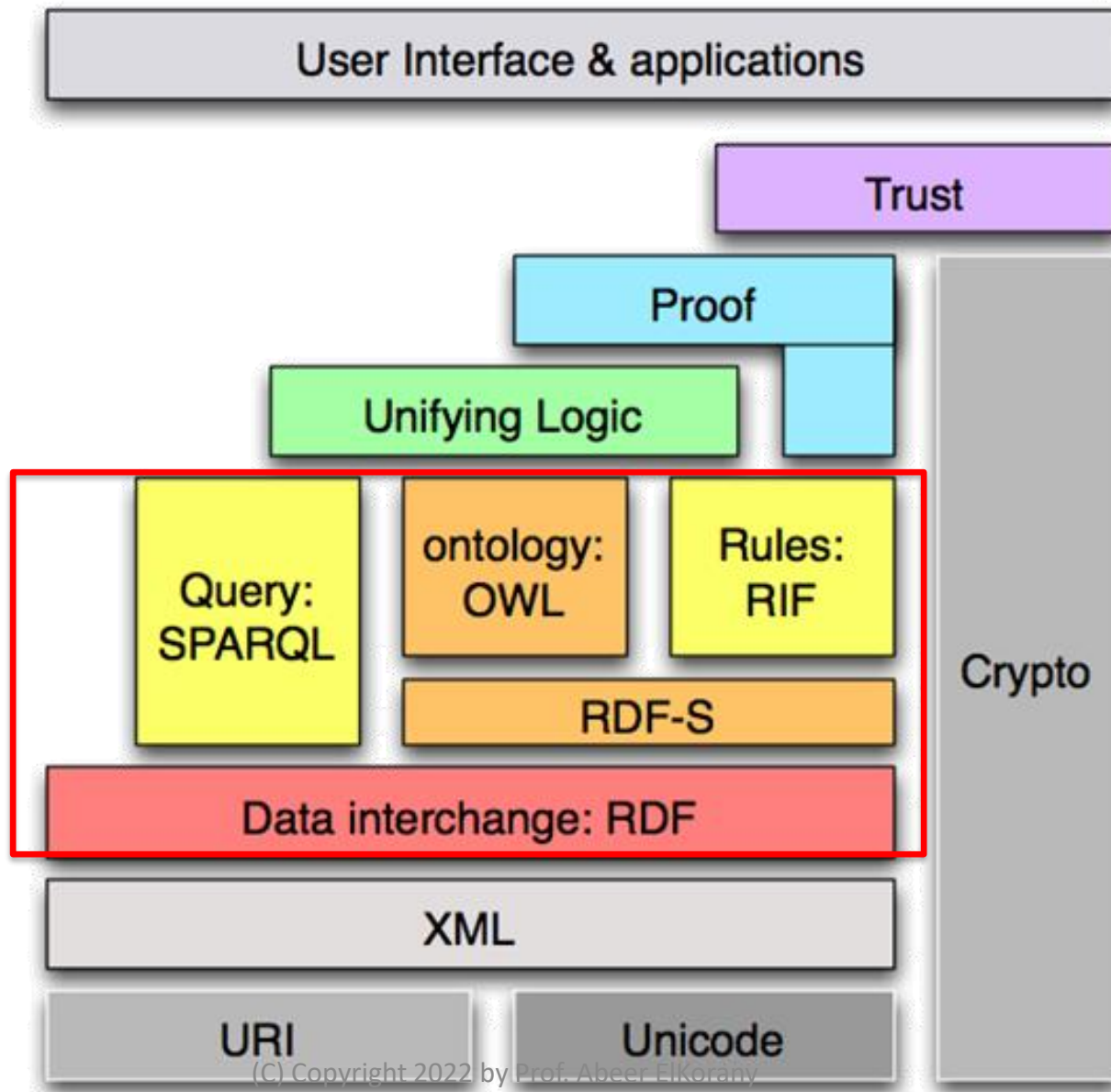


# Ontology Building using RDF

- Basic Ideas of RDF
- RDF Building Blocks
- RDF Data and graph Model
- Blank Node

# Semantic Web Stack

(updated, W3C, 2006)



# What is RDF?

## Resource Description Framework“

- A W3C standard data model since 2004
- originally for describing metadata for web pages, but has grown beyond that
  - structured information
  - universal, machine-readable data exchange format
  - main syntax uses XML for serialization

# RDF Basics

- RDF is based on the idea of identifying resources using **Web identifiers** and describing resources in terms of simple **properties** and property **values**.
- To identify resources, RDF uses **Uniform Resource Identifiers (URIs)** and **URI references (URIrefs)**.
- **Definition:** A **resource** is anything that is identifiable by a URIref.

# RDF Triples

- RDF statements can be written down using **triple notation**. In this notation, a statement is written as follows:

**subject**      **predicate**      **object** .

- **Example:**

```
<http://www.example.org/index.html>  
  <http://purl.org/dc/elements/1.1/creator>  
  <http://www.example.org/staffid/85740> .
```

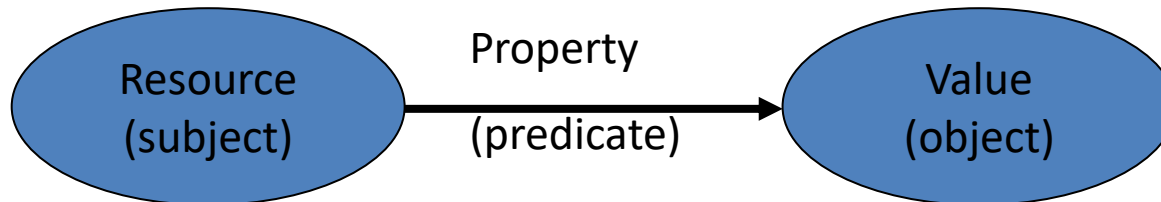
```
<http://www.example.org/index.html>  
  <http://www.example.org/terms/creation-date> "August 16, 1999" .
```

```
<http://www.example.org/index.html>  
  <http://purl.org/dc/elements/1.1/language> "en" .
```

- **Note:** In this notation URIs are written out completely, in angle brackets.

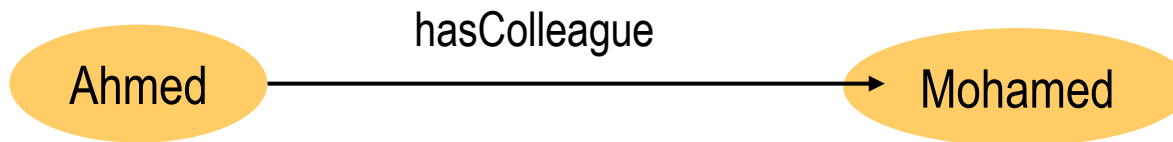
# RDF Triples

- Triple
  - A Resource (Subject) is anything that can have a URI: URIs or blank nodes
  - A Property (Predicate) is one of the features of the Resource: URIs
  - A Property value (Object) is the value of a Property, which can be literal or another resource: URIs, literal, blank nodes (anonymous label)



# Basic Ideas of RDF

- The fundamental elements of RDF are:
  - resources
  - properties
  - Statements
- Basic building block: **subject-predicate-object (triple)**.
  - E.g.: “**Prof. Abeer ElKorany** **teaches** **CS465**”.
  - This is called a statement.
  - View 1: Sentences in the form <subject, predicate, object>
    - <Abeer ElKorany **works-for** FCAI>
  - View 2: Directed graphs with labeled edges

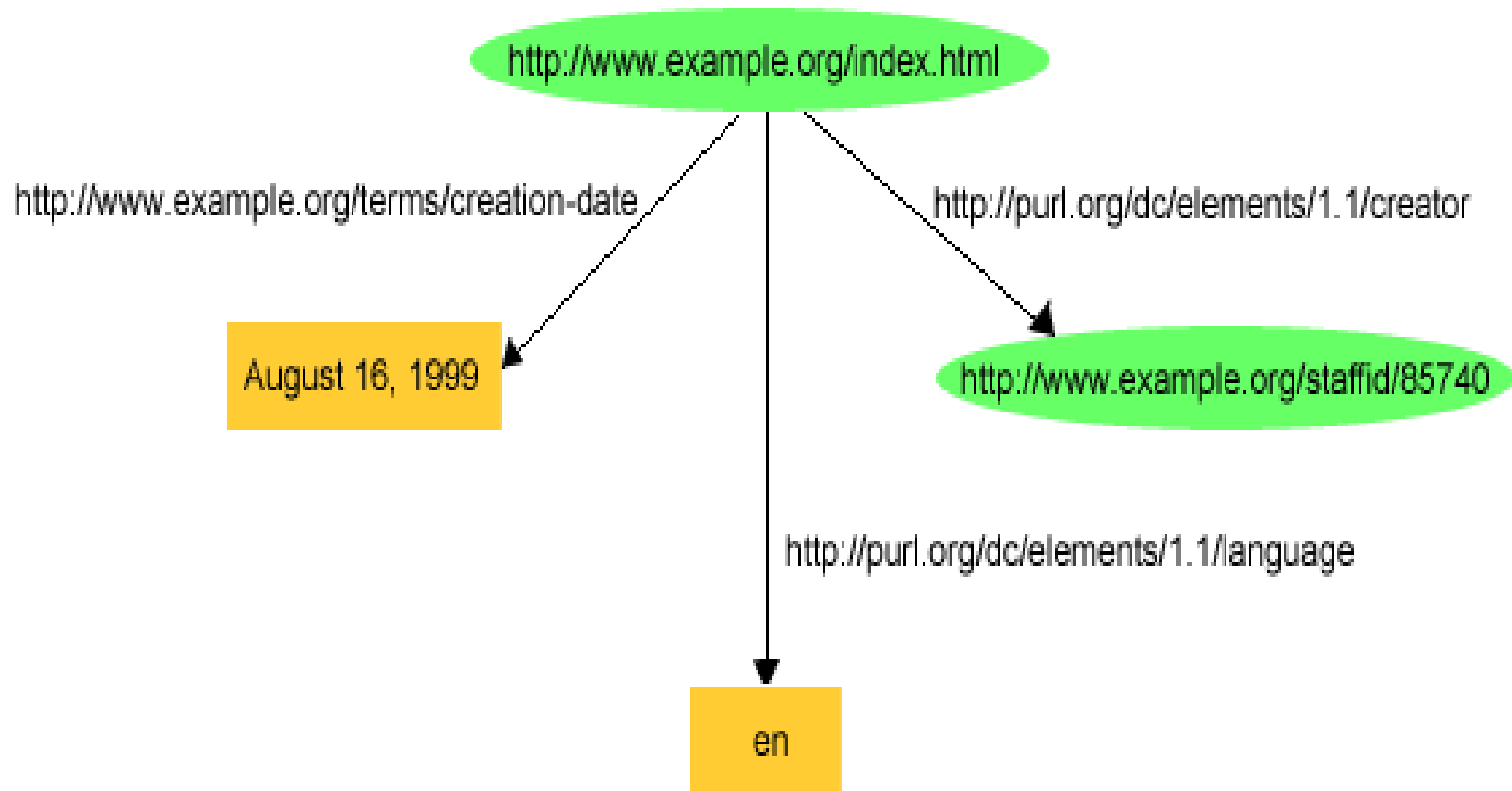


# Example

- Consider the following statements:
  - `http://www.example.org/index.html` has a **creation-date** whose value is August 16, 1999.
  - `http://www.example.org/index.html` has a **language** whose value is English.



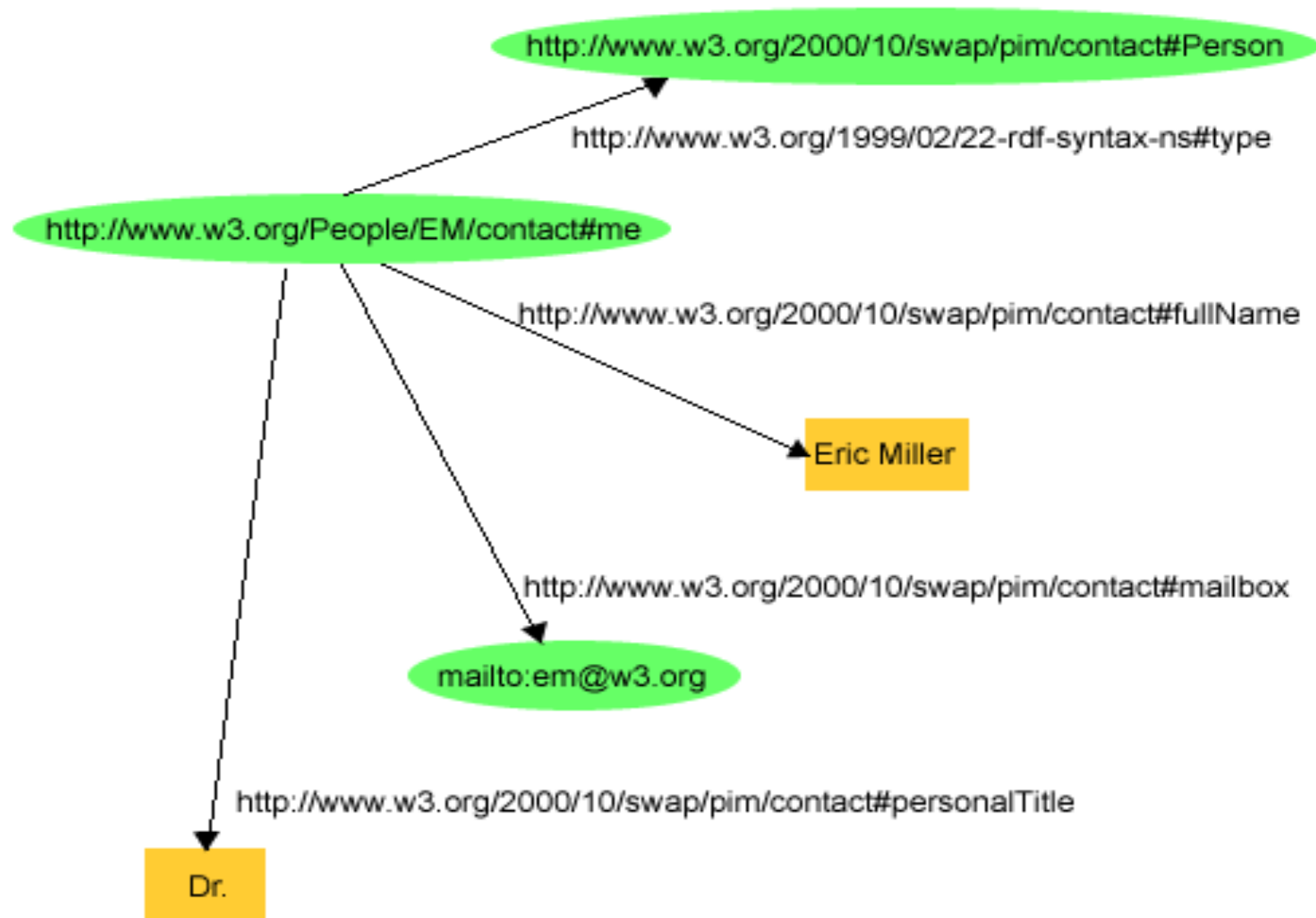
# The RDF Graph of the Example



We can add other info like:

`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 (cont'd)



# Example

`http://www.example.org/index.html` has  
a creator whose value is “John Smith”

- The **subject** is the URL

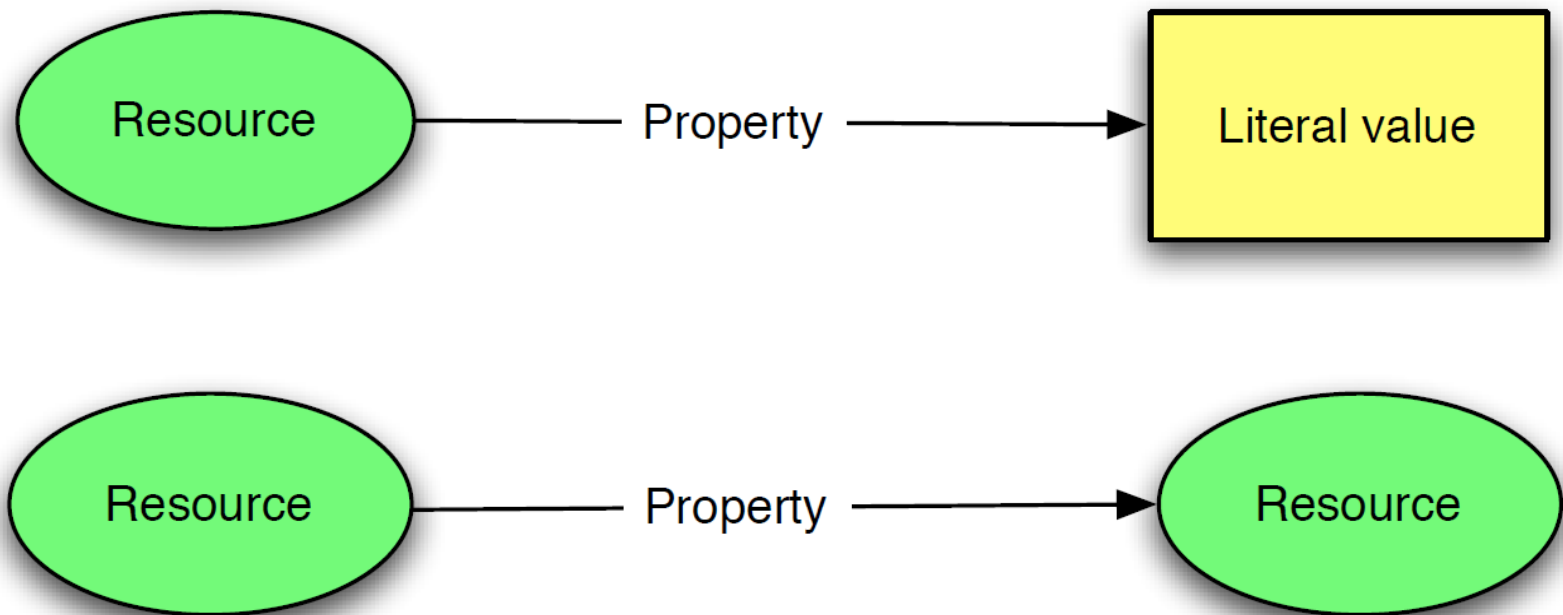
`http://www.example.org/index.html`

- The **predicate** is the word "creator"
- The **object** is the phrase “John Smith”

# What does a URI mean?

- RDF uses a URI (Uniform Resource Identifier) to identify a web resource, and properties to describe the resource.
- Sometimes URIs denote a **web resource**
  - “<http://umbc.edu/~finin/finin.jpg>” denotes a file
  - We can use RDF to make assertions about the resource, e.g., it’s an image and depicts a person with name Tim Finin, ...
- Sometimes **concepts** in the **external world**
  - E.g., “<http://umbc.edu/>” denotes a particular University located in Baltimore.

# RDF Data Model



# Graph Data

1. **Resource nodes:** A resource is anything that can have things said about it (e.g., book, person, etc). In a visual representation, resources are represented by **ovals**.
2. **Literal nodes:** a literal is the same as a value. In a visual representation, literals are represented by **rectangles**.

Resource: “<http://www.cambridge.../about/rob>”

Value of the foaf:name property is "[Rob Gonzalez](#)".

# RDF Triples

- Since a cell is represented with three values, the basic building block for RDF is called the triple.
- A sample triple

Subject	Predicate	Object
Shakespeare	wrote	King Lear

- Subject is what the sentence is about
- Predicate is the property of the subject
- Finally, the object is the value of the property

# Example of RDF Statement

*Ora Lassila is the creator of the resource  
<http://www.w3.org/Home/Lassila>.*

Subject (resource)	<a href="http://www.w3.org/Home/Lassila">http://www.w3.org/Home/Lassila</a>
Predicate (property)	Creator
Object (literal)	“Ora Lassila”





# Statements

- Statements assert the properties of resources
- A statement is an object-attribute-value triple
  - It consists of a resource, a property, and a value
- Values can be resources or **literals**
  - Literals are atomic values (strings)

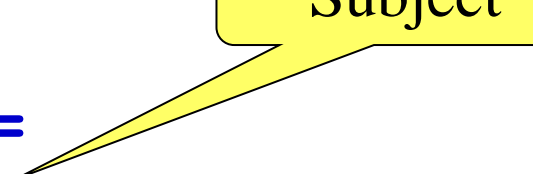
# Statement examples

- Statement: “The author of <http://www.w3schools.com/RDF> is Jan Egil Refsnes”
  - Subject: <http://www.w3schools.com/RDF>
  - Predicate: author
  - Object: Jan Egil Refsnes
- Statement: “The homepage of <http://www.w3schools.com/RDF> is <http://www.w3schools.com>”
  - Subject: <http://www.w3schools.com/RDF>
  - Predicate: homepage
  - Object: <http://www.w3schools.com>

# RDF Example (subject of statement)

*Ora Lassila is the creator of the resource  
<http://www.w3.org/Home/Lassila>.*

```
<rdf:RDF>  
  <rdf:Description rdf:about=  
    "http://www.w3.org/Home/Lassila">  
    <s:Creator>Ora Lassila</s:Creator>  
  </rdf:Description>  
</rdf:RDF>
```



# RDF Example (predicate of statement)

*Ora Lassila is the creator of the resource  
<http://www.w3.org/Home/Lassila>.*

```
<rdf:RDF>
  <rdf:Description about=
    "http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

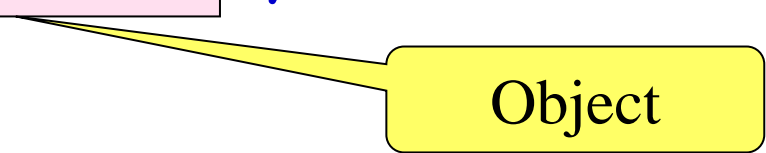


Predicate

# RDF Example (object of statement)

*Ora Lassila is the creator of the resource  
<http://www.w3.org/Home/Lassila>.*

```
<rdf:RDF>
  <rdf:Description about=
    "http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```



The diagram illustrates the object of the statement. A yellow callout box labeled "Object" points to the text "Ora Lassila" within the XML snippet, which is highlighted with a pink background.

# RDF Example (reference to ontology)

*Ora Lassila is the creator of the resource  
<http://www.w3.org/Home/Lassila>.*

```
<rdf:RDF>  
  <rdf:Description about=  
    "http://www.w3.org/Home/Lassila">  
    <s:Creator>Ora Lassila</s:Creator>  
  </rdf:Description>  
</rdf:RDF>
```

a specific namespace prefix as reference to ontology where predicates are defined, e.g.  
xmlns: s="http://description.org/schema/"

# Resources (subject of statement)

- A **resource** is any object (“things” we want to talk about) that can be pointed to by a **URI**
  - a document, a picture, a library book, a real person, authors, books, publishers, places, people, hotels
- Every resource has a **URI**, a Universal Resource Identifier
- A URI can be
  - Web address [http://dbpedia.org/page/Tim\\_Berners-Lee](http://dbpedia.org/page/Tim_Berners-Lee) **OR**
  - some other kind of unique identifier

# Properties (predicate of statement)

- Properties are a special kind of resources that describe relations between resources
  - E.g. “written by”, “age”, “title”, “author-of” etc.
- Properties are also identified by URIs
- Advantages of using URIs:
  - A global, worldwide, unique naming scheme
  - Reduces the homonym problem of distributed data representation



# Namespaces

- A namespace is a URI that is defined once at the beginning and used in sentences later without referring to URI again.

– Example:

***lit*** stands for <http://www.WorkingOntologist.com/Examples/Chapter3/Shakespeare.owl#>

***geo*** stands for <http://www.WorkingOntologist.com/Examples/Chapter3/geography.owl#>

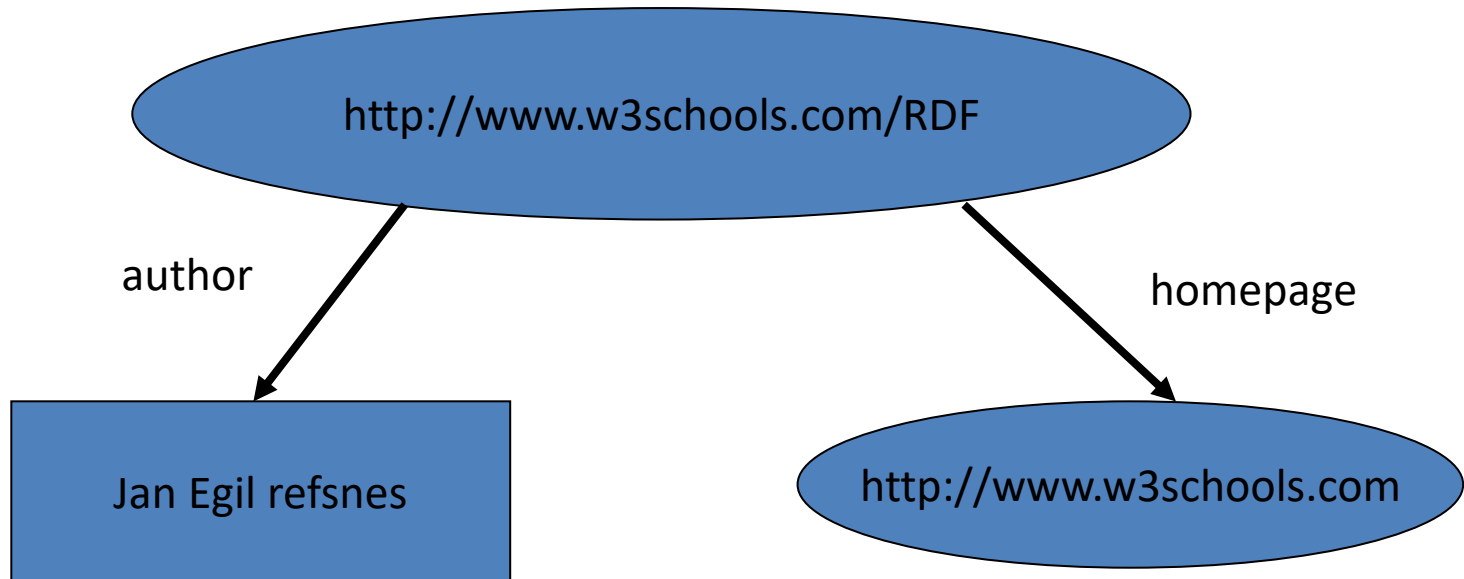
***lit*** and ***geo*** are called Namespaces.

# Replace with namespaces

Subject	Predicate	Object
Shakespeare	lit:wrote	KingLear
Shakespeare	lit:wrote	MacBeth
AnneHathaway	bio:married	Shakespeare
AnneHathaway	bio:livedWith	Shakespeare
Shakespeare	bio:livedIn	Stratford
Stratford	geo:isIn	England
England	geo:partOf	UK
Scotland	geo:partOf	UK

# Simple example of RDF

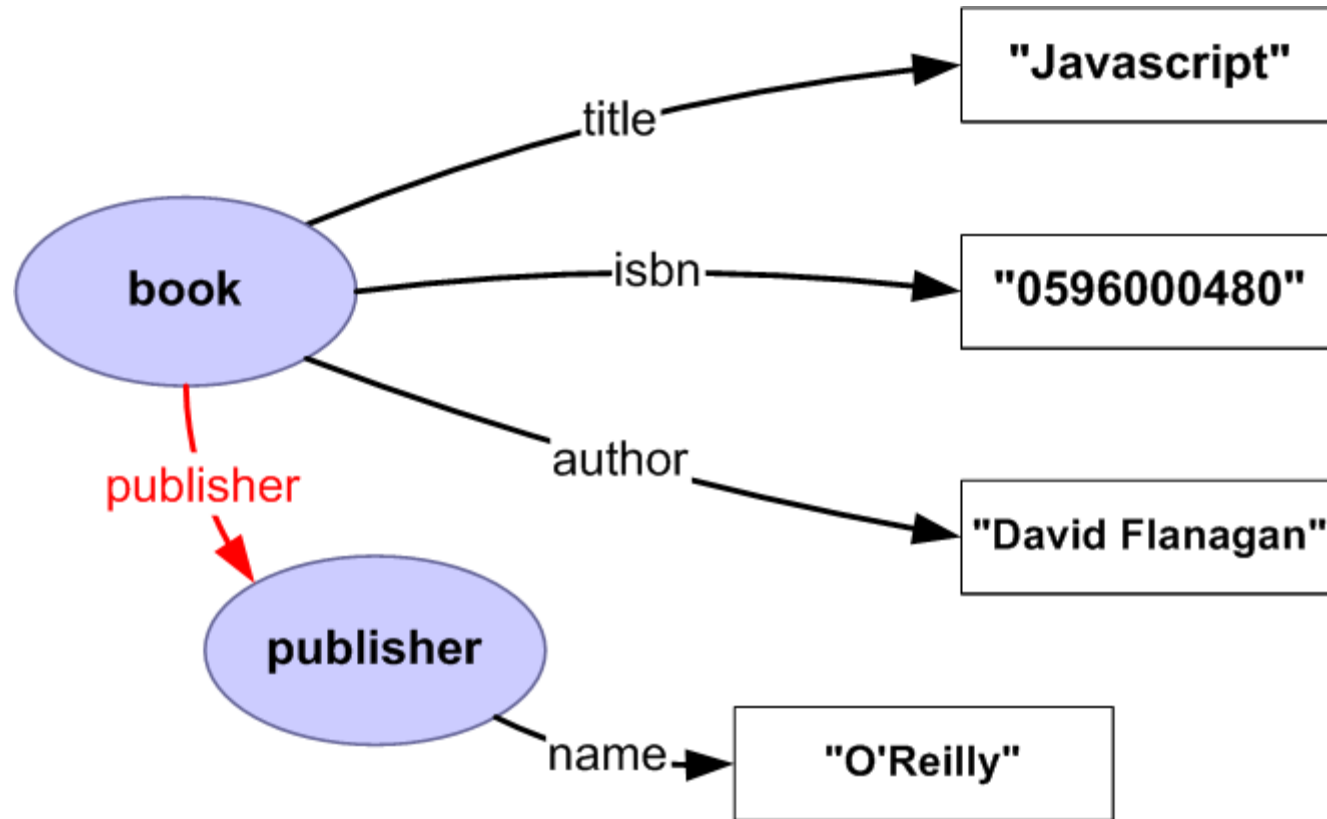
```
<RDF>
  <Description about="http://www.w3schools.com/RDF">
    <si:author>Jan Egil refsnes</author>
    <si:homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```



# The RDF Graph

- An RDF document is an unordered collection of triples.
- The subject of one triple can be the object of another.
- So the result is a directed, labelled graph.
- A triple's object can also be a literal, e.g., a string.

# RDF is a graph

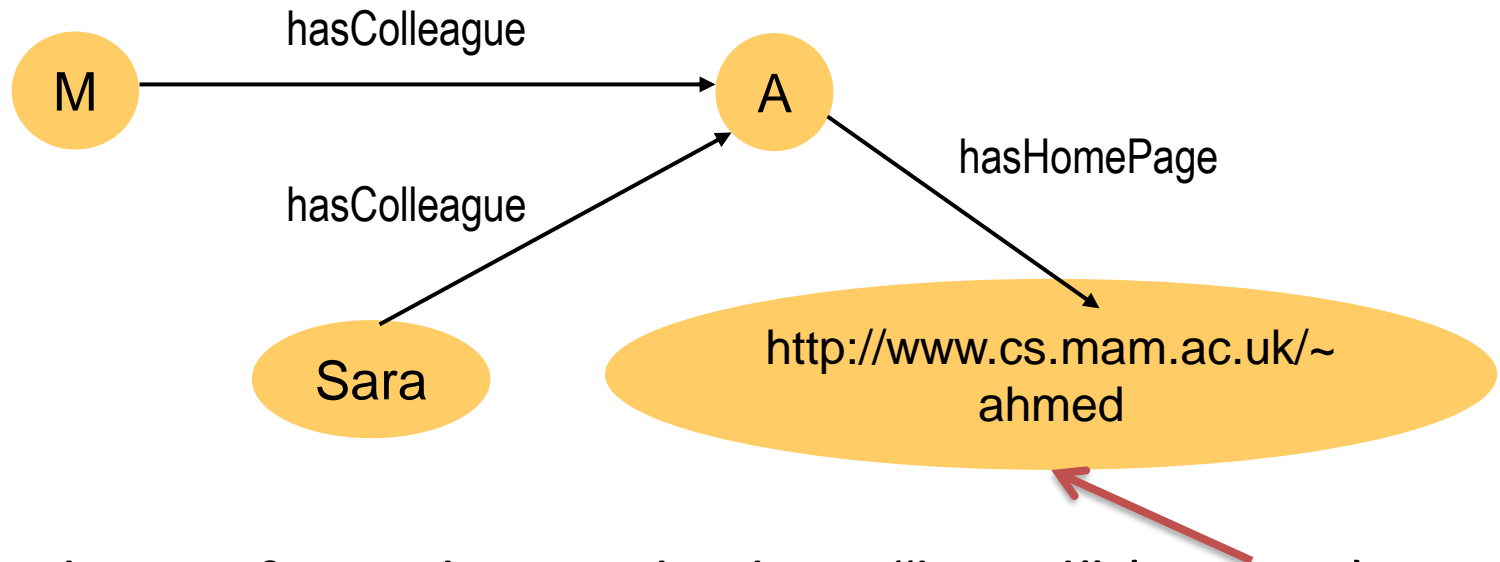


# RDF Graph(cont.)

- Graphically, RDF models statements by **nodes** and **arcs** in a **graph**.
- In the RDF graph notation, a **statement** is represented by:
  - a node for the subject
  - a node for the object
  - an arc for the predicate, directed from the subject node to the object node.
- A **node** may be identified by a **URIref** or it can be a **literal** (it can also be a **blank node**; will explain this later).
- An **arc** is identified by a **URIref**.

# The RDF Graph (cont.)

- The subject of a statement can be the object of another.
- Such collections of statements form a directed, labeled graph.



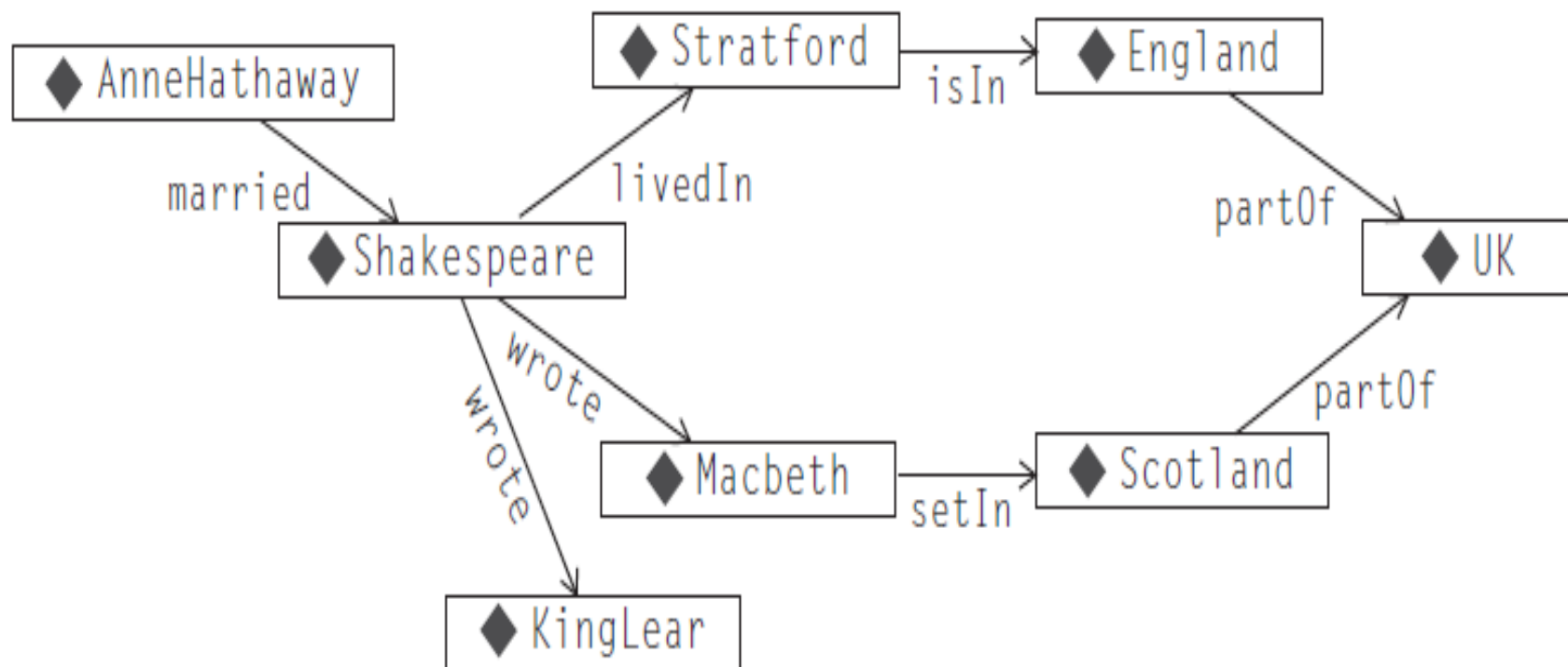
- The object of a triple can also be a “literal” (a string).

# RDF Tabular Representation

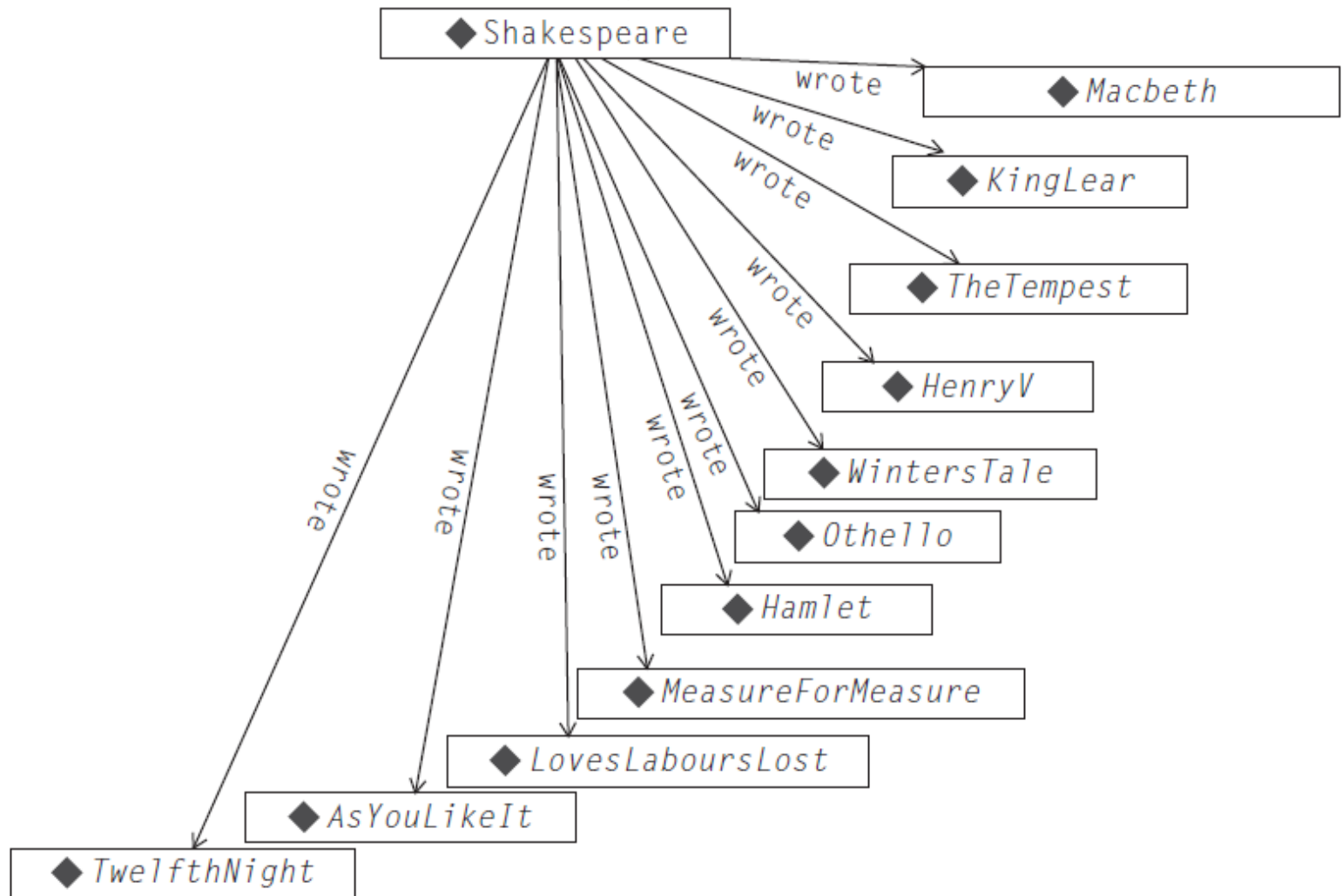
Subject	Predicate	Object
Shakespeare	wrote	King Lear
Shakespeare	wrote	Macbeth
Anne Hathaway	married	Shakespeare
Shakespeare	lived in	Stratford
Stratford	is in	England
Macbeth	set in	Scotland
England	part of	The UK
Scotland	part of	The UK



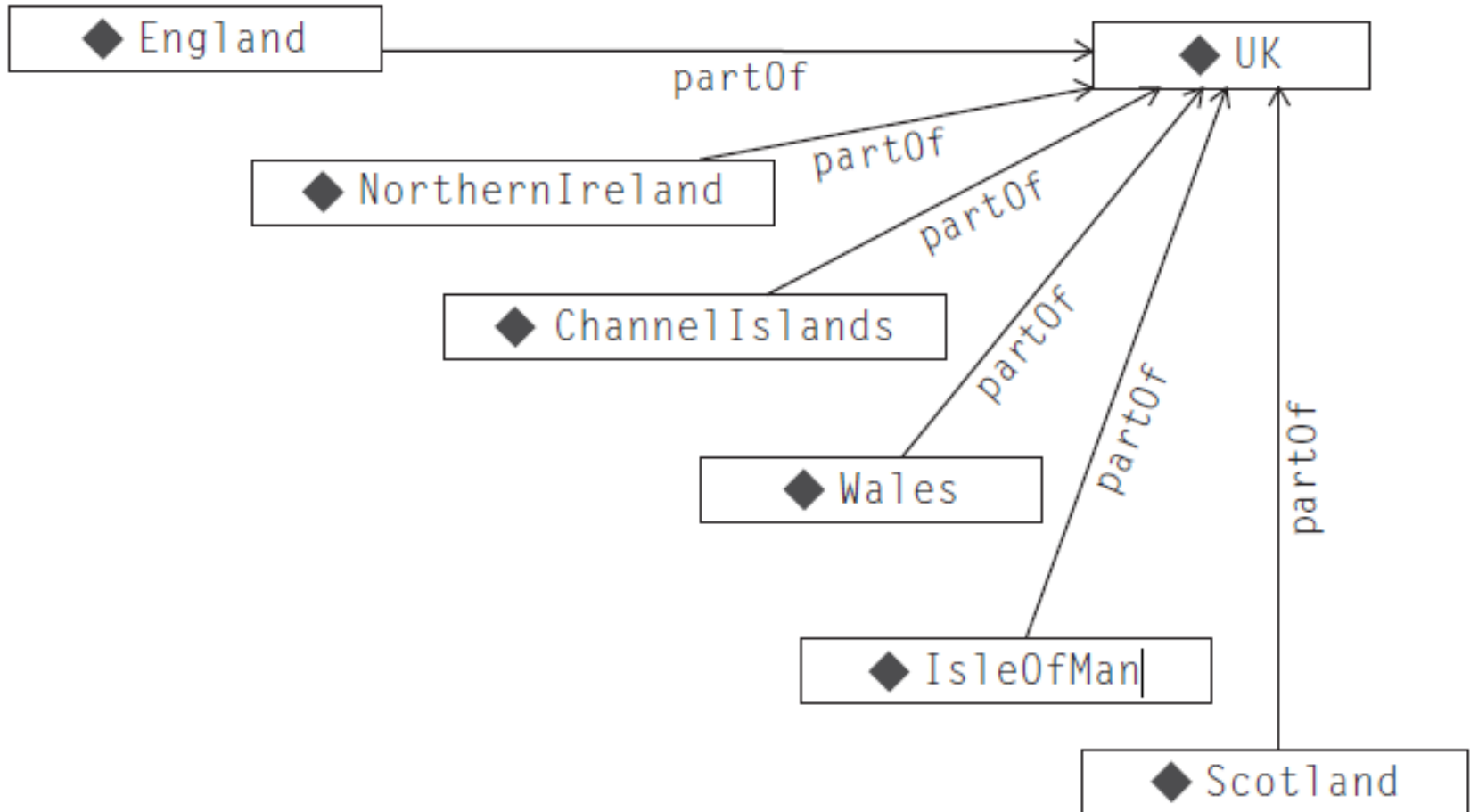
# RDF Tabular as a Graph



# Expanding of data



# More data



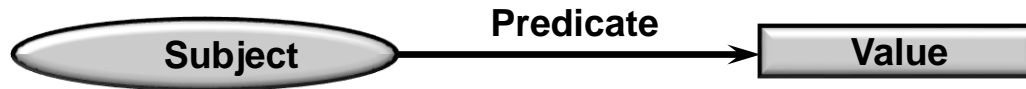
# RDF Syntax

- RDF has an XML syntax that has a specific meaning.
- An RDF document starts by an **rdf:RDF** element
  - The content of that element is a number of statements.
- A **namespace** mechanism is used
  - Disambiguation
  - Namespaces are expected to be RDF documents defining resources that can be reused
  - Large, distributed collections of knowledge

# RDF syntax(cont.)

- Starting with `<rdf:RDF>` and end with `</rdf:RDF>`
- `<rdf:RDF>`: the root element
- `<rdf:Description>` is the main element to define the subject, predicate and object of the statement
- RDF Namespace
  - <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
- File format: `.rdf`

# Template/frame for RDF statement



- A subject can contain several property elements
- Object-descriptions can be used as subject-descriptions for further triples

```
<rdf:Description rdf:about="http://semantic-web-book.org/uri">
  <ex:title>Foundations of Semantic Web Technologies</ex:title>
  <ex:publishedBy>
    <rdf:Description rdf:about="http://crcpress.com/uri">
      <ex:name>CRC Press</ex:name>
    </rdf:Description>
  </ex:publishedBy>
</rdf:Description>
```

# RDF Syntax

- Statements are defined using the tag Descriptions “**rdf:Description**”.
- Every description makes a statement about a resource
- Every attribute or nested element inside a **Description** is a **property** of that Resource.
- URIs are in <angle brackets>
  - Literals are ”enclosed in quotes”
  - whitespace (blanks, line feeds) is ignored
- Recall the previous example

```
<rdf:Description about="some.uri/person/mohamed">  
  <hasColleague resource="some.uri/person/ahamed"/>  
</rdf:Description>  
<rdf:Description about="some.uri/person/ahmed">  
  <si:hasHomePage>http://www.cs...ac.uk/~ahmed</hasHomePage>  
</rdf:Description>  
<rdf:Description about="some.uri/person/sara">  
  <si:hasColleague resource="some.uri/person/ahmed"/>  
</rdf:Description>
```

# RDF Complete Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.rechshop.fake/cd/Empire Burlesque">
    <cd:artist>Bob Dylan</cd:artist>
    <cd:country>USA</cd:country>
    <cd:company>Columbia</cd:company>
    <cd:price>10.90</cd:price>
    <cd:year>1985</cd:year>
  </rdf:Description>
  <rdf:Description
    rdf:about="http://www.rechshop.fake/cd/Hide your heart">
    <cd:artist>Bonnie Tyler</cd:artist>
    <cd:country>UK</cd:country>
    <cd:company>CBS Records</cd:company>
    <cd:price>9.90</cd:price>
    <cd:year>1988</cd:year>
  </rdf:Description>
  <!-- more cds -->
</rdf:RDF>
```



# RDF Container Elements

- RDF Container used to collect a number of resources or attributes, about which we want to make statements as a whole It is used to describe group of things
  - <Bag>: a list of members without order
  - <Seq>: a list of members with order
  - <Alt>: a list of members that only one can be selected

# Containers

- Groups of things that has no way to close them :  
<bag> <seq> <alt>
  - <bag> unordered list; duplicates allowed
  - <seq> ordered list; duplicates allowed
  - <alt> list of alternatives; one will be selected
- While “RDF:Collection”: enumerates the specified members (the group only contains the specified members listed in the collection)

# Example <alt>

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

```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:cd="http://www.recshop.fake/cd#">
```

```
<rdf:Description  
  rdf:about="http://www.recshop.fake/cd/Beatles">
```

```
<cd:format>
```

```
<rdf:Alt>
```

```
<rdf:li>CD</rdf:li>
```

```
<rdf:li>Record</rdf:li>
```

```
<rdf:li>Tape</rdf:li>
```

```
</rdf:Alt>
```

Exactly one of these formats

```
</cd:format>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

# Limiting the scope

Collection - describes a group of things represented as a predefined list structure i.e contains only the specified members, no others.

```
<rdf:Description
  rdf:about="http://recshop.fake/cd/Beatles">
  <cd:artist rdf:parseType="Collection">
    <rdf:Description
      rdf:about="http://recshop.fake/cd/Beatles/George"/>
    <rdf:Description
      rdf:about="http://recshop.fake/cd/Beatles/John"/>
    <rdf:Description
      rdf:about="http://recshop.fake/cd/Beatles/Paul"/>
    <rdf:Description
      rdf:about="http://recshop.fake/cd/Beatles/Ringo"/>
  </cd:artist>
</rdf:Description>
```

# Blank node

- Blank node is called anonymous resource in RDF
- Blank node are nodes without URI:
  - Unnamed resources
  - More complex constructs
- Resources for which we do not have a URI as identifier, because
  - We may not know the URI
  - There is no point in minting a URI for them

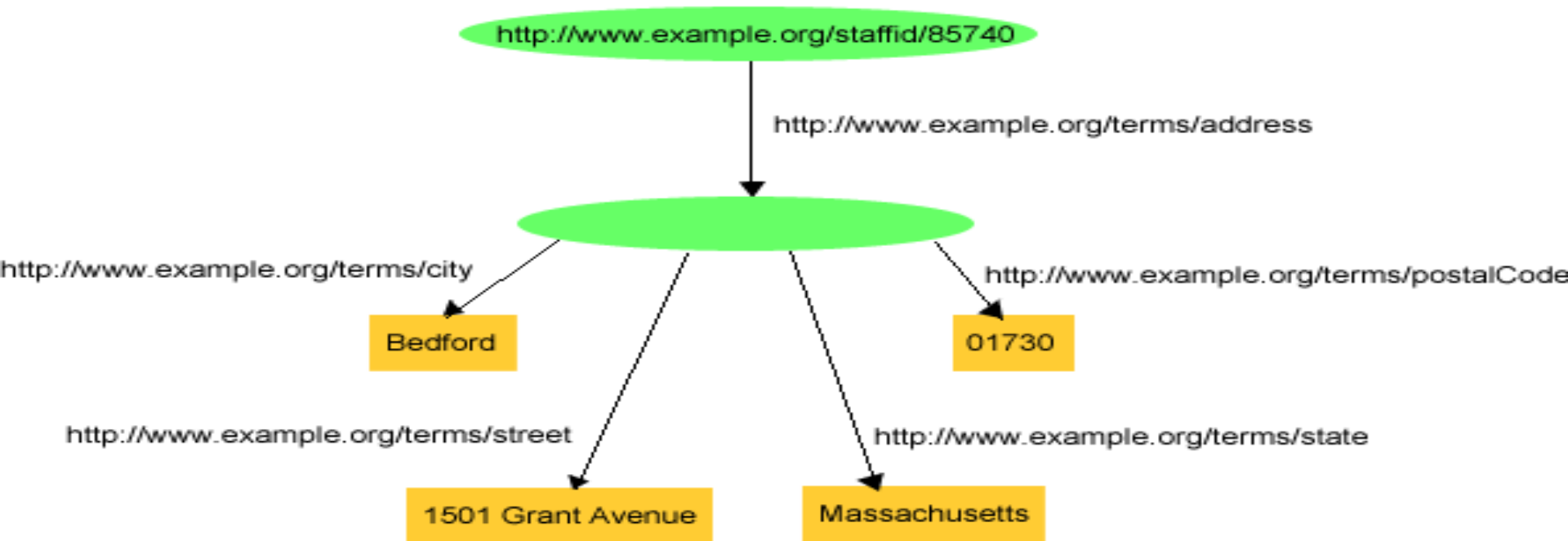
# Semantics of Blank Nodes

- In terms of **first-order logic**, a blank node corresponds to an existentially quantified variable (or a Skolem constant). Thus, a graph with blank nodes is similar to an **existentially quantified first order logic statement**.
- Blank nodes are useful to represent **n-ary relationships** in RDF (e.g., the relationship between John Smith and the street, city, state, and postal code components of his address).
- Blank nodes are also useful to more accurately make statements about **resources that may not have URIs**, but that are described in terms of relationships with other resources that do have URIs.

# Blank node(cont.)

- Blank nodes are also useful to more accurately make statements about **resources** are described in terms of relationships with other resources that do have URIs
- It just provides the necessary connectivity between various parts of the graph.
- Blank node in RDF can essentially be interpreted as a logical statement, “there exists.” That is, in these statements we assert “there exists student, who lived in cairo, who attend class SE465”

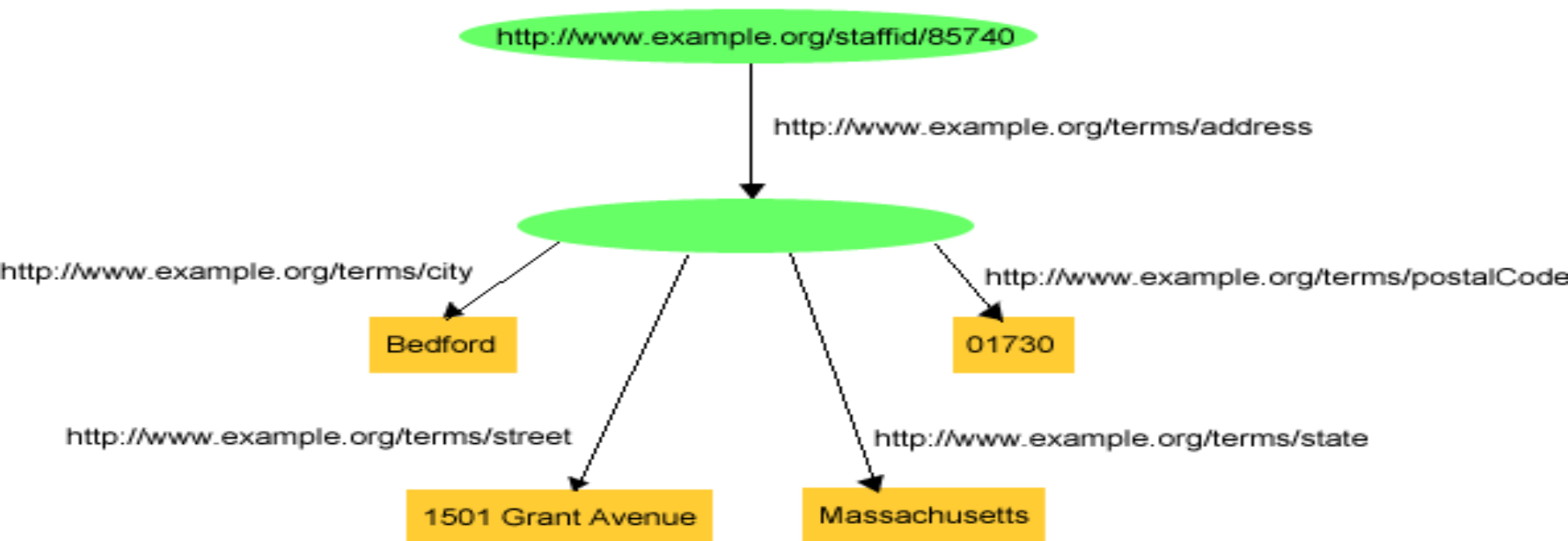
# Blank Node



**If we give `_:johnaddress` as the identifier for the blank node**

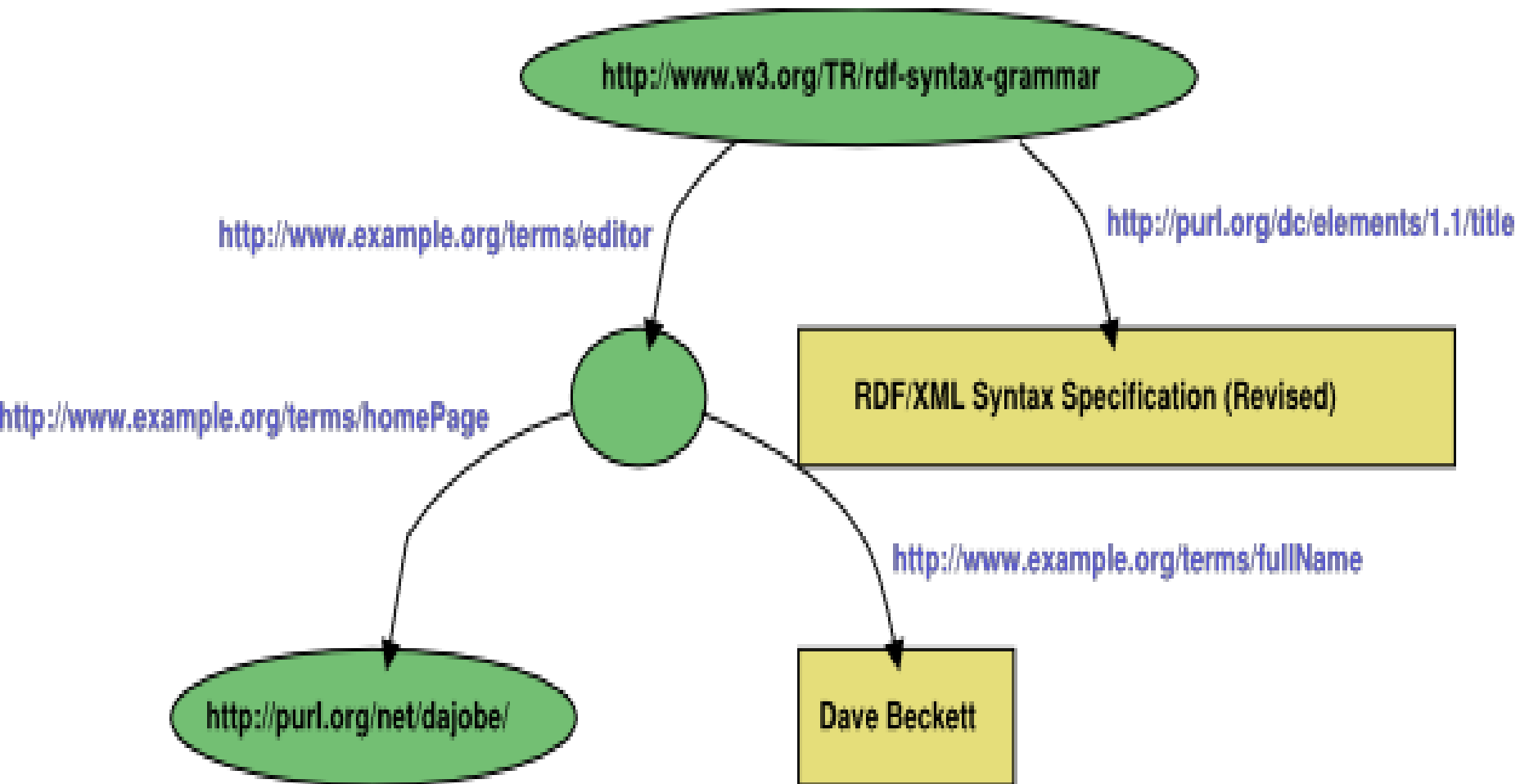
- Nodes like John's address may not require "universal" identifiers.
- Nodes with only local meaning can be blank.



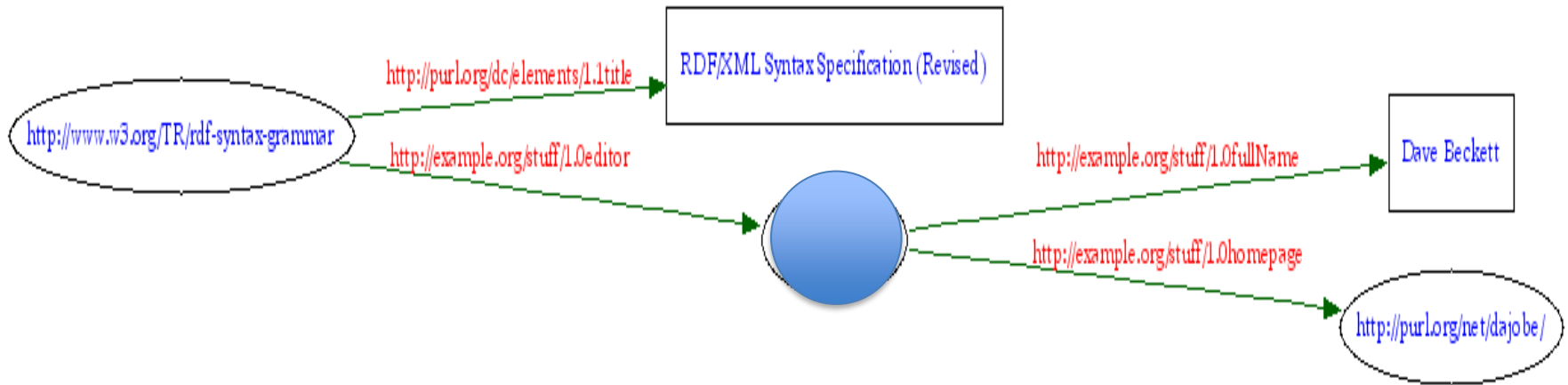


```
exstaff:85740 extterms:address _:johnaddress .  
_:johnaddress extterms:street "1501 Grant Avenue" .  
_:johnaddress extterms:city "Bedford" .  
_:johnaddress extterms:state "Massachusetts" .  
_:johnaddress extterms:postalCode "01730" .
```

# Blank node



# Blank node



No	Subject	Predicate	Object
1	<code>http://www.w3.org/TR/rdf-syntax-grammar</code>	<code>http://purl.org/dc/elements/1.1/title</code>	"RDF/XML Syntax Specification (Revised)"
2		<code>http://example.org/stuff/1.0fullName</code>	"Dave Beckett"
3	<code>http://www.w3.org/TR/rdf-syntax-grammar</code>	<code>http://example.org/stuff/1.0editor</code>	
4		<code>http://example.org/stuff/1.0homepage</code>	<code>http://purl.org/net/dajobe/</code>

# RDF Validator

- <http://www.w3.org/RDF/Validator/>
- Provides a **validator** for your RDF data.
- It creates a graph of your RDF statements
  - Choose the appropriate “Display Result Option”.
- Very useful to start writing RDF documents