Ingeniería de Aplicaciones para la Web Semántica

Clase 06

RDF y la web semántica

Mg. A. G. Stankevicius

Segundo Cuatrimestre

2005

Copyright

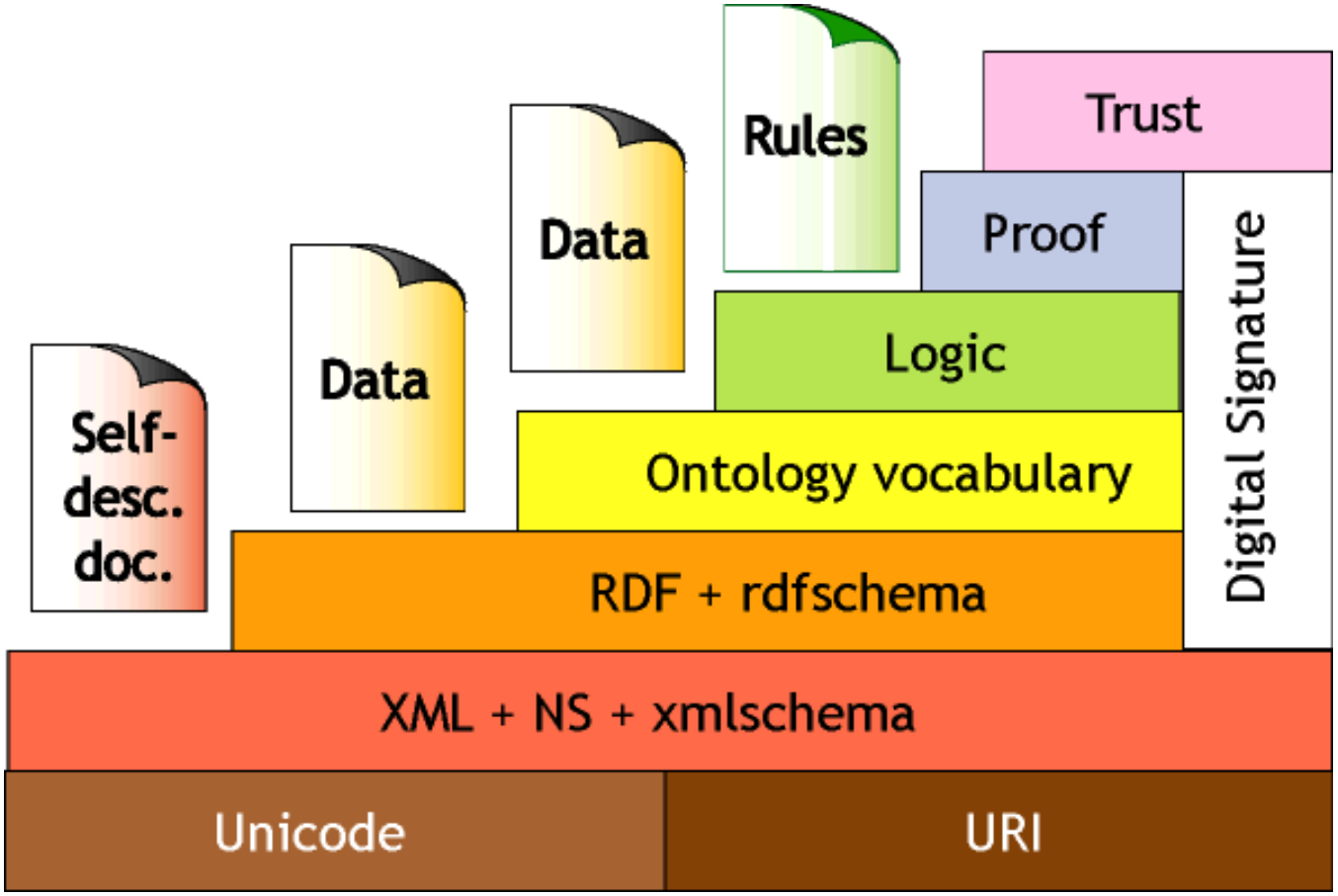
Contenidos

- El camino hacia la web semántica.
- Componentes de un documento RDF.
- Un documento, tres visiones.
- Reificación (cosificación).
- Una crítica constructiva a RDF.
- Sintaxis XML para RDF en detalle.

El camino hacia la web semántica

- Tanto el desarrollo como la puesta en funcionamiento de la web semántica adopta un esquema por capas.
- Cada capa se construyen en base a la anterior, respetando que:
  - Se preserve la compatibilidad con respecto a las capas inferiores.
  - Se mantenga de ser posible la comprensión parcial del contenido de las capas superiores.

Implementación basada en capas de la web semántica



Capas de la web semántica

- Capa XML:
  - Base sintáctica de la web semántica.
- Capa RDF:
  - Lenguaje RDF para modelar los aspectos más simples del dominio considerado.
  - Lenguaje RDFS para describir ontologías básicas.
- Capa de Ontología:
  - Más poder expresivo para describir ontologías (estandard actual: OWL).

Capas de la web semántica

- Capa Lógica:
  - Permite refinar las ontologías disponibles.
  - Admite representar conocimiento específico al dominio en consideración.
- Capa de Justificación:
  - Generación y verificación de las inferencias obtenidas en base a la capa lógica.
- Capa de Confianza:
  - Firma y certificados digitales.
  - Compilación de rankings de servicios, etc.

Cuestionamientos al XML

- XML sin duda provee un marco uniforme en el cual diversas aplicaciones pueden intercambiar datos y metadatos.
- Sin embargo, XML no provee mecanismo alguno para hacer referencia al significado de esos datos y metadatos.
  - Ejemplo: ¿cuál es el significado del anidamiento de tal o cual elemento XML?
- Las aplicaciones deben dotar de significado al documento XML.

Anidamiento de tags XML

- Suponiendo que Guillermo dicta IAWS, los siguientes anidamientos alternativos resultan enteramente razonables:

```
<course name="IAWS">
   <lecturer>Guillermo</lecturer>
</course>

<lecturer name="Guillermo">
   <teaches>IAWS</teaches>
</lecturer>
```

El lenguaje RDF

- RDF = *Resource Description Framework*.
- Los documentos RDF se componen esencialmente de un conjunto de ternas objeto-atributo-valor.
- Tanto estas ternas como el documento RDF se pueden formular como un documento XML convencional.
  - RDF hereda todos los beneficios de XML.
  - Es posible utilizar otras codificaciones alternativas orientadas a los humanos.

El lenguaje RDF

- Al estar basado en XML se respetan los dos objetivos del diseño por capas:
  - Compatibilidad hacia abajo.
  - Entendimiento parcial hacia arriba.
- Los principales conceptos asociados a los documentos RDF son:
  - Los recursos.
  - Las propiedades.
  - Las declaraciones.

Los recursos

- Los recursos son aquellos objetos o cosas de las cuales deseamos hablar.
  - Por caso, los cursos, profesores, alumnos, aulas, etc.
- Cada recurso se asocia a un URI (Universal Resource Identifier).
- Los URIs puede ser:
  - URLs (direcciones de la web).
  - Cualquier otro tipo de identificadores unívocos.

Las propiedades

- Las propiedades son, en cierto sentido, un tipo especial de recurso.
- Capturan relaciones entre recursos.
  - Por caso, “dicta”, “participa-en”, “edad”, etc.
- Las propiedades también son identificadas mediante URIs:
  - Los URIs proveen una nomenclatura global.
  - Simplifica el problema que los homónimos introducen en la representación de conocimiento distribuido.

Las declaraciones

- Las declaraciones especifican las propiedades de los recursos.
- Una declaración es simplemente *una terna* objeto-propiedad-valor.
- Los valores pueden ser otros recursos, o bien simplemente valores literales.
- Estos valores literales deben ser atómicos, por ejemplo:
  - Una cadena de caracteres.
  - Un valor numérico.

Un documento, tres visiones

- Las declaraciones puede considerarse como:
  - Una terna.
  - Una porción de un grafo.
  - Una porción de un documento XML.
- Por ende, un documento RDF puede verse como:
  - Un conjunto de ternas.
  - Un grafo (denominado *red semántica*).
  - Un documento XML.

Visión como ternas

(“Guillermo R. Simari”,
http://micasita.com.ar/responsable-de,
http://cs.uns.edu.ar/materias/IAWS)

- En general, toda terna *(x, P, y)* puede ser reinterpretada de forma directa como una fórmula lógica *P(x,y)*.
  - Esta fórmula se interpreta como que la relación *P* abarca a los objetos *x* e *y*.
- RDF sólo contempla predicados binarios.

Visión como red semántica

- Grafo dirigido con nodos y arcos etiquetados, donde:
  - Los arcos se originan en los recursos,
  - ...y arriban a los valores.
- Cuando el valor asociado a un recurso sea otro recuros, las burbujas se terminan encadenando entre sí.

Visión como documento XML

- Los grafos son potentes herramientas para facilitar el entendiento por parte de las personas... *pero*,
- La web semántica tiene por objeto facilitar el acceso y el procesamiento automático de la información.
- Por ende, existe una tercer visión en términos de documentos XML:
  - No obstante, el lenguaje XML *no forma parte* de la especificación del documento RDF.

Visión como documento XML

- Codificación en términos de XML:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:miNS="http://micasita.com.ar/ns#">
    <rdf:Description
        rdf:about="http://cs.uns.edu.ar/materias/IAWS">
        <miNS:responsable-de>
            Guillermo R. Simari
        </miNS:responsable-de>
    </rdf:Description>
</rdf:RDF>
```

Documentos RDF expresados en lenguaje XML

- Los documentos RDF se codifican en XML mediante el tag `rdf:RDF`.
- Anidados en ese tag aparecen un conjunto de descripciones formuladas a través del tag `rdf:Description`.
- Las descripciones hacen declaraciones acerca recursos, los cuales usan:
  - El atributo `rdf:about`, en caso de existir.
  - El atributo `rdf:ID`, en caso de crear uno nuevo.

Volviendo al ejemplo anterior...

- El elemento `rdf:Description` está haciendo una declaración acerca del recurso `http://cs.uns.edu.ar/materias/IAWS`.
- Dentro de esta descripción, podemos observar en concreto que:
  - La propiedad en cuestión es definida en términos de un nuevo tag XML.
  - El contenido de ese tag es el valor asociado a la propiedad anterior.

Reificación en acción

- RDF permite hacer declaraciones acerca de otras declaraciones:
  - AGS asegura que GRS es el responsable de la materia IAWS.
- Este tipo de declaraciones permiten modelar creencias o certezas acerca de otras declaraciones.
- Para esto, es necesario que cada declaración pueda ser referenciada unívocamente.

Reificación (cosificación)

- Solución: introducir un recurso auxiliar, el cual será asociado a las tres partes de la declaración en cuestión.
- Por caso, en el ejemplo anterior, sea `belief1` un recurso auxiliar con las siguiente propiedades:
  - `subject` asociada al recurso “GRS”.
  - `predicate` asociada al recurso “responsable-de” .
  - `object` al recurso “IAWS”.

Tipado de datos en RDF

- Los valores literales pueden conformar a un determinado tipo de dato:

  (“Alejandro G. Stankevicius”,
  http://micasita.com.ar/miNS#edad,
  “30”^^http://www.w3.org/2001/XMLSchema#integer)

- ^^ denota el tipo asociado a un literal.
- En la práctica se adopta a los esquemas XML como mecanismo de tipado.
  - No obstante, se pueden utilizar cualquier otra definición externa de tipado.

Críticas constructivas a RDF

- ¡RDF sólo utiliza predicados binarios!
  - Solemos utilizar predicados de mayor aridad.
  - Pero, los predicados binarios pueden simular predicados de mayor aridad.
- Por caso, para simular el predicado `debe(Argentina, FMI, Monto)`, podemos crear el objeto auxiliar deuda, el cual:
  - Se asocia mediante `deudor` a `Argentina`.
  - Se asocia mediante `acreedor` a `FMI`.
  - Se asocia mediante `importe` a `Monto`.

Críticas constructivas a RDF

- El mecanismo de reificación es extremadamente poderoso.
- ¿No será mucho para un lenguaje tan simple como RDF?
  - Puede introducir un nivel de complejidad excesivo, quizás innecesario en esta capa inicial de la web semántica.
- Una posible alternativa hubiera sido contemplar la reificación en capas superiores (*e.g.*, en la capa lógica).

Críticas constructivas a RDF

- RDF puede no ser el mejor lenguaje de modelado, pero...
  - ¡Ya se trata de un estandar de facto!
- Cuenta con el suficiente poder expresivo, al menos para construir otras capas más significativas por encima.
- La principal ventaja es que la todo documento RDF se mapea de manera unívoca a un determinado modelo.

Sintáxis basada en XML

- De las tres visiones posibles para un documento RDF, la que posiblemente requiera mayor empeño es aquella que resulta más alejada a los humanos:La sintáxis para RDF basada en XML.
- A continuación repasaremos algunos aspectos adicionales en relación a esta representación en particular para los documentos RDF.

Un ejemplo más complejo

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XLMSchema#"
    xmlns:uns="http://cs.uns.edu.ar/ns#">
    <rdf:Description rdf:about="#1234">
        <uns:name>Juan Perez</uns:name>
        <uns:title>Profesor Adjunto</uns:title>
        <uns:age rdf:datatype="&xsd:integer">
            36
        </uns:age>
    </rdf:Description>
```

Un ejemplo más complejo

```
<rdf:Description rdf:about="#CS101">
    <uns:course-name>RPA</uns:name>
    <uns:taught-by>Juan Perez</uns:taught-by>
</rdf:Description>

<rdf:Description rdf:about="#CS102">
    <uns:course-name>EP</uns:name>
    <uns:taught-by>Mengue Chón</uns:taught-by>
</rdf:Description>

</rdf:RDF>
```

rdf:about vs. rdf:ID

- El elemento XML `rdf:Description` cuenta con los siguientes atributos:
  - `rdf:about`, cuando se haga mención a recursos definidos en otra parte.
  - `rdf:ID`, cuando se haga mención a un recurso que está siendo definido en ese momento.
- No se distigue la declaración de un recurso de su referencia (pueden aparecer en cualquier orden).

Estructura del elemento XML rdf:Description

- Al considerar el contenido del siguiente elemento `rdf:Description`:

```
<rdf:Description rdf:about="#CS101">
    <uns:course-name>RPA</uns:name>
    <uns:taught-by>Juan Perez</uns:taught-by>
</rdf:Description>
```

- ...se puede apreciar que `<uns:course-name>` y `<uns:taught-by>` definen sendos valores que deben asociarse al curso `CS101`.
  - La lectura debe ser hecha de manera conjuntiva.

Tipado de datos

- El atributo rdf:datatype=”&xsd:integer” indica el tipo de dato asociado a la propiedad edad:

```
<rdf:Description rdf:about="#1234">
    <uns:name>Juan Perez</uns:name>
    <uns:title>Profesor Adjunto</uns:title>
    <uns:age rdf:datatype="&xsd:integer">
        36
    </uns:age>
</rdf:Description>
```

Tipado de datos

- En este caso, se ha especificado que la propiedad `edad` tiene por dominio a "`&xsd:integer`".
  - Cada vez que sea utilizada esta propiedad se deberá especificar el tipo delvalor asociado.
  - La idea es asegurar que quien procese el documento RDF pueda asignar el tipo correcto a las propiedades, aún sin tener que haber accedido a la definición asociada.
  - ¡Este es un escenario altamente factible en el marco de la web!

El atributo rdf:resource

- En el ejemplo anterior la conexión entre cursos y profesores es implícita.
  - Se puede observar que ambos referencian por su nombre a la misma persona.
- El uso de la misma cadena de caracteres pudo haber sido mera casualidad.
- Para explicitar formalmente la conexión entre ambos recursos, podemos hacer uso del atributo `rdf:resource`.

El atributo rdf:resource

- Explicitando la conexión entre cursos y profesores en el ejemplo anterior:

```
<rdf:Description rdf:about="#1234">
   <uns:name>Juan Perez</uns:name>
   <uns:title>Profesor Adjunto</uns:title>
   <uns:age rdf:datatype="&xsd:integer">
      36
   </uns:age>
</rdf:Description>
<rdf:Description rdf:about="#CS101">
   <uns:course-name>RPA</uns:name>
   <uns:taught-by rdf:resource=”1234”/>
</rdf:Description>
```

Haciendo referencia definiciones externas

- Es posible hacer referencia a recursos definidos externamente mediante la sintaxis "`http://foo.bar/ns#resource`" como valor de la propiedad `rdf:about`.
  - `http://foo.bar/ns` es el URI en donde se puede encontrar la definición del recurso `resource`.
- Sólo las descripciones conteniendo el atributo `ID` puede ser referenciadas de esta forma.

Descripciones anidadas

- En RDF es posible anidar las descripciones:

```
<rdf:Description rdf:about="#CS101">
   <uns:course-name>RPA</uns:course-name>
   <uns:taught-by>
      <rdf:Description rdf:about="#1234">
      <uns:name>Juan Perez</uns:name>
      <uns:title>Profesor Adjunto</uns:title>
      </rdf:Description></uns:taught-by>
</rdf:Description>
```

- A pesar del anidamiento, el alcance sigue siendo global.

Incorporando estructura a los documentos RDF

- El elemento `rdf:type` permite explicitar qué restricción de estructura sigue un determinado recurso RDF:

```
<rdf:Description rdf:about="#CS101">
    <rdf:type rdf:resource="http://cs.uns.edu.ar/ns#course"/>
    <uns:course-name>RPA</uns:course-name>
    <uns:taught-by>
        <rdf:Description rdf:about="#1234">
        <rdf:type rdf:resource="http://cs.uns.edu.ar/ns#teacher"/>
        <uns:name>Juan Perez</uns:name>
        <uns:title>Profesor Adjunto</uns:title>
        </rdf:Description></uns:taught-by>
</rdf:Description>
```

Convenciones de notación

- Reglas de simplificación de la notación:
  1) Las propiedades expresadas como elementos sin componentes anidados puede ser reemplazados por atributos XML.
  2) Las descripciones que hagan referencia a su estructura pueden usar el nombre especificado en el elemento rdf:type en vez del tag rdf:Description.
- Estas reglas crean variantes sintácticas de un mismo modelo de datos, aquel unívocamente asociado al documento.

Primer regla en acción

- Los elementos <uns:course-name>, <uns:name> y <uns:title> se reescriben como atributos:

```
<rdf:Description rdf:about="#CS101"
  uns:course-name="RPA">
    <rdf:type rdf:resource="http://cs.uns.edu.ar/ns#course"/>
    <uns:taught-by>
        <rdf:Description rdf:about="#1234"
          uns:name="Juan Perez"
          uns:title="Profesor Adjunto">
        <rdf:type rdf:resource="http://cs.uns.edu.ar/ns#teacher"/>
        </rdf:Description></uns:taught-by>
</rdf:Description>
```

Segunda regla en acción

- El elemento `rdf:type` permite explicitar qué restricción de estructura sigue un cierto recurso RDF:

```
<uns:course rdf:about="#CS101"
  uns:course-name=”RPA”>
    <uns:taught-by>
        <uns:teacher rdf:about="#1234"
          uns:name=”Juan Perez”
          uns:title=”Profesor Adjunto”/>
    </uns:taugh-by>
</uns:course>
```

Elementos contenedores

- Los elementos contenedores permiten hacer declaraciones acerca de una agrupación de recursos o de atributos.
  - Por caso, “todos los cursos dictados por un cierto profesor”.
- El contenido del contenedor se puede acceder mediante los nombres `rdf:_1`, `rdf:_2`, etc.
  - En caso de no quere especificar el orden, se puede usar la notación alternativa `rdf:li`.

Tipos de contenedores

- Existen tres tipos de contenedores:
  - `rdf:Bag`: que no especifica orden alguno entre sus elementos y que puede contener elementos repetidos.
  - `rdf:Seq`: que especifica un cierto orden entre sus elementos y que también puede contener elementos repetidos.
  - `rdf:Alt`: que especifica la existencia de alternativas múltiples.

Ejemplo del uso de rdf:Bag

```
<uns:teacher rdf:about="#1234"
   uns:name="Juan Perez"
   uns:title="Profesor Adjunto">
   <uns:courses-taught>
      <rdf:Bag>
         <rdf:_1 rdf:resource="#CS101"/>
         <rdf:_2 rdf:resource="#CS105"/>
      </rdf:Bag>
   </uns:courses-taught>
</uns:teacher>
```

Colecciones

- Los contenedores presentan como limitación el no poder clausurar su contenido:
  - No es posible indicar que “sólo éstos son todos los elementos del contenedor”.
- RDF permite describir grupos de exáctamentos tales o cuales elementos, denominados colecciones:
  - Se representan como listas en el grafo asociado, usando las colecciones `rdf:List`, `rdf:first`, `rdf:rest` y `rdf:nil`.

Sintaxis alternativa para las colecciones

- Para evitar una sintaxis tan enrevezada, aprovechar la propia estructura inducida por el documento RDF mediante el atributo `rdf:parseType`.
- Por caso:

```
<rdf:Description rdf:about="#1234"
   <uns:courses-taught rdf:parseType="Collection">
      <rdf:Description rdf:about="#CS101"/>
      <rdf:Description rdf:about="#CS105"/>
   </uns:courses-taught>
</rdf:Description>
```

Ejemplo de reificación

- La siguiente declaración:

```
<rdf:Description rdf:about="#1234">
    <uns:name>Juan Perez</uns:name>
</rdf:Description>
```

- ...puede ser reificada como:

```
<rdf:Statement rdf:ID="AcercaDe1234">
    <rdf:subject rdf:resource="#1234"/>
    <rdf:predicate
      rdf:resource="http://cs.uns.edu.ar/ns#name"/>
    <rdf:object>Juan Perez</rdf:object>
</rdf:Statement>
```

Acerca de la reificación

- A través de los elementos `rdf:subject`, `rdf:predicate` y `rdf:object` se puede acceder a las diversas partes de la declaración que fuera reificada.
- Se usa `rdf:Description` en el caso de no necesitar referenciar posteriormente a esa declaración.
- Se usa `rdf:Statement` en el caso de querer poder hacer referencia a esa declaración desde otras partes.