

Curso: “*Tecnología Web*”

Profesores:

Jose Emilio Labra Gayo (Universidad de Oviedo, España)
Raúl Monge (UTFSM, Chile)

***** Curso sobre Tecnología Web ***** Versión 2005

Contenido

1.- Tecnologías XML

Definición y validación: DTDs, Espacios de nombres, XML Schema
Transformación y consulta de XML: XSLT, Xpath, Xquery
Programación XML: SAX, DOM

2.-Servicios Web

SOAP, WSDL, UDDI

3.-Web Semántica

Fundamentos, RDF, OWL

4.-Nuevas aplicaciones

***** Curso sobre Tecnología Web ***** Versión 2005

Reflexiones sobre la WWW

Web actual = mayor almacén de información recopilado por personas humanas

Características:

Grandes cantidades de información sobre cualquier asunto

Acceso *casi instantáneo* desde cualquier lugar con conexión a Internet

Sistema no centralizado \Rightarrow Cualquier persona puede añadir más información

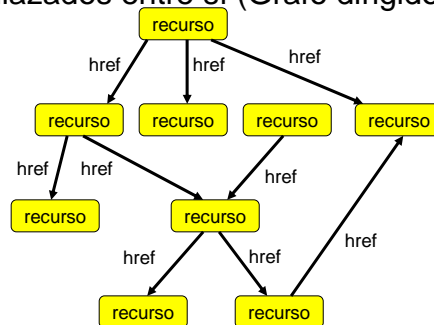
Plataforma Multimedia (Texto, Imágenes, Vídeo, etc.)

Identificación de recursos unificada (URIs)

Reflexiones sobre la Web

Actualmente, estamos en la **Web Sintáctica**

Un conjunto de recursos enlazados entre sí (Grafo dirigido)



Los ordenadores realizan la presentación visual (tarea fácil) y las personas navegan e interpretan el contenido (tarea difícil)

¿Sería posible que los ordenadores hiciesen algo más?

Tareas difíciles en la Web sintáctica

Buscar fotos del profesor de este curso (Labra)



Curso sobre Tecnología Web

Versión 2005

Tareas difíciles en la Web Sintáctica

Buscar información sobre la Universidad de **Beihang** en China...



Curso sobre Tecnología Web

Versión 2005

Tareas difíciles en la Web Sintáctica

Otras tareas:

- Buscar si hubo alguien que se autoproclamó rey de los Estados Unidos
- Buscar un ave que utilice el oído para orientarse y que no sea un murciélago

Tareas difíciles en la Web Sintáctica

Búsquedas complejas

Localizar información en almacenes de datos

Búsqueda de viajes

Comparar Precios de productos

Encontrar y utilizar “servicios web”

Delegar tareas complejas a agentes de la Web

Organizar un viaje en algún lugar con playa no demasiado caro en el que hablen chino

Buscar y comparar noticias que hablen de las últimas elecciones

Encargar una comida en el restaurante que no tenga demasiadas calorías

El problema de la Web Sintáctica

Tipos de Pizzas - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security

Bookmarks Location

Pizzas del Restaurante Al Capone

Tipos de Pizzas

Pizza	Ingredientes	Precio
Barbacoa	Salsa barbacoa, Mozzarella, Pollo, Bacon, Ternera	8 €
Hawaiana	Tomate, Mozzarella, Jamón, Piña, Queso	7 €
4 quesos	Tomate, Mezcla de 4 quesos	7 €
Margarita	Tomate, Jamón, Queso	6 €

```
<html><head><title>Pizzeria Al Capone</title></head>
<body bgcolor="blue" text="yellow">
<h1>Pizzas del Restaurante Al Capone</h1>
<table>
<caption>Tipos de Pizzas</caption>
<tr>
<td>Barbacoa</td><td>Salsa barbacoa, mozzarella, Pollo
Bacon, Ternera </td><td>8&euro;</td>
</tr>
...
</body>
</html>
```

Las marcas de HTML incluyen información sobre la visualización (tipo de letra, color, etc.)

El significado es accesible a las personas pero no es accesible a los ordenadores...

El problema de la Web Sintáctica

La información que vemos nosotros...

Pizzas del Restaurante Al Capone
Tipos de Pizzas
Barbacoa
Salsa barbacoa, mozzarella, Pollo, Bacon, Ternera 8€
...

Lo que ve la máquina...

```
<html><head><title>Pizzeria Al Capone</title></head>
<body>
<h1>Pizzas del Restaurante Al Capone</h1>
<table>
<caption>Tipos de Pizzas</caption>
<tr>
<td>Barbacoa</td><td>Salsa barbacoa, mozzarella, Pollo
Bacon, Ternera </td><td>8&euro;</td>
</tr>
...
</body>
</html>
```

¿XML como posible solución?

Possible solution: XML. Etiquetas con significado propio

```
<título>Pizzas del Restaurante Al Capone</título>
<pizza>
<nombre>Barbacoa</nombre><ingredientes>Salsa barbacoa,
mozzarella, Pollo, Bacon, Ternera
</ingredientes><precio>8€ </precio></pizza>
...
```

La máquina vería...

[illegible]

¿XML como posible solución?

Problema: Si se cambia de contexto, ¿se definen nuevas etiquetas?

<ave><nombreComún>Lechuza Común</nombreComún>
<nombre>Tyto alba</nombre>
<clasificación>Orden Estrigiformes, familia Titónidos.
</clasificación>
<características>33-39 cm. 300-380 g. </características>
...

Pero la aplicación que trabaja con Pizzas no entendería las nuevas etiquetas

[illegible]

Problema: Las etiquetas no tienen un significado compartido

¿Cómo compartir etiquetas?

Posibilidad: **Acuerdo global** sobre el significado de las anotaciones

Ejemplo: Dublin Core

Especifica etiquetas globales de anotación de recursos. Por ejemplo:
dc:creator

Problemas:

Inflexibilidad

Limita el número de cosas que pueden expresarse

¿Cómo compartir etiquetas?

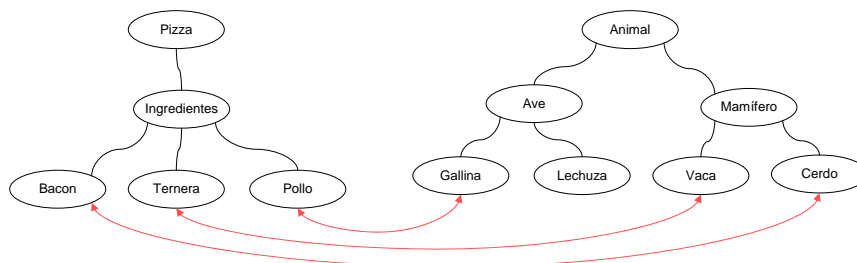
Otra posibilidad: Utilización de **Ontologías**

Definen el significado de las anotaciones

Los términos nuevos pueden formarse a partir de otros anteriores

El significado se define formalmente

Pueden especificarse relaciones entre términos de varias ontologías...



Web Semántica

La **Web semántica** (Tim Berners-Lee) es una **visión**:

*“**disponer datos** en la Web **definidos y enlazados** de forma que puedan ser **utilizados por las máquinas** no solamente para visualizarlos sino también para:*

***automatizar** tareas,*

***integrar** y*

***reutilizar** datos entre aplicaciones”*

Web Semántica

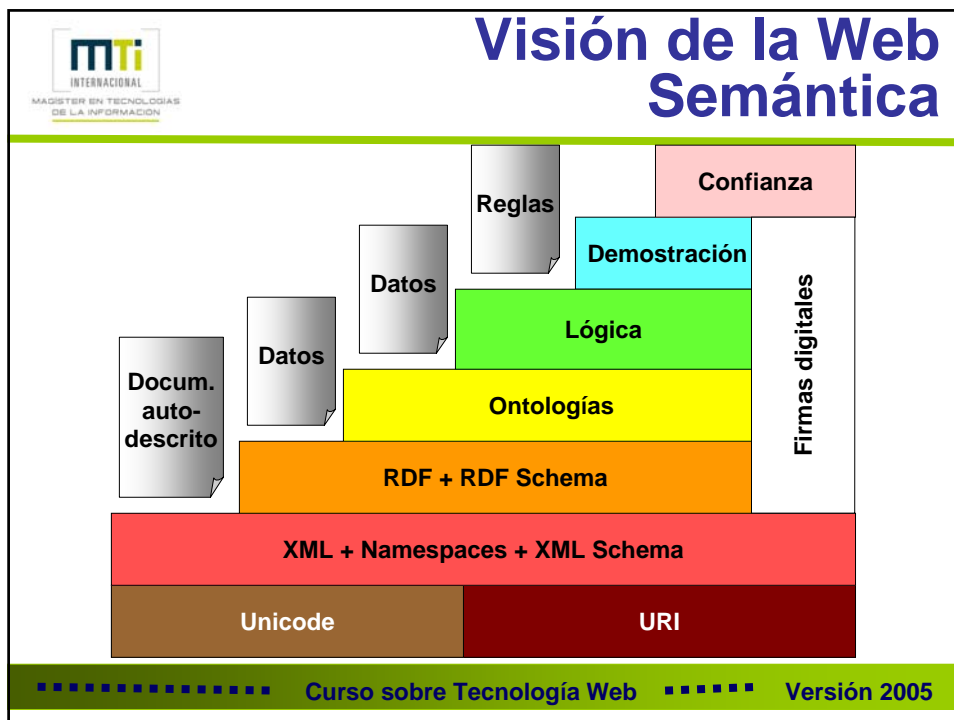
Características de la Web que deben tenerse en cuenta...

No centralizada: problemas para garantizar integridad de la información)

Información Dinámica: puede cambiar la información e incluso el conocimiento sobre esa información

Mucha información: El sistema no puede pretender acaparar toda la información

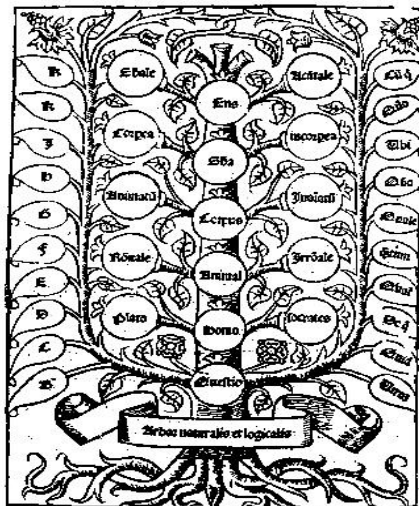
Es abierta: Muchos sistemas anteriores usaban la *Closed World Assumption*



mti
INTERNACIONAL
MAGISTER EN TECNOLOGÍAS
DE LA INFORMACIÓN

Antecedentes

..... Curso sobre Tecnología Web Versión 2005



Árbol de la naturaleza y de la lógica
 Ramón Llull (1235-1316)

Orígenes: Aristóteles (-342 a. de C.)

Desarrollo de la Lógica formal a finales s. XIX (De Morgan, Fregge)

Lógica computacional (Hilbert, Church, Turing, Herbrand, Tarski, ...)

Lógica proposicional

Lógica de Predicados de Primer Orden

$\forall x(\text{elefante}(x) \rightarrow \text{mamífero}(x))$ $\forall x(\text{mamífero}(x) \rightarrow \text{lactante}(x))$ $\forall x(\text{elefante}(x) \rightarrow \text{númeroPatas}(x,4))$ $\exists x(\text{elefante}(x) \wedge \text{seLlama}(\text{Pipo}))$	Premisas	} Razonamiento
$\exists x (\text{lactante}(x))$	Conclusión	

Sistemas de Razonamiento

Sistema de razonamiento: Conjunto de Axiomas + Reglas de Inferencia
Propiedades

Expresividad: Capacidad de representar un problema

Consistencia: Todo lo que se deduce es correcto

Compleitud: Todo lo que es correcto puede deducirse

Decidibilidad: Existe un algoritmo para decidir si se cumple una conclusión

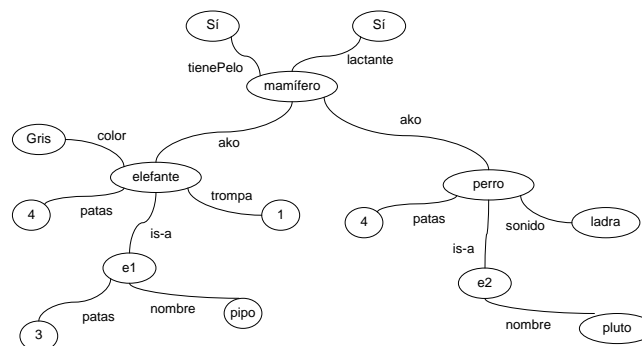
Tratabilidad: El algoritmo de decisión tiene una complejidad razonable

Clases de complejidad

$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME \subseteq EXPSPACE$

Redes Semánticas

Redes Semánticas (Quillian, 68): Grafos dirigidos donde los vértices son conceptos y los enlaces son relaciones entre conceptos
2 tipos especiales de relaciones: is-a (pertenencia) y ako (inclusión)



Desarrollados en los años 70 para estructurar el conocimiento de las redes semánticas

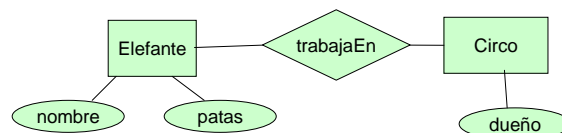
Un *frame* o marco = colección de atributos (slots) que describen una entidad

Puede representar un concepto (o clase) y un individuo (o instancia)

<p>Clase: Mamífero</p> <p> tienePelo: Sí</p> <p> lactante: Sí</p>	<p>Individuo: e1</p> <p> isa: Elefante</p> <p> patas: 3</p> <p> nombre: Pipo</p>
<p>Clase: Elefante</p> <p> ako: Mamifero</p> <p> patas: 4</p> <p> trompa: 1</p> <p> color: gris</p>	<p>Individuo: e2</p> <p> is-a: Perro</p> <p> nombre: Pluto</p>
<p>Clase: Perro:</p> <p> ako: Mamifero</p> <p> patas: 4</p> <p> sonido: ladra</p>	

Diagramas Entidad-Relación (Chen, 1976): Representaciones gráficas utilizadas para capturar modelos de dominio.

Utilizados en el desarrollo de Bases de Datos



Mapas de Tópicos (Topic Maps)

Mapas de tópicos (<http://www.topicmaps.org/>)

Estándar de definición de índices

XTM es un vocabulario para mapas de tópicos basado en XML

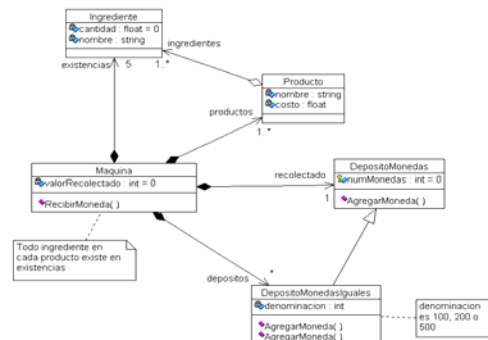
```

<topic id="pizzas"/> ...
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#barbacoa"/>
    </instanceOf>
    <scope>
      <topicRef xlink:href="#pizza"/>
    </scope>
    <resourceRef xlink:href="barbacoa.jpg"/>
  </occurrence>
  ...
</topic>
    
```

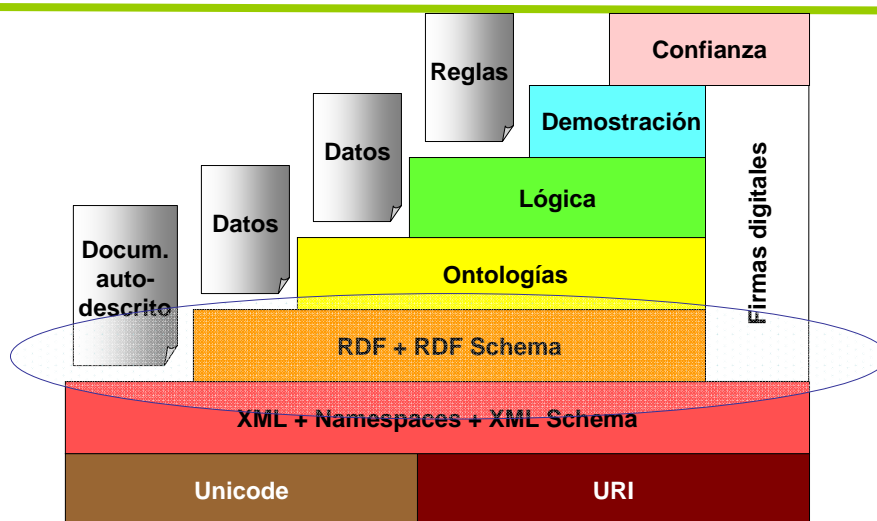
Modelos Orientados a Objetos

Modelos Orientados a Objetos: Especificación de herencia y jerarquía de objetos

Lenguajes de modelado. UML incluye diagramas de clase que describen la estructura de objetos, atributos, operaciones, etc.



RDF



RDF (Resource Description Framework)

Mecanismo para describir recursos

Recurso: Cualquier cosa que pueda nombrarse mediante una URI

Propiedad: Característica o atributo de un recurso

Tiene asociada una URI y un significado concreto

Puede relacionarse con otras propiedades

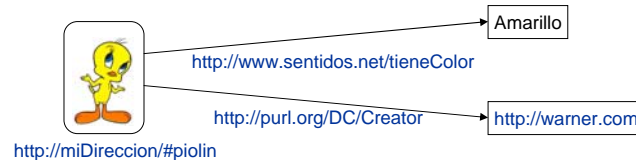
Enunciado: Asocia el valor de una propiedad a un recurso

RDF define un modelo (= Grafo dirigido)

Nodos Orígen = URIs

Arcos = URIs

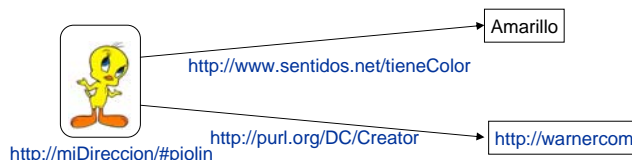
Nodos destino (URIs o literales)



Se podría representar como:

(<http://miDireccion/#Piolín>, <http://www.sentidos.net/tieneColor>, "Amarillo")
(<http://miDireccion/#Piolín>, <http://purl.org/DC/Creator>, <http://warner.com>)

El modelo RDF se puede representar en sintaxis XML (serialización)



```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.sentidos.net/"
  xmlns:dc="http://purl.org/DC/">
  <rdf:Description about="http://miDireccion/#piolin">
    <s:tieneColor>Amarillo</s:tieneColor>
    <dc:Creator resource="http://warner.com" />
  </rdf:Description>
</rdf:RDF>

```

Sintaxis abreviada

```

<rdf:Description rdf:ID="Piolin">
  <s:tieneColor>Amarillo</s:tieneColor>
  <rdf:type resource="#Canario" />
</rdf:Description>

```

Si las propiedades no se repiten,
pueden incluirse como atributos

```

<rdf:Description rdf:ID="Piolin" s:tieneColor="Amarillo">
  <rdf:type resource="#Canario" />
</rdf:Description>

```

La declaración de "type" puede
incluirse en la etiqueta

```

<rdf:Canario rdf:ID="Piolin" s:tieneColor="Amarillo" />

```


Sintaxis Abreviada Notación 3

N3 es una sintaxis RDF no XML

Facilita la representación/manipulación humana de tripletas

Objetivo: Legibilidad por desarrolladores

Utilizada por CWM

```

@prefix : <http://miDireccion/#> .
@prefix dc: <http://purl.org/DC/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix s: <http://www.sentidos.net/> .

:piolin    dc:Creator <http://warner.com>.
:piolin    s:tieneColor "Amarillo" .
    
```

RDF: Contenedores

Tipos

Bag: Conjunto no ordenado (permite duplicados)

Seq: Lista ordenada (permite duplicados)

Alt: Valor único alternativo (elección de un elemento del contenedor)

Los elementos se indican con

Pueden tener ID (pero no [about](#))

Pueden aplicarse propiedades a los elementos con [aboutEach](#)

```

<rdf:Description rdf:ID="Película">
  <dc:title>El canario peleón</dc:title>
  <m:personajes>
    <rdf:Bag ID="ps1">
      <li resource="http://warner.com/#piolin" />
      <li resource="http://warner.com/#silvester" />
    </rdf:Bag>
  </m:personajes>
</rdf:Description>

<rdf:Description aboutEach="#ps1">
  <dc:Creator resource="http://warner.com" />
</rdf:Description>
    
```

RDF: Reificación

Permite definir sentencias sobre sentencias (orden superior)

Ej. *El sitio Web de la EUITIO dice que Labra es el profesor de Lógica*

(WebEUITIO, dice, s1)

(s1, Subject, labra)

(s1, Predicate, esProfesorDe)

(s1, Object, lógica)

Una descripción define implícitamente un [Bag](#) de sentencias

RDF define automáticamente las sentencias como recursos y las incluye en un Bag

Las sentencias se representan con el tipo predefinido [rdf:Statement](#)

Los atributos de [rdf:Statement](#) son:

[rdf:Subject](#)

[rdf:Predicate](#)

[rdf:Object](#)

Es posible añadir otros atributos a las sentencias

RDF: Tipos de Datos

Es posible utilizar tipos de datos

En general se utilizan los tipos de XML Schema

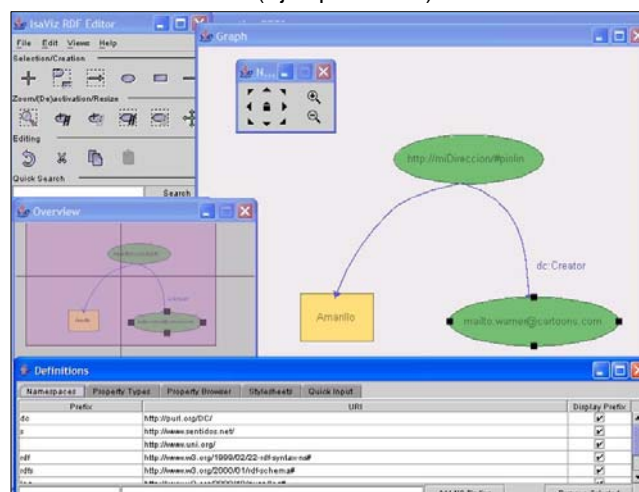
Podrían utilizarse otros tipos de datos

Se usa la notación ^^

```
<rdf:Description rdf:about="949318">  
  <uni:nombre>Jose Labra</uni:nombre>  
  <uni:cargo>Profesor Titular</uni:cargo>  
  <uni:edad  
    rdf:datatype="&xsd:integer">35</uni:age>  
</rdf:Description>
```

Creación de RDF

Existen varios editores visuales (Ejemplo **IsaViz**)



Curso sobre Tecnología Web Versión 2005

Consultas sobre RDF SPARQL

Los ficheros RDF pueden considerarse bases de datos de tripletas
 SPARQL (W3C, 2004) es un lenguaje de consulta para datos RDF
 Similar a SQL

```

...
<rdf:Description about="http://miDireccion/#piolin">
  <s:tieneColor>Amarillo</s:tieneColor>
  <dc:Creator resource="http://warner.com" />
</rdf:Description>
...

```

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?c
WHERE ( <http://miDireccion/#piolin> dc:Creator ?c )

```

```
http://warner.com
```

Curso sobre Tecnología Web Versión 2005

Aplicaciones de RDF RSS

RSS 1.0 es un vocabulario de RDF

Creación de resúmenes de sitios Web (syndication)

NOTA: Existe RSS 0.92, 0.93 y 2.0 que nose basa en RDF

```

<rdf:RDF>
- <channel rdf:about="http://www.webweavertech.com/ovidiu/weblog/">
  <title>Ovidiu Predescu's Weblog</title>
  <link>http://www.webweavertech.com/ovidiu/weblog/</link>
  <description>Technology ramblings</description>
  <dc:language>en-us</dc:language>
  <dc:creator>ovidiu</dc:creator>
  <dc:date>2004-03-02T20:17:36-08:00</dc:date>
  <admin:generatorAgent rdf:resource="http://www.movabletype.org/?v=2.64"/>
  <admin:errorReportsTo rdf:resource="mailto:ovidiu@webweavertech.com"/>
  <sy:updatePeriod>hourly</sy:updatePeriod>
  <sy:updateFrequency>1</sy:updateFrequency>
  <sy:updateBase>2000-01-01T12:00+00:00</sy:updateBase>
- <items>
  - <rdf:Seq>
    <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000318.html"/>
    <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000317.html"/>
    <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000316.html"/>
    <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000314.html"/>
    <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000307.html"/>
  </rdf:Seq>

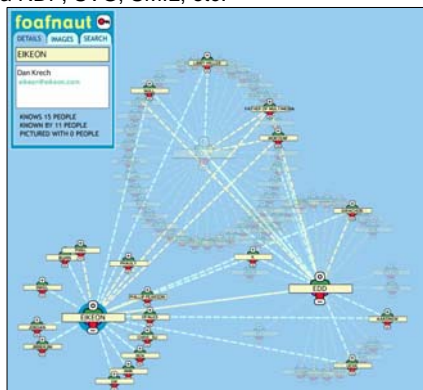
```

Aplicaciones de RDF FOAF

FOAF = Friend of a Friend (<http://rdfweb.org>)

Vocabulario para definir páginas Personales: redes sociales

FOAFNaut: Usa RDF, SVG, SMIL, etc.



Adobe XMP (eXtensible Metadata Platform)

Incluir meta-información en ficheros multimedia: imágenes, vídeos, etc.

Extensible Metadata Platform (XMP)

XMP main

Downloads

- [XMP SDK](#)
- [Custom File Info panels](#)

XMP information

- [Overview](#)
- [In depth](#)
- [Reviews and news](#)
- [Related applications](#)
- [Subscribe to the XMP newsletter](#)

Support

- [Developer forums](#)
- [Developer support](#)



Adding intelligence to media

"As any content or production professional knows, developing a workflow that actually works can be a major challenge. Keeping track of important files and assets at each stage is critical. Effective file management is an important and necessary part of the creative process, but the available tools have never been adequate.

Adobe's Extensible Metadata Platform (XMP) is a labeling technology that allows you to embed data about a file, known as metadata, into the file itself. With XMP, desktop applications and back-end publishing systems gain a common method for capturing, sharing, and leveraging this valuable metadata — opening the door for more efficient job processing, workflow automation, and rights management, among many other possibilities. With XMP, Adobe has taken the "heavy lifting" out of metadata integration, offering content creators an easy way to embed meaningful information about their projects and providing industry partners with standards-based building blocks to develop optimized workflow solutions.

[Find out more: What is XMP?](#)

"[XMP] is an important piece that brings the Semantic Web closer to realization."

— Eric Miller, W3C Semantic Web Activity Lead



RDF Schema



RDF Schema Motivación

RDF es un modelo universal que permite describir recursos
definidos en ciertos vocabularios
RDF no asume ni define ningún tipo de semántica de un dominio
concreto

En RDF Schema pueden definirse:

- Clases y propiedades
- Jerarquías y herencia entre clases
- Jerarquías de propiedades

RDF Schema Clases e individuos

Hay que distinguir entre:

Cosas concretas (individuos) del dominio.

Ej. "Jose Labra", "Lógica"

Clases o conceptos = Conjuntos de individuos que comparten
algunas propiedades (***rdfs:Class***)

Ej. "Profesor", "Asignatura", "Estudiante", ...

Los individuos que pertenecen a una clase también se llaman
instancias

La relación entre un individuo y la clase se define mediante
rdf:type

RDF Schema Rango y Dominio

El uso de clases permite declarar restricciones de Rango y de Dominio

Ejemplo: daClaseDe

Dominio: Persona

Rango: Asignatura

RDF Schema Jerarquías

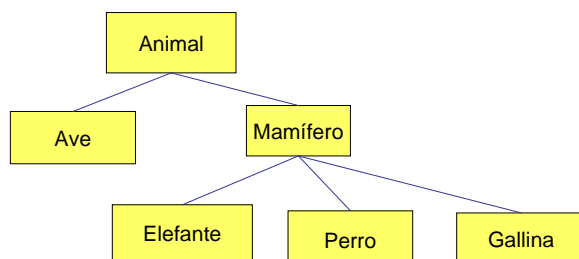
Las clases pueden organizarse en jerarquías

subclassOf define que una clase es una subclase de otra

A es una subclase de B si todo individuo de A pertenece a B

Entonces, B es una superclase de A

Una clase puede tener múltiples superclases

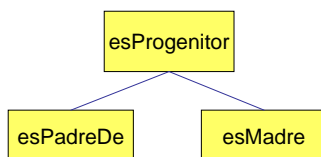


RDF Schema Jerarquía de Propiedades

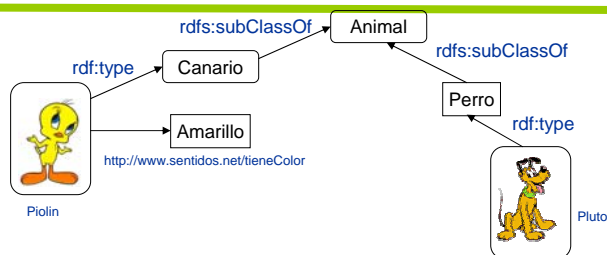
Jerarquías entre propiedades **subPropertyOf**

Ej. Ser padre es una subpropiedad de ser progenitor

Si P es una subpropiedad de Q entonces, $P(x,y)$ se cumple sólo si $Q(x,y)$



RDF Schema Sintaxis XML



```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.sentidos.net/">
  <rdf:Description rdf:ID="Piolin">
    <s:tieneColor>Amarillo</s:tieneColor>
    <rdf:type resource="#Canario" />
  </rdf:Description>

  <rdf:Description rdf:ID="Canario"> <rdf:subClassOf rdf:resource="#Animal"/></rdf:Description>
  <rdf:Description rdf:ID="Pluto"><rdf:type rdf:resource="#Perro" /></rdf:Description>
  <rdf:Description rdf:ID="Perro"><rdf:subClassOf rdf:resource="#Animal" /></rdf:Description>
  <rdf:Description rdf:ID="Animal" />
</rdf:RDF>
  
```


RDF Schema Restricciones

Propiedades básicas:

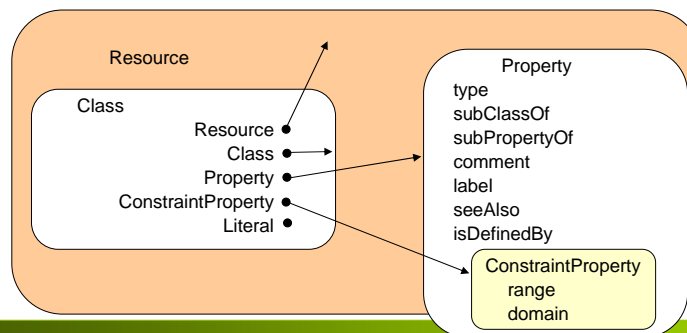
type: indica pertenencia (el valor debe ser instancia de class)

subClassOf: relación de subconjunto entre 2 clases

Relación transitiva

Una clase puede ser subclase de más de una clase (herencia múltiple)

Una clase no puede ser subclase de sí misma



RDF Schema

Otras propiedades

seeAlso: Especifica un recurso que puede proporcionar más información

isDefinedBy: Subpropiedad de **seeAlso** que permite indicar un recurso autor

comment: Asocia un comentario a un recurso

label: Asocia una etiqueta a un recurso

Restricciones

ConstraintResource: Recurso que permite definir restricciones

ConstraintProperty: Propiedad que expresa una restricción

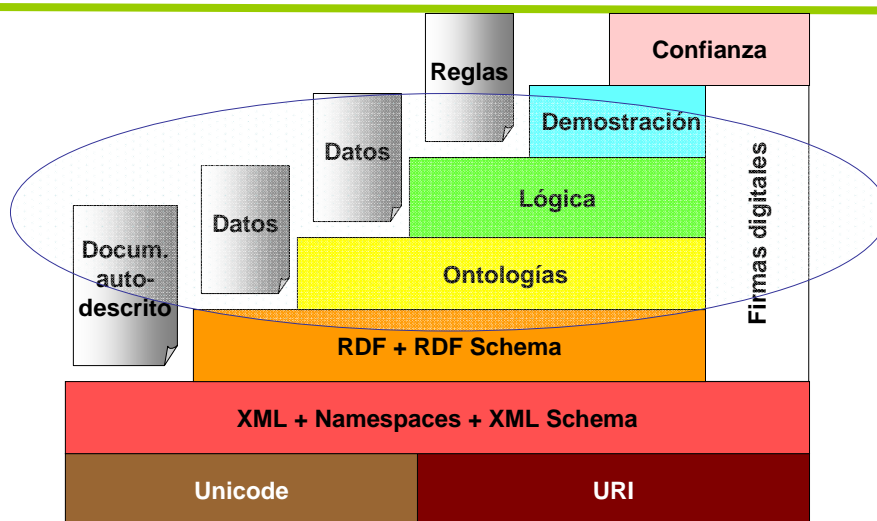
range: Se aplica a una propiedad. El valor debe ser una clase

domain: Se aplica a una propiedad. El valor debe ser una clase



Ontologías Lenguaje OWL

..... Curso sobre Tecnología Web Versión 2005



..... Curso sobre Tecnología Web Versión 2005

El término *Ontología* se utiliza en filosofía como una *disciplina que estudia la naturaleza y organización de la realidad*

En Aristóteles (Metafísica, IV, 1) se define como *la ciencia del ser*

En Informática, se utiliza como un artefacto que define:

- Un **vocabulario compartido** que describe un determinado dominio

- Un **conjunto de supuestos** sobre los términos de dicho vocabulario, generalmente se utiliza un **lenguaje formal** manipulable automáticamente.

Cyc (<http://www.cyc.com>).

- Conceptos de sentido común para Inteligencia Artificial
- Utiliza lógica de predicados mediante lenguaje CycL

Frame Ontology y OKBC Ontology

- Disponibles en Ontolingua (<http://www-ksl-svc.stanford.edu/>)
- Utiliza KIF (Knowledge Interchange Format)

Ontologías en campos concretos:

- Lingüística: WordNet (<http://www.globalwordnet.org/>)
- Medicina: GALEN (<http://www.opengalen.org/>)
- etc.

Ejemplos de Ontologías Dublin Core

Dublin Core Metadata Initiative (<http://www.dcmi.org>)

Utilizado para la catalogación de documentos

Espacio de nombres: <http://purl.org/dc/elements/1.1/>

Conjunto de elementos básicos cuyo significado es compartido

Contenido: Coverage, Description, Type, Relation, Source, Subject, Title

Propiedad Intelectual: Contributor, Creator, Publisher, Rights

Instanciación: Date, Format, Identifier, Language

Cada elemento básico admite una serie de cualificadores

Refinamiento de elementos

Ejemplo: Date.created, Description.tableOfContents

Esquema de codificación

Ejemplos: Identifier.URI, Date.DCMIPeriod

Evolución de las Ontologías para la Web

SHOE (Simple HTML Ontology Extensions) Univ. Maryland, 1996

Permite definir ontologías en documentos HTML

Objetivo = Facilitar búsquedas y anotaciones de documentos

OIL (Ontology Inference Layer)

Sintaxis RDF(S) y primitivas de representación del conocimiento en marcos

Se basa en el uso de *description logics*

DAML (DARPA Agent Markup Language)

Proyecto americano de creación de lenguaje para ontologías

DAML-OIL. Proyecto conjunto que será la base de OWL

OWL (Web Ontology Language) desarrollado en W3C (2004)

OWL (Web Ontology Language)

Desarrollado por el consorcio W3C (2004)

3 niveles:

- OWL Full. Unión de sintaxis OWL y RDF (sin restricciones)

 - No se garantiza la eficiencia ni siquiera la decidibilidad

- OWL DL (Description Logics). Limita la expresividad intentando conseguir decidibilidad

 - Se pierde compatibilidad con RDF(S)

- OWL Lite. Subconjunto de OWL DL más fácil de implementar (y se espera que más eficiente)

OWL DL se basa en Lógica Descriptiva (Description Logics)

- En realidad equivale al formalismo $\mathcal{SHOIN}(\mathcal{Dn})$

Características

- Semántica bien definida

- Propiedades formales (decidibilidad, complejidad)

- Algoritmos de razonamiento conocidos

- Varios Sistemas que lo implementan (ej. Racer, Fact, etc.)

- Incluye tipos de datos primitivos de XML Schema

Lógica Descriptiva

La lógica descriptiva consiste en:

Conceptos (o clases)

Ejemplo: Padre, Madre, Persona

Propiedades (o roles): Relaciones entre conceptos

Ejemplo: tieneHijo, esPadreDe

Individuos: Elementos del dominio

Ejemplo: Juan, Sergio, ...

Lógica Descriptiva

La base de conocimiento contiene 2 partes

TBox (Términos): Descripción de conceptos

Ejemplo: $\text{Padre} \subseteq \text{Persona} \wedge \exists \text{ tieneHijo Persona}$

$\text{Orgullosa} \subseteq \text{Persona} \wedge \exists \text{ tieneHijo ReciénNacido}$

$\text{ReciénNacido} \subseteq \text{Persona}$

ABox (Aserciones): Descripción de individuos

Ejemplo: $\text{ReciénNacido}(\text{Sergio})$

$\text{tieneHijo}(\text{Jose}, \text{Sergio})$

$\text{Persona}(\text{Jose})$

Lógica Descriptiva Definición de Conceptos

Definición de conceptos

Equivalencia: $C \equiv D$

Ejemplo: $\text{Asturiano} \equiv \text{NacidoEnAsturias}$

Subclase: $C \subseteq D$ (C está incluido en D ó D subsume a C)

Ejemplo: $\text{Asturiano} \subseteq \text{Español}$

Intersección: $C \cap D$

Ejemplo: $\text{Mujer} \equiv \text{Persona} \cap \text{Femenino}$

Unión: $C \cup D$

Ejemplo: $\text{Persona} \equiv \text{Hombre} \cup \text{Mujer}$

Complemento: $\neg C$

Ejemplo: $\text{Masculino} \equiv \neg \text{Femenino}$

Concepto vacío: \perp

Clases Disjuntas: $C \cap D \equiv \perp$

Lógica Descriptiva Cuantificadores

Descripción de Propiedades

Existencial ($\exists R C$)

x pertenece a $\exists R C$ si existe algún valor $y \in C$ tal que $R(x,y)$

Ejemplo: $\text{Madre} \equiv \text{Mujer} \cap \exists \text{ tieneHijo Persona}$

Universal ($\forall R C$)

x pertenece a $\forall R C$ si para todo y, si $R(x,y)$ entonces $y \in C$

Ejemplo: $\text{MadreFeliz} \equiv \text{Madre} \cap \forall \text{ tieneHijo Sano}$

Una Madre es feliz si todos sus hijos están sanos

NOTA: Si no tuviese hijos, también se cumpliría...

Lógica Descriptiva

Restricciones Cuantitativas

Restricciones cuantitativas

Cardinalidad ($P = n$)

x pertenece a ($P = n$) si existen n $y \in C$ tales que $R(x,y)$

Ejemplo: $\text{Elefante} \subseteq \text{Animal} \cap \text{tienePatatas} = 4$

Cardinalidad mínima ($P \geq n$)

x pertenece a ($P \geq n$) si existen n ó más $y \in C$ tales que $R(x,y)$

Ejemplo: $\text{BuenEstudiante} \equiv \text{Estudiante} \cap \text{tieneAprobada} \geq 3$

Cardinalidad máxima ($P \leq n$)

x pertenece a ($P \leq n$) si existen n ó menos $y \in C$ tales que $R(x,y)$

Ejemplo: $\text{MalEstudiante} \equiv \text{Estudiante} \cap \text{tieneAprobada} \leq 3$

Lógica Descriptiva

Atributos de propiedades

Inversa: P es inversa de $Q \Rightarrow P(x,y) \Leftrightarrow Q(y,x)$

Ejemplo: daClaseDe es inversa de tieneProfesor

SubPropiedad: P es subpropiedad de Q si siempre que $P(x,y)$ entonces $Q(x,y)$

Ejemplo: esHijoDe es subpropiedad de esDescendienteDe

Transitividad: Si $P(x,y)$ y $P(y,z)$ entonces $P(x,z)$

Ejemplo: antepasado

Simetría: Si $P(x,y)$ entonces $P(y,x)$

Ejemplo: hermano

Lógica Descriptiva

Atributos de propiedades

Propiedad Funcional. Como mucho tiene un valor para cada objeto.

Si $P(x,y)$ y $P(x,z)$ entonces $y = z$

Ejemplo: *edad*

Propiedad Funcional inversa. Dos objetos diferentes no pueden tener el mismo valor. Si $P(x,y)$ y $P(z,y)$ entonces $x = z$

Ejemplo: *dni*

Lógica Descriptiva

Razonamiento

A partir de una base de conocimiento Σ se ofrecen varios mecanismos de inferencia:

1.- Satisfacibilidad de conceptos: De Σ no se deduce que $C \equiv \perp$

Ejemplo: *Orgullosos* \cap *ReciénNacidos*

2.- Subsunción: Deducir si un concepto está incluido en otro

$\Sigma \Rightarrow C \subseteq D$

Ejemplo: *Orgullosos* \subseteq *Padre*

$\text{Padre} \subseteq \text{Persona} \cap \exists \text{ tieneHijo Persona}$ $\text{Orgullosos} \subseteq \text{Persona} \cap \exists \text{ tieneHijo ReciénNacido}$ $\text{ReciénNacido} \subseteq \text{Persona}$ $\text{Padre} \subseteq \neg \text{ReciénNacido}$

$\text{ReciénNacido}(\text{Luis})$ $\text{tieneHijo}(\text{Jose}, \text{Luis})$ $\text{Persona}(\text{Jose})$

Lógica Descriptiva Razonamiento

3.- Instanciación: $\Sigma \Rightarrow a \in C$

Ejemplo: Orgullosa(José)

4.- Recuperación de Información

Dado un concepto C, obtener a tales que $a \in C$

Ejemplo: ? Orgullosa

José

5.- Realización/Comprensión (realizability).

Dado un elemento a, obtener concepto más específico C tal que $a \in C$

Ejemplo: ? José

Orgullosa

Padre \subseteq Persona $\cap \exists$ tieneHijo Persona Orgullosa \subseteq Persona $\cap \exists$ tieneHijo ReciénNacido ReciénNacido \subseteq Persona Padre $\subseteq \neg$ ReciénNacido
--

ReciénNacido(Luis) tieneHijo(José,Luis) Persona(José)

OWL Sintaxis XML

OWL se basa en RDF (utiliza sintaxis XML de RDF)

También existen otras formas sintácticas más sencillas

Las ontologías comienzan por owl:Ontology

```

<owl:Ontology rdf:about="">
  <rdfs:comment>Ejemplo de Ontología</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.uniovi.es/viejo"/>
  <owl:imports
    rdf:resource="http://www.uniovi.es/personas"/>
  <rdfs:label>Ontología de la Universidad</rdfs:label>
</owl:Ontology>
  
```

owl:imports es una propiedad transitiva

Clases en OWL

Las clases se definen mediante owl:Class

owl:Class es una subclase de rdfs:Class

Clases equivalentes mediante equivalentClass

```

<owl:Class rdf:ID="Profesor">
  <owl:equivalentClass rdf:resource="#PersonalDocente"/>
</owl:Class>
  
```

owl:Thing es la clase más general

owl:Nothing es la clase vacía

Las clases disjuntas se definen mediante owl:disjointWith

```

<owl:Class rdf:about="#ProfesorAsociado">
  <owl:disjointWith rdf:resource="#catedrático"/>
  <owl:disjointWith rdf:resource="#titular"/>
</owl:Class>
  
```

Propiedades en OWL

Hay 2 tipos de propiedades

Propiedades de Objetos relacionan un objeto con otro objeto. ej. "daClaseDe"

```

<owl:ObjectProperty rdf:ID="esHijoDe">
  <owl:domain rdf:resource="#Persona"/>
  <owl:range rdf:resource="#Persona"/>
  <rdfs:subPropertyOf rdf:resource="#esDescendienteDe"/>
</owl:ObjectProperty>
  
```

Propiedades de tipos de datos relacionan un objeto con valores de tipos de datos (enteros, literales, etc.), ej. "edad"

Habitualmente, se utilizan los tipos de datos de XML Schema

```

<owl:DatatypeProperty rdf:ID="edad">
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
  
```

Propiedades en OWL

Clases como restricciones de propiedades

Se puede declarar que la clase C satisface ciertas condiciones (todos sus individuos deben satisfacerlas)

Es lo mismo que decir que C es una subclase de C' donde C' incluye todos los objetos que satisfacen dichas condiciones

```
<owl:Class rdf:about="#personalDocente">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#daClaseDe"/>
      <owl:someValuesFrom rdf:resource="#Asignatura"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Propiedades en OWL Restricciones

allValuesFrom (\forall) indica que todos los valores deben ser de un tipo

NOTA: Los que no tiene ningún valor, también cumplen la condición

someValuesFrom (\exists) Al menos un valor de la propiedad debe tener un tipo

Ejemplo: Un estudiante es una persona que cursa al menos una asignatura

hasValue Al menos uno de los valores tiene un valor

minCardinality, **maxCardinality** restringen el número máximo/mínimo de valores

Propiedades en OWL

Combinaciones booleanas

Combinaciones booleanas

complementOf, unionOf, intersectionOf

```
<owl:Class rdf:ID="personasUniversidad">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#personalDocente"/>
    <owl:Class rdf:about="#estudiantes"/>
    <owl:Class rdf:about="#PAS"/>
  </owl:unionOf>
</owl:Class>
```

Propiedades en OWL

Enumeraciones

oneOf permite realizar enumeraciones

```
<owl:oneOf rdf:parseType="Collection">
  <owl:Thing rdf:about="#Lunes"/>
  <owl:Thing rdf:about="#Martes"/>
  <owl:Thing rdf:about="#Miércoles"/>
  <owl:Thing rdf:about="#Jueves"/>
  <owl:Thing rdf:about="#Viernes"/>
  <owl:Thing rdf:about="#Sábado"/>
  <owl:Thing rdf:about="#Domingo"/>
</owl:oneOf>
```

Individuos en OWL

Se declaran igual que en RDF

```
<rdf:Description rdf:ID="jose">  
  <rdf:type rdf:resource= "#profesor"/>  
</rdf:Description>
```

```
<personalDocente rdf:ID="jose">  
  <uni:edad rdf:datatype="&xsd;integer">35<uni:edad>  
</personalDocente>
```

Otras Características de OWL

No se asume nombres únicos

Si 2 individuos tienen nombre diferente no quiere decir que sean diferentes

Ej. Si se declara que cualquier curso sólo tiene un profesor y que un curso tiene 2 profesores, entonces se infiere que ambos profesores son el mismo.

Para indicar que son diferentes, hay que declararlo explícitamente mediante "differentFrom", "distinctMembers"

OWL admite los tipos de datos predefinidos de XML

Schema pero no trabaja con los tipos definidos por el usuario

Información de versiones

Existen varias herramientas de manipulación de documentos OWL

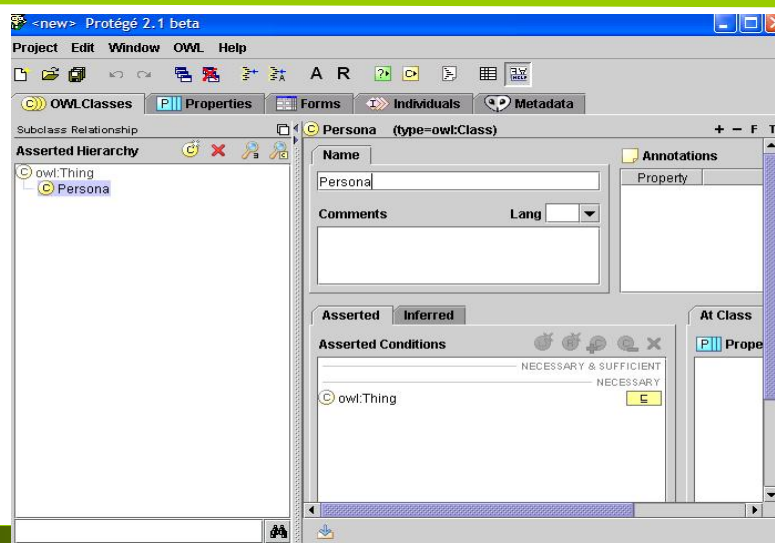
OILED, Triple20 (Prolog), SWOOP (Java), Protégé

Protégé (<http://protege.stanford.edu>) es una herramienta para creación de ontologías desarrollada en Stanford (se basa en Frames)

Arquitectura que facilita el desarrollo de plugins

Plugin para edición de documentos OWL

Swoop: Herramienta inspirada en un visualizador web con la posibilidad de editar ontologías



OWL

Sistemas de Inferencia

CWM. Desarrollado por Tim Berners Lee en Python
Incluye sistema de inferencia
Sintaxis n3 y RDF
Euler. Sistema de inferencia Admite n3
SWISH (Semantic Web Inference System in Haskell)
JENA. API Java para RDF. Incluye sistema de inferencia
SwiProlog. Incluye librerías de Prolog y el sistema Triple20 que permite
editar OWL
MetaLog. Basado en Prolog, permite sintaxis en pseudo-lenguaje natural
RACER. Sistema de inferencia implementado en Lisp
Funciona como un servicio http

Nuevos Retos

Nuevos Retos Limitaciones de OWL

Sistema de módulos

Mecanismo de importación primitivo (sin ocultación de información)

Valores por defecto

La adopción de *Open World Assumption* puede limitar capacidad de razonamiento

No asunción de nombres únicos (individuos con diferentes nombres no tienen porqué ser diferentes)

Algunos razonadores no la incluyen (ej. Racer)

Inclusión de Reglas

Además de Lógica Descriptiva, existen otros subconjuntos de lógica:
Cláusulas Horn

Las Cláusulas Horn son la base de la programación lógica

Existen implementaciones eficientes

C. Horn vs. L. Descriptiva

Algunas propuestas de simplificar OWL (ej. OWL Flight) para alcanzar
Programación Lógica Descriptiva

SWRL (Semantic Web Rule Language) es una propuesta de creación de un lenguaje de reglas que añada reglas tipo Prolog a OWL

Orígenes: RuleML (Iniciativa internacional)

`progenitor(?x,?y) ∧ hermano(?x,?z) → tío(?z,?y)`

Lógicas Monótonas vs. no monótonas

Lógica Descriptiva = Lógica Monótona

Al añadir nueva información, la información existente se mantiene

Problema: Un vendedor *on-line* quiere hacer un descuento especial si es cumpleaños del cliente

Solución 1

R1. Si cumpleaños entonces descuento

R2. Si no cumpleaños entonces no descuento

¿Qué ocurre si una persona no declara cuándo es su cumpleaños?

R2'. Si no se sabe si es cumpleaños entonces no descuento

R2' no está en lógica de primer orden, se basa en información incompleta.

Servicios Web Semánticos

Reto: Descripción de servicios Web mediante tecnologías de Web Semántica

WSDL describe la interfaz

...pero es necesario otro tipo de descripciones:

Descubrimiento automático

Composición e interoperación entre servicios

Monitorización de servicios

Procesos

Calidad de Servicio

- etc.

Propuesta de vocabulario: OWL-S

<http://www.w3.org/Submission/OWL-S>

El futuro de la Web Semántica

La Web Semántica está de modapuede ser un problema...

Compromiso Expresividad vs Eficiencia

Razonamiento con individuos limitado

Complejidades exponenciales

Aplicaciones de muestra rudimentarias

Necesidad de una *Killer Application*

Generación de meta-información

Representación de meta-información

Depuración de ontologías

¿Y la confianza?

Inclusión de Técnicas de certificación

Explicación de Respuestas (D. McGuiness)

Ejercicios

1.- Arquitecturas Orientadas a Servicios

<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>

2.- Servicios Web Semánticos

<http://www.swsi.org/>

3.- Sistemas de reglas para la Web Semántica

<http://www.w3.org/2004/12/rules-ws/report/>

4.- Confianza en la Web

<http://trust.mindswap.org>

Selección de Enlaces



Página del consorcio: <http://www.w3c.org/RDF>
SemanticWeb: <http://www.semanticweb.org>
Directorio de Ontologías: <http://www.schemaweb.info>
Ontologías: <http://www.ontology.org>
Dublin Core: <http://www.dcmi.org>
Darpa Markup Language: <http://www.daml.org>
Open Directory Project: <http://www.dmoz.org>
OntoWeb: <http://www.ontoweb.org>
Topic Maps: <http://easytopicmaps.com>

Fin de la Presentación

