



Introducción a la Web Semántica

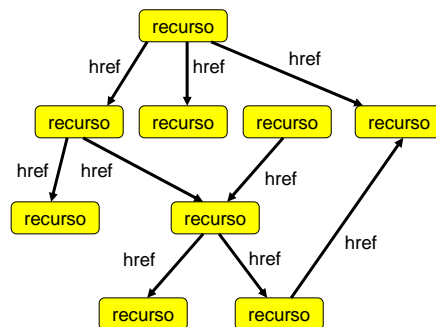
Jose Emilio Labra Gayo

Departamento de Informática
Universidad de Oviedo

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)



Tareas difíciles en la Web Sintáctica

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



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

Información multimedia: imágenes, vídeos, audio

Información en otros idiomas

Información imprecisa

Búsquedas conceptuales

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



Pizza	Ingredientes	Precio
Barbacoa	Salsa barbacoa, Mozzarella, Pollo, Bacon, Ternera	8 €
Hawaiiana	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€</td>
</tr>
...
</body>
</html>
```

¿XML como posible solución?

Possible solution: XML. Tags with own meaning

```
<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>
...
```

La aplicación que trabajaba con Pizzas no entendería las nuevas etiquetas

[illegible]

Problema: Las etiquetas no tienen un significado compartido

Web Semántica

Propuesta de la **Web semántica** (Tim Berners-Lee):

“**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 de datos en lugar de Web de documentos

Web Semántica

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

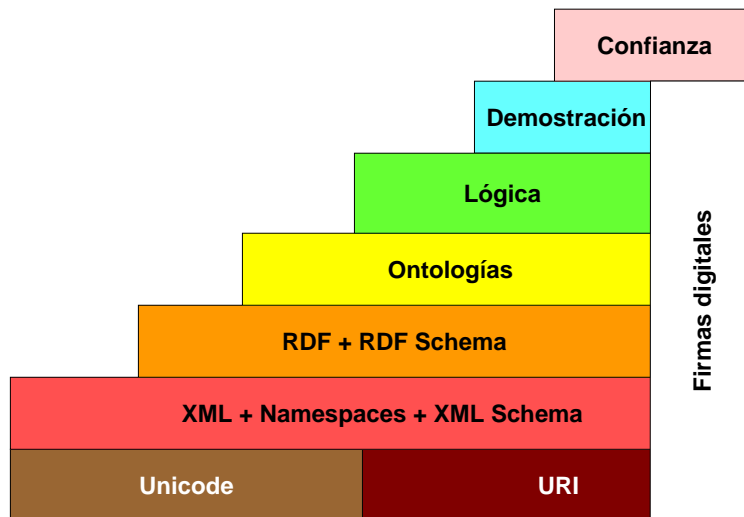
No centralizada: problemas para garantizar integridad de la información)

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

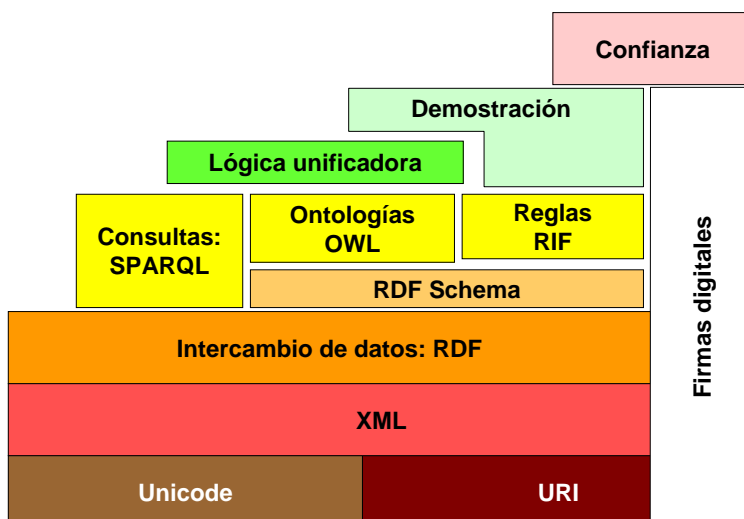
Mucha información: El sistema no puede pretender acaparar toda la información

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

Tarta de la Web

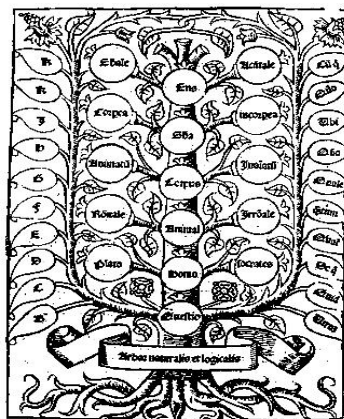


Cambios en la tarta...



Antecedentes

Ontologías



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

Lógica

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, ...)

Varios sistemas de Lógica

Lógica proposicional

Lógica de predicados

Otras lógicas: lógica modal, lógica descriptiva, lógica borrosa, etc.

Lógica Proposicional

Cada frase que puede ser verdadera o falsa es una proposición o enunciado (p)

Varias conectivas:

- Negación: $\neg p$
- Conjunción: $p \wedge q$
- Disyunción: $p \vee q$
- Implicación: $p \rightarrow q$
- Equivalencia: $p \leftrightarrow q$

*“Si Juan juega al fútbol, se cansa y Juan juega al fútbol
Por tanto: Juan se cansa”*

$$\begin{array}{r} p \rightarrow q \\ p \\ \hline q \end{array}$$

Lógica proposicional

Existen sistemas de demostración que comprueban si un razonamiento es correcto

Ejemplo: Deducción natural

Propiedades:

Consistente: todos los razonamientos que se demuestran son correctos

Completo: todos los razonamientos correctos pueden demostrarse

Complejidad: NP (es uno de los problemas NP clásicos)

Expresividad: Muy poca expresividad

Ejemplo: *“Todos los hombres son mortales, Sócrates es un hombre, luego Sócrates es mortal”*

Lógica de predicados

Extiende la lógica proposicional con predicados, funciones y cuantificadores

Ejemplo de predicado: $P(x,y)$ = x es padre de y

Ejemplo de función: $m(x)$ = “madre de x”

Cuantificadores: Existencial: $\exists xP(x)$

Universal: $\forall xP(x)$

Ejemplo: *“Todos los hombres son mortales, Sócrates es un hombre, luego Sócrates es mortal”*

$$\forall x(H(x) \rightarrow M(x))$$

$$H(s)$$

$$\hline M(s)$$

Lógica de Predicados

Existen varios sistemas de demostración en lógica de predicados.

Propiedades:

Consistente: Todo lo que demuestran es correcto

Completo: Todos lo que es correcto es demostrable

Semidecidible: Si una fórmula es correcta, lo detectan, si no lo es, pueden no detectarlo

Para resolver ese problema se han buscado subconjuntos de lógica de predicados de primer orden que sean decidibles:

Clausulas Horn

Lógica descriptiva

Complejidad

Los sistemas de demostración pueden ser muy complejos

Existe una jerarquía de clases de complejidad

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

P = Problemas que pueden resolverse por una máquina de Turing determinista en tiempo polinómico

NP= Problemas que pueden resolverse por una máquina de Turing no determinista en tiempo polinómico

PSPACE= Problemas que pueden resolverse por una máquina de Turing determinista en espacio polinómico

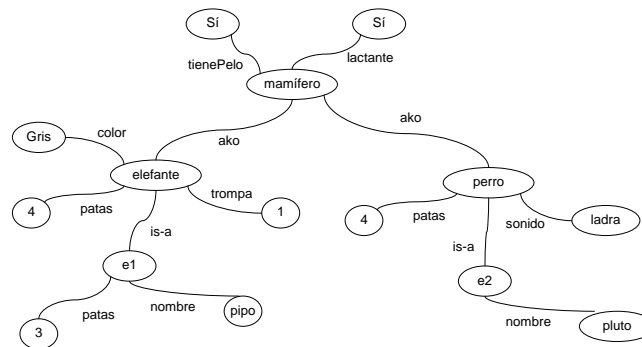
EXPTIME=Problemas que pueden resolverse por una máquina de Turing determinista en tiempo $O(2^{p(n)})$

NEXPTIME=Problemas que pueden resolverse por una máquina de Turing no determinista en tiempo $O(2^{p(n)})$

EXPSPACE=Problemas que pueden resolverse por una máquina de Turing determinista en espacio $O(2^{p(n)})$

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)



Frames

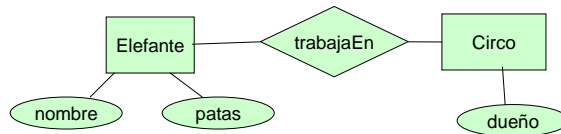
Desarrollados en los años 70 para estructurar el conocimiento de las redes semánticas

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> <ul style="list-style-type: none">tienePelo: Sílactante: Sí <p>Clase: Elefante</p> <ul style="list-style-type: none">ako: Mamíferopatas: 4trompa: 1color: gris <p>Clase: Perro:</p> <ul style="list-style-type: none">ako: Mamíferopatas: 4sonido: ladra	<p>Individuo: e1</p> <ul style="list-style-type: none">isa: Elefantepatas: 3nombre: Pipo <p>Individuo: e2</p> <ul style="list-style-type: none">is-a: Perronombre: Pluto
---	---

Diagramas Entidad-Relación

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

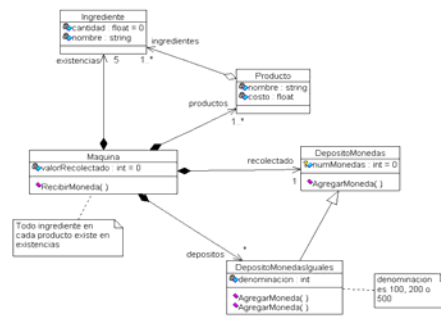
XTM es un vocabulario para mapas de tópicos basado en XML

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

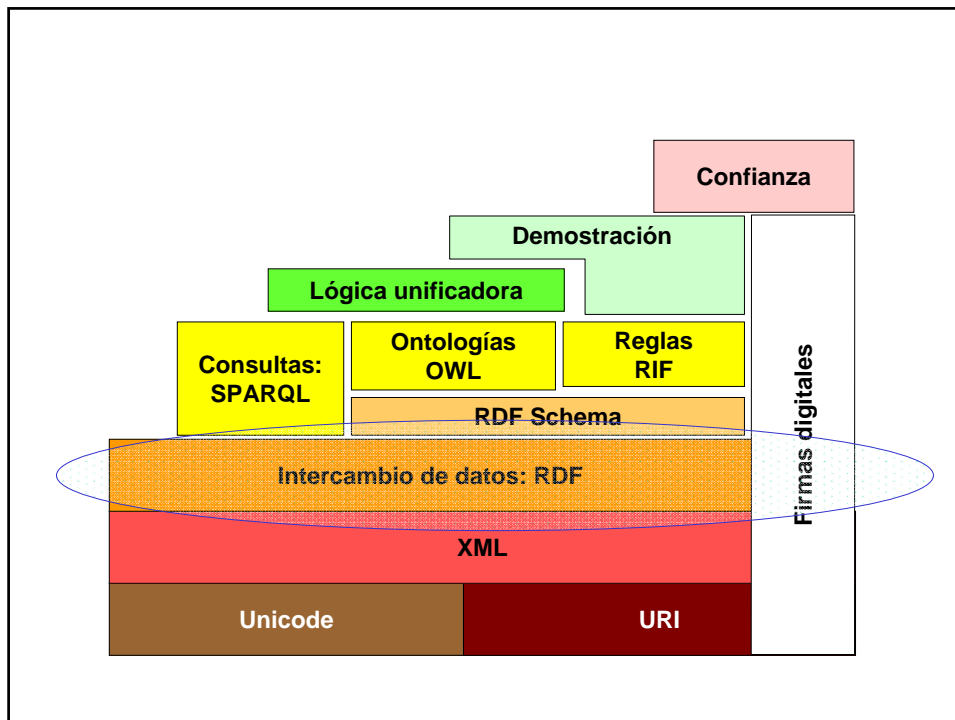
Modelos Orientados a Objetos

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

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)

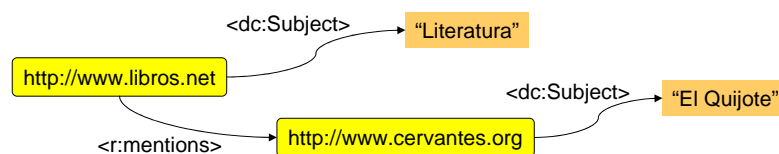
Nodos Origen = URIs o nodos anónimos

Arcos = URIs

Nodos destino (URIs, literales o nodos anónimos)

Ejemplo:

```
<http://www.libros.net> <dc:Subject> "Literatura".
<http://www.libros.net> <r:mentions> <http://www.cervantes.org>.
<http://www.cervantes.org> <dc:Subject> "El Quijote".
```



Mezcla de grafos

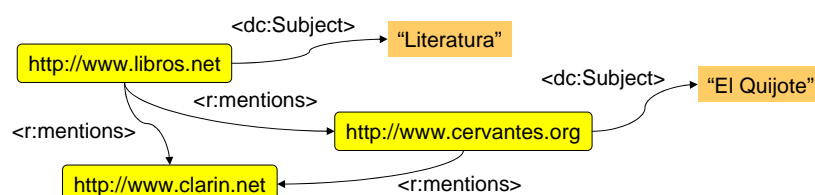
Cada descripción RDF = un grafo

Varios agentes pueden realizar descripciones independientes

Es posible mezclar dichas descripciones...

```
<http://www.libros.net> <dc:Subject> "Literatura".
<http://www.libros.net> <r:mentions> <http://www.cervantes.org>.
<http://www.cervantes.org> <dc:Subject> "El Quijote".
```

```
<http://www.libros.net> <r:mentions> <http://www.clarin.net>.
<http://www.cervantes.org> <r:mentions> <http://www.clarin.net>.
```

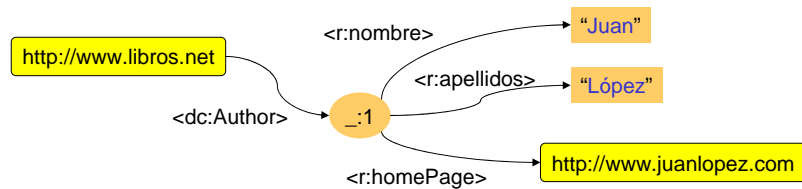


Nodos blancos (blank nodes)

Los nodos blancos son nodos que no tienen asociada una URI
Permite hacer descripciones sobre cualquier tipo de elemento, por ejemplo, personas.

La sintaxis depende de la aplicación, habitualmente: `_:nº`

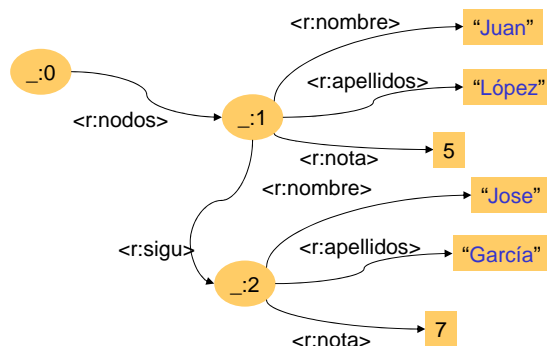
```
<http://www.libros.net>    <dc:Author> _:1 .
_:1                        <r:nombre>   "Juan".
_:1                        <r:apellidos>  "López".
_:1                        <r:homePage> <http://juanlopez.com>
```



Codificación de Tablas

Los formatos tabulares pueden codificarse en RDF

Nombre	Apellidos	Nota
Juan	López	5
Jose	García	7



RDF *Sintaxis RDF/XML*

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

```
<http://www.libros.net> <dc:Subject> "Literatura".  
<http://www.libros.net> <r:mentions> <http://www.cervantes.org>.  
<http://www.cervantes.org> <dc:Subject> "El Quijote".
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:r="http://www.relations.org/"  
  xmlns:dc="http://purl.org/DC/">  
  <rdf:Description about="http://www.libros.net">  
    <dc:Subject>Literatura</dc:Subject>  
    <r:mentions resource="http://www.cervantes.org" />  
  </rdf:Description>  
  <rdf:Description about="http://www.cervantes.org">  
    <dc:Subject>El Quijote</dc:Subject>  
  </rdf:Description>  
</rdf:RDF>
```

RDF *Notación 3*

N3 es una sintaxis RDF no XML

Facilita la representación/manipulación humana de tripletas

Objetivo: Legibilidad por desarrolladores

Utilizada por CWM

```
@prefix r: <http://www.relaciones.org/>.  
@prefix dc: <http://purl.org/DC/>  
  
<http://www.libros.net> <dc:Subject> "Literatura".  
<http://www.libros.net> <r:mentions> <http://www.cervantes.org>.  
<http://www.cervantes.org> <dc:Subject> "El Quijote".
```

Sintaxis RDF/XML

Reglas para abreviar expresiones

```
<rdf:Description rdf:about="http://www.libros.net">
  <s:subject>Literatura</s:subject>
  <rdf:type rdf:resource="#WebPage" />
</rdf:Description>
```

Si las propiedades no se repiten,
pueden incluirse como atributos

```
<rdf:Description rdf:about="http://www.libros.net"
  s:subject="Literatura">
  <rdf:type rdf:resource="#WebPage" />
</rdf:Description>
```

La declaración de “**type**” puede
incluirse en la etiqueta

```
<s:WebPage rdf:about="http://www.libros.net" s:subject="Literatura" />
```

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

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://www.ejemplos.org">

  <rdf:Description about="http://www.uniovi.es/logica">
    <a:tieneAlumnos>
      <rdf:Bag>
        <rdf:li rdf:resource="http://www.alumnos.org/juan" />
        <rdf:li rdf:resource="http://www.alumnos.org/luis" />
        <rdf:li rdf:resource="http://www.alumnos.org/marcos" />
      </rdf:Bag>
    </a:tieneAlumnos>
  </rdf:Description>
</rdf:RDF>
```

RDF: Colecciones

Pueden definirse listas de elementos
Las listas se definen como elementos cerrados.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://www.ejemplos.org">

  <rdf:Description about="http://www.uniovi.es/logica">
    <a:tieneAlumnos rdf:parseType="Collection">
      <rdf:Description rdf:about="http://www.alumnos.org/juan" />
      <rdf:Description rdf:about="http://www.alumnos.org/luis" />
      <rdf:Description rdf:about="http://www.alumnos.org/marcos" />
    </a:tieneAlumnos>
  </rdf:Description>
</rdf:RDF>
```

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

Las sentencias se representan con el tipo predefinido **rdf:Statement**

Los atributos de **rdf:Statement** son: **rdf:Subject**, **rdf:Predicate** y **rdf:Object**

Es posible añadir otros atributos a las sentencias

```
<rdf:Description rdf:about="http://www.ejemplos.org/#labra">
  <a:esProfesorDe rdf:ID="enunciado1">Logica</a:esProfesorDe>
</rdf:Description>

<rdf:Statement rdf:about="http://www.uniovi.es" >
  <a:declara rdf:resource="#enunciado1" />
</rdf:Statement>
```

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

En RDF/XML

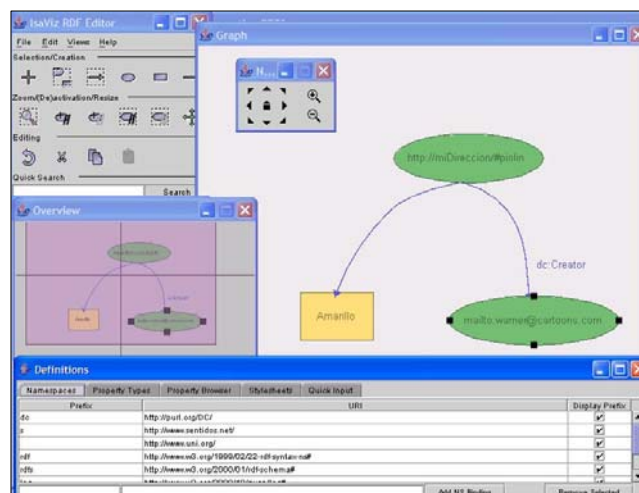
```
<rdf:Description rdf:about="949318">
  <uni:nombre>Jose Labra</uni:nombre>
  <uni:cargo>Profesor Titular</uni:cargo>
  <uni:edad rdf:datatype="&xsd:integer">35</uni:age>
</rdf:Description>
```

En n3

```
<uni:9394072> <uni:nombre> "Jose Labra".
<uni:9394072> <uni:cargo> "Profesor Titular".
<uni:9394072> <uni:edad> "35"^^<http://www.w3.org/2001/XMLSchema#integer>.
```

Edición de RDF

Editor visual **IsaViz**



RDF en HTML?

Anotación de páginas HTML mediante RDF

Problema: Incluir descripciones RDF en páginas HTML

La sintaxis RDF/XML impide la validación de HTML

Soluciones:

Incluir RDF como comentarios

Difícil de generar con las herramientas XML

Fácil para el usuario

Un comentario no deja de ser un comentario

Extender XHTML para incluir RDF

2 formas:

Añadirlo sin más \Rightarrow XHTML no válido

Extender la DTD de XHTML

Utilizar `<link>` para enlazar a un fichero RDF externo

Problema: mantenimiento de 2 ficheros independientes

Incluir RDF como comentarios

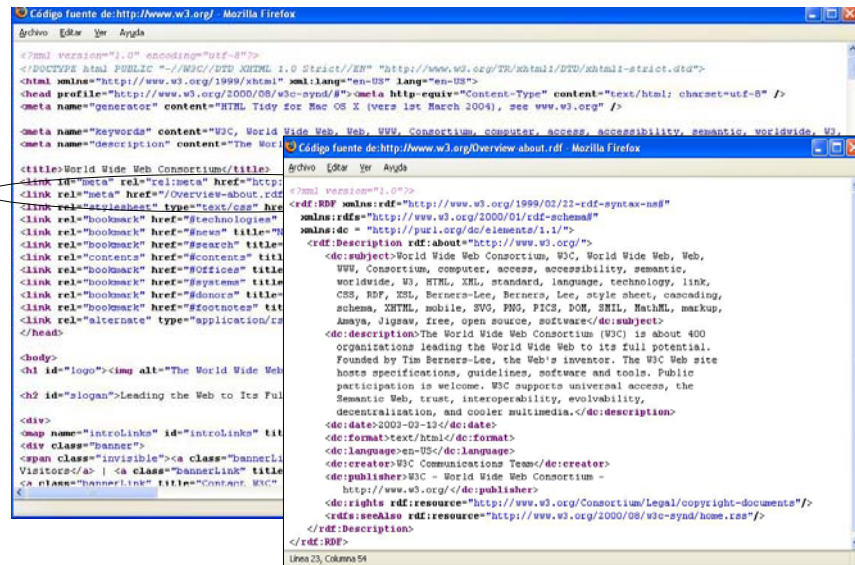
```
<div id="f-lastmod"> This page was last modified 12:26, 11 July 2006.</div>
<div id="f-copyright">This wiki is licensed to the public under a
<a href="http://creativecommons.org/licenses/by/3.0/"
class='external '
title="http://creativecommons.org/licenses/by/3.0/"
rel="nofollow">Creative Commons Attribution 3.0</a> license.<br/>
Your use of this wiki is governed by the
<a href="/CWiki:Terms_of_Use">Terms of Use</a>.
<!-- rdf:RDF xmlns="http://web.resource.org/cc/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
<Work rdf:about=""
license rdf:resource="http://creativecommons.org/licenses/by/2.5/"
dc:type rdf:resource="http://purl.org/dc/dcmitype/Text" />
</Work>
<License rdf:about="http://creativecommons.org/licenses/by/2.5/"
permits rdf:resource="http://web.resource.org/cc/Reproduction"/>
permits rdf:resource="http://web.resource.org/cc/Distribution"/>
requires rdf:resource="http://web.resource.org/cc/Notice"/>
requires rdf:resource="http://web.resource.org/cc/Attribution"/>
permits rdf:resource="http://web.resource.org/cc/DerivativeWorks"/>
</License>
</rdf:RDF> -->
</div>
<div id="f-about">
<a href="/CWiki:About" title="CWiki:About">About CWiki</a>
```

Añadir RDF en HTML

```
<!DOCTYPE html SYSTEM
"http://infomesh.net/2002/m12n/test/rdf.txt">

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:lang="en" >
<head>
<title>Pagina de Libros</title>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://www.libros.net"
dc:subject="Literatura"/>
</rdf:RDF>
</head>
...
```

Utilizar <link> para enlazar a un fichero



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US" lang="en-US">
<head profile="http://www.w3.org/2000/08/w3c-synd/#" ><meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="generator" content="HTML Tidy for Mac OS X (vers 1st March 2004), see www.w3.org" />

<meta name="Keywords" content="W3C, World Wide Web, Web, WWW, Consortium, computer, access, accessibility, semantic, worldwide, W3,
<meta name="description" content="The World Wide Web Consortium (W3C) is an international community that develops standards for the World Wide Web, such as HTML, CSS, and XML." />
<title>World Wide Web Consortium</title>
<link id="meta" rel="stylesheet" href="http://www.w3.org/2000/08/w3c-synd/#" type="text/css" />
<link rel="stylesheet" href="http://www.w3.org/2000/08/w3c-synd/#" type="text/css" />
<link rel="bookmarks" href="#technologies" title="Technologies" />
<link rel="bookmarks" href="#news" title="News" />
<link rel="bookmarks" href="#research" title="Research" />
<link rel="bookmarks" href="#contents" title="Contents" />
<link rel="bookmarks" href="#offices" title="Offices" />
<link rel="bookmarks" href="#systems" title="Systems" />
<link rel="bookmarks" href="#donors" title="Donors" />
<link rel="bookmarks" href="#footnotes" title="Footnotes" />
<link rel="alternate" type="application/atom+xml" href="http://www.w3.org/atom.xml" />
</head>
<body>
<div id="logo"><img alt="The World Wide Web Consortium logo" data-bbox="241 348 418 368"/>
<div id="slogan">Leading the Web to Its Full Potential</div>
<div>
<div name="introlinks" id="introlinks" title="Introductory Links" />
<div class="banner">
<div class="invisible"><a class="bannerlink" href="http://www.w3.org/2000/08/w3c-synd/home.css" title="Bannerlink" />
<a class="bannerlink" href="http://www.w3.org/2000/08/w3c-synd/home.css" title="Bannerlink" />
</div>
</div>
</div>
</body>
</html>
```

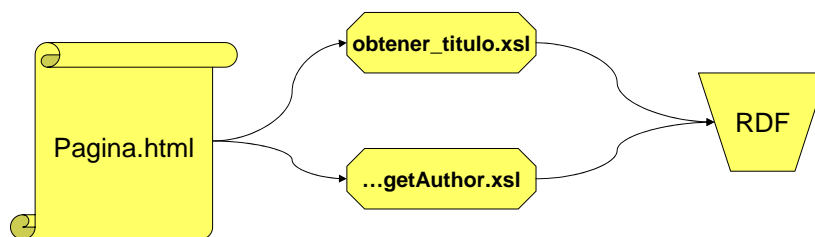
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://www.w3.org/">
<dc:subject>World Wide Web Consortium, W3C, World Wide Web, Web,
WWW, Consortium, computer, access, accessibility, semantic,
worldwide, W3, HTML, XML, standard, language, technology, link,
CSS, RDF, XML, Berners-Lee, Berners, Lee, style sheet, cascading,
schemas, XHTML, mobile, SVG, PNG, PICS, DOM, SMIL, MathML, markup,
Amaya, Jigsaw, free, open source, software</dc:subject>
<dc:description>The World Wide Web Consortium (W3C) is about 400
organizations leading the World Wide Web to its full potential.
Founded by Tim Berners-Lee, the Web's inventor. The W3C Web site
hosts specifications, guidelines, software and tools. Public
participation is welcome. W3C supports universal access, the
Semantic Web, trust, interoperability, evolvability,
decentralization, and cooler multimedia.</dc:description>
<dc:date>2002-02-13</dc:date>
<dc:format>text/html</dc:format>
<dc:language>en-US</dc:language>
<dc:creator>W3C Communications Team</dc:creator>
<dc:publisher>W3C - World Wide Web Consortium -
http://www.w3.org/</dc:publisher>
<dc:rights rdf:resource="http://www.w3.org/Consortium/Legal/copyright-documents"/>
<dc:seeAlso rdf:resource="http://www.w3.org/2000/08/w3c-synd/home.css"/>
</rdf:Description>
</rdf:RDF>
```

GRDDL

GRDDL = Gleaning Resource Descriptions over Dialects of Languages
Permite obtener información RDF a partir de ficheros XML/HTML
Se utiliza XSLT para definir una transformación de XML/HTML a RDF
Mecanismo para asociar transformaciones XSLT a tipos de documentos

Ejemplo en XHTML

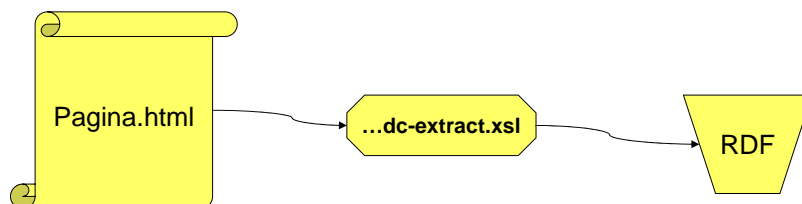
```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:grddl="http://www.w3.org/2003/g/data-view#"
      grddl:transformation="obtener_titulo.xsl
                           http://www.w3.org/2001/sw/grddl-wg/td/getAuthor.xsl" >
<head>
<title>Are You Experienced?</title>
[...]
```



Ejemplo en HTML con DTDs

Se utiliza el atributo profile (ya existía en HTML 4.02)

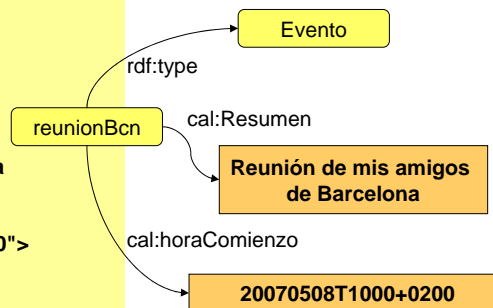
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://www.w3.org/2003/g/data-view">
  <title>Libros</title>
  <link rel="transformation"
        href="http://www.w3.org/2000/06/dc-extract/dc-extract.xsl" />
  <meta name="DC.Subject" content="Literatura" /> ...
</head>
...
</html>
```



Añadir RDF en HTML mediante atributos: RDFa

RDFa define atributos generales para añadir meta-datos a cualquier elemento HTML

```
<html
  xmlns:cal="http://www.w3.org/2002/12/cal/ical#">
...
<body>
<p class="cal:Evento"
  about="#reunionBcn">
  Voy a ir a una
  <span property="cal:Resumen">
    reunión de mis amigos de Barcelona
  </span>, el día
  <span property="cal:horaComienzo"
    content="20070508T1000+0200">
    8 de Mayo a las 10am
  </span> </p>
...
</body>
</html>
```



Aplicaciones de RDF

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 no se basa en RDF

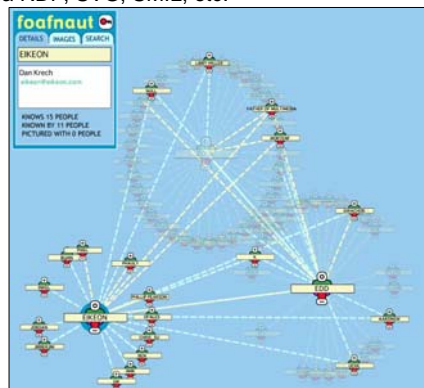
```
<rdf:RDF>
- <channel rdf:about="http://www.webweavertech.com/ovidiu/weblog/">
  <title>Ovidiu Predescu's Weblog</title>
  <link>http://www.webweavertech.com/ovidiu/weblog/</link>
  <description>Technology ramblings</description>
  <dc:language>en-us</dc:language>
  <dc:creator>ovidiu</dc:creator>
  <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>
</rdf:RDF>
```

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.



Aplicaciones de RDF XMP

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](#)

Powered By



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

DBPedia

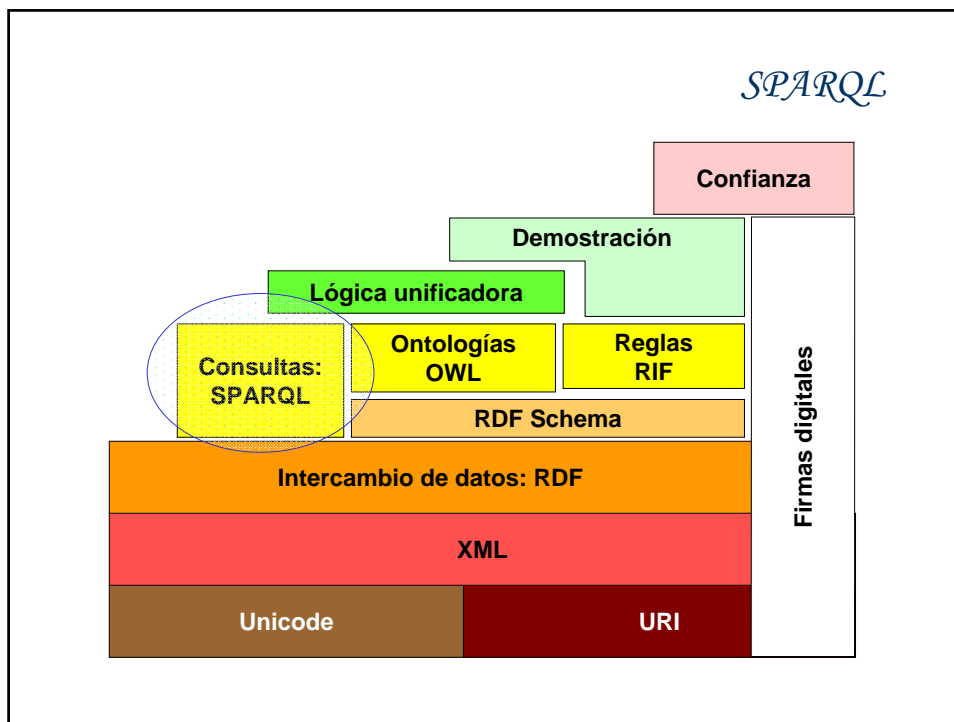
Spain at DBpedia.org

http://dbpedia.org/resource/Spain

Depiction of Spain

Spain, officially the Kingdom of Spain (, Reino de España), is a country located in Southern Europe, with two small exclaves in North Africa (both bordering Morocco). Spain is a democracy which is organized as a parliamentary monarchy. It is a developed country with the ninth-largest economy in the world. It is the larger of two sovereign states that make up the Iberian Peninsula — the other is Portugal.

Property	Value
is p: Born of	dbpedia:Amalia_Montero dbpedia:Carlos_Berlanga dbpedia:David_Díaz dbpedia:Juan%3F%40_Cameras dbpedia:Jose_Luis_País dbpedia:P%3F%40do_Domingo
is p: Country of	dbpedia:Tercer_%20Regimiento
is p: Died of	dbpedia:Blas_Casby dbpedia:Carlos_Berlanga dbpedia:Papa_Marcelo
p: GDP_PPP	1.029E+12 (units: Dollar)
p: GDP_PPP_per_capita	26300 (units: Dollar)
p: GDP_PPP_per_capita_rank	26 (units: Rank)
p: GDP_PPP_year	2005 (xsd:integer)
p: HDI	0.998 (xsd:decimal)
p: HDI_category	high
p: HDI_rank	19 (units: Rank)
p: HDI_year	2004 (xsd:integer)
is p: Host_city_of	dbpedia:1992_Summer_Olympics dbpedia:1997_European_Grand_Prix dbpedia:2006_Southwest_Forest_Park
is p: Location of	



Consultas sobre RDF
SPARQL

Los ficheros RDF pueden considerarse bases de datos de tripletas

SPARQL (Abril 2006) es un lenguaje de consulta para datos RDF

- Similar a SQL para RDF
- Lenguaje de consultas
 - Basado en RDQL
 - Modelo = patrones sobre grafos
- También describe un protocolo de transporte

SPARQL Sintaxis

URLs entre < >

<http://www.uniovi.es>

Prefijos para espacios de nombres

PREFIX x: <http://www.alumnos.org/>

x:profesor

Nodos anónimos

_:nombre ó **[]**

Literales entre “ ”

“Jose Emilio”

“234”^^xsd:integer

Variables empiezan por ?

?nombre

Comentarios empiezan por #

esto es un comentario

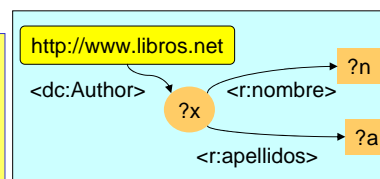
SPARQL Encaje de patrones

SPARQL se basa en encajar patrones de preguntas en el grafo

Ejemplo:

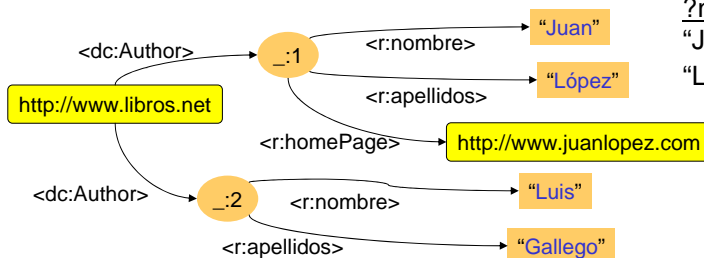
```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX r: <http://www.books.org/>

SELECT ?n ?a
WHERE { <http://www.libros.net> dc:Author ?x.
        ?x                      r:nombre ?n.
        ?x                      r:apellidos ?a. }
```



Resultados

?n	?a
“Juan”	“López”
“Luis”	“Gallego”



SPARQL *Filtros*

FILTER añade restricciones a los valores encajados

```
PREFIX p: <http://www.personas.org/>
SELECT ?n
WHERE { ?x p:nombre ?n .
        ?x p:edad ?e .
        FILTER (?e > 18) .
}
```

FILTER utiliza funciones y operadores de XPath 2.0

Tipos de datos: Boolean, Integer, Float, dateTime, etc.

Operadores habituales: >, <, >=, <=, =, !=

Otras expresiones: BOUND, isURI, isLiteral, isBlank, regex

SPARQL *Combinaciones de grafos*

UNION combina resultados de varios grafos

```
SELECT ?n
WHERE {
  { ?x foaf:name ?n }
  UNION { ?y p:nombre ?n }
}
```

OPTIONAL obtener valores aunque algunas tripletas no existan

```
SELECT ?n ?e
WHERE { ?person p:nombre ?n .
        OPTIONAL { ?person p:edad ?e }
}
```

SPARQL

Especificar conjuntos de entrada

FROM indica la URL de la que proceden los datos

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?n
FROM <http://www.di.uniovi.es/~labra/labraFoaf.rdf>
WHERE { ?x foaf:name ?n }
```

Pueden especificarse varios conjuntos de entrada

FROM NAMED permite asignar un nombre al grafo de entrada

GRAPH encaja contra un determinado nombre de grafo

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?n ?g
FROM NAMED <http://www.w3.org/People/Berners-Lee/card.rdf>
FROM NAMED <http://www.di.uniovi.es/~labra/labraFoaf.rdf>
WHERE
{ GRAPH ?g { ?x foaf:name ?n } }
```

SPARQL

Especificar formato de salida

SELECT devuelve las variables que encajan en el grafo

Los resultados pueden devolverse como una lista de enlaces en formato XML

CONSTRUCT genera un grafo RDF a partir de las variables

DESCRIBE devuelve una descripción del grafo en RDF

ASK devuelve sí o no

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX r: <http://relaciones.org/>

CONSTRUCT { ?x r:nombre ?n }
WHERE { ?x foaf:name ?n }
```


SPARQL *Limitaciones*

Lenguaje de consulta solamente

Propuestas para añadir "update", "delete", "insert", etc.

Resultados calculados

Ejemplo: SELECT (?x + ?y) WHERE ...

Funciones de agregación

Ejemplo: SELECT count(?x) WHERE...

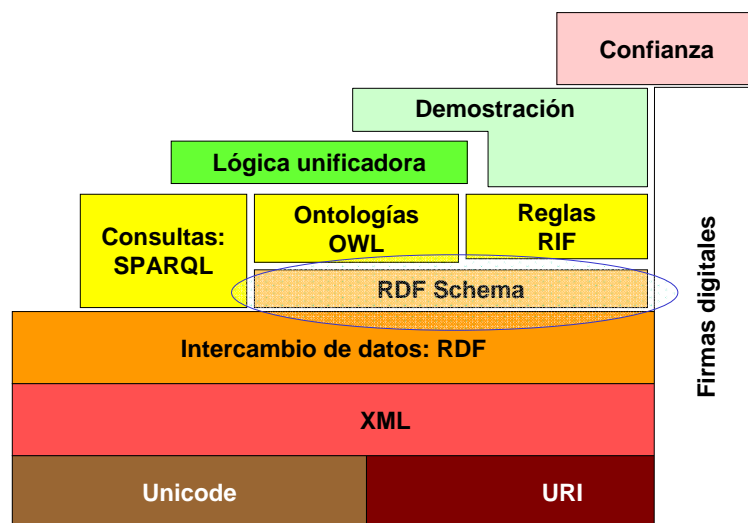
Operadores de búsqueda de texto

Soporte para colecciones

Caminos de longitud variable

Soporte para tipos de datos

RDF Schema



RDF Schema

Motivación

RDF permite establecer propiedades pero no dice nada acerca de las propiedades

RDF Schema tiene varias clases y propiedades predefinidas que permiten definir vocabularios.

Es permite definir:

- 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", ...

rdf:type indica que un individuo pertenece a una clase

rdfs:subClassOf indica que una clase está incluida en otra

Nota

rdf:type = `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>`

rdfs:Class = `<http://www.w3.org/2000/01/rdf-schema#Class>`

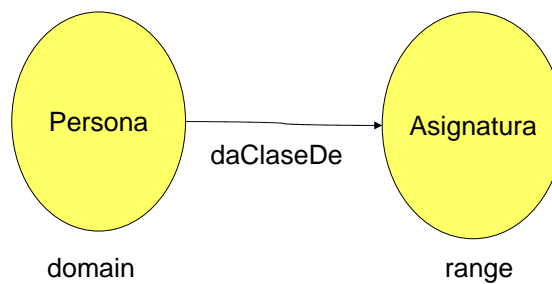
RDF Schema *Rango y Dominio*

Pueden declararse restricciones de Rango y Dominio

Ejemplo: **daClaseDe**

`rdfs:domain:` **Persona**

`rdfs:range:` **Asignatura**



RDF Schema *Jerarquías*

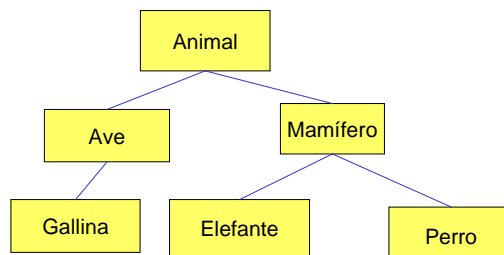
Las clases pueden organizarse en jerarquías

`rdfs: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



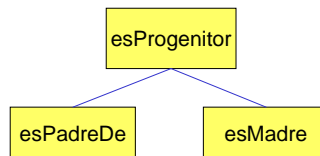
RDF Schema

Jerarquía de Propiedades

Jerarquías entre propiedades **subPropertyOf**

Ej. Ser padre es una subpropiedad de ser progenitor

P es una subpropiedad de Q sii si $P(x,y)$ entonces $Q(x,y)$



RDF Schema: Inferencias

RDF Schema tiene una semántica predefinida que permite inferir nuevas declaraciones a partir de las existentes

En realidad, se genera un grafo nuevo a partir del grafo anterior

Ejemplos:

$X \text{ rdf:type } A \wedge A \text{ subClassOf } B \rightarrow X \text{ rdf:type } B$

$A \text{ rdfs:subClassOf } B \wedge B \text{ rdfs:subClassOf } C \rightarrow A \text{ rdfs:subClassOf } C$

$P \text{ rdfs:domain } A \wedge X P Y \rightarrow X \text{ rdf:type } A$

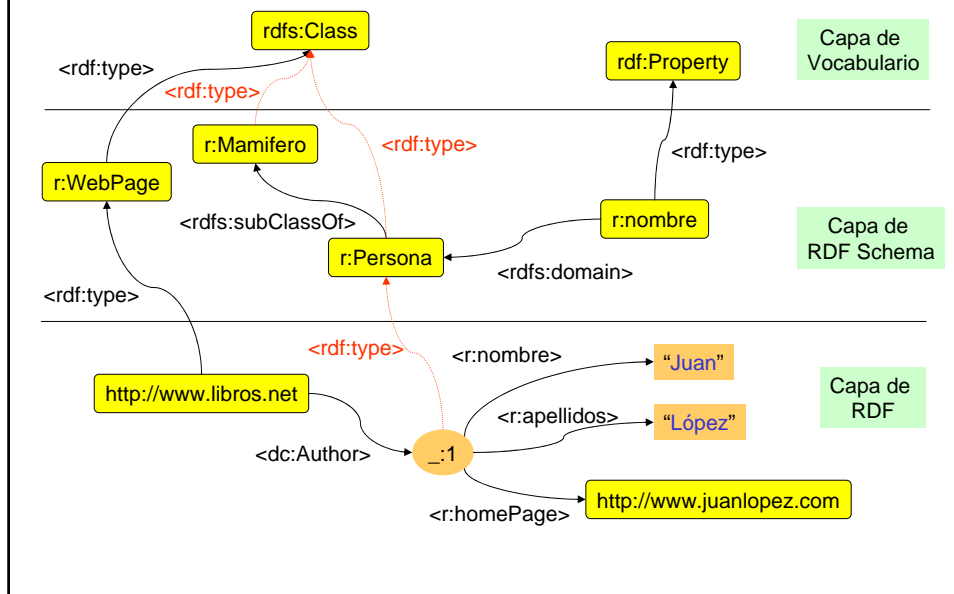
$P \text{ rdfs:range } B \wedge X P Y \rightarrow B \text{ rdf:type }$

$P \text{ rdfs:subPropertyOf } Q \wedge Q \text{ rdfs:subPropertyOf } R \rightarrow P \text{ rdfs:SubpropertyOf } R$

$P \text{ rdfs:subPropertyOf } Q \wedge X P Y \rightarrow X Q Y$

etc.

RDF Schema: Inferencias



Limitaciones de RDF Schema

RDF Schema sólo permite declarar información clases y propiedades

Es un primer paso hacia las ontologías pero se queda corto

Carece de expresividad para:

- Información negativa (ej. Los hombres no son mujeres)

- Cuantificadores (ej. Para que alguien sea considerado padre debe tener al menos un hijo)

- Cardinalidad (ej. Un buen estudiante tiene que tener aprobadas más de 3 asignaturas)

- No permite declarar atributos de propiedades (transitiva, simétrica, inversa, etc.)

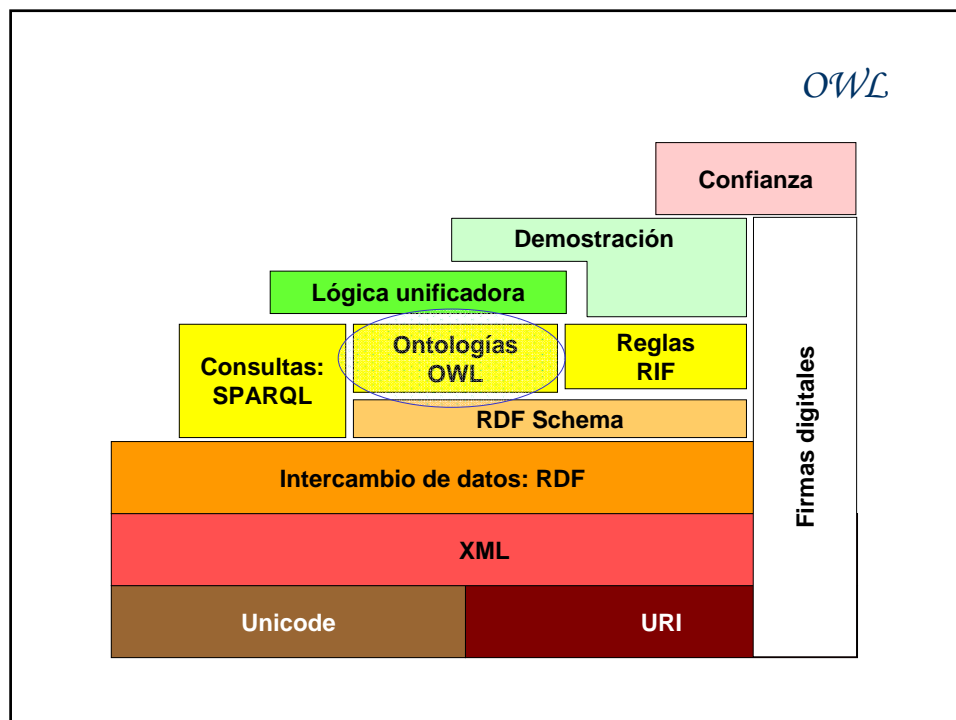
Problema de RDF Schema

RDF Schema es demasiado liberal permitiendo mezclar clases con individuos y propiedades

Por ejemplo:

x rdf:type x

Pueden llegar a producirse paradojas en RDF Schema



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.

Ejemplos de Ontologías

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 Dublín 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)

Profiles. Subconjuntos de OWL DL

OWL Lite, EL++, etc.

Se supone que son más eficientes, pero menos expresivos

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 lógica descriptiva es un subconjunto de la lógica de primer orden

Características:

Sólo se usan predicados de un argumento (clases) y dos argumentos (propiedades)

El uso de las variables está restringido

Lógica Descriptiva

La base de conocimiento contiene 2 niveles

Términos (TBox): Descripción de conceptos

$\text{Padre} \equiv \text{Persona} \cap \exists \text{ tieneHijo Persona}$

$\text{Orguloso} \equiv \text{Persona} \cap \exists \text{ tieneHijo ReciénNacido}$

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

Aserciones (ABox): Descripción de individuos

$\text{ReciénNacido}(\text{Sergio})$

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

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

Lógica descriptiva

Las expresiones en lógica descriptiva pueden representarse en lógica de primer orden

$\text{Padre} \equiv \text{Persona} \cap \exists \text{ tieneHijo Persona}$

$\forall x (\text{Padre}(x) \leftrightarrow (\text{Persona}(x) \wedge \exists y (\text{tieneHijo}(x,y) \wedge \text{Persona}(y))))$

$\text{Orguloso} \equiv \text{Persona} \cap \exists \text{ tieneHijo ReciénNacido}$

$\forall x (\text{Orguloso}(x) \leftrightarrow (\text{Persona}(x) \wedge \exists y (\text{tieneHijo}(x,y) \wedge \text{RecienNacido}(y))))$

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

$\forall x (\text{RecienNacido}(x) \rightarrow \text{Persona}(x))$

Lógica Descriptiva *Definición de Conceptos*

Definición de conceptos

Equivalencia: $C \equiv D$

Ejemplo: *Asturiano* \equiv *NacidoEnAsturias*

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

Ejemplo: *Asturiano* \subseteq *Español*

Intersección: $C \cap D$

Ejemplo: *Mujer* \equiv *Persona* \cap *Femenino*

Unión: $C \cup D$

Ejemplo: *Persona* \equiv *Hombre* \cup *Mujer*

Complemento: $\neg C$

Ejemplo: *Masculino* \equiv \neg *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: *Madre* \equiv *Mujer* $\cap \exists$ *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: *MadreFeliz* \equiv *Madre* $\cap \forall$ *tieneHijo Sano*

Una Madre es feliz si todos sus hijos están sanos

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

Lógica Descriptiva *Restricciones Cuantitativas*

Cardinalidad ($P = n$)

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

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

Cardinalidad máxima ($P \leq n$)

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

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

Cardinalidad mínima ($P \geq n$)

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

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

$$\begin{aligned} \forall x (\text{BuenEstudiante}(x) \leftrightarrow \text{Estudiante}(x) \wedge \\ \exists y_1 \exists y_2 \exists y_3 (\text{tieneAprobada}(x, y_1) \wedge \\ \text{tieneAprobada}(x, y_2) \wedge \\ \text{tieneAprobada}(x, y_3) \wedge \\ y_1 \neq y_2 \wedge y_2 \neq y_3 \wedge y_1 \neq y_3)) \end{aligned}$$

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**

Padre \equiv Persona $\cap \exists$ tieneHijo Persona Orgullosos \equiv Persona $\cap \exists$ tieneHijo ReciénNacido ReciénNacido \subseteq Persona Padre $\subseteq \neg$ ReciénNacido

ReciénNacido(Sergio) tieneHijo(Jose , Sergio) Persona(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 \equiv Persona $\cap \exists$ tieneHijo Persona
Orgullosa \equiv Persona $\cap \exists$ tieneHijo ReciénNacido
ReciénNacido \subseteq Persona
Padre $\subseteq \neg$ ReciénNacido

ReciénNacido(Sergio)
tieneHijo(José,Sergio)
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="http://www.uniovi.es/ontologia_1.1">
  <rdfs:comment>Ejemplo de Ontología</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.uniovi.es/ontologia_1.0"/>
  <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

2 tipos de propiedades

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

```
<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

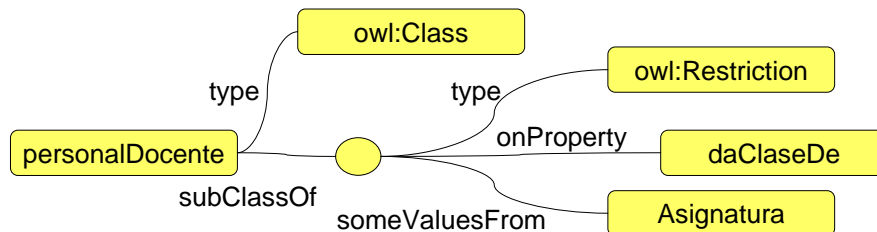
```
<owl:DatatypeProperty rdf:ID="edad">  
  <rdfs:range  
    rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>  
</owl:DatatypeProperty>
```


Definición de Clases

Clases como restricciones de propiedades

```
<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>
```

personalDocente $\subseteq \exists$ daClaseDe Asignatura



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>
```

OWL – Web semántica No asume nombres únicos

Web = modelo abierto

Información incompleta

2 URIs diferentes podrían identificar el mismo objeto

No soporta UNA (Unique name assumption)

Permite **inferir** que 2 elementos son iguales

No está pensado para validar modelos

Ejemplo

Persona \subseteq tienePadre = 1

tienePadre(luis,jose)
tienePadre(luis,pepe)
Persona(luis)

No indica error en el modelo

Infiere que “pepe” y “jose” son iguales

OWL – Web Semántica

Asumción de mundo abierto

Web = Sistema abierto

Sistemas tradicionales usaban *closed world assumption*

En OWL se usa *open world assumption*

Ejemplo

$\text{Soltero} \subseteq \neg \exists \text{ estaCasadoCon Persona}$
 $\text{Casado} \subseteq \exists \text{ estaCasadoCon Persona}$

Persona(pepe)
Persona(Maria)
Persona(luis)
estaCasadoCon(maria, pepe)
Casado(luis)

El sistema infiere que
María está casada

El sistema no infiere que
pepe esté casado ni soltero

El sistema infiere que luis
Está casado con alguien...
pero no sabe con quién.

OWL

Herramientas

Herramientas para manipulación de documentos OWL

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

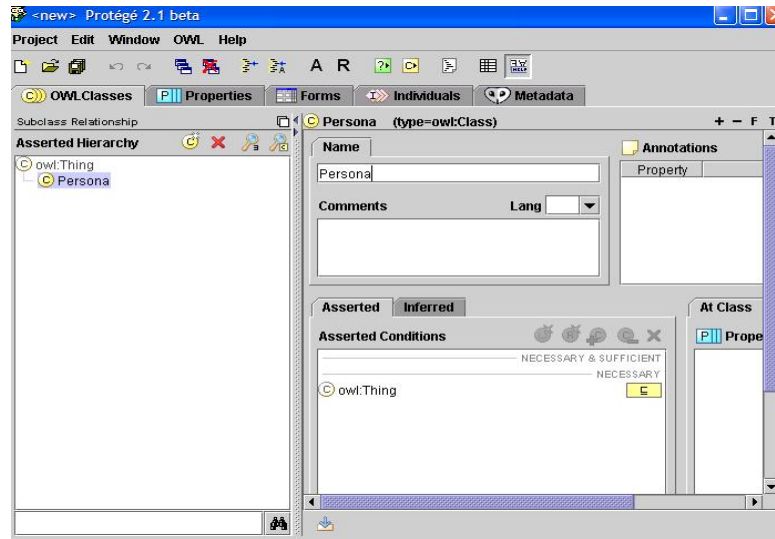
Arquitectura que facilita el desarrollo de plugins

Plugin para edición de documentos OWL

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

TopBraid (Comercial)

OWL Herramientas



OWL Sistemas de Inferencia

- Pellet (Java) incluye razonador para OWL
 - Soporte en línea de comandos o mediante interfaz DIG
- Fact++ (C++) razonador
- RACER. Sistema de inferencia implementado en Lisp
- JENA. API Java para RDF. Incluye sistema de inferencia
- CWM. Desarrollado por Tim Berners Lee en Python
 - Incluye sistema de inferencia
 - Sintaxis n3 y RDF

Ampliaciones OWL 2

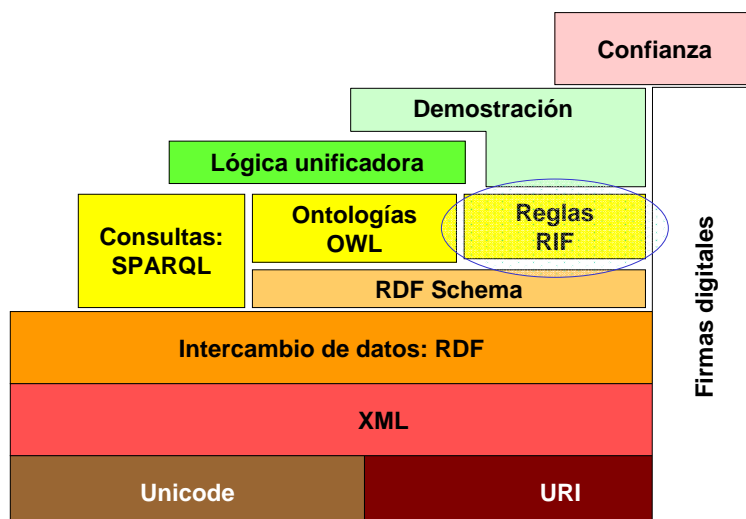
OWL 2 (En desarrollo)

Extensión de OWL 1.0 con más capacidad expresiva

Pasa de $\mathcal{SHOIN}(\mathcal{Dn})$ a $\mathcal{SROIQ}(\mathcal{Dn})$

- Propiedades disjuntas
hija, madre
- Propiedades reflexivas, irreflexivas, simétricas, antisimétricas
- Restricciones de cardinalidad cualificadas
 $\text{Influyente} \equiv \text{amigo} \geq 3 \text{ Político}$
- Axiomas de inclusión complejos
 $\text{dueño} \cdot \text{contiene} \subseteq \text{dueño} \ (x \text{ dueño } y, y \text{ contiene } z \rightarrow x \text{ dueño } z)$
- Tipos de datos definidos por el usuario

RIF: Reglas



Ampliaciones Reglas

Las Cláusulas Horn son la base de la programación lógica
SWRL (Semantic Web Rule Language) es una propuesta de creación de un lenguaje de reglas que añade reglas tipo Prolog a OWL

Orígenes: RuleML (Iniciativa internacional)

RIF (Rule Interchange Format) = Intercambio de reglas

$\text{hermano}(x,y) \wedge \text{progenitor}(y,z) \rightarrow \text{tío}(x,z)$

$\text{hermano} \cdot \text{progenitor} \subseteq \text{tío}$

Problema: Indecidible al unirse con OWL

Monotonidad y Reglas

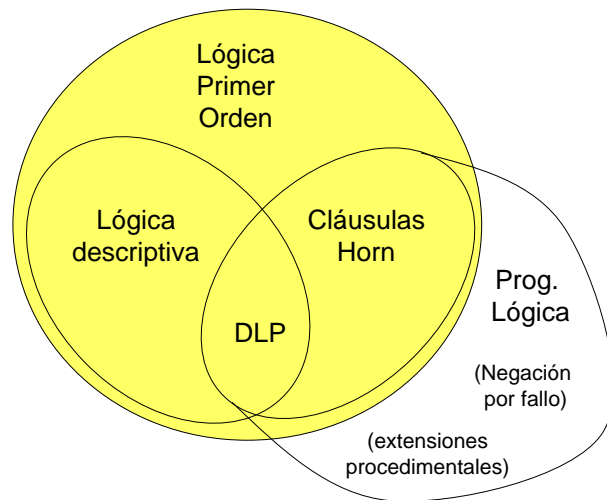
Problema: negación por fallo

Lógica de primer orden es monótona

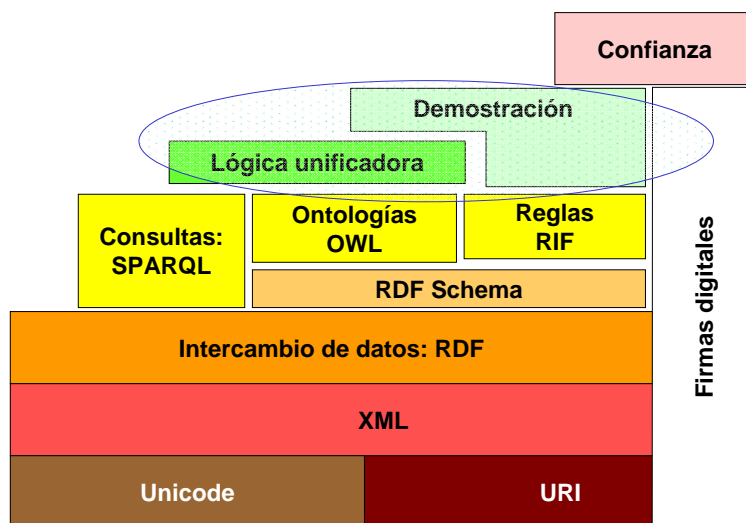
Negación por fallo no es monótona

Programación lógica con negación por fallo no es un subconjunto de lógica de primer orden

Interacción entre programación lógica y lógica descriptiva



Lógica y Demostración



Sistema de Inferencia Semantic Tableaux

Semantic Tableaux

Detecta si es insatisfacible

Va buscando modelos que cumplan las condiciones

Va generando un árbol de posibles modelos

Cierra las ramas cuando encuentra inconsistencias (clash)

Expresividad vs. Decidibilidad

Sopa de letras

<http://www.cs.man.ac.uk/~ezolin/logic/complexity.html>

Semantic Tableaux Ejemplo

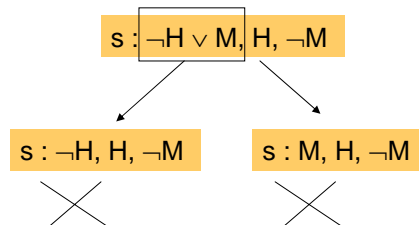
Hombre \subseteq Mortal

Hombre(Sócrates)

Mortal(Sócrates)

Razonamiento: $\{ H \subseteq M, H(s) \} \Rightarrow M(s)$

Forma normal: $\{ \neg H \vee M, H(s), \neg M(s) \}$



Regla de inferencia:

Si A contiene $x : C \vee D$ entonces

$A' = A \cup x : C$

$A' = A \cup x : D$

Semantic Tableaux
Algunas Reglas de inferencia

Si A contiene $x : C \vee D$ entonces

$$A' = A \cup x : C$$

$$A' = A \cup x : D$$

Si A contiene $x : C \wedge D$ entonces

$$A' = A \cup x : C, D$$

Si A contiene $x : \exists R C$ y $\nexists z$ tal que $R(x,z)$ y $z : C$ entonces

$$A' = A \cup \{ y : C, R(x,y) \} \text{ para un } y \notin A$$

Si A contiene $x : \forall R C$ y $R(x,y)$ pero no contiene $y : C$

$$A' = A \cup y : C$$

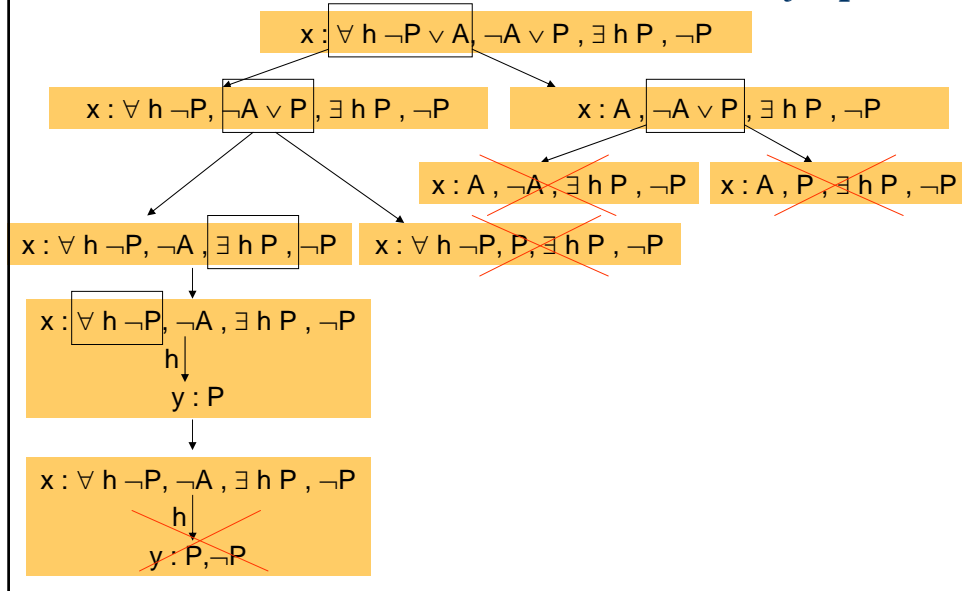
Semantic Tableaux
Otro ejemplo

$\{ \exists \text{ hijo Persona} \subseteq \text{Padre}, \text{Padre} \subseteq \text{Persona} \} \Rightarrow \exists \text{ hijo Persona} \subseteq \text{Persona}$

Cambiando nombres: $\{ \exists h P \subseteq A, A \subseteq P \} \Rightarrow \exists h P \subseteq P$

Forma normal: $\{ \forall h \neg P \vee A, \neg A \vee P, \exists h P, \neg P \}$

Semantic Tableaux
Otro ejemplo



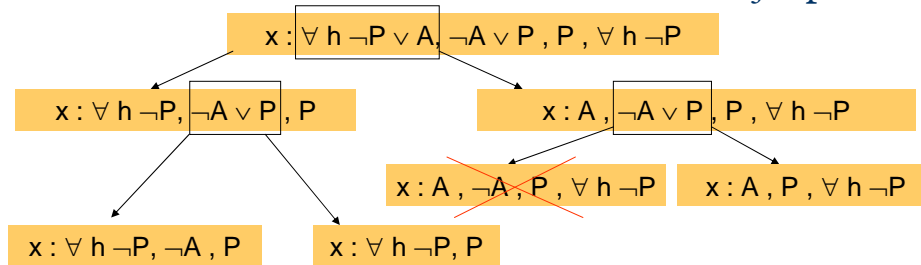
Semantic Tableaux
Otro ejemplo

$\{ \exists \text{ hijo Persona} \subseteq \text{Padre}, \text{Padre} \subseteq \text{Persona} \} \Rightarrow \text{Persona} \subseteq \exists \text{ hijo Persona}$

Cambiando nombres: $\{ \exists h P \subseteq A, A \subseteq P \} \Rightarrow P \subseteq \exists h P$

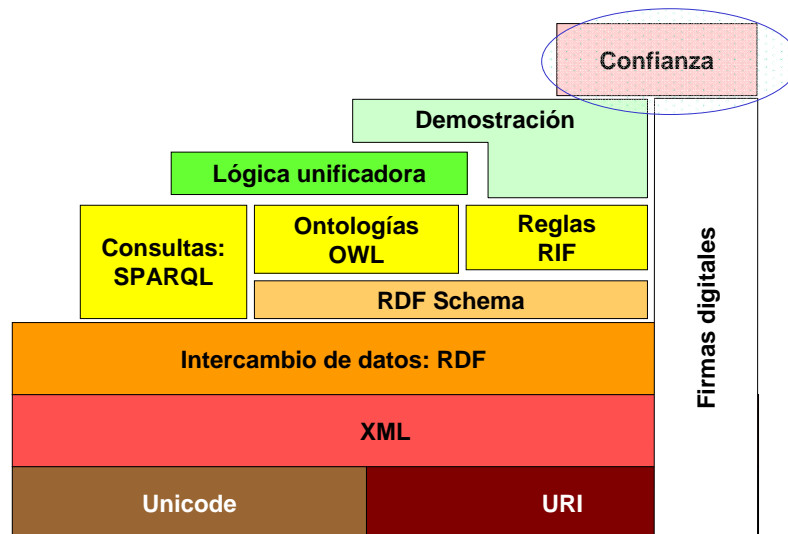
Forma normal: $\{ \forall h \neg P \vee A, \neg A \vee P, P, \forall h \neg P \}$

Semantic Tableaux Otro ejemplo



Se encuentra un modelo \Rightarrow No se cumple

Confianza



Confianza

Las declaraciones que aparecen en la Web pueden no ser fiables
Es necesario desarrollar sistemas que tengan en cuenta el contexto de las declaraciones

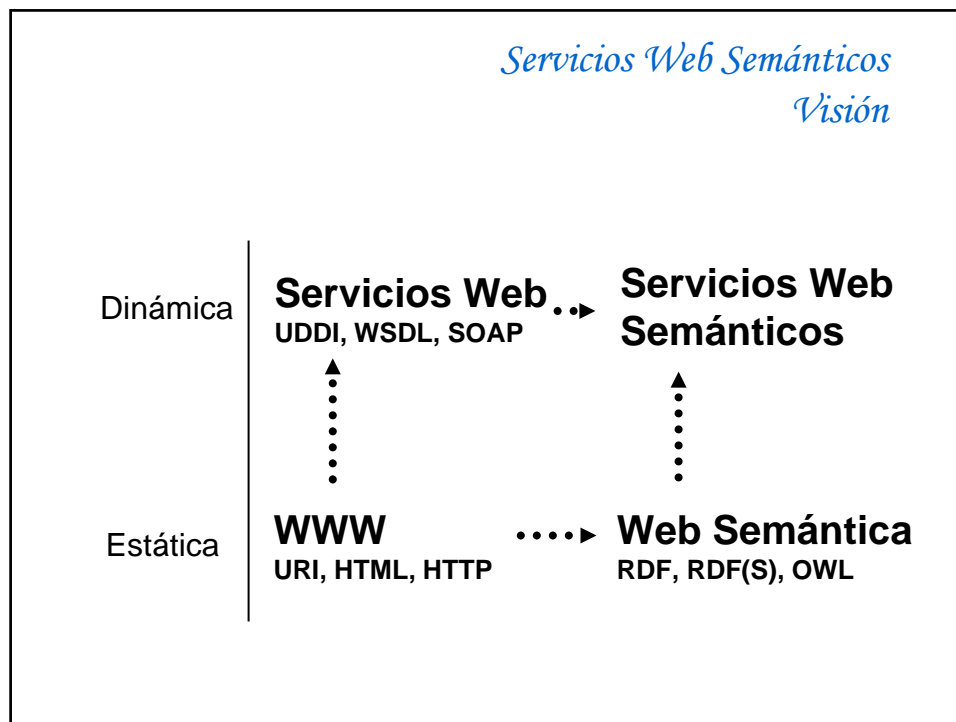
Problemas/Retos:

- ¿Es el autor quien dice que es?
- ¿Cómo se pueden comprobar sus credenciales?
- ¿Es posible confiar en el sistema de inferencia?

Algunas propuestas:

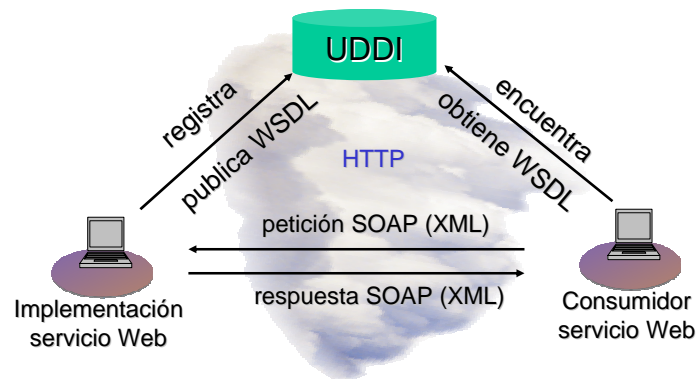
- Redes de confianza
- Reificación en RDF permite realizar afirmaciones sobre declaraciones RDF

Servicios Web Semánticos



- Servicios Web*
- Componentes débilmente acoplados y reutilizables
 - Contienen funcionalidad discreta
 - Distribuidos
 - Accesibles de forma automática mediante protocolos estándar

Servicios Web *Protocolos tradicionales*

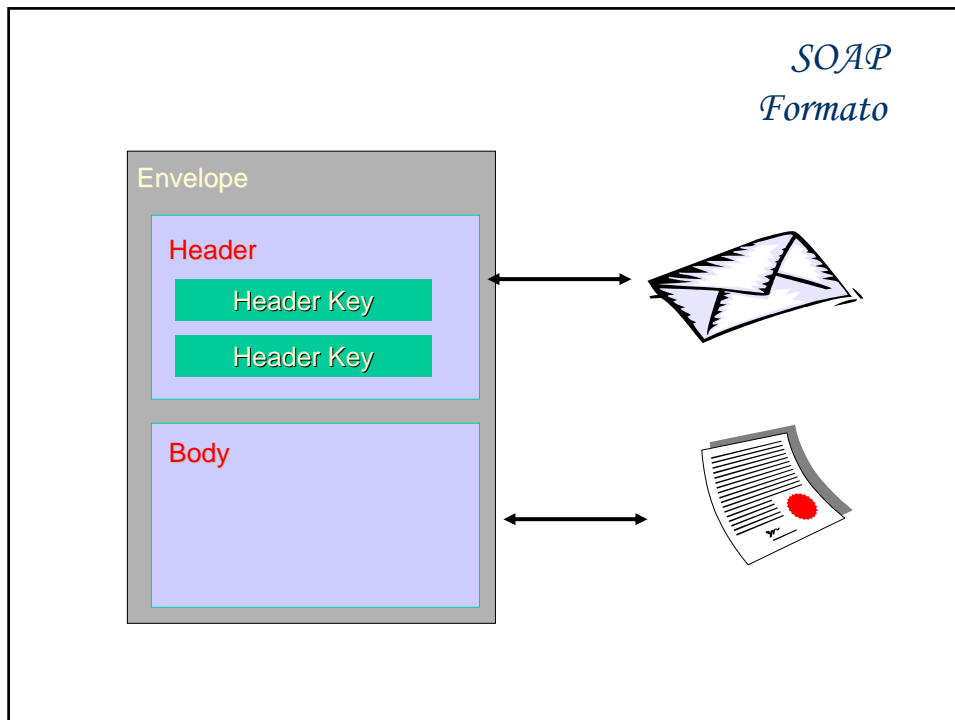


SOAP *Evolución*

SOAP: Define el formato de los mensajes
SOAP = Simple Object Access Protocol
Aunque tiene poco de objetos...

Evolución

- Desarrollado a partir de XML-RPC
- SOAP 1.0 (1999), 1.1 (2000), 1.2 (2002)
- Participación inicial de Microsoft
- Adopción posterior de IBM, Sun, etc.
- Aceptación industrial



*Servicios Web
WSDL*

WSDL (Web Services Description Language)

Describe:

- Qué puede hacer el servicio
- Dónde reside
- Cómo invocarlo

Vocabulario basado en capas

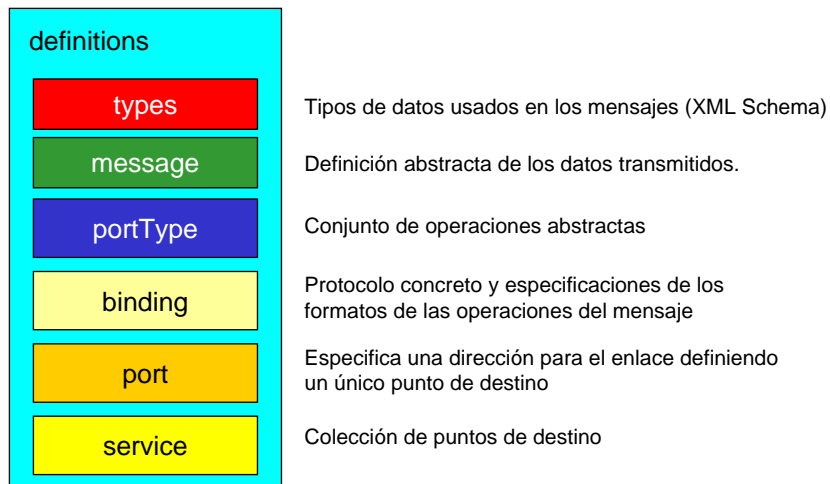
Es posible concentrarse en una capa cada vez

Evolución: Iniciativa conjunta de Ariba, IBM y Microsoft

(2001) Propuesto a W3C como recomendación (WSDL 1.1)

En desarrollo WSDL 2.0

Servicios Web *WSDL*



UDDI *Definición*

UDDI (Universal Discovery, Description and Integration)

Consorcio formado por IBM, Hp, Sun, Microsoft, Oracle, etc.

UDDI 1.0 (2000) Fundación del registro

UDDI 2.0 (2001) Alineación con estándares y taxonomía de servicios más flexible

UDDI 3.0 (2002) Interacción de implementaciones públicas y privadas

2 partes

Descripción de negocios

Páginas blancas (información de contacto)

“ amarillas (información de la industria)

“ verdes (información técnica y especificaciones)

Registro de servicios

Problemas Actuales de los Servicios Web

Descripciones puramente sintácticas

Uso e integración de servicios web debe
inspeccionarse manualmente

No se realiza un marcado semántico del contenido
de los servicios

No hay soporte para descubrimiento e invocación
automáticos

Servicios Web Semánticos

Web semántica

Permite interpretación de datos automática

Uso de ontologías como modelos de datos

Servicios Web

Descubrimiento, selección y composición
automáticos

Ejecución de Servicios a través de Web

Servicios Web Semánticos

Funcionalidades requeridas

Publicación: Ofrecer descripciones de las posibilidades y capacidades de un servicio

Descubrimiento: Localizar diferentes servicios apropiados para una determinada tarea

Seleccionar: Elegir los servicios más apropiados entre los disponibles

Composición: Combinar servicios para alcanzar un objetivo

Orquestación: Coordinación centralizada

Coreografía: Coordinación independiente

Mediación: Resolver problemas entre los servicios que se combinan (datos, protocolos, procesos...)

Ejecución: Invocar servicios siguiendo convenciones de forma automática

Monitorización: Controlar el proceso de ejecución

Compensación: Proporcionar soporte transaccional y deshacer o mitigar efectos no deseados

Substitución: Facilitar la substitución de servicios por otros equivalentes

Auditoría: Verificar que la ejecución de servicios se produjo de la forma deseada

Servicios Web Semánticos

2 Tecnologías principales

OWL-S

WSMO

Servicios Web Semánticos OWL-S

OWL-S: ontología para describir servicios Web

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

Basado en DAML-S (Darpa)

Utiliza OWL para definir una ontología de servicios Web

Objetivos:

Proporcionar descubrimiento de servicios Web basado en capacidades

Composición automática de servicios Web

Invocación automática de servicios Web

OWL-S pretende complementar protocolos ya existentes

Utiliza WSDL para invocación

Aumenta UDDI para descubrimiento

OWL-S Estructura General de la Ontología



WSMO

WSMO = Web Service Modeling Ontology (<http://www.wsmo.org>)

Iniciativa de grupo de trabajo de la Unión Europea

Modelo conceptual para servicios Web

- Ontología de los elementos básicos de servicios Web

- Lenguaje de descripción formal (WSML)

- Entorno de ejecución (WSMX)

Basado en WSMF (Web service Modeling Framework)

Resuelve problema de integración y mediación entre servicios Web

Servicios Web Semánticos Más información

Semantic Web Services Interest Group

- Grupo de interés creado en W3C (<http://www.w3.org/2002/ws/swsig/>)

SWSI: Iniciativa internacional para estandarizar servicios web semánticos (<http://www.swsi.org>)

- Incluye 2 comités

 - SWSL: Lenguaje para servicios web semánticos

 - SWSA: Requisitos de arquitectura de servicios web semánticos

Otros sistemas:

- IRS (Internet Reasoning Service)

- Meteor-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. McGuinness)

Algunos artículos ...

A tableaux decision procedure for SHOIQ, I. Horrocks, U. Sattler

Semantic tableaux para OWL 1.0

The even more irresistible *SROIQ* , I. Horrocks, O. Kutz, U. Sattler

Bases teóricas de OWL 1.1

Combining Rules and Ontologies: a survey, J. Maluszyński (ed)

Repasa diferentes propuestas para OWL + Reglas

Query answering for OWL DL with rules, B. Motik, U. Sattler, R. Studer

Explica indecidibilidad de OWL + Reglas

Extending the SHOIQ(D) Tableaux with DL-safe Rules, V. Kolovski,

B. Parsia, E. Sirin

Subconjunto de reglas que es decidible al juntarlo a OWL

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>

Fin de la Presentación

