Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web

Mariano Fernández-López
University Foundation San Pablo CEU

Ontological Engineering

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Ontological Engineering

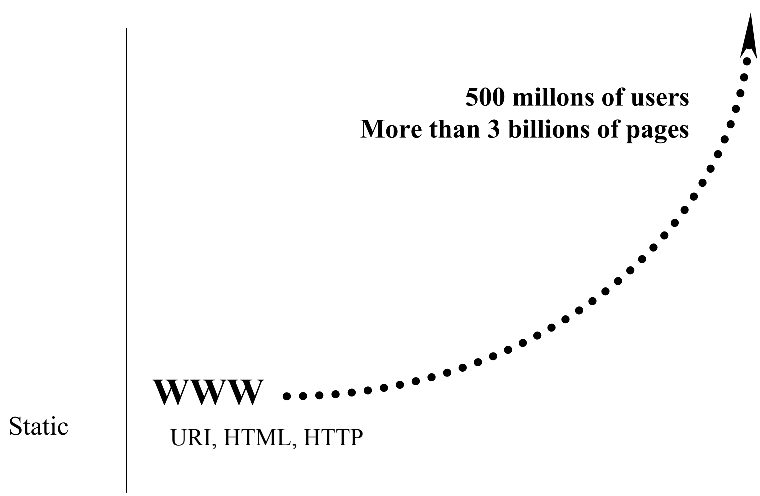
Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Table of Contents

The Role of Ontologies in the Semantic Web

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

The problem: Information overload on the WEb



The current Web is based on HTML

Arab

الاسم: الهندسة فى علم التطو ر
المؤلفون: آسنسيون غومز-بيرز
السّعر: $74.95
المنتج: الكتاب

```
الهندسة فى علم التطو ر <b>:الاسم</b><BR>
آسنسيون غومز-بيرز <b>:المؤلفون</b><BR>
$74.95 <b>:السّعر</b><BR>
المنتج <b>:الكتاب</b><BR>
```

Japanese

タイトル: 存在論工学
著者: アスンスィオン　ゴメスペレス
価格: $74.95
産品: 本

```
<b>タイトル:</b> 存在論工学<BR>
<b>著者:</b> アスンスィオン　ゴメスペレス... <BR>
<b>価格:</b> $74.95<BR>
<b>産品:</b> 本<BR>
```

Norwegian

Skjøte: Ontological Ingeniørarbeid
Forfatter: Overtakelse Gómez-Pérez...
Pris: 74.95€
Produkt: Bok

```
<b>Skjøte:</b> Ontological Ingeniørarbeid<BR>
<b>Forfatter:</b> Overtakelse Gómez-Pérez... <BR>
<b>Pris:</b> 74.95€<BR>
<b>Produkt:</b> Bok<BR>
```

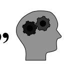
English

Title: Ontological Engineering
Authors: Asunción Gómez-Pérez...
Price: $74.95
Product: Book

```
<b>Title:</b> Ontological Engineering <BR>
<b>Authors:</b>  Asunción Gómez-Pérez... <BR>
<b>Price:</b> $74.95<BR>
<b>Product:</b> Book<BR>
```

.- HTML is useful for browsing the information
.- Content is language-dependent
.- High cost for keeping the information up-to-date

XML allows the creation of metada with “meaning”



Arab

الاسم: الهندسة فى علم التطو ر
المؤلفون: آسنسيون غومز-بيرز
السّعر: $74.95
المنتج: الكتاب

```
<الاسم>الهندسة فى علم التطو ر</الاسم>
<المؤلفون>آسنسيون غومز-بيرز</المؤلفون>
<السّعر>$74.95</السّعر>
<المنتج>الكتاب</الكتاب>
```

English

Title: Ontological Engineering
Authors: Asunción Gómez-Pérez...
Price: $74.95
Product: Book

```
<Title>Ontological Engineering</Title>
<Author>Asunción Gómez-Pérez...</Author>
<Price>$74.95</Price>
<Product>Book</Product>
```

¿What do the tags mean for the machine?

The problem of choosing information

.- Find the information
.- Extract relevant information
.- Interpretation by human users
.- Sinthesis

The problem of content agregation: From Madrid to Tokyo

.- Content in different languages (Spanish, English, Japanesse,...)
.- Find out relevant information from heterogeneous sources
  .- Extract
  .- Interpretation
  .- Agregation
.- Consistency of the information

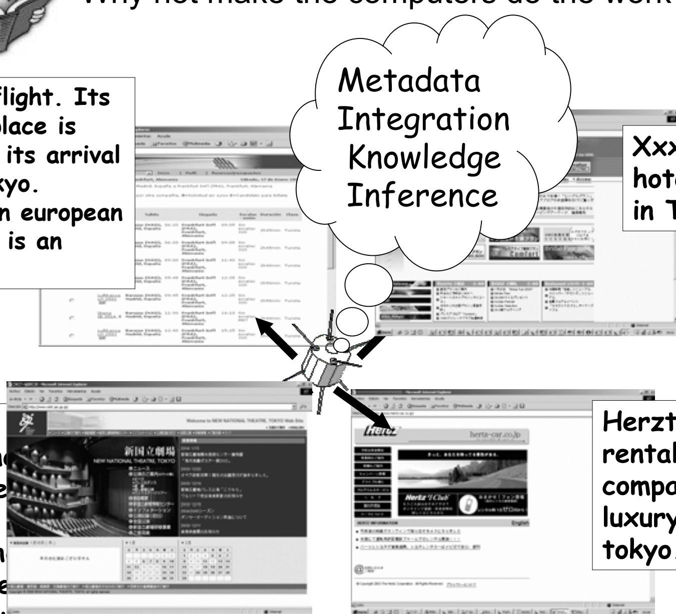
What was the Web intended to be?

"... a goal of the Web was that, if the interaction between person and hypertext could be so intuitive that the machine-readable information space gave an accurate representation of the state of people's thoughts, interactions, and work patterns, then machine analysis could become a very powerful management tool, seeing patterns in our work and facilitating our working together through the typical problems which beset the management of large organizations."

[Berners-Lee 1996]

Why not make the computers do the work?

IBXX is a flight. Its departure place is Madrid and its arrival place is Tokyo. Madrid is an european city. Tokyo is an asian city.



Xxx is a hotel placed in Tokyo

The new national the[...] is a theate[...] located in Tokyo.It h[...] peformance every Satu[...]

Herzt is a rental car company with luxury cars in tokyo.

What is the Semantic Web?

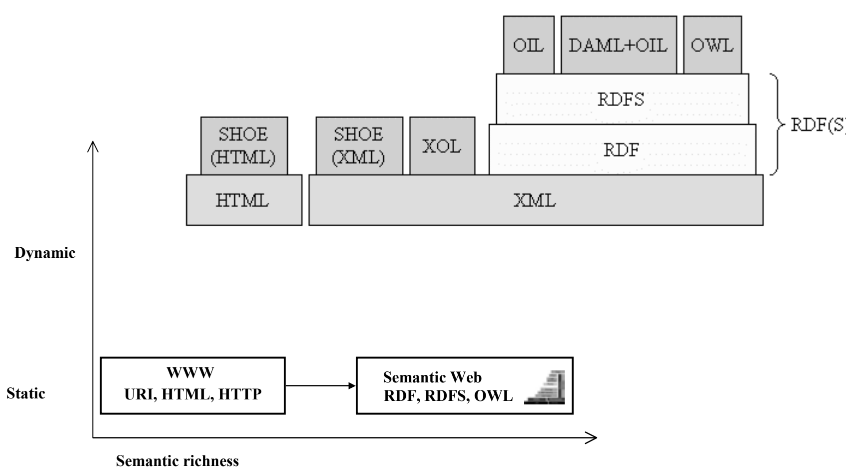
"The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the Web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications."

Ontologies

Annotation

Hendler, J., Berners-Lee, T., and Miller, E.
Integrating Applications on the Semantic Web, 2002,
http://www.w3.org/2002/07/swint.html

Semantic Web Languages



Ontologies and Metadata

Ontologies

xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
xmlns:NS0='http://www.esperonto.net/semanticportal/RDFS/Person_Ontology#'
xmlns:NS1='http://www.esperonto.net/semanticportal/RDFS/Organization_Ontology#'

Person —Belongs_To→ Organization
Organization —Has_contact_Person→ Person
Associate Prof. —Subclass of→ Person
Partner —Subclass of→ Organization

Instance of

Instance of

Annotation (RDF)

```
<rdf:Description rdf:about='Asunción Gómez-Pérez'>
<rdf:type rdf:resource='Associate Prof'/>
<NS0:Full_Name>A_GomezPerez</NS0:Full_Name>
<NS0:Belongs_To>UPM</NS0: Belongs_To >
<NS0:e-mail>asun@fi.upm.es</NS0:e-mail>
```

```
<rdf:Description rdf:about='UPM'>
<rdf:type rdf:resource='Partner'/>
<NS1:Acronym>UPM</NS1:Acronym>
<NS1:Has_Contact_Person>Asunción Gómez-Pérez
        </NS1:Has_Contact_Person >
```

Web Page

Full Name	Asuncion Gomez-Perez
e-mail	asun@fi.upm.es
Photo	

Asunción Gómez-Pérez is contact person UPM (Partner).
Asunción Gómez-Pérez belongs to UPM (Partner).

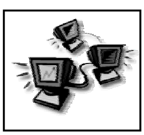
Full Name	Universidad Politécnica de Madrid
Acronym	UPM
Logo	UPM

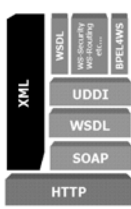
UPM has contact person Asunción Gómez-Pérez (Associate Professor).
UPM participates in Esperonto (Project).
UPM team is formed by :

URL http://www.esperonto.net http://www.esperonto.net

Web Services

Peer Web for information exchange between machines





To describe control flows
To register Services
Service Access Interfaz
Communication Protocol

Terminological problems

Dynamic — Web Services UDDI, WSDL, SOAP

Declaratively described Program Access Interfaces that are accessible through the Web

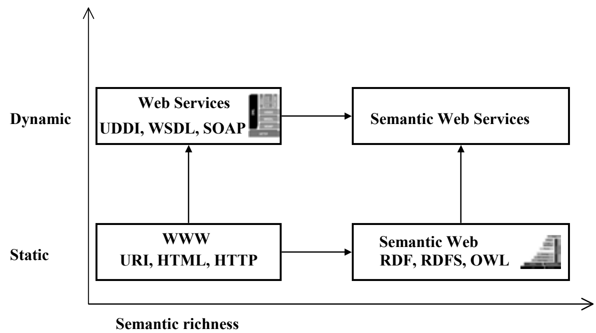
Static — WWW URI, HTML, HTTP → Semantic Web RDF, RDFS, OWL

Semantic richness

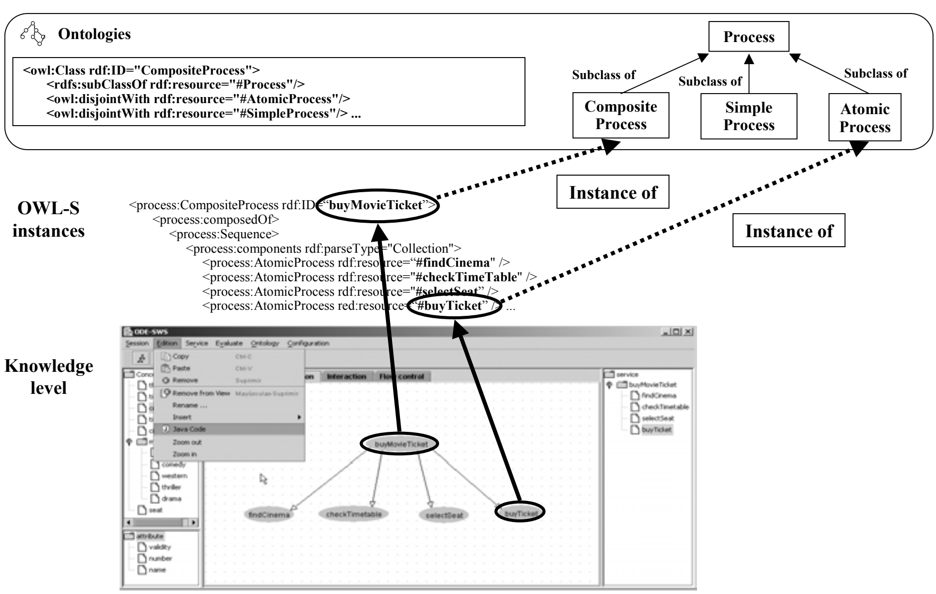
The Semantic Web and the Semantic Web Services

Web Services that describe their properties and capabilities using

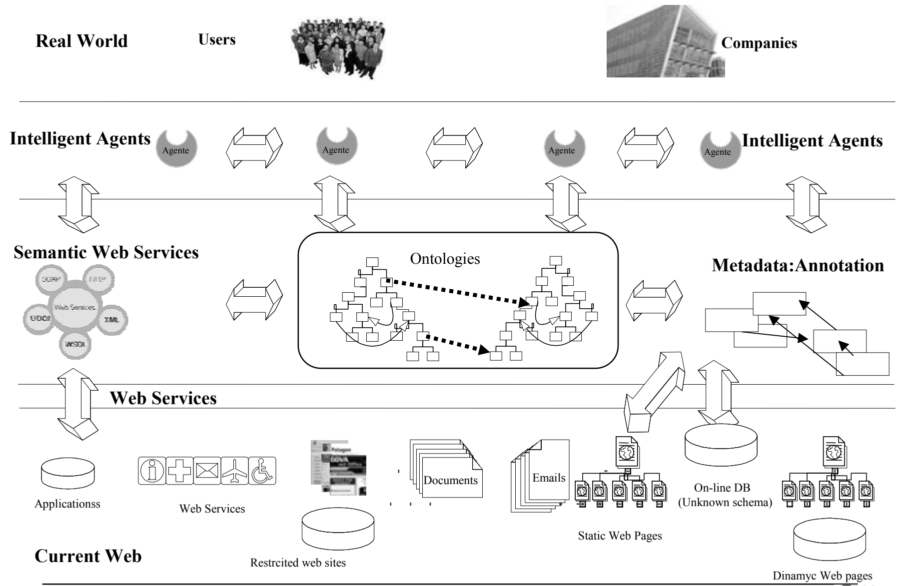
the vocabulary of an ontology, and they are expressed in some semantic markup language



Semantic Web Services



A Semantic Web Scenario



Ontological Enginnering for the Semantic Web

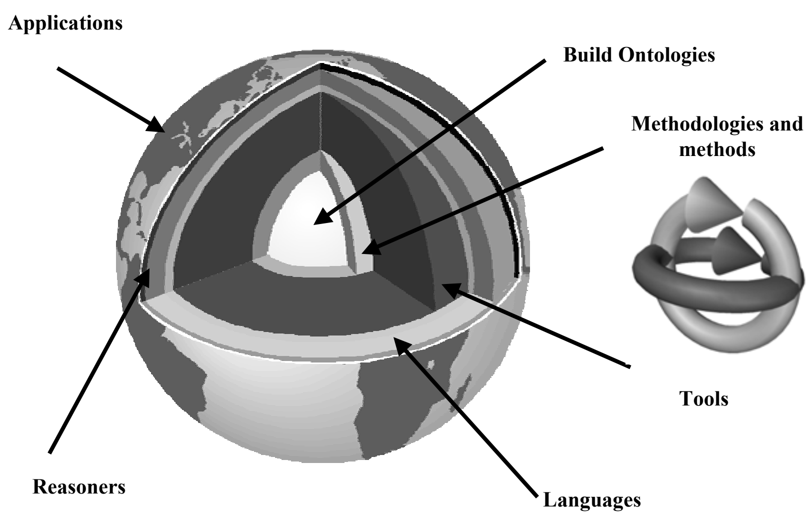


Table of Contents

Theoretical Foundations of Ontologies

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Main References

Gómez-Pérez, A.; Fernández-López, M.; Corcho, O. **Ontological Engineering.** *Springer Verlag. 2003*

http://www.ontoweb.org

Deliverables
•D1.1
•D1.2
•D1.3
•D1.4
•D1.5

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.

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

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

Outline

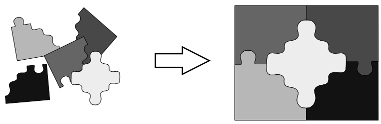
The Knowledge Sharing Initiative
Definitions of Ontologies
Modeling of Ontologies
Types of Ontologies
Libraries of Ontologies
Ontological Commitments

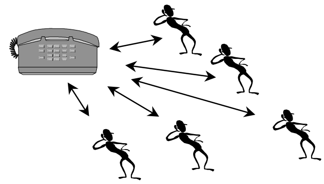
⇧

Reuse and Sharing

Reuse means to build new applications assembling components already built

Sharing is when different applications use the some resources





Advantages:

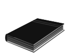
- **Less money**
- **Less time**
- **Less resources**

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.

Outline

The Knowledge Sharing Initiative
Definitions of Ontologies
Modeling of Ontologies
Types of Ontologies
Libraries of Ontologies
Ontological Commitments

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 (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 RE, Finin T, Gruber TR, Senator T, Swartout WR (1991) *Enabling technology for knowledge sharing*. **AI Magazine** 12(3):36–56
2. "An ontology is an explicit specification of a conceptualization"	Gruber TR (1993a) *A translation approach to portable ontology specification*. **Knowledge Acquisition** 5(2):199–220
3. "An ontology is a formal, explicit specification of a shared conceptualization"	Studer R, Benjamins VR, Fensel D (1998) *Knowledge Engineering: Principles and Methods*. **IEEE Transactions on Data and Knowledge Engineering** 25(1-2):161–197
4. "A logical theory which gives on explicit, partial account of a conceptualization"	Guarino N, Giaretta P (1995) *Ontologies and Knowledge Bases: Towards a Terminological Clarification*. In: Mars N (ed) **Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95)**. University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, pp 25–32
5. "A set of logical axioms designed to account for the intended meaning of a vocabulary"	Guarino N (1998) *Formal Ontology in Information Systems*. In: Guarino N (ed) 1st International Conference on **Formal Ontology in Information Systems (FOIS'98)**. Trento, Italy. IOS Press, Amsterdam, pp 3–15

Definitions of Ontologies (II)

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.**

Outline

⇧

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**

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

Primitivas necesarias para modelizar conocimientos disjuntos en taxonomías

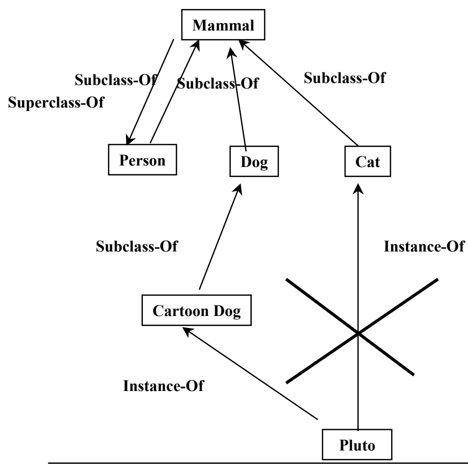
class-Partition: Conjunto de clases que son disjuntas entre sí

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.

How to build taxonomies (II)

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

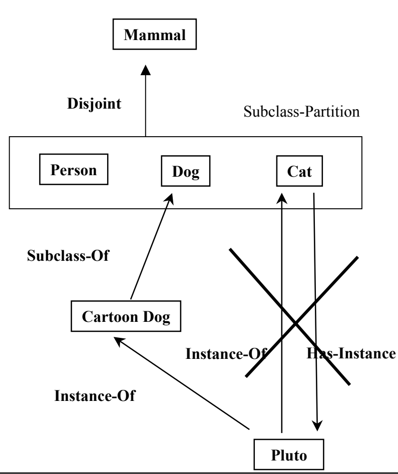


Pluto could be an instance of cat and dog

Semantic Error

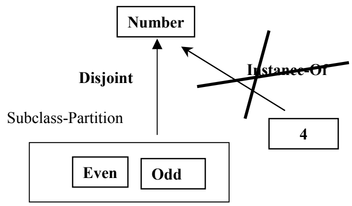
How to build taxonomies (III)

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

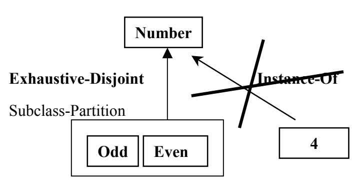
How to build taxonomies (IV)



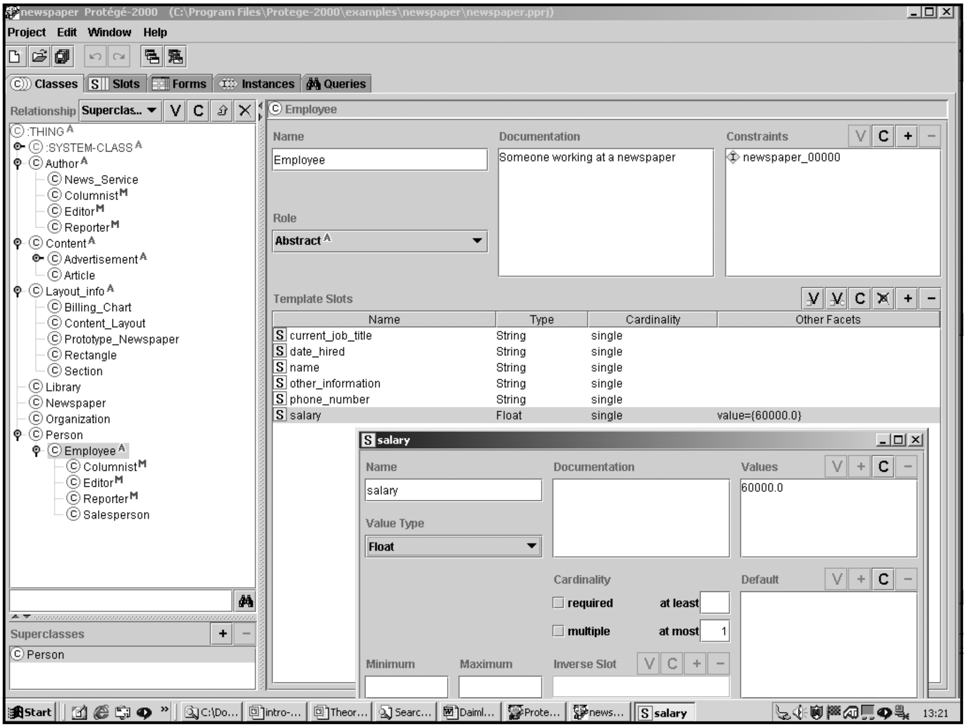
Four is an instance of Par

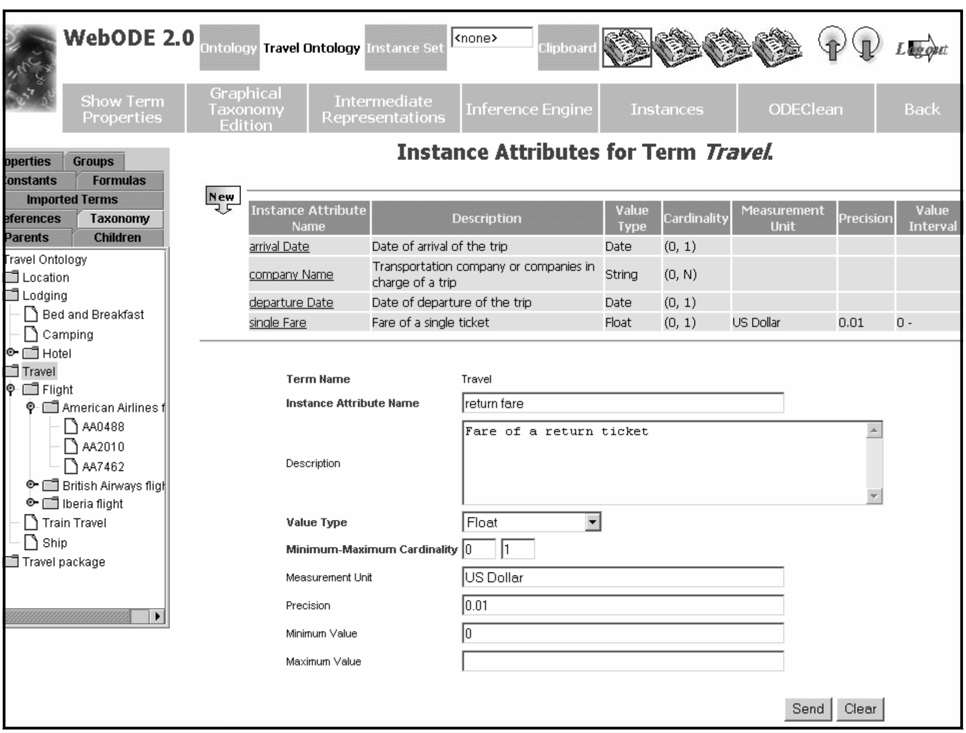
A. Gómez-Pérez. *Evaluation of Ontologies.* **International Journal of Intelligent Systems.** Vol. 16, Nº3. March 2001. PP391-410

How to build taxonomies (V)



Four is an instance of something in the partition


newspaper Protégé-2000 (C:\Program Files\Protege-2000\examples\newspaper\newspaper.pprj)
Project Edit Window Help
Classes Slots Forms Instances Queries
Relationship Superclas...
:THING
:SYSTEM-CLASS
Author
News_Service
Columnist
Editor
Reporter
Content
Advertisement
Article
Layout_info
Billing_Chart
Content_Layout
Prototype_Newspaper
Rectangle
Section
Library
Newspaper
Organization
Person
Employee
Columnist
Editor
Reporter
Salesperson
Superclasses
Person
Employee
Name
Employee
Documentation
Someone working at a newspaper
Constraints
newspaper_00000
Role
Abstract
Template Slots
Name Type Cardinality Other Facets
current_job_title String single
date_hired String single
name String single
other_information String single
phone_number String single
salary Float single value={60000.0}
salary
Name
salary
Documentation
Values
60000.0
Value Type
Float
Cardinality
required at least
multiple at most 1
Default
Minimum Maximum Inverse Slot
Start
13:21


WebODE 2.0
Ontology Travel Ontology Instance Set <none> Clipboard
Logout
Show Term Properties
Graphical Taxonomy Edition
Intermediate Representations
Inference Engine
Instances
ODEClean
Back
Instance Attributes for Term Travel.
operties Groups
onstants Formulas
Imported Terms
eferences Taxonomy
Parents Children
Travel Ontology
Location
Lodging
Bed and Breakfast
Camping
Hotel
Travel
Flight
American Airlines f
AA0488
AA2010
AA7462
British Airways fligh
Iberia flight
Train Travel
Ship
Travel package
New
Instance Attribute Name | Description | Value Type | Cardinality | Measurement Unit | Precision | Value Interval
arrival Date | Date of arrival of the trip | Date | (0, 1)
company Name | Transportation company or companies in charge of a trip | String | (0, N)
departure Date | Date of departure of the trip | Date | (0, 1)
single Fare | Fare of a single ticket | Float | (0, 1) | US Dollar | 0.01 | 0 -
Term Name Travel
Instance Attribute Name return fare
Description Fare of a return ticket
Value Type Float
Minimum-Maximum Cardinality 0 1
Measurement Unit US Dollar
Precision 0.01
Minimum Value 0
Maximum Value
Send Clear

What does an Explicit Ontology look Like?

Highly informal: → in natural language

An html ontology for linking documents

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

Example

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

Principles for the Design of Ontologies (I)

Clarity:

To communicate the intended meaning of defined terms

Coherence:

To sanction inferences that are consistent with definitions

Extendibility:

To anticipate the use of the shared vocabulary

example

Minimal Encoding Bias:

To be independent of the symbolic level

Minimal Ontological Commitments:

To make as few claims as possible about the world

• Gruber, T.; *Towards Principles for the Design of Ontologies.* **KSL-93-04.** Knowledge Systems Laboratory. Stanford University. 1993

Clarity

An ontology should communicate effectively the intended meaning of defined terms. Definitions should be objective. Definitions can be stated on formal axioms, and a complete definition (defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). . .

No Clarity

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Subclass-Of Travel Thing)
      (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)))
```


Clarity

Clarity

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

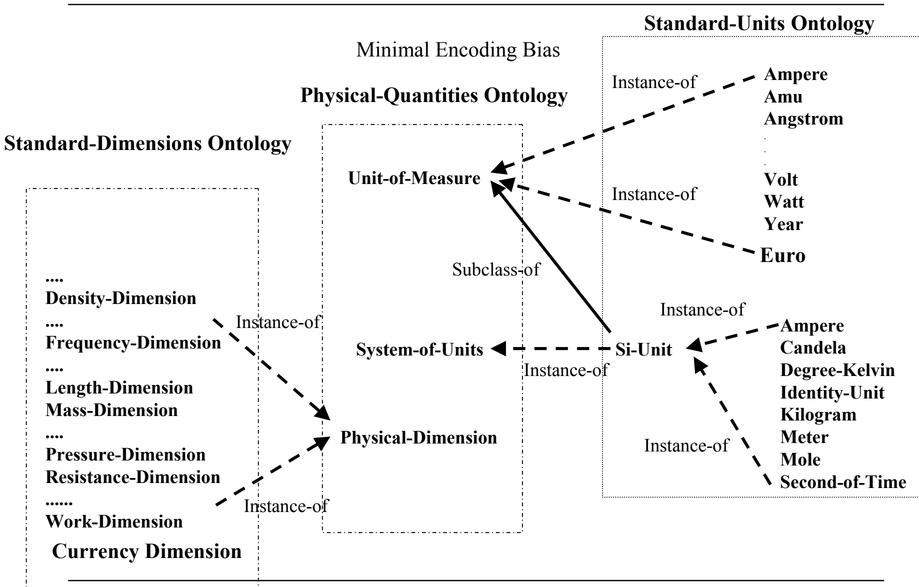
Minimal Encoding Bias

> ***"The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding".***

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

No minimal Encoding Bias

Minimal Encoding Bias



Minimal Encoding Bias

```
(singleFare ?travel Number)
```

should be substituted by:

```
(singleFare ?travel CurrencyQuantity)
```


Extensibility

> ***"One should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions".***

- **Currency dimension**
- **Definition of currencies**
- **Relationship between currencies**

```
(define-individual Euro (Unit-of-Measure)
   "An Euro is the currency on the European Union"
:= (* 0,96 USDollar)
:axiom-def
  (= (Quantity.dimension Euro) CurrencyDimension))
```

Coherence

"An ontology should be coherent: that is, it should sanction inferences that are consistent with the definitions.[…] If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent".

```
(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)))))))
(define-instance Madrid (EuropeanLocation))
(define-instance NewYork (USALocation))
```


Minimal Ontological Commitments

"Since ontological commitment is based on the consistent use of the vocabulary, ontological commitment can be minimized by specifyingthe weakest theory and defining only those terms that are essential to the communication of knowledge consistent with the theory".

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 ( .... )
:iff-def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date))
:def
  (and (singleFare ?travel Number)
       (companyName ?travel String)))
```

- **What is a date?**
- **Absolute/relative date?**
- **could be an interval?**
- **date= month + year**
- **date= day + month +year**
- **date = month +day +year**

Principles for the Design of Ontologies (IV)

- **The representation of disjoint and exhaustive knowledge.** If the set of subclasses of a concept are disjoint, we can define a disjoint decomposition. The decomposition is exhaustive if it defines the superconcept completely.

- **To improve the understandability and reusability of the ontology,** we should implement the ontology trying to minimize the syntactic distance between sibling concepts.

- **The standardization of names.** To ease the understanding of the ontology the same naming conventions should be used to name related terms.

Arpírez JC, Gómez-Pérez A, Lozano A, Pinto HS (1998) *(ONTO)^{2}Agent: An ontology-based WWW broker to select ontologies*. In: Gómez-Pérez A, Benjamins RV (eds) **ECAI'98 Workshop on Applications of Ontologies and Problem-Solving Methods**. Brighton, United Kingdom, pp 16–24

Approaches for Modeling Ontologies

- **Using frames and first order logic**
- **Using description logic**
- **Using UML**
- **Using the entity relationship model**

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

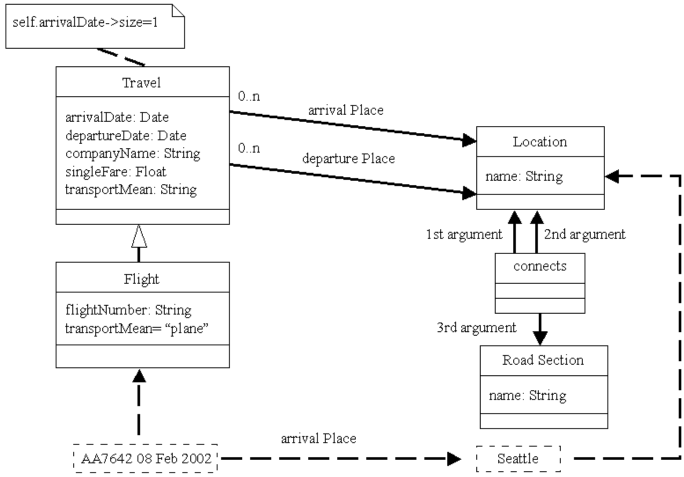
```
(defconcept Travel
  "A journey from place to place"
:is-primitive
  (:and
   (:all arrivalDate Date)(:exactly 1 arrivalDate)
   (:all departureDate Date)(:exactly 1
departureDate)
   (:all companyName String)
   (:all singleFare Number)(:at-most singleFare 1)))
```

```
(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))
```

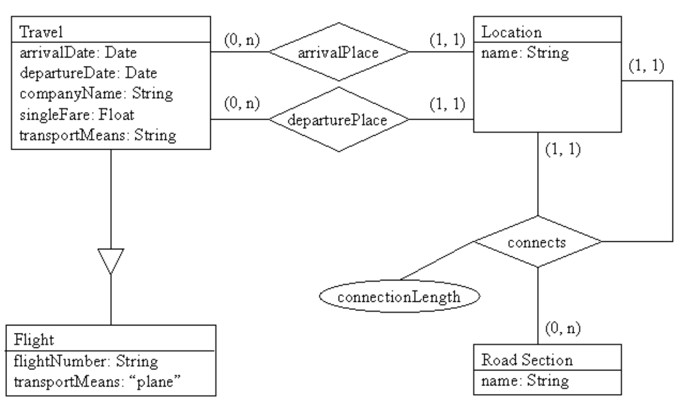
```
(defrelation Pays
:is
 (:function (?room ?Discount)
  (- (Price ?room) (/(*(Price ?room) ?Discount) 100)))
:domains (Room Number)
:range Number)
```

```
(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)))))
```

Using UML for Modeling Ontologies



Using the Entity Relationship Model for Modeling Ontologies



Conclusions on the 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

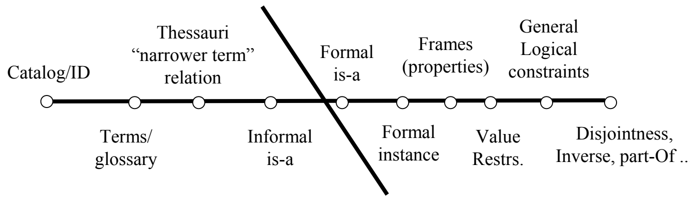
•A domain model is not necessarily an ontology only because it is written in Ontolingua or OWL, for the same reasons that we cannot say that a program is a knowledge-based system because it is written in Prolog

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

Outline

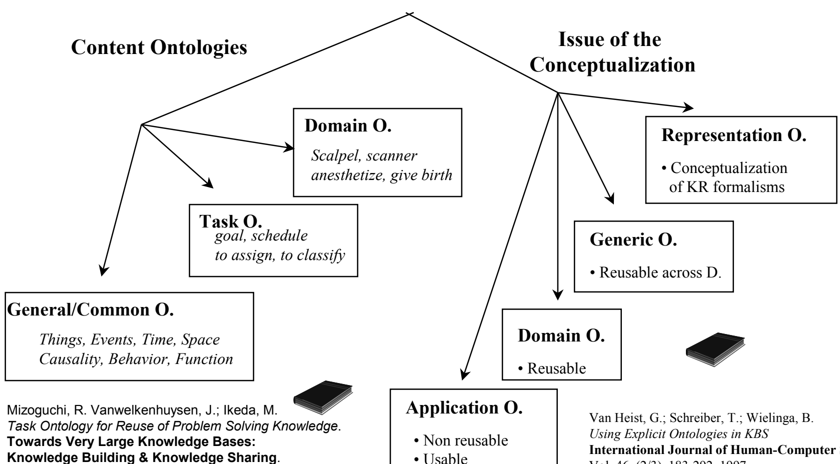
⇧

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.

Types of Ontologies



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)

•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/*

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

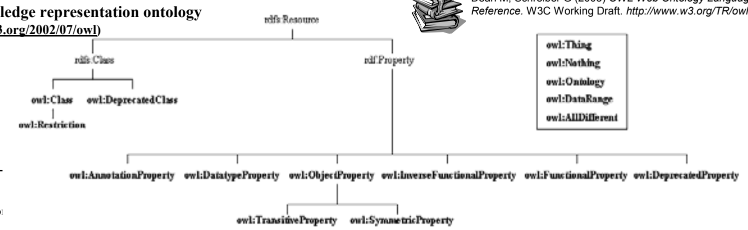
Horrocks I, Fensel D, Harmelen F, Decker S, Erdmann M, Klein M (2000) *OIL in a Nutshell.* In: Dieng R, Corby O (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 1–16

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

Horrocks I, van Harmelen F (eds) (2001) *Reference Description of the DAML+OIL (March 2001) Ontology Markup Language.* Technical report. *http://www.daml.org/2001/03/reference.html*

•**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/*



Top-level Ontologies

•**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

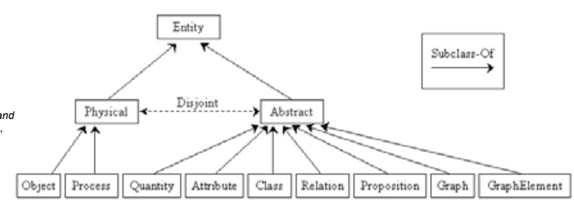
•Gangemi A, Guarino N, Oltramari A (2001) *Conceptual analysis of lexical taxonomies: the case of Wordnet top-level*. In: Smith B, Welty C (eds) International Conference on Formal Ontology in Information Systems (FOIS'01). Ogunquit, Maine. ACM Press, New York, pp 3–15

•**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

Linguistic Ontologies

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

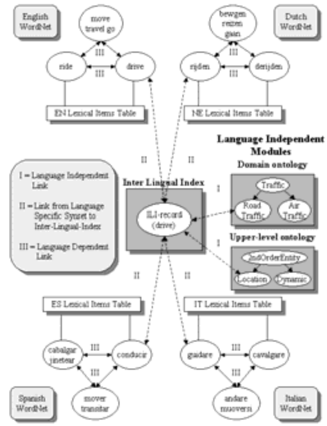
•Miller GA (1995) *WordNet: a lexical database for English*. Communications of the ACM 38(11):39–41

•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

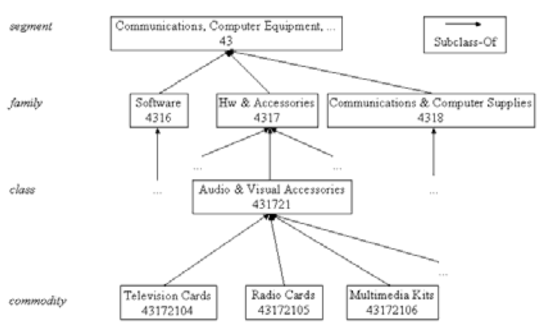
•**The United Nations Standard Products and Services Codes (UNSPSC)**
(**http://www.unspsc.org/**)

•**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

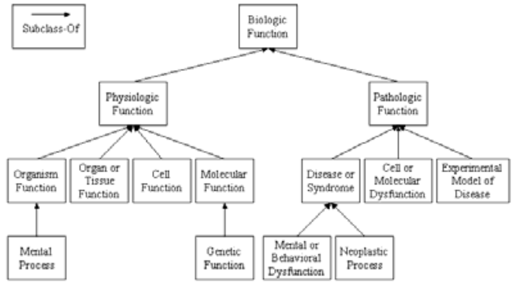
•**GALEN** (http://www.opengalen.org/)

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



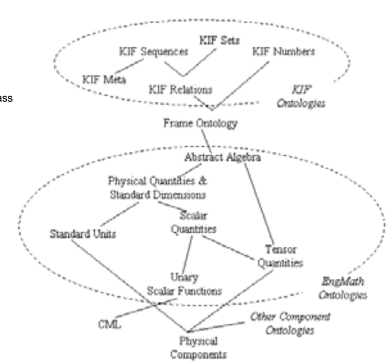
Domain Ontologies: Engineering Ontologies

•**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

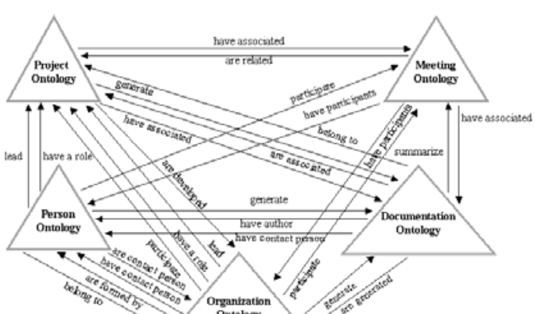
- Enterprise Ontologies
  - Enterprise Design Ontology
  - Project Ontology
  - Material Flow Ontology
  - Business Process Ontology
- Derivative Ontologies
  - Transportation Ontology
  - Inventory Ontology
  - Quality Ontology
  - Product Design Ontology
  - Goals Ontology
  - Scheduling Ontology
  - 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/)

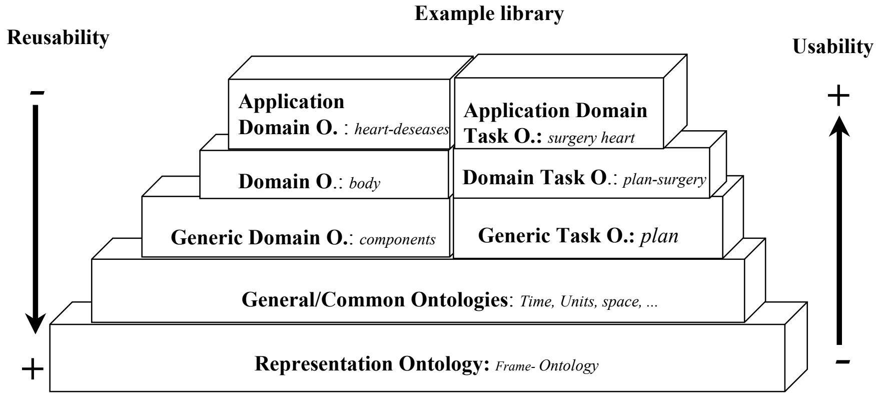


Outline

Libraries of Ontologies (I)

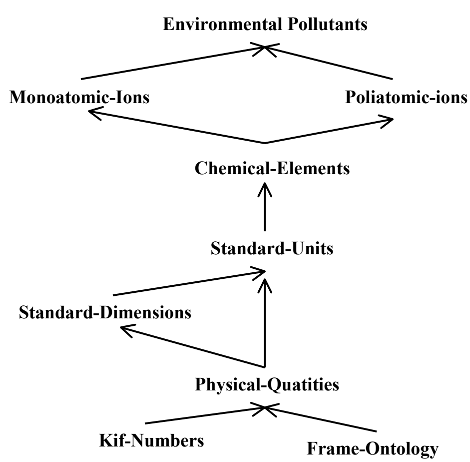
DAML ontology library	http://www.daml.org/ontologies/	see library
Protege ontology library	http://protege.stanford.edu/ontologies.html	
Ontolingua ontology library	http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html	
WebOnto ontology library	http://webonto.open.ac.uk	
SHOE ontology library	http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html	
WebODE ontology library	http://webode.dia.fi.upm.es/	
$(KA)^2$ ontology library	http://ka2portal.aifb.uni-karlsruhe.de/	

Libraries of Ontologies (II)



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

Relationship between Ontologies in the Library



Outline

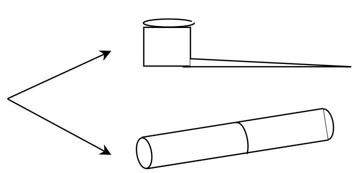
Ontological Commitments

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?



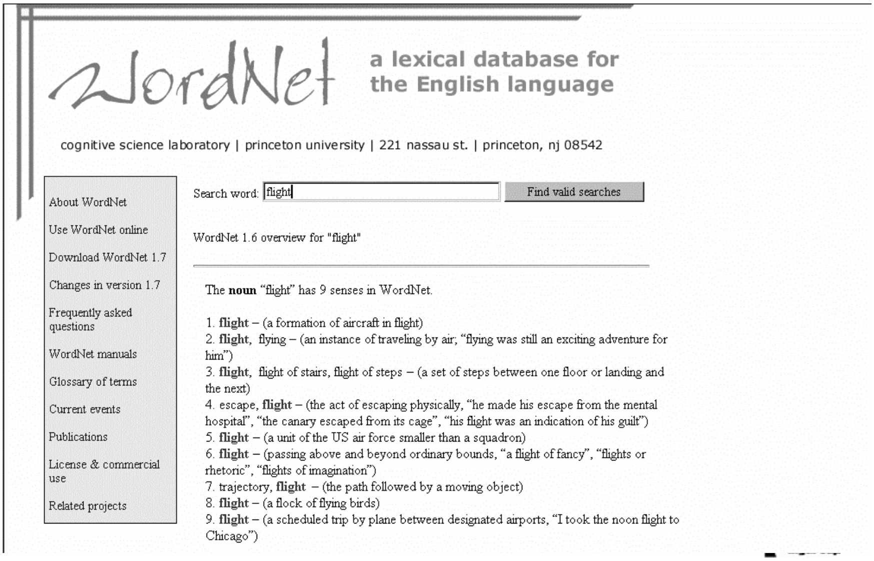
9 definitions of the term flight from wordnet

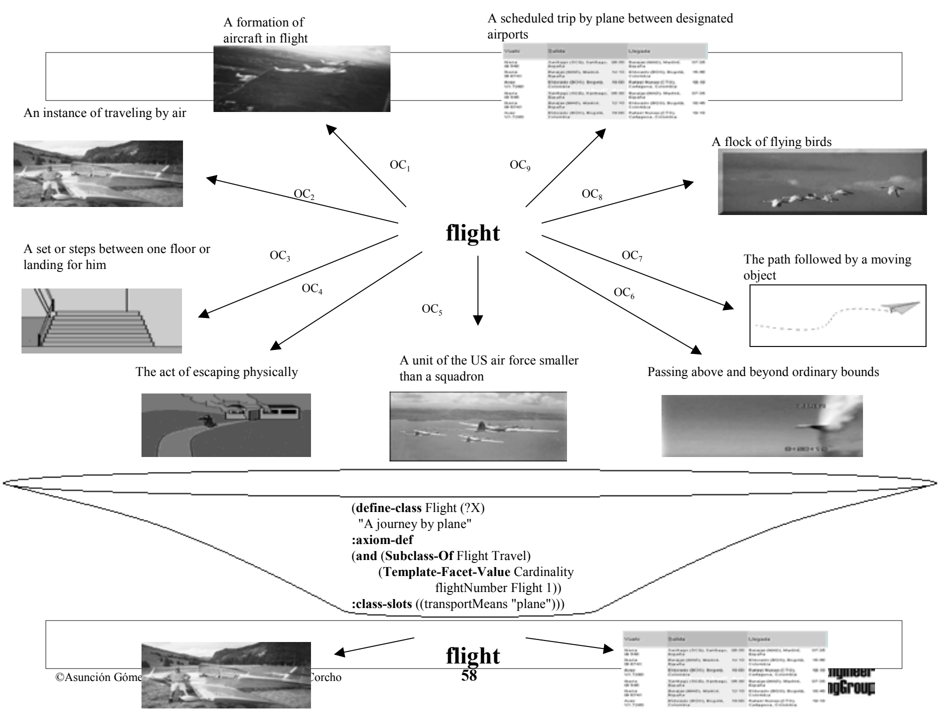
Identification of the ontological commitment

• 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





What is an Ontology?

Shared understanding of a domain

Repository of vocabulary

- Formal definitions
- Informal definitions

Table of Contents

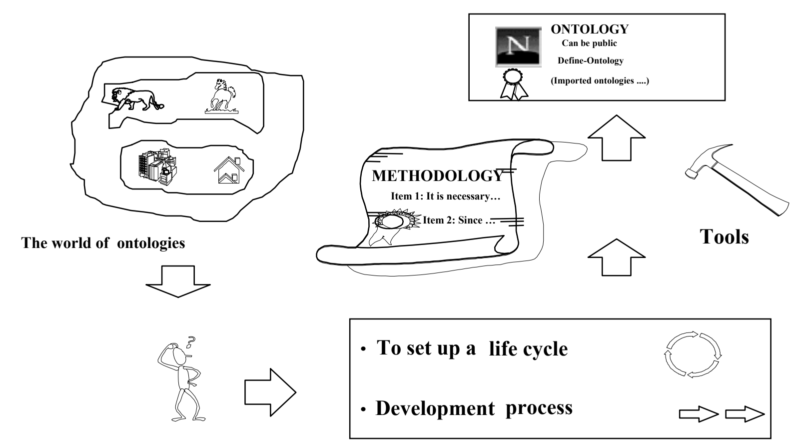
Ontological Engineering: Methodologies and Tools

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Outline

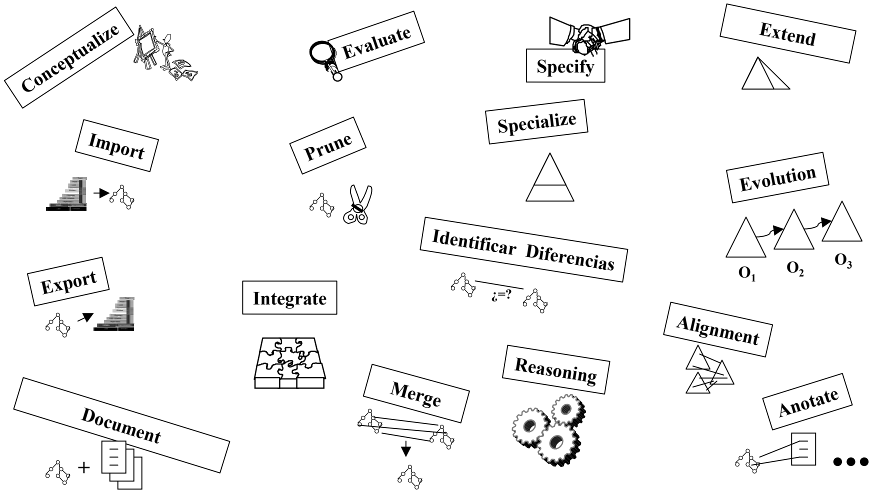
The Ontology Development Process
Methodologies for building ontologies
Methods and tools for
- **Conceptualizing**
- **Learning ontologies**
- **Merging**
- **Evaluating**
- **Evolving**

The Framework

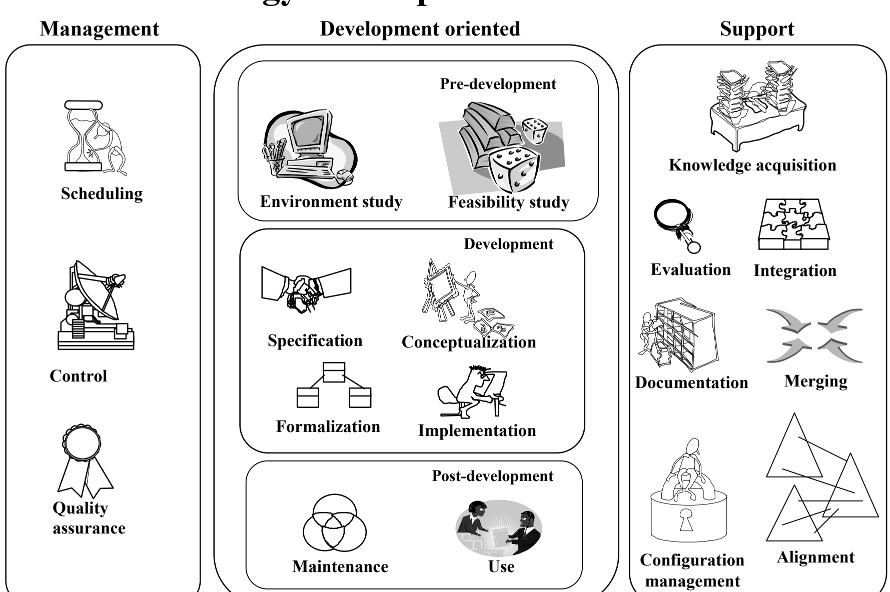


Gómez-Pérez, A. *Knowledge Sharing and Reuse*. **In the Handbook of Applied Expert Systems.** CRC Press. 1998.

Building ontologies

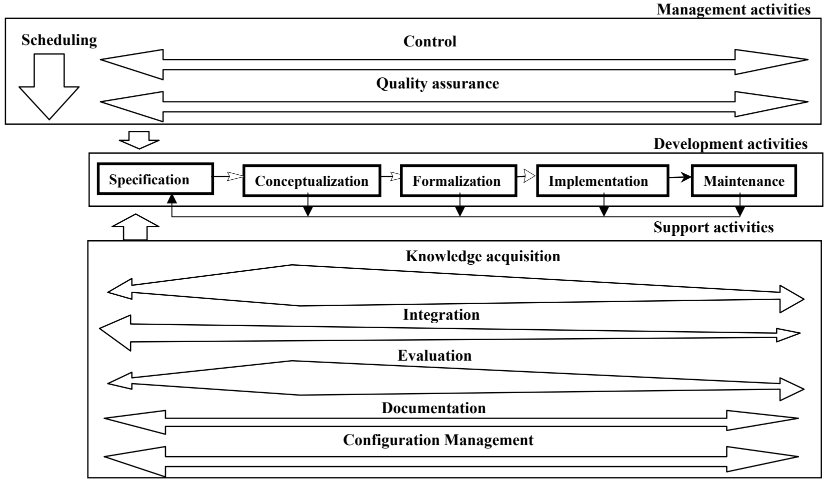


Ontology Development Process



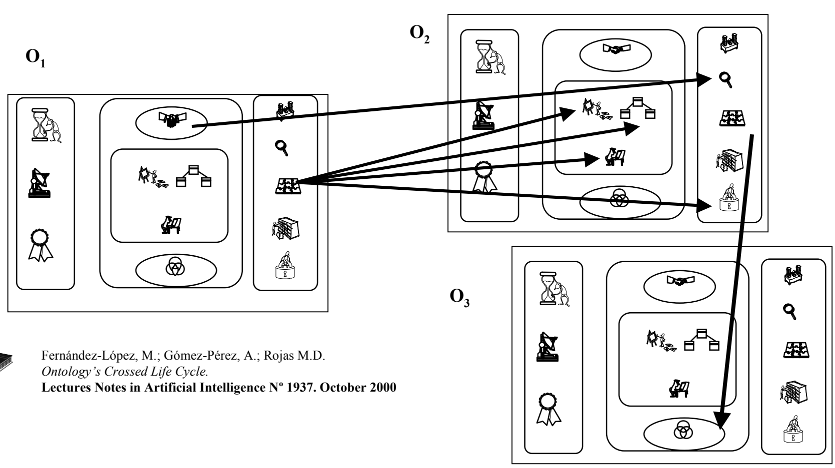
Intra-dependencies

Ontology Life Cycle



Inter-dependencies

Inter-dependencies refer the relationship between activities carried out when building different ontologies



Fernández-López, M.; Gómez-Pérez, A.; Rojas M.D.
Ontology's Crossed Life Cycle.
Lectures Notes in Artificial Intelligence Nº 1937. October 2000

Methodologies and methods for building ontologies from scratch

Methods and Methodologies analysed (7):

- **Cyc method**
- **Uschold and King's method**
- **Grüninger and Fox's methodology**
- **KACTUS method**
- **METHONTOLOGY**
- **SENSUS method**
- **On-To-Knowledge methodology**

- **Framework for comparing methodologies**
- **Methodology/method description**
- **Comparison of the approaches against the framework**
- **Conclusions**

METHONTOLOGY Framework

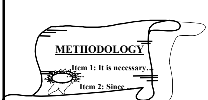
Ontology Development Process (which activities)

- **Management, Development, Support**

Life Cycle (Order of activities)

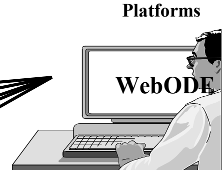
- **Evolving Prototype.**

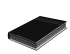
Methodology (how to carry out)



- **Specification**
- **Knoweldge Acquisition**
- **Conceptualization**
- **Integration**
- **Implementation.**
- **Evaluation**
- **Documentation**

Ontology Building Platforms



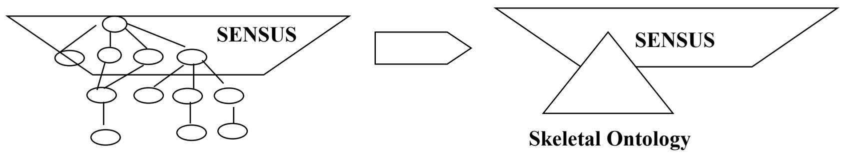


Gómez-Pérez, A. *Knowledge Sharing and Reuse.* **In the Handbook of Applied Expert Systems.** CRC Press. 1998.

SENSUS as a basis for a domain-specific ontology (I)

Linking Domain Specific Terms to a broad Coverage Ontology

To identify the terms in SENSUS that are relevant to a particular domain and then prune the skeletal ontology using heuristics



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.

SENSUS as a basis for a domain-specific ontology (II)

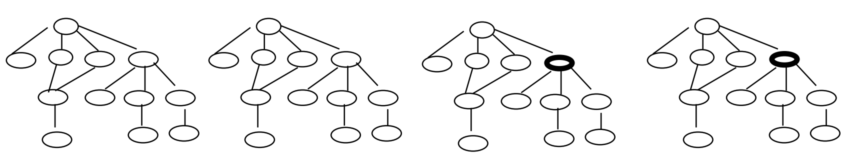
see example

METHOD

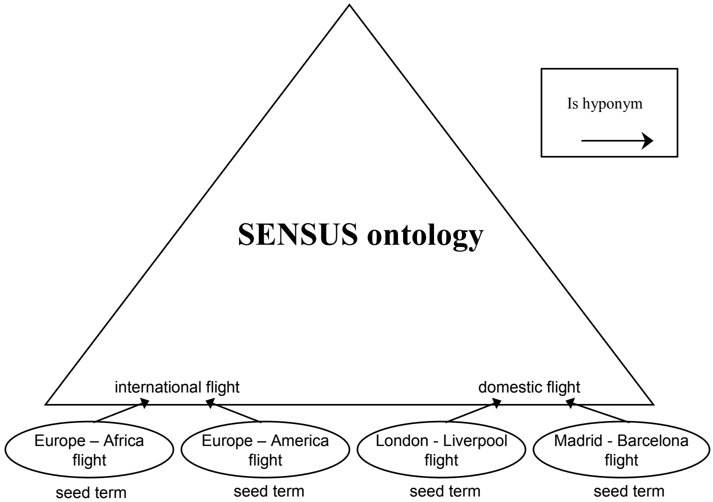
1. **Identify "seed" terms**
2. **Link seed terms to SENSUS by hand**
3. **Include nodes on the path to root**
4. **Add entire subtrees using the heuristic:**

   If many nodes in a subtree are relevant, the other nodes in the subtree are relevant

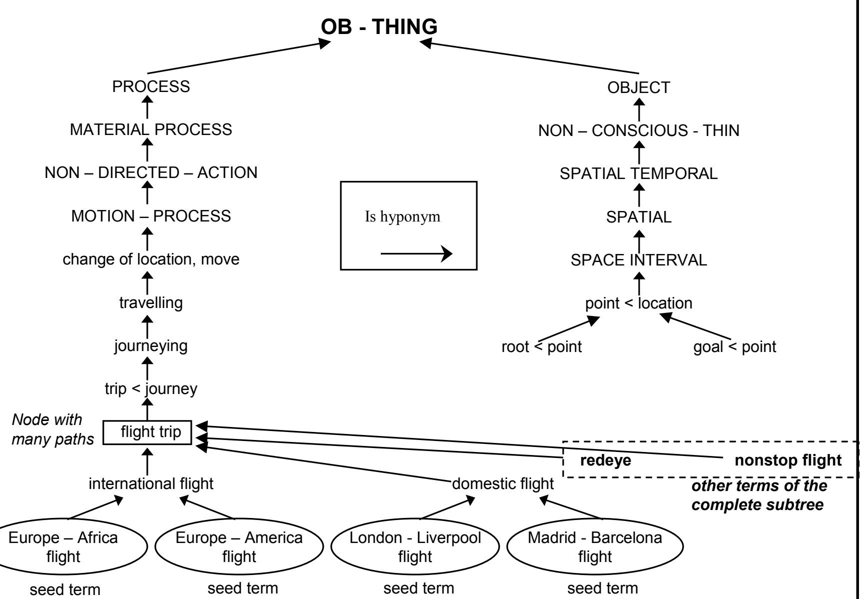
- **Sensus Term**
- **Seed**
- **Path to root**
- **Frequent Parent**
- **Subtree Term**



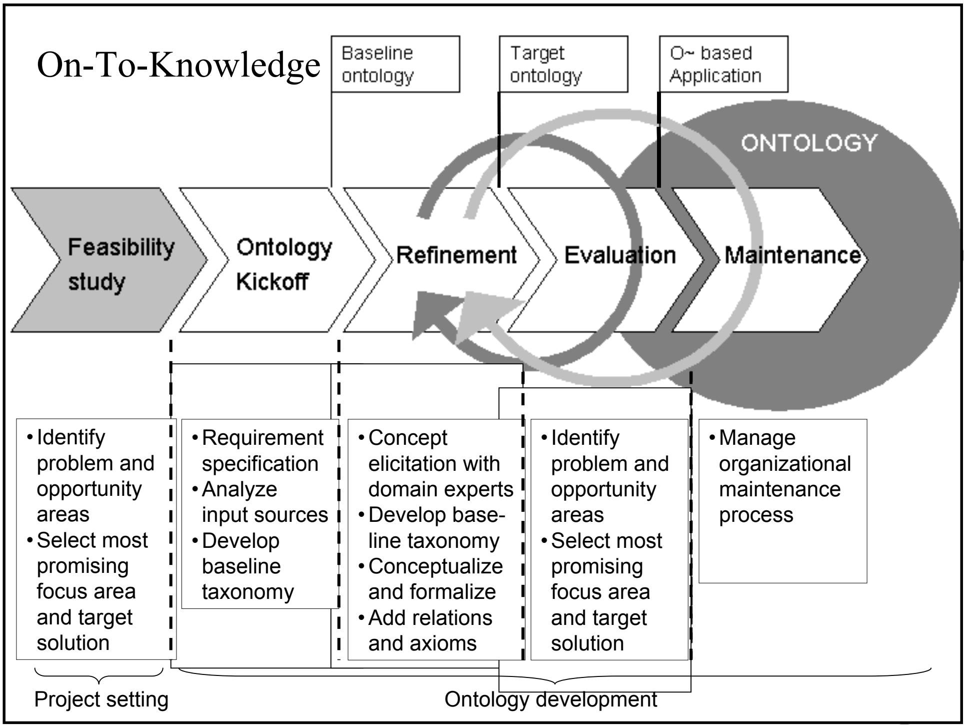
B. Swartout; R. Patil; k. Knight; T. Russ. *Toward Distributed Use of Large-Scale Ontologies*
Ontological Engineering. AAAI-97 Spring Symposium Series. 1997. 138-148.

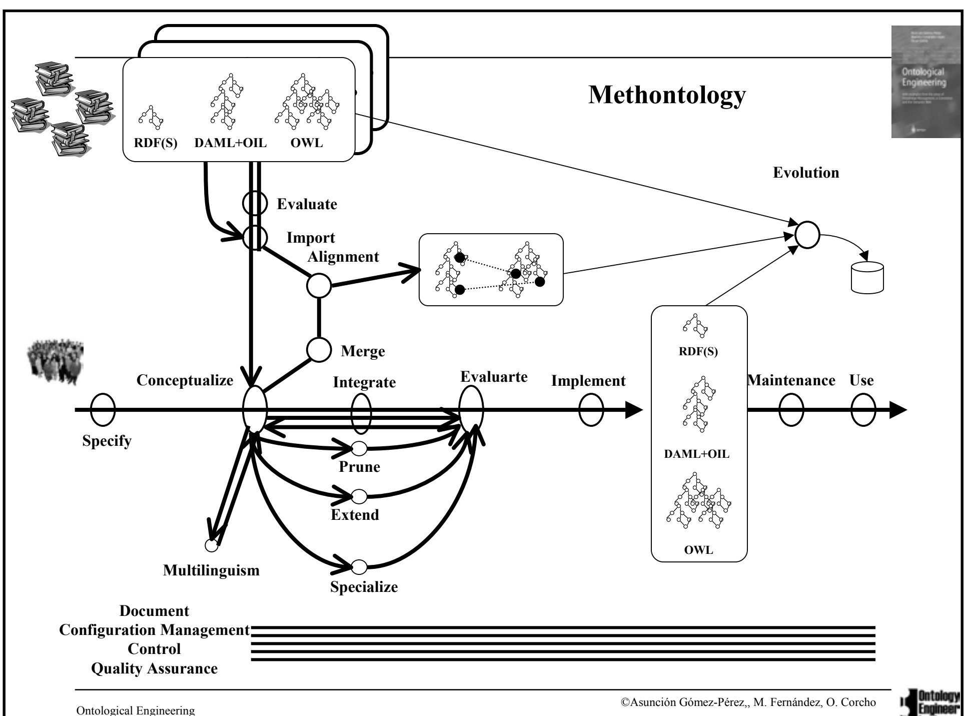


SENSUS ontology
Is hyponym
international flight
domestic flight
Europe – Africa flight
Europe – America flight
London - Liverpool flight
Madrid - Barcelona flight
seed term
seed term
seed term
seed term



OB - THING
PROCESS
MATERIAL PROCESS
NON – DIRECTED – ACTION
MOTION – PROCESS
change of location, move
travelling
journeying
trip < journey
Node with many paths
flight trip
international flight
domestic flight
OBJECT
NON – CONSCIOUS - THIN
SPATIAL TEMPORAL
SPATIAL
SPACE INTERVAL
point < location
root < point
goal < point
Is hyponym
redeye
nonstop flight
other terms of the complete subtree
Europe – Africa flight
Europe – America flight
London - Liverpool flight
Madrid - Barcelona flight
seed term
seed term
seed term
seed term


On-To-Knowledge
Baseline ontology
Target ontology
O~ based Application
ONTOLOGY
Feasibility study
Ontology Kickoff
Refinement
Evaluation
Maintenance
• Identify problem and opportunity areas
• Select most promising focus area and target solution
• Requirement specification
• Analyze input sources
• Develop baseline taxonomy
• Concept elicitation with domain experts
• Develop base-line taxonomy
• Conceptualize and formalize
• Add relations and axioms
• Identify problem and opportunity areas
• Select most promising focus area and target solution
• Manage organizational maintenance process
Project setting
Ontology development


Methontology
RDF(S)
DAML+OIL
OWL
Evaluate
Import
Alignment
Evolution
Merge
Conceptualize
Integrate
Evaluarte
Implement
Maintenance
Use
Specify
Prune
Extend
Multilinguism
Specialize
RDF(S)
DAML+OIL
OWL
Document
Configuration Management
Control
Quality Assurance
Ontological Engineering
©Asunción Gómez-Pérez,, M. Fernández, O. Corcho

Evaluation framework for methodologies and me

Construction Strategy

- **Life Cycle Proposal**
- **Strategy with respect the application**
- **Use of core ontologies**
- **Strategy to identify concepts**

Proposed Ontology development process

- **Project Management processes:**
- **Ontology development-oridented processes**
- **Integral Processes:**

Acceptation of the methodology by other groups

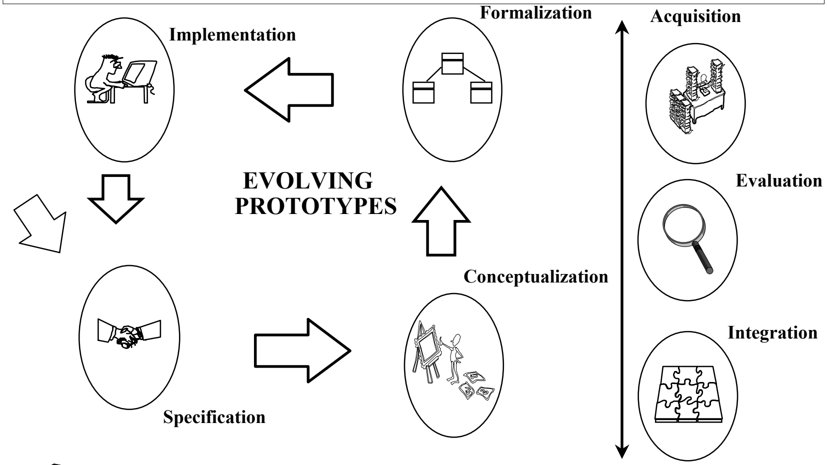
Technological support to the methodology

Summary of the ontology development process

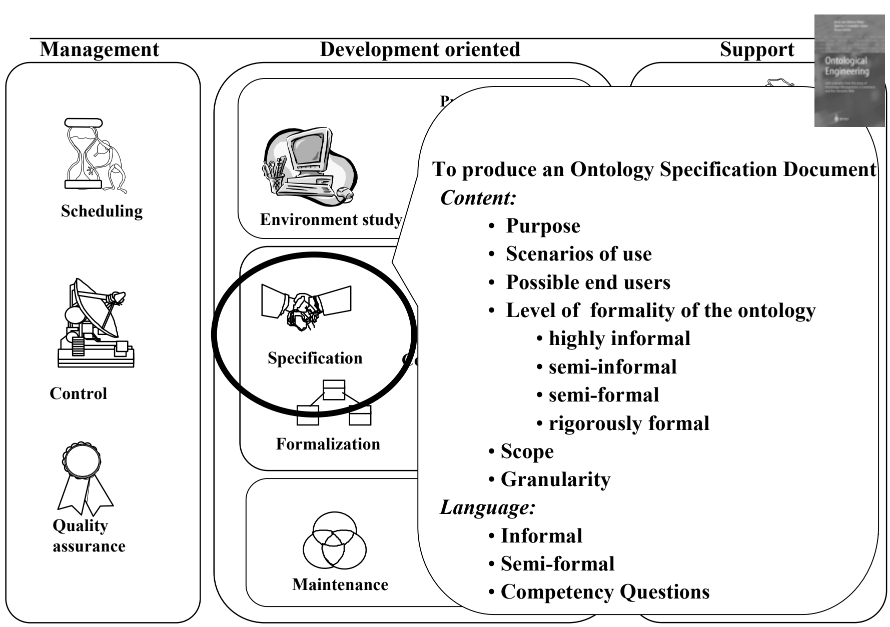
Feature			Cyc	Uschold & King	Grüninger & Fox	KACTUS	METHONTOLOGY	SENSUS	On-To-Knowledge
Ontology management activities	Scheduling		NP	NP	NP	NP	*Proposed*	NP	*Described*[12]
	Control		NP	NP	NP	NP	*Proposed*	NP	*Described*
	Quality assurance		NP	NP	NP	NP	NP	NP	*Described*
Ontology development-oriented activities	Pre-development processes	Environment study	NP	NP	NP	NP	NP	NP	*Proposed*
		Feasibility study	NP	NP	NP	NP	NP	NP	*Described*
	Development processes	Specification	NP	*Proposed*	*Described in detail*	*Proposed*	*Describe in detail*	*Proposed*	*Describe in detail*
		Conceptualization	NP	NP	*Described in detail*	*Proposed*	*Described in detail*	NP	*Proposed*
		Formalization	NP	NP	*Described in detail*	*Described*	*Described*	NP	*Described*
		Implementation	*Proposed*	*Proposed*	*Described*	*Proposed*	*Described in detail*	*Described*	*Described*
	Post-development processes	Maintenance	NP	NP	NP	NP	*Proposed*	NP	*Proposed*
		Use	NP	NP	NP	NP	NP	NP	*Proposed*
Ontology support activities	Knowledge acquisition		*Proposed*	*Proposed*	*Proposed*	NP	*Described in detail*	NP	*Described*
	Evaluation		NP	Proposed	*Described in detail*	NP	*Described in detail*	NP	*Proposed*
	Integration		*Proposed*[13]	*Proposed*	*Proposed*	*Proposed*	*Proposed*	NP	*Proposed*
	Configuration management		NP	NP	NP	NP	*Described*	NP	*Proposed*
	Documentation		*Proposed*	*Proposed*	*Proposed*	NP	*Described in detail*	NP	*Described*
	Merging and Alignment		NP	NP	NP	NP	NP	NP	NP

...

Ontology Life Cycle



Gómez-Pérez, A. *Knowledge Sharing and Reuse*. **In the Handbook of Applied Expert Systems.** CRC Press. 1998.



Getting terminology using Competency Questions

Motivating Scenarios → **Informal Competency Questions** → **Formal Terminology**

Identify intuitively possible applications and solutions

Identify Queries:
• Answers: Axioms
Formal definitions
• Questions: Terminology

Classes
Relations
Attributes
Axioms
Instances

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

Find stories which include Person P

Identify Queries:
• Questions: Story, Person, involved-in, includes
• Answers: Story S1 includes person P

Classes: Story, Person
Relations: Involved-in, includes
Attributes: ---
Axioms
Instances: P, S1

Getting terminology using Competency Questions

Find all the events attended by participants working on semantic web projects

Each project has a property storing its type

Identify Queries:
Questions: Event, Researcher, Project, work-at, attend-at, type-of-Project
Answers: ISWC, EKAW, K-cap,....

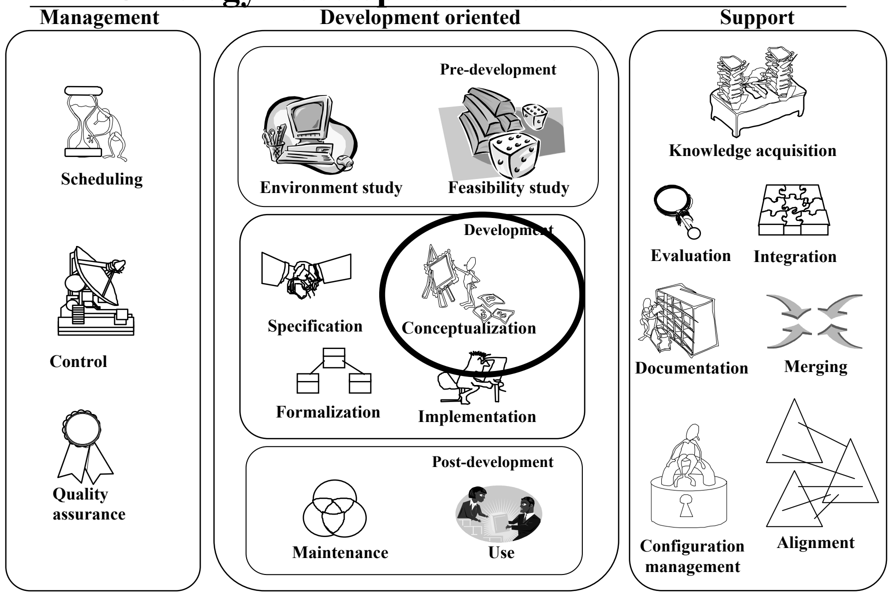
Classes: Event, International Conference, National Conference, Workshop, Project Meetings, Researcher, Person, Project,
Relations: Work-at, attend-at
Attributes: Type of Project
Axioms: For all...
Instances: ISWC, EKAW, K-cap,....

•Taxonomy of Topics
•There exist a relation that connects projects and topics

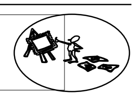
Identify Queries:
Questions: Event, Researcher, Project, work-at, attend-at, Semantic Web Topics, main-topics
Answers: ISWC, EKAW, K-cap,....

Classes: Event, International Conference, National Conference, Workshop, Project Meetings, Researcher, Person, Project, Topics, Ontologies, mark-up languages, semantic web services, annotations, ...
Relations: Work-at, attend-at, main-topics, topic-of
Attributes: ---
Axioms: For all ....
Instances:ISWC, EKAW, K-cap,....

Ontology Development Process



METHONTOLOGY: Conceptualization

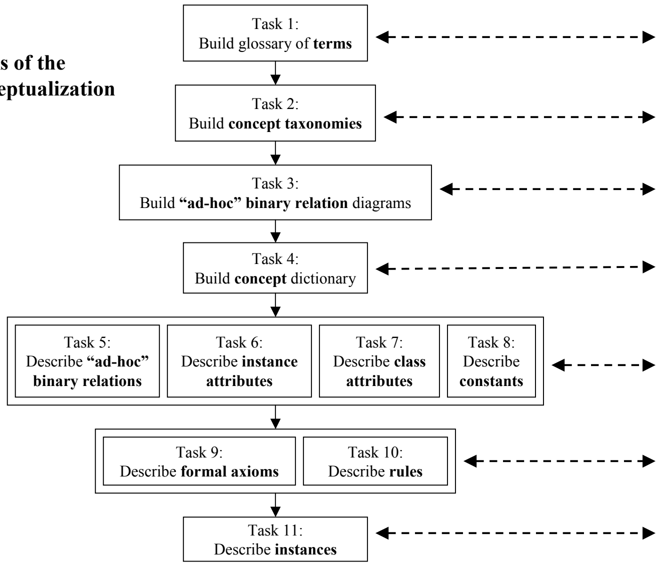


It organizes and structures the knowledge acquired during the knowledge acquisition activity using external representations that are independent of the knowledge representation paradigms and implementation languages in which the ontology will be formalized and implemented.

- **We can use Ontology Editors for conceptualizing the Ontology**
- **The ontology editors transforms the conceptualization into executable code using translators**

Gómez-Pérez, A. *Knowledge Sharing and Reuse*. **In the Handbook of Applied Expert Systems.** CRC Press. 1998.

Tasks of the conceptualization



Terms glossary

Name	Synonyms	Acronyms	Description	Type
American Airlines Flight	--	AA Flight	Flight operated by American Airlines.	Concept
Bed and Breakfast	--	--	An establishment (as an inn) offering lodging and breakfast	Concept
British Airways Flight	--	BA Flight	Flight operated by British Airways.	Concept
Business Trip	--	--	A special package for businessmen, consisting of a flight and a good quality hotel.	Concept
Camping	--	--	Temporal lodging in a camp.	Concept
Economy Trip	--	--	An economic package, usually costing less than 1000$.	Concept
European Location	--	--	A location in Europe.	Concept
Five-stars Hotel	--	--	High quality hotel.	Concept
Flight	--	--	A journey by plane identified by a flight number.	Concept
Hotel	--	--	An establishment that provides lodging and usually meals, entertainment, and various personal services for the public	Concept
Iberia Flight	--	IB Flight	Flight operated by Iberia.	Concept
Japan Location	--	--	A location in Japan.	Concept
Location	Place	--	A position or site occupied or available for occupancy or marked by some distinguishing feature.	Concept
Lodging	Accommodation	--	A temporary place to stay during a trip, sleeping accommodations.	Concept
Luxury Trip	--	--	A luxury and expensive trip.	Concept
Spain Location	--	--	A location in Spain.	Concept
Train Travel	Rail Travel	--	A journey by train.	Concept
Travel	--	--	A journey from place to place.	Concept
Travel Package	--	--	A travel package that a person can ask for. It consists of one or several means of transport and one or several accommodations.	Concept

Primitives for Modelling Taxonomies

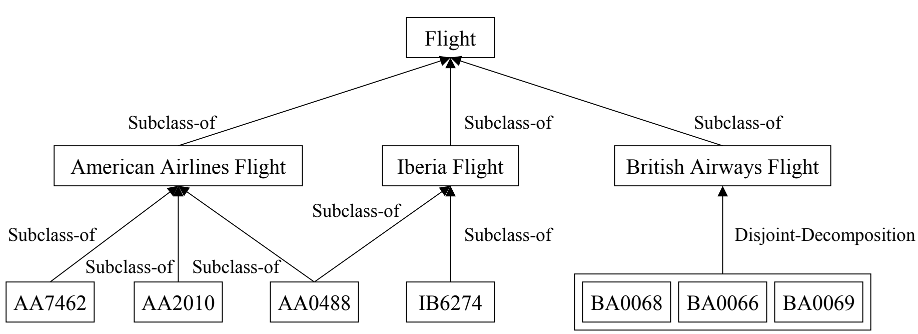
Subclass-of:

Disjoint decomposition: a set of subclasses of C that do not have common instances and do not cover C

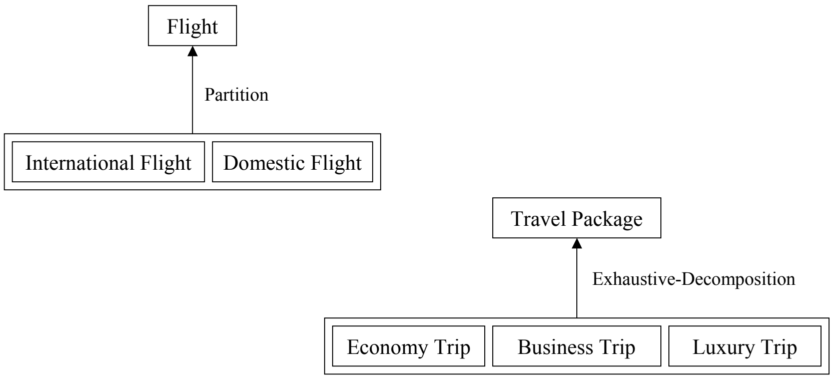
Partition: a set subclasses of C that cover C and do not have common instances or subclasses

Exhaustive-Decomposition: a set subclasses of C that cover C and may have common instances or subclasses

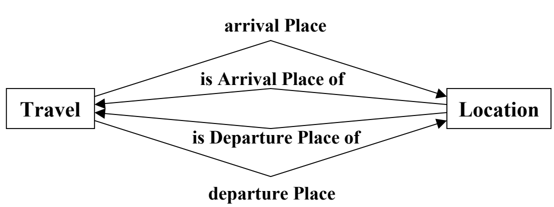
Example of a Taxonomy (I)



Example of a Taxonomy (II)



Identify Ad-hoc relations



Define a Concept Dictionary

Concept name	Class attributes	Instance attributes	Relations
AA7462	--	--	same Flight as
American Airlines Flight	company Name	--	--
British Airways Flight	company Name	--	--
Five-stars Hotel	number of Stars	--	--
Flight	--	--	same Flight as
Location	--	name size	is Arrival Place of is Departure Place of
Lodging	--	price of Standard Room	placed in
Travel	--	arrival Date company Name departure Date return Fare single Fare	arrival Place departure Place
Travel Package	--	budget final Price name number of Days travel Restrictions	arrival Place departure Place accommodated in travels in
USA Location	--	--	--

Define in detail Instance Attributes

Instance attribute name	Concept name	Value type	Measurement unit	Preci-sion	Range of values	Cardi-nality
budget	Business Trip	Float	Currency Quantity	0.01	1000...3000	(0,1)
budget	Economy Trip	Float	Currency Quantity	0.01	0...1000	(0,1)
name	Location	String	--	--	--	(1,N)
size	Location	Integer	Square Meters	1	--	(1,1)
price of Standard Room	Lodging	Float	--	--	--	(0,1)
budget	Luxury Trip	Float	Currency Quantity	0.01	--	(0,1)
arrival Date	Travel	Date	--	--	--	(0,1)
company Name	Travel	String	--	--	--	(0,N)
departure Date	Travel	Date	--	--	--	(0,1)
return Fare	Travel	Float	Currency Quantity	0.01	--	(0,1)
single Fare	Travel	Float	Currency Quantity	0.01	--	(0,1)
budget	Travel Package	Float	Currency Quantity	0.01	--	(0,1)
final Price	Travel Package	Float	Currency Quantity	0.01	--	(0,1)
number of Days	Travel Package	Integer	days	1	--	(0,1)
travel Restrictions	Travel Package	String	--	--	--	(0,1)

Define Class Attributes

Attribute name	Defined at concept	Value type	Measurement unit	Precision	Cardinality	Values
company Name	American Airlines Flight	String	--	--	(1,1)	AA
company Name	British Airways Flight	String	--	--	(1,1)	BA
company Name	Iberia Flight	String	--	--	(1,1)	IB
number of Stars	Five-stars Hotel	Integer	star	1	(1,1)	5
number of Stars	Four-stars Hotel	Integer	star	1	(1,1)	4
number of Stars	Three-stars Hotel	Integer	star	1	(1,1)	3
number of Stars	Two-stars Hotel	Integer	star	1	(1,1)	2
number of Stars	One-stars Hotel	Integer	star	1	(1,1)	1

Define formal axioms

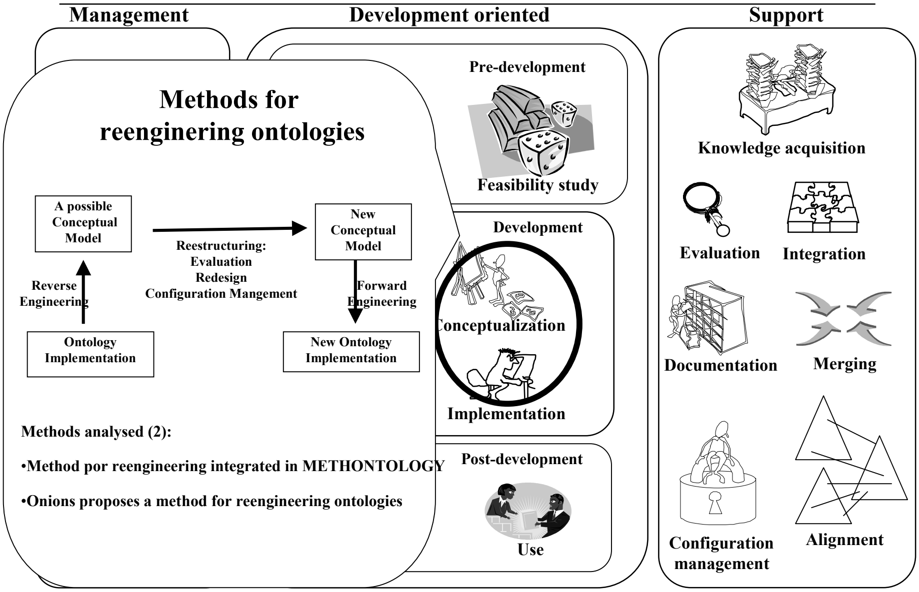
Axiom name	Train inside Europe
Description	Every train that departs from a European location must arrive at another European location
Expression	forall(?X,?Y,?Z) ([Train Travel](?X) and [departure Place](?X,?Y) and [arrival Place](?X,?Z) and [European Location](?Y) -> [European Location](¿Z))
Concepts	Train Travel European Location
Referred attributes	--
Ad-hoc binary relations	departure Place arrival Place
Variables	?X ?Y ?Z

Define rules

Rule name	Costa Cruises rule
Description	Every ship that departs from Europe is arranged by the company Costa Cruises
Expression	if [European Location](?Y) and Ship(?X) and [departure Place](?X,?Y) then [company Name](?X, "Costa Cruises")
Concepts	Ship European Location
Referred attributes	company Name
Ad-hoc binary relations	departure Place
Variables	?X ?Y

Define Instances

Instance Name	Concept Name	Attribute	Values
AA7462_Feb08_2002	AA7462	company Name	American Airlines
		departure Date	02/08/2002
		arrival Date	02/08/2002
		single Fare	300
AA7462_Feb16_2002	AA7462	company Name	American Airlines
		departure Date	02/16/2002
		arrival Date	02/16/2002
		single Fare	300



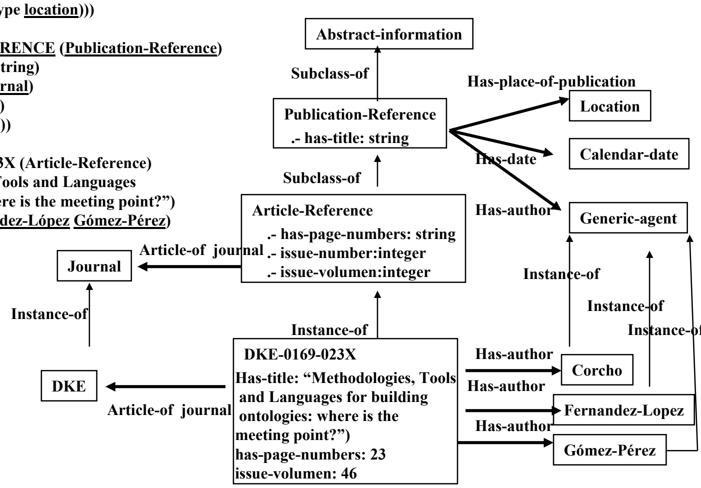
Ontology Libraries

DAML ontology library	**http://www.daml.org/ontologies/**	see library
Protege ontology library	**http://protege.stanford.edu/ontologies.html**	
Ontolingua ontology library	**http://ontolingua.stanford.edu/**	
WebOnto ontology library	**http://webonto.open.ac.uk**	
SHOE ontology library	**http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html**	
WebODE ontology library	**http://webode.dia.fi.upm.es/**	
(KA)^2 ontology library	**http://ka2portal.aifb.uni-karlsruhe.de/**	
AKT ontology	**http://www.aktors.org/ontology/**	

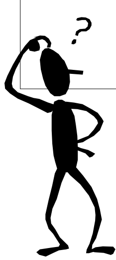
(def-class PUBLICATION-REFERENCE (abstract-information)
"we have decided that a publication reference is an intangible, abstract information"
((has-title :type string)
(has-author :type generic-agent)
(has-date :type calendar-date)
(has-place-of-publication :type location)))

(def-class ARTICLE-REFERENCE (Publication-Reference)
((has-page-numbers :type string)
(article-of-journal :type journal)
(issue-number :type integer)
(issue-volume :type integer)))

(def-instance DKE-0169-023X (Article-Reference)
(has-title "Methodologies, Tools and Languages
for building ontologies: where is the meeting point?")
(has-author Corcho Fernández-López Gómez-Pérez)
(has-date July-2003)
(has-page-numbers 23)
(article-of-journal DKE)
(issue-volume 46))



Selecting a tool for building the ontology



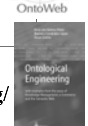
I must develop an ontology.
What Tool do I use to conceptualize it???

- **The one(s) I like the most?**
- **The one(s) I know the best?**
- **The one(s) that import/export an ontology from/to a given ontology implementation language?**
- **The one(s) that best fit(s) my needs?**

Main criteria for selecting an ontology editor

- **Which activities of the ontology development process are supported by each tool?**
- **What is the expressiveness of the underlying knowledge model attached to the tool?**
- **What kinds of user interface does the tool provide to model ontology terms?**
- **Does the tool provide an advanced user interface to model formal axioms or complex expressions?**
- **Does the tool need to be installed locally or not?**
- **Can it be used with a Web browser?**
- **Where are the ontologies stored (in databases or files)?**
- **Does the tool have an inference engine and querying tools?**
- **Which ontology languages or formats does the tool generate?**
- **Is the tool able to import ontologies implemented in ontology languages or in other formats?**
- **Is it possible to export an ontology from one tool to another without losing knowledge?**
- **How can ontology-based applications use ontologies developed with a tool?**
- **What types of consistency checking and content evaluation does the tool perform?**

Ontology development Tools



KAON from AIFB and FZI at the University of Karlsruhe http://kaon.semanticweb.org/

OilEd from University of Manchester http://oiled.man.ac.uk/

Ontolingua from KSL (Stanford University) http://www-ksl.stanford.edu

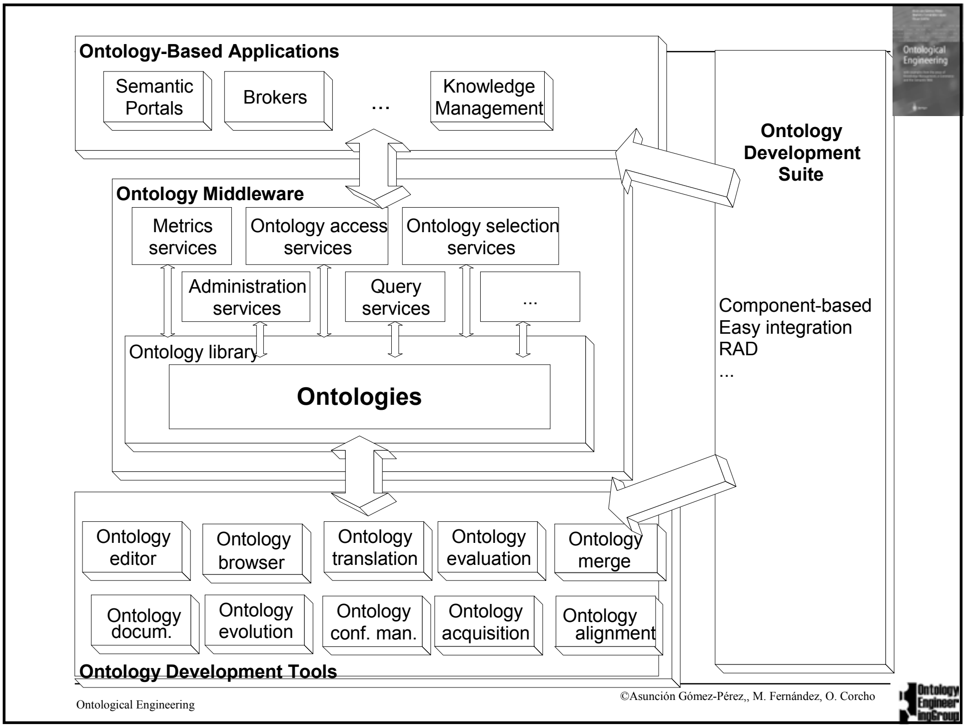
OntoSaurus from ISI (USA) http://www.isi.edu/isd/ontosaurus.html

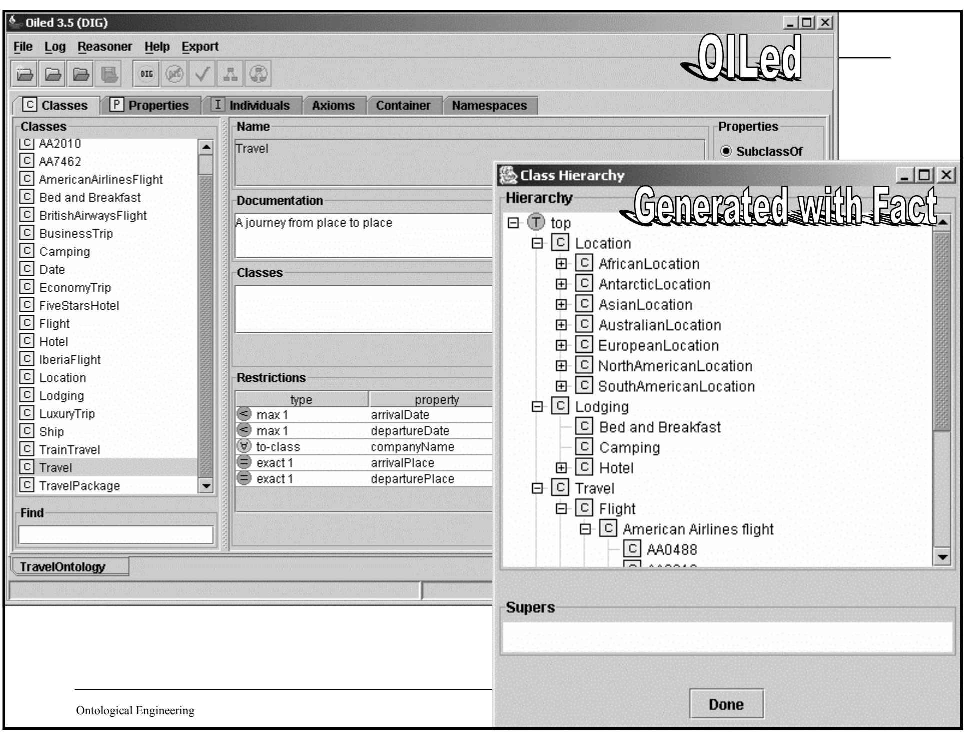
OntoEdit from Karlsrhue Univ. http://ontoserver.aifb.unikarlsruhe.de/ontoedit/

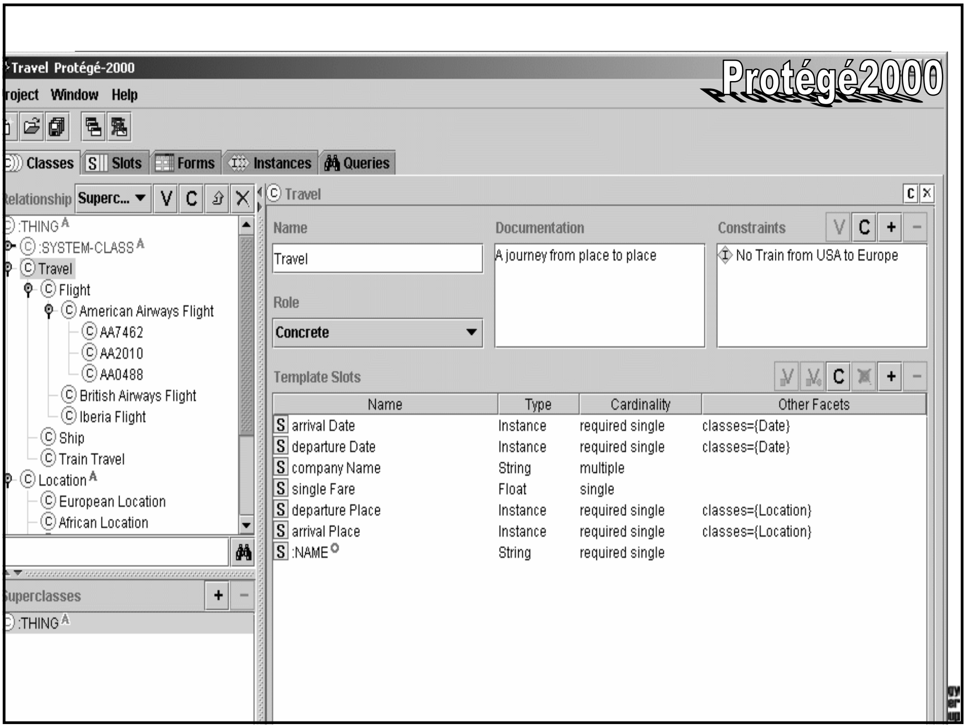
Protégé 2000 from SMI (Stanford University) http://protege.stanford.edu/

WebOnto from KMI (Open University) http://kmi.open.ac.uk/projects/webonto/

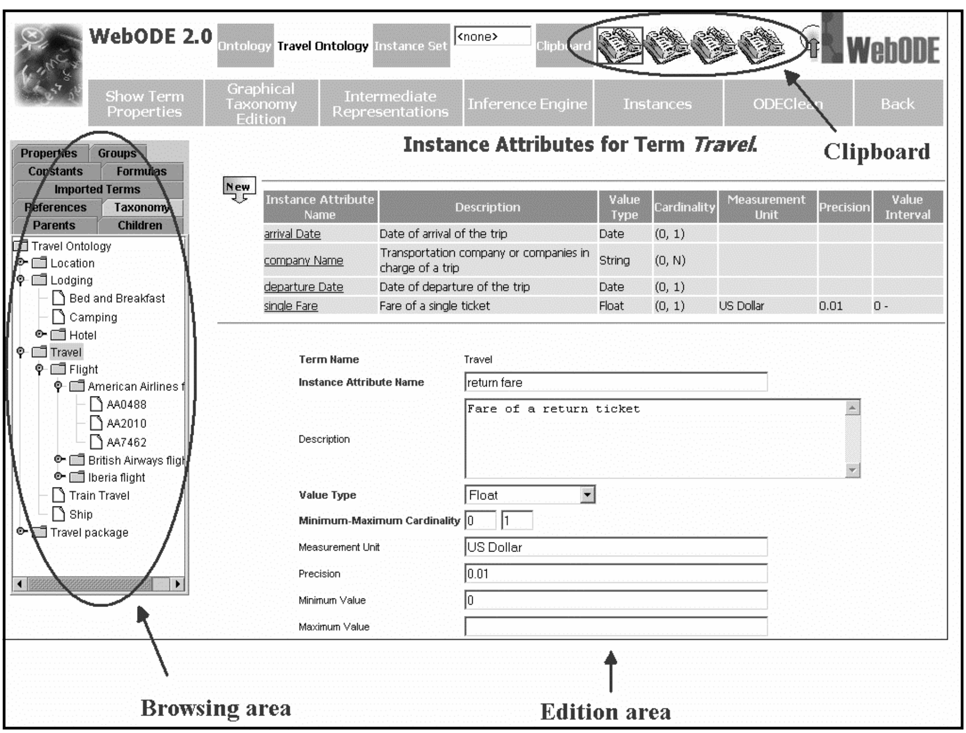
WebODE from UPM http://webode.dia.fi.upm.es/webODE/


Ontology-Based Applications
Semantic Portals
Brokers
...
Knowledge Management
Ontology Development Suite
Ontology Middleware
Metrics services
Ontology access services
Ontology selection services
Administration services
Query services
...
Ontology library
Ontologies
Component-based
Easy integration
RAD
...
Ontology editor
Ontology browser
Ontology translation
Ontology evaluation
Ontology merge
Ontology docum.
Ontology evolution
Ontology conf. man.
Ontology acquisition
Ontology alignment
Ontology Development Tools
Ontological Engineering
©Asunción Gómez-Pérez,, M. Fernández, O. Corcho


OIled 3.5 (DIG)
OILed
Generated with Fact
File Log Reasoner Help Export
Classes Properties Individuals Axioms Container Namespaces
Classes
AA2010
AA7462
AmericanAirlinesFlight
Bed and Breakfast
BritishAirwaysFlight
BusinessTrip
Camping
Date
EconomyTrip
FiveStarsHotel
Flight
Hotel
IberiaFlight
Location
Lodging
LuxuryTrip
Ship
TrainTravel
Travel
TravelPackage
Find
Name
Travel
Documentation
A journey from place to place
Restrictions
type property
max 1 arrivalDate
max 1 departureDate
to-class companyName
exact 1 arrivalPlace
exact 1 departurePlace
Properties
SubclassOf
TravelOntology
Class Hierarchy
Hierarchy
top
Location
AfricanLocation
AntarcticLocation
AsianLocation
AustralianLocation
EuropeanLocation
NorthAmericanLocation
SouthAmericanLocation
Lodging
Bed and Breakfast
Camping
Hotel
Travel
Flight
American Airlines flight
AA0488
Supers
Done
Ontological Engineering



Travel Protégé-2000
Protégé2000
roject Window Help
Classes
Slots
Forms
Instances
Queries
Relationship Superc...
: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
Travel
Name
Travel
Documentation
A journey from place to place
Constraints
No Train from USA to Europe
Role
Concrete
Template Slots
Name | Type | Cardinality | Other Facets
arrival Date | Instance | required single | classes={Date}
departure Date | Instance | required single | classes={Date}
company Name | String | multiple
single Fare | Float | single
departure Place | Instance | required single | classes={Location}
arrival Place | Instance | required single | classes={Location}
:NAME | String | required single



WebODE 2.0
Ontology
Travel Ontology
Instance Set
<none>
Clipboard
WebODE
Show Term Properties
Graphical Taxonomy Edition
Intermediate Representations
Inference Engine
Instances
ODEClean
Back
Properties
Groups
Constants
Formulas
Imported Terms
References
Taxonomy
Parents
Children
Travel Ontology
Location
Lodging
Bed and Breakfast
Camping
Hotel
Travel
Flight
American Airlines f
AA0488
AA2010
AA7462
British Airways fligh
Iberia flight
Train Travel
Ship
Travel package
Instance Attributes for Term Travel.
Clipboard
New
Instance Attribute Name | Description | Value Type | Cardinality | Measurement Unit | Precision | Value Interval
arrival Date | Date of arrival of the trip | Date | (0, 1)
company Name | Transportation company or companies in charge of a trip | String | (0, N)
departure Date | Date of departure of the trip | Date | (0, 1)
single Fare | Fare of a single ticket | Float | (0, 1) | US Dollar | 0.01 | 0 -
Term Name
Travel
Instance Attribute Name
return fare
Description
Fare of a return ticket
Value Type
Float
Minimum-Maximum Cardinality
0
1
Measurement Unit
US Dollar
Precision
0.01
Minimum Value
0
Maximum Value
Browsing area
Edition area

Comparison of Ontology building tool

Criteria:

- **General Description**
- **Tools' architecture: architecture, extendibility, ontology storage, back-up**
- **Tools' interoperability: with tools, export/import from/to languages**
- **KR paradigm supported by the tool**
- **Methodological Support**
- **Tools' inference services**
- **Tools' usability**

- **Framework for comparing tools**
- **Tool description**
- **Comparison of the tools against the framework**
- **Conclusions**
- **Recommendations**

Ontology Development Tools.
General description

	KAON	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protégé 2000	WebODE	WebOnto
Developers	AIFB-FZI (University of Karlsruhe)	IMG (University of Manchester)	Ontoprise	KSL (Stanford University)	ISI (University of Southern California)	SMI (Stanford University)	Ontology Group (Universidad Politécnica de Madrid)	KMI (Open University)
Version (date)	1.2.5 (Jan2003)	3.5 (Dec2002)	2.6 (Nov2002)	1.0.649 (Nov2001)	1.9 (Mar2002)	1.8 (Apr2003)	2.1 (Mar2003)	2.3 (May2001)
Availability	Open source	Open source	Freeware Licenses	Free access	Open source Free access	Open source	Free access	Free access
Methodological support	No	No	On-To-Knowledge	No	No	No	METHONTOLOGY	No

Knowledge Representation Approach

Knowledge model	KAON	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protégé 2000	WebODE	WebOnto
KR formalism	Frames	DL	Frames + FOL	Frames + FOL	DL	Frames + FOL	Frames + FOL	Frames + FOL
Underlying KR language	KAON	DAML+OIL	OXML	Ontolingua	LOOM	--	--	OCML
Formal axiom language	--	--	FLogic	KIF	LOOM	PAL	WAB	OCML

Software architecture

	KAON	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protégé 2000	WebODE	WebOnto
Application architecture	n-tier	3-tier	3-tier	Client/server	Client/server	3-tier	n-tier	Client/server
Extensibility	Application server	Backends	Plug-ins	--	--	Plug-ins	Application server	--
Ontology storage	Files DBMS	Files	Files	Files	Files	Files DBMS	DBMS	Files
Programming language	Java	Java	Java	Lisp	Lisp	Java	Java	Java + Lisp

Main Features of the editor and Inference Engine

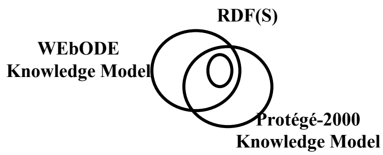
Ontology editors	KAON	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protégé 2000	WebODE	WebOnto
Front-end	Java Swing	Java Swing	Java Swing	HTML	HTML	Java Swing	HTML and applets	Applets
Graphical concept taxonomy edition	Yes	No	Yes	No	No	Yes	Yes	Yes
Graphical prunes	Yes	No	Yes	No	No	Yes	Yes	Yes
Zooms	Yes	No	Yes	No	No	Yes	No	No
Formal axiom editor	No	Yes	No	No	No	Yes	Yes	No
Collaborative edition	Yes	No	Yes	Yes	Yes	No	Yes	Yes
Inference engine	--	FaCT (built-in) RACER DIG compliant	OntoBroker	JTP	LOOM classifier	PAL (built-in) Jess FaCT Prolog FLORA Algernon	Prolog (built-in) Jess	OCML KR system
Consistency checking	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Automatic classification	No	Yes	No	No	Yes	No	No	No

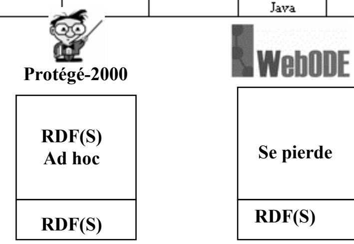
SIG3
EON WS

Interoperability

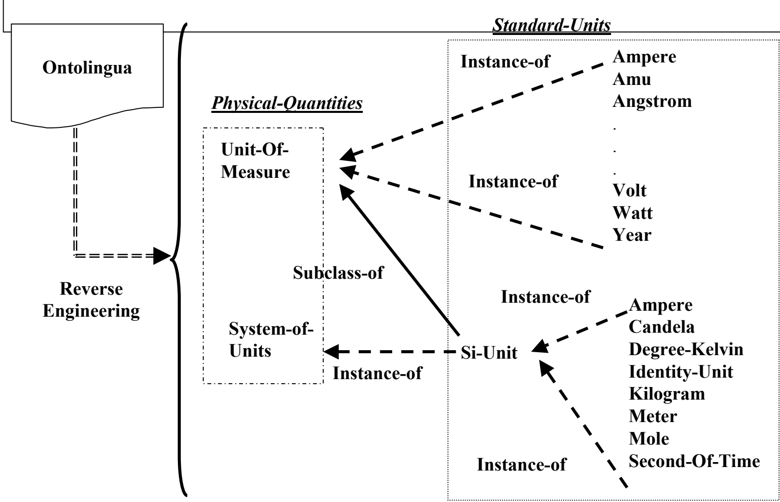
	KAON	OilEd	OntoEdit	Ontolingua	OntoSaurus	Protégé 2000	WebODE	WebOnto
With other tools	KAON-Portal	--	OntoAnnotate OntoMat Semantic Miner	Chimaera OKBC	OKBC	PROMPT OKBC ArgoUML	ODE-KM ODE-SeW ODE-SWS ODEAnnotate Protégé-2000	MnM
Import	KAON RDF(S)	RDF(S) OIL DAML+OIL SHIQ	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF CML IDL	LOOM PowerLOOM Stella IDL	XML RDF(S) OWL XML Schema XMI	XML RDF(S) OWL DAML+OIL CARIN	OCML RDF(S)
Export	KAON RDF(S)	RDF(S) OIL DAML+OIL OWL SHIQ DIG	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF LOOM CLIPS CML Epikit Prolog IDL	LOOM PowerLOOM KIF Ontolingua Stella IDL C++	XML RDF(S) OWL XML Schema FLogic CLIPS Java XMI	XML RDF(S) OWL DAML+OIL OIL CARIN FLogic Prolog Jess Java	OCML Ontolingua RDF(S)

Why low interoperability?

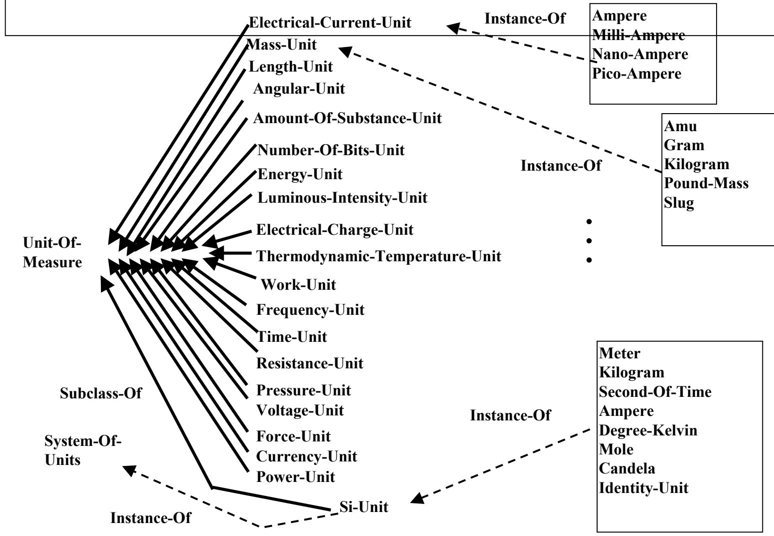


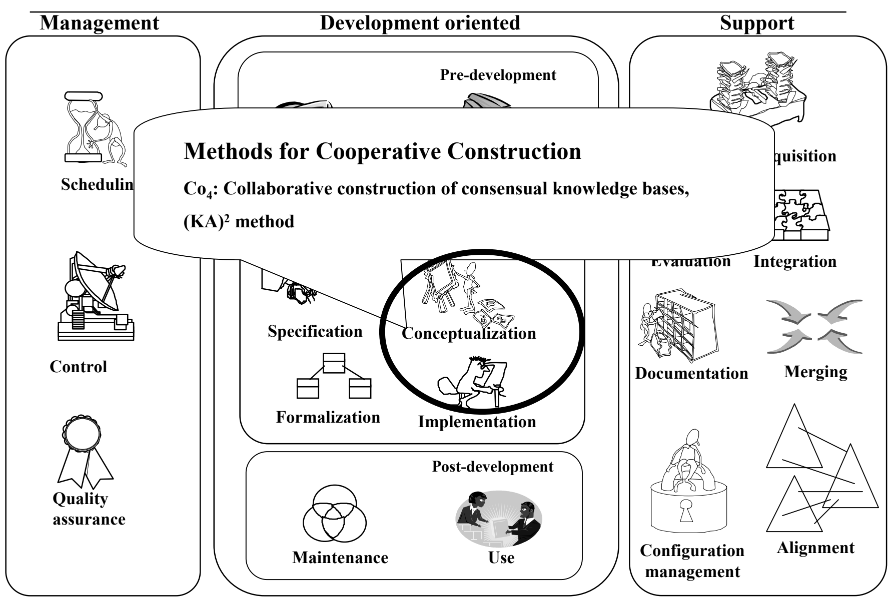


Standard-Units Reverse Engineering

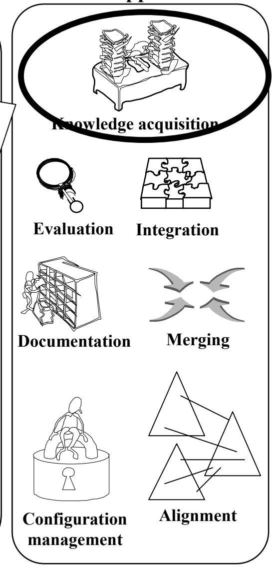


Restructuring Standard-Units Conceptual Model





Support



Ontology Learning is the set of methods and techniques used for building an ontology from scratch, enriching, or adapting an existing ontology in a semi-automatic fashion using several sources. It aims to reduce the time and the effort necessary in the knowledge acquisition process.

- Approaches:
  - **Ontology learning from text**
  - **Ontology learning from dictionary**
  - **Ontology learning from knowledge bases**
  - **Ontology learning from semi-structured schemata**
  - **Ontology learning from relational schemata**

Approaches for Ontology Learning

Main techniques

Concept learning
Linguistic patterns
NLP techniques
Machine learning techniques
Ontological techniques
Reverse engineering
Statistical approach
Text-mining

Sources

Texts

Dictionaries

Knowledge bases

Semi-structured schema

Relational Schema

Criteria

Methods

Tools

Approaches for Ontology Learning

OL from text

- 18 methods
- 18 tools

OL from dictionary

- 3 methods
- 2 tools

OL from knowledge bases

- 1 method and tool

OL from semi-structured schemata

- 4 methods
- 1 tool

OL from relational schemata

- 4 methods

For each group of methods:

- Framework for comparing OL methods
- Method description
- Comparison of each Method against the framework
- Conclusions
- Recommendations

For each group of tools:

- Framework for comparing OL Tools
- Tool description
- Comparison of each Tool against the framework
- Conclusions
- Recommendations

Techniques used in different OL approaches

OL from text

- Natural Language Techniques
- Clustering techniques
- Machine learning
- Statistical aproach

OL from dictionary

- Natural Language Processing
- Statistical aproach

OL from knowledge bases

- Rules

OL from semi-structured schemata

- Graph Theory
- Machine Learning
- Pattern Recognition
- Clustering
- Ontological Techniques

OL from relational schemata

- Mapping Techniques
- Reverse Engineering

OL from texts:methods and techniques

Aguirre and colleagues' method	**URL:** Not available
Alfonseca and Manandhar's method	**URL:** **http://www.ii.uam.es/~ealfon**
Aussenac-Gilles and colleagues' approach	**URL:** **http://www-lipn.univ-paris13.fr/~szulman/TERMINAE.html**
Bachimont's method	**URL:** **http://opales.ina.fr/public/**
Faatz and Steinmetz approach	**URL:** Not available
Gupta and colleagues' approach	**URL:** Not available
Hahn and colleagues' method	**URL:** Not available
Hearst's approach	**URL:** **http://www.ii.uam.es/~ealfon**
Hwang's method	**URL:** **http://www.argreenhouse.com/InfoSleuth/index.shtml**
Khan and Luo's method	**URL:** Not available
Kietz and colleagues' method	**URL:** **http://ontoserver.aifb.uni-karlsruhe.de/texttoonto/**
Lonsdale and colleagues' method	**URL:** **http://www.ttt.org/salt/index.html**
Missikoff and colleagues' method	**URL:** Not available
Moldovan and Girju's method	**URL:** Not available
Nobécourt approach	**URL:** Not available
Roux and colleagues' approach	**URL:** Not available
Wagner approach	**URL:** Not available
Xu and colleagues' approach	**URL:** Not available

OL from dictionary

Hearst's method	**URL:** Not available
Rigau and colleagues' method	**URL:** http://www.lsi.upc.es/~rigau/
Jannink and Wiederhold's approach	**URL:** Not available

OL from knowledge bases

OL from semi-structured schemata

Deitel and colleagues' approach	**URL:** http://mondeca-publishing.com/s/anonymous/title11884.html
Doan and colleagues approach	**URL:** Not available
Papatheodorou and colleagues' method	**URL:** http://www.educanext.org/
Volz and colleagues' approach	**URL:** http://www.aifb.uni-karlsruhe.de/WBS/rvo/raphael-bib.html#wonderweb-D11

OL from relational schemata

Johannesson's method	**URL:** Not available
Kashyap's method	**URL:** Not available
Rubin and colleagues' approach	**URL:** http://www.nigms.nih.gov/funding/pharmacogenetics.html
Stojanovic and colleagues' approach	**URL:** http://wonderweb.semanticweb.org/publications.shtml

Criteria to descr ibe methods and techniques

- General Description, including its main goals and scope
- General steps used for learning
- Knowledge sources used for learning
- Main techniques applied in the process
- Possibility of reusing other ontologies
- Domains in which it has been tested
- Tools associated
- Most relevant ontologies built following it
- Bibliography
- URL

Comparison of OL methods from texts

Name	Main goal	Main techniques used	Reuse other ontologies	Sources used for learning	Tool associated	Evaluation	Bibliography
Aguirre and colleagues' method	To enrich concepts in existing ontologies	Statistical approach Clustering Topic signatures	Yes	Domain Text WordNet	Information not available in papers	User	Agirre et al., 2000
Alfonseca and Manandhar's method	To enrich an existing ontology with new concepts	Topic signatures Semantic distance	Yes	Domain text WordNet	Welkin	Expert	Alfonseca et al., 2002
Aussenac-Gilles and colleagues' approach	To learn concepts and relations among them	Linguistic analysis Clustering techniques	Yes	Domain Text Domain ontologies	GEDITERM TERMINAE	User	Aussenac-Gilles and colleagues, 2000a and 2000b
Bachimont's method	To build a taxonomy	NLP techniques	No	Domain text	DOE	Expert	Bachimont et al., 2002
Faatz and Steinmetz approach	To enrich an existing ontology with new concepts	Statistical approach Semantic distance	Yes	Domain corpus Domain ontology	Any ontology workbench	Expert	Faatz et al 2002
Gupta and colleagues' approach	To build sub-languages in WordNet	NLP techniques Term-extraction techniques	Yes	Domain text WordNet	SubWordNet Engineering tool	Expert	Gupta et al. 2002
Hahn and colleagues' method	To learn new concepts	Concept hypothesis based on linguistic and conceptual quality labels Statistical approach	No	Domain text	Information not available in papers	Empirical measures and by an expert	Hahn et al 1998
Hearst's approach	To enrich an existing ontology	NLP techniques Linguistic patterns	Yes	Domain Text WordNet	Welkin	Expert	Hearst 1998 and Alfonseca et al. 2002

...

Criteria followed to describe tools

- General Description including its main goals and scope
- Main techniques used by the tool
- Method followed
- Software architecture
- Interoperability with other tools
- Inport and export facilities
- Interface facilities
- URL
- Bibliography

OL from texts: tools

18 tools described

ASIUM	**URL:** **http://www.lri.fr/~faure/Demonstration/Presentation_Demo.html**
Caméléon	**URL:** Not available
Corporum-Ontobuilder	**URL:** **http://ontoserver.cognit.no**
DOE	**URL:** **http://opales.ina.fr/public/**
KEA	**URL:** **http://www.nzdl.org/Kea/**
LTG	**URL:** **http://www.ltg.ed.ac.uk/%7Emikheev/workbench.html**
MO'K Workbench	**URL:** Not available
OntoLearn	**URL:** Not available
Prométhée	**URL:** **http://www.sciences.univ-nantes.fr/info/perso/permanents/morin/promethee/**
SOAT	**URL:** **http://www.iis.sinica.edu.tw/IASL/en/index.htm**
SubWordNet E.P.	**URL:** **http://www.aic.nrl.navy.mil/~aha/cbr/luikm.html**
SVETLAN'	**URL:** **http://www.limsi.fr/Individu/gael/ManuscritThese/**
TDIDF	**URL:** Not available
TERMINAE	**URL:** **http://www-lipn.univ-paris13.fr/~szulman/TERMINAE.html**
TextStorm and Clouds	**URL:** Not available
TextToOnto	**URL:** **http://ontoserver.aifb.uni-karlsruhe.de/texttoonto/**
Welkin	**URL:** **http://www.ii.uam.es/~ealfon**
WOLFIE	**URL:** Not available

OL from dictionary

SEID **URL:** http://www.lsi.upc.es/~rigau/

DOODLE **URL:** Not available

OL from knowledge bases

OL from semi-structured schemata

OntoBuilder **URL:** http://www.cs.msstate.edu/~gmodica/Education/OntoBuilder/

OL from relational schemata

OL from texts. Tools

Name	Goal and scope	Learning technique	Method followed to learn	Sources	User intervention	Interoperability	Bibliography
ASIUM	To learn taxonomic relations	Conceptual clustering techniques	Own method	Text syntactically analysed	Whole process	Can be used to perform the knowledge acquisition to any other ontology development tool	Faure et al 2000, 1999, and 1998
Caméléon	To tune generic lecixo-syntactic patterns or build new ones. To find taxonomic and non taxonomic lexical relations in texts and to enrich a conceptual model with these lexical relations	To reuse and tuning of generic patterns (mainly for taxonomic relations), hearst's proposal, and pattern identification in text help to learn lexical relations and their validation leads to conceptual relations	Own method	Texts processed by taggers Its own base of generic patterns	Validates, adapts, or defines new domain specific patterns and relations Domain expert just validates the model	Imports lists of terms from any term extractor	Aussenac-Gilles and Seguela 2000
Corporum-Ontobuilder	To extract initial taxonomy	Linguistic and semantic techniques	Own method	Text	Not necessary	OntoWrapper and OntoExtract	Engels 2001 and 2000
DOE	To help to the ontologist in the process of building an ontology	Differential Semantic	Bachimont's method	NL text	Whole process	None	Bachimont 2000

...

Conclusions about Ontology learning

- **Ontology learning is a suitable process:**
  - **to accelerate the knowledge acquisition process necessary to build an ontology from scratch,**
  - **to reduce the time required to enrich an existing ontology,**
  - **to speed up the construction of ontologies to be used for different purposes in the Semantic Web.**
- **integrated methods and techniques are needed for achieving the goal.**

Ontology-based annotation tools

Ontology based annotation tools

- **Used for Ontology population**
- **Main Features**
  - **Language for storing the annotations**
  - **Language for handling ontologies**
  - **Automatization degree of the annotation process**
  - **Static/dynamic page annotation**
  - **Text/image annotation**

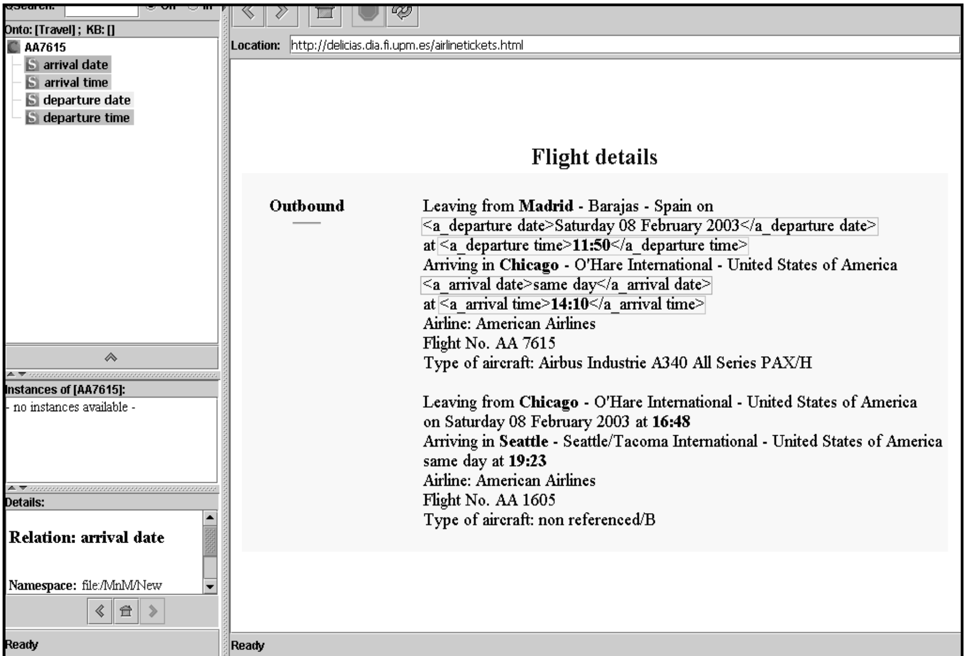
AEroDAML

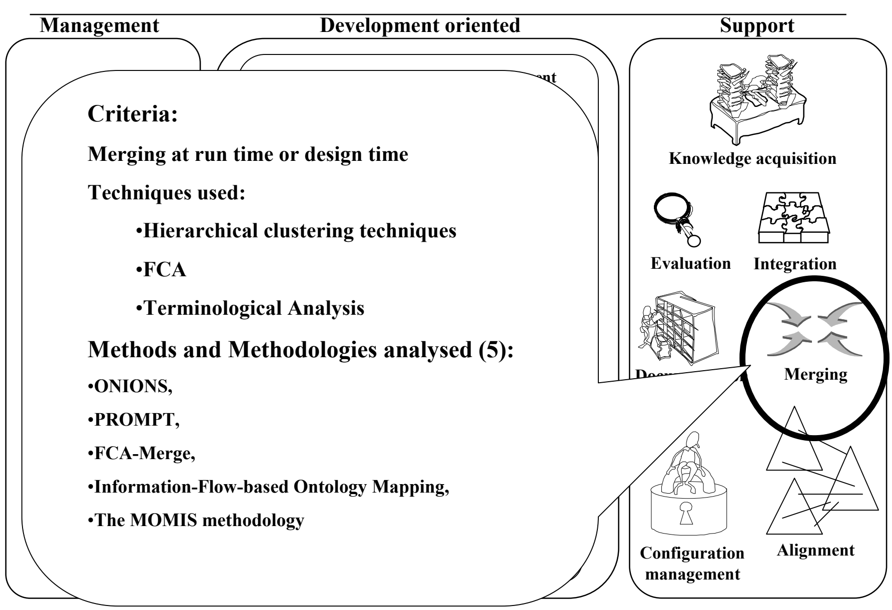
COHSE

MnM

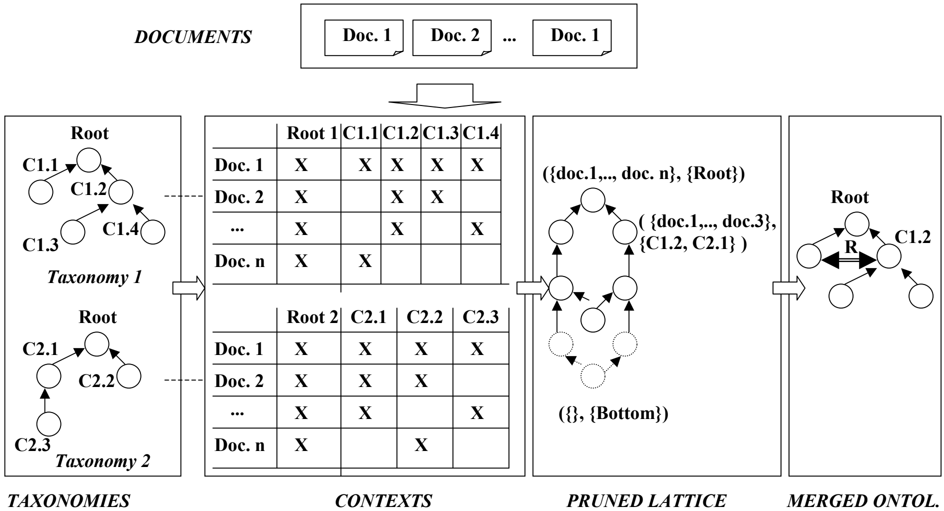
OntoAnnotate

SHOE Knowledge Annotator


Onto: [Travel] ; KB: []
AA7615
arrival date
arrival time
departure date
departure time
Instances of [AA7615]:
- no instances available -
Details:
Relation: arrival date
Namespace: file:/MnM/New
Ready
Location: http://delicias.dia.fi.upm.es/airlinetickets.html
Flight details
Outbound
Leaving from Madrid - Barajas - Spain on
<a_departure date>Saturday 08 February 2003</a_departure date>
at <a_departure time>11:50</a_departure time>
Arriving in Chicago - O'Hare International - United States of America
<a_arrival date>same day</a_arrival date>
at <a_arrival time>14:10</a_arrival time>
Airline: American Airlines
Flight No. AA 7615
Type of aircraft: Airbus Industrie A340 All Series PAX/H
Leaving from Chicago - O'Hare International - United States of America
on Saturday 08 February 2003 at 16:48
Arriving in Seattle - Seattle/Tacoma International - United States of America
same day at 19:23
Airline: American Airlines
Flight No. AA 1605
Type of aircraft: non referenced/B
Ready


Management
Development oriented
Support
Knowledge acquisition
Evaluation
Integration
Merging
Configuration management
Alignment
Criteria:
Merging at run time or design time
Techniques used:
•Hierarchical clustering techniques
•FCA
•Terminological Analysis
Methods and Methodologies analysed (5):
•ONIONS,
•PROMPT,
•FCA-Merge,
•Information-Flow-based Ontology Mapping,
•The MOMIS methodology

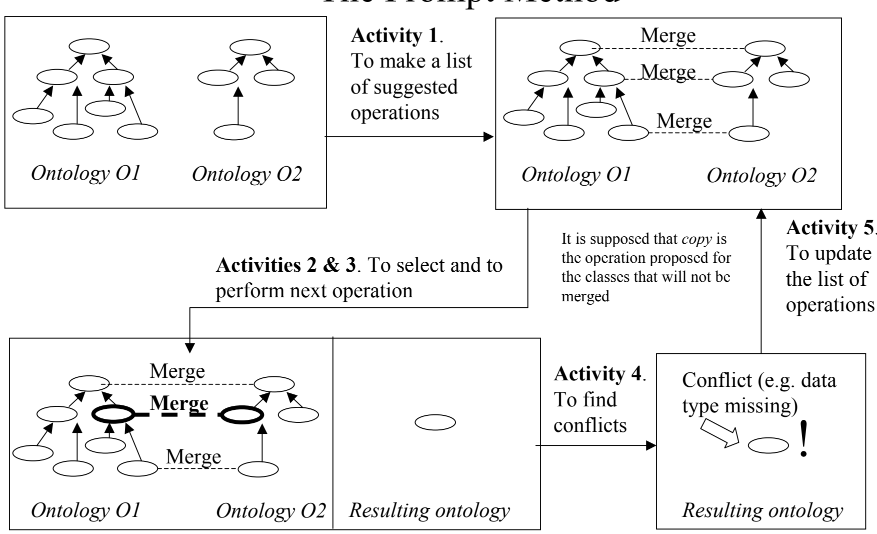
FCA-Merge



	Root 1	C1.1	C1.2	C1.3	C1.4
Doc. 1	X	X	X	X	X
Doc. 2	X		X	X	
...	X		X		X
Doc. n	X	X			

	Root 2	C2.1	C2.2	C2.3
Doc. 1	X	X	X	X
Doc. 2	X	X	X	
...	X	X		X
Doc. n	X		X	

The Prompt Method

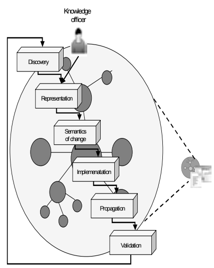


Management **Development oriented** **Support**

Ontology Evolution:
The ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology [Noy and Klein, 02].

Approaches

1. **METHONTOLOGY,**
   - **Activity during the life cycle [Fernández-López et al., 97]**
   - **Identification of the elements to be controlled [Gómez-Pérez and Rojas, 99]**
   - **Control of changes**
   - **Generation of status reports.**
2. **Types of changes [Noy and Klein, 02].**
3. **Klein and Fensel [Klein and Fensel, 01]:**
   - **Identification**
   - **Change specification**
   - **Transparent evolution**
4. **Stojanovic's Process [Stojanovic et al., 02]:**



Knowledge acquisition

Evaluation **Integration**

Documentation **Merging**

Configuration management **Alignment**

Management **Development oriented** **Support**

Pre-development

Criteria:

•Content Evaluation on taxonomies

•Criteria: consistency, completeness

Methods analysed (3):

•Gómez-Pérez approach for taxonomy evaluation

•OntoClean Method

•Ontological Constrains Manager (OCM)

dy **Development** **alization** **plementation**

Quality assurance

Post-development

Maintenance **Use**

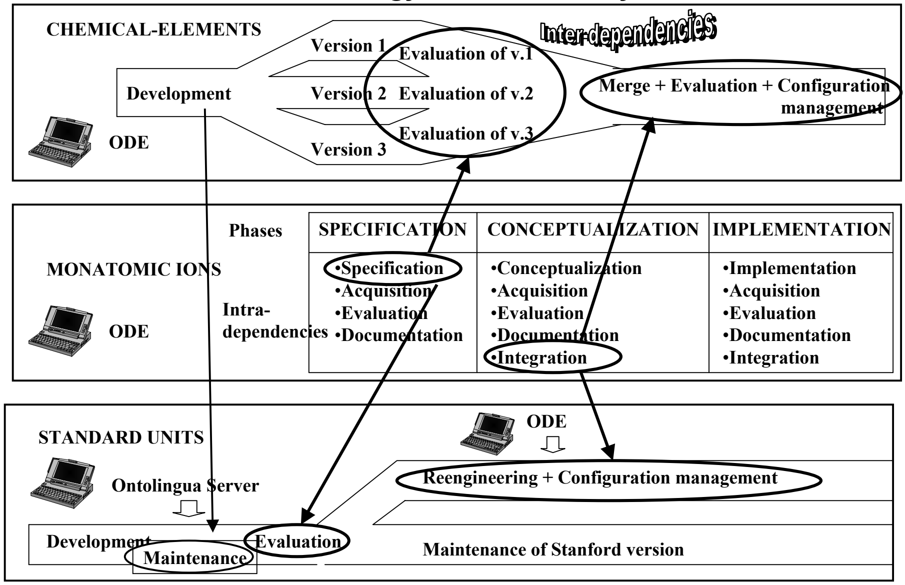
Knowledge acquisition

Evaluation **Integration**

Documentation **Merging**

Configuration management **Alignment**

Ontology's crossed life cycles



Conclusions

- **There exist stable methodologies and tools for building ontologies, but they do not cover all the process of the ontology development process.**
  - **Methontology (the recommended methodology to ontology development by FIPA)**
  - **On-To-Knowledge**
- **There exist methods and tools for specific tasks**
  - **Reengineering**
  - **Collaborative construction**
  - **Merging**
  - **Evaluating**
  - **Evolution**
  - **Ontology Learning**
- **Integration of specific methods in methodologies are needed**
- **Technological support for the whole ontology development process**

Methodologies for building ontologies (I)

Methodologies for building ontologies from the scratch.

Cyc methodology	**URL: http://www.cyc.com**
Uschold and King	**URL:** ***Not available***
Grüninger and Fox	**URL:** ***Not available***
KACTUS methodology	**URL:** ***Not available***
METHONTOLOGY	**URL:** ***Not available***
SENSUS methodology	**URL:** ***Not available***
On-To-Knowledge Methodology	**URL: http://www.ontoknowledge.org/**

Methodologies for reengineering ontologies

Method for reengineering ontologies integrated in Methontology **URL:** ***Not available***

Methodologies for cooperative construction of ontologies

CO4 methodology	**URL:** ***Not available***
$(KA)^2$ methodology	**URL:** ***Not available***

Methodologies for building ontologies (II)

Ontology learning methodologies

Aussenac-Gille's and colleagues methodology	**URL: http://www.biomath.jussieu.fr/TIA/**
Maedche and colleagues' methodology	**URL:** ***Not available***

Ontology merge methodologies

FCA-merge	**URL:** ***Not available***
PROMPT	**URL:** ***Not available***
ONIONS	**URL:** ***Not available***

Ontology evaluation methods

OntoClean: Guarino's group methodology	**URL:** ***Not available***
Gómez Pérez's evaluation methodology	**URL:** ***Not available***

To know more about this topics

Gómez-Pérez, A.; Fernández-López, M.; Corcho, O. Ontological Engineering. Springer Verlag. 2003

Ontoweb WP1: D1.1.1 http://www.ontoweb.org

WP1: D1.3 Survey on Tools

WP1: D1.4 Survey on methodologies

WP1: D1.5 Survey on ontology learning

OntoRoadMap

http://babage.dia.fi.upm.es/ontoweb/wp1/OntoRoadMap/index.html

Table of Contents

Ontology Languages

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
{asun, mfernandez, ocorcho}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

References

[Bray et al, 98] Bray, T., Paoli, J., Sperberg, C. Extensible Markup Language (XML) 1.0. W3C Recommendation. Feb 1998. http://www.w3.org/TR/REC-xml.

[Brickley et al, 99] Brickley, D., Guha, R.V. Resource Description Framework (RDF) Schema Specification. W3C Proposed Recommendation. March, 1999. http://www.w3.org/TR/PR-rdf-schema.

[Chaudhri et al, 97] Chaudhri, V., Farquhar, A, Fikes, R., Karp, P., Rice, J. The Generic Frame Protocol 2.0. July, 1997.

[Corcho et al, 00] Corcho, O., Gómez-Pérez, A. The Languages of the Semantic Web. Submitted to IEEE Intelligent Systems & their applications. Special Issue on Semantic Web. 2000.

[Farquhar et al, 97] Farquhar, A., Fikes, R., Rice, J. The Ontolingua Server: A Tool for Collaborative Ontology Construction. International Journal of Human Computer Studies. 46(6). pp: 707-728.

[Genesereth et al, 92] Genesereth, M., Fikes, R. Knowledge Interchange Format. Technical Report. Computer Science Department. Stanford University. Logic-92-1. 1992.

[Gruber, 93] Gruber, R. A translation approach to portable ontology specification. Knowledge Acquisition. #5: 199-220. 1993.

[Harmelen et al, 99] Harmelen, F., Fensel, D. Surveying notations for machine-processable semantics of Web sources. Proceedings of the IJCAI'99 Workshop on Ontologies & PSMs. 1999.

References

[Horrocks et al, 00] Horrocks, I., Fensel, D., Harmelen, F., Decker, S., Erdmann, M, Klein, M. OIL in a Nutshell. Proceedings of the ECAI'00 Workshop on Application of Ontologies and PSMs. Berlin. Germany. August, 2000.

[Karp et al, 99] Karp, R., Chaudhri, V., Thomere, J. XOL: An XML-Based Ontology Exchange Language. July, 1999.

[Kent, 98] Kent, R. Conceptual Knowledge Markup Language (version 0.2). 1998. .

[Kifer et al, 95] Kifer, M., Lausen, G., Wu, J. Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM. 1995.

[Lassila et al, 99] Lassila, O., Swick, R. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. January, 99. .

[Lenat et al, 90] Lenat, D.B., Guha, R.V. Building Large Knowledge-based systems. Representation and Inference in the Cyc Project. Addison-Wesley. Reading. Massachusetts. 1990.

[Luke et al, 00] Luke S., Heflin J. SHOE 1.01. Proposed Specification. SHOE Project. February, 2000.

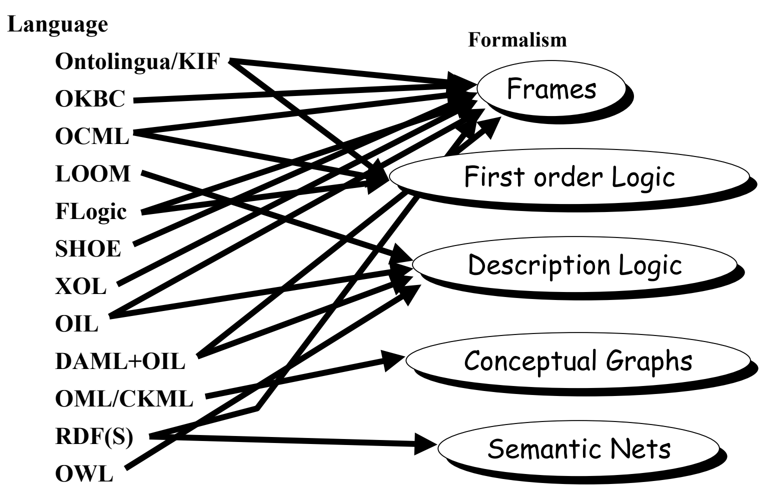
[MacGregor, 91] MacGregor, R. Inside the LOOM clasifier. SIGART bulletin. #2(3):70-76. June, 1991.

References

[Motta, 99] Motta, E. Reusable Components for Knowledge Modelling. IOS Press. Amsterdam. 1999.

[Van Harmelen et al, 01] van Harmelen, F., Patel-Schneider, P., Horrocks, I. Reference description of the DAML+OIL (March 2001) ontology markup language. Technical Report. March, 2001.

KR Formalisms



Index

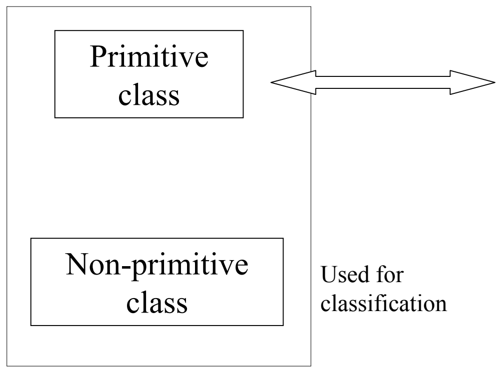
- 1. Evaluation Framework
- 2. Ontology Languages
  - 2.1 Traditional languages
  - 2.2 Ontology markup languages
- 3. Comparison between languages
- 4. Examples

Evaluation Framework

KR (Expressiveness)	Inference mechanisms (Reasoning)
Classes: Metaclasses, Attributes, Facets	Exceptions, Automatic Classifications
Taxonomies	Inheritance: Monotonic, non-monotonic; Simple, Multiple
Procedures	Execution of procedures
Relations/Functions, Instances/Individuals/ Facts/Claims, Axioms	Constraint Checking
Production Rules	Reasoning with rules

Classes and attributes (I)

Different types of classes



OKBC, LOOM, OIL **Others**

Classes and attributes (II)

Cardinality constraints

Class Desk	
Attributes	(Cardinality)
Price	(0,2)
Number of legs	(1,1)
Made of	(2,N)

Ontolingua

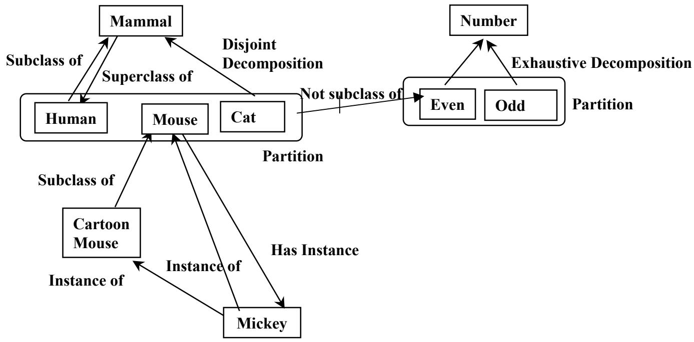
Class Desk	
Attributes	(Cardinality)
Price	(0,N)
Number of legs	(0,1)
Made of	(0,N)

FLogic

Class taxonomies (I)

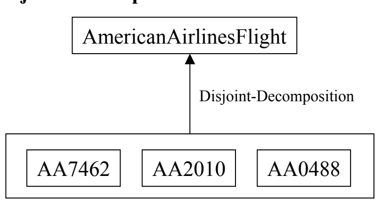
Taxonomies

- **Subclass of / Not subclass of**
- **Partition**
- **Disjoint Decomposition**
- **Exhaustive Decomposition**

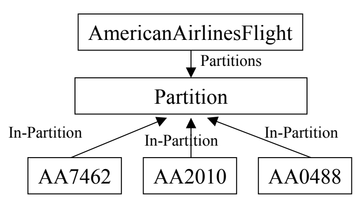


Class taxonomies (II)

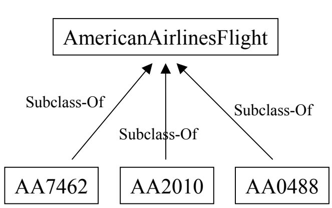
Disjoint Decompositions



a) In Ontolingua

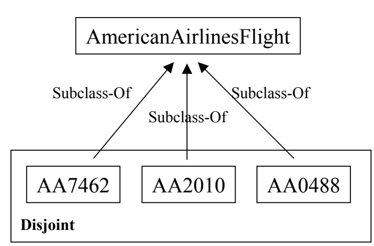


b) In LOOM



FORALL flight flight:AA7462 <-> NOT flight:AA2010.
FORALL flight flight:AA2010 <-> NOT flight:AA0488.
FORALL flight flight:AA7462 <-> NOT flight:AA0488.

c) In FLogic



d) In OIL

Relations and functions (I)

```
(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)))
```

Ontolingua
Not executable

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

OCML
Executable

```
Buyer[pays@Room,Number => Number].
B[pays@Prod,D -> F] ß
    B:Buyer AND Prod:Room AND D:Number AND F:Number
    AND Prod[price->P] AND F = P – (P * D / 100).
```

FLogic
Executable

Formal axioms (I)

Axioms
- **First order logic**
- **Second order logic**
- **Named axioms**

Ontolingua	LOOM	FLogic
Class Dog: Embedded axiom1, Embedded axiom2	Class Dog: Embedded axiom1, Embedded axiom2	Class Dog: ~~Embedded axiom1~~, ~~Embedded axiom2~~
Named axiom 1	~~Named axiom 1~~	~~Named axiom 1~~
		Unnamed axiom 1

Instances (I)

```
<AA7462 rdf:ID="AA7462Feb082002">
  <singleFare>300 US Dollars</singleFare>
  <departureDate rdf:datatype="&xsd;date">
    2002-02-08
  </departureDate>
  <arrivalPlace rdf:resource="#Seattle"/>
</AA7462>
```

RDF(S)

```
<INSTANCE KEY="AA7462-Feb08-2002">
  <USE-ONTOLOGY ID="Travel-Ontology"
    URL="http://delicias.dia.fi.upm.es/SHOE/travel.html"
    VERSION="1.0" PREFIX="travel">
  <CATEGORY NAME="travel.AA7462">
  <RELATION NAME="travel.singleFare">
    <ARG POS=1 VALUE="me">
    <ARG POS=2 VALUE="300USDollars">
  </RELATION>
  <RELATION NAME="travel.departureDate">
    <ARG POS=1 VALUE="me">
    <ARG POS=2 VALUE="Feb8-2002">
  </RELATION>
  <RELATION NAME="travel.arrivalPlace">
    <ARG POS=1 VALUE="me">
    <ARG POS=2 VALUE="Seattle">
  </RELATION>
</INSTANCE>
```

SHOE

Inference mechanisms (I)

Several definitions depending on the existence of an inference engine for the language

Ontolingua (no inference engine)

```
(define-Axiom NoTrainfromUSAtoEurope
 "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)))))))
```

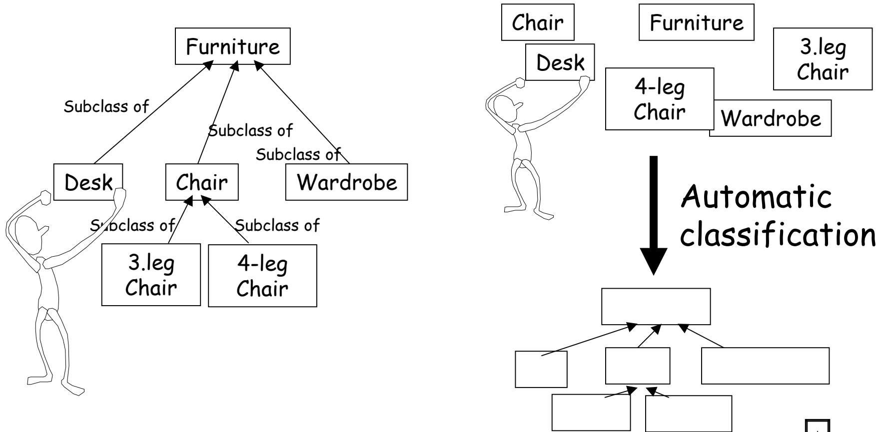
LOOM (with inference engine)

```
(defconcept Train-Travel
:is           (:and  Travel
                 (:satisfies ?x (:for-all ?y (:for-all ?z
                   (:not (:and (arrivalPlace ?x ?y)
                                               (EuropeanLocation ?y)
                                               (departurePlace ?x ?z)
                                               (USALocation ?z))))))))
```

(Valente et al, 99) Valente, Russ, MacGregor, Swartout. **Building and (Re)Using a Ontology of Air Campaign Planning.** ***IEEE Intelligent Systems.*** **14. #1 (1999) 27-36**

Inference mechanisms (II)

Automatic classifications performed by the inference engine of a language



RDF(S)

No automatic classification

LOOM

Automatic classification

Ontology Languages (I)

Traditional ontology languages

- **Ontolingua/KIF**
- **OKBC**
- **OCML**
- **LOOM**
- **FLogic**

Ontology Languages (II)

Ontology markup languages

Standards & Recommendations of W3C

XML

RDF(S)

Ontology specification languages

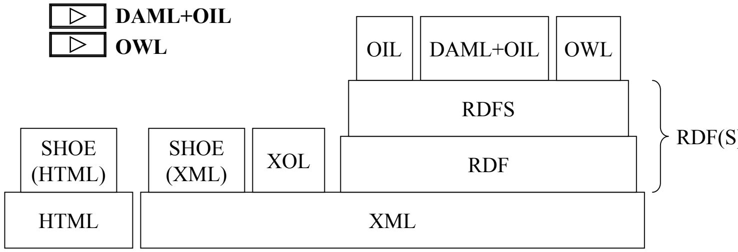
SHOE

XOL

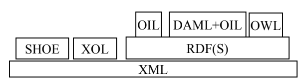
OIL

DAML+OIL

OWL



OWL



Web Ontology Language

see code

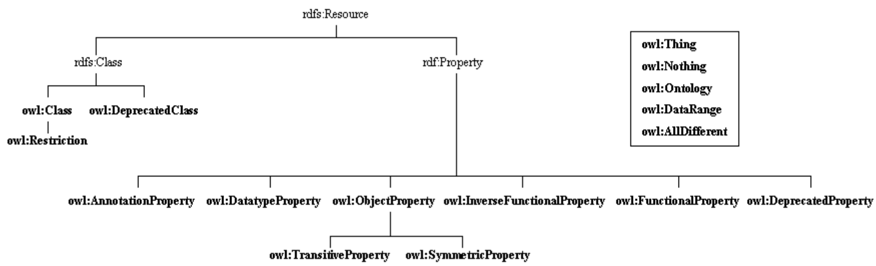
Built on top of RDF(S) and renaming DAML+OIL primitives

3 layers:
-OWL Lite: a small subset, easier for frame-based tools to transition to, easier reasoning
-OWL DL: description logic, decidable reasoning
-OWL Full: RDF extension, allows metaclasses

Several syntaxes:
-Abstract syntax: easier to read and write manually, closely corresponds to DL
-RDF/XML: OWL can be parsed as an RDF document, more verbose

Dean M, Schreiber G. OWL Web Ontology Language 1.0 Reference. December 2003.

Class taxonomy of the OWL KR ontology



Property list of the OWL KR ontology

Property name	domain	range
owl:intersectionOf	owl:Class	rdf:List
owl:unionOf	owl:Class	rdf:List
owl:complementOf	owl:Class	owl:Class
owl:oneOf	owl:Class	rdf:List
owl:onProperty	owl:Restriction	rdf:Property
owl:allValuesFrom	owl:Restriction	rdfs:Class
owl:hasValue	owl:Restriction	*not specified*
owl:someValuesFrom	owl:Restriction	rdfs:Class
owl:minCardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:maxCardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:cardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:inverseOf	owl:ObjectProperty	owl:ObjectProperty
owl:sameAs	owl:Thing	owl:Thing
owl:equivalentClass	owl:Class	owl:Class
owl:equivalentProperty	rdf:Property	rdf:Property
owl:sameIndividualAs	owl:Thing	owl:Thing
owl:differentFrom	owl:Thing	owl:Thing
owl:disjointWith	owl:Class	owl:Class
owl:distinctMembers	owl:AllDifferent	rdf:List
owl:versionInfo	*not specified*	*not specified*
owl:priorVersion	owl:Ontology	owl:Ontology
owl:incompatibleWith	owl:Ontology	owl:Ontology
owl:backwardCompatibleWith	owl:Ontology	owl:Ontology
owl:imports	owl:Ontology	owl:Ontology

OWL DL

Class expressions allowed in: rdfs:domain, rdfs:range, rdfs:subClassOf
owl:intersectionOf, owl:equivalentClass, owl:allValuesFrom, owl:someValuesFrom
Values are not restricted (0..N) in: owl:minCardinality, owl:maxCardinality, owl:cardinality

owl:DataRange, rdf:List, rdf:first, rdf:rest, rdf:nil

owl:hasValue (*daml:hasValue*)
owl:oneOf (*daml:oneOf*)
owl:unionOf (*daml:unionOf*), owl:complementOf (*daml:complementOf*)
owl:disjointWith (*daml:disjointWith*)

OWL Lite

owl:Ontology (*daml:Ontology*),
owl:versionInfo (*daml:versionInfo*),
owl:imports (*daml:imports*),
owl:backwardCompatibleWith,
owl:incompatibleWith, owl:priorVersion,
owl:DeprecatedClass,
owl:DeprecatedProperty

owl:Class (*daml:Class*),
owl:Restriction (*daml:Restriction*),
owl:onProperty (*daml:onProperty*),
owl:allValuesFrom (*daml:toClass*) (only with class identifiers and named datatypes),
owl:someValuesFrom (*daml:hasClass*) (only with class identifiers and named datatypes),
owl:minCardinality (*daml:minCardinality*, restricted to {0,1}),
owl:maxCardinality (*daml:maxCardinality*, restricted to {0,1}),
owl:cardinality (*daml:cardinality*, restricted to {0,1})

owl:intersectionOf (only with class identifiers and property restrictions)

owl:ObjectProperty (*daml:ObjectProperty*),
owl:DatatypeProperty (*daml:DatatypeProperty*),
owl:TransitiveProperty (*daml:TransitiveProperty*),
owl:SymmetricProperty,
owl:FunctionalProperty (*daml:UniqueProperty*),
owl:InverseFunctionalProperty (*daml:UnambiguousProperty*),
owl:AnnotationProperty

owl:Thing (*daml:Thing*)
owl:Nothing (*daml:Nothing*)

owl:inverseOf (*daml:inverseOf*),
owl:equivalentClass (*daml:sameClassAs*) (only with class identifiers and property restrictions),
owl:equivalentProperty (*daml:samePropertyAs*),
owl:sameAs (*daml:equivalentTo*),
owl:sameIndividualAs,
owl:differentFrom (*daml:differentIndividualFrom*),
owl:AllDifferent, owl:distinctMembers

RDF(S)

rdf:Property
rdfs:subPropertyOf
rdfs:domain
rdfs:range (only with class identifiers and named datatypes)
rdfs:comment, rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy
rdfs:subClassOf (only with class identifiers and property restrictions)

OWL DL

Class expressions allowed in: rdfs:domain, rdfs:range, rdfs:subClassOf
owl:intersectionOf, owl:equivalentClass, owl:allValuesFrom, owl:someValuesFrom
Values are not restricted (0..N) in: owl:minCardinality, owl:maxCardinality, owl:cardinality

owl:DataRange, rdf:List, rdf:first, rdf:rest, rdf:nil

owl:hasValue (*daml:hasValue*)
owl:oneOf (*daml:oneOf*)
owl:unionOf (*daml:unionOf*), owl:complementOf (*daml:complementOf*)
owl:disjointWith (*daml:disjointWith*)

OWL Lite

owl:Ontology (*daml:Ontology*),
owl:versionInfo (*daml:versionInfo*),
owl:imports (*daml:imports*),
owl:backwardCompatibleWith,
owl:incompatibleWith, owl:priorVersion,
owl:DeprecatedClass,
owl:DeprecatedProperty

owl:Class (*daml:Class*),
owl:Restriction (*daml:Restriction*),
owl:onProperty (*daml:onProperty*),
owl:allValuesFrom (*daml:toClass*) (only with class identifiers and named datatypes),
owl:someValuesFrom (*daml:hasClass*) (only with class identifiers and named datatypes),
owl:minCardinality (*daml:minCardinality*, restricted to {0,1}),
owl:maxCardinality (*daml:maxCardinality*, restricted to {0,1}),
owl:cardinality (*daml:cardinality*, restricted to {0,1})

owl:intersectionOf (only with class identifiers and property restrictions)

owl:ObjectProperty (*daml:ObjectProperty*),
owl:DatatypeProperty (*daml:DatatypeProperty*),
owl:TransitiveProperty (*daml:TransitiveProperty*),
owl:SymmetricProperty,
owl:FunctionalProperty (*daml:UniqueProperty*),
owl:InverseFunctionalProperty (*daml:UnambiguousProperty*),
owl:AnnotationProperty

owl:Thing (*daml:Thing*)
owl:Nothing (*daml:Nothing*)

owl:inverseOf (*daml:inverseOf*),
owl:equivalentClass (*daml:sameClassAs*) (only with class identifiers and property restrictions),
owl:equivalentProperty (*daml:samePropertyAs*),
owl:sameAs (*daml:equivalentTo*),
owl:sameIndividualAs,
owl:differentFrom (*daml:differentIndividualFrom*),
owl:AllDifferent, owl:distinctMembers

RDF(S)

rdf:Property
rdfs:subPropertyOf
rdfs:domain
rdfs:range (only with class identifiers and named datatypes)
rdfs:comment, rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy
rdfs:subClassOf (only with class identifiers and property restrictions)

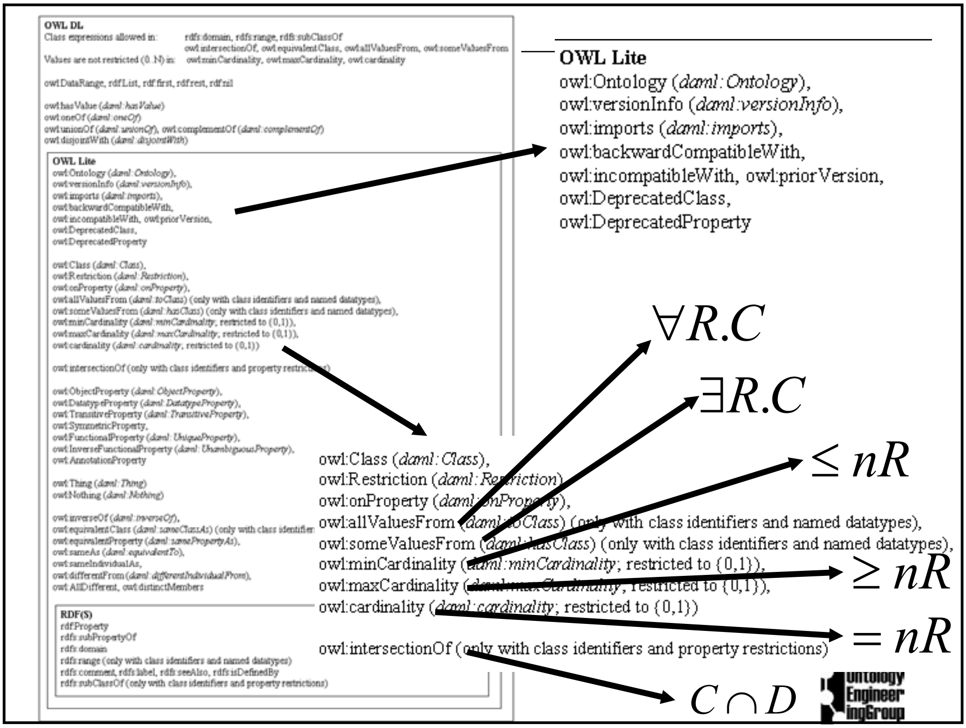
R

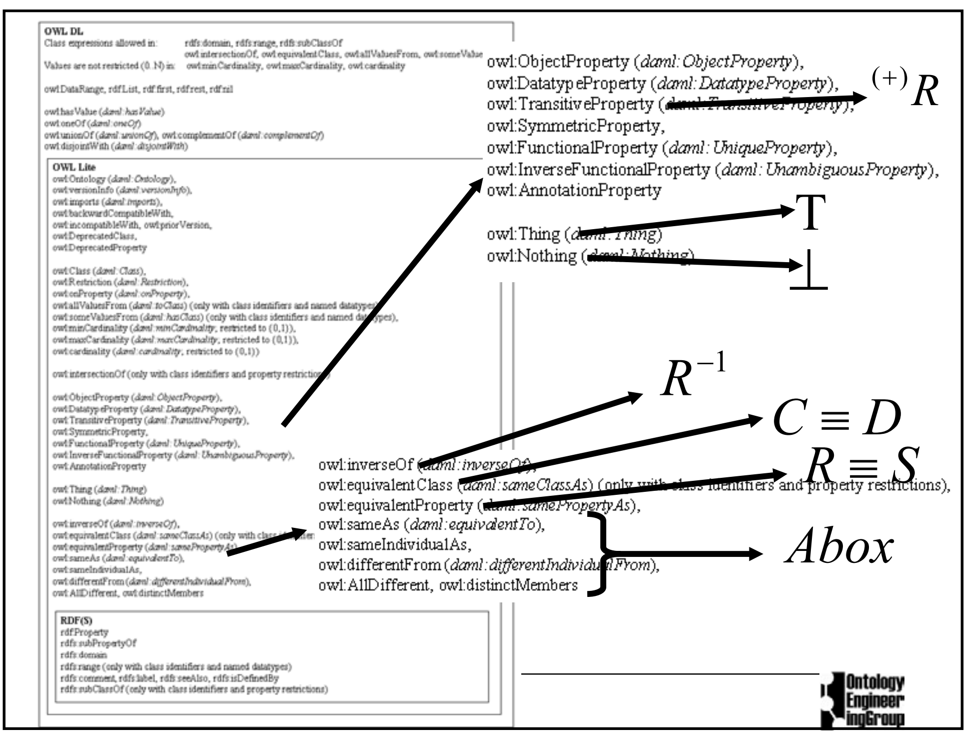
$R \subseteq S$

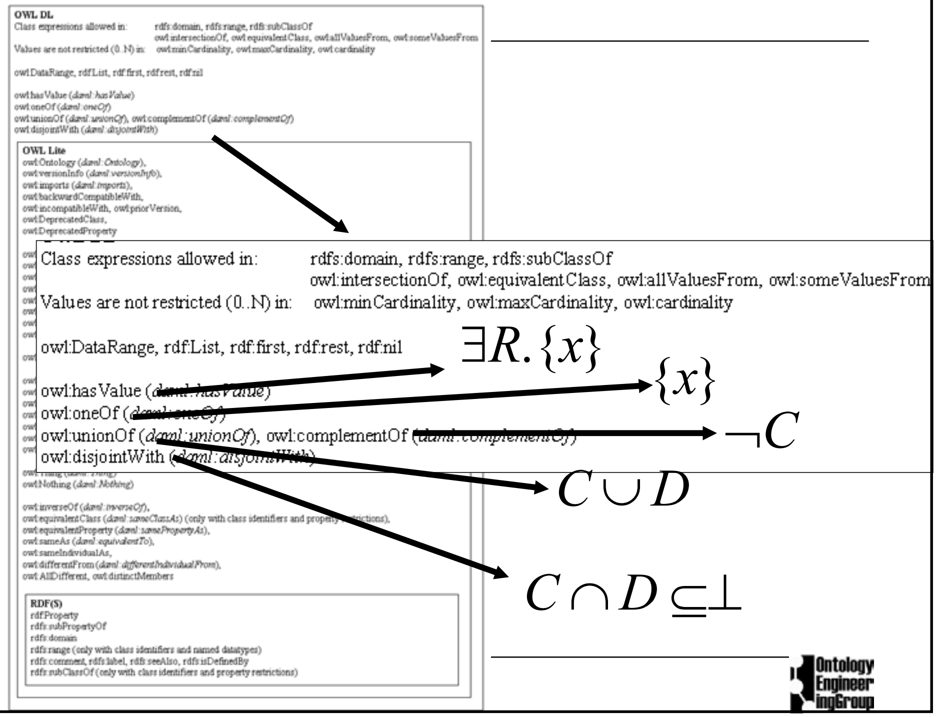
RDF(S)

rdf:Property
rdfs:subPropertyOf
rdfs:domain
rdfs:range (only with class identifiers and named datatypes)
rdfs:comment, rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy
rdfs:subClassOf (only with class identifiers and property restrictions)

$C \subseteq D$


OWL Lite
owl:Ontology (daml: Ontology),
owl:versionInfo (daml: versionInfo),
owl:imports (daml: imports),
owl:backwardCompatibleWith,
owl:incompatibleWith, owl:priorVersion,
owl:DeprecatedClass,
owl:DeprecatedProperty
$\forall R.C$
$\exists R.C$
$\leq nR$
owl:Class (daml: Class),
owl:Restriction (daml: Restriction),
owl:onProperty (daml: onProperty),
owl:allValuesFrom (daml: toClass) (only with class identifiers and named datatypes),
owl:someValuesFrom (daml: hasClass) (only with class identifiers and named datatypes),
owl:minCardinality (daml: minCardinality; restricted to {0,1}),
owl:maxCardinality (daml: maxCardinality; restricted to {0,1}),
owl:cardinality (daml: cardinality; restricted to {0,1})
$\geq nR$
$= nR$
owl:intersectionOf (only with class identifiers and property restrictions)
$C \cap D$
Ontology Engineering Group


owl:ObjectProperty (daml: ObjectProperty),
owl:DatatypeProperty (daml: DatatypeProperty),
owl:TransitiveProperty (daml: TransitiveProperty),
owl:SymmetricProperty,
owl:FunctionalProperty (daml: UniqueProperty),
owl:InverseFunctionalProperty (daml: UnambiguousProperty),
owl:AnnotationProperty
$^{(+)}R$
owl:Thing (daml: Thing)
owl:Nothing (daml: Nothing)
$\top$
$\bot$
R^{-1}
$C \equiv D$
$R \equiv S$
owl:inverseOf (daml: inverseOf),
owl:equivalentClass (daml: sameClassAs) (only with class identifiers and property restrictions),
owl:equivalentProperty (daml: samePropertyAs),
owl:sameAs (daml: equivalentTo),
owl:sameIndividualAs,
owl:differentFrom (daml: differentIndividualFrom),
owl:AllDifferent, owl:distinctMembers
Abox
Ontology Engineering Group



OWL Example

Develop a sample ontology in the domain of people, pets, vehicles, and newspapers

- **Practice with DL syntax, OWL abstract syntax and OWL RDF/XML syntax**
- **Understand the basic primitives of OWL Lite and OWL DL**
- **Understand the basic reasoning mechanisms of OWL DL**

  Subsumption

    Automatic classification: an ontology built collaboratively

    Instance classification

    Detecting redundancy

  Consistency checking: unsatisfiable restrictions in a Tbox (are the classes coherent?)

Chunk 1. Formalize in DL, and then in OWL DL

1. Concept definitions:
   Grass and trees must be plants. Leaves are parts of a tree but there are other parts of a tree that are not leaves. A dog must eat bones, at least. A sheep is an animal that must only eat grass. A giraffe is an animal that must only eat leaves. A mad cow is a cow that eats brains that can be part of a sheep.

2. Restrictions:
   Animals or part of animals are disjoint with plants or parts of plants.

3. Properties:
   Eats is applied to animals. Its inverse is eaten_by.

4. Individuals:
   Tom.
   Flossie is a cow.
   Rex is a dog and is a pet of Mick.
   Fido is a dog.
   Tibbs is a cat.

Chunk 2. Formalize in DL, and then in OWL DL

1. Concept definitions:
   Bicycles, buses, cars, lorries, trucks and vans are vehicles. There are several types of companies: bus companies and haulage companies.
   An elderly person must be adult. A kid is (exactly) a person who is young. A man is a person who is male and is adult. A woman is a person who is female and is adult. A grown up is a person who is an adult. And old lady is a person who is elderly and female. Old ladies must have some animal as pets and all their pets are cats.

2. Restrictions:
   Youngs are not adults, and adults are not youngs.

3. Properties:
   Has mother and has father are subproperties of has parent.

4. Individuals:
   Kevin is a person.
   Fred is a person who has a pet called Tibbs.
   Joe is a person who has at most one pet. He has a pet called Fido.
   Minnie is a female, elderly, who has a pet called Tom.

© Oscar Corcho, Mariano Fernánde-López, Asunción Gómez-Pérez

Chunk 3. Formalize in DL, and then in OWL DL

1. Concept definitions:

   A magazine is a publication. Broadsheets and tabloids are newspapers. A quality broadsheet is a type of broadsheet. A red top is a type of tabloid. A newspaper is a publication that must be either a broadsheet or a tabloid.
   White van mans must read only tabloids.

2. Restrictions:

   Tabloids are not broadsheets, and broadsheets are not tabloids.

3. Properties:

   The only things that can be read are publications.

4. Individuals:

   Daily Mirror
   The Guardian and The Times are broadsheets
   The Sun is a tabloid

Chunk 4. Formalize in DL, and then in OWL DL

1. Concept definitions:

   A pet is a pet of something. An animal must eat something. A vegetarian is an animal that does not eat animals nor parts of animals. Ducks, cats and tigers are animals. An animal lover is a person who has at least three pets. A pet owner is a person who has animal pets. A cat liker is a person who likes cats. A cat owner is a person who has cat pets. A dog liker is a person who likes dogs. A dog owner is a person who has dog pets.

2. Restrictions:

   Dogs are not cats, and cats are not dogs.

3. Properties:

   Has pet is defined between persons and animals. Its inverse is is_pet_of.

4. Individuals:

   Dewey, Huey, and Louie are ducks.
   Fluffy is a tiger.
   Walt is a person who has pets called Huey, Louie and Dewey.

Chunk 5. Formalize in DL, and then in OWL DL

1. Concept definitions
   A driver must be adult. A driver is a person who drives vehicles. A lorry driver is a person who drives lorries. A haulage worker is who works for a haulage company or for part of a haulage company. A haulage truck driver is a person who drives trucks ans works for part of a haulage company. A van driver is a person who drives vans. A bus driver is a person who drives buses. A white van man is a man who drives white things and vans.

2. Restrictions:
   --

3. Properties:
   The service number is an integer property with no restricted domain

4. Individuals:
   Q123ABC is a van and a white thing.
   The42 is a bus whose service number is 42.
   Mick is a male who read Daily Mirror and drives Q123ABC.

Chunk 1. Formalisation in DL

$$grass \subseteq plant$$

$$tree \subseteq plant$$

$$leaf \subseteq \exists partOf.tree$$

$$dog \subseteq \exists eats.bone$$

$$sheep \subseteq animal \cap \forall eats.grass$$

$$giraffe \subseteq animal \cap \forall eats.leaf$$

$$madCow \equiv cow \cap \exists eats.(brain \cap \exists partOf.sheep)$$

$$(animal \cup \exists partOf.animal) \cap (plant \cup \exists partOf.plant) \subseteq \bot$$

Chunk 2. Formalisation in DL

$bicycle \subseteq vehicle; bus \subseteq vehicle; car \subseteq vehicle; lorry \subseteq vehicle; truck \subseteq vehicle$

$busCompany \subseteq company; haulageCompany \subseteq company$

$elderly \subseteq person \cap adult$

$kid \equiv person \cap young$

$man \equiv person \cap male \cap adult$

$woman \equiv person \cap female \cap adult$

$grownUp \equiv person \cap adult$

$oldLady \equiv person \cap female \cap elderly$

$oldLady \subseteq \exists hasPet.animal \cap \forall hasPet.cat$

$young \cap adult \subseteq \perp$

$hasMother \subseteq hasParent$

$hasFather \subseteq hasParent$

Chunk 3. Formalisation in DL

$magazine \subseteq publication$

$broadsheet \subseteq newspaper$

$tabloid \subseteq newspaper$

$qualityBroadsheet \subseteq broadsheet$

$redTop \subseteq tabloid$

$newspaper \subseteq publication \cap (broadsheet \cup tabloid)$

$whiteVanMan \subseteq \forall reads.tabloid$

$tabloid \cap broadsheet \subseteq \perp$

Chunk 4. Formalisation in DL

$pet \equiv \exists isPetOf.\top$

$animal \subseteq \exists eats.\top$

$vegetarian \equiv animal \cap \forall eats.\neg animal \cap \forall eats.\neg(\exists partOf.animal)$

$duck \subseteq animal; cat \subseteq animal; tiger \subseteq animal$

$animalLover \equiv person \cap (\geq 3 hasPet)$

$petOwner \equiv person \cap \exists hasPet.animal$

$catLike \equiv person \cap \exists likes.cat; catOwner \equiv person \cap \exists hasPet.cat$

$dogLike \equiv person \cap \exists likes.dog; dogOwner \equiv person \cap \exists hasPet.dog$

$dog \cap cat \subseteq \perp$

Chunk 5. Formalisation in DL

$driver \subseteq adult$

$driver \equiv person \cap \exists drives.vehicle$

$lorryDriver \equiv person \cap \exists drives.lorry$

$haulageWorke \equiv \exists worksFor.(haulageCompany \cup \exists partOf.haulageCompany)$

$haulageTruckDriver \equiv person \cap \exists drives.truck \cap$
$\quad \exists worksFor.(\exists partOf.haulageCompany)$

$vanDriver \equiv person \cap \exists drives.van$

$busDriver \equiv person \cap \exists drives.bus$

$whiteVanMan \equiv man \cap \exists drives.(whiteThing \cap van)$

Formalisation in OWL DL. Initial concept taxonomy

OWL Abstract syntax **people+pets.abs.txt**

OWL RDF/XML syntax **people+pets.owl**

Initial concept taxonomy → inferencing log → final concept taxonomy

4_ConceptTaxonomies.doc

Interesting results (I). Automatic classification

And old lady is a person who is elderly and female.
Old ladies must have some animal as pets and all their pets are cats.

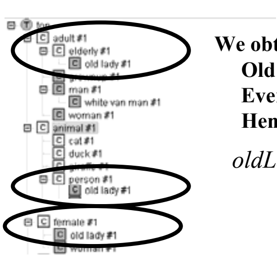
$$elderly \subseteq person \cap adult$$

$$woman \equiv person \cap female \cap adult$$

$$catOwner \equiv person \cap \exists hasPet.cat$$

$$oldLady \equiv person \cap female \cap elderly$$

$$oldLady \subseteq \exists hasPet.animal \cap \forall hasPet.cat$$



We obtain:

- **Old ladies must be women.**
- **Every old lady must have a pet cat**
- **Hence, every old lady must be a cat owner**

$$oldLady \subseteq woman \cap elderly \cap catOwner$$

Interesting results (II). Instance classification

A pet owner is a person who has animal pets
Old ladies must have some animal as pets and all their pets are cats.
Has pet has domain person and range animal
Minnie is a female, elderly, who has a pet called Tom.

$$petOwner \equiv person \cap \exists hasPet.animal$$
$$oldLady \subseteq \exists hasPet.animal \cap \forall hasPet.cat$$
$$hasPet \subseteq (person, animal)$$
$$Minnie \in female \cap elderly$$
$$hasPet(Minnie, Tom)$$

We obtain:
Minnie is a person
Hence, Minnie is an old lady
Hence, Tom is a cat

$$Minnie \in person; Tom \in animal$$
$$Minnie \in petOwner$$
$$Minnie \in oldLady$$
$$Tom \in cat$$

Interesting results (III). Instance classification and redundancy detection

An animal lover is a person who has at least three pets
Walt is a person who has pets called Huey, Louie and Dewey.

$$animalLover \equiv person \cap (\geq 3 hasPet)$$
$$Walt \in person$$
$$hasPet(Walt, Huey)$$
$$hasPet(Walt, Louie)$$
$$hasPet(Walt, Dewey)$$

We obtain:
Walt is an animal lover
Walt is a person is redundant

$$Walt \in animalLover$$

© Oscar Corcho, Mariano Fernánde-López, Asunción Gómez-Pérez

Interesting results (IV). Instance classification

A van is a type of vehicle
A driver must be adult
A driver is a person who drives vehicles
A white van man is a man who drives vans and white things
White van mans must read only tabloids
Q123ABC is a white thing and a van
Mick is a male who reads Daily Mirror and drives Q123ABC

$$van \subseteq vehicle$$
$$driver \subseteq adult$$
$$driver \equiv person \cap \exists drives.vehicle$$
$$whiteVanMan \equiv man \cap \exists drives.(van \cap whiteThing)$$
$$whiteVanMan \subseteq \forall reads.tabloid$$
$$Q123ABC \in whiteThing \cap van$$
$$Mick \in male$$
$$reads(Mick, DailyMirror)$$
$$drives(Mick, Q123ABC)$$

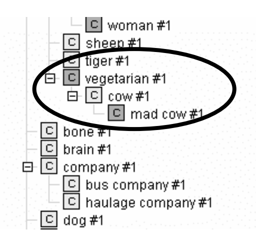
We obtain:
- **Mick is an adult**
- **Mick is a white van man**
- **Daily Mirror is a tabloid**

$$Mick \in adult$$
$$Mick \in whiteVanMan$$
$$DailyMirror \in tabloid$$

Interesting results (V). Consistency checking

Cows are vegetarian.
A vegetarian is an animal that does not eat animals nor parts of animals.
A mad cow is a cow that eats brains that can be part of a sheep

$$cow \subseteq vegetarian$$
$$vegetarian \equiv animal \cap \forall eats.\neg animal \cap$$
$$\forall eats.\neg(\exists partOf.animal))$$
$$madCow \equiv cow \cap \exists eats.(brain \cup \exists partOf.sheep)$$
$$(animal \cup \exists partOf.animal) \cap (plant \cup \exists partOf.plant) \subseteq \bot$$



We obtain:
- **Mad cow is unsatisfiable**

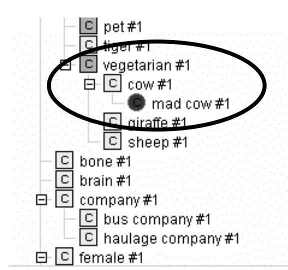


Table of Contents

Ontological Engineering: Semantic Web Portals

Asunción Gómez-Pérez
Mariano Fernández-López
Oscar Corcho
Angel López Cima
{asun, mfernandez, ocorcho, alopez}@fi.upm.es
Grupo de Ontologías
Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

ODESeW

Ontology-based application that automatically generates and manages Knowledge portals for intranets and extranets

1. **Semantic Driven**
2. **Permission-based**
3. **User Oriented**
4. **Interoperate**
5. **Sincronization**

IST-2001-34373

http://www.esperonto.net

ODESeW portal is one of the two main front-end applications of the WebODE workbench

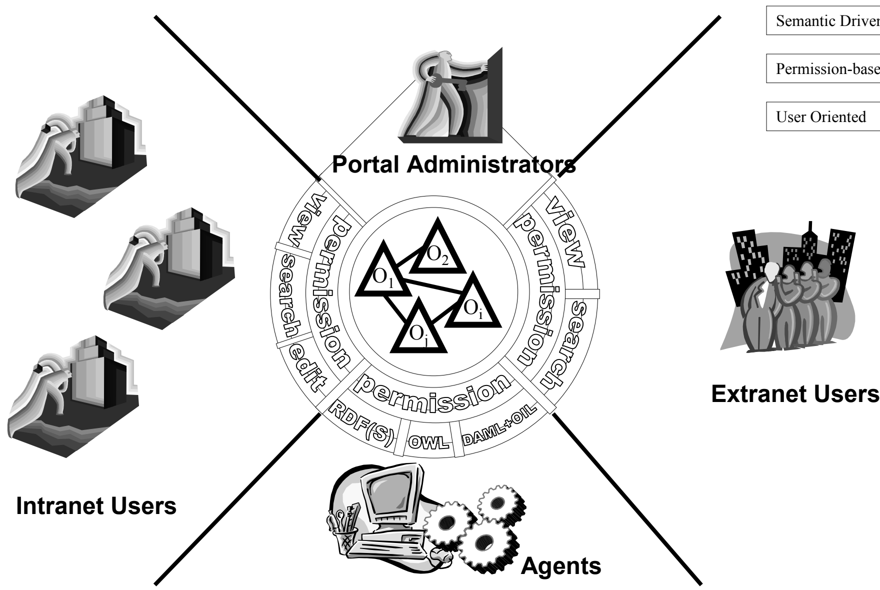
ODESeW is used for building the EsperOnto Web site

Different users = Different perspectives

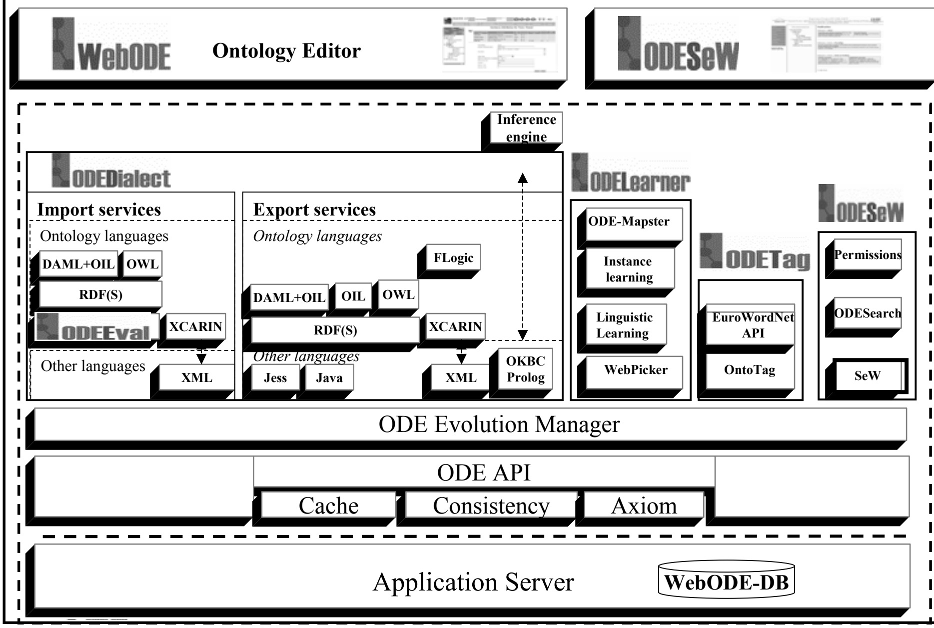
Semantic Driven

Permission-based

User Oriented



Ontology Workbench Status



Portal Generation Process

1. Build the Ontologies

↓

2. Deploy ontologies into ODESeW

↓

3. Create and Update typologies of users and users

↓

4. Define permissions at the concept and instance level

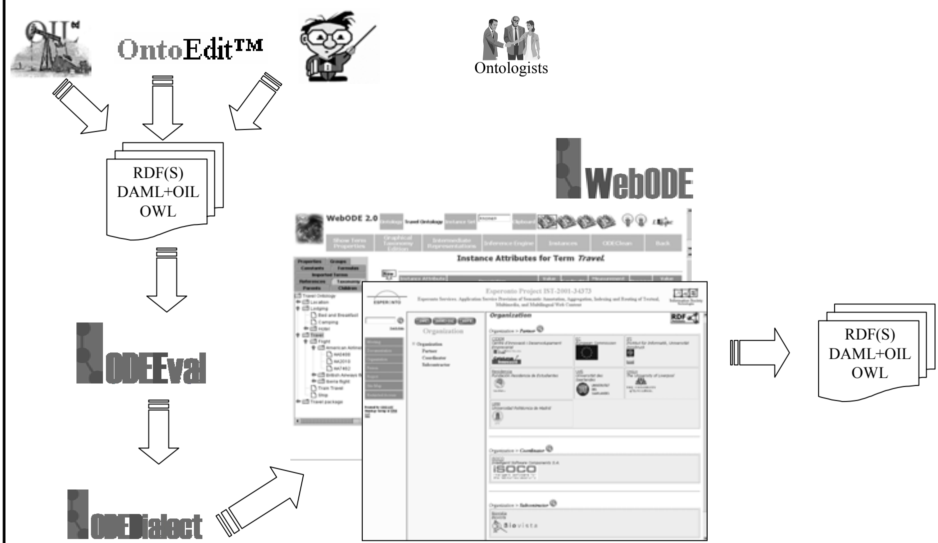
↓

5. Define the attribute visualization ordering

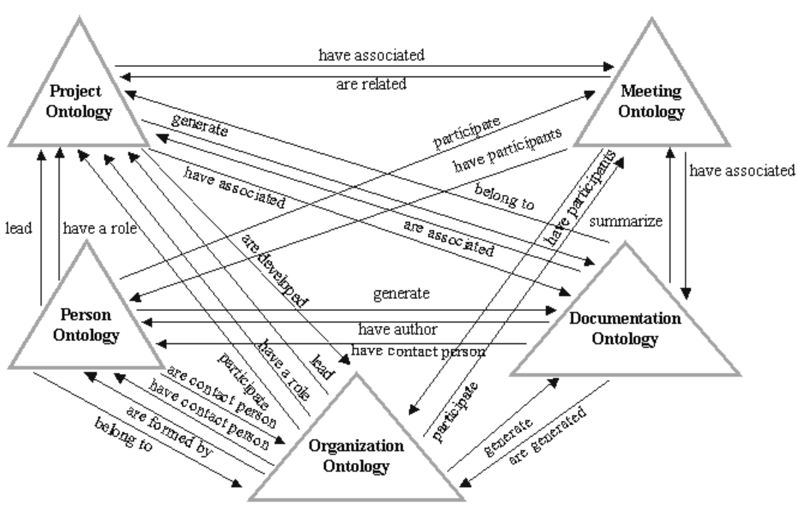
↓

6. Select and compose the attributes for summarization purposes

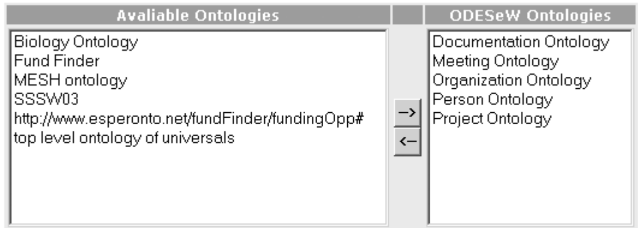
Ontology Modelling



Ontology Modelling

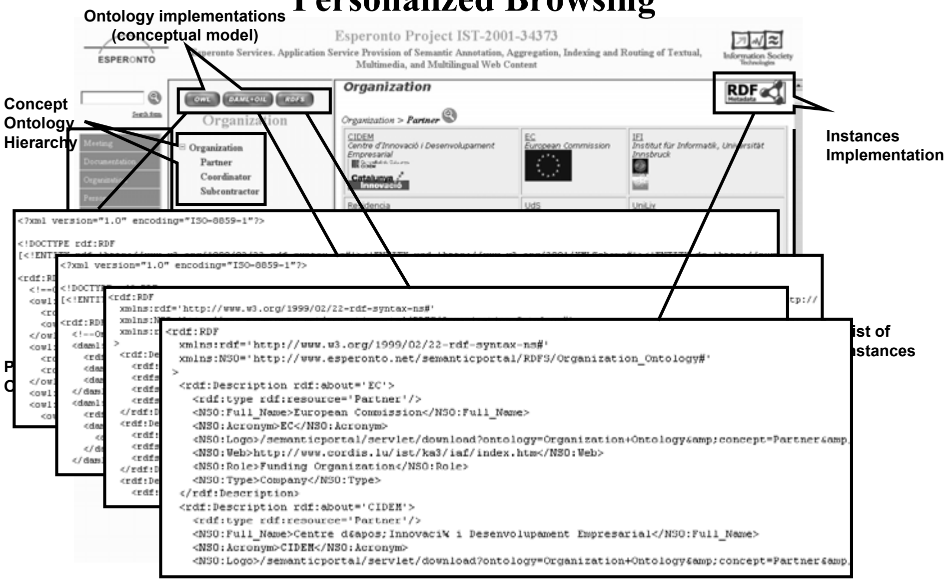


Deploy ontologies into ODESeW



Select the ontologies that will be made available on the portal for browsing

Personalized Browsing



Semantics driven visualization (guest)

Deliverable List

WP1: Ontologies
- D1.1: State of the art in ontologies from the SW perspective
- D1.2: Kernel Ontology Specification. Knowledge architecture *(Restricted)*
- D1.3: Ontology Workbench Specification *(Restricted)*
- D1.4: Ontology Alignment Solution *(Restricted)*

WP2: Window on Semantic Web languages
- D2.1: State of the art on Semantic Web languages
- D2.2: Report on SW languages evolution

WP3: Annotation services
- D3.1: State of the art on annotation tools and services
- D3.2: Methodology for the development of wrappers and annotation tools *(Restricted)*
- D3.3: Annotation services for static resources
- D3.4: Annotation services for dynamic resources
- D3.5: Annotation services for multimedia content
- D3.6: Annotation services for web services

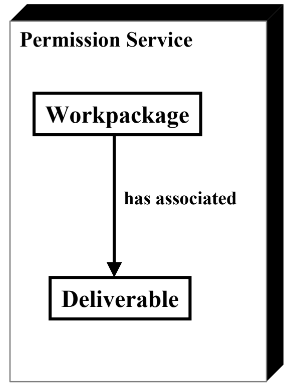
WP4: Semantic indexation and routing
- D4.1: State of the art on indexation, routing techniques and negotiation techniques
- D4.2: Semantic Index Solution *(Restricted)*
- D4.3: Routing Solution *(Restricted)*

WP5: Multilinguality
- D5.1: State of the art on multilinguality for ontologies, annotation services and user interfaces
- D5.2: Multilinguality and ontologies *(Restricted)*
- D5.3: Multilinguism and annotation services *(Restricted)*
- D5.4: Multilingual user interface *(Restricted)*

WP6: User interface and visualisation services
- D6.1: State of the art on visualisation technologies feasible for the Semantic Web
- D6.2: Ontology visualisation core services
- D6.3: Semantic Web content visualisation services
- D6.4: Semantic Index and Routing Monitor service

WP7: Definition and integration
- D7.1: System specification *(Restricted)*
- D7.2: Cooperation protocol definition *(Restricted)*
- D7.3: Application development guidelines *(Restricted)*
- D7.4: Integration test plan *(Restricted)*

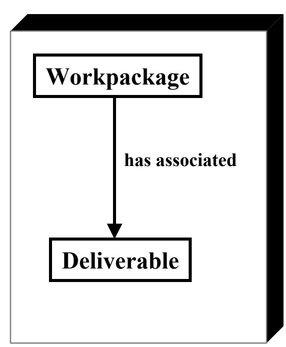
WP8: Test case 1. Fund finder for R&D
- D8.1: Test case system specification *(Restricted)*
- D8.2: Test case ontology specification *(Restricted)*



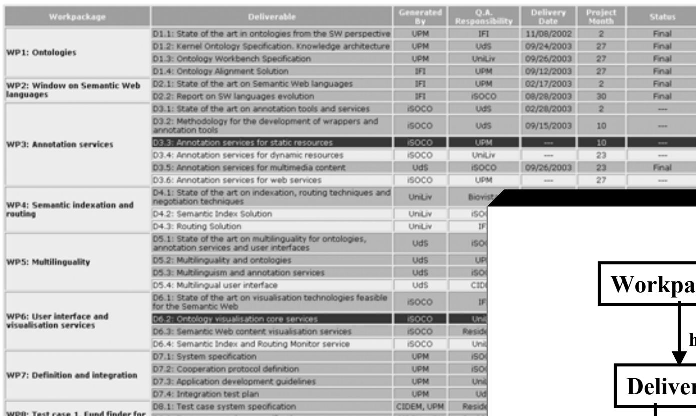
Semantics driven visualization (intranet)

Deliverable List

Workpackage	Deliverable	upload date	PDF
WP1: Ontologies	D1.1: State of the art in ontologies from the SW perspective	11/08/2002	
	D1.2: Kernel Ontology Specification. Knowledge architecture	09/24/2003	
	D1.3: Ontology Workbench Specification	09/26/2003	
	D1.4: Ontology Alignment Solution	09/12/2003	
WP2: Window on Semantic Web languages	D2.1: State of the art on Semantic Web languages	02/17/2003	
	D2.2: Report on SW languages evolution	08/28/2003	
WP3: Annotation services	D3.1: State of the art on annotation tools and services	02/28/2003	
	D3.2: Methodology for the development of wrappers and annotation tools	09/15/2003	
	D3.5: Annotation services for multimedia content	09/26/2003	
WP4: Semantic indexation and routing	D4.1: State of the art on indexation, routing techniques and negotiation techniques	02/11/2002	
WP5: Multilinguality	D5.1: State of the art on multilinguality for ontologies, annotation services and user interfaces	02/28/2003	
	D5.2: Multilinguality and ontologies	09/26/2003	
	D5.3: Multilinguism and annotation services	09/26/2003	
WP6: User interface and visualisation services	D6.1: State of the art on visualisation technologies feasible for the Semantic Web	03/07/2003	
	D6.3: Semantic Web content visualisation services	09/25/2003	
WP7: Definition and integration	D7.1: System specification	07/29/2003	
	D7.2: Cooperation protocol definition	01/16/2003	
	D7.3: Application development guidelines	04/03/2003	
	D7.4: Integration test plan	09/18/2003	
WP8: Test case 1. Fund finder for R&D	D8.1: Test case system specification	04/07/2003	
	D8.2: Test case ontology specification	08/29/2003	
	D8.3: Test case application development	09/25/2003	
WP9: Test case 2. Cultural tour	D9.1: Test case system specification	07/21/2003	
	D9.2: Test case ontology specification	09/26/2003	
WP10: Test case 3. Scientific discovery	D10.2: Test case ontology specification	07/10/2003	
WP11: Project management	D11.1: Quality Plan and Development Plan	02/12/2003	



Semantics driven visualization



Status of the Deliverables

Workpackage	Deliverable	Generated By	Q.A. Responsibility	Delivery Date	Project Month	Status
WP1: Ontologies	D1.1: State of the art in ontologies from the SW perspective	UPM	IFI	11/08/2002	2	Final
	D1.2: Kernel Ontology Specification. Knowledge architecture	UPM	UdS	09/24/2003	27	Final
	D1.3: Ontology Workbench Specification	UPM	UniLiv	09/26/2003	27	Final
	D1.4: Ontology Alignment Solution	IFI	UPM	09/12/2003	27	Final
WP2: Window on Semantic Web languages	D2.1: State of the art on Semantic Web languages	IFI	UPM	02/17/2003	2	Final
	D2.2: Report on SW languages evolution	IFI	iSOCO	08/28/2003	30	Final
WP3: Annotation services	D3.1: State of the art on annotation tools and services	iSOCO	UdS	02/28/2003	2	---
	D3.2: Methodology for the development of wrappers and annotation tools	iSOCO	UdS	09/15/2003	10	---
	D3.3: Annotation services for static resources	iSOCO	UPM	---	10	---
	D3.4: Annotation services for dynamic resources	iSOCO	UniLiv	---	23	---
	D3.5: Annotation services for multimedia content	UdS	iSOCO	09/26/2003	23	Final
	D3.6: Annotation services for web services	iSOCO	UPM	---	27	---
WP4: Semantic indexation and routing	D4.1: State of the art on indexation, routing techniques and negotiation techniques	UniLiv	Biovist			
	D4.2: Semantic Index Solution	UniLiv	iSO			
	D4.3: Routing Solution	UniLiv	IF			
WP5: Multilinguality	D5.1: State of the art on multilinguality for ontologies, annotation services and user interfaces	UdS	iSO			
	D5.2: Multilinguality and ontologies	UdS	UP			
	D5.3: Multilinguism and annotation services	UdS	iSO			
	D5.4: Multilingual user interface	UdS	CID			
WP6: User interface and visualisation services	D6.1: State of the art on visualisation technologies feasible for the Semantic Web	iSOCO	IF			
	D6.2: Ontology visualisation core services	iSOCO	Uni			
	D6.3: Semantic Web content visualisation services	iSOCO	Reside			
	D6.4: Semantic Index and Routing Monitor service	iSOCO	Uni			
WP7: Definition and integration	D7.1: System specification	UPM	iSO			
	D7.2: Cooperation protocol definition	UPM	iSO			
	D7.3: Application development guidelines	UPM	Uni			
	D7.4: Integration test plan	UPM	Ud			
WP8: Test case 1. Fund finder for	D8.1: Test case system specification	CIDEM, UPM	Reside