RDF
Resource Description Framework

Fulvio Corno, Laura Farinetti

Politecnico di Torino

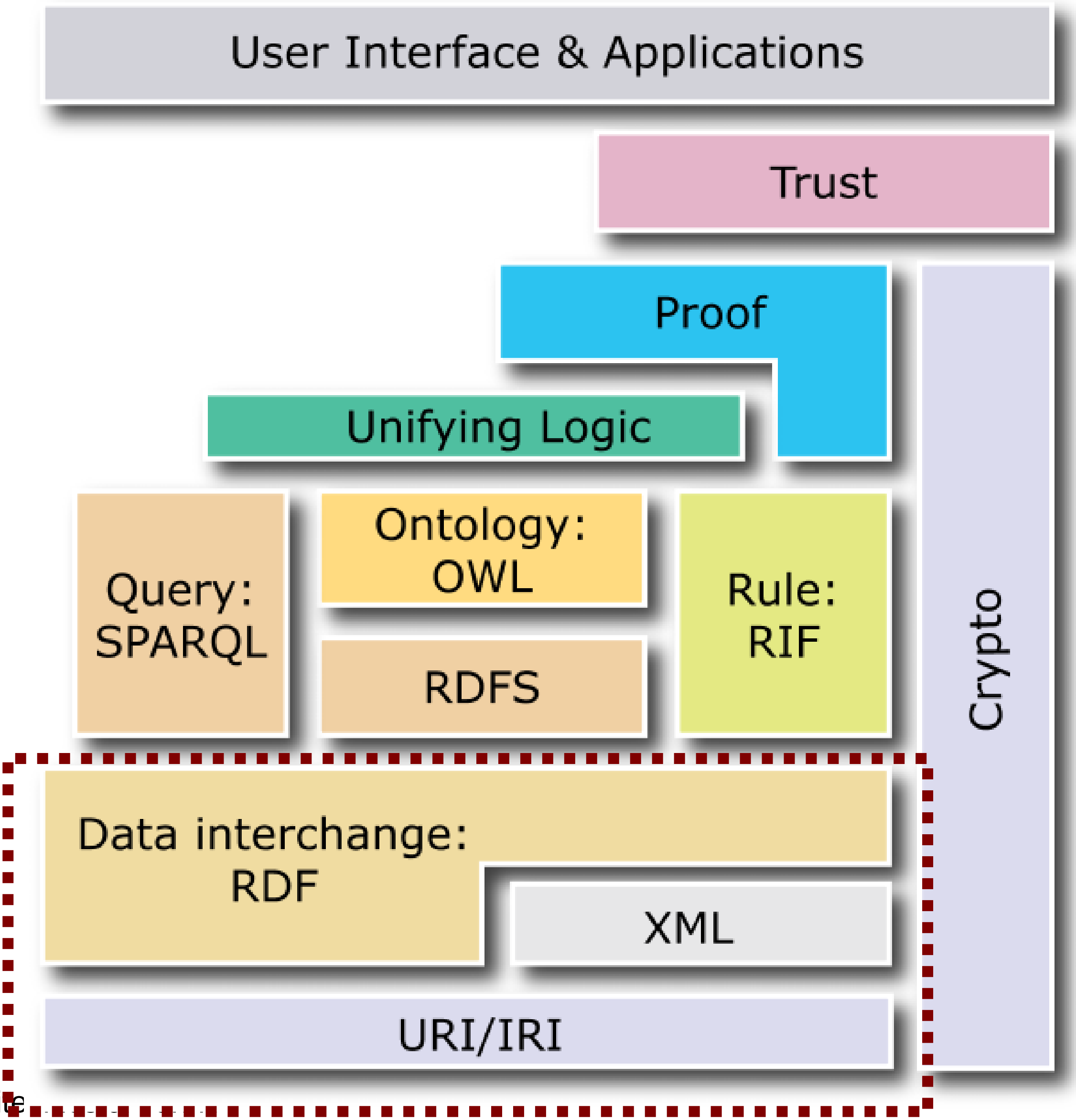
Dipartimento di Automatica e Informatica

e-Lite Research Group – http://elite.polito.it

Outline

- **RDF Design objectives**
- RDF General structure
- RDF Vocabularies
- Serialization: XML
- Semantic features
- RDF Schema

SW Technology Stack



User Interface & Applications

Trust

Proof

Unifying Logic

Query: SPARQL

Ontology: OWL

RDFS

Rule: RIF

Crypto

Data interchange: RDF

XML

URI/IRI

A common language for describing resources

- The Resource Description Framework (RDF) is a language for **representing information about resources** in the World Wide Web
- Particularly intended for representing **metadata** about Web resources
- RDF can also be used to represent information about things that can be **identified** on the Web, even when they cannot be directly **retrieved** on the Web

RDF Design goals

- having a **simple data model**
- having **formal semantics** and **provable inference**
- using an extensible **URI-based vocabulary**
- using an XML-based syntax
- supporting use of XML schema **datatypes**
- allowing **anyone** to make statements about **any** resource

Simple yet powerful

- RDF has an **abstract syntax** that reflects a simple graph-based data model
- RDF has **formal semantics** with a rigorously defined notion of entailment providing a basis for well founded deductions

Basic principles (1/2)

- Clearly separate
  - **Model** structure (RDF graph)
  - Interpretation **Semantics** (Entailment)
  - Concrete **Syntaxes** (XML, TN, N3, …)
- Only two datatypes
  - URI/URIref: everything is a URI
  - Literal
    - String or other XSD datatype

Basic principles (2/2)

- Integrated with the Web
  - Uses XMLSchema datatypes
  - May reference http-retrievable resources
- **Open world** assumption
  - Allows anyone to make statements about any resource
  - No guaranteed completeness
  - No guaranteed consistency

Outline

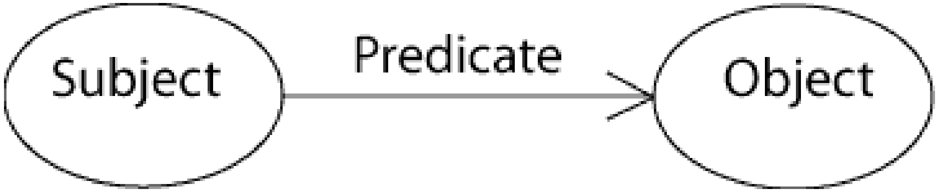
- RDF Design objectives
- **RDF General structure**
- RDF Vocabularies
- Serialization: XML
- Semantic features
- RDF Schema

Key concepts

- Graph data model
- URI-based vocabulary
- Datatypes
- Literals
- XML serialization syntax
- Expression of simple facts
- Entailment

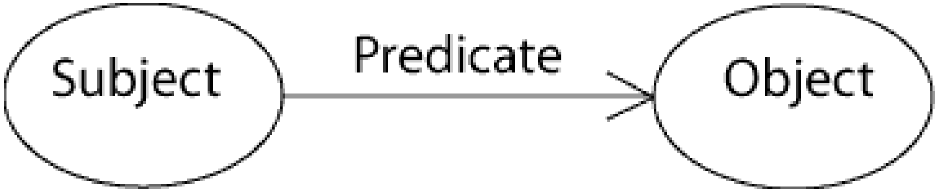
Graph data model

- Triple: subject, predicate, object
- Expression: collection of triples
  - RDF graph



Terminology and constraints

- *Subject* and *Object* are called *Nodes*
- *Predicate* and *Property* are synonyms
- Special unnamed nodes: Blank Nodes
- Subject may be: URI reference or blank node
- Predicate must be: URI reference
- Object may be: URI reference, literal or blank node



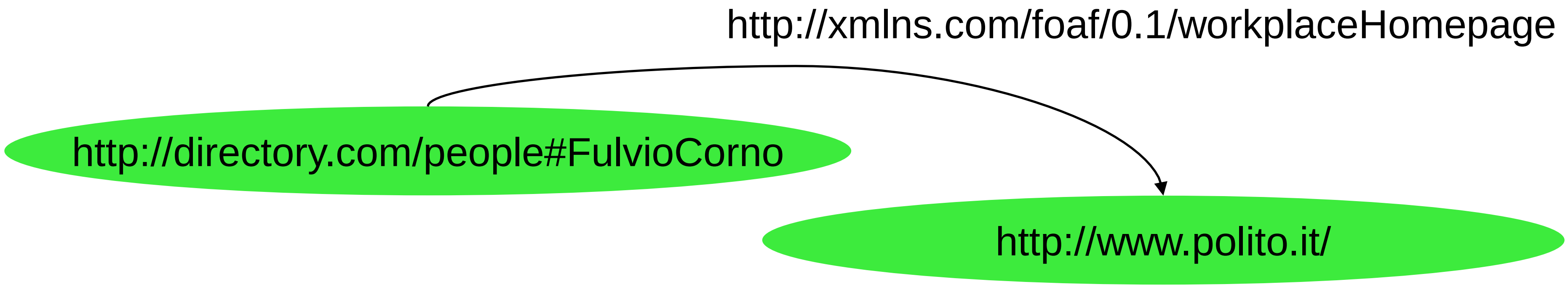
The Triples and the Graph

- The *assertion of an RDF triple* says that some relationship, indicated by the predicate, holds between the things denoted by subject and object of the triple.
- The *assertion of an RDF graph* amounts to asserting all the triples in it, so the meaning of an RDF graph is the conjunction (logical AND) of the statements corresponding to all the triples it contains.

Expression of Simple Facts

- Some simple facts indicate a relationship between two things → one triple
    - the predicate names the relationship
    - the subject and object denote the two things

Information in triples


http://xmlns.com/foaf/0.1/workplaceHomepage
http://directory.com/people#FulvioCorno
http://www.polito.it/

RDF

CompanyHomePage

PersonID	Homepage
FulvioCorno	http://www.polito.it/

Relational database

First order logic predicate

HasCompanyHomePage(
‘FulvioCorno’,
‘http://www.polito.it/’) ;

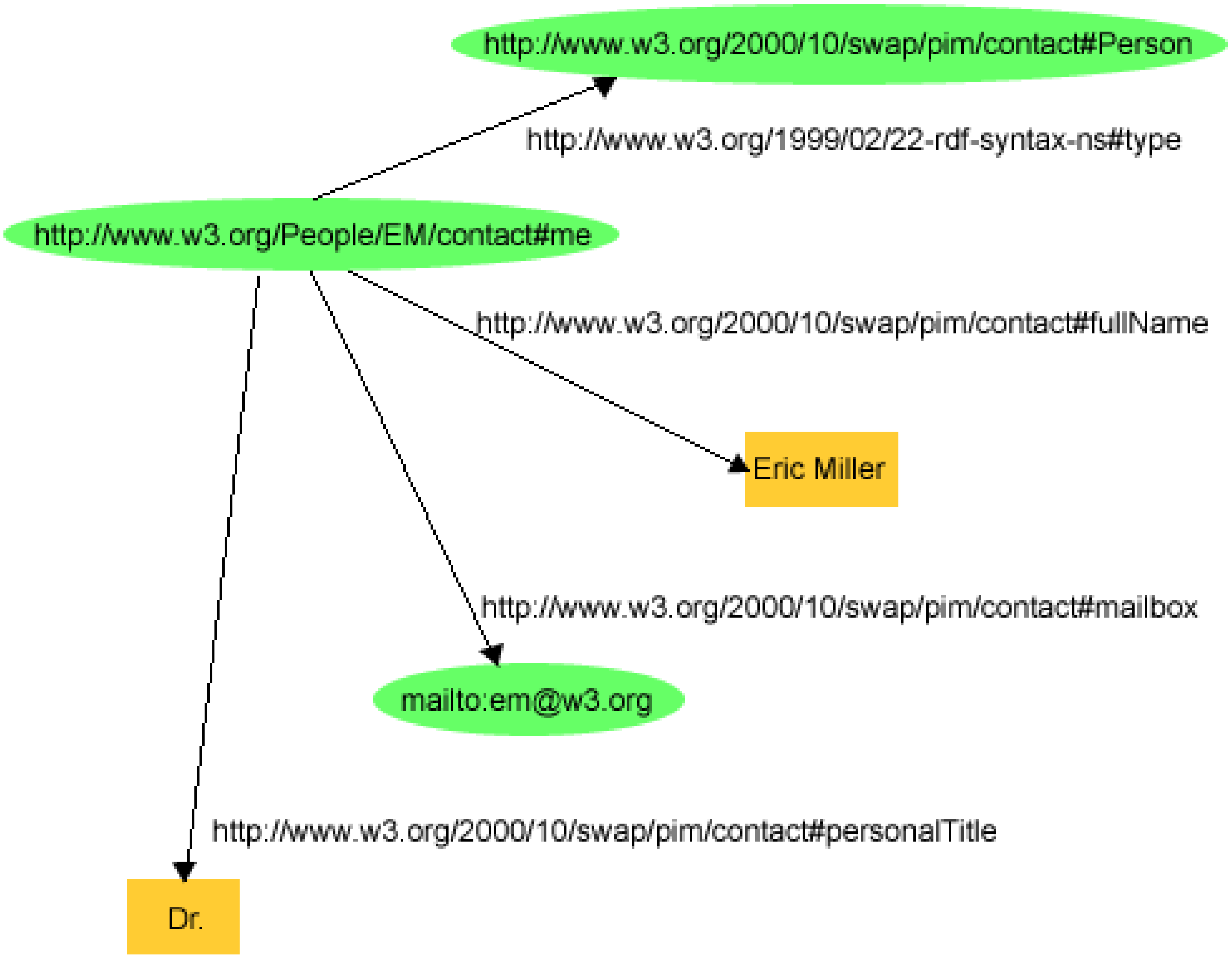
But...

- Relational database tables may have an arbitrary number of columns
- First order logic predicates may have an arbitrary number of places (arguments)
- RDF triples may only have **one** subject and **one** object
    - Complex statements have to be decomposed for representation as RDF triples

Example

- Represent in RDF the following statement
- "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."

Example


http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
Eric Miller
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:em@w3.org
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Dr.

URIs represent (almost) everything

- Nodes (subject or object)
  - □ individuals: Eric Miller, identified by `http://www.w3.org/People/EM/contact#me`
  - □ kinds of things: Person, identified by `http://www.w3.org/2000/10/swap/pim/contact#Person`
  - □ values of properties: `mailto:em@w3.org` as the value of the mailbox property
- Predicates
  - □ properties of things: mailbox, identified by `http://www.w3.org/2000/10/swap/pim/contact#mailbox`

Non-URI information

- Literals (only as objects, never as subjects)
  - The name `"Eric Miller"`
  - The title `"Dr."`
  - May be localized
    - `"Dr."@en`
    - `"Dott."@it`
  - May be typed with XMLSchema data types
    - `"27"^^<http://www.w3.org/2001/XMLSchema#integer>`
    - `"37"^^xsd:integer`
    - `"1999-08-16"^^xsd:date`

URIs are more than URLs

- URL = uniform resource *locator*
  - Designed to locate, and <u>retrieve</u>, resources on the web
- URI = uniform resource *identifier*
  - More general
  - Identifies also resources that do <u>not</u> have a network location
  - Every person or organization can independently create URIs, and use them to identify “things” (either concrete or abstract)

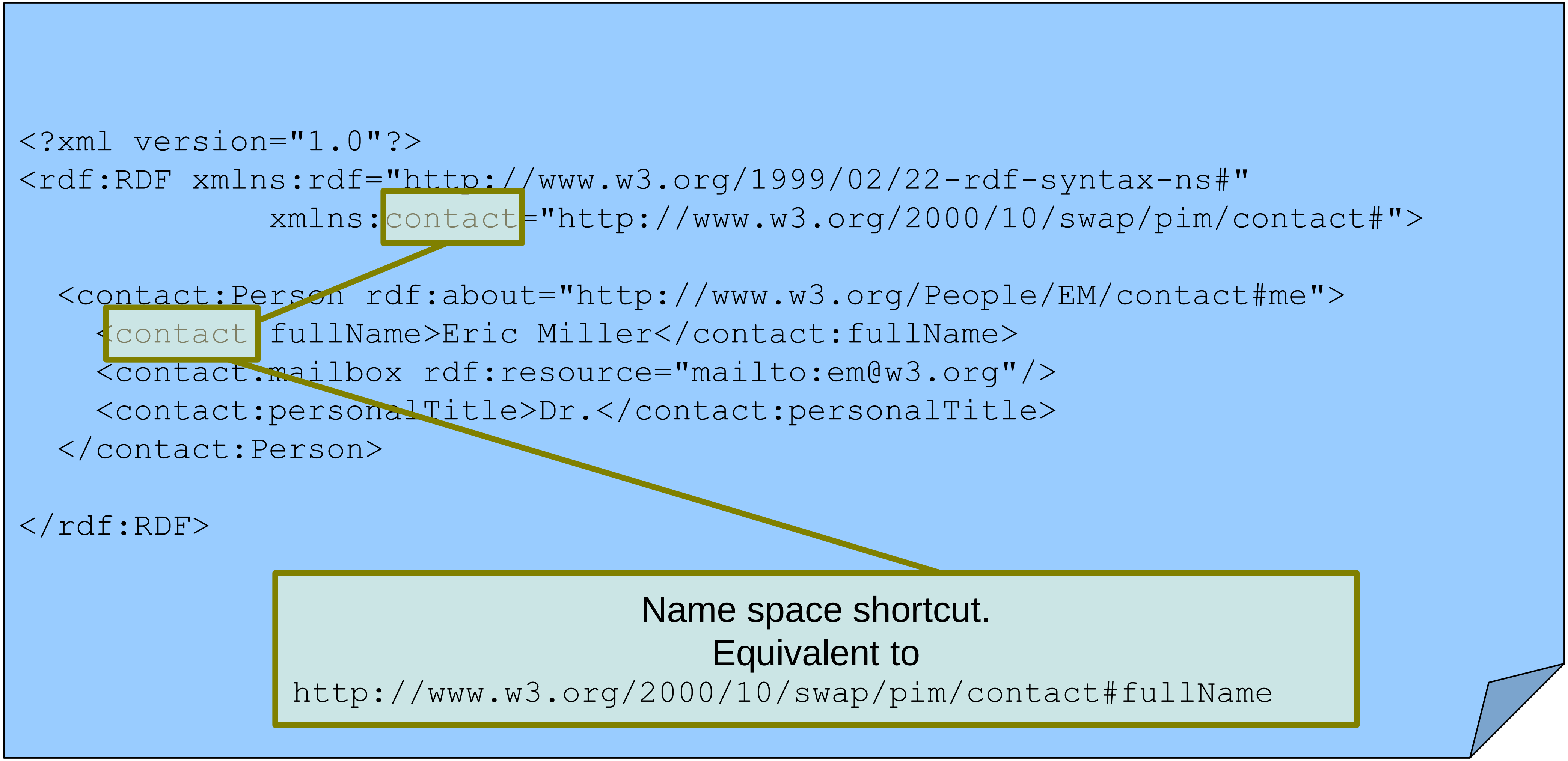
URIref = URI#fragmet

- URIref = URI reference
- A single URI may define many different resources
  - E.g., the URI references an RDF file with many definitions
- To identify a single **fragment** inside the URI, we use the ‘#’ notation
  - E.g., http://example.org/index#person

RDF/XML Syntax

```
<?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>
```

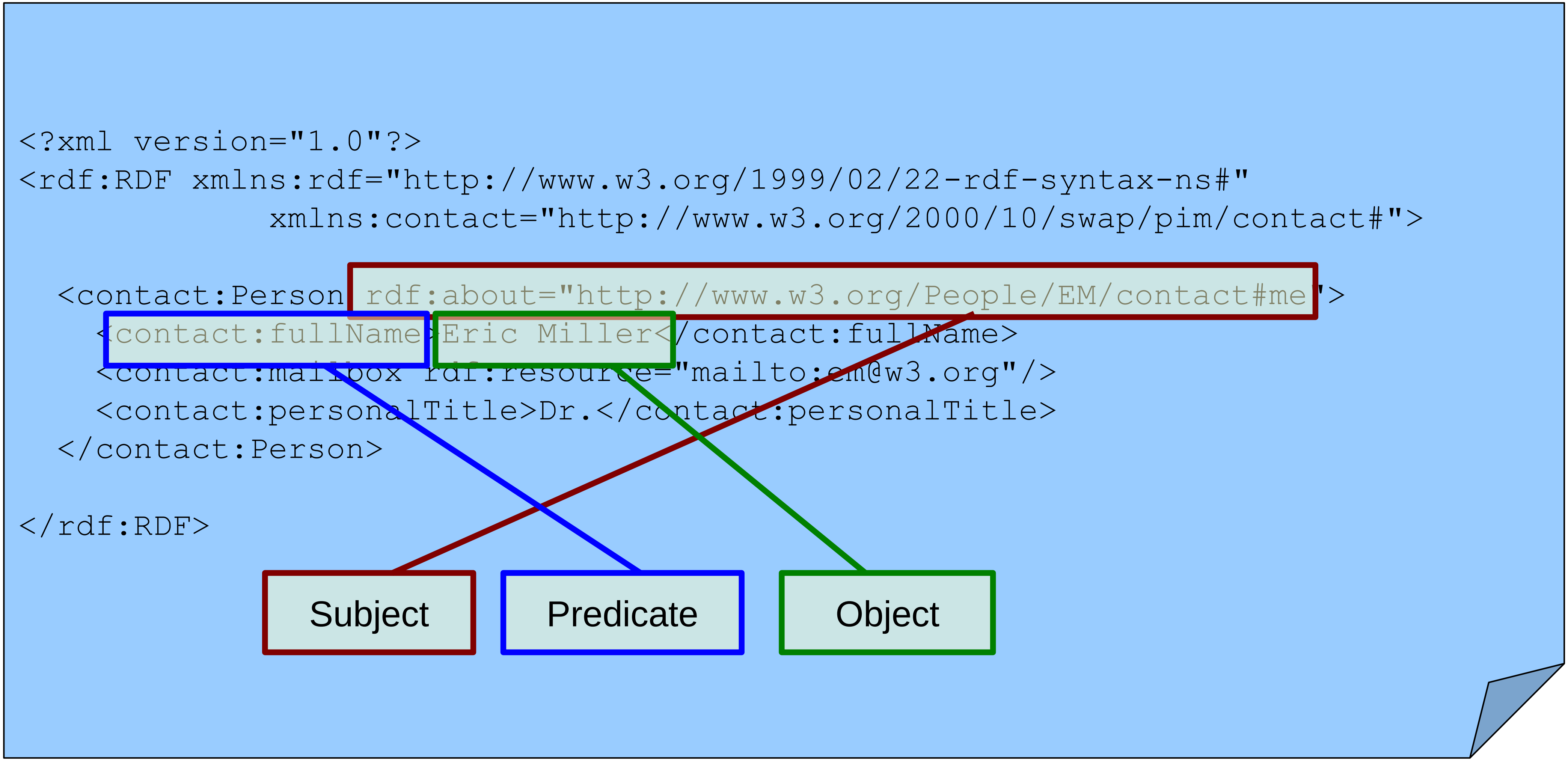
RDF/XML Syntax



```
<?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>
```

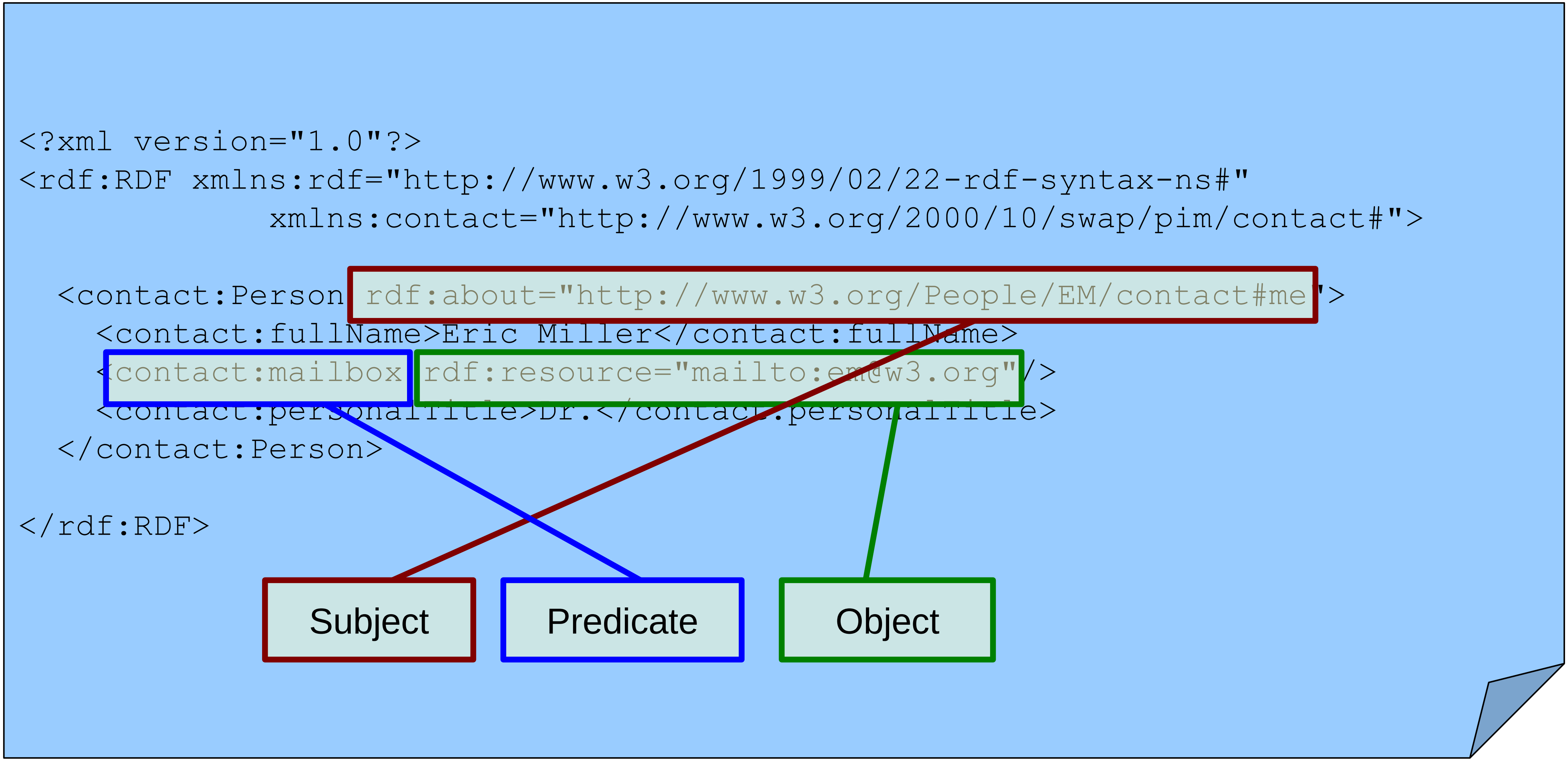
Name space shortcut.
Equivalent to
`http://www.w3.org/2000/10/swap/pim/contact#fullName`

RDF/XML Syntax



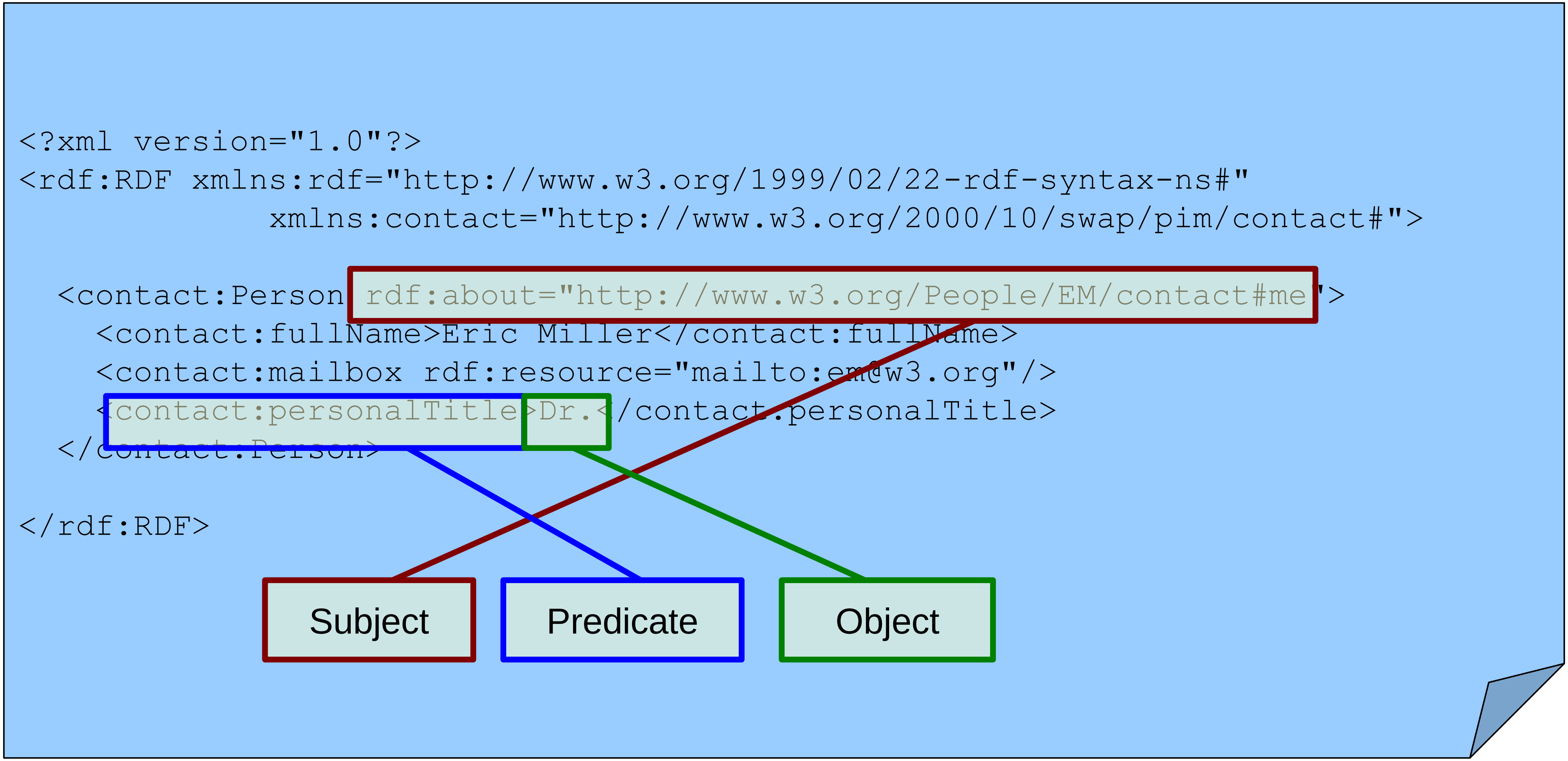
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <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>
```

RDF/XML Syntax



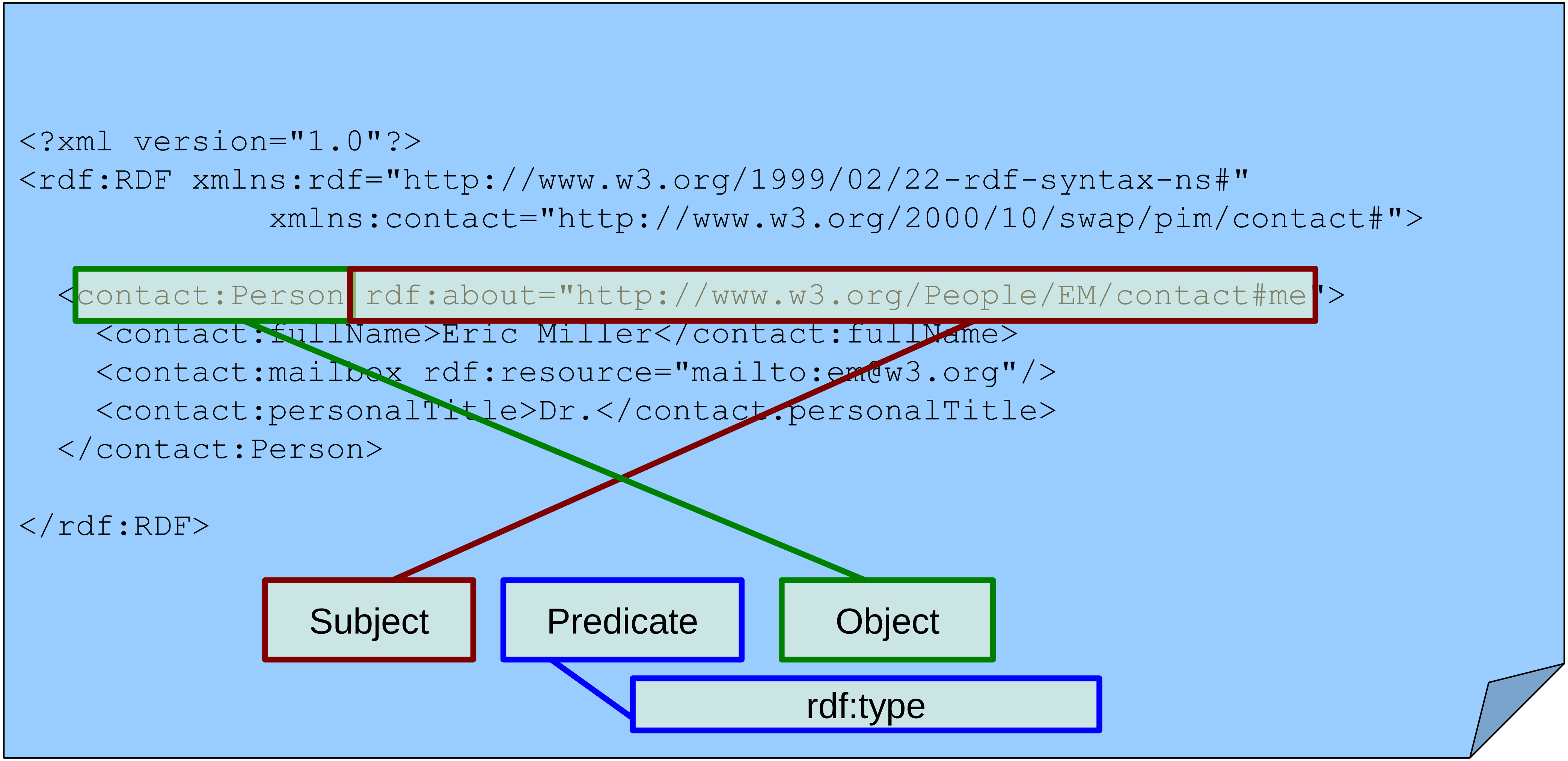
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <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>
```

RDF/XML Syntax



```
<?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>
```

RDF/XML Syntax



```
<?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>
```

Subject

Predicate

Object

rdf:type

“Triple” or “Turtle” notation

```
<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#fullName>
"Eric Miller" .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#mailbox>
<mailto:em@w3.org> .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#personalTitle>
"Dr." .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2000/10/swap/pim/contact#Person> .
```

“Triple” or “Turtle” notation (abbreviated)

```
w3people:EM#me contact:fullName "Eric Miller" .

w3people:EM#me contact:mailbox <mailto:em@w3.org> .

w3people:EM#me contact:personalTitle "Dr." .

w3people:EM#me rdf:type contact:Person .
```

More details on the turtle syntax and further abbreviations will be shown in the SPARQL chapter

Example

```
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntaxns# .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/#> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  :editor [
    :fullName "Dave Beckett";
    :homePage <http://purl.org/net/dajobe/>
  ] .
```

Hands-on exercise

- Model as an RDF graph a subset of the following assertions:
  - Oracle Corporation (NASDAQ: ORCL) and Sun Microsystems (NASDAQ: JAVA) announced today they have entered into a definitive agreement under which Oracle will acquire Sun common stock for $9.50 per share in cash.
  - […]
  - Sun Microsystems, Inc. (NASDAQ: JAVA) develops the technologies that power the global marketplace. [...] Sun can be found in more than 100 countries and on the Web at http://www.sun.com.
  - Oracle (NASDAQ: ORCL) is the world's largest enterprise software company. For more information about Oracle, please visit our Web site at http://www.oracle.com.

Source: http://www.oracle.com/us/corporate/press/018363

Outline

- RDF Design objectives
- RDF General structure
- **Serialization: XML**
- XML Serialization
- Semantic features
- RDF Schema

RDF vocabularies

- A set of URIref is called *vocabulary*
- Common vocabularies collect URIrefs under the same *name space*, so that all nodes may be reached with QNames such as:
  - prefix:nodeName
- The name space is chosen to represent the organization responsible for the definitions
- Every elaboration in RDF must first *resolve all prefixes*, so that only **absolute URIs** are used by the algorithms

Common prefixes

- prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
- prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
- prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
- prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
- prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#
- prefix ex:, namespace URI: http://www.example.org/ (or http://www.example.com/)

Vocabulary reuse

- Extremely easy to re-use other vocabularies in our RDF graph... just define a prefix to point to the proper name space
- When using a predicate, **always** check if its semantics is already satisfied by some property defined in well-known vocabularies
  - Never re-define, with a different URIref, some already existing predicate
- The same applies for names, but with somewhat less importance.

Hands-on: let's explore some useful vocabularies...

- Dublin Core
  - Specification: http://dublincore.org/documents/dces/
  - Namespace: xmlns:dc="http://purl.org/dc/elements/1.1/"
- FOAF
  - Specification: http://xmlns.com/foaf/spec/
  - Namespace: xmlns:foaf="http://xmlns.com/foaf/0.1/"

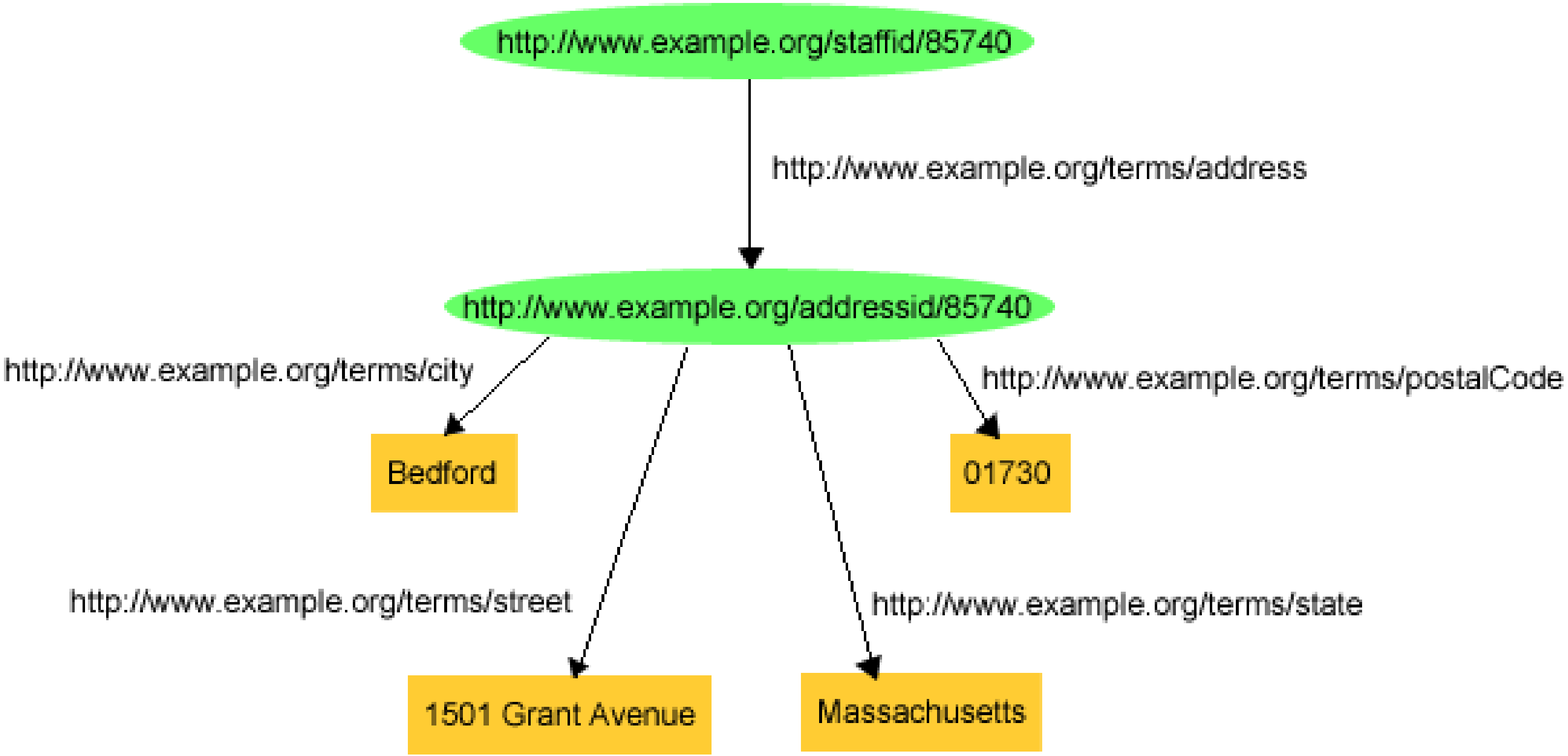
Hands-on: let's explore some useful vocabularies...

- Recent Dublin Core enhancement: DCMI Metadata Terms
  - Specification: http://dublincore.org/documents/dcmi-terms/
  - Namespace: xmlns:dcterms="http://purl.org/dc/terms/"
- RSS 1.0
  - Information: http://en.wikipedia.org/wiki/RSS_(file_format)

Blank nodes

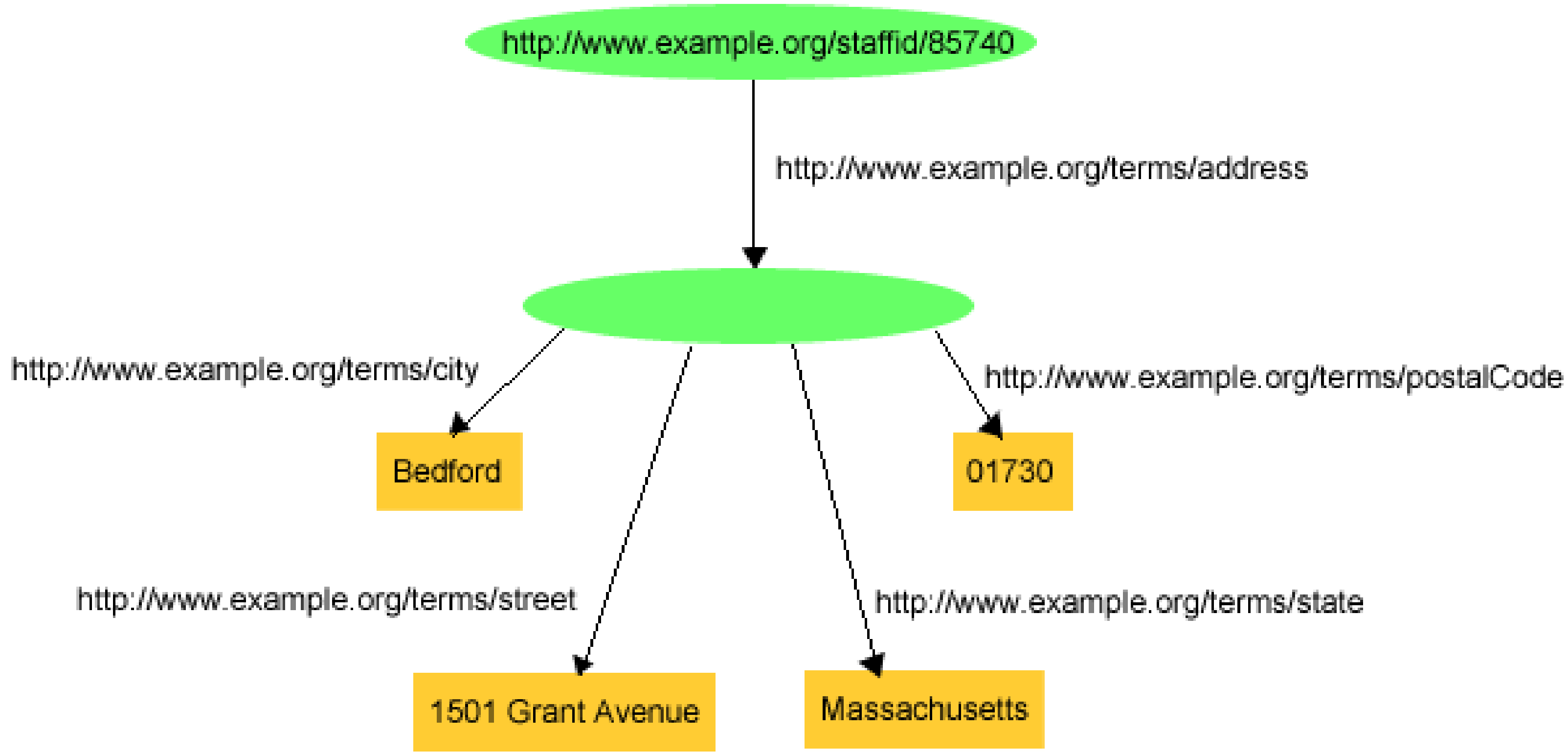
- RDF just supports triples, i.e., binary relationships
- Higher-order relationships must be broken down into many binary pieces
- Breaking down means creating additional nodes
- Such additional nodes will never be referenced from outside the current sub-graph → the don’t need a name!
- A subject or object may be left “blank”

Example


http://www.example.org/staffid/85740
http://www.example.org/terms/address
http://www.example.org/addressid/85740
http://www.example.org/terms/city
http://www.example.org/terms/postalCode
Bedford
01730
http://www.example.org/terms/street
http://www.example.org/terms/state
1501 Grant Avenue
Massachusetts

```
exstaff:85740       exterms:address       exaddressid:85740 .
exaddressid:85740   exterms:street        "1501 Grant Avenue" .
exaddressid:85740   exterms:city          "Bedford" .
exaddressid:85740   exterms:state         "Massachusetts" .
exaddressid:85740   exterms:postalCode    "01730" .
```

Example – with blank node


http://www.example.org/staffid/85740
http://www.example.org/terms/address
http://www.example.org/terms/city
http://www.example.org/terms/postalCode
Bedford
01730
http://www.example.org/terms/street
http://www.example.org/terms/state
1501 Grant Avenue
Massachusetts

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

Outline

- RDF Design objectives
- RDF General structure
- RDF Vocabularies
- **Serialization: XML**
- Semantic features
- RDF Schema

Details on the XML serialization

- The XML document has a root node <**rdf:RDF**>
- Specifying the subject:
    - <**rdf:Description rdf:about**=”SubjectURIref”>
- Specifying properties, in the body of the rdf:Description tag
    - <ex:propertyName>ObjectLiteral</ex:propertyName>
    - <ex:otherProperty **rdf:resource**=”ObjectURIref” />
- Several triples sharing the same subject may be collected in the same rdf:Description body

Examples

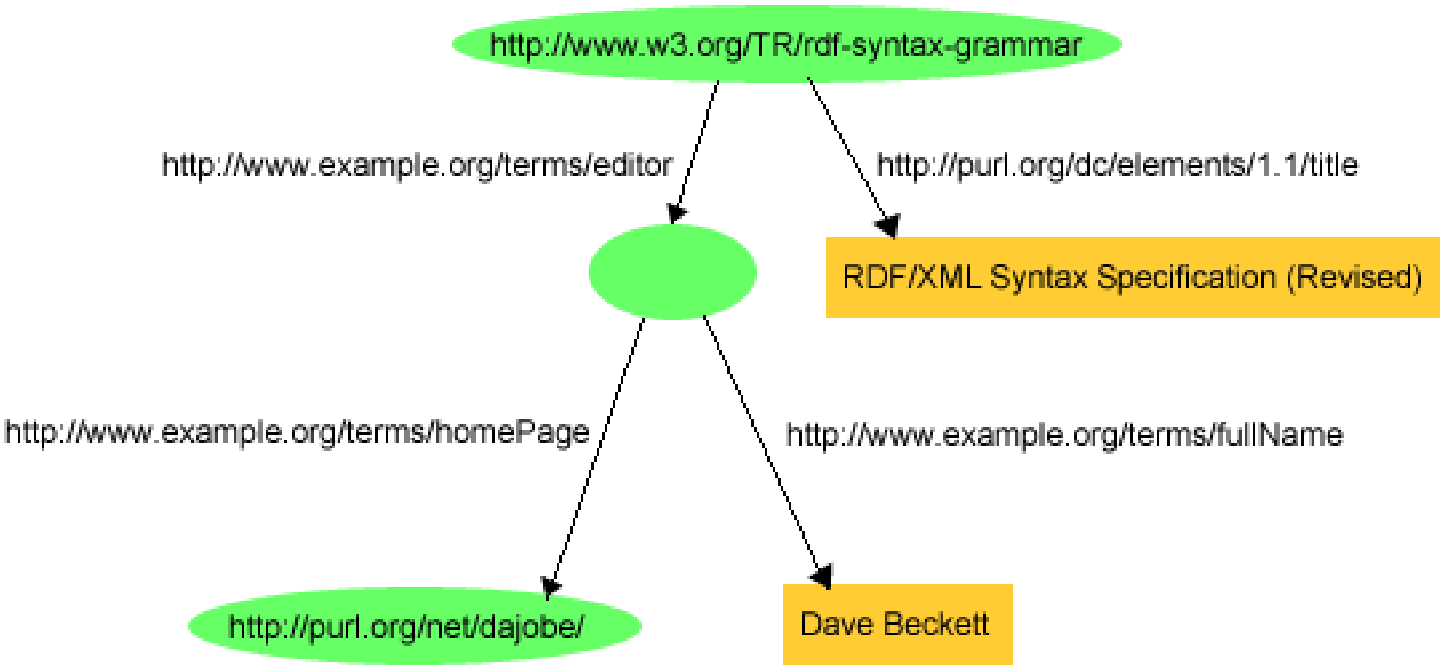
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:exterms="http://www.example.org/terms/">

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

</rdf:RDF>
```

Examples

```
<?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>
```

```
<?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>
```

Blank nodes in XML: rdf:nodeID


http://www.w3.org/TR/rdf-syntax-grammar
http://www.example.org/terms/editor
http://purl.org/dc/elements/1.1/title
RDF/XML Syntax Specification (Revised)
http://www.example.org/terms/homePage
http://www.example.org/terms/fullName
http://purl.org/net/dajobe/
Dave Beckett

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
  <exterms:editor rdf:nodeID="abc"/>
</rdf:Description>

<rdf:Description rdf:nodeID="abc">
    <exterms:fullName>Dave Beckett</exterms:fullName>
    <exterms:homePage rdf:resource="http://purl.org/net/dajobe/"/>
</rdf:Description>
```

Typed literals in XML

```
ex:index.html   exterms:creation-date   "1999-08-16"^^xsd:date .
```

```
<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>
```

Outline

- RDF Design objectives
- RDF General structure
- RDF Vocabularies
- Serialization: XML
- **Semantic features**
- RDF Schema

RDF Data structures

- Containers (unbounded)
  - `rdf:Bag` (unordered)
  - `rdf:Seq` (ordered)
  - `rdf:Alt` (one-of)
  - Semantically equivalent, the different beween Bag/Seq/Alt is only in its “intended usage”
  - Does not limit the member elements to the ones declared
- Collections (bounded)
  - `rdf:List`
  - Only the mentioned elements are part of the collection

Reification

- It may be sometimes useful to assert a statement *about* another statement.
  - For example, I want to say *who* added a fact (a triple) to my set of statements
- In this case, instead of writing the triple, we *describe* the triple by
  - **Giving a name** to the statement (`rdf:Statement`)
  - Giving the **elements of the triple** with `rdf:subject`, `rdf:predicate`, `rdf:object`

Example

```
exproducts:item10245   exterms:weight   "2.4"^^xsd:decimal .
```

reification

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

… and now the statement has a URIref: `this.rdf#triple12345`

Example (cont.)

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

```
exproducts:triple12345   dc:creator       exstaff:85740 .
```

We expressed the `dc:creator` of the previous statement!

Entailment

- An RDF expression A is said to **entail** another RDF expression B if every possible arrangement of things in the world that makes A true also makes B true. On this basis, if the truth of A is presumed or demonstrated then the truth of B can be inferred.
- The mechanism for defining formal semantics for RDF
- The ultimate mechanism for creating reasoning engines in the semantic web
- Never asserts anything about “the things in the world”, only about the propagation of truth in RDF statements/assertions

More on this in the RDF Semantics chapter!

Outline

- RDF Design objectives
- RDF General structure
- RDF Vocabularies
- Serialization: XML
- Semantic features
- **RDF Schema**

RDF Schema

- Special RDF vocabulary for describing the properties and the content of... RDF vocabularies
- Think of a definition (**schema**) of the nodes and predicates used in an RDF document.
  - However, this definition is expressed in RDF, too, by using the **RDFS vocabulary**
- With RDFS we may restrict the usage of RDF nodes and predicates, by introducing coherency and a sort of data types
- RDF Schema provides a **type system** for RDF

RDFS nature

- RDFS does **not** specify a vocabulary of *descriptive* properties such as “author”
- RDFS specifies **mechanisms** that may be used to name and describe properties and the classes of resource they describe
- Similar to the type systems of object-oriented programming languages, but:
  - OO languages define a class in terms of the properties its instances may have
  - RDFS describes properties in terms of the classes of resource to which they apply (domain & range)

Example

- OO language
  - define a class `eg:Book`
  - with an attribute called `eg:author`
  - of type `eg:Person`
- RDFS
  - define the `eg:author` property
  - to have a domain of `eg:Document`
  - and a range of `eg:Person`

- Why?
  - Easy for others to subsequently define additional properties with a domain of `eg:Document` or a range of `eg:Person`
  - This can be done without the need to re-define the original description of these classes
  - It allows anyone to extend the description of existing resources, one of the architectural principles of the Web

Defining Classes in RDFS

- `rdf:type`
  - Defines the ‘type’ of the subject node
  - The object of ‘type’ must be a class
- `rdfs:Class`
  - The set of all possible classes
  - A class is any resource having an `rdf:type` property whose value is the resource `rdfs:Class`

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

exthings:companyCar   rdf:type   ex:MotorVehicle .
```

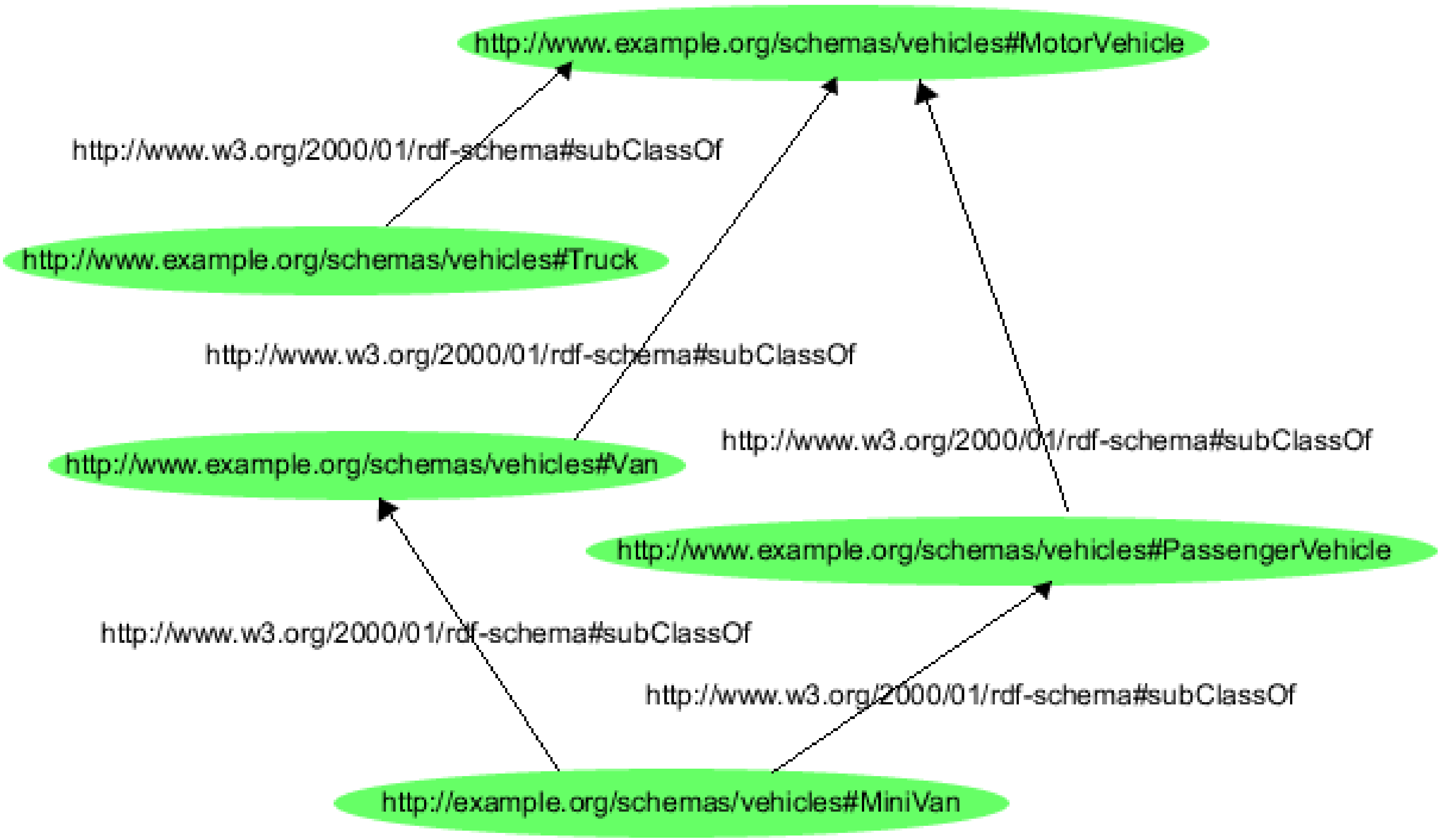
Defining class hierarchies

- `rdfs:subClassOf`
  - Defines a narrower class
  - Any instance of class `ex:Van` is also an instance of class `ex:MotorVehicle`
  - A transitive predicate

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

exthings:companyCar    rdf:type    ex:MotorVehicle .
```

```
ex:Van      rdf:type    rdfs:Class .
ex:Truck    rdf:type    rdfs:Class .

ex:Van    rdfs:subClassOf    ex:MotorVehicle .
```

Class hierarchies


http://www.example.org/schemas/vehicles#MotorVehicle
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.example.org/schemas/vehicles#Truck
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.example.org/schemas/vehicles#Van
http://www.example.org/schemas/vehicles#PassengerVehicle
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://example.org/schemas/vehicles#MiniVan

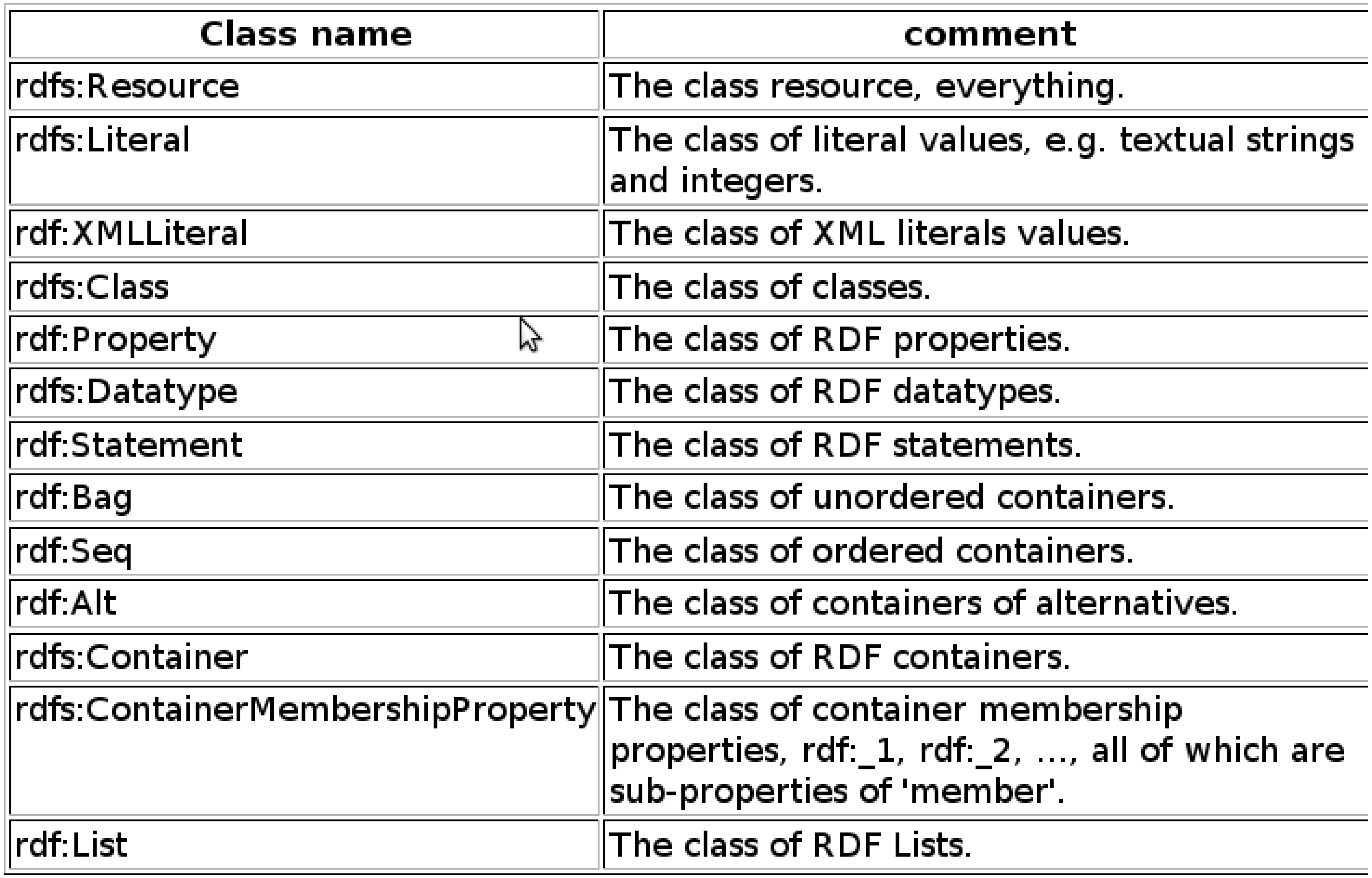
Defining properties in RDFS

- `rdf:Property`
  - Any URIref used as a predicate has an `rdf:type` of `rdf:Property`
- `rdfs:domain`, `rdfs:range`
  - Define the domain and the range of the property
  - Domain and range are Classes
- `rdfs:subPropertyOf`
  - Defines hierarchies of properties

Example

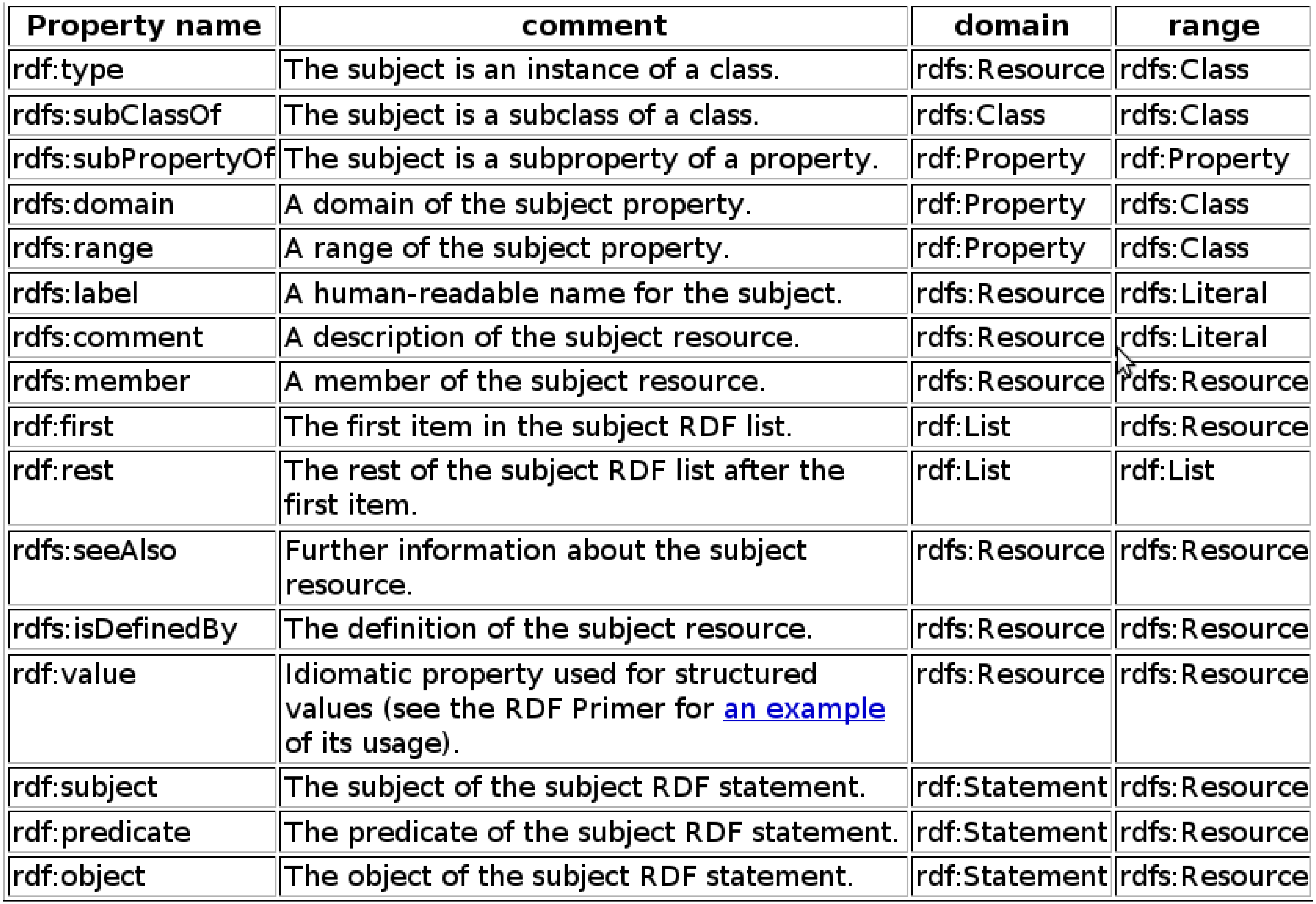
```
<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>

<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"/>
```

RDF/RDFS Classes



Class name	comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

RDF/RDFS Properties



Property name	comment	domain	range
rdf:type	The subject is an instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a subproperty of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	A human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	A description of the subject resource.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdf:List
rdfs:seeAlso	Further information about the subject resource.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:value	Idiomatic property used for structured values (see the RDF Primer for an example of its usage).	rdfs:Resource	rdfs:Resource
rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

References

- RDF Primer – W3C Recommendation 10 February 2004
  - http://www.w3.org/TR/rdf-primer/
- Resource Description Framework (RDF): Concepts and Abstract Syntax – W3C Recommendation 10 February 2004
  - http://www.w3.org/TR/rdf-concepts/
- RDF Vocabulary Description Language 1.0: RDF Schema – W3C Recommendation 10 February 2004
  - http://www.w3.org/TR/rdf-schema/

License