ONTOLOGICAL ENGINEERING

Asunción Gómez-Pérez
asun@fi.upm.es

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

Scope of the tutorial

- What is an ontology?
- What design principles should I follow to build an Ontology?
- What types of ontologies already exist?
- How are ontologies organized in libraries?
- What are the relationships between ontologies and knowledge bases?
- What methodology/steps should I use to build my own ontology?
- Which techniques are appropriate for each step?
- How do software tools support the process of building and using ontologies?
- What are the most well known ontologies?
- What are the uses of ontologies in applications?

Outline

1. Theoretical Foundations
2. Most Relevant Ontologies
3. Methodologies to build Ontologies
4. Tools
5. Applications

Ontological Engineering: Theoretical Foundations

Asunción Gómez-Pérez
asun@fi.upm.es

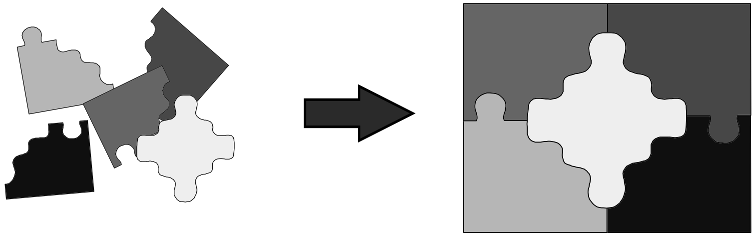
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Outline

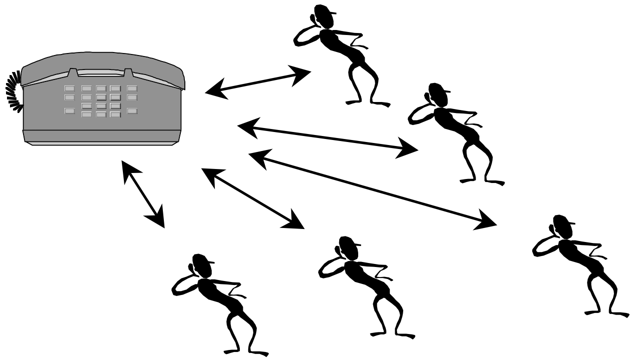
- Reuse and Sharing
- Problems in Building KBS from Scratch
- Problems when you reuse/share knowledge in KBS
- The Knowledge Sharing Initiative
- Definitions of Ontologies
- Ontological Commitments
- Components of an Ontology
- Types of Ontologies
- Libraries of Ontologies
- What does an explicit ontology look like?
- Principles for the Design of Ontologies
- Ontologies “versus” knowledge bases
- Uses of Ontologies

Reuse and Sharing

Reuse means to build new applications assembling components already built



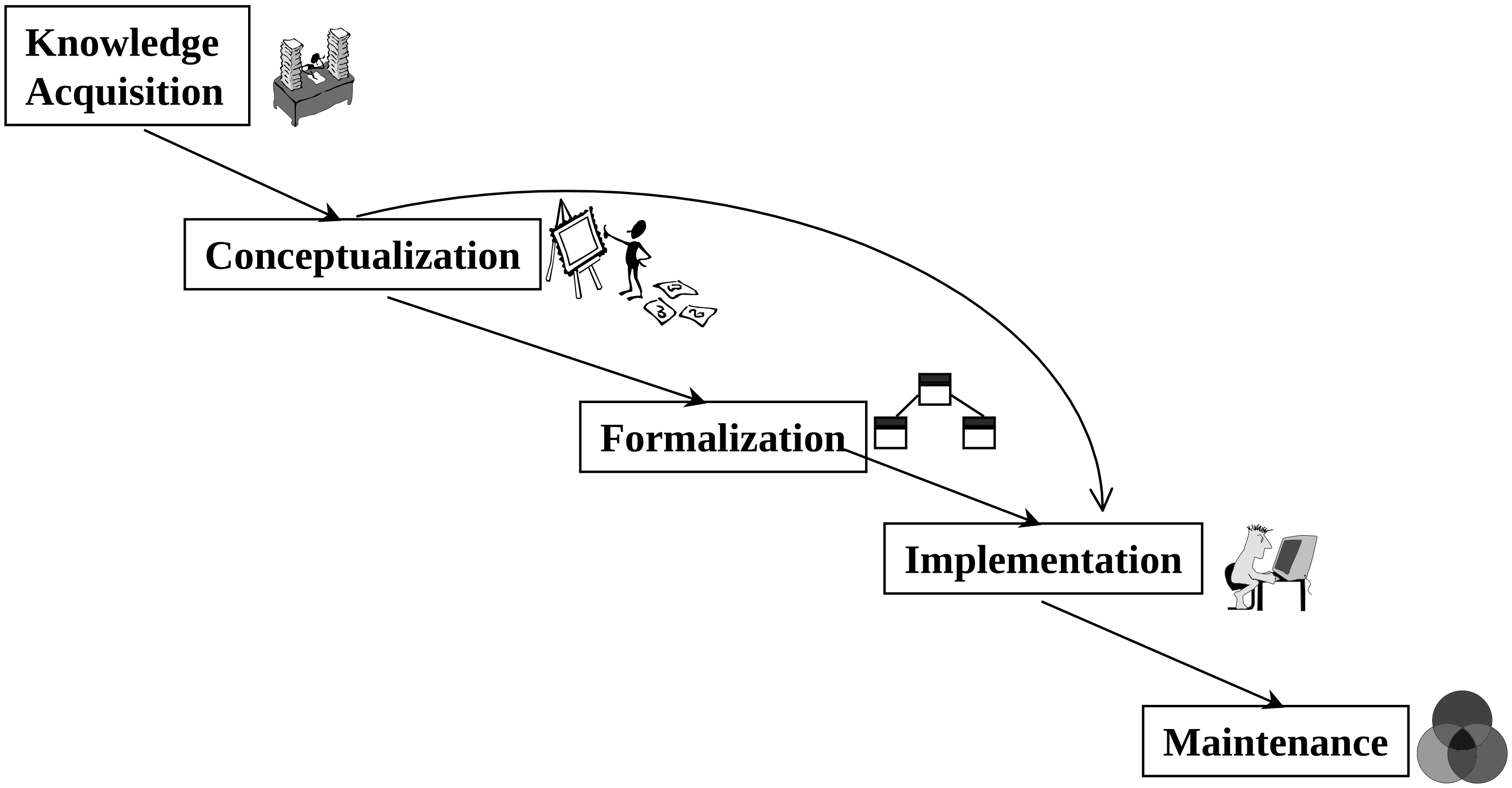
Sharing is when different applications use the some resources



Advantages:

- **Less money**
- **Less time**
- **Less resources**

Problems in building KBS from scratch



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.

A Declaration of Intentions

"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.

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)

2. Ontology as a specification of a conceptualization

3. Ontology as a philosophical discipline

4. Ontology as an informal conceptual system

5. Ontology as a formal semantic account

→ **Knowledge Level** (4–5)

6. Ontology as a representation of a conceptual system via a logical theory

7. Ontology as the vocabulary used by a logical theory

8. Ontology as a (meta-level) specification of a logical theory

→ **Symbolic Level** (6–8)

Guarino, N.; Giaretta, P. *Ontologies and Knowledge Bases: Towards a Terminological Clarification*. **Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing**. IOS Press. 1995. 25-32.

Definitions of Ontologies (III)

9. 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.

10. 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 (IV)

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

Machine-readable

Consensual Knowledge

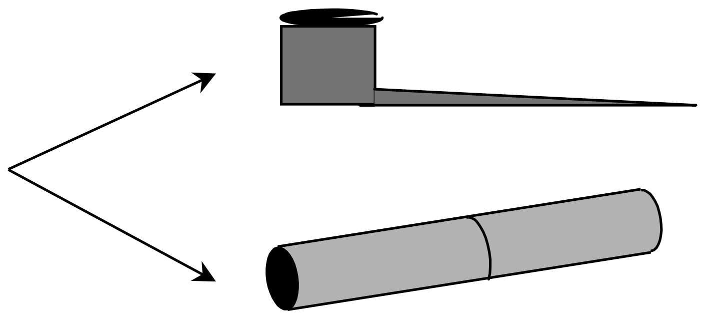
Concepts, properties relations, functions, constraints, axioms, are explicitly defined

Abstract model of some phenomenon in the world

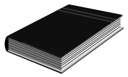
Studer, Benjamins, Fensel. **Knowledge Engineering: Principles and Methods.** ***Data and Knowledge Engineering*. 25 (1998) 161-197**

Ontological Commitments

Example: What is a pipe?



Agreements to use the vocabulary in a coherent and consistent manner



• 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

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

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

Instances **Elements**

Axioms **Sentences which are always true**

How to build taxonomies (I)

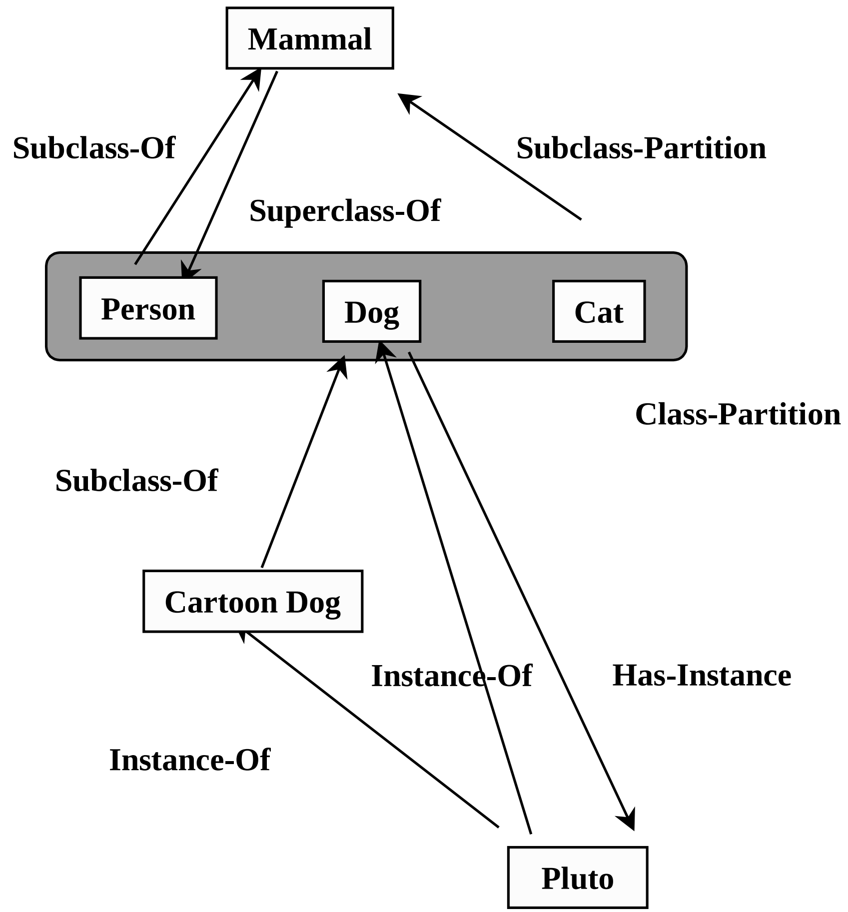
Main Relations between classes

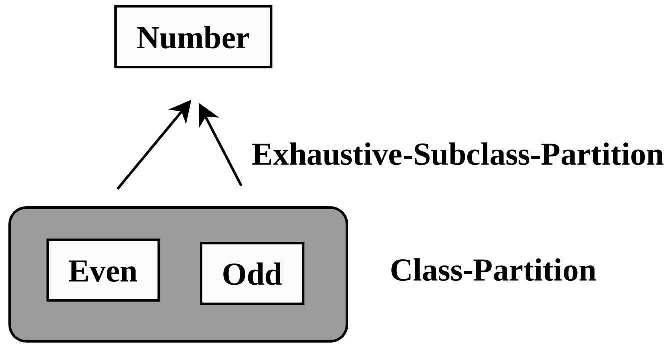
- **Subclass-of:**
- **Subclass-partition:**
- **Exhaustive-subclass-partion**

Main Relation between instances and classes

- **Instance-of**

How to build taxonomies (II)





Example 1: Bibliographic Data Ontology

Example 2: (KA)^2 Ontologies

What does an explicit ontology look like?

Highly informal: → in natural language

Semi-informal: → in a restricted and structured form of natural language

Example

Semi-formal: → in an artificial and formally defined language

Example

Rigorously formal: → in a language with formal semantics, theorems and proofs of such properties as soundness and completeness

Uschold, M.; Grüninger, M. *ONTOLOGIES: Principles, Methods and Applications.*
Knowledge Engineering Review. Vol. 11; N. 2; June 1996.

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.

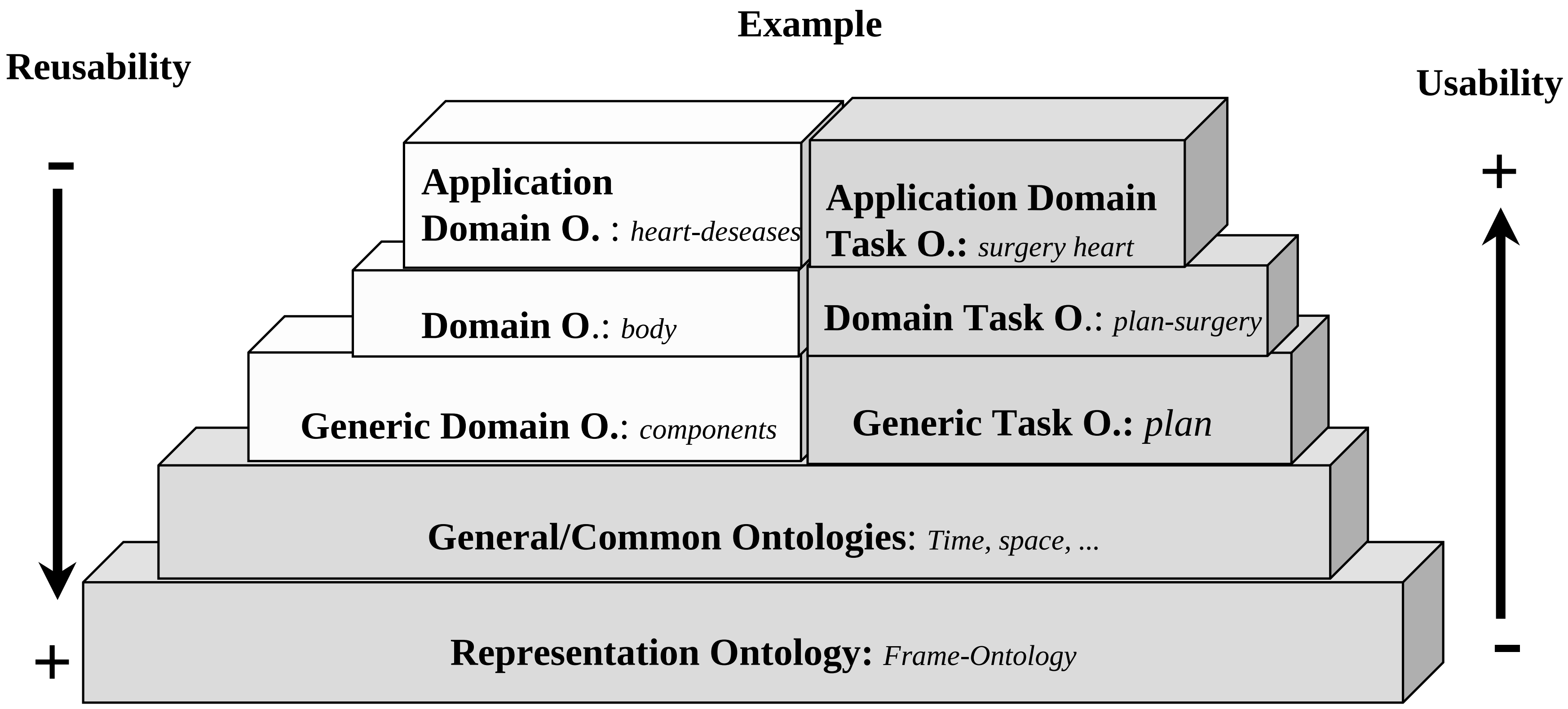
- Reusable

Application O.

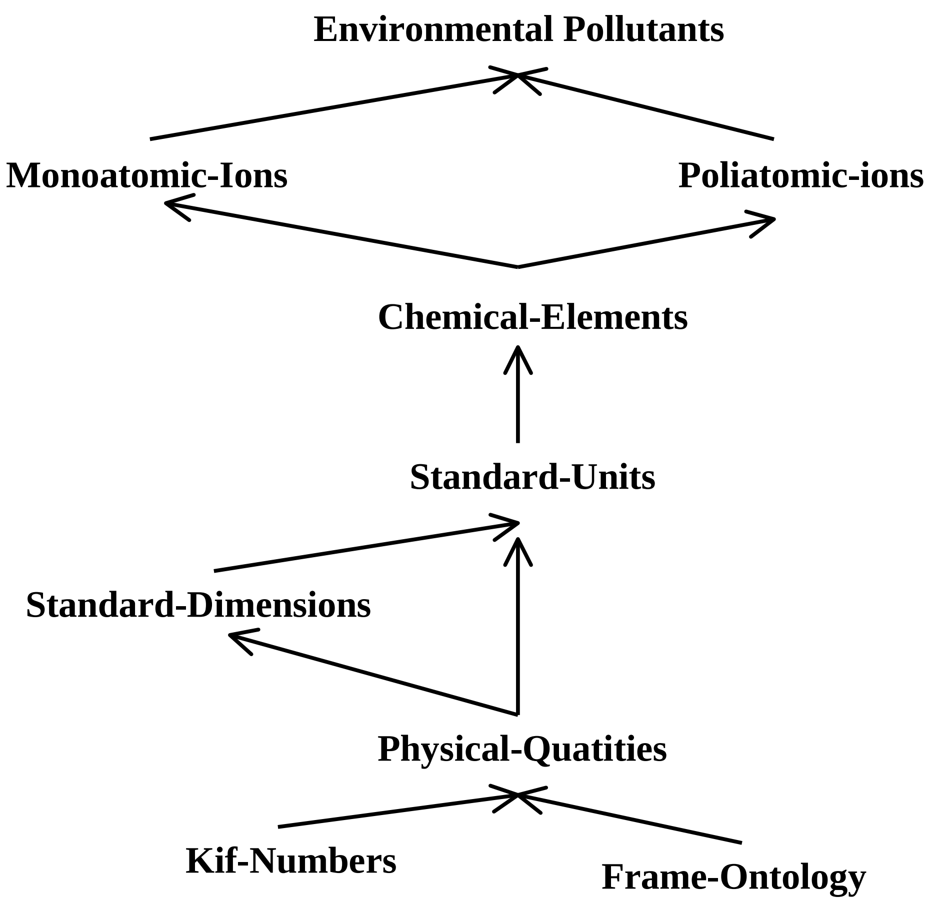
- Non reusable
- Usable

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

Libraries of Ontologies (I)



Relationship between ontologies in the library



Most Well-Known Ontologies (I)

Freely Available:

Ontolingua Ontologies: http://www-ksl.stanford.edu

Mirror site at Madrid: http://www-ksl-svc-lia.dia.fi.upm.es:5915

WordNet: http://www.tio.darpa.mil/Summaries95/B370-Princeton.html

Partially freely Available:

Cyc Ontologies: http://www.cyc.com/

TOP: The Ontology Page

http://www.medg.lcs.mit.edu/doyle/top/

Most Well-Known Ontologies (II)

• Interlingua: KIF and PIF

• Knowledge Representation Ontology: The Frame Ontology

• Top Level Ontologies: PANGLOSS, Penman Upeer Level, Cyc, MikroKosmos
Guarino’s and Sowa’s Top Level Ontology

• Linguistic Ontologies: Generalized Upper Model, WordNet, SENSUS

• Engineering Ontologies: EngMath, PhysSys

• Knowledge Management Ontologies: $(KA)^2$ Ontologies, Reference Ontology

• Modeling Enterprise: Enterprise and TOVE

• Domain Ontologies: CHEMICALS

KIF

Is a format to interchange knowledge, although it could be used to represent knowledge

Features:

- **Semantically Declarative**
- **Logically comprehensive: a prefix version of first-order predicate calculus**
- **Very expressive**
- **Extensions: meta-knowledge, nonmonotonic reasoning, ...**

What can be represented in KIF?

- **Objects: symbols, numbers, lists, sets, etc.**
- **Relations and functions of variable arity**

M. Genesereth ; R. Fikes. G. *Knowledge Interchange Format* Version 3.0 Reference Manual. **Report Logic 92-1** Computer Science Department. Stanford University. 1992.

KIF definitions

(Define-Relation <relation-name> (?arg$_1$ ... ? arg$_n$)
(Define-Function <function-name> (?arg$_1$ ... ? arg$_{n-1}$) -> ? arg$_n$
(Define-Class <class-name> (?arg$_1$)
(Define-Instance <instance-name> (<class-name>)
(Define-Axiom <axiom-name>

"text in natural language"
:Def <KIF-sentence>
:Iff-Def <KIF-sentence>
:Lambda-Body <KIF-sentence>
:Axiom-Def <KIF-sentence>
:Constraint <KIF-sentence>)

KIF examples

- **Relations**

```
(define-relation Conection (?comp1 ?comp2)
"Binary relation between two components. No component is a part of the other."
:def (and (component ?comp1) (component ?comp2)
          (not (part-of ?comp1 ?comp2) (not (part-of ?comp2 ?comp1))))
```

- **Functions**

```
(define-function Square (?n) --> value
"The square of a number is the product by itself "
:def (and (number ?n) (nonnegative-number ?value)
:lambda-body (* ?n ?n))
```

- **Axioms**

```
(define-axiom Two-numbers-are-prime-Cond
"If two numbers are prime between their, then they are different"
:= (forall ?num1 ?num2 (=> (Prime-between-numbers ?num1 ?num2)
                           (<> ?num1 ?num2)))))
```

The Frame Ontology

Knowledge Representation Ontology

Captures the representation primitives used in frame-based languages

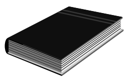
Ontological Commitments:

- **relations are tuples**
- **functions are special cases of relations**
- **classes are unary relations**

Implemented in KIF 3.0

http://www-ksl.stanford.edu

See the Ontology



Gruber, G. *A Translation approach to Portable Ontology Specifications.*
Knowledge Acquisition. Vol. 5. 1993.

Definition of the class CLASS in the Frame Ontology

(define-class Class (?class)

"A class can be thought of as a collection of individuals. Formally, a class is a unary relation, a set of tuples (lists) of length one. Each tuple contains an object which is said to be an instance of the class. An individual, or object, is any identifiable entity in the universe of discourse (anything that can be denoted by an object constant in KIF), including classes themselves. The notion of CLASS is introduced in addition to the relation vocabulary because of the importance of classes and types in knowledge representation practice. The notion of class and relation are merged to unify relational and object-centered representational conventions. Classes serve the role of `sorts' and `types'..."

:iff-def (and (Relation ?class)
(= (Arity ?class) 1))
:issues
((:see-also "In CycL, classes are called collections."
"In Loom, classes are called concepts."
"In KEE, classes are called classes."
"In Epikit, classes are not explicitly part of the language but are conventionally denoted by unary relations, or using a binary relation such as (ISA <instance> <class>).")))

http://www-ksl.stanford.edu

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'.")))
```

Most Well-Known Ontologies (II)

• Interlingua: KIF and PIF

• Knowledge Representation Ontology: The Frame Ontology

• Top Level Ontologies: PANGLOSS, Penman Upeer Level, Cyc, MikroKosmos
Guarino's and Sowa's Top Level Ontology

• Linguistic Ontologies: Generalized Upper Model, WordNet, SENSUS

• Engineering Ontologies: EngMath, PhysSys

• Knowledge Management Ontologies: $(KA)^2$ Ontologies, Reference Ontology

• Modeling Enterprise: Enterprise and TOVE

• Domain Ontologies: CHEMICALS

Generalized Upper Model: GUM

A general and domain-independent linguistic ontology

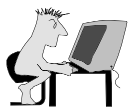
Linguistic categories are organized in taxonomies:

- **Concepts (processes)**
- **Relations (participants, circumstances)**

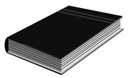
Does not represent information about:

- **relations between grammatical components**
- **context in which sentence applies**
- **relations bewtween emittor and receiver: inperative, questioning, ...**

Implemented in LOOM



http://www-darmstadt.gmd.de/publish/komet/gen-um/newUM.html



Bateman, J.; Magnini, B.; Fabris, G.
The Generalized Upper Model Knowledge Base: Organization and Use.
Towards Very Large Knowledge Bases. N. Mars. IOS Press. Amsterdam. 1995. 60-72.

WordNet

Lexical Database

Correspondence between terms and meanings (f, s)

Categories:

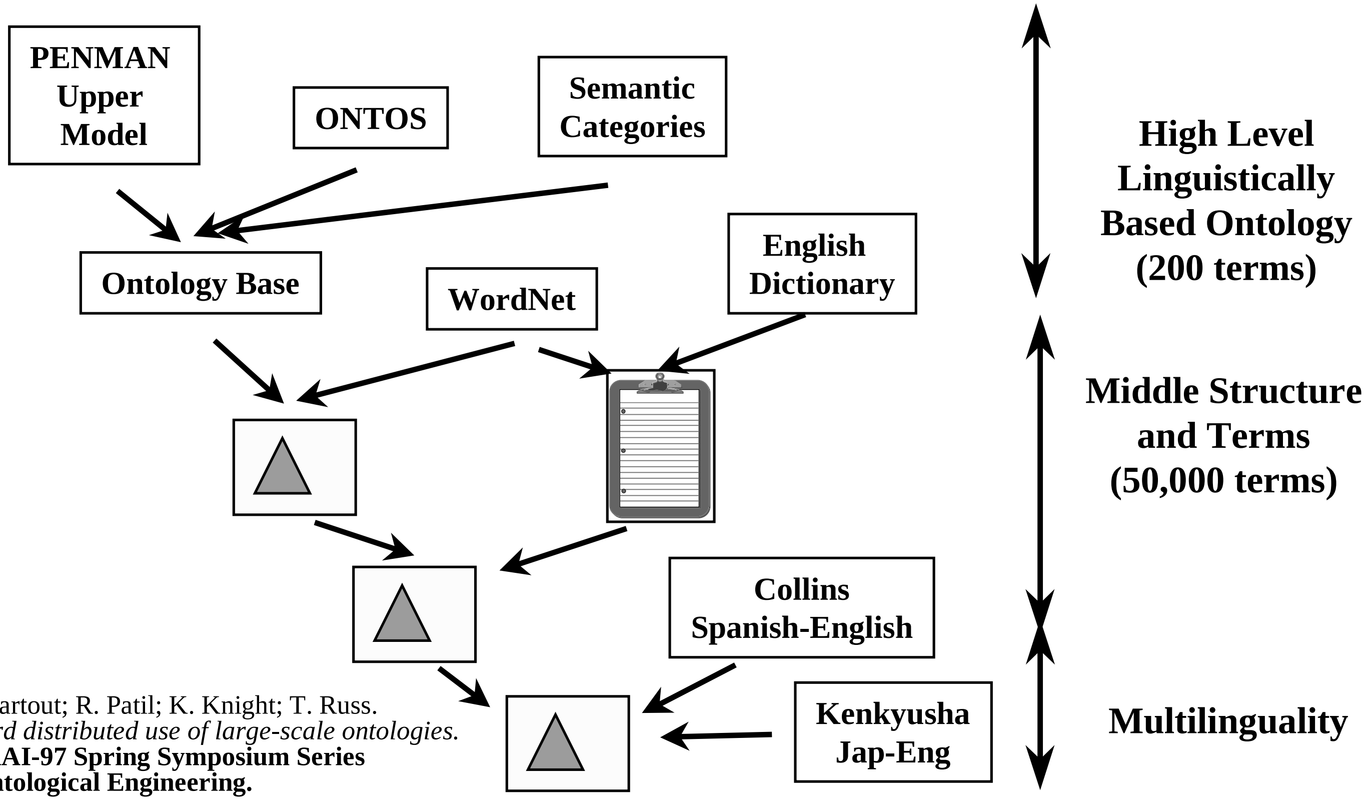
- **Nouns: organized in hierarchies**
- **Verbs: Implication relationships**
- **Adjectives and Adverbs: N-dimensionals hyperspaces**

Board --> {board, plank}
Board --> {board, committee}
Board --> {board} --> {“a person’s meals, provided regularly for money}

WordNet: http://www.tio.darpa.mil/Summaries95/B370-Princeton.html
EuroWordNet: http://www.cds.shef.ac.uk/research/groups/nlp/funded/eurowordnet.html

SENSUS Ontology

Developed by extracting and merging information from existing electronic resources



B. Swartout; R. Patil; K. Knight; T. Russ.
Toward distributed use of large-scale ontologies.
In **AAAI-97 Spring Symposium Series on Ontological Engineering.**

EngMath

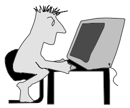
Mathematical modeling ontology

Includes:

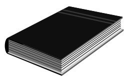
- **Scalar**
- **Vector**
- **Tensor quantities**
- **Physical dimensions**
- **Units of Measure**
- **Functions of quantities**
- **Dimensions of quantities**

Used by:

- **PhysSys Ontology**
- **SHADE project**



http://www-ksl.stanford.edu



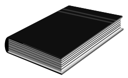
Gruber, T; Olsen, G.
An Ontology for Engineering Mathematics.
Fourth International Conference on Principles of Knowledge Representation.
DE by Doyle and Torasso. Morgan Kaufmann. 1994.

PhysSys

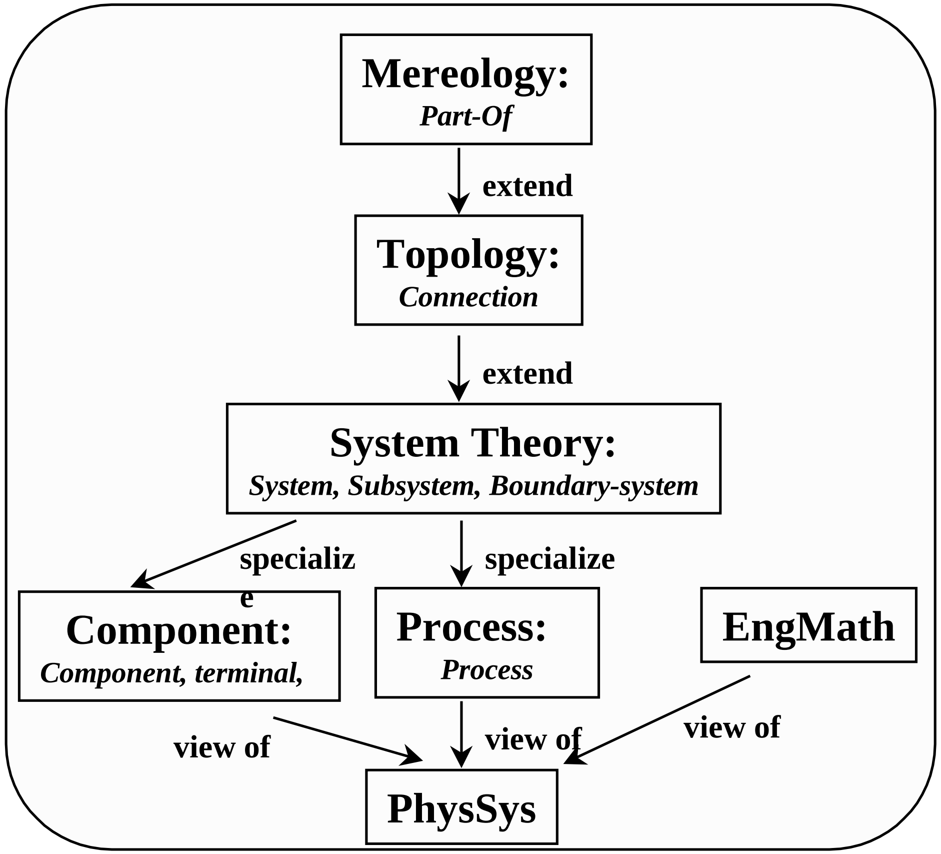
Engineering Ontology: Modeling, Simulating and Designing Physical Systems

Viewpoints

- **System Layout**
- **Physical Process Behaviour**
- **Descriptive Mathematical Relations**



Borst, P.; Benjamin, J.; Wilinga, B.; Akkermans, H.
An Application of Ontology Construction.
Workshop on Ontological Engineering.
ECAI'96. 17-28.



Enterprise

A set of ontologies for enterprise modeling

Meta-Ontology:
Entity
Relationship
Role
Actor
State of Affairs

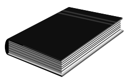
Activities and Processes:
Activity
Resource
Plan
Capability

Organization:
Organizational Unit
Legal Entity
Manage
Ownership

Strategy:
Purpose
Strategy
Help Achieve
Assumption

Marketing:
Sale
Product
Vendor
Customer
Market

http://www-ksl.stanford.edu



Uschold, M.; Grüninger, M. *ONTOLOGIES: Principles, Methods and Applications.* **Knowledge Engineering Review**. Vol. 11; N. 2; June 1996.

CHEMICALS

Built using METHONTOLOGY and ODE

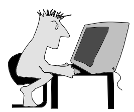
Codified in Ontolingua

A set of domain ontologies in the domain of chemical substances:

- **CHEMICAL-ELEMENTS:**
  - **16 classes**
  - **21 relations**
  - **3 functions**
  - **103 instances**
  - **27 axioms**
- **CHEMICAL-CRYSTALS:**
  - **19 classes**
  - **8 relations**
  - **1 functions**
  - **66 instances**
  - **26 axioms**

Extensions of:

- **Standard Units**
- **Standard Dimensions**



http://www-ksl.stanford.edu

Fernández-López. **CHEMICALS: Ontología de Elementos Químicos**. Facultad de Informática. UPM. December 1996.

Ontologies “versus” Knowledge Bases (I)

Features of the language used to codify the knowledge

- **Expressive**
- **Declarative**
- **Portable**
- **Domain independent**
- **Semantically well defined**

CycL

KIF

LOOM

Ontolingua

CycL: Lenat, D.B., Guha, R.V.; **Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project**. Addison-Wesley Publishing Company, Inc. CA. 1990.

KIF: Genesereth, M.; Fikes, R. *Knowledge Interchange Format*. Version 3.0. Reference Manual. **Report Logic-92-1.** Computer Science Department. Stanford University. CA. 1992.

LOOM: MacGregor, R. *The evolving technology of classification-based knowledge representation systems*. In J. Sowa, Ed. **Principles of Semantic Networks: Explorations in the Representation of Knowledge**. San Mateo, CA. Morgan Kaufmann. 1

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

Ontologies “versus” Knowledge Bases (II)

Goal of knowledge codification

Knowledge Base

PART-OF (cylinder, engine)
PART-OF (battery, engine)

a) Definitions in a Knowledge Base

Physical-Devices

Concept: *Component*

Relation: *Part-of*
Number of Arguments: *2*
Type of Arg. #1: *Component*
Type of Arg. #2: *Component*

Mechanical-Devices

Concept: *Cylinder*
Subclass-of: *Component*
Part-of: *Engine*

Concept: *Battery*
Subclass-of: *Component*
Part-of: *Engine*

Concept: *Engine*
Subclass-of: *Component*

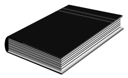
b) Definitions in ontologies

Requirements Specification

Ontologies “versus” Knowledge Bases (III)

Functional characterization

- Does it express the consensus knowledge of a community of people?
- Do people use it as a reference of precisely defined terms?
- Is the language used expressive enough for people to say what they want to say?
- Is it stable?
- Can it be used to solve a variety of different sorts of problems?
- Can it be used as a starting point to construct multiple (sorts of) applications?



By A. Farquhar at ontolingua@hpp.stanford.edu electronic mailing list