Ontologies

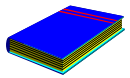
Asunción Gómez-Pérez, Mari Carmen Suárez-Figueroa
{asun, mcsuarez}@fi.upm.es
http://www.oeg-upm.net

Omtological Engineering Group
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Table of Content

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.

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,

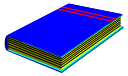
and the tool suites

and languages that support them.

Table of Content

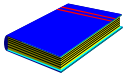
Definitions of Ontologies (I)

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.

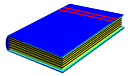
2. "An ontology is an explicit specification of a conceptualization"



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

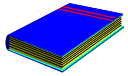
Definitions of Ontologies (II)

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)

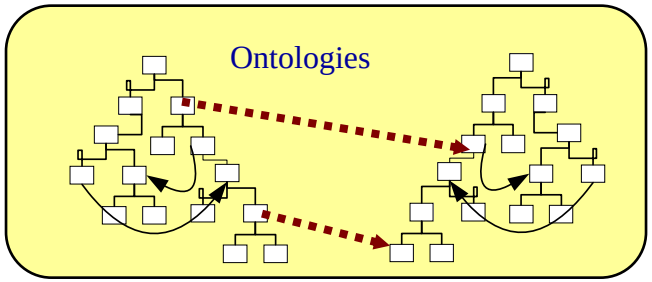
5. "An ontology is a formal, explicit specification of a shared conceptualization"

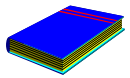
Machine-readable

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)

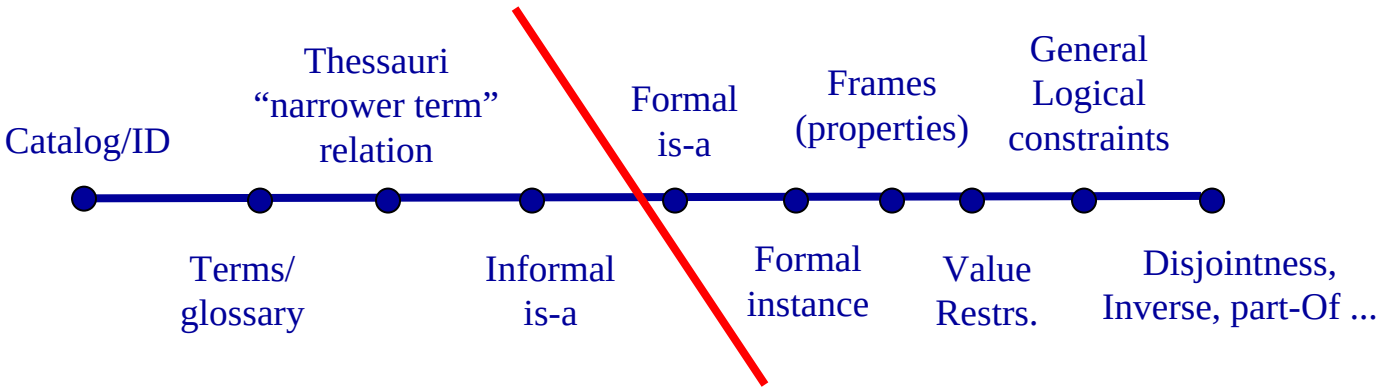
Lightweight Ontologies :

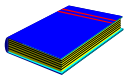
- **Include concepts with properties and taxonomies**
- **Do not include axioms and constraints**

Heavyweight Ontologies :

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

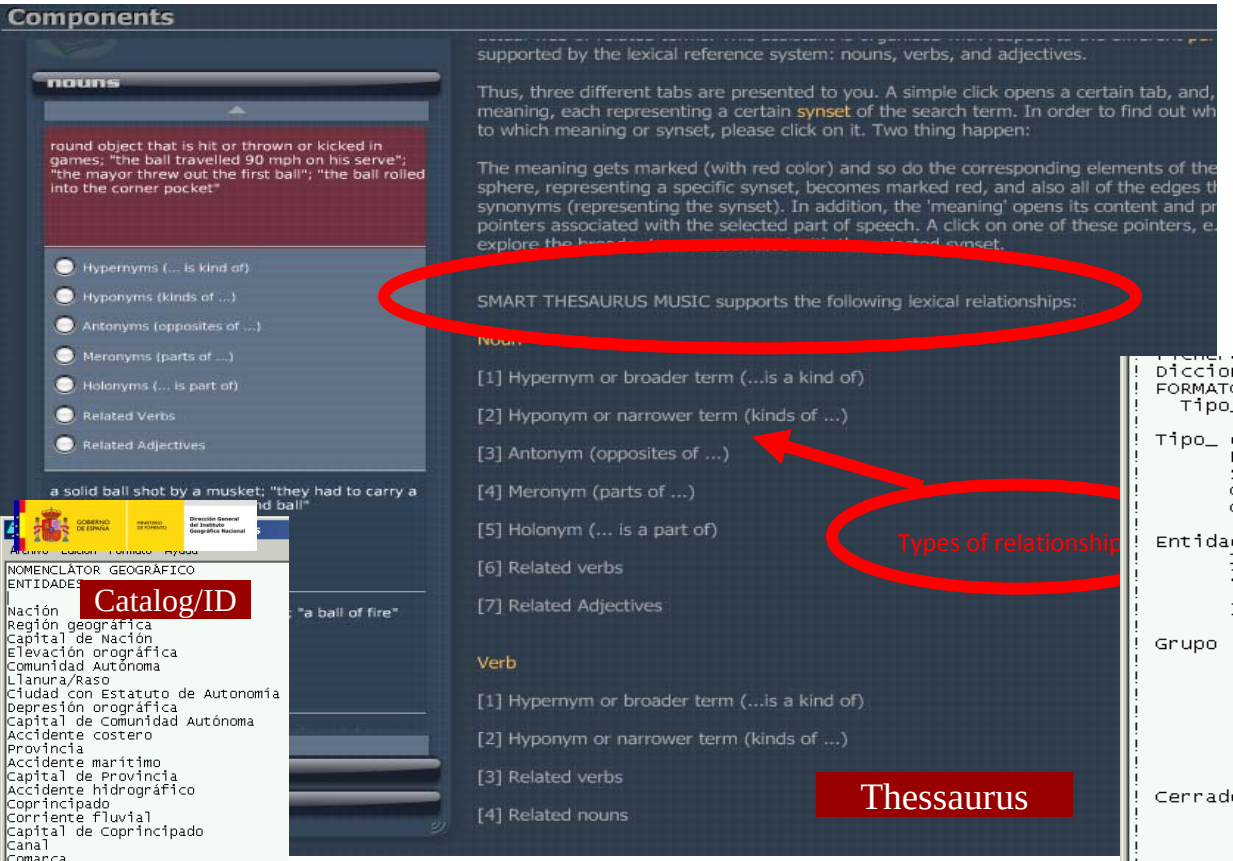
Types of Ontologies
Lassila and McGuiness Classification





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



Catalog/ID

```
NOMENCLÀTOR GEOGRÀFICO
ENTIDADES
Nación
Región geográfica
Capital de Nación
Elevación orográfica
Comunidad Autónoma
Llanura/Raso
Ciudad con Estatuto de Autonomía
Depresión orográfica
Capital de Comunidad Autónoma
Accidente costero
Provincia
Accidente marítimo
Capital de Provincia
Accidente hidrográfico
Coprincipado
Corriente fluvial
Capital de Coprincipado
Canal
Comarca
Embalse
Capital de Comarca
Lago/Laguna
Isla Humedal
Capital de Isla
Isla fluvial
Municipio
Isla marítima
Capital de Municipio
Garganta/Hoz
E.A.T.I.M.
Lugar/Paraje
Capital de E.A.T.I.M.
Paso/Collado
Población
Puerto de montaña
Comunidad de Municipios
Puerto comercial
Enclave
Helipuerto comercial
Territorio anejo
Aeródromo/Aeropuerto
Territorio autonómico
Estación de ferrocarril
Zona neutral
```

Informal is-a

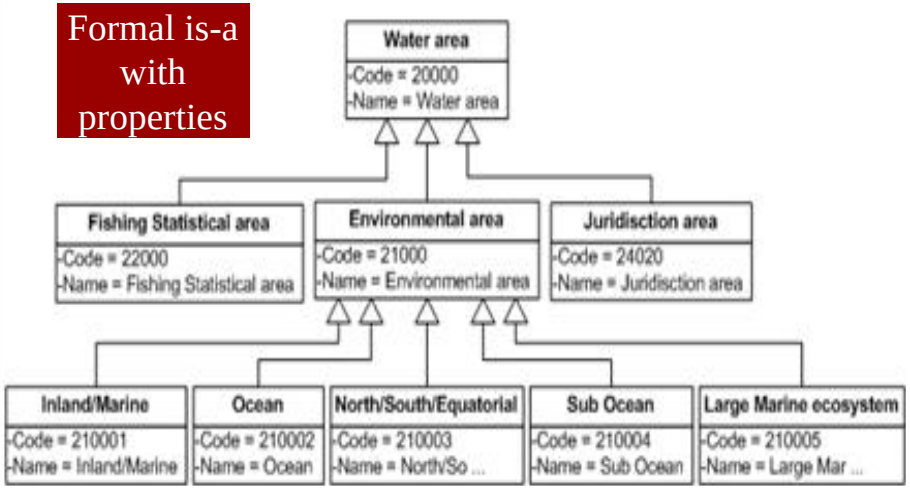
Id	Category Name	Parent
20000	Water area	1
21000	Environmental area	20000
22000	Fishing Statistical area	20000
24020	Jurisdiction area	20000
21001	Inland/marine	21000
21002	Ocean	21000
21003	North/South/Equatorial	21000
21004	Sub Ocean	21000
21005	Large Marine ecosystem	21000

```
! 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 Microstatio
!
! 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.

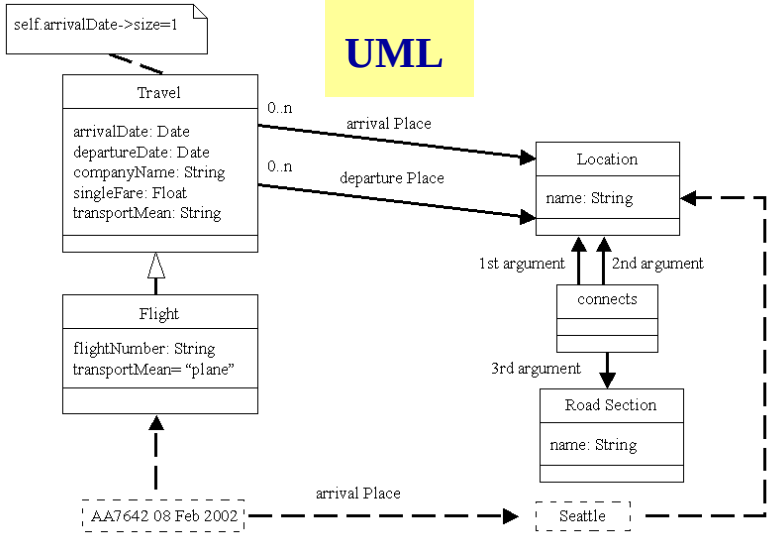
```
(define-class AmericanAirlinesFlight (?X)
:def (Flight ?X)
:axiom-def
  (Disjoint-Decomposition AmericanAirlinesFlight
          (Setof AA7462 AA2010 AA0488)))

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

Disjointness

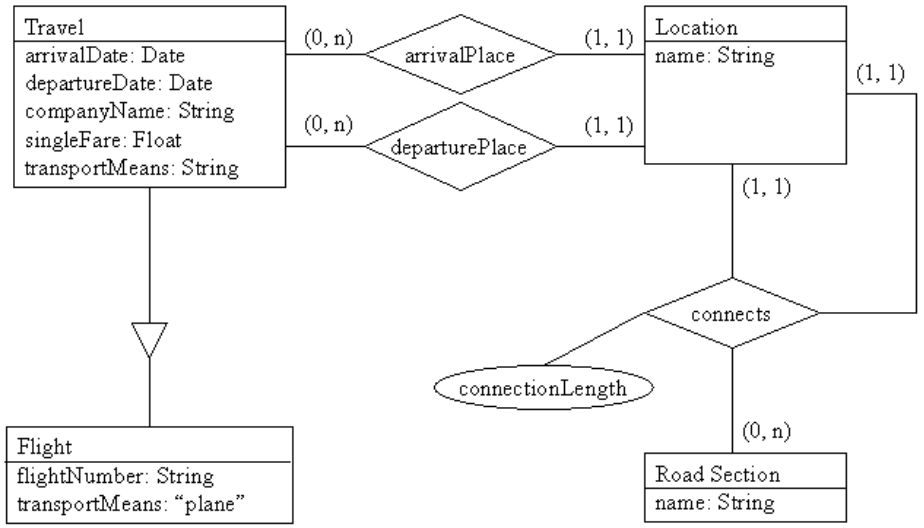
Table of Content

Approaches for building ontologies

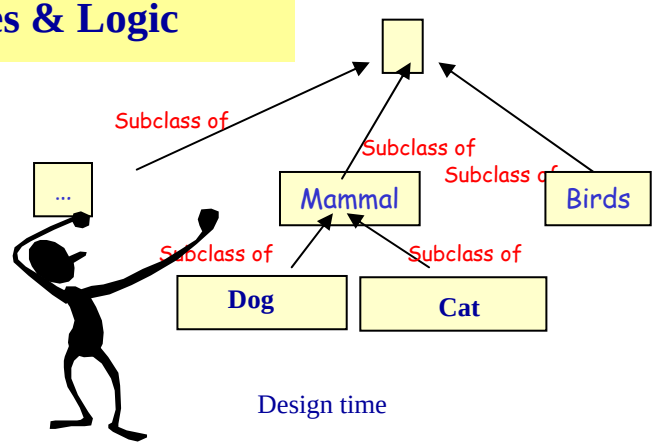
UML



E/R Model

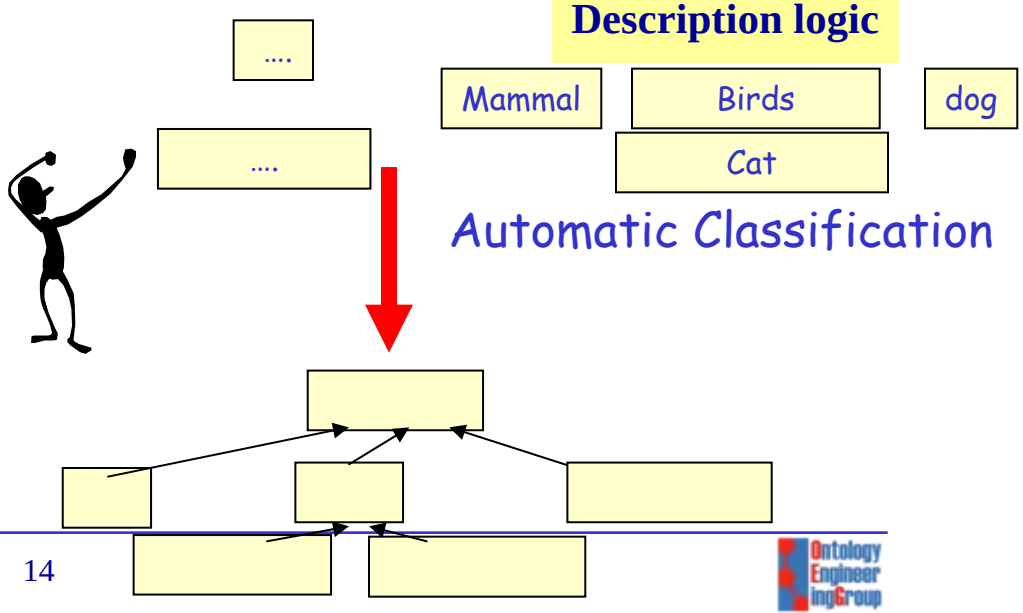


Frames & Logic



Design time

Description logic



Components of an Ontology

Concepts are organized in **taxonomies**

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

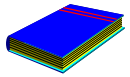
Subclass-of: Concept 1 x Concept2
Connected to: Component1 x Component2

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

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

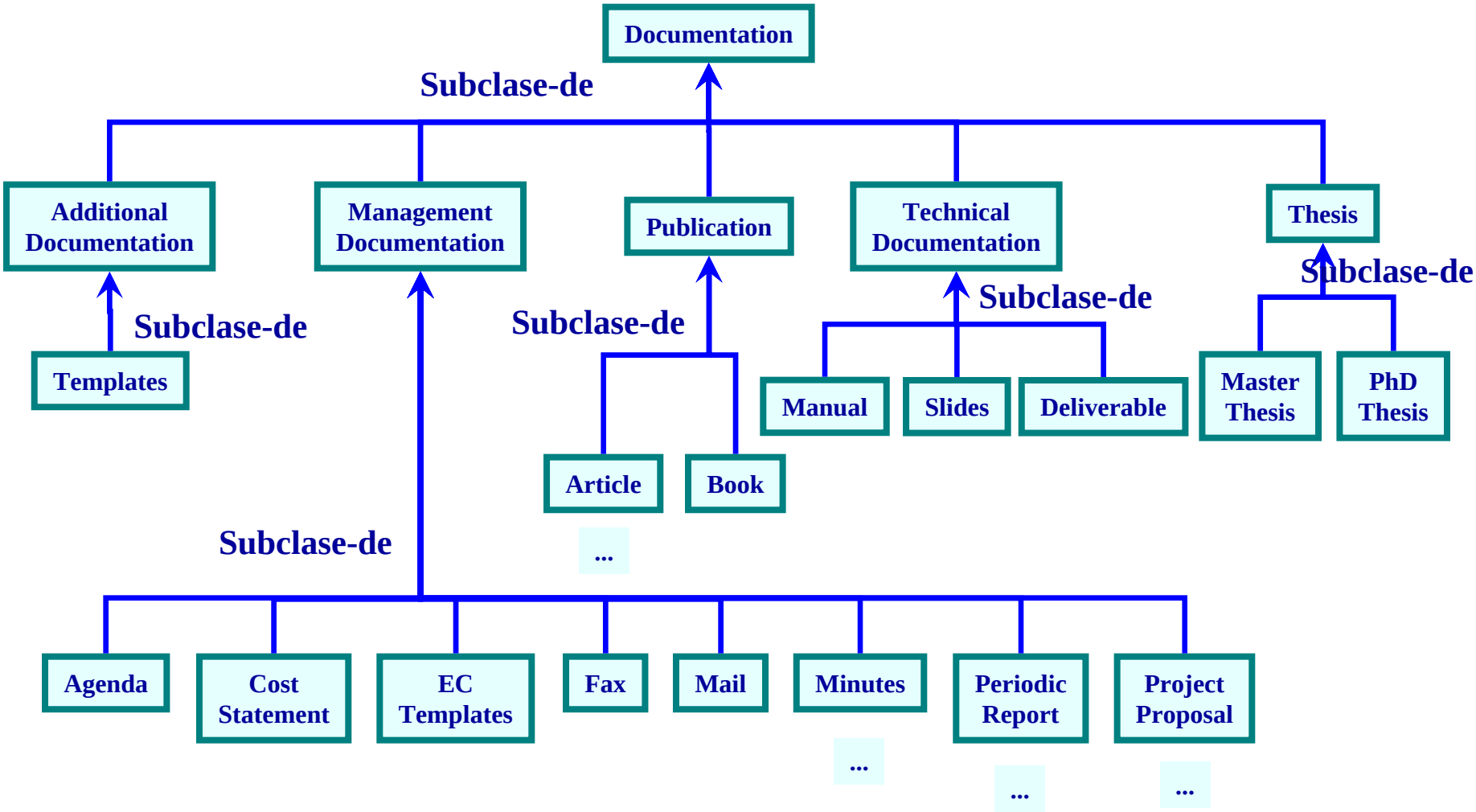
Instances Elements

Axioms Sentences which are always true



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

Documentation Taxonomy



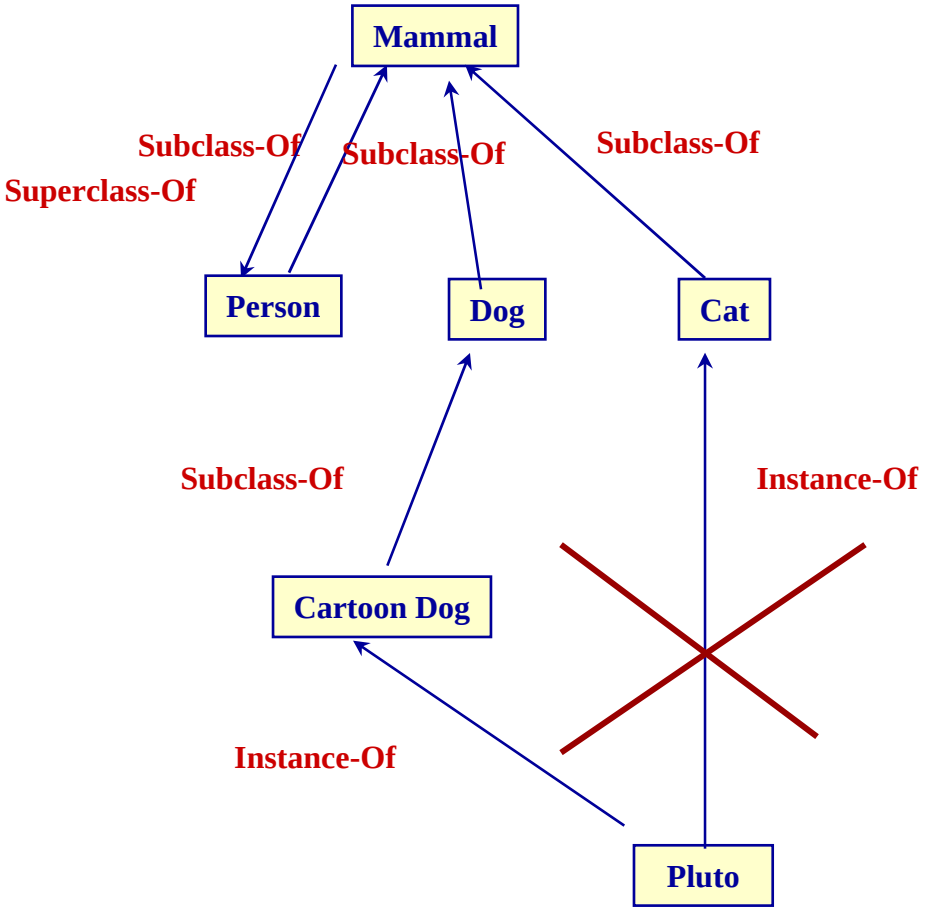
Modelling disjoint knowledge

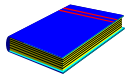
class-Partition: a set of disjoint classes

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)



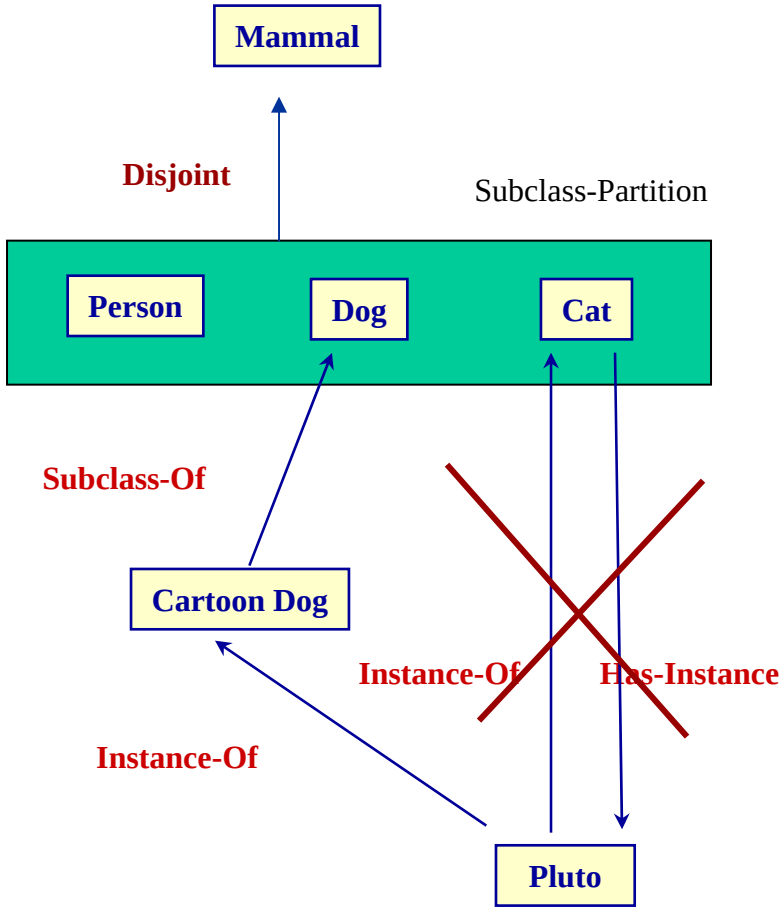


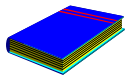
A. Gómez-Pérez. *Evaluation of Ontologies.*
International Journal of Intelligent Systems.
Vol. 16, Nº3. March 2001. PP391-410

Pluto could be an instance of cat and dog

Semantic Error

Why disjoint knowledge is important (II)

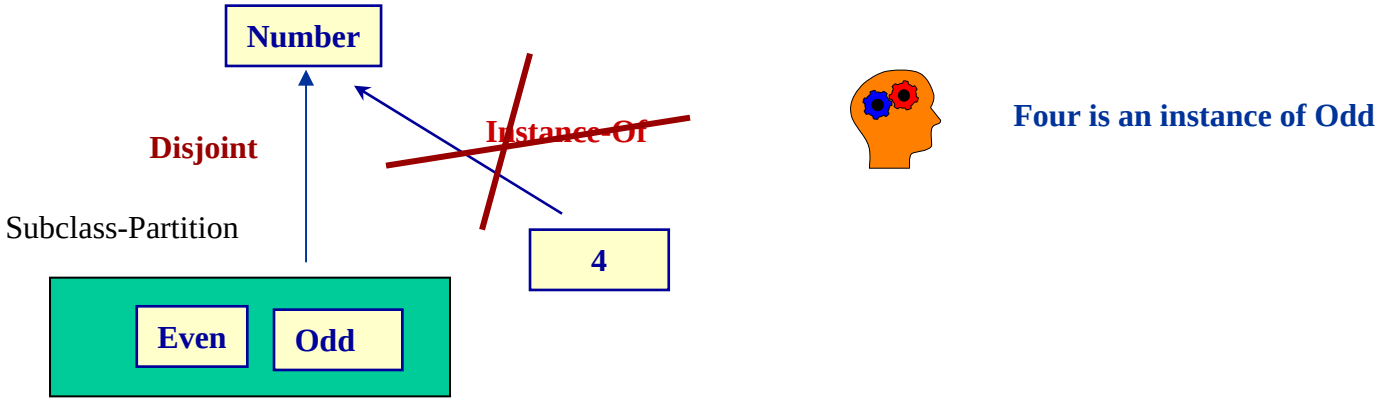


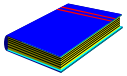


A. Gómez-Pérez. *Evaluation of Ontologies.*
International Journal of Intelligent Systems.
Vol. 16, Nº3. March 2001. PP391-410

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

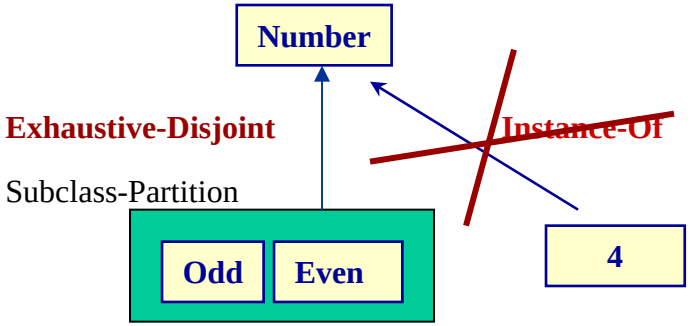
Why disjoint knowledge is important (III)





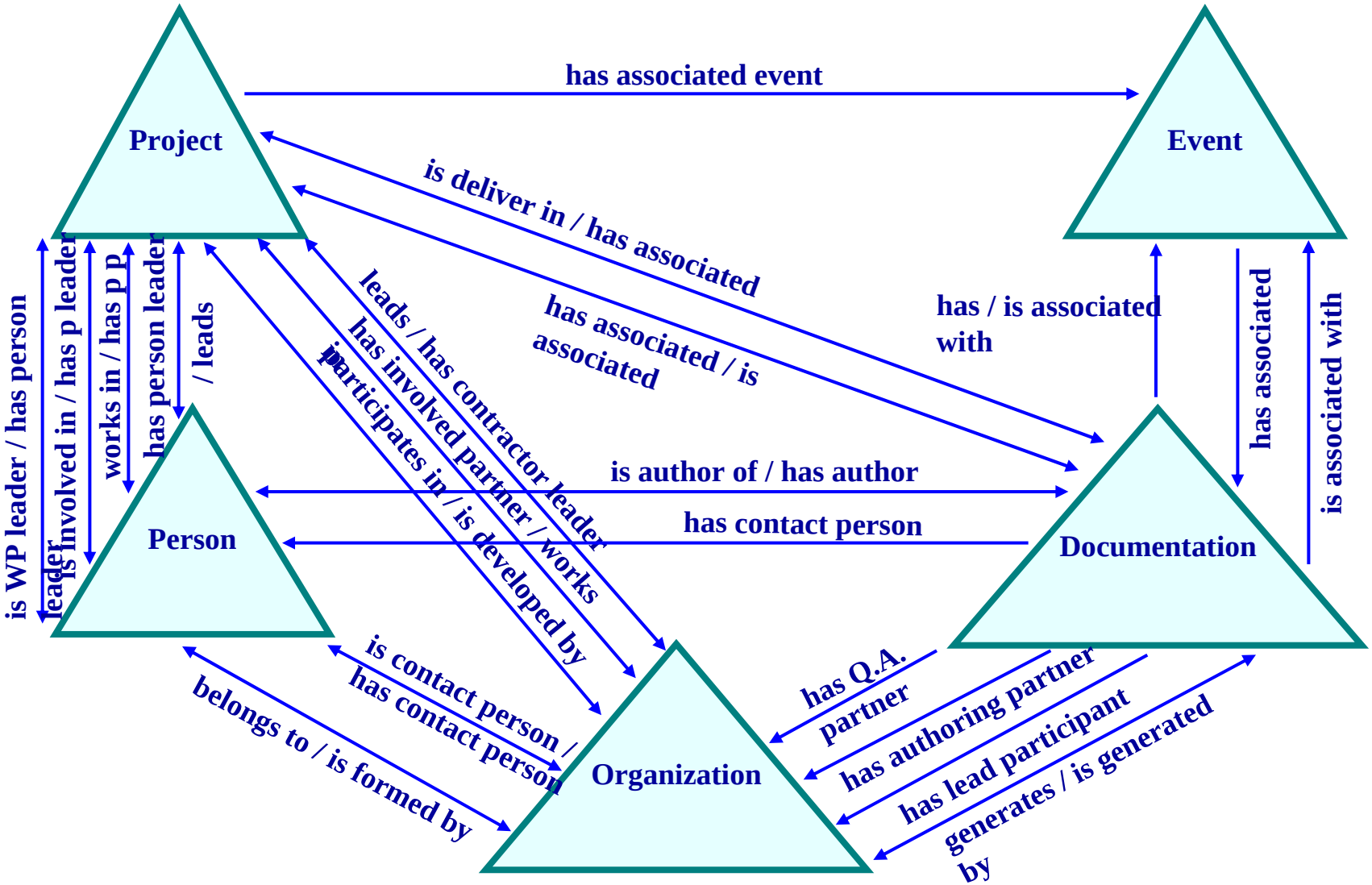
A. Gómez-Pérez. *Evaluation of Ontologies.* **International Journal of Intelligent Systems.** Vol. 16, Nº3. March 2001. PP391-410

Why disjoint knowledge is important (IV)

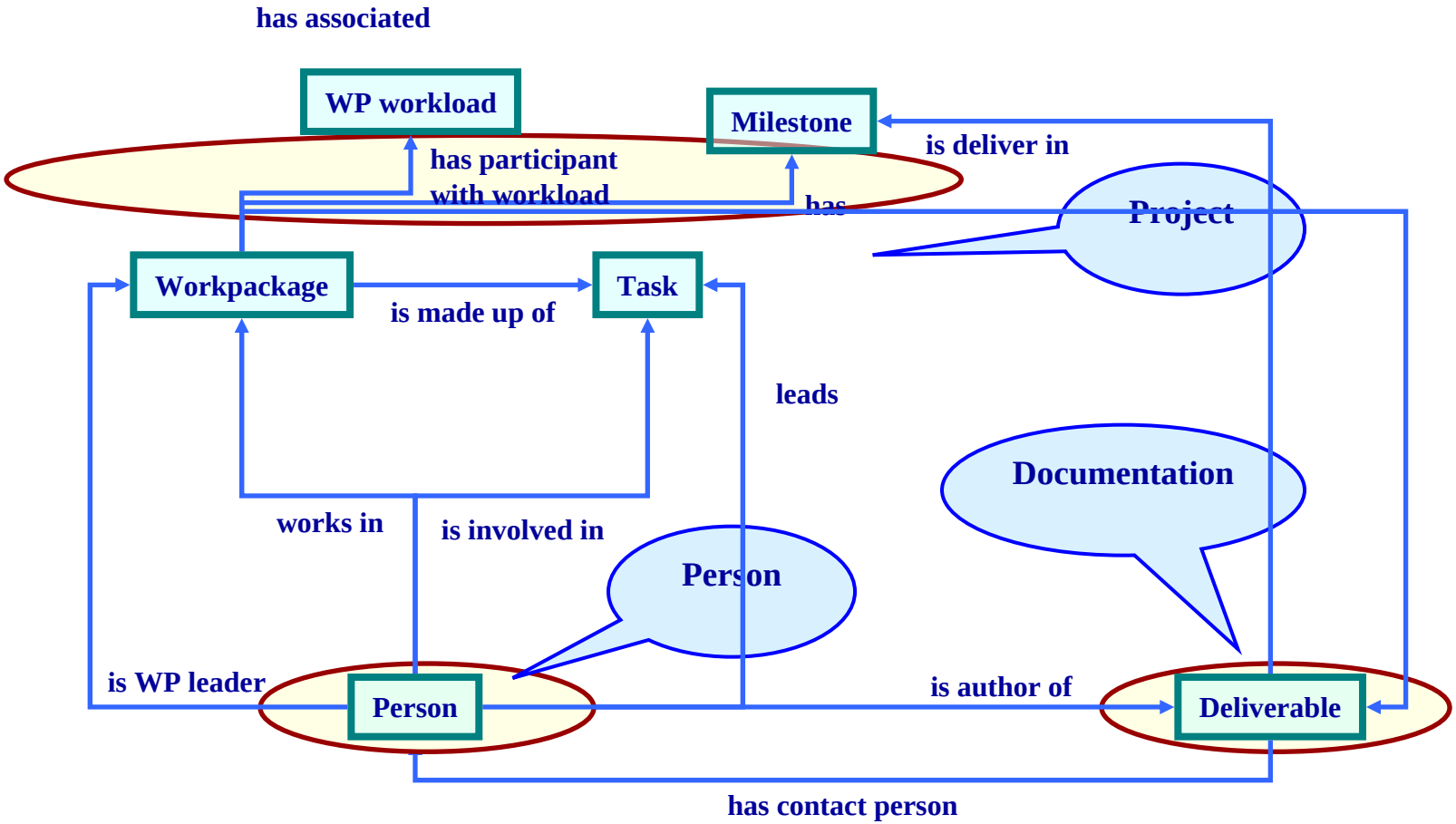


Four is an instance of something in the partition

Relations between concepts



Relationships between Person, Project and Documentation



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)))))))
```

Ontology Modeling: General Issues

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

- Ontology development is necessarily an iterative process.

- 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.

- It is important the use of naming conventions
  - Concepts: Human, Man, Woman
  - Relationships: is-Married-with, is-parent-of
  - Attributes: name, age, dateOfBirthday

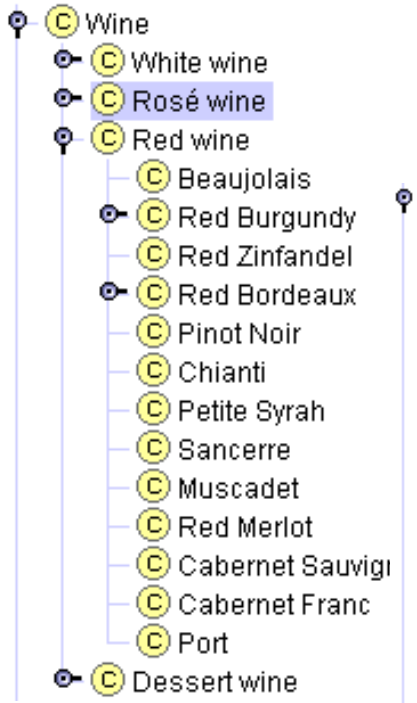
Classes and Class Hierarchy (I)

- Classes represent concepts in the domain and not the words that denote these concepts.
  - Synonyms for the same concept do not represent different classes
- Classes as unary predicates—questions that have one argument.
  - For example, “Is this object a wine?”
  - By contrast binary predicates (or slots)—questions that have two arguments. For example, “Is the flavor of this wine strong?” “What is the flavor of this wine?”
- A subclass of a class represents a concept that is a “kind of” the concept that the superclass represents.

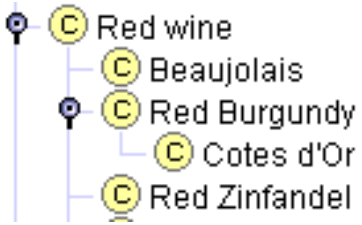
Classes and Class Hierarchy (II)

- Organize the classes into a hierarchical taxonomy by asking if by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other class.
  - *If a class A is a superclass of class B, then every instance of B is also an instance of A*
  - Apple is a subclass of Fruit → *Every apple is a fruit*
  - Red wines is a subclass of Wine → *Every red wine is a wine*
  - Chianti wine is a subclass of Red wine → *Every Chianti wine is a red wine*
- *A single person is not a subclass of all persons*

Class Hierarchy

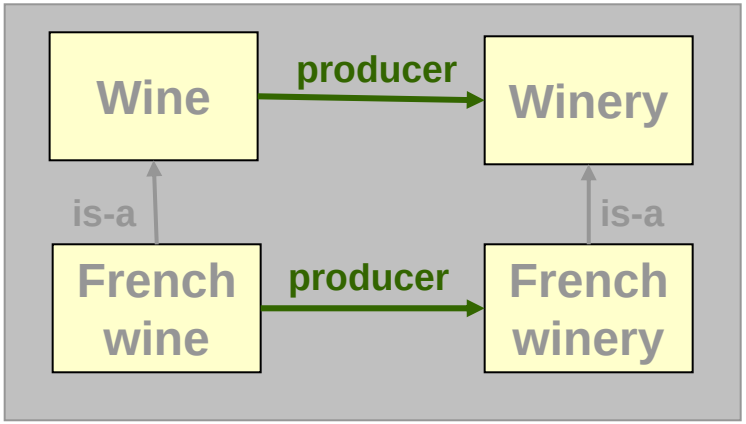


- **All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality.**
- **Subclasses of a class usually**
  - have additional properties that the superclass does not have, or
  - restrictions different from those of the superclass, or
  - participate in different relationships than the superclasses
- **If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary.**
- **If a class has only one direct subclass there may be a modeling problem or the ontology is not complete.**



Properties and Class Inheritance

- **Properties** are inherited and should be attached to the most general class in the hierarchy
- A subclass inherits all the properties from the superclass
  - *If a wine has a name and flavor, a red wine also has a name and flavor*
- If a class has multiple superclasses, it inherits properties from all of them
  - *Port is both a dessert wine and a red wine. It inherits "sugar content: high" from the former and "color:red" from the latter*



Domain and Range of Properties

- **When defining a domain or range find the most general class or classes**
  - *The domain of flavor should be Wine and not Red wine or White wine*
  - *The range of produces for a Winery should be Wine and not Red, White or Rosé wine*

- **General patterns**
  - A class and a superclass – replace with the superclass
  - All subclasses of a class – replace with the superclass
  - Most subclasses of a class – consider replacing with the superclass

Common Modelling Errors

P1. Creating polysemous elements
P2. Creating synonyms as classes
P3. Creating the relationship "is" instead of using "subclassOf", "instanceOf" or "sameIndividual"
P4. Creating unconnected ontology elements
P5. Defining wrong inverse relationships
P6. Including cycles in the hierarchy
P7. Merging different concepts in the same class
P8. Missing annotations
P9. Missing basic information
P10. Missing disjointness
P11. Missing domain or range in properties
P12. Missing equivalent properties
P13. Missing inverse relationships
P14. Misusing "allValuesFrom"
P15. Misusing "not some" and "some not"
P16. Misusing primitive and defined classes
P17. Specializing too much a hierarchy
P18. Specifying too much the domain or the range
P19. Swapping intersection and union
P20. Swapping Label and Comment
P.21 Using a miscellaneous class
P22. Using different naming criteria in the ontology
P23. Using incorrectly ontology elements
P24. Using recursive definition

Table of Content

Types of Ontologies

Content Ontologies

Issue of the Conceptualization

Domain O.

Scalpel, scanner
anesthetize, give birth

Representation O.

• Conceptualization of KR formalisms

Task O.

goal, schedule
to assign, to classify

Generic O.

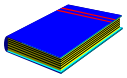
• Reusable across D.

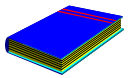
General/Common O.

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

Domain O.

• Reusable





Application O.

• Non reusable
• Usable

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.

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

Knowledge Representation Ontologies

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

- **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)

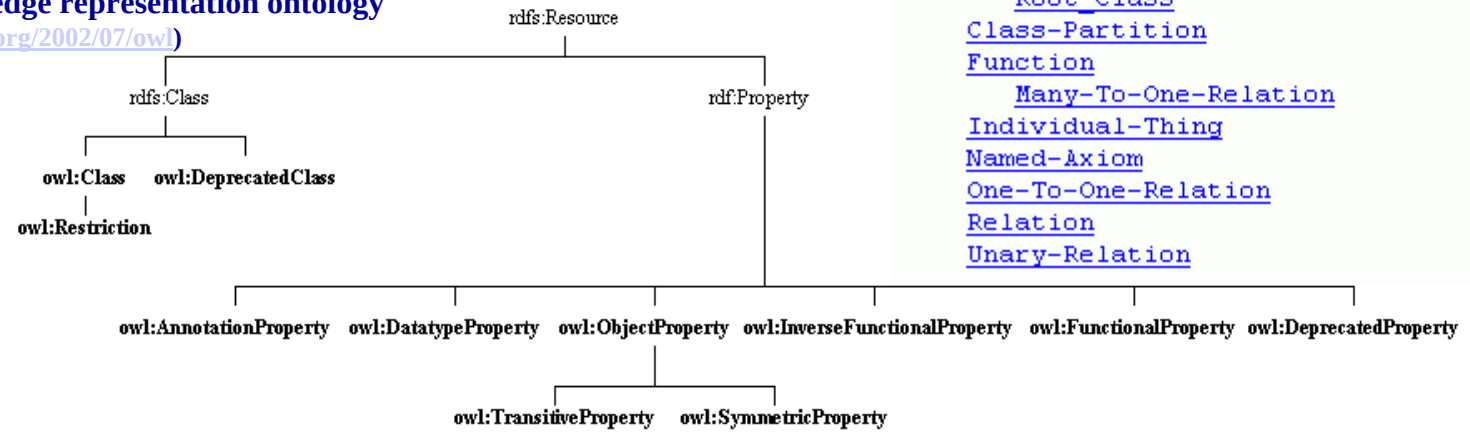
- **OIL knowledge representation ontology**
  (http://www.ontoknowledge.org/oil/rdf-schema/2000/11/10-oil-standard)

- **DAML+OIL knowledge representation ontology**
  (http://www.daml.org/2001/03/daml+oil)

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

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
```



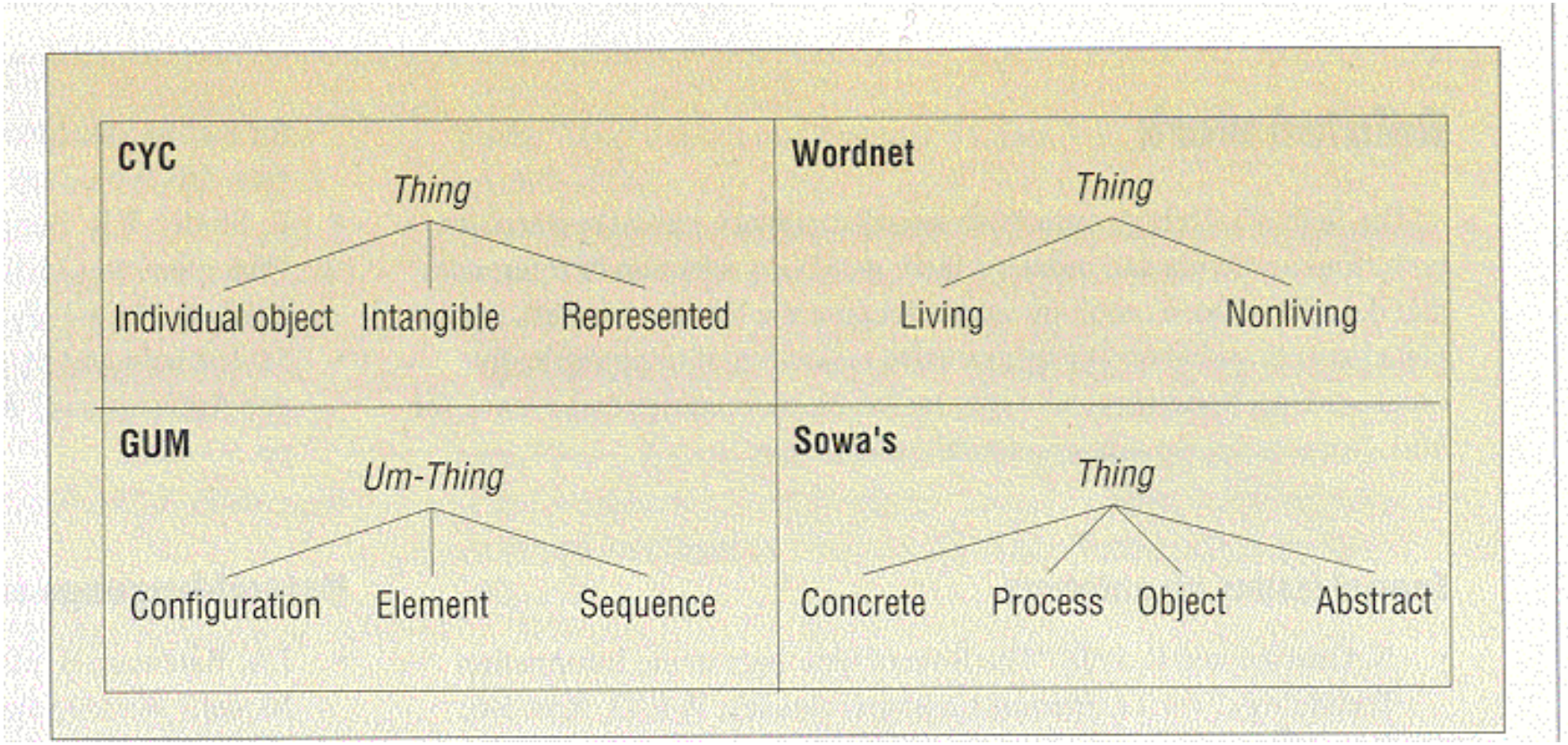
Definition of the relation SUBCLASS-OF in the Frame Ontology

```
(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

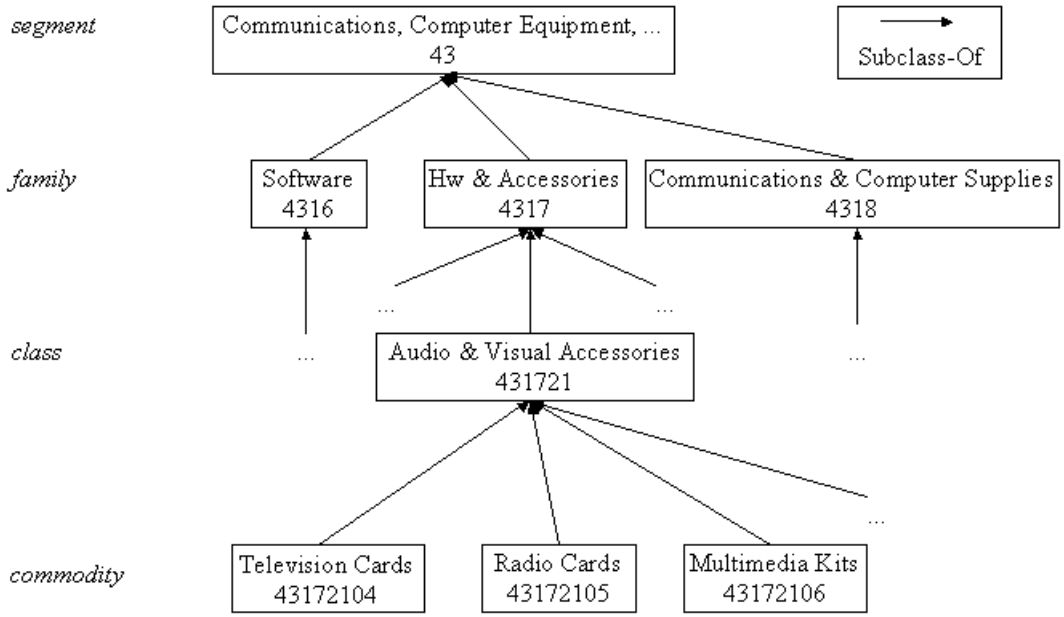
One Unique Top-Level Ontology?

Various proposals



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

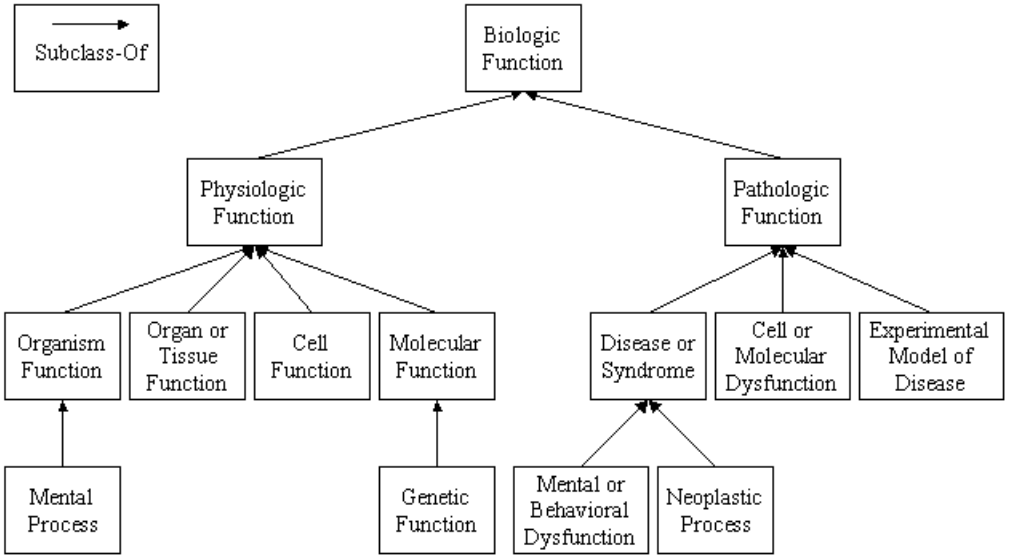
•**GALEN** (http://www.opengalen.org)

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

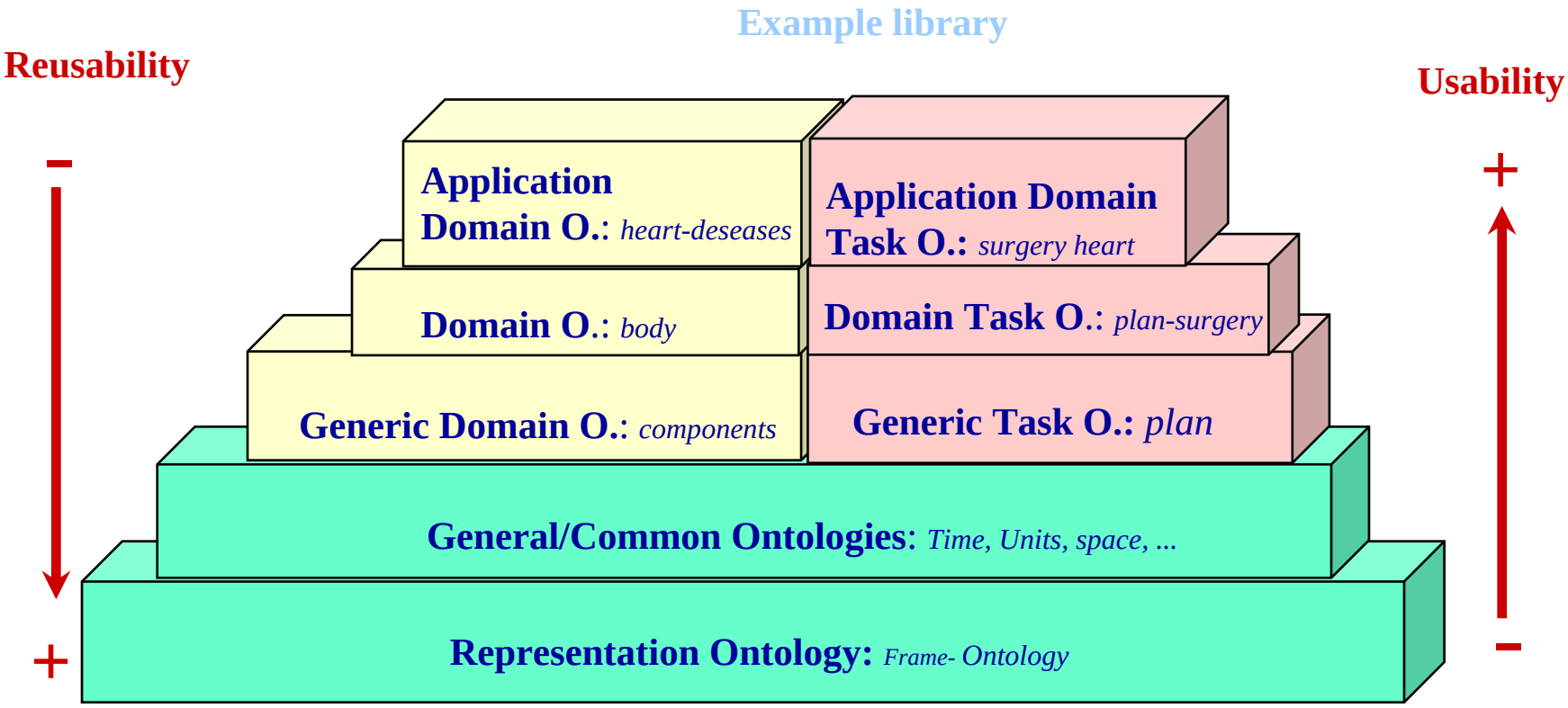
•**UMLS (Unified Medical Language System)**
(http://www.nih.gov/research/umls/)

•**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



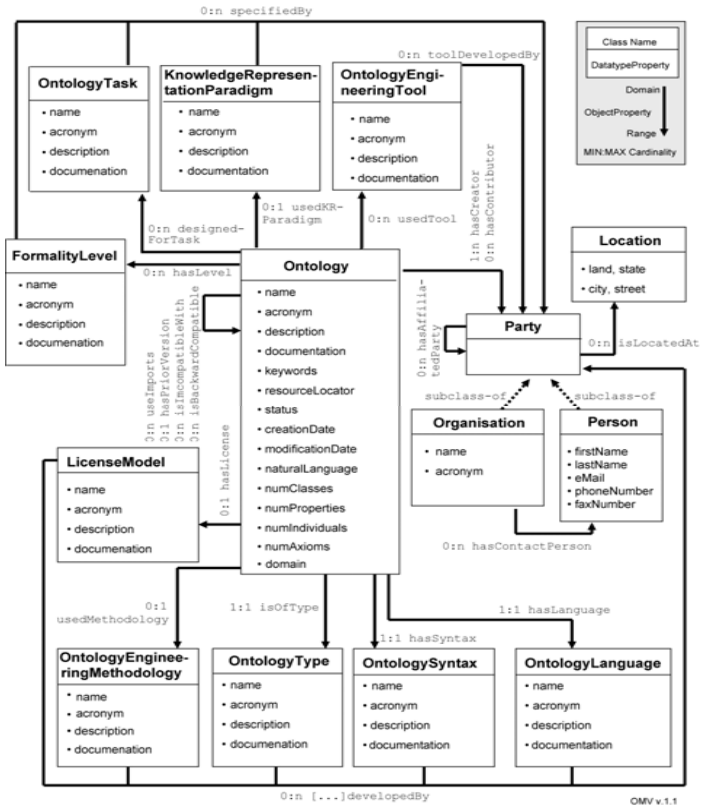
Libraries of Ontologies



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

Searching Ontologies

- OMV: Ontology Metadata Vocabulary



- Ontology registries

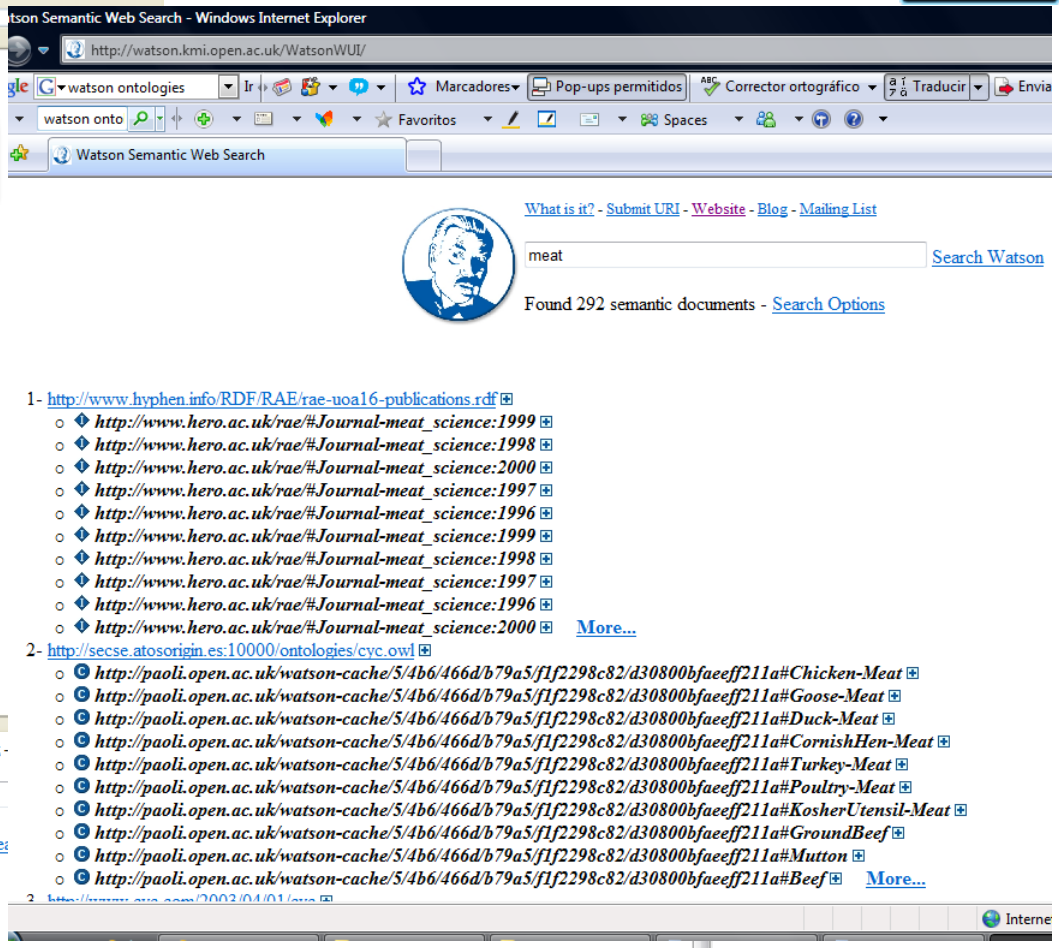
CYSTER

Onthology.org

knowledge zone one stop shop for ontologies

Swoogle semantic web search 2006

watson exploring the semantic web



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

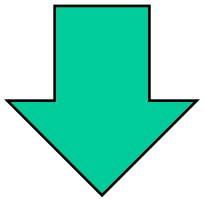
14- http://reliant.teknowledge.com/DAML/SUMO.daml
- http://kmi-web05.open.ac.uk:81/cache/1/f5d/857a/f0cae/8b96c6b3ac/1e19b0c5cf7f2a849#Meat
- http://kmi-web05.open.ac.uk:81/cache/1/f5d/857a/f0cae/8b96c6b3ac/1e19b0c5cf7f2a849#Beverage

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

What is an Ontology?

Shared understanding of a domain



Repository of vocabulary

- Formal definitions
- Informal definitions