

## Tecnologías para la web semántica

## Lenguajes para definir ontologías

- ♦ Cálculo de predicados de primer orden extendido para razonar sobre relaciones
  - CLASSIC (1989), CYCL (1990), LOOM (1991), KIF (1992), Ontolingua (1993), Frame Logic (1995)
- ♦ Nuevos lenguajes orientados a la web
  - SHOE (1996), RDF (1999), OIL (2000), DAML (2000), DAML+OIL (2002), OWL (2003)
  - TopicMaps, OCML, WebODE...

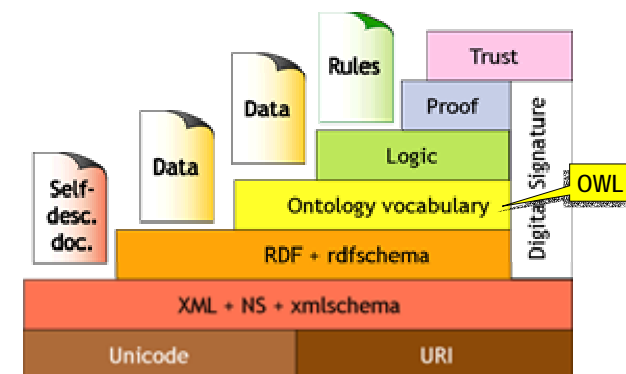
2

## Herramientas para definir ontologías

- ♦ Librerías y API's
  - Jena, Sesame, RDF Gateway...
- ♦ Entornos de desarrollo
  - Protégé, Kaon, WebODE...

3

## Niveles de soporte para la web semántica



4

## Estatus de los estándares

W3C Recommendation	HTML 4.0.1	24/12/99
	XML 1.0	10/02/98
	XML Schema	02/05/01
	RDF model & syntax	22/02/99
W3C Proposed recommendation	XML 1.1	05/11/03
W3C Candidate recommendation	OWL	18/08/03
W3C Working draft	RDF semantics, abstract syntax & RDF Schema	10/10/03

5

## Resource Description Framework (RDF)

## Lenguajes en internet

- ♦ HTML
  - Documentos, formato, enlaces
  - Muy simple, facilidad de uso
  - Demasiado simple, falta de semántica (títulos, tablas, párrafos)
- ♦ XML
  - Tags semánticos
  - Estructura de datos: árboles arbitrarios
  - Tags específicos para aplicaciones, intercambio de datos
  - Semántica y estructura mezclados
- ♦ RDF
  - Asignar semántica a un documento sin hacer suposiciones sobre su estructura
  - Jerarquía de clases

7

## Metadatos en HTML

- ♦ No hay un estándar
- ♦ No permite dar estructura

```
<html>
<head>
  <title>Distributed Metadata</title>
  <meta name="description" content="This article addresses...">
  <meta name="subject" content="metadata, rdf, peer-to-peer">
  <meta name="creator" content="Dan Brickley and Rael Dornfest">
  <meta name="publisher" content="O'Reilly">
  <meta name="date" content="2000-10-29T00:34:00+00:00">
  <meta name="type" content="article">
  <meta name="language" content="en-us">
  <meta name="rights" content="Copyright 2000, O'Reilly, Inc.">
</head>
...
```

8

## Metadatos en RDF

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <Item rdf:about="http://www.oreillynet.com/.../article.html">
    <title>Distributed Metadata</title>
    <link>http://www.oreillynet.com/.../article.html </link>
    <dc:description>This article addresses...</dc:description>
    <dc:subject>metadata, rdf, peer-to-peer </dc:subject>
    <dc:creator>Dan Brickley and Rael Dornfest </dc:creator>
    <dc:publisher>O'Reilly</dc:publisher>
    <dc:date>2000-10-29T00:34:00+00:00</dc:date>
    <dc:type>article</dc:type>
    <dc:language>en-us</dc:language>
    <dc:format>text/html</dc:format>
    <dc:rights>Copyright 2000, O'Reilly, Inc.</dc:rights>
  </Item>
  ...
```

9

## Resource Description Framework (RDF)

- ♦ W3C, octubre 97 (WD), febrero 99 (REC), octubre 03 (WD)
- ♦ Lenguaje para la descripción (metadatos) de recursos *explícitos* o *implícitos* en la web
- ♦ Definición de ontologías para la web semántica
- ♦ RDF Schema: propiedades y valores válidas en una descripción + jerarquía
- ♦ Esquema URI para identificar instancias
- ♦ Tipos de datos y literales
- ♦ Distintas sintaxis: XML, N3, N-Triples
- ♦ Lenguajes de query para RDF: RDQL, RQL, otros.
- ♦ Parsers: HP Jena API y otros  
ver <http://www.ilrt.bris.ac.uk/discovery/rdf/resources/#sec-tools>
- ♦ Espacio de nombres: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

10

## RDF vs. otros lenguajes

- ♦ RDF = XML + restricciones estructurales para asegurar consistencia
  - Orden es relevante en XML
  - XML permite mezclar texto y tags
  - RDF tiene una interpretación (semántica) propia (graph data model)
- ♦ Ventajas de RDF
  - Definición de clases (también XML Schema)
  - Jerarquía de clases, se puede examinar y manipular en tiempo de ejecución
  - Definición dispersa de ontología y de instancias (y metadatos separado de recursos)
- ♦ Diferencia con modelo ER
  - Las relaciones son objetos
  - Gran facilidad de extensión: la estructura de los schemas es abierta, se pueden añadir relaciones a posteriori
- ♦ OWL y otros = modelo RDF + constraints en propiedades + definición lógica de clases (con un predicado, unión o intersección de otras clases...)

11

## Bibliografía

<http://www.w3c.org/RDF>

- ♦ RDF Primer
- ♦ RDF Model & Syntax Specification (feb. 99)
- ♦ RDF Concepts and Abstract Syntax
- ♦ RDF/XML Syntax Specification
- ♦ RDF Vocabulary Description Language 1.0: RDF Schema
- ♦ RDF Semantics
- ♦ RDF Test Cases

12

## De las ontologías a RDF



- ♦ Ontología + red semántica
- ♦ Asignar un URI a cada nodo (unificar noción de identificador y URL)
- ♦ Asignar un tipo a cada nodo, en su caso definir una jerarquía de tipos
- ♦ Etiquetar arcos (tipos de arco)
- ♦ Permitir descripciones de arcos (arcos son también nodos)
- ♦ Permitir descripciones de tipos de arcos (qué tipo de nodos conecta, otras propiedades)
- ♦ Permitir descripciones de tipos de nodo (tipos de nodos son nodos)
- ♦ Al final RDF representa *todo* en términos de arcos (+ URIs y literales)

13

## El modelo básico RDF (1)

- ♦ Recurso (i.e. nodo)
  - Un URI o un literal (literal = typed literal, plain literal, ó XML literal)
  - Puede representar un documento, parte de él (p.e. un elemento XML), colección de documentos (p.e. un sitio web), objetos externos a la web (p.e. una persona)
  - Comparable a un objeto en POO o un registro en BD
- ♦ Propiedad (i.e. predicado, relación binaria)
  - Siempre es un URI
  - Característica, atributo o relación que describe un recurso, p.e. título, autor, etc.
  - Comparable a variable en POO o campo en BD
- ♦ Sentencia (i.e. triple): asignación de valores a propiedades de un recurso
  - Sujeto (recurso, URI) + predicado (propiedad, URI) + objeto (recurso, URI ó literal)
- ♦ Grafo: conjunto de sentencias

14

## El modelo básico RDF (2)

- ♦ Esquemas (RDFS)
  - Definición de jerarquías de clases
  - Atribución de instancias a clases
  - Sujetos (domain) y objetos (range) permitidos para una propiedad
- ♦ Reificación (meta-metadatos)
  - Las propiedades son también recursos
  - Las sentencias son también recursos
  - Las clases son también recursos

15

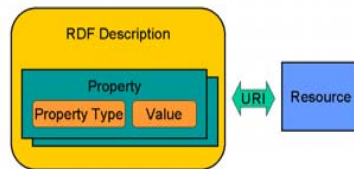
## Ejemplo (1)

- ♦ El documento <http://www.darbuka.com/index.html> tiene un autor cuyo valor es “Vedat Demirkan”
- ♦ El documento <http://www.darbuka.com/index.html> tiene un título cuyo valor es “Darbuka Method”
- ♦ El documento <http://www.darbuka.com/index.html> tiene una fecha cuyo valor es “1998-01-01”
- ♦ El documento <http://www.darbuka.com/index.html> tiene un tema cuyo valor es “Music, Percussion”

16

## Ejemplo (2)

```
<? xml version="1.0" ?>
<RDF xmlns = "http://w3.org/TR/1999/02/22-rdf-syntax-ns#"
  xmlns:dc = "http://purl.org/dc/elements/1.1/" >
  <Description about = "http://www.darbuka.com/index.html" >
    <dc:title> Darbuka Method </dc:title>
    <dc:creator> Vedat Demirkan </dc:creator>
    <dc:date> 1998-01-01 </dc:date>
    <dc:subject> Music, Percussion </dc:subject>
  </Description>
</RDF>
```



17

## Ejemplo (3)

### Sujeto

<http://www.darbuka.com/index.html>

### Predicados

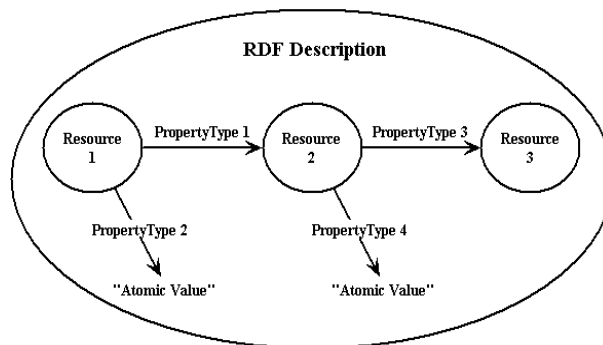
<http://purl.org/dc/elements/1.1/title>  
<http://purl.org/dc/elements/1.1/creator>  
<http://purl.org/dc/elements/1.1/date>  
<http://purl.org/dc/elements/1.1/subject>

### Objetos (en este caso, literales)

"Vedat Demirkan"  
 "Darbuka Method"  
 "1998-01-01"  
 "Music, Percussion"

18

## Encadenamiento de propiedades: URIs como valores



19

## Ejemplo (inline)

```
<Description about = "http://www.darbuka.com/index.html" >
  <dc:title> Darbuka Method </dc:title>
  <dc:creator>
    <Description about = "http://www.darbuka.com/vedat">
      <vCard:fn> Vedat Demirkan </vCard:fn>
      <vCard:title> Director </vCard:title>
      <vCard:email> vedat@darbuka.com </vCard:email>
      <vCard:role> Teacher </vCard:role>
    </Description>
  </dc:creator>
  <dc:date> 1998-01-01 </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>
```

20

## Ejemplo (offline)

```
<Description about = "http://www.darbuka.com/index.html" >
  <dc:title> Darbuka Method </dc:title>
  <dc:creator resource = "http://www.darbuka.com/vedat" />
  <dc:date> 1998-01-01 </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>

<Description about = "http://www.darbuka.com/vedat">
  <vCard:fn> Vedat Demirkan </vCard:fn>
  <vCard:title> Director </vCard:title>
  <vCard:email> vedat@darbuka.com </vCard:email>
  <vCard:role> Teacher </vCard:role>
</Description>
```

21

## Propiedades: recursos anónimos

```
<Description about = "http://www.darbuka.com/index.html" >
  <dc:title> Darbuka Method </dc:title>
  <dc:creator parseType="Resource">
    <vCard:fn> Vedat Demirkan </vCard:fn>
    <vCard:title> Director </vCard:title>
    <vCard:email> vedat@darbuka.com </vCard:email>
    <vCard:role> Teacher </vCard:role>
  </dc:creator>
  <dc:date> 1998-01-01 </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>
```

Equivalente a una  
descripción anónima

22

## Propiedades: valores literales XML

```
<Description about = "http://www.darbuka.com/vedat">
  <vCard:fn> Vedat Demirkan </vCard:fn>
  <vCard:title> Director </vCard:title>
  <vCard:email> vedat@darbuka.com </vCard:email>
  <vCard:role> Teacher </vCard:role>
  <dc:address parseType = "Literal">
    <b> Mr. Vedat Demirkan </b>
    <center> World Music School </center>
    <i> 500 Street </i>
    <b> Amsterdam </b>
  </dc:address>
</Description>
```

23

## Propiedades: valores literales tipados

```
<Description about = "http://www.darbuka.com/index.html" >
  <dc:title> Darbuka Method </dc:title>
  <dc:creator resource = "http://www.darbuka.com/vedat" />
  <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    1998-01-01
  </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>
```

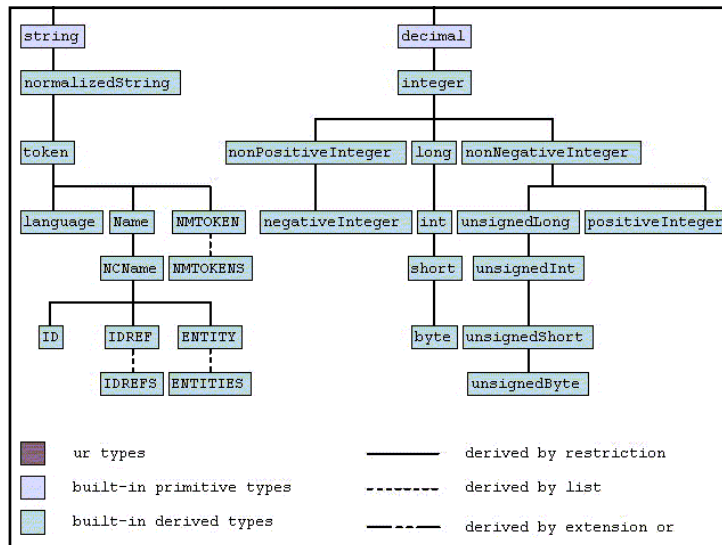
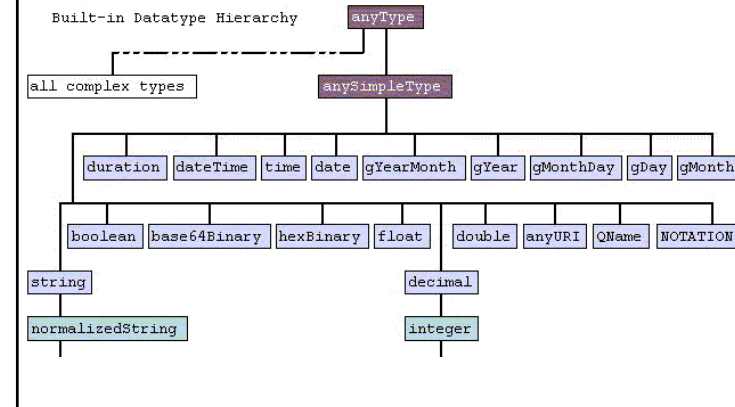
24

## Tipos de valores (recursos) para una propiedad

1. URI  
`<dc:creator resource = "http://www.darbuka.com/vedat"/>`
2. Plain literal: texto sin etiquetas XML  
`<dc:creator> Vedat Demirkan </dc:creator>`
3. XML Literal: puede tener etiquetas XML (o no)  
`<dc:creator rdf:parseType = "Literal">`  
`<vCard:fn> Vedat Demirkan </vCard:fn>`  
`<vCard:email> vedat@darbuka.com </vCard:email>`  
`<age> 45 </age>`  
`</dc:creator>`
4. Typed literal: tipos de datos XML Schema  
`<age rdf:datatype="http://www.w3.org/2001/XMLSchema#int">`  
`45 </age>`

25

## Tipos primitivos XMLS



## about vs. ID

- ♦ ID da lugar a un '#' antes de la cadena  
`<rdf:Description ID = "xxx"> ... </rdf:Description>`  
`<rdf:Description about = "#xxx"> ... </rdf:Description>`
- ♦ Un ID no se puede repetir en un mismo documento  
  - Se supone que un parser de RDF debe verificarlo
- ♦ ID tiene un significado especial cuando se usa con una propiedad: reificación
- ♦ Por lo demás about e ID son equivalentes

28

## URIs (1)

- Los URIs pueden ser URLs (i.e. recursos web accesibles, como `http://purl.org/dc/elements/1.1/`) pero no tienen por qué
  - En ocasiones es una convención, pero no es obligatorio
- Al usar propiedades (y clases) como etiquetas, su URI se separa en namespace:nombre

```
<rdf:RDF xmlns:dc = "http://purl.org/dc/elements/1.1/">
  <rdf:Property about="http://purl.org/dc/elements/1.1/title"/>
  <rdf:Description about = "...">
    <dc:title> ... <dc:title>
  </rdf:Description>
  ...
</rdf:RDF>
```

29

## URIs (2)

- `xml:base` se puede usar para abreviar los URIs en ID, about y resource

```
<rdf:RDF xmlns:dc = "http://purl.org/dc/elements/1.1/"
  xml:base = "http://purl.org/dc/elements/1.1/">
  <rdf:Property about = "title"/>
  <rdf:Description about = "...">
    <dc:title> ... <dc:title>
  </rdf:Description>
  ...
</rdf:RDF>
```

30

## URIs (3)

- El uso de XML entities simplifica la sintaxis y es útil cuando se necesitaría más de una XML base

```
<!DOCTYPE rdf:RDF
  [ <!ENTITY card "http://www.w3.org/2001/vcard-rdf/3.0#">
    <!ENTITY dcore "http://purl.org/dc/elements/1.1/"> ]>
<rdf:RDF xmlns:vCard = "%card;"
  xmlns:dc = "%dcore;">
  <rdf:Property about = "%dcore;title"/>
  <rdf:Description about = "...">
    <dc:title> ... <dc:title>
  </rdf:Description>
  ...
</rdf:RDF>
```

31

## Reificación: sentencias sobre sentencias

```
<Description about = "http://www.darbuka.com/index.html">
  <dc:creator ID = "xxx" Vedat Demirkan </dc:creator>
  <dc:title> Darbuka Method </dc:title>
</Description>

<Description about = "#xxx">
  <a:saysWho> Kim Nair </a:saysWho>
</Description>
```

32



## Sentencias como recursos

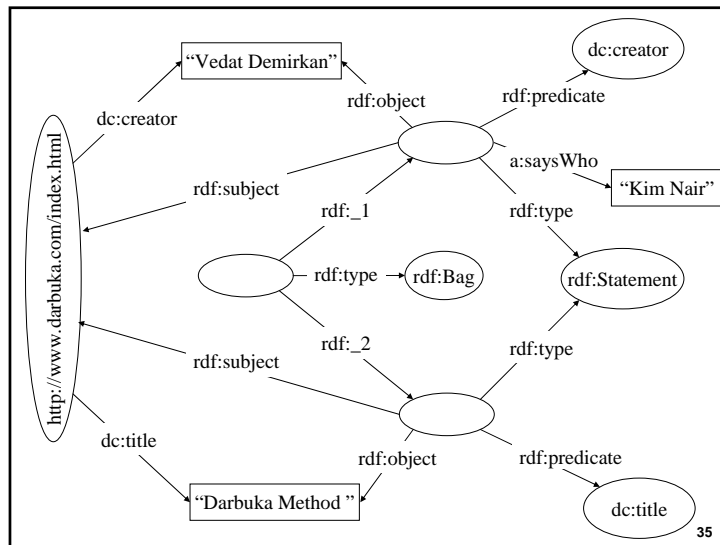
- ♦ Las sentencias tienen el tipo predefinido `rdf:Sentence`
- ♦ Atributos de `rdf:Sentence`
  - `rdf:Subject`
  - `rdf:Predicate`
  - `rdf:Object`
- ♦ Es posible añadir otros atributos a las sentencias
- ♦ Una descripción define implícitamente un Bag de sentencias
  - Una sentencia por cada propiedad asignada

33

## Ejemplo

```
<Description about = "http://www.darbuka.com/index.html">
  <dc:creator> Vedat Demirkan </dc:creator>
  <dc:title> Darbuka Method </dc:title>
</Description>
```

34



35

## Sentencias sobre sentencias

```
<rdf:Description>
  <rdf:subject resource = "http://www.darbuka.com/index.html" />
  <rdf:predicate resource = "http://description.org/schema/Creator" />
  <rdf:object> Vedat Demirkan </rdf:object>
  <rdf:type resource =
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement" />
  <a:saysWho> Kim Nair </a:saysWho>
</rdf:Description>
```

36

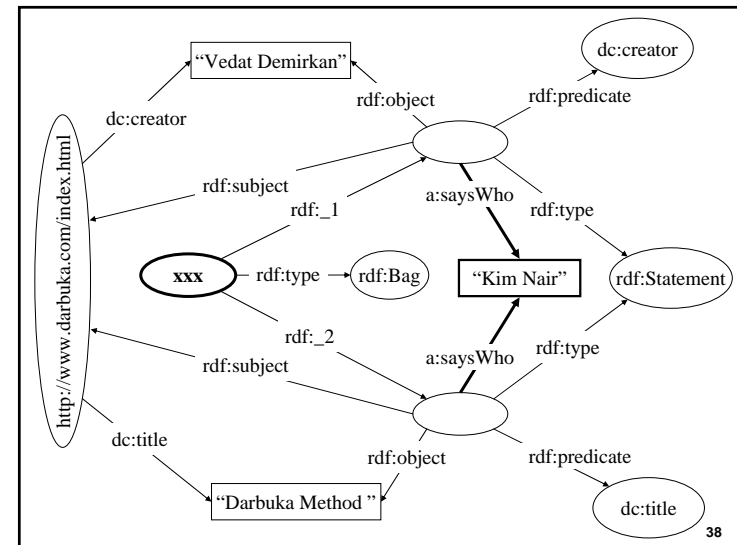
## Sentencias sobre grupos de sentencias

```
<Description about = "http://www.darbuka.com/index.html"
  bagID = "xxx">
  <dc:creator> Vedat Demirkan </dc:creator>
  <dc:title> Darbuka Method </dc:title>
</Description>
```

```
<Description aboutEach = "#xxx">
  <a:saysWho> Kim Nair </a:saysWho>
</Description>
```

- aboutEachPrefix añade propiedades a recursos
- aboutEach añade propiedades a las sentencias de una descripción con bagID (o a los elementos de una lista con ID)

37



38

## RDF abreviado

- ♦ Invisible para browsers HTML (todo queda en tags y atributos)
- ♦ Los valores de las propiedades tienen que ser strings
- ♦ Las propiedades no se pueden repetir

```
<Description about = "http://www.darbuka.com/index.html"
  dc:title = "Darbuka Method"
  dc:creator = "Vedat Demirkan"
  dc:date = "1998-01-01"
  dc:subject = "Music, Percussion"
/>
```


39

```
<Description about = "http://www.darbuka.com/index.html"
  dc:title = "Darbuka Method"
  dc:date = "1998-01-01"
  dc:subject = "Music, Percussion"
>
  <dc:creator resource = "http://www.darbuka.com/vedat"
    vCard:fn = "Vedat Demirkan"
    vCard:title = "Director"
    vCard:email = "vedat@darbuka.com"
    vCard:role = "Teacher"
  />
</Description>
```

40

```
<Description about = "http://www.darbuka.com/index.html" >
  <rdf:type resource = "http://percussion.org/schema/Lesson">
  <dc:title> Darbuka Method </dc:title>
  <dc:creator> Vedat Demirkan </dc:creator>
  <dc:date> 1998-01-01 </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>
```

Recursos  
tipados



```
<Lesson
  dc:title = "Darbuka Method"
  dc:creator = "Vedat Demirkan"
  dc:date = "1998-01-01"
  dc:subject = "Music, Percussion"
/>
```

41

## Valores múltiples

- ♦ Bag
  - Conjunto no ordenado
  - Permite duplicados
- ♦ Seq
  - Lista ordenada
  - Permite duplicados
- ♦ Alt
  - Valor único
  - Elección de un elemento de la lista
  - Al menos un elemento (valor por defecto)
- ♦ List
  - Colección de elementos fijos
- ♦ Los elementos se indican con <li> (mismas reglas gramaticales que una propiedad)
- ♦ Puede usarse una lista allí donde se permitiría una descripción
- ♦ Puede tener ID (pero no about)

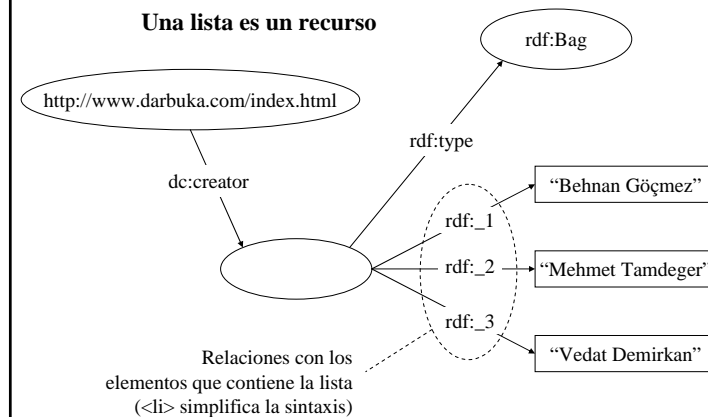
42

## Ejemplo

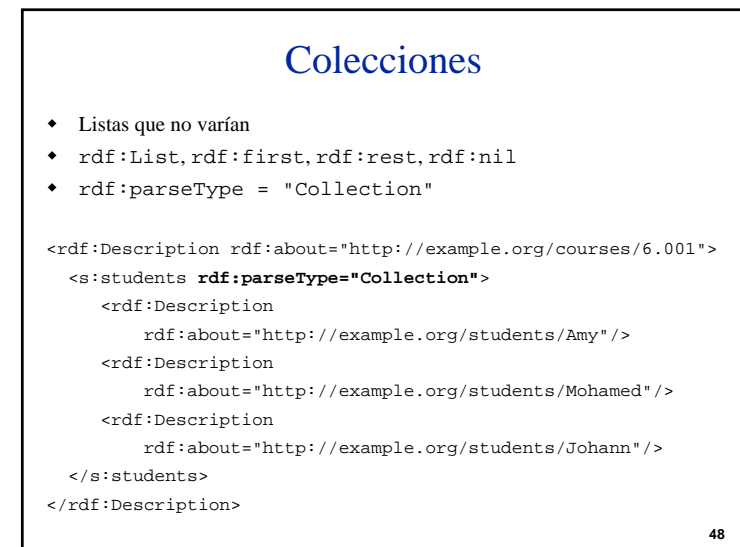
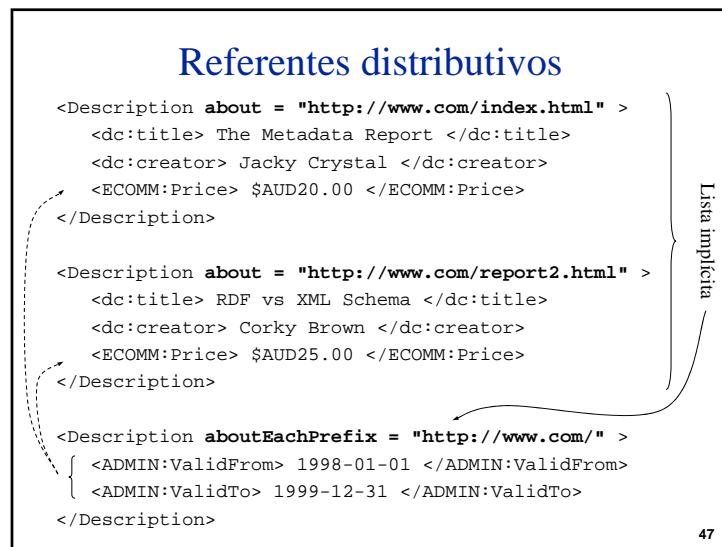
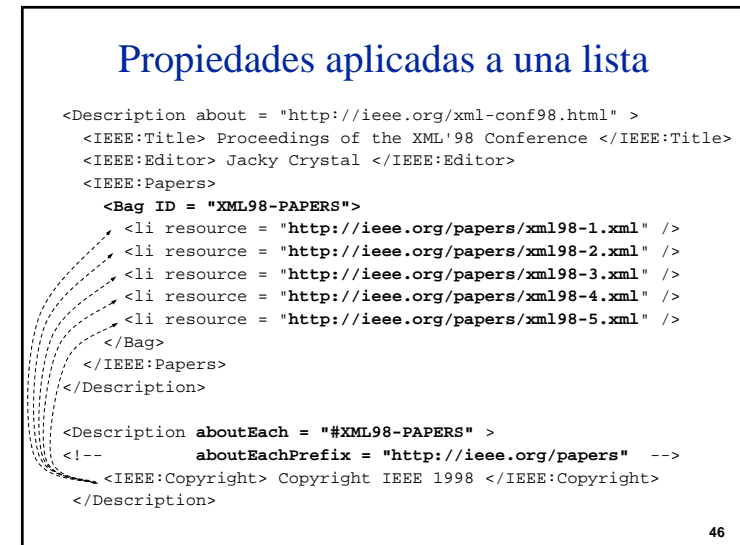
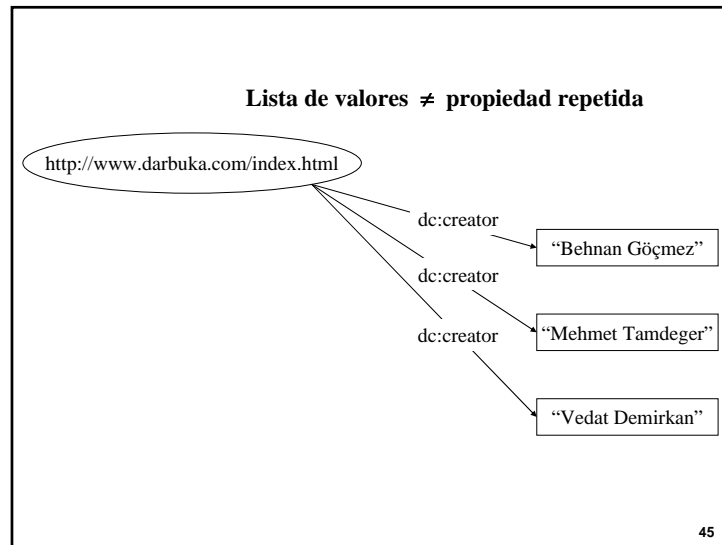
```
<Description about = "http://www.darbuka.com/index.html" >
  <dc:title> Darbuka Method </dc:title>
  <dc:creator>
    <Bag>
      <li> Behnan Göçmez </li>
      <li> Mehmet Tamdeger </li>
      <li> Vedat Demirkan </li>
    </Bag>
  </dc:creator>
  <dc:date> 1998-01-01 </dc:date>
  <dc:subject> Music, Percussion </dc:subject>
</Description>
```

43

### Una lista es un recurso



44



```

<rdf:Description rdf:about="http://example.org/courses/6.001">
  <s:students rdf:nodeID="sch1"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch1">
  <rdf:first rdf:resource="http://example.org/students/Amy"/>
  <rdf:rest rdf:nodeID="sch2"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch2">
  <rdf:first
    rdf:resource="http://example.org/students/Mohamed"/>
  <rdf:rest rdf:nodeID="sch3"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch3">
  <rdf:first
    rdf:resource="http://example.org/students/Johann"/>
  <rdf:rest
    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
</rdf:Description>

```

49

## RDF Schema

- ♦ Extensión de RDF para definir clases
  - Pertenencia de instancias a clases
  - Dominio y rango de propiedades: en qué clases está cada propiedad y qué valores puede tomar
  - Jerarquía de clases (induce herencia de tipos implícita)
- ♦ Establece una forma de restricciones sobre las sentencias RDF
- ♦ Extensible: las clases no quedan cerradas
- ♦ Influenciado por
  - Frames, redes semánticas
  - Modelo relacional BD
  - Lenguajes de lógica de predicados (CYC, KIF)
    - (menos expresivo, más eficiente)
- ♦ Espacio de nombres: <http://www.w3.org/2000/01/rdf-schema#>

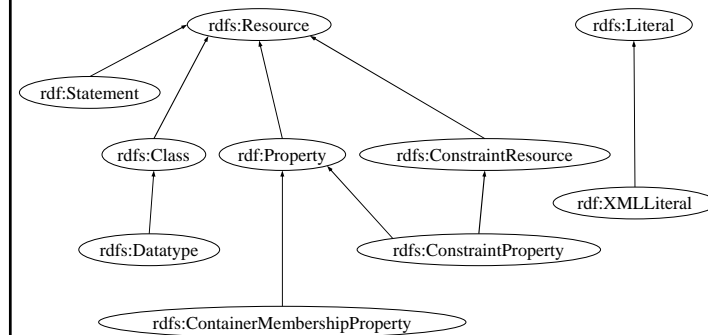
50

## RDFS vs. modelo OO

- ♦ OO describe clases en función de las propiedades que tienen
- ♦ RDFS define propiedades en función de las clases a las que es aplicable
- ♦ El enfoque centrado en propiedades permite la extensibilidad descentralizada de la definición de las clases (principio arquitectural de la web)

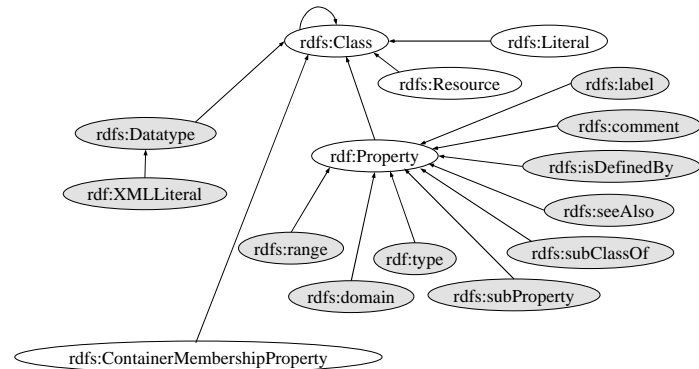
51

## Clases RDFS predefinidas



52

## Relaciones de clase / instancia



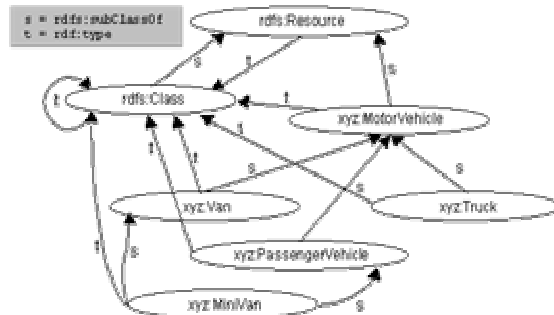
53

## Primitivas RDFS

- ♦ Clases (recursos de tipo Class)
  - Resource: corresponde al conjunto *Recursos*
  - Property: corresponde al conjunto *Propiedades*
  - Class: tipo a asignar a las definiciones de clases
  - Literal: strings, elementos XML (no RDF), corresponde al conjunto *Literales*
- ♦ Propiedades (instancias de Property)
  - type
    - Indica que un recurso pertenece a una clase (el recurso es una instancia de la clase)
    - El objeto debe ser un recurso de tipo Class (o subtipo de Class)
    - Se permite que un recurso tenga varios tipos
  - subClassOf
    - Relación de subconjunto entre clases (una instancia de una clase lo es también de todas las superclases de ésta)
    - Transitiva y no se permiten ciclos
    - Sujeto y objeto son recursos de tipo Class (o subtipo de Class)
    - Se permite herencia múltiple
  - ...

54

## Ejemplo



55

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xyz="http://www.w3.org/2000/03/example/vehicles#">

  <rdf:Description ID="MotorVehicle">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
      rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>

  <rdf:Description ID="PassengerVehicle">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  ...
```

56

```

...
<rdf:Description ID="Truck">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="Van">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="MiniVan">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>

</rdf:RDF>

```

57

#### ♦ Propiedades (cont.)

- ...
- seeAlso
  - Sujeto y objeto son recursos
- isDefinedBy
  - Subpropiedad de seeAlso
  - Sujeto y objeto son recursos

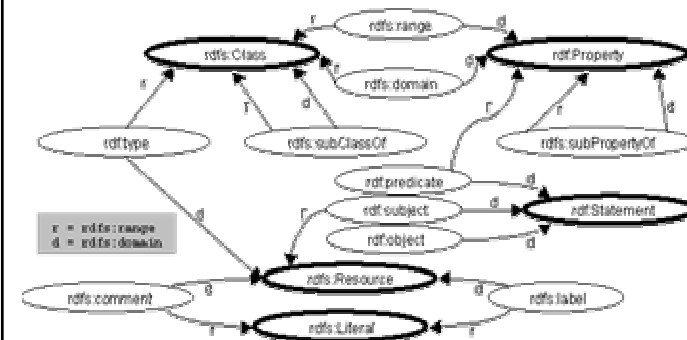
58

## Constraints

- ♦ Se utilizan para asociar propiedades a clases
- ♦ RDF no explicita cómo se deben procesar (en principio para validación)
- ♦ Propiedades
  - range
    - El sujeto es una propiedad, el objeto es una clase
    - Declara la clase del valor que puede tomar la propiedad
    - Una propiedad puede tener como máximo un range (ninguno  $\Rightarrow$  sin restricción)
  - domain
    - El sujeto es una propiedad, el objeto es una clase
    - Indica la clase a la que pertenece la propiedad
    - Una propiedad puede tener 0 ( $\Rightarrow$  sin restricción) ó más domain
  - ...

59

## Constraints predefinidas para las propiedades primitivas de RDFS



60

## Constraints: ejemplo

```
<rdf:Description ID = "registeredTo">
  <rdf:type resource =
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource = "#MotorVehicle"/>
  <rdfs:range rdf:resource = "#Person"/>
</rdf:Description>

<rdf:Description ID = "rearSeatLegRoom">
  <rdf:type resource =
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource = "#PassengerVehicle"/>
  <rdfs:domain rdf:resource = "#Minivan"/>
  <rdfs:range rdf:resource =
    "http://www.w3.org/2000/03/example/classes#Number"/>
</rdf:Description>
```

61

## ♦ Propiedades (cont.)

```
- ...
- subPropertyOf
  • Sujeto y objeto son propiedades
  • p1 subpropiedad de p2, y {p1, sub, obj} ⇒ {p2, sub, obj}
  • Ejemplo:

  <rdf:Description ID = "Parent">
    <rdf:type resource =
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>

  <rdf:Description ID = "Father">
    <rdf:type resource =
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource = "#biologicalParent"/>
  </rdf:Description>

- ...
```

62

## ♦ Clases (cont.)

- ...
- Container
  - Subclase de Resource
  - Superclase de Bag, Seq, Alt
- ContainerMembershipProperty
  - Subclase de Property
  - Clase de las propiedades \_1, \_2, \_3 ...

## ♦ Propiedades (cont.)

- ...
- comment
- label

63

## Ejemplo de RDFS (síntaxis abreviada)

```
<rdfs:Class rdf:ID = "Person">
  <rdfs:comment> The class of people. </rdfs:comment>
  <rdfs:subClassOf rdf:resource =
    "http://www.w3.org/2000/03/example/classes#Animal"/>
</rdfs:Class>

<rdf:Property ID = "ssn">
  <rdfs:comment>Social Security Number</rdfs:comment>
  <rdfs:domain rdf:resource = "#Person"/>
  <rdfs:range rdf:resource =
    "http://www.w3.org/2000/03/example/classes#Integer"/>
</rdf:Property>

<rdf:Property ID = "age">
  <rdfs:domain rdf:resource = "#Person"/>
  <rdfs:range rdf:resource =
    "http://www.w3.org/2000/03/example/classes#Integer"/>
</rdf:Property>
...
```

64



```
...  
<rdf:Property ID = "maritalStatus">  
  <rdfs:domain rdf:resource = "#Person"/>  
  <rdfs:range rdf:resource = "#MaritalStatus"/>  
</rdf:Property>  
  
<rdfs:Class rdf:ID = "MaritalStatus"/>  
  
<MaritalStatus rdf:ID = "Married"/>  
<MaritalStatus rdf:ID = "Divorced"/>  
<MaritalStatus rdf:ID = "Single"/>  
<MaritalStatus rdf:ID = "Widowed"/>  
  
<Person rdf:ID = "abcde0012" ssn = "123456" age = "42">  
  <maritalStatus rdf:resource = "#Married" />  
</Person>
```

65