

Web Semántica

Pedro J. Muñoz Merino
<http://www.it.uc3m.es/pedmume/>



Web Semántica: Introducción

- Concepto introducido por Tim Berners-Lee, el creador de la Web
- Web inteligente, donde los recursos tengan no sólo texto plano sino estén anotados con información semántica, relaciones entre ellos, relaciones de conceptos, etc.
- Las máquinas puedan razonar sobre la información y los recursos, en lugar de sólo realizar búsquedas planas
- Ejemplo: Reserva de un viaje en función de una serie de condiciones

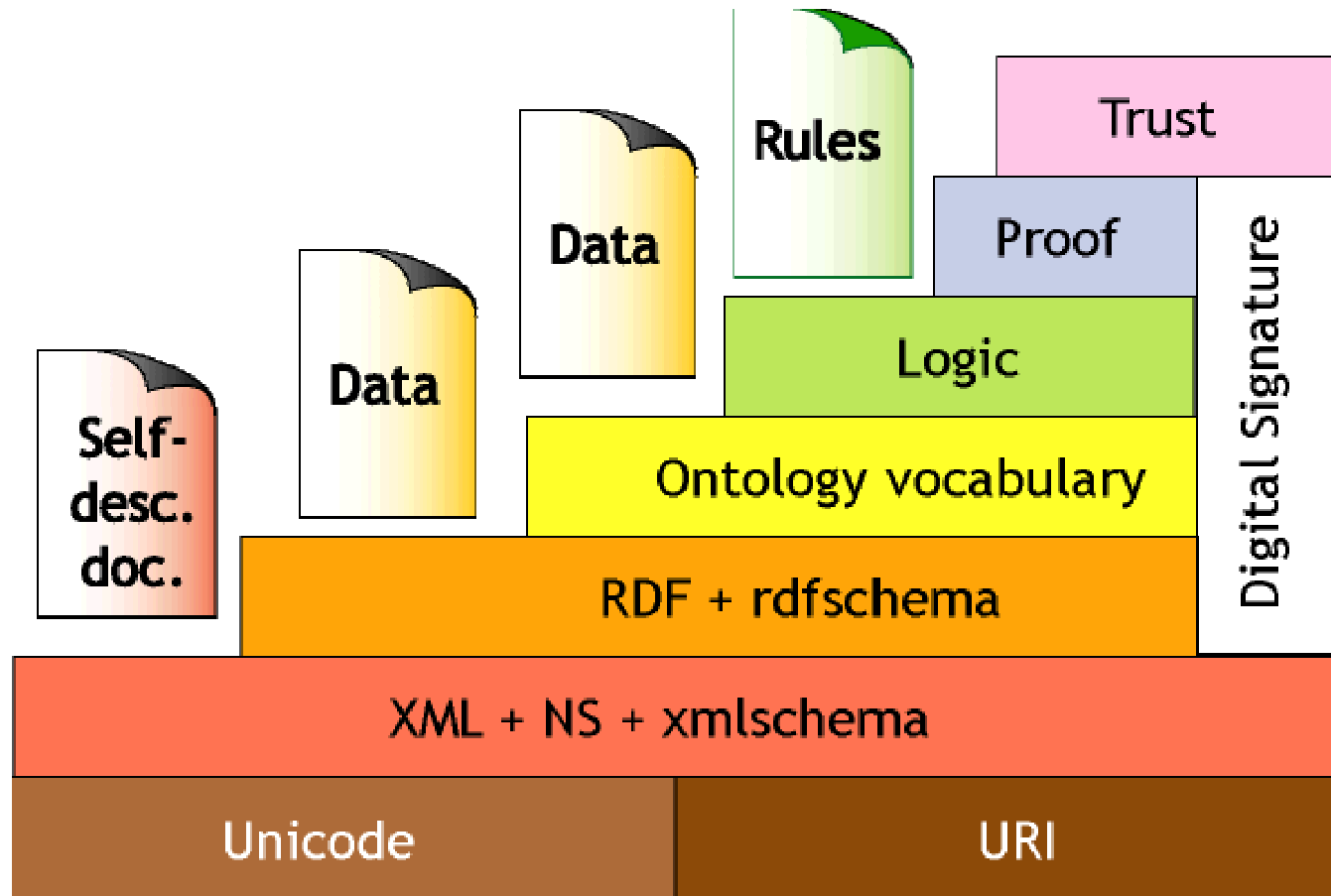
Web Semántica: Introducción

- Se describe la realidad como un conjunto de sentencias:

Sujeto Verbo Objeto

- Elementos fundamentales:
 - Ontologías
 - Anotaciones
 - Reglas
- Herramientas:
 - Lenguajes: N3, RDF/XL, Ntriples, etc.
 - Razonadores: CWM, etc.
 - Búsquedas en datos: SPARQL, etc.
 - Herramientas Autor: Protegé para ontologías, etc.

Web Semántica: Arquitectura



RDF: Anotaciones semánticas

- Cada recurso (sujeto u objeto) como verbo (o propiedad) es una URI (ejemplo URLs)
 - URL#nombre
- Una vez que se forman las diferentes sentencias, se produce un grafo entre ellas (imagen del W3C)



N3: Anotaciones semánticas

- `<#pedro> <#plays> <#paddle> .`
 - # representa el espacio de nombres actual, que tendrá una URI
 - En otro caso, habría que poner el espacio de nombres completo
- `<#pedro> <#origen> "Cuenca" .`
 - El objeto es lo único que puede ser una constante de un determinado tipo, en lugar de una URI
- Conjunto de sentencias combinando este tipo de sujeto, verbo y objeto

RDF: Anotaciones semánticas

- `<#pedro> <#plays> [<#uses> <#raquet>]`.
 - Nodos en blanco
- `@prefix st: <http://www.it.uc3m.es/pedmume/study/>`
- `@prefix dc: <http://purl.org/dc/elements/1.1/> .`
- `:pedro st:study [dc:subject st:maths]`.
 - Espacios de nombres: Se pone un “:” y a la izquierda va el identificador del espacio de nombres y a la derecha va el concepto definido en dicho espacio de nombres
 - Espacio de nombre actual: El relacionado con la URI del actual documento. Esas sentencias van sin nada delante

Ontologías

- Lenguajes para definir vocabularios (RDFS y OWL):
 - Definición de clases
 - Herencia entre clases
 - Objetos que pertenecen a una clase
 - Todos los verbos o propiedades de una clase
 - Los valores que se pueden alcanzar con cada propiedad y a quien se puede aplicar dicha propiedad
 - Decir que clases y verbos son diferentes y cuales iguales
 - etc., etc.

Definición y herencia de clases y propiedades

- Elementos relevantes:
 - Definición de clases: `rdfs:Class`
 - Relaciones de herencia entre clases: `rdfs:subClassOf`
 - Definición de propiedades: `rdf:Property`
 - Relaciones de herencia entre propiedades: `rdfs:subPropertyOf`
 - Instanciación de un objeto de la clase o de una propiedad: `rdf:type`

Caracterización de una propiedad

- Elementos relevantes:
 - Conjunto de clases posibles como sujeto de dicha propiedad: `rdfs:domain`
 - Conjunto de clases posibles como objeto de dicha propiedad: `rdfs:range`
 - `owl:equivalentProperty`
 - `owl:inverseOf`
 - `owl:FunctionalProperty`
 - `owl:InverseFunctionalProperty`
 - `owl:TransitiveProperty`
 - `owl:SymmetricProperty`
 - `owl:maxcardinality`
 - `owl:minCardinality`

Caracterización de clases

- Elementos relevantes:
 - owl:equivalentClass
 - owl:disjointWith
 - owl:intersectionOf
 - owl:unionOf
 - owl:complementOf

Caracterización de individuos

- Elementos relevantes:
 - owl:sameAs
 - owl:differentFrom

Propiedades de anotación

- Elementos relevantes:

- `rdfs:label`
- `rdfs:comment`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`

Reglas de razonamiento

- Forma de una regla:
 - { expresión1 } log:implies { expresión2 }
 - expresión1 y expresión2 suelen ser un conjunto de sentencias. Las sentencias van separadas con un punto y es como un AND lógico
 - Si se cumple la expresión1, entonces se realizarán las sentencias de expresión2
- Se pueden introducir variables, que luego se pueden utilizar en las expresiones:
 - @forAll :var1, :var2, ... :varn.
 - @forSome :var1, :var2, ... :varn.

Búsquedas con SPARQL

- SPARQL parecido a SQL, pero orientado a realizar peticiones sobre triplas
- Hay diferentes sentencias embebidas en las sentencias SPARQL
- <http://jena.sourceforge.net/ARQ/Tutorial/>

Ontologías: Ejemplo daml-ex Parentesco

```
Woman rdf:type rdfs:Class;  
      rdfs:subClassOf :Female, :Person .
```

```
:ancestor rdf:type owl:TransitiveProperty;  
          rdfs:label "ancestor" .
```

```
:child rdf:type rdfs:Property;  
       owl:inverseOf :parent .
```

```
:descendant rdf:type owl:TransitiveProperty .
```

```
:father rdf:type rdfs:Property;  
        owl:cardinality "1";  
        rdfs:range :Man;  
        rdfs:subProperty :parent .
```

Basado en el ejemplo “Family Relationships – GEDCOM” que se encuentra en <http://www.w3.org/2000/10/swap/Examples.html>

Anotaciones: Ejemplo Parentesco



Goedele gc:childIn :dt; :gender :F.
:Veerle gc:childIn :dt; :gender :F.
:Nele gc:childIn :dt; :gender :F.
:Karel gc:childIn :dt; :gender :M.
:Maaïke gc:spouseIn :dt; :gender :F.
:Jos gc:spouseIn :dt; :gender :M.

:Jos gc:childIn :dp; :gender :M.
:Rita gc:childIn :dp; :gender :F.
:Geert gc:childIn :dp; :gender :M.
:Caroline gc:childIn :dp; :gender :F.

Basado en el ejemplo “Family Relationships – GEDCOM” que se encuentra en <http://www.w3.org/2000/10/swap/Examples.html>

Reglas: Ejemplo Parentesco



If a child is a child in a family and a parent is a spouse in the # family, then the parent is a parent of the child.“

@forAll :child, :family, :parent. { :child gc:childIn :family. :parent gc:spouseIn :family} log:implies { :child gc:parent :parent}.

"If a child has parent and the parent is male, then then the
child's father is the parent.“

**@forAll :child, :parent. { :child gc:parent :parent. :parent :gender :M}
log:implies { :child gc:father :parent} .**

Basado en el ejemplo “Family Relationships – GEDCOM” que se encuentra en
<http://www.w3.org/2000/10/swap/Examples.html>

Razonador CWM: Ejemplo Parentesco



```
Goedele    :gender :F;  
           gc:ancestor :Frans,  
             :Jos,  
             :Maaike,  
             :Maria;  
           gc:aunt :Caroline,  
             :Christine,  
             :Greta,  
             :Karolien,  
             :Rita;  
           gc:brother :Karel;  
           gc:childIn :dt;  
           gc:cousin :Ann,  
             :Ann_Sophie,
```

Basado en el ejemplo “Family
Relationships – GEDCOM” que se
encuentra en
<http://www.w3.org/2000/10/swap/Examples.html>

Jena

- Entorno de Web Semántica en Java
 - API de RDF
 - API de OWL
 - Leer ficheros RDF y guardarlos en memoria
 - Acceso a Bases de Datos semánticas: SDB (soporta SPARQL) y TDB
- API RDF de Jena
 - http://jena.sourceforge.net/tutorial/RDF_API/index.html

Entorno de Jena: Importación de librerías

```
import com.hp.hpl.jena.rdf.model.*;
```

```
import com.hp.hpl.jena.sdb.SDBFactory;
```

```
import com.hp.hpl.jena.sdb.Store;
```

```
import com.hp.hpl.jena.sdb.store.DatasetStore;
```

```
/* Package for the execution of SPARQL queries
```

```
*/
```

```
import com.hp.hpl.jena.query.*;
```

Ejecución de SPARQL en Jena

```
Query query = QueryFactory.create(queryString);
//Connect to the database
Store store = SDBFactory.connectStore("sdb.ttl") ;
Dataset ds = SDBFactory.connectDataset(store) ;
// Execute the query and obtain results
QueryExecution qe = QueryExecutionFactory.create(query, ds);
ResultSet results = qe.execSelect();

while (results.hasNext()){
    results.nextSolution();
}
int countevents = results.getRowNumber();
qe.close();
store.getConnection().close() ;
store.close() ;
return countevents;
```

Trabajo propio del alumno asociado a la sesión

- Tutorial de Web Semántica utilizando Notation 3:
<http://www.w3.org/2000/10/swap/Primer>,
<http://www.w3.org/2000/10/swap/doc/formats>,
<http://www.w3.org/2000/10/swap/doc/ontologies>,
<http://www.w3.org/2000/10/swap/doc/Rules>,
<http://www.w3.org/2000/10/swap/doc/Processing>
- Tutorial básico de Web semántica:
<http://www.linkeddatatools.com/semantic-web-basics>
- Tutorial de SPARQL:
<http://jena.sourceforge.net/ARQ/Tutorial/>
- API RDF de Jena:
http://jena.sourceforge.net/tutorial/RDF_API/index.html

Referencias Extra

- RDF por el W3C, <http://www.w3.org/RDF/>
- Tutorial de N3 completo, <http://www.w3.org/2000/10/swap/doc/>
- RDFS por el W3C, <http://www.w3.org/TR/rdf-schema/>
- OWL por el W3C, <http://www.w3.org/2004/OWL/>
- Jena, <http://jena.sourceforge.net/>
- SPARQL del W3C, <http://www.w3.org/TR/rdf-sparql-query/>
- Protegé (editor de ontologías), <http://protege.stanford.edu/>