Main Components

Applications

Build Ontologies

Methodologies and Methods

Technological Support

Reasoners

Languajes
Table of Content

Ontologies

1. Reuse and Sharing
2. Definitions of Ontologies
3. Modeling of Ontologies
4. Type of Ontologies
5. Libraries of Ontologies
Reuse and Sharing

**Reuse** means to build new applications assembling components already built

**Sharing** is when different applications use the same 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 infrastructure to support such sharing and reuse would lead to greater ubiquity of these systems, potentially transforming the knowledge industry ...”

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]

Ingeniería Ontológica

Conjunto de actividades relativas al proceso de desarrollo de ontologías su ciclo de vida, métodos y metodologías para construirlas, y el conjunto de herramientas y lenguajes en los que se implementan.
Table of Content

Ontologies

1. Reuse and Sharing
2. Definitions of Ontologies
3. Modeling of Ontologies
4. Type of Ontologies
5. Libraries of Ontologies
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”


2. “An ontology is an explicit specification of a conceptualization”

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.


4. An ontology provides the means for describing explicitly the conceptualization behind the knowledge represented in a knowledge base.

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

Consensual Knowledge

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

Machine-readable

Ontologías

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

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

Ontologies

1. Reuse and Sharing
2. Definitions of Ontologies
3. Modeling of Ontologies
4. Type of Ontologies
5. Libraries of Ontologies
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

Strategies for building taxonomies: Botton up strategy
Strategies for building taxonomies: Top Down strategy

- Object
  - Concrete object
    - Taxi
    - Bus
    - Train
  - Abstract object
    - Transport by underground
    - Transport by bus
    - Transport by taxi

uses
uses
uses
Strategies for building taxonomies: Middle out strategy

![Taxonomy Diagram]

- Transport mean
  - Underground
  - Bus
    - Local bus
    - Shuttle
  - Taxi
    - Coach

Starting concept

Subclass of
Primitivas necesarias para modelizar conocimientos disjuntos en taxonomías

**class-Partition:** Conjunto de clases que son disjuntas entre sí

**Disjoint:** un conjunto de clases que son disjuntas entre sí son subclase de una clase padre

**Exhaustive-Disjoint:** un conjunto de clases que son disjuntas entre sí son subclase de una clase padre y el conjunto de clases definen completamente a la clase padre.
Why disjoint knowledge is important (I)

Pluto could be an instance of cat and dog

Semantic Error

Why disjoint knowledge is important (II)

*International Journal of Intelligent Systems.*

Pluto cannot be simultaneously a class of Cat and Dog because they are disjoint.
Why disjoint knowledge is important (III)

Why disjoint knowledge is important (IV)

Four is an instance of something in the partition
Propiedades

Que se definen en la clase

Son Atributos del concepto o clase
Se definen y rellenan en la clase
El valor es el mismo para todas los individuos

Que se definen en la clase y se rellenan en la Instancia

Atributos específicos de cada individuo
Se definen en el marco clase
Se rellenan en el marco instanciado
En cada individuo puede tomar un valor diferente
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)))))))))
## Instance Attributes for Term **Travel**

<table>
<thead>
<tr>
<th>Instance Attribute Name</th>
<th>Description</th>
<th>Value Type</th>
<th>Cardinality</th>
<th>Measurement Unit</th>
<th>Precision</th>
<th>Value Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival Date</td>
<td>Date of arrival of the trip</td>
<td>Date</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>company Name</td>
<td>Transportation company or companies in charge of a trip</td>
<td>String</td>
<td>(0, N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>departure Date</td>
<td>Date of departure of the trip</td>
<td>Date</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single Fare</td>
<td>Fare of a single ticket</td>
<td>Float</td>
<td>(0, 1)</td>
<td>US Dollar</td>
<td>0.01</td>
<td>0 -</td>
</tr>
</tbody>
</table>

**New**

- Travel
  - Flight
    - American Airlines flight
    - AA0488
    - AA2010
    - AA7462
  - British Airways flight
  - Iberia flight
  - Train Travel
  - Ship
- Travel package

**Edition area**

- **Term Name**: Travel
- **Instance Attribute Name**: return fare

**Description**

Fare of a return ticket

**Value Type**

- Float

**Minimum-Maximum Cardinality**

- Minimum: 0
- Maximum: 1

**Measurement Unit**

- US Dollar

**Precision**

- 0.01

**Minimum Value**

- 0

**Maximum Value**

- 0

**Browsing area**
Using Frames and First Order Logic for Modeling Ontologies

### Define Class Travel

```
(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))
:define
```

### Define Function Pays

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

### Define Relation Connects

```
(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."
:define (and (SpatialPoint ?source)
  (SpatialPoint ?target)
  (Edge ?edge))
:axiom-def
((=> (connects ?edge ?source ?target)
  (connects ?edge ?target ?source)) ; symmetry
 (=> (connects ?edge ?target ?source)
  (not (or (part-of ?source ?target) ; irreflexivity
   (part-of ?target ?source)))))
```

### Define Instance AA7462-Feb-08-2002

```
(define-instance AA7462-Feb-08-2002 (AA7462)
  :define ((singleFare AA7462-Feb-08-2002 300)
    (departureDate AA7462-Feb-08-2002 Feb8-2002)
    (arrivalPlace AA7462-Feb-08-2002 Seattle)))
```
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
Approaches for building ontologies

UML

Marcos y Lógica

Lógicas Descriptivas

Clasificación automática

Modelo entidad Relación

©Asunción Gómez-Pérez
Table of Content

Ontologies

1. Reuse and Sharing
2. Definitions of Ontologies
3. Modeling of Ontologies
4. Type of Ontologies
5. Libraries of Ontologies
Types of Ontologies
Lassila and McGuinness Classification

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

**Issue of the Conceptualization**

- **Domain O.**
  - Reusable

- **Application O.**
  - Non reusable
  - Usable

**Representation O.**

- **Generic O.**
  - Reusable across D.

- **Conceptualization of KR formalisms**


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


Ontological Commitments

The noun “flight” has 9 senses in WordNet

1. flight – (a formation of aircraft in flight)
2. flight, flying – (an instance of traveling by air; “flying was still an exciting adventure for him”)
3. flight, flight of stairs, flight of steps – (a set of steps between one floor or landing and the next)
4. escape, flight – (the act of escaping physically, “he made his escape from the mental hospital”, “the canary escaped from its cage”, “his flight was an indication of his guilt”)
5. flight – (a unit of the US air force smaller than a squadron)
6. flight – (passing above and beyond ordinary bounds, “a flight of fancy”, “flights or rhetoric”, “flights of imagination”)
7. trajectory, flight – (the path followed by a moving object)
8. flight – (a flock of flying birds)
9. flight – (a scheduled trip by plane between designated airports, “I took the noon flight to Chicago”)
A formation of aircraft in flight

An instance of traveling by air

A set or steps between one floor or landing for him

The act of escaping physically

A unit of the US air force smaller than a squadron

Passing above and beyond ordinary bounds

A flock of flying birds

The path followed by a moving object

(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")))
Table of Content

Ontologies

1. Reuse and Sharing
2. Definitions of Ontologies
3. Modeling of Ontologies
4. Type of Ontologies
5. Libraries of Ontologies
Libraries of Ontologies

Example library

Reusability

Application Domain O.: heart-deseases
Application Domain Task O.: surgery heart

Domain O.: body
Domain Task O.: plan-surgery

Generic Domain O.: components
Generic Task O.: plan

General/Common Ontologies: Time, Units, space, ...

Representation Ontology: Frame-Ontology

http://delicias.dia.fi.upm.es/mirror-server/ont-serv.html
Relationship between Ontologies in the Library

- Environmental Pollutants
  - Monoatomic-Ions
  - Poliatomic-ions
  - Chemical-Elements
    - Standard-Units
    - Standard-Dimensions
      - Physical-Quantities
        - Kif-Numbers
        - Frame-Ontology
Ontology Searching in Ontology Metadata Repositories

Ontology to describe ontology metadata information
  - OMV – Ontology Metadata Vocabulary (http://ontoware.org/projects/omv)
  - Knowledge Zone vocabulary (http://tinyurl.com/qfp2s)

4 Ontology Metadata Repositories
  - Oyster (P2P system, http://oyster.ontoware.org)
  - ONTHOLOGY.org (centralized, http://www.onthology.org/)
  - Knowledge Zone (centralized, http://smiprotege.stanford.edu:8080/KnowledgeZone/)
  - Swoogle (http://swoogle.umbc.edu/)

©Asunción Gómez-Pérez 11/10/2007 11:08 - 39
**Searching ontologies: Obtain the set of candidate ontologies using Oyster**

<table>
<thead>
<tr>
<th>Scope</th>
<th>OWL</th>
<th>UPML-MAIN Person</th>
<th>UPML-Main Peer</th>
<th>rapOntos, UPML-Main Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Pe</td>
<td>The RDF Schema vocabulary (RDFS)</td>
<td>OWL</td>
<td>UPML-Main Peer</td>
<td>rapOntos, UPML-Main Peer</td>
</tr>
<tr>
<td>Auto</td>
<td>The RDF Vocabulary (RDF)</td>
<td>OWL</td>
<td>UPML-Main Peer</td>
<td>rapOntos, UPML-Main Peer</td>
</tr>
<tr>
<td>Automat</td>
<td>Topic Maps</td>
<td>OWL</td>
<td>UPML-Main Peer</td>
<td>rapOntos, UPML-Main Peer</td>
</tr>
<tr>
<td>Date</td>
<td>Thread Description Language (TDL)</td>
<td>DAML+OIL</td>
<td>rapOntos, UPML-Main Peer</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Topic Maps</td>
<td>OWL</td>
<td>UPML-Main Peer</td>
<td>rapOntos, UPML-Main Peer</td>
</tr>
<tr>
<td>Entry Details</td>
<td>Trust Networks on the Semantic Web</td>
<td>DAML+OIL</td>
<td>rapOntos, UPML-Main Peer</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>Trust Ontology</td>
<td>OWL</td>
<td>rapOntos, UPML-Main Peer</td>
<td></td>
</tr>
</tbody>
</table>

**Integration of Results**

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:omv="http://omv.ontoware.org/2005/05/ontology#">
  <rdf:Description rdf:about="http://keg.cs.tsinghua.edu.cn/ontology/travel">
    <rdf:type rdf:resource="http://omv.ontoware.org/2005/05/ontology#OntologyDocument"/>
    <omv:ontologyLanguage>OWL</omv:ontologyLanguage>
    <omv:creationDate>2005-08-28T10:51:24+01:00</omv:creationDate>
    <omv:numClasses>84</omv:numClasses>
    <omv:numProperties>211</omv:numProperties>
    <omv:docCreator>Po Zhang</omv:docCreator>

    <omv:Language>en</omv:Language>
    <omv:Description>
      <omv:docTitle>Travel Ontology</omv:docTitle>
      <omv:docAbstract>This Ontology contains three subdomains in travel domain, such as flight, hotel and car. It also provides domain knowledge of travel, such as flight, hotel and car.</omv:docAbstract>
    </omv:Description>
```

**Entry Details**

- Shopp
- Computer
- News
- World
- Science
- Health
- Art
- Games

©Asunción Gómez-Pérez
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

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

(\texttt{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
\[(\texttt{and (Class ?parent-class)}\]
\[(\texttt{Class ?child-class)}\]
\[(\texttt{forall (?instance)}\]
\[(\texttt{=> (Instance-Of ?instance ?child-class)}\]
\[(\texttt{Instance-Of ?instance ?parent-class)}\])\]

:axiom-constraints
\[(\texttt{(Transitive-Relation Subclass-Of)})\]
:iissued
\[(:(\texttt{see-also direct-subclass-of})\]
\[(:(\texttt{see-also "In CycL, subclass-of is called #\%allGenls because it is a slot from a collection to all of its generalizations (superclasses)."})\]
\[("In the KL-ONE literature, subclass relationships are also called subsumption relationships and ISA is sometimes used for subclass-of.")\]
\[("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 of universals and particulars (http://webode.dia.fi.upm.es/)


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


• 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/)

One Unique Top-Level Ontology?

Various proposals

<table>
<thead>
<tr>
<th>CYC</th>
<th>Wordnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>Thing</td>
</tr>
<tr>
<td>Individual object</td>
<td>Living</td>
</tr>
<tr>
<td>Intangible</td>
<td>Nonliving</td>
</tr>
<tr>
<td>Represented</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUM</th>
<th>Sowa's</th>
</tr>
</thead>
<tbody>
<tr>
<td>Um-Thing</td>
<td>Thing</td>
</tr>
<tr>
<td>Configuration</td>
<td>Concrete</td>
</tr>
<tr>
<td>Element</td>
<td>Process</td>
</tr>
<tr>
<td>Sequence</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
</tr>
</tbody>
</table>
Linguistic Ontologies

- **WordNet** ([http://www.hum.uva.nl/~ewn/gwa.htm](http://www.hum.uva.nl/~ewn/gwa.htm))


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

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

- **SENSUS** ([http://www.isi.edu/natural-language/projects/ONTOGIES.html](http://www.isi.edu/natural-language/projects/ONTOGIES.html))
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/](http://www.co-ode.org/galen/)


• **UMLS (Unified Medical Language System)**

• **ON9** [http://saussure.irmkant.rm.cnr.it/ON9/index.html](http://saussure.irmkant.rm.cnr.it/ON9/index.html)

Domain Ontologies: Engineering Ontologies

• EngMath


• PhysSys

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

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


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

Domain Ontologies: Knowledge Management Ontologies

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


• R&D projects (http://www.esperonto.net)
What is an Ontology?

Shared understanding of a domain

Repository of vocabulary

• Formal definitions
• Informal definitions