

## *Tecnologías Web* *Web Semántica*

Jose Emilio Labra Gayo  
Departamento de Informática  
Universidad de Oviedo

## *Esquema de la exposición*

### Lenguajes para la Web

- Diseño de lenguajes

- XML

- Vocabularios XML

### Arquitecturas Web

- Esquemas Cliente/servidor

- Sistemas distribuidos

- Servicios Web

- Otras Arquitecturas: agentes, *peer-to-peer*, *grid*

### Web Semántica

- Descripción de recursos

- Representación del conocimiento

- Ontologías

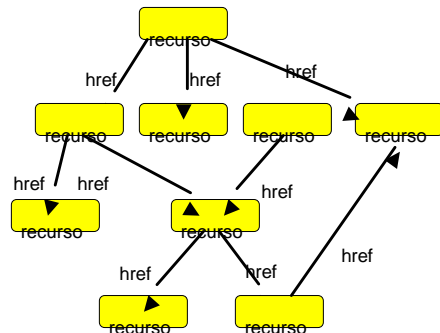
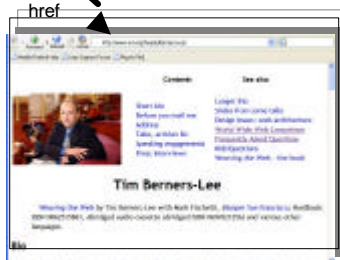
Web actual = mayor almacén de información recopilado por personas humanas

Grandes cantidades de información sobre cualquier asunto

Acceso casi instantáneo desde cualquier lugar con conexión a Internet

Sistema no centralizado ⇒ Cualquier persona puede añadir más información

Podría considerarse que ahora estamos en la **Web Sintáctica**



Características actuales de la Web sintáctica

Biblioteca Digital con hipertexto

Enorme biblioteca con documentos (llamados *páginas Web*) conectados entre sí mediante enlaces

Una base de datos (o plataforma común de aplicaciones)

Un portal común de aplicaciones accesibles a través de páginas Web y que muestran sus resultados como páginas Web

Una plataforma para multimedia

Una nueva forma de transmitir programas de radio, TV y vídeos

Un esquema de nombres

Identidad única para los documentos

*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?

Buscar fotos del profesor de este curso (Labra)



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



Tecnologías Web

Buscar un ave que utilice el oído para orientarse y que no sea un murciélago

y si os digo que en inglés es *barn owl* ?



Tecnologías Web

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 inglés

Buscar y comparar noticias que hablen de las últimas elecciones

Encargar una comida en el restaurante que no tenga demasiadas calorías

The screenshot shows a web browser window with a table titled "Pizzas del Restaurante Al Capone". The table has three columns: "Pizza", "Ingredientes", and "Precio". The rows are:

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

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

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

```
<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>
```



Pero si vamos a otro contexto

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

La máquina vería...



El problema es que las etiquetas no tienen un significado compartido

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

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...

La Web semántica (Tim Berners-Lee) pretende desarrollar lenguajes que faciliten la inclusión en la Web de contenido legible por las máquinas

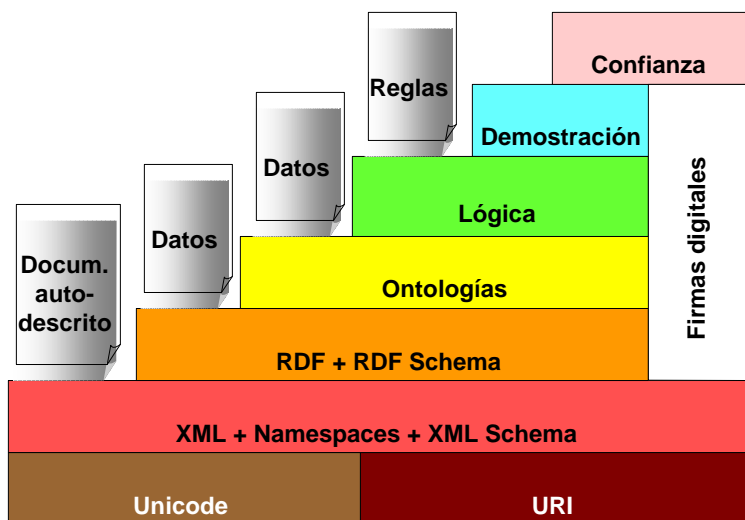
Características de la Web que deben tenerse en cuenta...

**No centralizada:** problemas para garantizar integridad de la información)

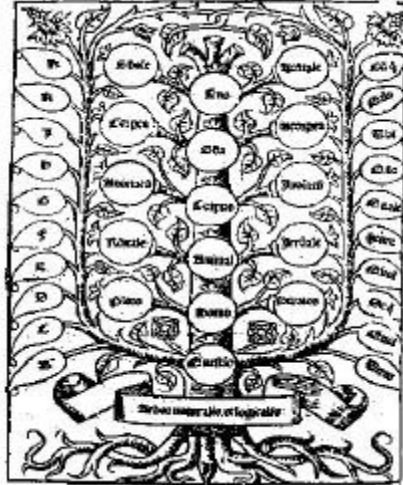
**Información Dinámica:** puede cambiar la información e incluso el conocimiento sobre esa información

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

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



Sistemas de representación del conocimiento

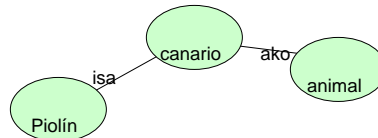


Árbol de la naturaleza y de la lógica  
Ramón Llull (1235-1316)

Tecnologías Web

Sistemas de representación del conocimiento

Redes semánticas

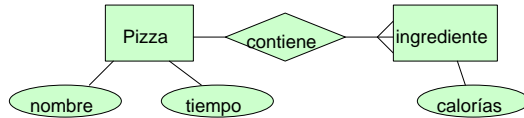


Lógica de primer orden

$$\frac{\forall x(\text{canario}(x) \rightarrow \text{animal}(x))}{\text{canario}(\text{piolín})} \\ \text{animal}(\text{piolín})$$

Tecnologías Web

Diagramas entidad-relación



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>
    
```

RDF (Resource Description Framework)

Permite describir recursos mediante propiedades y valores de propiedades

Recurso: Cualquier cosa que pueda nombrarse mediante una URI

Propiedad: Característica o atributo de un recurso

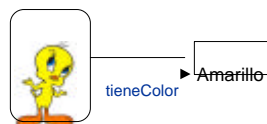
Tiene asociada una URI y un significado concreto

Puede relacionarse con otras propiedades

Enunciado: Asocia el valor de una propiedad a un recurso

Ejemplo: *Piolín es de color amarillo*

Representación gráfica



RDF = Grafo acíclico dirigido

Nodos Orígen = URIs (atributo *about*) o identificadores (atributo *ID*)

Arcos = URIs

Nodos destino (URIs o literales)



Sintaxis XML

```
<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="mailto:warner@cartoons.com" />
  </rdf:Description>
</rdf:RDF>
```

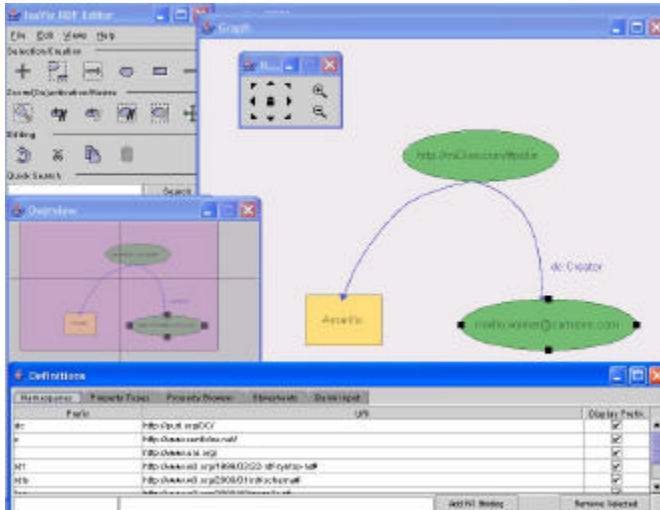


Sintaxis N3

```
@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 <mailto:warner@cartoons.com>.
:piolin s:tieneColor "Amarillo" .
```

Ejemplo de editor Visual: IsaViz



### 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" />
```

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 <li>

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="mailto:warner@cartoos.com" />
</rdf:Description>

```

Permite definir sentencias sobre sentencias (orden superior)

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

Recursos que pueden usarse para describir propiedades de otros recursos (o de otras propiedades)

Establece restricciones sobre las posibles combinaciones

Elementos:

**Resource:** Todo son recursos

**Class:** Tipo o categoría

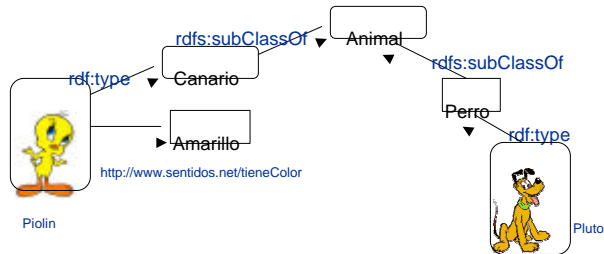
Las clases pueden ser subclases de otras

El tipo de un recurso debe ser un elemento de una clase

**Property:** Subconjunto de recursos que son propiedades

Las propiedades tienen un rango y un dominio

RDF: Propiedades predefinidas



```

<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>
  
```

Propiedades básicas:

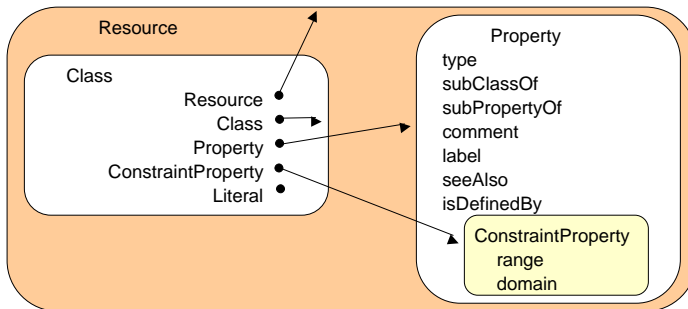
**type**: indica pertenencia (el valor debe ser instancia de class)

**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



Propiedades básicas (continuación)

**subPropertyOf**: una propiedad es una especialización de otra

Ej: padre es una especialización de progenitor

**seeAlso**: Especifica un recurso que puede proporcionar más información

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

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



Extensibilidad: Definir nuevas clases y propiedades mediante especialización



XMP = eXtensible Metadata Platform  
Incluir meta-información en imágenes

**Extensible Metadata Platform (XMP)**

**Powered By xmp**

**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 offer a common method for capturing, sharing, and leveraging the 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 Architect Lead

Tecnologías Web

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

Tecnologías Web

vCard: Información personal (<http://www.imc.org>)

```
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vCard = "http://www.w3.org/2001/vcard-rdf/3.0#">
<rdf:Description rdf:about = "http://www.mafia.it/#vito" >
<vCard:FN>Vito Corleone</vCard:FN>
<vCard:TITLE>Jefe de la pizzería</vCard:TITLE>
<vCard:TEL rdf:parseType="Resource">
  <rdf:value>+985 203040 </rdf:value>
</vCard:TEL>
<vCard:EMAIL rdf:parseType="Resource">
  <rdf:value>vito@mafia.it</rdf:value>
</vCard:EMAIL>
<vCard:ADR rdf:parseType="Resource">
  <vCard:Street>C/ Génova Nº1</vCard:Street>
  <vCard:Locality>Oviedo </vCard:Locality>
  <vCard:Country>España</vCard:Country>
</vCard:ADR>
</rdf:Description>
</rdf:RDF>
```

*Ontologías*

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.

Normalmente las ontologías tienen dos componentes:

Nombres de los conceptos

**Elefante:** es un concepto cuyos miembros son animales

**Hervívoro:** es un concepto cuyos miembros son exactamente aquellos animales que comen solamente plantas o partes de plantas

**Elefante Adulto:** es un concepto cuyos miembros son animales elefantes que tienen una edad superior a 20 años

**Conocimiento de base o restricciones:**

Los **Elefantes Adultos** pesan más de 2000kg

Todos los **Elefantes** son elefantes africanos o indios

Ningún individuo es carnívoro y hervívoro

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.

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

Permite definir ontologías en documentos HTML

Objetivo = Facilitar búsquedas y anotaciones de documentos

XOL (XML Ontology exchange Language)

Objetivo = Intercambio de definiciones de ontologías entre sistemas

OIL (Ontology Inference Layer)

Utiliza sintaxis RDF(S) y añade 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

Será la base de OWL

OWL (Web Ontology Language)

Estándar desarrollado en el consorcio Web

Desarrollado por el consorcio W3C

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

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

OWL DL se basaba en el sistema SHIQ de *description logics*

Semántica bien definida

Propiedades formales (decidibilidad, complejidad)

Algoritmos de razonamiento conocidos

Varios Sistemas que lo implementan

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human $\sqcap$ Male	$C_1 \wedge \dots \wedge C_n$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1 \vee \dots \vee C_n$
complementOf	$\neg C$	$\neg$ Male	$\neg C$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x_1 \vee \dots \vee x_n$
allValuesFrom	$\forall P.C$	$\forall$ hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	$\exists$ hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leq nP$	$\leq 1$ hasChild	$[P]_{n+1}$
minCardinality	$\geq nP$	$\geq 2$ hasChild	$\langle P \rangle_n$

**Admite tipos de XML Schema y clases en  $\exists P.C$  and  $\exists P.C$**

–E.g.,  $9 \text{ edad.nonNegativeInteger}$

**Los constructores pueden**

–E.g.,  $\text{Persona} \sqcup 8 \text{ tieneHijo.Doctor} \sqcap 9 \text{ tieneHijo.Doctor}$

Persona u 8 tieneHijo.Doctor t 9 tieneHijo.Doctor:

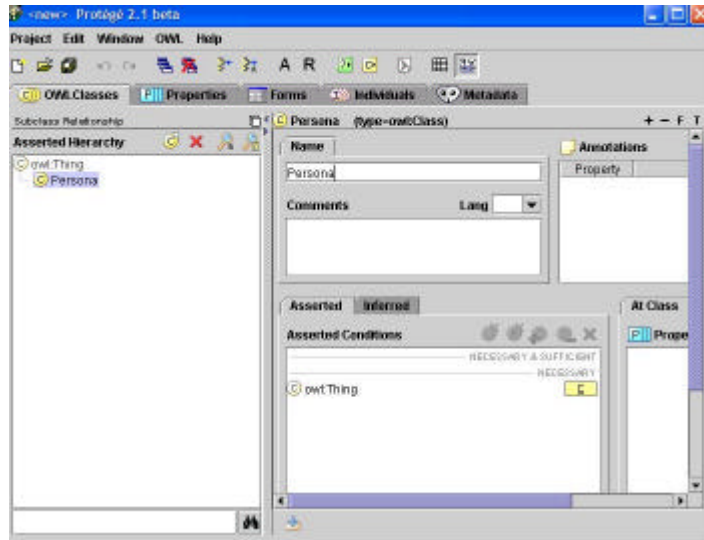
```
<owl:Class>
  <owl:intersectionOf rdf:parseType=" collection">
    <owl:Class rdf:about="#Persona"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#tieneHijo"/>
      <owl:toClass>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#tieneHijo"/>
            <owl:hasClass rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:toClass>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G.W.Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent <sup>-</sup>
transitiveProperty	$P^+ \sqsubseteq P$	ancestor <sup>+</sup> $\sqsubseteq$ ancestor
functionalProperty	$T \sqsubseteq \leq 1P$	T $\sqsubseteq \leq 1$ hasMother
inverseFunctionalProperty	$T \sqsubseteq \leq 1P^-$	T $\sqsubseteq \leq 1$ hasSSN <sup>-</sup>

Axiomas reducibles a inclusión (v)

$$C \dot{\sqsubseteq} D \text{ si y sólo si } C \vee D \text{ y } D \vee C$$

OILED  
Protégè



Tecnologías Web

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

Tecnologías Web

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.

Primeros pasos: OWL-S

<http://www.daml.org/services/owl-s/1.0/>



Página del consorcio: <http://www.w3c.org/RDF>

SemanticWeb: <http://www.semanticweb.org>

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*

