

COMS 4995

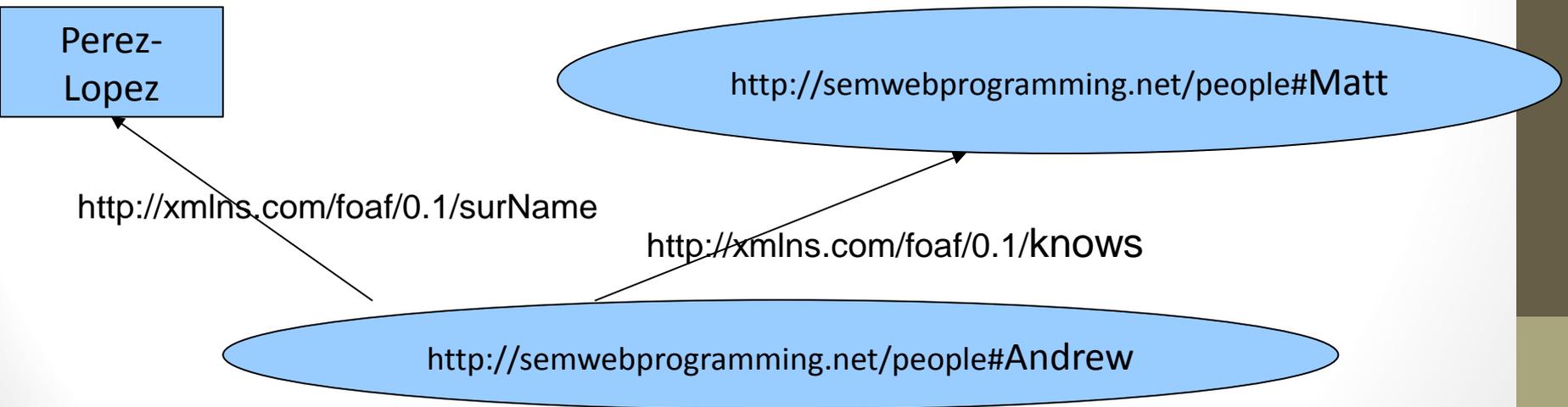
Introduction to Semantic Web

Lecture 4: RDF

Dr. Knarig Arabshian

RDF Statements

```
<rdf:Description rdf:about="http://semwebprogramming.net/people#Andrew">  
  <foaf:surname>Perez-Lopez</foaf:surname>  
  <foaf:knows rdf:resource="http://semwebprogramming.net/people#Matt"/>  
</rdf:Description>
```



XML Syntax for RDF

```
<rdf:RDF
```

```
  xmlns:people="http://semwebprogramming.net/people#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ext="http://semwebprogramming.net/2008/06/ont/foaf-extension#">
```

```
  <rdf:Description rdf:about=http://semwebprogramming.net/people#Ryan>
```

```
    <ext:worksWith>
```

```
      <rdf:Description rdf:about=http://semwebprogramming.net/people#John>
```

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

```
    </ext:worksWith>
```

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

```
  <rdf:Description rdf:about="http://semwebprogramming.net/people#Matt">
```

```
    <foaf:knows
```

```
      rdf:resource="http://semwebprogramming.net/people#John"/>
```

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

```
  <rdf:Description rdf:about="http://semwebprogramming.net/people#Andrew">
```

```
    <foaf:surname>Perez-Lopez</foaf:surname>
```

```
    <foaf:knows
```

```
      rdf:resource="http://semwebprogramming.net/people#Matt"/>
```

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

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

*rdf:RDF element used as
the root of any RDF/XML
document*

```
<rdf:RDF  
  xmlns:people="http://semwebprogramming.net/people#"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:foaf="http://xmlns.com/foaf/0.1/"  
  xmlns:ext="http://semwebprogramming.net/2008/06/ont/foaf-extension#">
```

Namespace declarations

*Subject and object are described
by elements of type rdf:Description*

```
<rdf:Description rdf:about="http://semwebprogramming.net/people#Ryan">
```

rdf:about defines identifier of resource

```
<ext:worksWith>
```

Predicate is represented directly as an element

```
<rdf:Description rdf:about="http://semwebprogramming.net/people#John">
```

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

```
</ext:worksWith>
```

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

```
<rdf:Description rdf:about="http://semwebprogramming.net/people#Matt">
```

```
<foaf:knows
```

```
  rdf:resource="http://semwebprogramming.net/people#John"/>
```

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

*Shortcut to describe an
object by using
rdf:resource as an
attribute of the predicate*

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

RDF Vocabulary

- RDF vocabulary defined in:
”<http://www.w3.org/1999/02/22-rdf-syntax-ns#>”
- STATEMENT
 - rdf:Description
- SUBJECT
 - rdf:about
 - rdf:ID:
- OBJECT
 - rdf:resource

RDF Resources

- Resources treated *differently* depending on whether they are **subject** or **object** of a statement
- Subjects of statements are designated with **rdf:about** or **rdf:ID** attribute of a **rdf:Description** tag
- Objects of statements appear in **rdf:resource** attributes of **predicate** tags
- RDF/XML is XML so uses standard XML namespace conventions to abbreviate full URIs that appear as XML elements

RDF Literals

- Literals appear as text content of predicate element
- Can be assigned a datatype using standard XML Schema Datatypes (XSD)
- Any URI can be used as a datatype for a literal so you can create your own custom types
- `xml:lang` attribute indicates language of text
- `rdf:datatype` indicates type of literal values

RDF Literals

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:zoo="http://example.org/zoo/">
  <rdf:Description rdf:about="http://example.org/zoo/Animal-123456">
    <zoo:numberOfLegs rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">4</zoo:numberOfLegs>
    <zoo:scientificName>Canis lupus familiaris</zoo:scientificName>
    <zoo:commonName xml:lang="en">Dog</zoo:commonName>
    <zoo:commonName xml:lang="es">Perro</zoo:commonName>
  </rdf:Description>
</rdf:RDF>
```

Shorthand

```
<foaf:Person rdf:ID="http:.../#Ryan" />
```

Same as:

```
<rdf:Description rdf:about="http:.../#Ryan" />
```

```
<rdf:type rdf:resource="http:../Person" />
```

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

Namespaces

- RDF/XML is XML so uses standard XML namespace conventions to abbreviate full URIs that appear as XML elements
- The use of the colon ':' in XML attributes is not allowed unless it is used with a namespace.
- Problem-- namespaces cannot be used to describe *values* of XML attributes

Namespaces

```
<rdf:RDF
  xmlns:people="http://semwebprogramming.net/people#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ext="http://semwebprogramming.net/2008/06/ont/foaf-extension#">
  <rdf:Description rdf:about="people:#Ryan">
    <ext:worksWith
      rdf:resource="people:#John"/>
  </rdf:Description>
</rdf:RDF>
```

`people:#Ryan` and `people:#John` is **WRONG** since it will be interpreted as the schema of a URI

Namespaces

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF[!ENTITY people 'http://semwebprogramming.net/people#']>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ext="http://semwebprogramming.net/2008/06/ont/foaf-extension#">
  <rdf:Description rdf:about="&people;Ryan">
    <ext:worksWith
      rdf:resource="&people;John"/>
  </rdf:Description>
</rdf:RDF>
```

SOLUTION: use XML entities. An entity in XML is a shortcut that can be declared at the beginning of a document and referred to later in the document instead of giving its complete value.

Done via document type declaration which provides a Document Type Definition (DTD)

Namespaces

- Base namespace
 - Another common case in which URIs might be abbreviated
 - XML has concept of a base URI that can be set for elements in a document using the attribute `xml:base`
 - Other attributes in the XML document then may use the relative references instead of full URIs

Namespaces

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org"
```

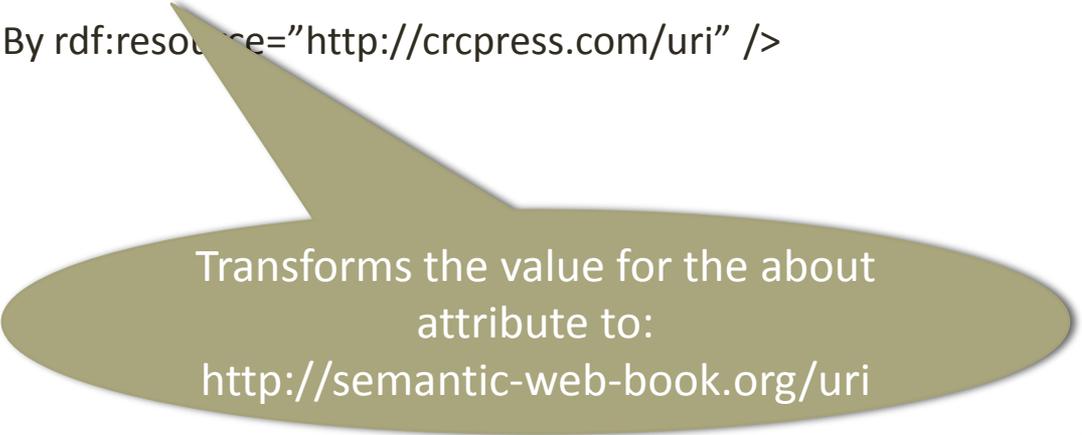
```
  xml:base = "http://semantic-web-book.org/">
```

```
  <rdf:Description rdf:about="uri">
```

```
    <ex:publishedBy rdf:resource="http://crcpress.com/uri" />
```

```
  </rdf:description>
```

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



Transforms the value for the about
attribute to:
<http://semantic-web-book.org/uri>

Datatypes in RDF

- Datatypes in RDF use XML datatypes

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.w3.org/TR/rdf-primer>

<http://example.org/title> "RDF Primer"^^xsd:string ;

<http://example.org/publicationDate> "2004-02-10"^^xsd:date .

Many-Valued Relationships with Blank Nodes

- Blank nodes: conceptually have no names at all
- Why blank nodes?
 - RDF supports only binary predicates: statement expresses a relationship between two resources
- What if we need more complicated data structure?
- For example:

@prefix ex: <http://example.org/> .

ex:Chutney ex:hasIngredient "1lb green mango",
"1tsp Cayenne pepper" .

- Ingredients and their amounts are modeled as plain strings of text.
- Not possible to query for all recipes that contain green mango unless the whole text including the amount is queried for

Blank Nodes

Another Example...

Want to express idea of a person living in a city and state

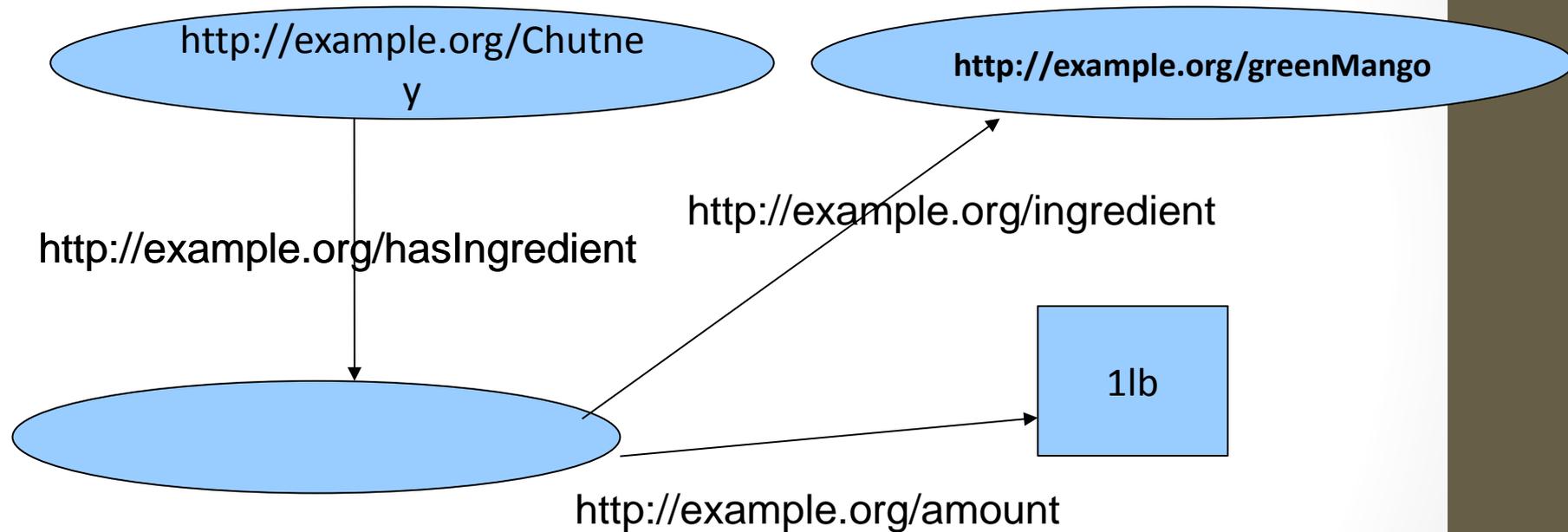
- No single statement that can connect a resource to both city and state
- You can use two predicates like `residesInCity` and `residesInState`
 - If person has two or more residences can't express this
 - Impossible to determine which city belongs with which state

Blank Nodes

```
<rdf:Description rdf:about="http://example.org/Chutney">  
  <ex:hasIngredient rdf:nodeID="id1" />  
</rdf:Description>
```

```
<rdf:Description rdf:nodeID="id1">  
  <ex:ingredient rdf:resource="http://example.org/greenMango" />  
  <ex:amount>1lb</ex:amount>  
</rdf:Description>
```

Blank Nodes



Blank Nodes

```
<rdf:Description rdf:about="http://example.org/Chutney">  
  <ex:hasIngredient rdf:parseType="Resource">  
    <ex:ingredient rdf:resource="http://example.org/greenMango" />  
    <ex:amount>1lb</ex:amount>  
  </ex:hasIngredient>  
</rdf:Description>
```

Blank Nodes

- The value *Resource* of the attribute *rdf:parseType* leads to automatic creation of a new blank node which does not have a node id within the given document
- *rdf:parseType*
 - Modifies the way parts of the XML document are interpreted
 - Leads to generation of additional triples that have not been specified directly

Blank Node in Turtle Syntax

@prefix ex: <http://example.org/> .

ex:Chutney ex:hasIngredient _:id1 .
_:id1 ex:ingredient ex:greenMango;
ex:amount "1lb" .

Blank Node in Turtle Syntax

@prefix ex: <http://www.example.org/> .

ex:Chutney ex:hasIngredient

[ex:ingredient ex:greenMango;
ex:amount "1lb"] .

Blank Nodes

- Blank nodes can not be addressed globally by means of URIs
- They do not carry any additional information within RDF graphs
- Due to how RDF is serialized, blank nodes can be referred to only in the context of a given document

Container Elements

- Collect a number of resources or attributes about which we want to make statements as a whole
- E.g., we may wish to talk about the courses given by a particular lecturer
- The content of container elements are named **rdf:_1**, **rdf:_2**, etc.
 - Alternatively **rdf:li**

Three Types of Container Elements

- **rdf:Bag** an unordered container, allowing multiple occurrences
 - E.g. members of the faculty board, documents in a folder
- **rdf:Seq** an ordered container, which may contain multiple occurrences
 - E.g. modules of a course, items on an agenda, an alphabetized list of staff members (order is imposed)
- **rdf:Alt** a set of alternatives
 - E.g. the document home and mirrors, translations of a document in various languages

Example for a Bag

```
<uni:lecturer rdf:ID="949352" uni:name="Grigoris  
  Antoniou"  
  uni:title="Professor">  
  <uni:coursesTaught>  
    <rdf:Bag>  
      <rdf:_1 rdf:resource="#CIT1112"/>  
      <rdf:_2 rdf:resource="#CIT3116"/>  
    </rdf:Bag>  
  </uni:coursesTaught>  
</uni:lecturer>
```

Example for Alternative

```
<uni:course rdf:ID="CIT1111"  
  uni:courseName="Discrete Mathematics">  
  <uni:lecturer>  
    <rdf:Alt>  
      <rdf:li rdf:resource="#949352"/>  
      <rdf:li rdf:resource="#949318"/>  
    </rdf:Alt>  
  </uni:lecturer>  
</uni:course>
```

RDF Example

Write the following statements in Turtle Syntax with meaningful properties and then draw the corresponding RDF Graph

- Peter and Frank are men.
- Frank is born in 1949.
- Sophie, Sandra, and Susie are women.
- Sandra is born in 1980.
- Susie is born in 1956.
- Peter's father is Frank.
- Peter's mother is Susie.
- Frank and Susie are married.
- Sophie likes Peter.
- Peter likes Sandra.

RDF Example

Draw the RDF graph of the following paragraph

Jackie will host dinner at her apartment at 7PM. The dinner menu includes roasted turkey, mashed potatoes, gravy and salad. Jackie's friends, Jill, James and Jenn are coming over and they will each bring the desserts: cookies, cake and ice cream, respectively.

RDF Example

- Pick five menu items and draw corresponding RDF graphs for each item

RDFa (RDF with Attributes)

Linked Data

- Linked Data is a relatively new way of thinking about how we publish data
- The data we have today is disconnected and lives in data silos.
- These data silos are not linked to one another, which goes against the core design of the Web.
- If we fix the data linking problem, computers will be able to use the data more easily.

Linked Data NOW

- Sites have started to identify a page's title, content type, and preview image to provide appropriate information in a user's newsfeed when she clicks the "Like" button.
- Search engines have started to provide richer search results by extracting fine-grained structured details from the Web pages they crawl.
- In turn, web publishers are producing increasing amounts of structured data within their Web content to improve their standing with search engines.

RDFa

- A key enabling technology behind these developments is the ability to add structured data to HTML pages directly.
- RDFa (Resource Description Framework in Attributes) is a technique that allows just that: it provides a set of markup attributes to augment the visual information on the Web with machine-readable hints.
- RDFa is a “light” version of RDF

RDFa

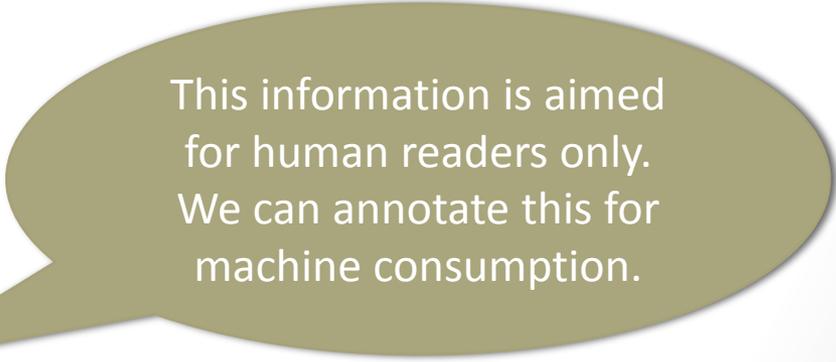
- By using RDFa your web pages will be displayed in an enhanced format on all major search engines.
- You can also use Facebook's Open Graph Protocol, which is based on RDFa, to express concepts that are contained in your web pages, like people, places, events, movies and recipes.
- The search and social companies are serious about indexing RDFa content

RDFa

Alice publishes a blog and would like to provide extra structural information on her pages like the publication date or the title. She would like to use the terms defined in the Dublin Core vocabulary [DC11], a set of terms that are widely used by, for example, the publishing industry or libraries. Her blog contains this information.

EXAMPLE 1

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2>The Trouble with Bob</h2>
  <p>Date: 2011-09-10</p>
  ...
</body>
```

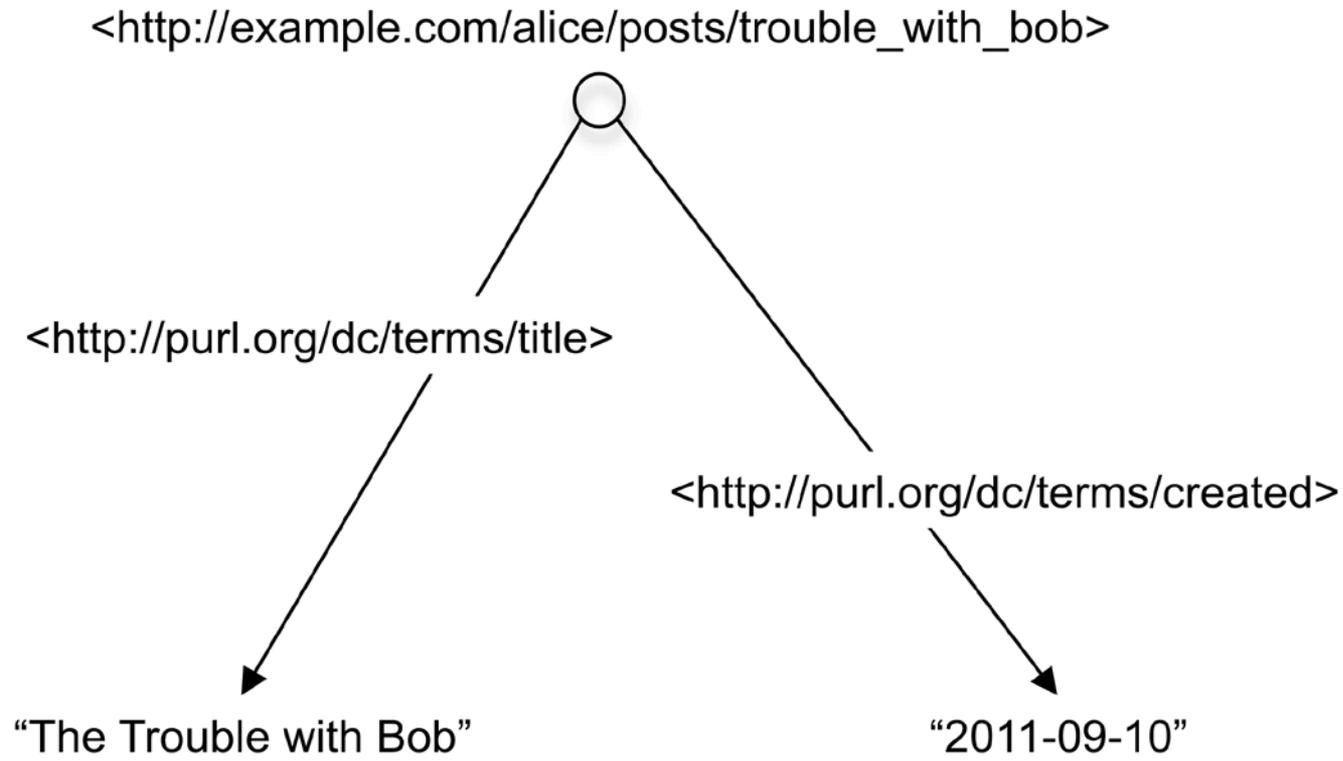


This information is aimed for human readers only. We can annotate this for machine consumption.

RDFa

```
<html>
<head>
...
</head>
<body>
...
  <h2 property="http://purl.org/dc/terms/title">The Trouble
with Bob</h2>
  <p>Date: <span
property="http://purl.org/dc/terms/created">2011-09-
10</span></p>
...
</body>
```

RDFa



RDFa

In her blog's footer, Alice already declares her content to be freely reusable, as long as she receives due credit when her articles are cited. The HTML includes a link to a Creative Commons [CC-ABOUT] license:

```
<p>All content on this site is licensed under  
<a href="http://creativecommons.org/licenses/by/3.0/">  
  a Creative Commons License</a>. ©2011 Alice Birpemschwick.</p>
```

- A human clearly understands this sentence, in particular the meaning of the link with respect to the current document
- Unfortunately, when Bob visits Alice's blog, his browser sees only a plain link that could just as well point to one of Alice's friends or to her CV.
- For Bob's browser to understand that this link actually points to the document's licensing terms, Alice needs to add some flavor, some indication of what kind of link this is.

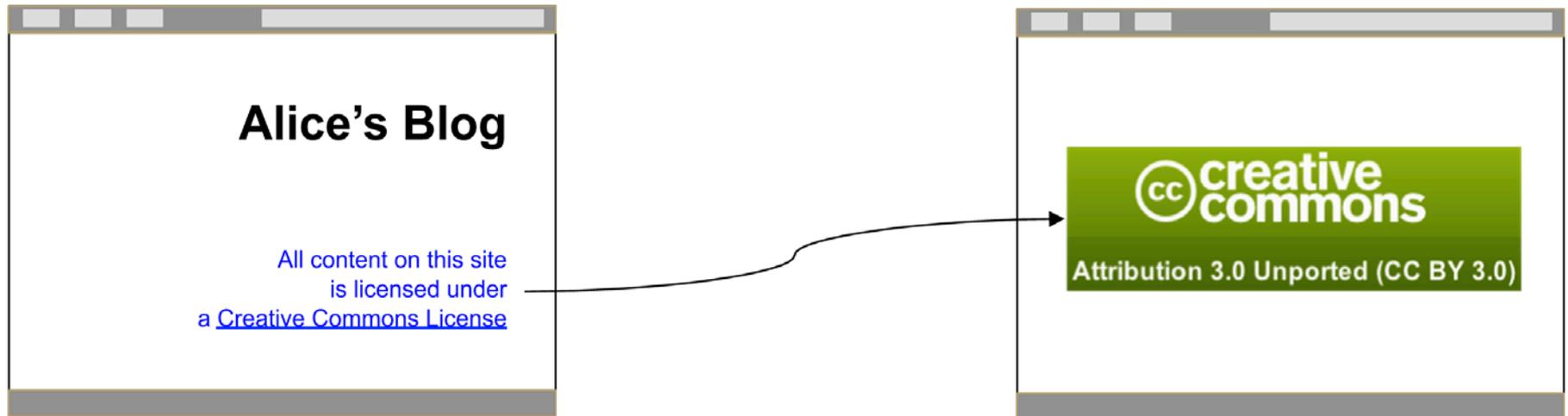
RDFa

- She can add this flavor using again the property attribute.
- The value of the attribute is the `http://creativecommons.org/ns#license`, defined by the Creative Commons:

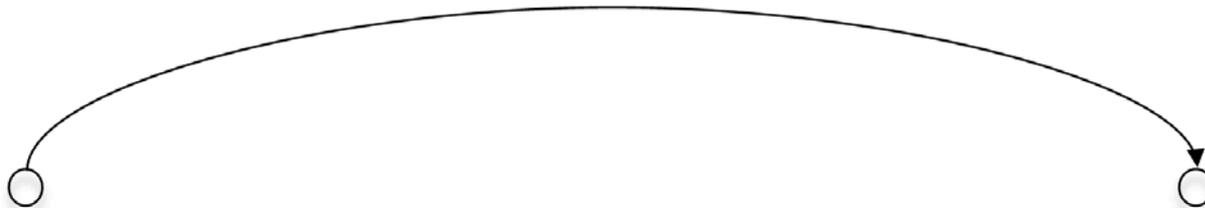
```
<p>All content on this site is licensed under  
  <a property="http://creativecommons.org/ns#license"  
  href="http://creativecommons.org/licenses/by/3.0/">  
  a Creative Commons License</a>. ©2011 Alice  
  Birpemschwick.</p>
```

- With this small update, Bob's browser will now understand that this link has a flavor: it indicates the blog's license

RDFa



`<http://creativecommons.org/ns#license>`



`<http://example.com/alice/posts/42>`

`<http://creativecommons.org/licenses/by/3.0/>`



RDF Reification

- How to express propositions about propositions?
 - Nested RDF statements
- “the detective supposes that the butler killed the gardener”
- One way to represent this:
 - `ex:detective ex:supposes “The butler killed the gardener.”` .
 - Literal will not be referenced in other triples
 - Another attempt:
 - `ex:detective ex:supposes ex:theButlerKilledTheGardener` .
 - Doesn't capture the full meaning of the proposition

RDF Reification

- Reification: process by which an abstract idea is turned into an explicit model
- Solution: nested triples
 - `ex:butler ex:killed ex:gardener .`
 - Object of the triple is a triple of its own
 - Draws the idea from many-valued relations (blank nodes)
- Auxiliary node is used as a “handle” to refer to the whole statement
 - Access to inner structure of represented triple is connected via the blank node with `rdf:subject`, `rdf:predicate` and `rdf:object` properties
 - Corresponding triple is called reified

RDF Reification

ex:detective

ex:theory

ex:theory

ex:theory

ex:supposes

rdf:subject

rdf:predicate

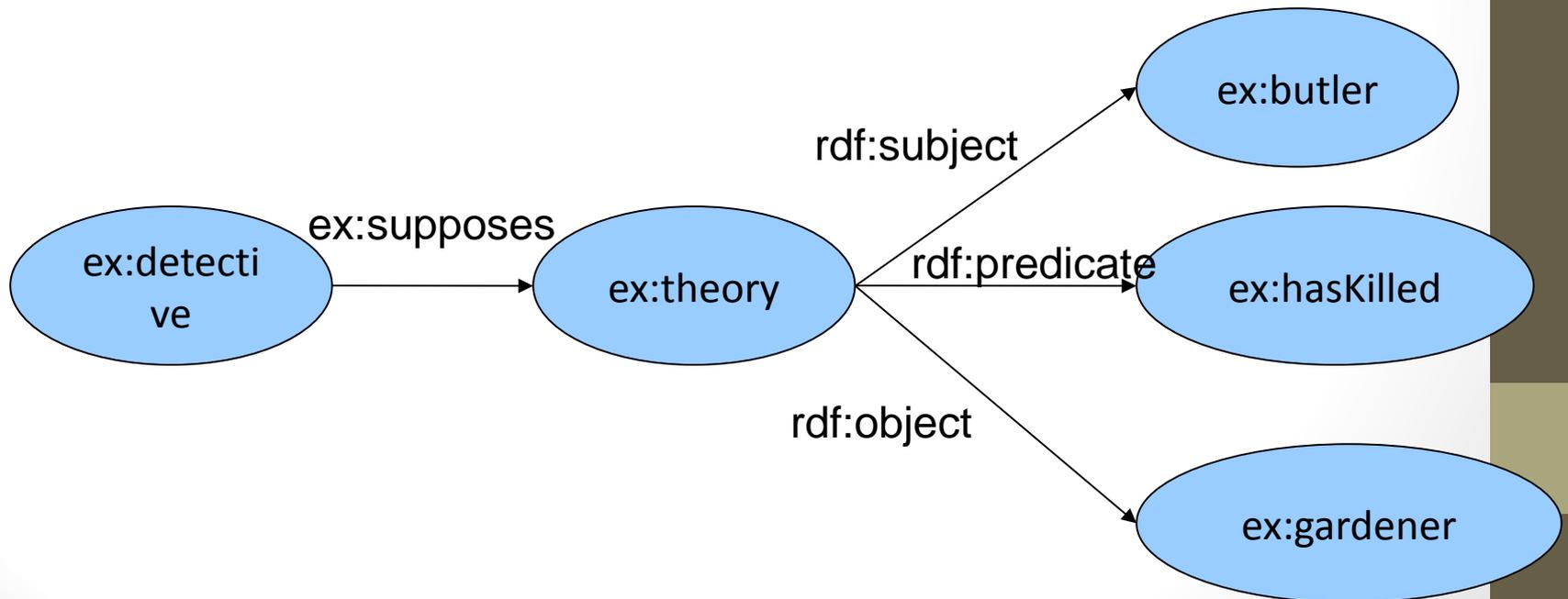
rdf:object

ex:theory .

ex:butler .

ex:hasKilled .

ex:gardener .



How can a model help us?

- Models help people communicate
 - Describe situation in a particular way that other people can understand
- Models describe and predict
 - Relates primitive concepts to one another and to more complex ones
- Models common ground for many viewpoints
 - Two people may not agree on how they perceive a domain but model can be a medium between two viewpoints

Modeling for communication

- Human to human communication tends to use informal modeling
- Informal modeling on the web
 - E-commerce systems
 - Online catalogs
 - Most online catalogs use similar modeling techniques
 - Tagging systems
 - community tagging
 - If any two people use the same tag, this becomes a common organizing entity
 - Tagging infrastructure shows which tags are used most
 - Helps browsers determine what tags to use in a search

Modeling for knowledge and prediction

- Models can be re-used to explain or predict information
- Example:
 - Knowledge
 - Understanding how fire is put out can help you understand why using a blanket helps put out a fire
 - Prediction
 - Know: ice is slippery
 - Predict:
 - Holding an ice cube might cause it to slip through your fingers
 - Driving on ice might cause your car to skid
 - Effective skating on ice would require blades

Modeling for knowledge and prediction

- This requires formal modeling
- Formal models are basic component of mathematics
- Basic arithmetic:
 - Agree on the numbering system (what does the symbol '1', '2', and '3' mean?)
 - Agree on symbols (what does '+' or '=' mean?)
 - Indisputable fact: $1 + 2 = 3$

Models Help Establish Common Grounds

- Variability is something to be expected
 - Example: is pluto a planet? Astrologers think it is; astronomers now consider it a dwarf planet.
- How to handle this?
 - Adopt the preferred point of view
 - Represent both points of view and let the information consumer decide how to model the information given the two differing models
 - This is current state of the Web

Modeling in Semantic Web

- Create an infrastructure that allows:
 - Any topic to be discussed
 - Community to work through misunderstandings
 - A framework to represent and organize common or different viewpoints (taxonomy)
- Layers of expressivity
 - Expressing the concept of “water”
 - Molecular structure (H₂O, H--O--H)
 - Different states (liquid, solid, vapor)

Inferencing in Semantic Web

- Relationships need to be added to the data so that it will seem more connected and consistently integrated
- Want to express the relationship between “apple” and “fruit”
- Referring to apple, also refers to fruit
- Restaurant in a specific location also means that the location contains this restaurant

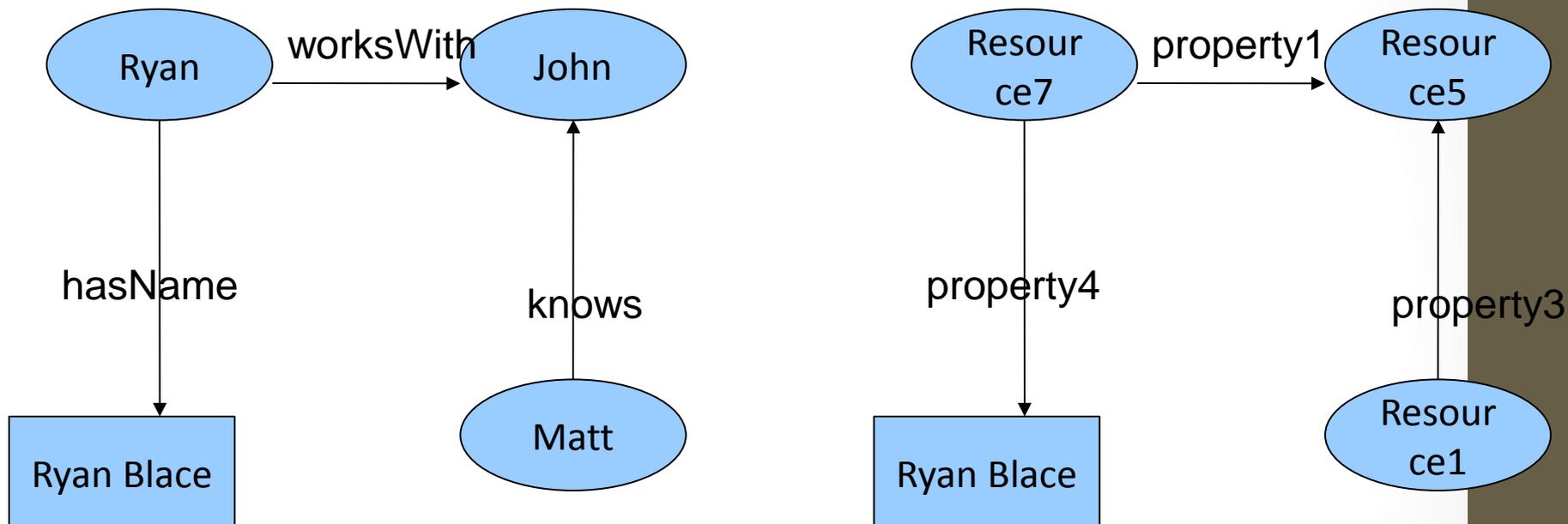
Inferencing in the Semantic Web

- RDF allows us to express propositions for single resources:
 - “Book hasName FOSWT”
 - “Book hasAuthor Hitzler”
- RDF describes *individuals* or *instances* such as “Book”, “Hitzler”, “FSWT” and puts them in relation to one-another

Inferencing in Semantic Web

- RDF lets individuals have limited types
 - Types indicate the class of entities an individual or instance belongs to
 - Literals can be in the class of natural numbers or character strings or ordered lists
- Need more information to understand the semantic meaning of a resource
 - What class of entities are these part of? Can't express this
 - “Pascal Hitzler” (person)
 - FOSWT (title)
 - Book (printed media)

Inferencing in Semantic Web



- RDF provides a way to model information
- Does not provide a way of specifying what that information means
- RDF graph alone can only be interpreted as a graph
- Meaning is apparent on your ability to recognize and interpret the URIs, literals, and general structure of graph

Inferencing in Semantic Web

- Vocabulary
 - Terms referring to individuals, relations and classes
 - User will have concrete idea about meanings
 - Restaurant is an Eatery
 - University is an Institution
 - Pascal Hitzler is a Person
 - To the computer system the vocabulary is just a series of character strings with no meaning
 - Semantic interrelations need to be explicitly communicated to the system in some format in order to enable it to draw conclusions that rely on human knowledge