

Methods and Tools for Management Information Systems

Lecture 4

17. November 2008



Resource Description Framework (RDF)

- Language for representing information about resources in the World Wide Web, particularly intended for representing metadata about Web resources
- RDF uses a more general notion of *Web resources*—not only *things* that can be *retrieved* on the Web but also things that can be *identified* on the Web via its URI
- Resources are described in terms of *properties* and property *values*



- RDF provides a common framework for expressing information that can be processed/exchanged by/between *applications*
- RDF defines the fundamental vocabulary to formulate arbitrary statements about *resources*
- Most important application is the *Semantic Web* which extends the existing Web by machine-processable content/metadata
- RDF specification consists of two parts:
 - *RDF Graph* represents the fundamental data model
 - *RDF/XML* provides the XML syntax to serialize those data



■ RDF Graph:

- Information is kept in form of *statements*, triples of **subject**, **predicate** and **object**, identified by URI references (URIrefs)
- Objects may also be constant values represented by text strings (so-called *literals*)
- URIrefs are represented as ellipses, literals as boxes
- Triples are connected in form of a labeled directed graph where nodes refer to subjects and objects, and arcs (directed from the subject node to the object node) to predicates
- Predicates themselves may also be resources and, as such, the subject of other statements (which can be used in other vocabularies, i. e., sets of URIrefs defined for a certain purpose)



Example:

There is a *Person* identified by

<http://www.w3.org/People/EM/contact#me>, whose *name* is Eric Miller, whose *email address* is em@w3.org, and whose *title* is Dr.



Serialization in RDF/XML:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```



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

```
<rdf:Property
  rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
  <rdfs:isDefinedBy
    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <rdfs:label>type</rdfs:label>
  <rdfs:comment>
    The subject is an instance of a class.
  </rdfs:comment>
  <rdfs:range
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:domain
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Property>
```



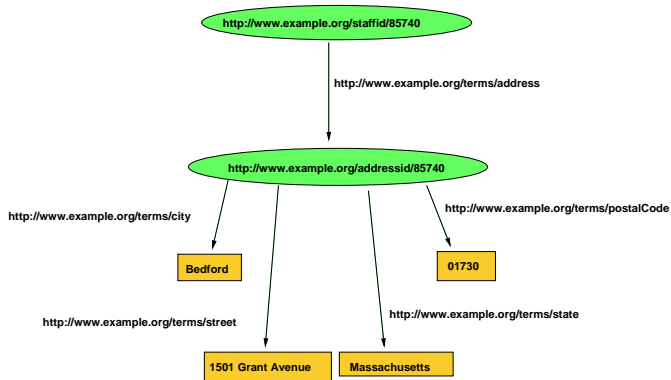
- Structured properties and blank nodes:

<http://www.example.org/staffid/85740>

<http://www.example.org/terms/address>

1501 Grant Avenue, Bedford, Massachusetts 01730

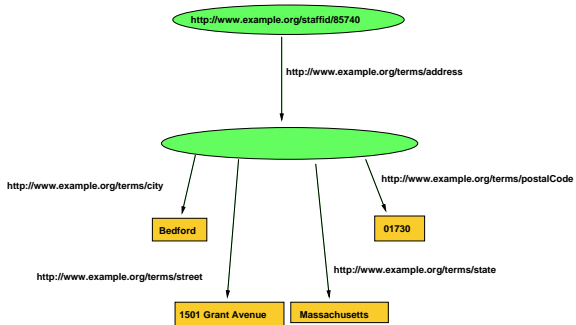




Triples notation:

<code>exstaff:85740</code>	<code>externs:address</code>	<code>exaddressid:85740 .</code>
<code>exaddressid:85740</code>	<code>externs:street</code>	<code>"1501 Grant Avenue"</code>
<code>exaddressid:85740</code>	<code>externs:city</code>	<code>"Bedford" .</code>
<code>exaddressid:85740</code>	<code>externs:state</code>	<code>"Massachusetts" .</code>
<code>exaddressid:85740</code>	<code>externs:postalCode</code>	<code>"01730" .</code>





■ Representation in triples notation using *Blank Node Identifiers*:

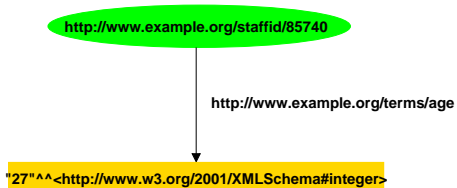
```
exstaff:85740          externs:address      _:johnaddress .
_:johnaddress         externs:street      "1501 Grant Avenue" .
_:johnaddress         externs:city        "Bedford" .
_:johnaddress         externs:state       "Massachusetts" .
_:johnaddress         externs:postalCode  "01730" .
```

■ Blank nodes ...

- ⇒ may not represent predicates
- ⇒ can illustrate certain relationships more precisely



- Typed literals:



Triples notation:

`exstaff:85740`

`exterms:age`

`"27"^^xsd:integer .`



- General rules:

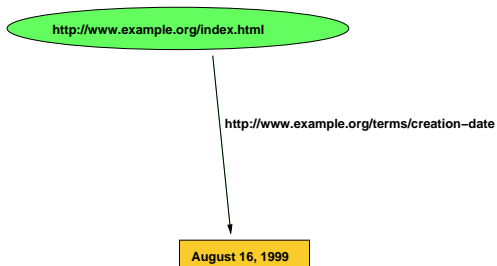
- ⇒ resources without URI (e. g. blank nodes) cannot be referenced

- ⇒ RDF can only represent binary relationships

- ⇒ type safety accomplishable by *typed literals*



- XML Syntax for RDF: RDF/XML
 - Serialization of RDF data in XML (*normativ*):



■ XML Syntax for RDF: RDF/XML

- Serialization of RDF data in XML (*normativ*):

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterms="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date>August 16, 1999</exterms:creation-date>
  </rdf:Description>

</rdf:RDF>
```



- Basic idea of RDF/XML: encode an RDF graph as XML elements, attributes, element content, and attribute values
- URIs of predicates (as well as some nodes) are written as XML QNames, i. e., consisting of a prefix denoting a namespace URI and a local name
- URIs of subject nodes (as well as some object nodes) are written as XML attribute values
- Literal nodes (which are always object nodes) become element text content or attribute values



- An RDF graph consisting of multiple statements can be represented using multiple description elements:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:exterms="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date>August 16, 1999</exterms:creation-date>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:language>en</dc:language>
  </rdf:Description>
</rdf:RDF>
```



- A description element may also contain multiple predicates:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:exterms="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date>August 16, 1999</exterms:creation-date>
    <dc:language>en</dc:language>
    <dc:creator
      rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>

</rdf:RDF>
```



■ Serialization of blank nodes using node identifiers:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:externs="http://example.org/stuff/1.0/">
  <rdf:Description
    rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
    <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
    <externs:editor rdf:nodeID="abc"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="abc">
    <externs:fullName>Dave Beckett</externs:fullName>
    <externs:homePage rdf:resource="http://purl.org/net/dajobe/">
  </rdf:Description>
</rdf:RDF>
```



■ Using typed literals:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterms="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date
      rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
      1999-08-16
    </exterms:creation-date>
  </rdf:Description>

</rdf:RDF>
```



■ Using XML entities:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF
  [
```



■ Abbreviating URIs using *fragment identifiers*:

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF
  [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:externs="http://www.example.com/terms/">
  <rdf:Description rdf:ID="item10245">
    <externs:model rdf:datatype="&xsd:string">
      Overnighter
    </externs:model>
    <externs:sleeps rdf:datatype="&xsd:integer">2</externs:sleeps>
    <externs:weight rdf:datatype="&xsd:decimal">2.4</externs:weight>
  </rdf:Description>
</rdf:RDF>
```



- Using the constructs described so far, an RDF graph is written in RDF/XML as follows:
 - 1 All blank nodes are assigned blank node identifiers
 - 2 Each node is listed in turn as the subject of an un-nested `rdf:Description` element, using an `rdf:about` attribute if the node has a `URIref`, or an `rdf:nodeID` attribute if the node is blank
 - 3 For each triple with this node as subject, an appropriate property element is created, with either literal content, an `rdf:resource` attribute specifying the object of the triple, or an `rdf:nodeID` attribute specifying the object of the triple
- ⇒ provides the most direct representation of the graph structure
- ⇒ recommended for applications that process RDF/XML further



- Other RDF capabilities:

- Using containers:

- `rdf:Bag`:

- ⇒ group of resources or literals, possibly including duplicate members, without order

- `rdf:Seq`:

- ⇒ group of resources or literals, possibly including duplicate members, where the order of the members is significant

- `rdf:Alt`:

- ⇒ group of resources or literals that are alternatives



■ Example:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">

  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students>
      <rdf:Bag>
        <rdf:li rdf:resource="http://example.org/students/Amy"/>
        <rdf:li rdf:resource="http://example.org/students/Tom"/>
        <rdf:li rdf:resource="http://example.org/students/Jim"/>
      </rdf:Bag>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```



■ Using collections:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">

  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/students/Amy"/>
      <rdf:Description rdf:about="http://example.org/students/Tom"/>
      <rdf:Description rdf:about="http://example.org/students/Jim"/>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```



■ Reification:

```
exproducts:triple12345    rdf:type          rdf:Statement .
exproducts:triple12345    rdf:subject       exproducts:item10245 .
exproducts:triple12345    rdf:predicate     externs:weight .
exproducts:triple12345    rdf:object        "2.4"^^xsd:decimal .
```



- `rdf:value` to represent the *main values* of a structured value:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF
  [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterm="http://www.example.org/terms/"

  <rdf:Description
    rdf:about="http://www.example.com/2002/04/products#item10245">
    <exterm:weight rdf:parseType="Resource">
      <rdf:value rdf:datatype="xsd:decimal">2.4</rdf:value>
      <exterm:units
        rdf:resource="http://www.example.org/units/kilograms"/>
    </exterm:weight>
  </rdf:Description>
</rdf:RDF>
```



■ Literals to represent fragments of XML:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.example.com/books">

  <rdf:Description rdf:ID="book12345">
    <dc:title rdf:parseType="Literal">
      <span xml:lang="en">
        The <em>&lt;br /&gt;</em> Element Considered Harmful.
      </span>
    </dc:title>
  </rdf:Description>
</rdf:RDF>
```



RDF Vocabulary Description Language (RDF Schema)

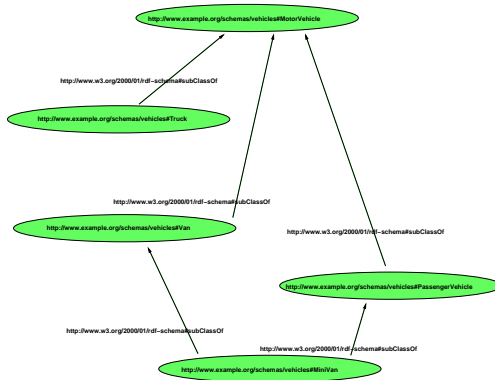
- RDF Schema is used to describe types and properties of resources
- Provides a type system similar to those used in object-oriented programming languages:
 - Class hierarchy
 - Resources as instances of one or more classes
- RDF Schema facilities are themselves provided in the form of an RDF vocabulary defined in a namespace that is bound to the URI <http://www.w3.org/2000/01/rdf-schema#>



- *Vocabulary descriptions* written in RDF Schema represent valid RDF graphs
- A class in RDF Schema corresponds to the generic concept of a *Type* or *Category* and can represent almost any category of thing, such as Web pages, people, document types, databases or abstract concepts
- Classes are described using the RDF Schema resources `rdfs:Class` and `rdfs:Resource`, and the attributes `rdf:type` and `rdfs:subClassOf`
- Properties are described using the RDF class `rdf:Property`, and the RDF Schema properties `rdfs:domain`, `rdfs:range`, and `rdfs:subPropertyOf`



■ *Example:*



Tripels notation:

ex:MotorVehicle	rdf:type	rdfs:Class .
ex:PassengerVehicle	rdf:type	rdfs:Class .
ex:Van	rdf:type	rdfs:Class .
ex:Truck	rdf:type	rdfs:Class .
ex:MiniVan	rdf:type	rdfs:Class .
ex:PassengerVehicle	rdfs:subClassOf	ex:MotorVehicle .
ex:Van	rdfs:subClassOf	ex:MotorVehicle .
ex:Truck	rdfs:subClassOf	ex:MotorVehicle .
ex:MiniVan	rdfs:subClassOf	ex:Van .
ex:MiniVan	rdfs:subClassOf	ex:PassengerVehicle .



RDF/XML:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
```



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



- Describing classes:

- Defining simple classes:

```
<rdf:Description rdf:ID="class_name">  
  <rdf:type  
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
</rdf:Description>
```

- Class definitions may be abbreviated:

```
<rdfs:Class rdf:ID="class_name"/>
```

- Specialization of classes using rdfs:subClassOf:

```
<rdfs:Class rdf:ID="class_name">  
  <rdfs:subClassOf rdf:resource="super_class"/>  
</rdfs:Class>
```



■ *Example:*

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

or abbreviated:

```
<rdfs:Class rdf:ID="MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class>
```



- Describing attributes:

- Properties in RDF are described as instances of class

```
ex:terms:weightInKg    rdf:type    rdf:Property .
```

- Intended usage of properties can be described using the RDF Schema properties `rdfs:range` and `rdfs:domain`:

- `rdfs:range`

- Values of a property are instances of a designated class:

```
ex:Person    rdf:type    rdfs:Class .
```

```
ex:author    rdf:type    rdf:Property .
```

```
ex:author    rdfs:range  ex:Person .
```



- Properties may have more than one `rdf:range` property:

```
ex:hasMother    rdfs:range    ex:Female .  
ex:hasMother    rdfs:range    ex:Person .  
exstaff:frank   ex:hasMother  exstaff:frances .
```

⇒ `exstaff:frances` has to be an instance of both classes
`ex:Female` *and* `ex:Person`

- Using typed literals for the `rdf:range` property:

```
ex:age    rdf:type    rdf:Property .  
ex:age    rdfs:range  xsd:integer .
```

Remark: while it is possible to explicitly name types, e. g.:

```
        xsd:integer    rdf:type    rdfs:Datatype .  
it is not possible to define new types using RDF Schema
```



■ rdfs:domain

- Property applies to a designated class:

```
ex:Book      rdf:type      rdfs:Class .
ex:author    rdf:type      rdf:Property .
ex:author    rdfs:domain   ex:Book .
```

- Properties may have more than one rdfs:domain property:

```
exterms:weight      rdfs:domain   ex:Book .
exterms:weight      rdfs:domain   ex:MotorVehicle .
exthings:companyCar exterms:weight "2500"^^xsd:integer .
```

⇒ `exthings:companyCar` has to be an instance of both classes
`ex:Book` *and* `ex:MotorVehicle`



■ Serializing in RDF/XML:

```
<rdf:Property rdf:ID="registeredTo">  
  <rdfs:domain rdf:resource="#MotorVehicle"/>  
  <rdfs:range rdf:resource="#Person"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="rearSeatLegRoom">  
  <rdfs:domain rdf:resource="#PassengerVehicle"/>  
  <rdfs:range rdf:resource="&xsd;integer"/>  
</rdf:Property>
```

```
<rdfs:Class rdf:ID="Person"/>  
<rdfs:Datatype rdf:about="&xsd;integer"/>
```



- Specialization of properties using `rdfs:subPropertyOf`:

```
ex:driver          rdf:type          rdf:Property .
ex:primaryDriver  rdf:type          rdf:Property .
ex:primaryDriver  rdfs:subPropertyOf ex:driver .
```

- Serializing in RDF/XML:

```
<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>

<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
```



- A property may have any number of `rdfs:subPropertyOf` relationships
- RDF schema properties that apply to a given property also apply to its subproperties



■ Example

■ RDF Schema document:

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>

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

  <rdfs:Class rdf:ID="MotorVehicle"/>
  <rdfs:Class rdf:ID="PassengerVehicle">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>
```



```
<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Person"/>
<rdfs:Datatype rdf:about="&xsd;integer"/>
<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>
```



```
<rdf:Property rdf:ID="rearSeatLegRoom">
  <rdfs:domain rdf:resource="#PassengerVehicle"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>
<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
</rdf:RDF>
```



■ Corresponding RDF instance document (ex:PassengerVehicle):

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/schemas/vehicles#"
  xml:base="http://example.org/things">
  <ex:PassengerVehicle rdf:ID="johnSmithsCar">
    <ex:registeredTo
      rdf:resource="http://www.example.org/staffid/85740"/>
    <ex:rearSeatLegRoom
      rdf:datatype="&xsd;integer">127</ex:rearSeatLegRoom>
    <ex:primaryDriver
      rdf:resource="http://www.example.org/staffid/85740"/>
  </ex:PassengerVehicle>
</rdf:RDF>
```



- RDF Schema supplies a number of builtin properties:
 - `rdfs:comment` to provide a human-readable description of a resource
 - `rdfs:label` to provide a more human-readable version of a resource's name
 - `rdfs:seeAlso` to indicate a resource that might provide additional information about the subject resource
 - `rdfs:isDefinedBy` to indicate a resource that *defines* the subject resource (subproperty of `rdfs:seeAlso`)



- Differences between RDF Schema declarations and type systems of object-oriented programming language:
 - Instead of describing a class as having a collection of specific properties, an RDF schema describes properties as applying to specific classes of resources
 - ⇒ Independence of classes and properties
 - Properties are always defined on a global level
 - RDF Schema descriptions are not necessarily *prescriptive*, but additional *descriptions* of resources (which *may* be used in instance documents)
 - ⇒ Properties vs. constraints



- Schema capabilities not provided by RDF Schema:
 - Cardinality constraints on properties
 - Specifying that a given property is transitive
 - Specifying that a given property is a unique identifier (or key) for instances of a particular class
 - Specifying that two different classes (having different URIs) actually represent the same class
 - Specifying that two different instances (having different URIs) actually represent the same individual
 - Specifying constraints on the range or cardinality of a property that depend on the class of resource to which a property is applied
 - Description of new classes in terms of combinations of other classes (union, intersection, disjoint)



RDF in the Field: Dublin Core Metadata Initiative

- Minimal set of descriptive elements that facilitate the description and the automated indexing of document-like networked objects
- Originally developed in March 1995 at a Workshop on Metadata Management in Dublin, Ohio
- Suitable for use by resource discovery tools on the Internet
- Sufficiently simple to be understood and used by a wide range of authors and casual publishers and widely used in documenting Internet resources



- Elements of the Dublin Core are defined in the Dublin Core Metadata Element Set, Version 1.1: Reference Description, and contain definitions for the following properties:



Property

Description

title

A name given to the resource

creator

An entity primarily responsible for making the content of the resource

subject

The topic of the content of the resource

description

An account of the content of the resource

publisher

An entity responsible for making the resource available



Property	Description
contributor	An entity responsible for making contributions to the content of the resource
date	A date associated with an event in the life cycle of the resource
type	The nature or genre of the content of the resource
format	The physical or digital manifestation of the resource
identifier	An unambiguous reference to the resource within a given context



Property	Description
source	A reference to a resource from which the present resource is derived
language	A language of the intellectual content of the resource
relation	A reference to a related resource
coverage	The extent or scope of the content of the resource
rights	Information about rights held in and over the resource



- Additional vocabulary is defined in <http://purl.org/dc/terms>
- Dublin Core Metadata may be captured in any suitable language (even in the form of HTML meta tags), but RDF is an ideal representation:

```
<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.dlib.org">
  <dc:title>D-Lib Program - Research in Digital
    Libraries</dc:title>
  <dc:description>The D-Lib program supports the community of
    people with research interests in digital libraries and
    electronic publishing.</dc:description>
  <dc:publisher>Corporation For National Research
    Initiatives</dc:publisher>
  <dc:date>1995-01-07</dc:date>
```



```
<dc:subject>
  <rdf:Bag>
    <rdf:li>Research; statistical methods</rdf:li>
    <rdf:li>Education, research, related topics</rdf:li>
    <rdf:li>Library use Studies</rdf:li>
  </rdf:Bag>
</dc:subject>
<dc:type>World Wide Web Home Page</dc:type>
<dc:format>text/html</dc:format>
<dc:language>en</dc:language>
</rdf:Description>
</rdf:RDF>
```



- Resource descriptions may either reside directly in the document:

```
<?xml version="1.0"?>
<svg width="4in" height="3in" version="1.1"
  xmlns = 'http://www.w3.org/2000/svg'>
  <metadata>
    <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://example.org/foo">
        <dc:creator>
          Mary Lambert
        </dc:creator>
      </rdf:Description>
    </rdf:RDF>
  </metadata>
</svg>
```



or in a separate file (being referenced in the original document):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">

<html>
  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=us-ascii"/>
  </head>
  <body>
    <!-- .... -->
    <a href="http://www.example.org/metadata.rdf">Metadata</a>
  </body>
</html>
```



- The following documents contribute to the specification of RDF:
 - *RDF Concepts and Abstract Syntax*
 - *RDF/XML Syntax Specification*
 - *RDF Vocabulary Description Language 1.0: RDF Schema*
 - *RDF Semantics*
 - *RDF Test Cases*
 - *RDF Primer*

