

Acuerdo de Bibliotecas Universitarias de Córdoba

Seminario

27 y 28 de septiembre de 2012

Web semántica, Web 3.0 y entornos Cloud Computing, nuevos horizontes para bibliotecarios, documentalistas y archivistas



Mela Bosch

melabosch@hotmail.com

Acuerdo de Bibliotecas Universitarias de Córdoba

Seminario

27 y 28 de septiembre de 2012



Séptimo encuentro: Los diferentes niveles en la representación del conocimiento en la Web Semántica

Qué es la representación del conocimiento en el medio digital. Qué es la representación del conocimiento en la Web Semántica. Los niveles de representación. Construcción de Ontologías. Ejemplos. Tareas y tendencias.

Cierre: Puesta en común. ¿qué lugar pueden ocupar los especialistas en información documental en la representación del conocimiento en el medio digital?

Qué es la representación del conocimiento en el medio digital

- Es un sustituto de la cosa en sí, utilizado para permitir que una entidad pueda determinar las consecuencias a partir de una conceptualización aún antes de actuar
- Es un conjunto de expresiones que indican en qué términos se representa un dominio
- Es una teoría fragmentaria de razonamiento inteligente indicado en base a tres componentes: **a.** Las representaciones de las bases fundamentales del razonamiento inteligente **b.** Las confirmación de esas representaciones por medio de una simbolización; **c.** Un conjunto de inferencias que es posible realizar.
- Es un medio para la expresión humana, un lenguaje para decir cosas sobre un mundo
- Es un medio para una pragmática y eficiente computabilidad

Ref: Davis, R., Shrobe, H., and Szolovits, P. What is a Knowledge Representation? *AI Magazine*, 14(1):17-33, 1993.
<http://groups.csail.mit.edu/medq/ftp/psz/k-rep.html>



Qué es la representación del conocimiento en la Web Semántica

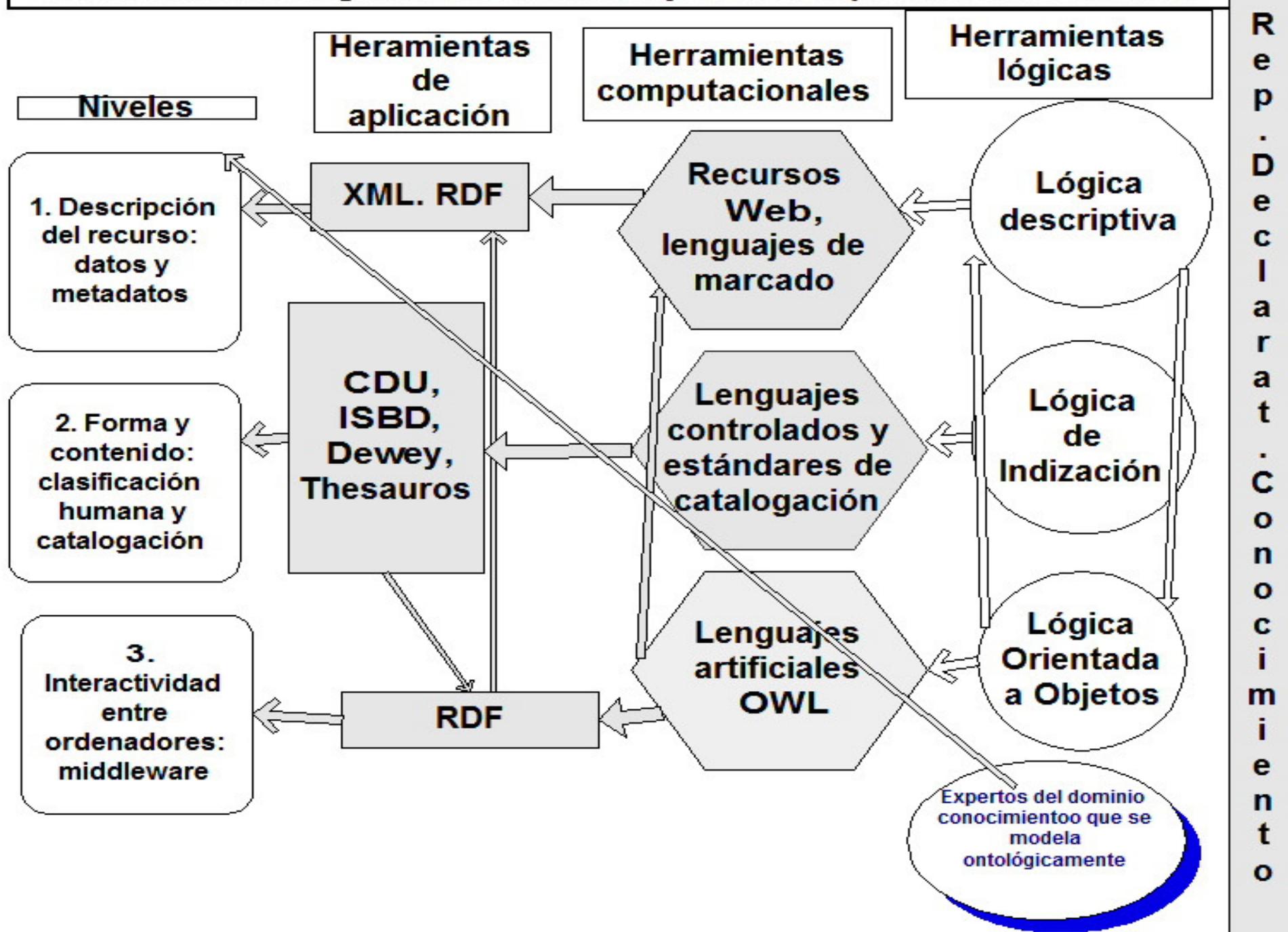
- Es la descripción de la forma y el contenido de recursos web
- Esta descripción está expresada por medio de una estructura de metadata en un lenguaje de marcado: RDF, Resource Description Format
- En un modo comprensibles por las máquinas



No hay una sola estructura de representación de conocimiento.

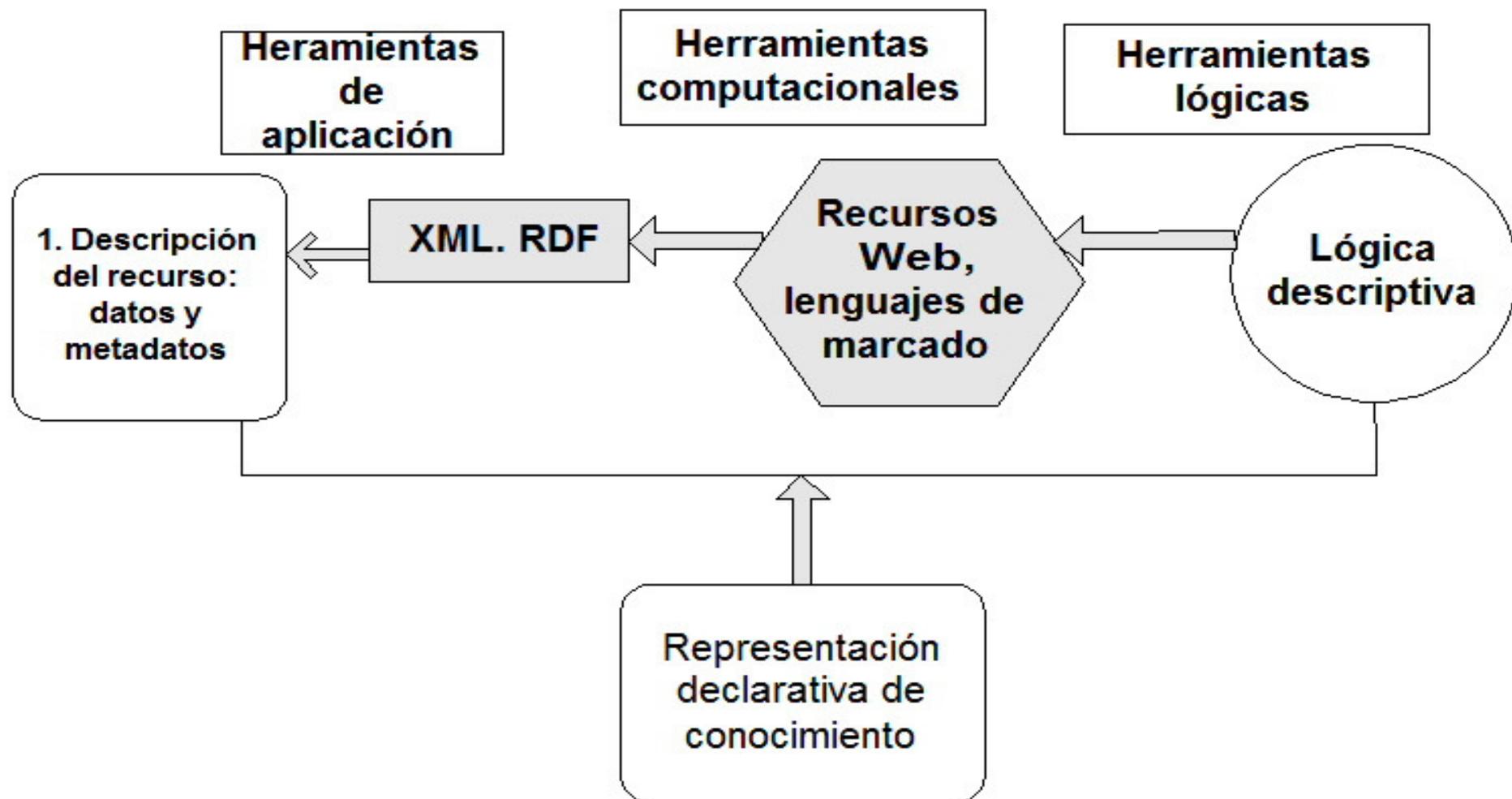
Los formalismos de representación de la Web Semántica consisten en diferentes tipos de lógicas con diferentes lenguajes y formatos

Web semántica: lógicas, herramientas y niveles represen. conocimiento



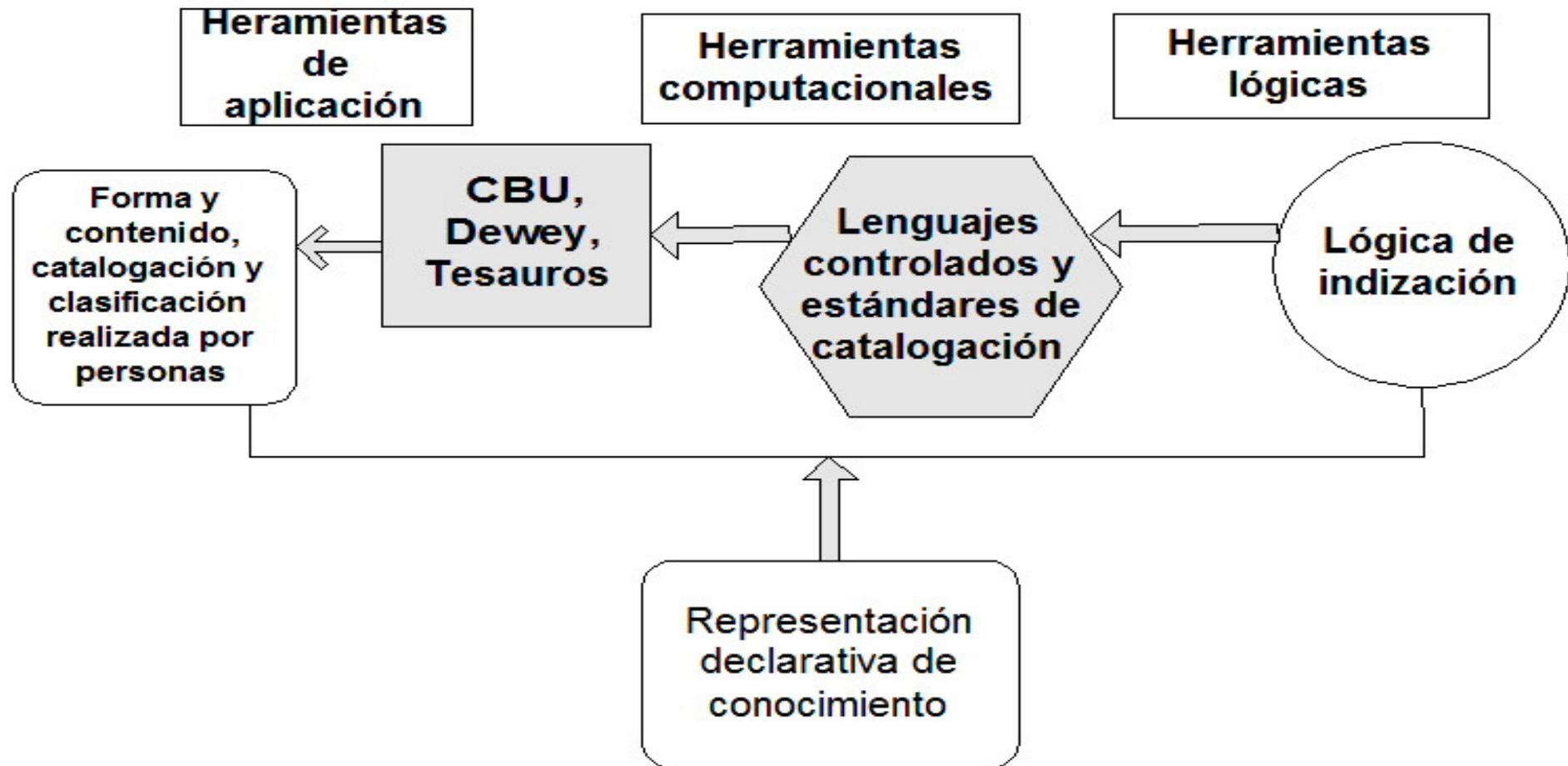
Primer nivel de representación

Web semántica: Nivel 1 Lógica descriptiva



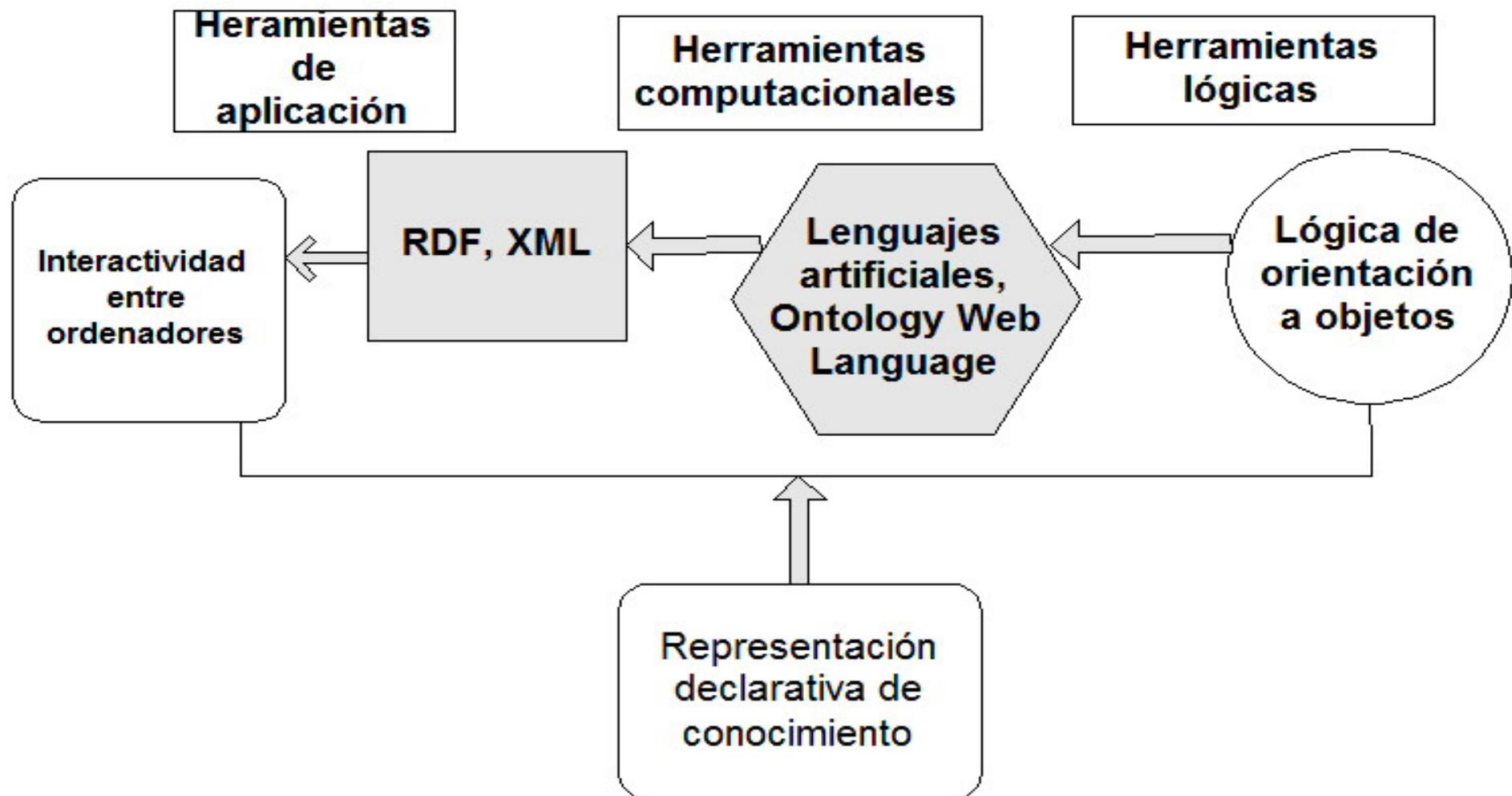
Segundo nivel de representación

Web semántica: Nivel 2 Lógica del tratamiento documental



Tercer nivel de representación

Web semántica: Nivel 3 Lógica de la Orientación a Objetos





Metodologías para los sistemas inteligentes

Procedimental → el conocimiento está integrado en el programa

Ventajas: *gran especificidad: algoritmos adaptados a cada caso*

Desventajas: *poca versatilidad dificultad para modificar*

Declarativa → La representación del conocimiento es independiente del proceso computacional

Ventajas : *flexible y con fuerte base lógica*

Desventajas: *alto nivel de abstracción, dificultad para mantener una lógica consistente*



Web Semántica: Representación declarativa utilizando metadata



La metadata se expresa en:

Lenguaje de marcado: Markup language:

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
</rdf:RDF>
```



Lógica descriptiva también llamada lógica terminológica

- lenguaje controlado cuya sintaxis es independiente de los procedimientos del sistema y la terminología está adaptada al dominio de aplicación
- Semántica bien definida muy expresiva



Lenguaje de marcado: tiene la especificidad procedimental con la capacidad de abstracción de la forma declarativa

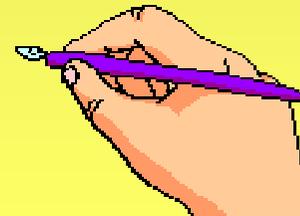


Segundo nivel de representación,
otro tipo de lógica



La lógica de la indización para catalogación y
clasificación

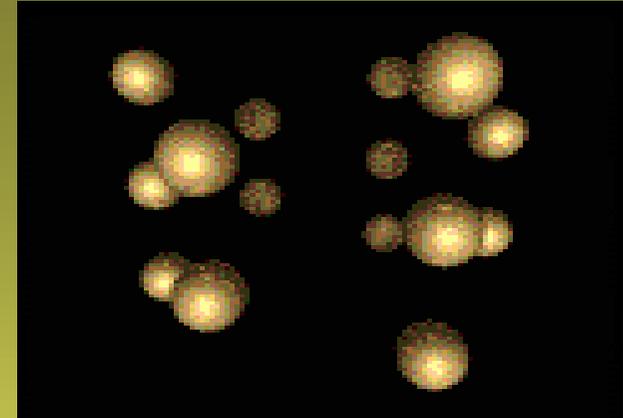
**Pero no es lo mismo indizar un
objeto que indizar la referencia a ese
objeto**





Web Semántica indiza recursos web

Es decir, objetos, no materiales,
sino objetos digitales.



Esos Objetos son descritos a través de una ontología

Los items de referencia a objetos digitales : autor, fecha etc.



**Son similares a los que se usan en los sistemas de
clasificación y catalogación documental**

**Otros aspectos de los objetos digitales como
atributos, comportamiento, cardinalidad, son
expresados con otra lógica:**

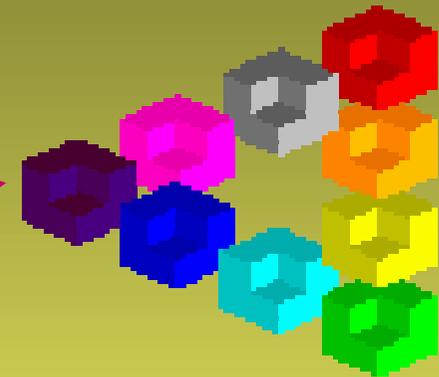


Tercer nivel de representación: Paradigma de orientación a objetos

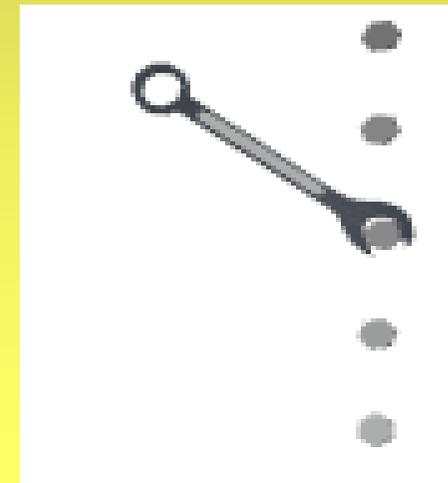
Orientación a objetos

Programa computación:
conjunto de unidades

individuales, los *objetos*, que administran por sí mismos su estado y sus operaciones



Se opone al tradicional enfoque en el que el programa es una colección de *view in funciones*, o una *lista de instrucciones* que se dan a la computadora.





Tercer nivel de representación del conocimiento, Lógica orientada objetos que permite:

- Hacer explícitos los presupuestos de un dominio
- Separar el dominio del conocimiento del conocimiento operacional
- Usan términos para representar conceptos que son una abstracción de las principales propiedades de los objetos



Ventajas → Intuitiva construcción directamente desde el mundo real

Desventajas → Dificultad para construir jerarquías de objetos,
→ trabajo manual



Orientación a Objetos y construcción de ontologías

La misma lógica, pero:

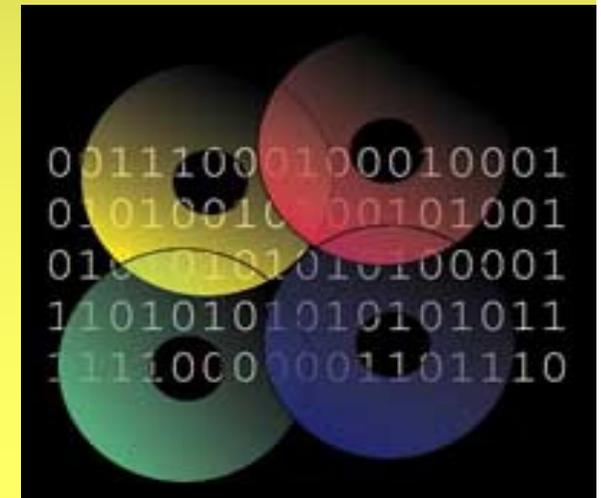
Una ontología

- Refleja la estructura del *mundo*
- es un estructura de *conceptos*
- No describe aspectos *físicos*

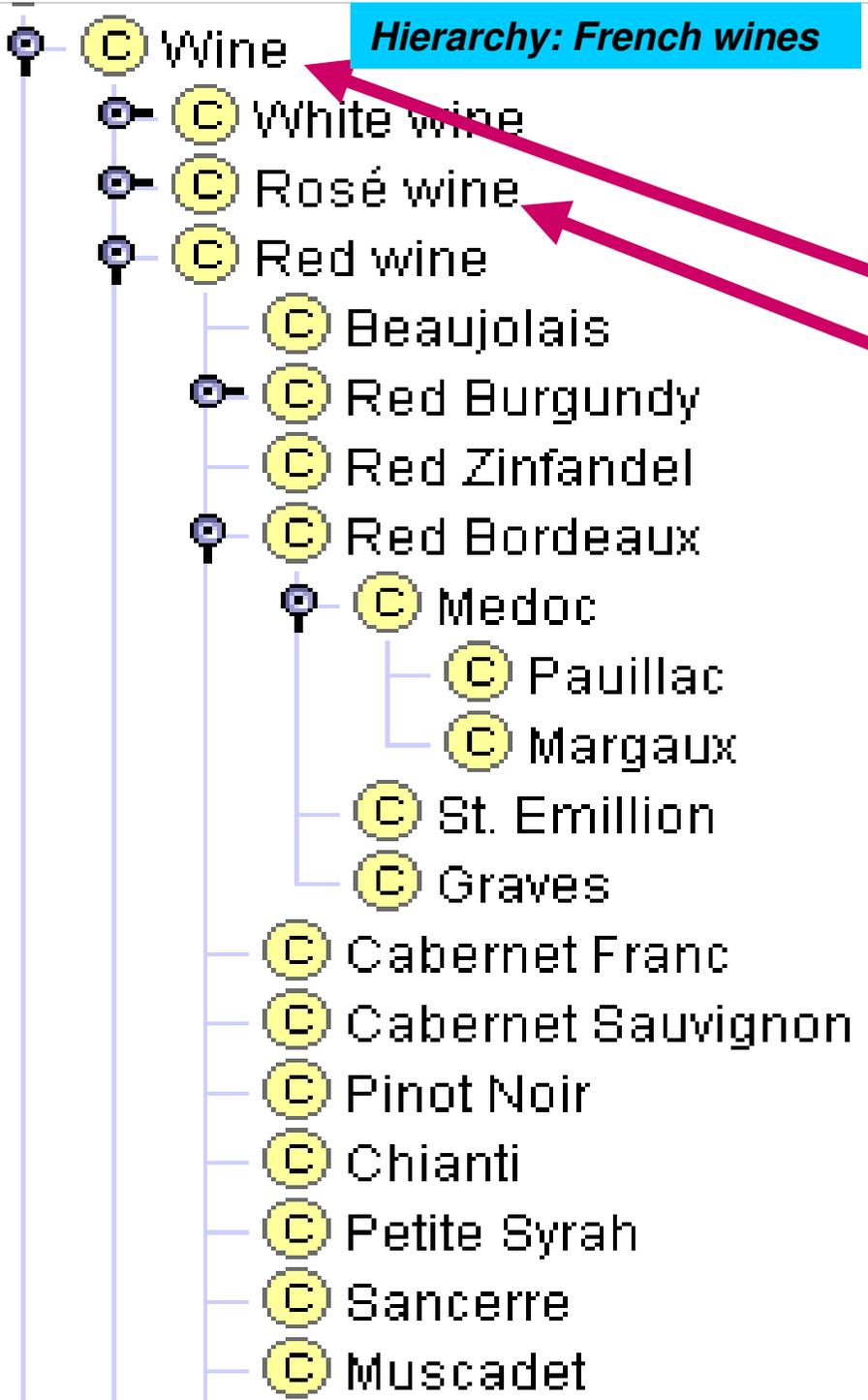


Estructura de clases en Orientación a Objetos

- Refleja la estructura de los *datos*
- Se ocupa de los *comportamientos*
- describes la *representación física de los datos* (long int, char, etc.)



(Ref: <http://protege.stanford.edu>)



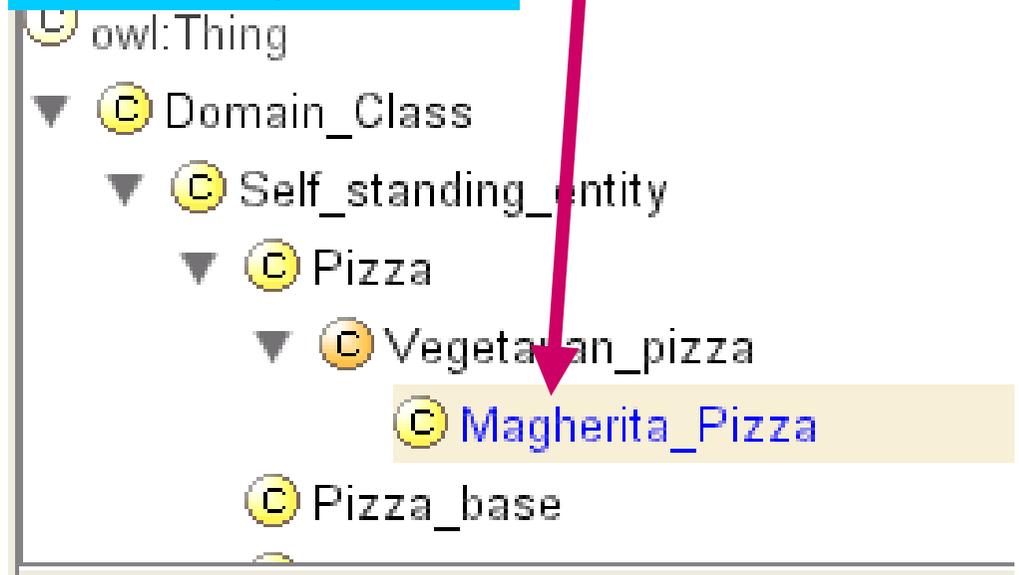
Hierarchy: French wines

Ejemplos: Protégé: a graphical ontology- development tool, open-source and freely available: <http://protege.stanford.edu>

Class structure usually constitute a taxonomic hierarchy:

Class
Subclass

Hierarchy: Pizza



Magherita_Pizza

Template Slots



Name	Type	Cardinality	Other Facets
S body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}
S color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}
S flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}
S grape	Instance	multiple	classes={Wine grape}
S maker I	Instance	single	classes={Winery}
S name	String	single	
S sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}

Slots in a class definition describe attributes of instances of the class and relations to other instances: Each wine will have color, sugar content, producer, etc.

Types of properties

- “intrinsic” properties: flavor and color of wine
- “extrinsic” properties: name and price of wine
- parts: ingredients in a dish
- relations to other objects: producer of wine (winery)

Terminology

Class=concept

Instance= object

Slot= property

Facet=values

Slot cardinality = the number of values a slot has (common facet)

Slot value type = the type of values a slot has (common facet): string, num, boolean

Minimum and maximum value = a range of values for a numeric slot (common facet)

Default value = the value a slot has unless explicitly specified otherwise (common facet)

Chateau Morgon Beaujolais (Beaujolais)

Name: Chateau Morgon Beaujolais

Area: Beaujolais region

Body: LIGHT

Color: RED

Maker: Chateau Morgon

Flavor: DELICATE

Sugar: DRY

Grape: Gamay grape

Tannin Level: LOW

Superclass
Wine



Subclass
French wine

A subclass **inherits** all the slots from the superclass, but with a list of own allowed values

Class instance creation

Examples from: Ref: http://www.ncess.ac.uk/insight/tutorials/datagrids/data_sh/ontologies/http://protege.stanford.edu/publications/ontology_development/ontology101.html



Common problems: Is a Margherita Pizza a Vegetarian Pizza?

Errors in understanding common logical constructs

A class can have more than one superclass
A subclass inherits slots and facet restrictions from all the parents
Different systems resolve conflicts differently

Magherita_Pizza

RDFS:COMMENT:
A magherita pizza kind of pizza and has has, amongst other things, Mozzarella and Tomato toppings

Asserted Inferred

ASSERTED CONDITIONS:

NECESSARY

- ⊆ Pizza
- ⊆ ∃ has_topping Mozzarella_topping
- ⊆ ∃ has_topping Tomato_topping

Name

Vegetarian_pizza

rdfs:comment
A vegetarian pizza is ANY pizza that has no meat topping and no fish topping.

Asserted Inferred

Asserted Conditions

NECESSARY &

- ⊆ Pizza
- ⊆ ¬(∃ has_topping Fish_topping)
- ⊆ ¬(∃ has_topping Meat_topping)

Ref: <http://www.co-ode.org>

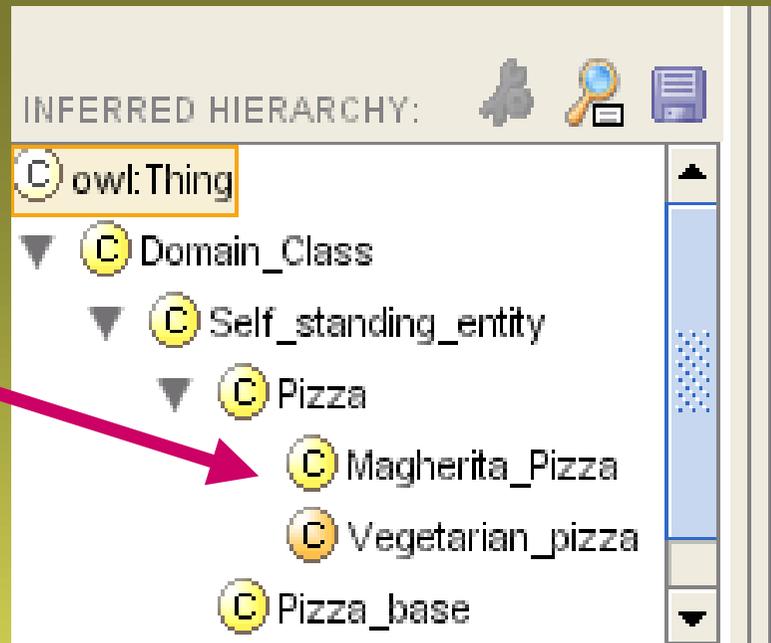


Common problems: Is a Margherita Pizza a Vegetarian Pizza?

Different subclasses

Closure Axiom: only

Correct hierarchy



Class Description

Paraphrase/descriptive syntax

Class: Magherita_Pizza

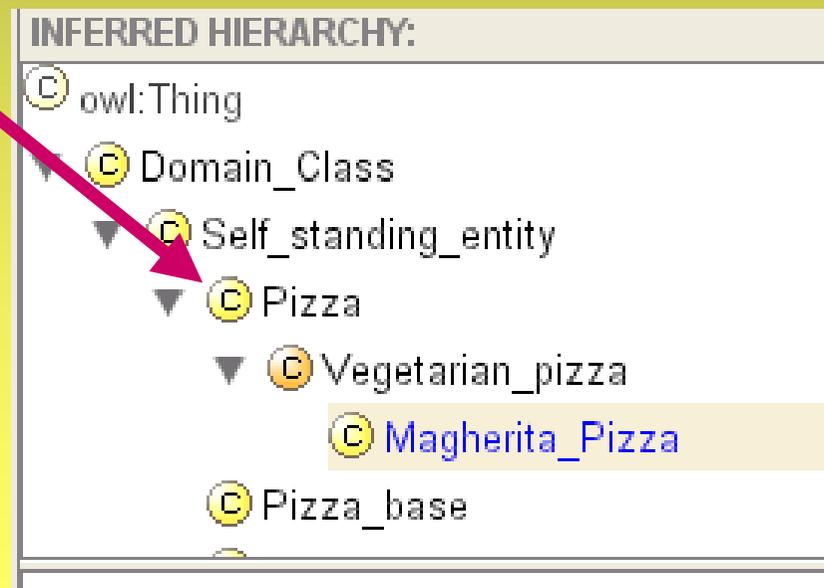
NECESSARILY

Pizza

has_topping some Mozzarella_topping

has_topping some Tomato_topping

has_topping only (Mozzarella_topping OR Tomato_topping)



Ref: <http://www.co-ode.org>



Conclusion

*“Every ontology is a treaty
– a social agreement –
among people with some
common motive in sharing”*

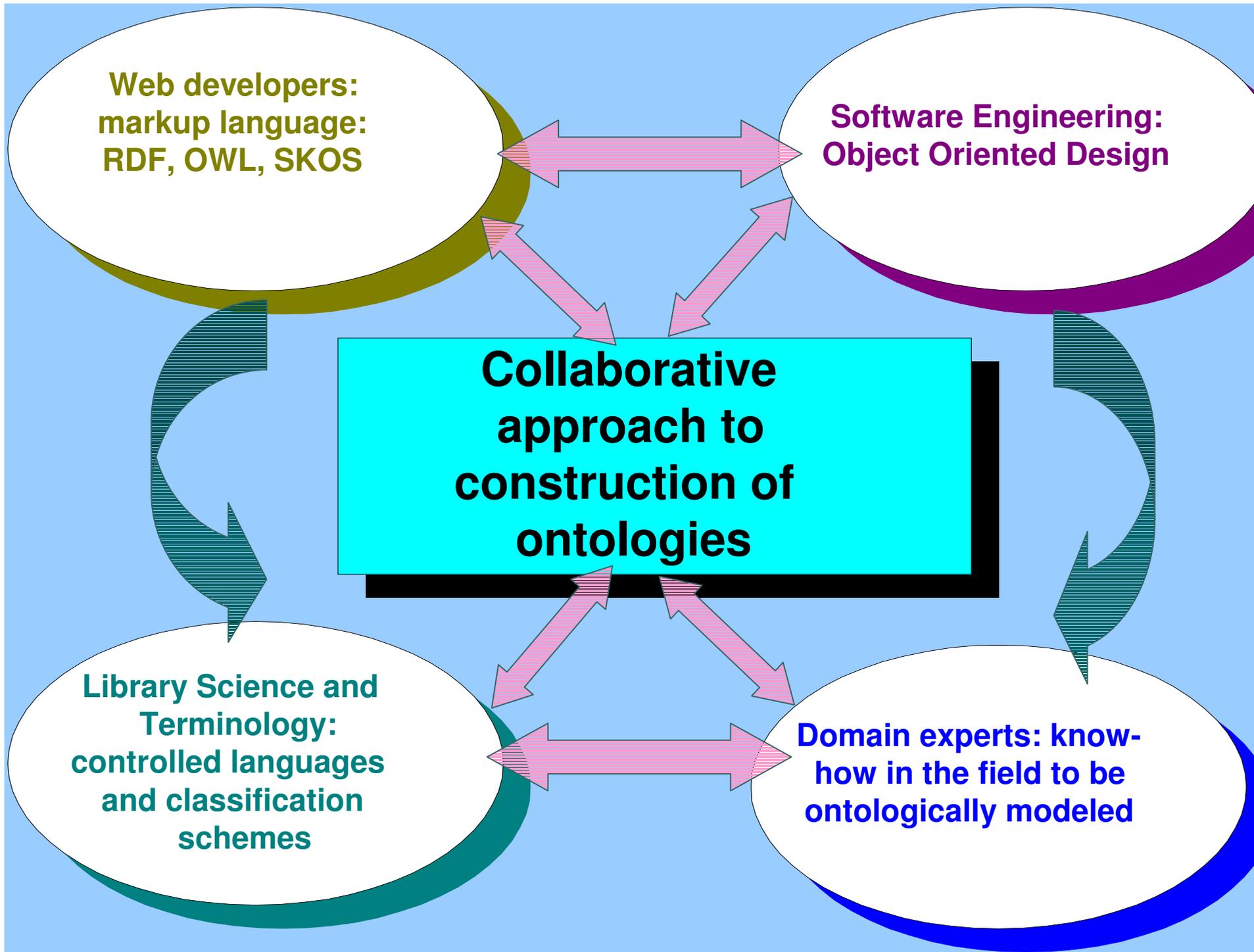
Gruber dijo:



Ref: http://www.sigsemis.org/newsletter/october2004/tom_gruber_interview_sigsemis

Enfoque colaborativo para la construcción de ontologías

- La comunicación entre los colaboradores de diferentes disciplinas es difícil
- La construcción de ontologías, el mantenimiento y reuso son actividades que toman tiempo y el análisis de costo es complejo





Conclusion

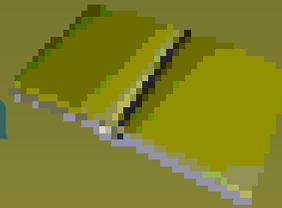
Tareas y tendencias



- Metodologías para creación y manejo de información compartida: Modelado de datos semánticamente heterogeneos
- Sistemas de apoyo para comunidades para creación de aplicaciones en forma colaborativa: Groupware tools for supporting collaborative ontology design, Semantic Wikis, semantic blogging.
- Estudios de caso y experiencias sobre en aplicaciones colaborativas.
- Estudios de estimación de costos de la ingeniería de ontologías en forma colaborativa



Bibliografía



Ejemplos de ontologías
ProtegeOntologiesLibrary
<http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary>

- Hodgson, Ralph; Keller, Paul. Collaborative Ontology-Based Systems. Innovator Perspectives and Demonstrations of New Open Standards and Technologies in Support of Ontology Engineered Solutions. TopQuadrant and NASA Ames. Collaborative Expedition Workshop #38. February 22, 2005 at NSF Semantic Conflict, Mapping, and Enablement: Making Commitments Together. [http://www.topquadrant.com/documents/talks/TQ%20Ontology-Based%20Collaborative%20Environments%20\(v4\).pdf](http://www.topquadrant.com/documents/talks/TQ%20Ontology-Based%20Collaborative%20Environments%20(v4).pdf)
- Díaz, Alicia, Baldo, G. CO-Protégé: A Groupware Tool for Supporting Collaborative Ontology Design with Divergence. Lifa, Fac. Informática- UNLP, La Plata, Argentina- Loria, Campus Scientifique, Vandœuvre-lès-Nancy cedex, France. http://protege.stanford.edu/conference/2005/slides/6.2_A.Diaz_Co-Protege_slices_and_flyer.pdf; <http://protege.stanford.edu/conference/2005/submissions/abstracts/accepted-abstract-diaz.pdf>
- Gamper, Johann, Nejd, Wolfgang; Wolpers, Martin. Combining Ontologies and Terminologies in Information Systems. European Academy Bolzano/Bozen, Scientific Area "Language and Law" Bozen, Italy - Institut für Rechnergestützte Wissensverarbeitung University of Hannover, Germany. <http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/1999/tke99/>
- W3C Working Draft 07 March 2002, Requirements for a Web Ontology Language. Latest version: <http://www.w3.org/TR/webont-req/>
- Institut für Informatik, Freie Universität Berlin. Ontology Engineering Cost Estimation with ONTOCOM. <http://ontocom.ag-nbi.de//index.html>