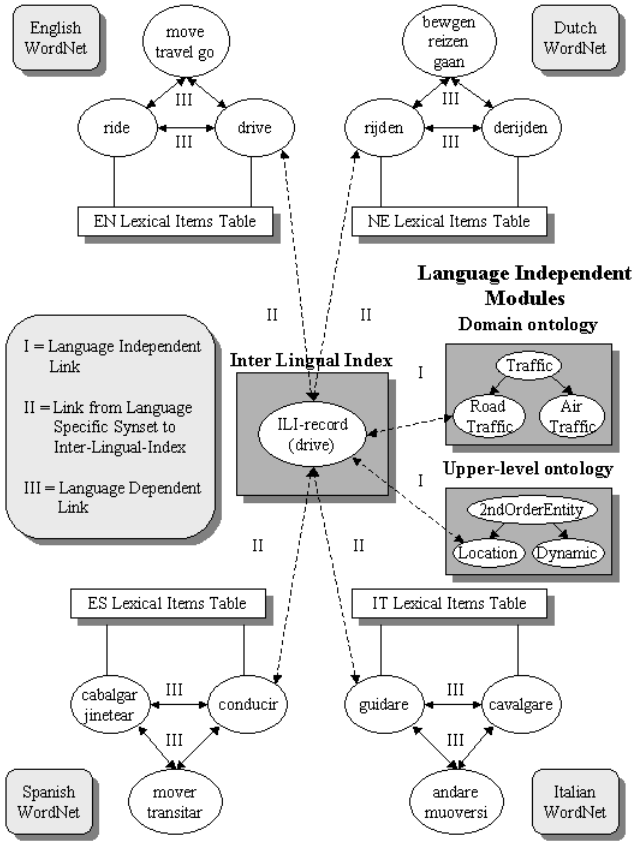
•The Mikrokosmos ontology (http://crl.nmsu.edu/mikro [user and password are required])

•Mahesh K (1996) *Ontology development for machine translation: Ideology and Methodology*. Technical Report MCCS-96-292. Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico. http://citeseer.nj.nec.com/mahesh96ontology.html

•Mahesh K, Nirenburg S (1995) *Semantic classification for practical natural language processing*. In: Schwartz RP, Kwasnik BH, Beghtol C, Smith PJ, Jacob E (eds) 6th ASIS SIG/CR Classification Research Workshop: An Interdisciplinary Meeting. Chicago, Illinois, pp 79–94

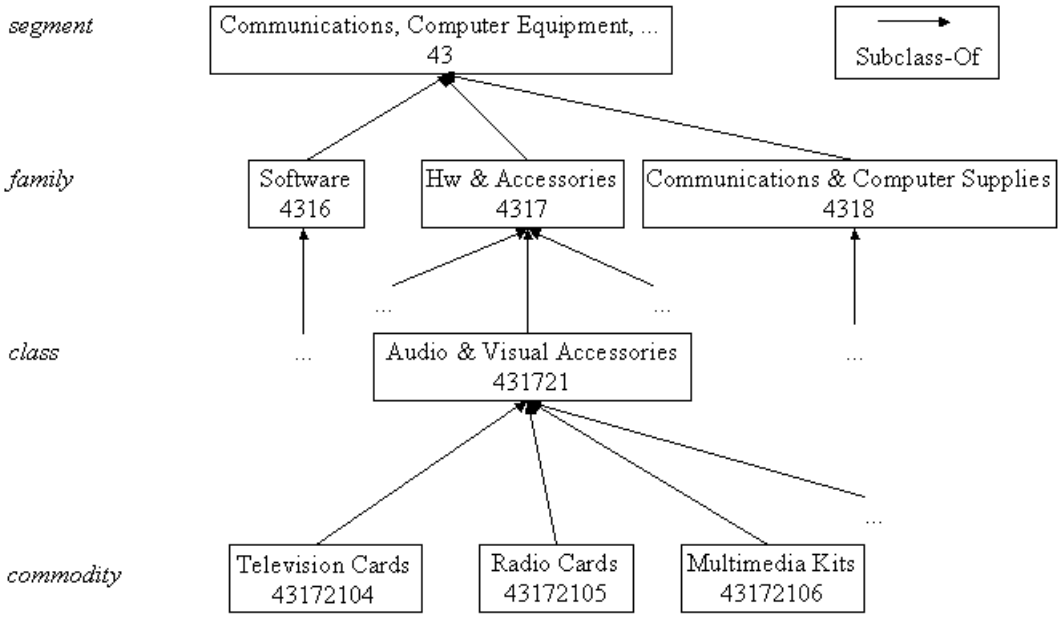
•SENSUS (http://www.isi.edu/natural-language/projects/ONTOLOGIES.html)

Swartout B, Ramesh P, Knight K, Russ T (1997) *Toward Distributed Use of Large-Scale Ontologies*. In: Farquhar A, Gruninger M, Gómez-Pérez A, Uschold M, van der Vet P (eds) AAAI'97 Spring Symposium on Ontological Engineering. Stanford University, California, pp 138–148



Domain Ontologies: e-Commerce Ontologies

- **The United Nations Standard Products and Services Codes (UNSPSC)**
  (http://www.unspsc.org/)
- **NAICS (North American Industry Classification System)**
  (http://www.census.gov/epcd/www/naics.html)
- **SCTG (Standard Classification of Transported Goods)**
  (http://www.statcan.ca/english/Subjects/Standard/sctg/sctg-menu.htm)
- **E-cl@ss**
  (http://www.eclass.de/)
- **RosettaNet**
  (http://www.rosettanet.org)



Domain Ontologies: Medical Ontologies

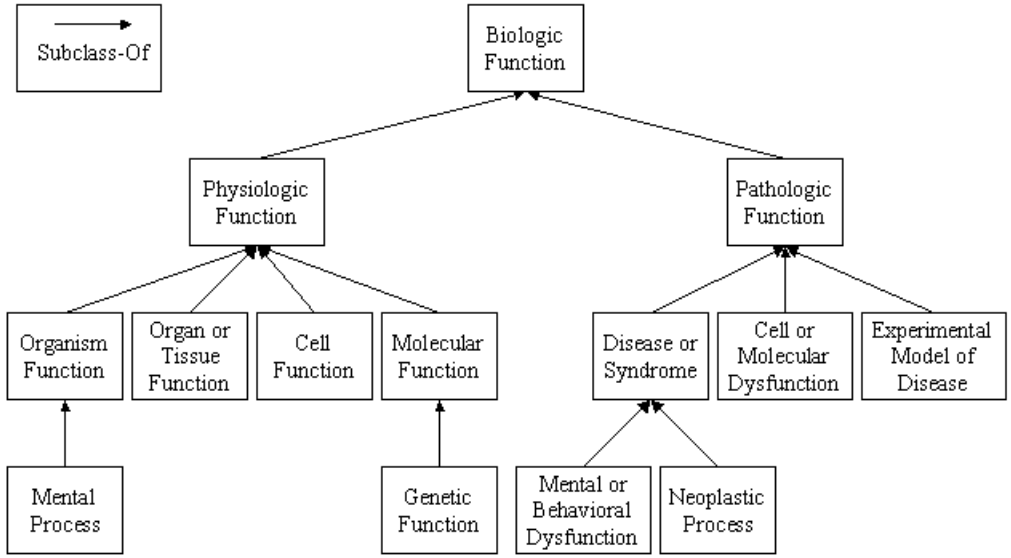
- **GALEN** **http://www.co-ode.org/galen/**

Rector AL, Bechhofer S, Goble CA, Horrocks I, Nowlan WA, Solomon WD (1997) *The GRAIL concept modelling language for medical terminology*. Artificial Intelligence in Medicine 9:139–171

- **UMLS (Unified Medical Language System)**
(http://www.nih.gov/research/umls/)

- **ON9** (http://saussure.irmkant.rm.cnr.it/ON9/index.html)

Gangemi A, Pisanelli DM, Steve G (1998) *Some Requirements and Experiences in Engineering Terminological Ontologies over the WWW*. In: Gaines BR, Musen MA (eds) 11th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'98). Banff, Canada, SHARE10:1–20



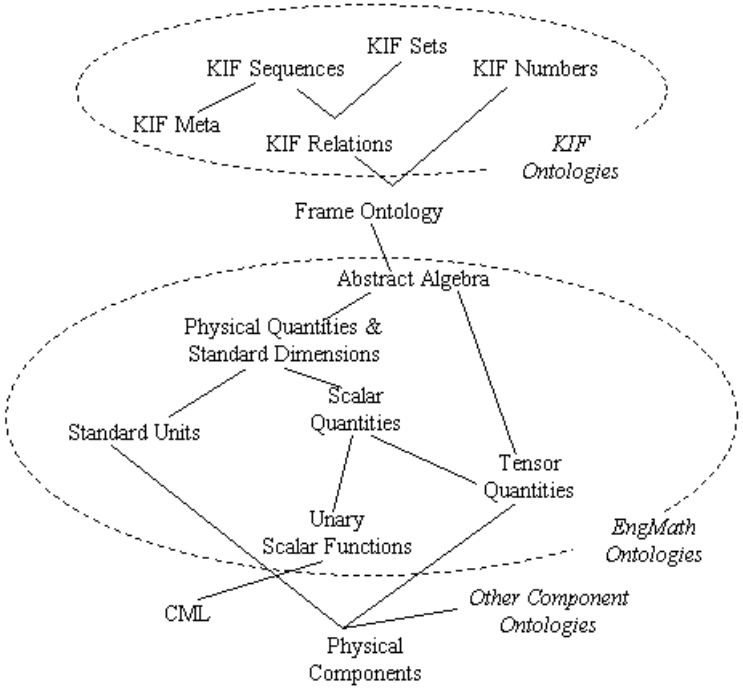
Domain Ontologies: Engineering Ontologies

- **EngMath**

Gruber TR, Olsen G (1994) *An ontology for Engineering Mathematics*. In: Doyle J, Torass Sandewall E (eds) Fourth International Conference on Principles of Knowledge Representation and Reasoning. Bonn, Germany. Morgan Kaufmann Publishers, San Francisco, California, pp 258–269

- **PhysSys**

Borst WN (1997) *Construction of Engineering Ontologies*. Centre for Telematica and Information Technology, University of Tweenty. Enschede, The Netherlands



Domain Ontologies: Enterprise Ontologies

- **Enterprise Ontology** (http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html)

Uschold M, King M, Moralee S, Zorgios Y (1998) *The Enterprise Ontology*. The Knowled Engineering Review 13(1):31–89

- **TOVE** (http://www.eil.utoronto.ca/tove/toveont.html)

Fox MS (1992) *The TOVE Project: A Common-sense Model of the Enterprise*. In: Belli F, Radermacher FJ (eds) Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. (Lecture Notes in Artificial Intelligence LNAI 604) Springer-Verlag, Berlin, Germany, pp 25–34

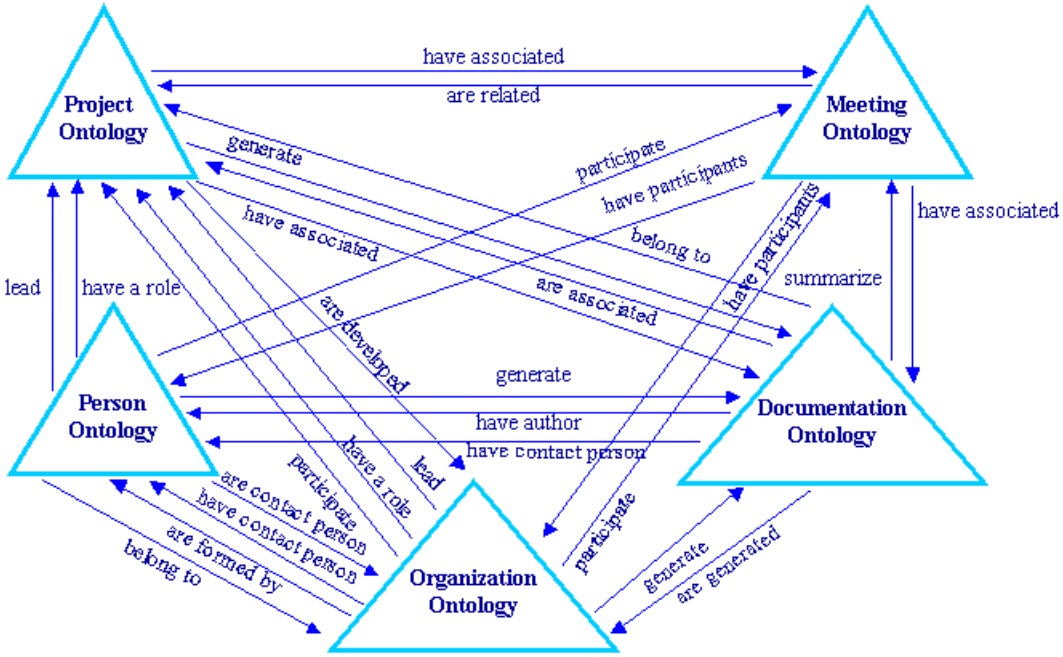
- Enterprise Ontologies
  - Enterprise Design Ontology
  - Project Ontology
  - Material Flow Ontology
  - Business Process Ontology
- Derivative Ontologies
  - Transportation Ontology
  - Inventory Ontology
  - Quality Ontology
  - Product Design Ontology
  - Goals Ontology
  - Scheduling Ontology
  - Operating Strategies Ontology
  - Product Requirements Ontology
  - Information Resource Ontology
  - Intended Action Ontology
  - Electro Mechanical Product Ontology
- Core Ontologies
  - Product Ontology
  - Service Ontology
  - Activity Ontology
  - Organization Ontology
  - Resource Ontology

Domain Ontologies: Knowledge Management Ontologies

- **(KA)^2 ontologies** (http://ka2portal.aifb.uni-karlsruhe.de)

Decker S, Erdmann M, Fensel D, Studer R (1999) *Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information*. In: Meersman R, Tari Z, Stevens S (eds) Semantic Issues in Multimedia Systems (DS-8), Rotorua, New Zealand. Kluwer Academic Publisher, Boston, Massachusetts. pp 351–369

- **R&D projects** (http://www.esperonto.net)



Linked data ontologies

http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/#whichvocabs

4. Which vocabularies should I use to represent information

In order to make it as easy as possible for client applications to process your data, you should reuse terms only define new terms yourself if you can not find required terms in existing vocabularies.

4.1 Reusing existing terms

A set of well-known vocabularies has evolved in the Semantic Web community. Please check whether you before defining any new terms:

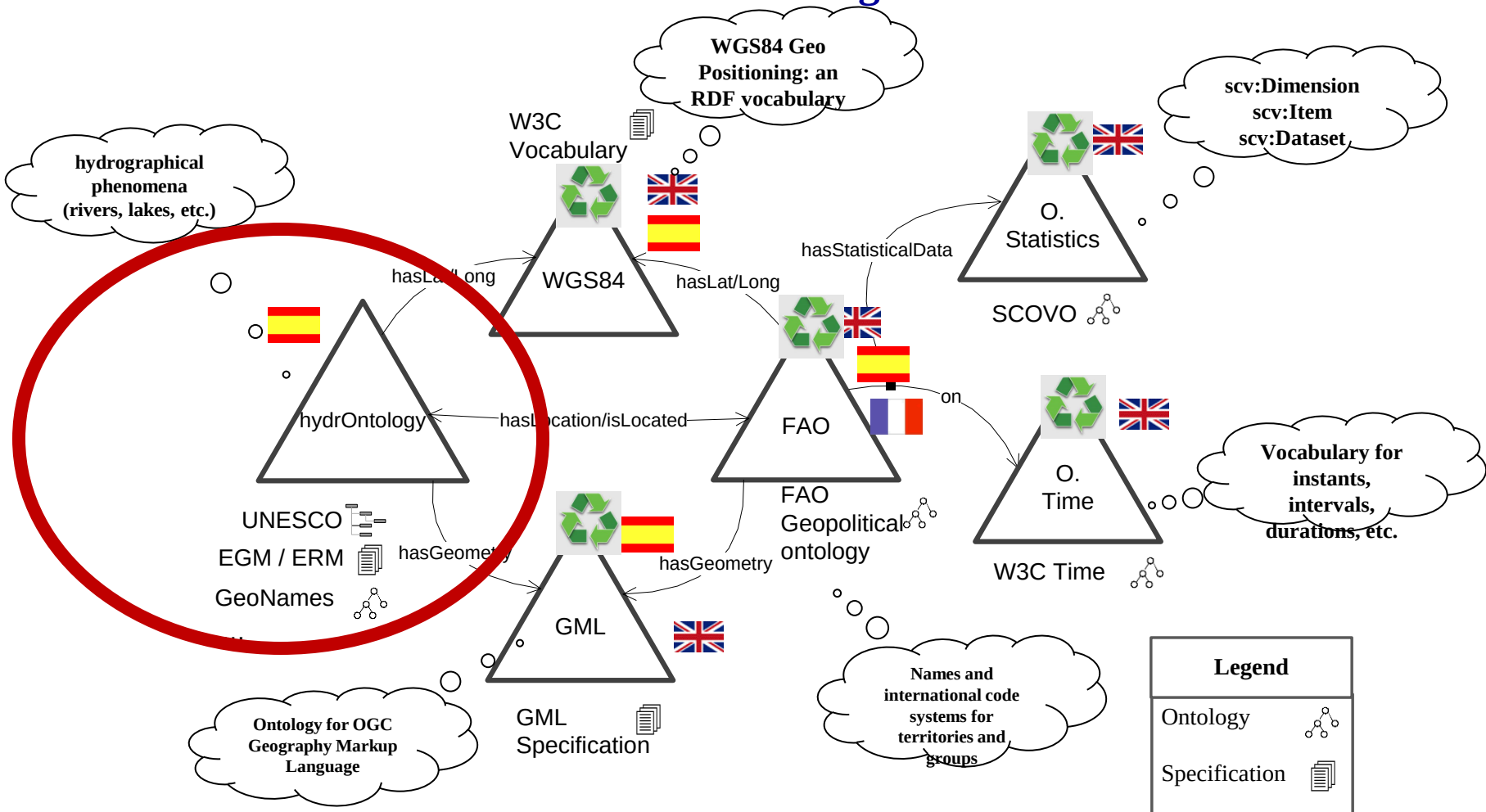
- Friend-of-a-Friend (FOAF), vocabulary for describing people.
- Dublin Core (DC) defines general metadata attributes. See also their new domains and ranges draf
- Semantically-Interlinked Online Communities (SIOC), vocabulary for representing online communitie
- Description of a Project (DOAP), vocabulary for describing projects.
- Simple Knowledge Organization System (SKOS), vocabulary for representing taxonomies and loos
- Music Ontology provides terms for describing artists, albums and tracks.
- Review Vocabulary, vocabulary for representing reviews.
- Creative Commons (CC), vocabulary for describing license terms.

This is not enough

Linked data ontologies

- Features
  - Lightweight :
    - Taxonomies and a few properties
  - Consensuated vocabularies
    - To avoid the mapping problems
  - Multilingual
    - Linked data are multilingual
- The NeOn methodology can help to
  - Re-enginer Non ontological resources into ontologies
    - Pros: use domain terminology already consensuated by domain experts
  - Withdraw in heavyweight ontologies those features that you don’t need
  - Reuse existing vocabularies

Geolinked data ontologies

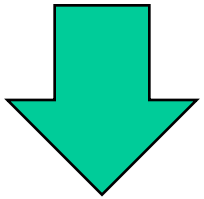


Classes	33	33
Object Properties	44	44
Data Properties	318	318

Following the INSPIRE (INfrastructure for SPatial InfoRmation in Europe) recommendation. hydrOntology,SCOVO, FAO Geopolitcal, WGS84, GML, and Time

What is an Ontology?

Shared understanding of a domain



Repository of vocabulary

- Formal definitions
- Informal definitions