

# Knowledge Representation for the Semantic Web

Winter Quarter 2010

Slides 3 – 01/19/2010

**Pascal Hitzler**

Kno.e.sis Center

Wright State University, Dayton, OH

<http://www.knoesis.org/pascal/>



# Slides are based on

**Pascal Hitzler, Markus Krötzsch,  
Sebastian Rudolph**

**Foundations of Semantic Web  
Technologies**

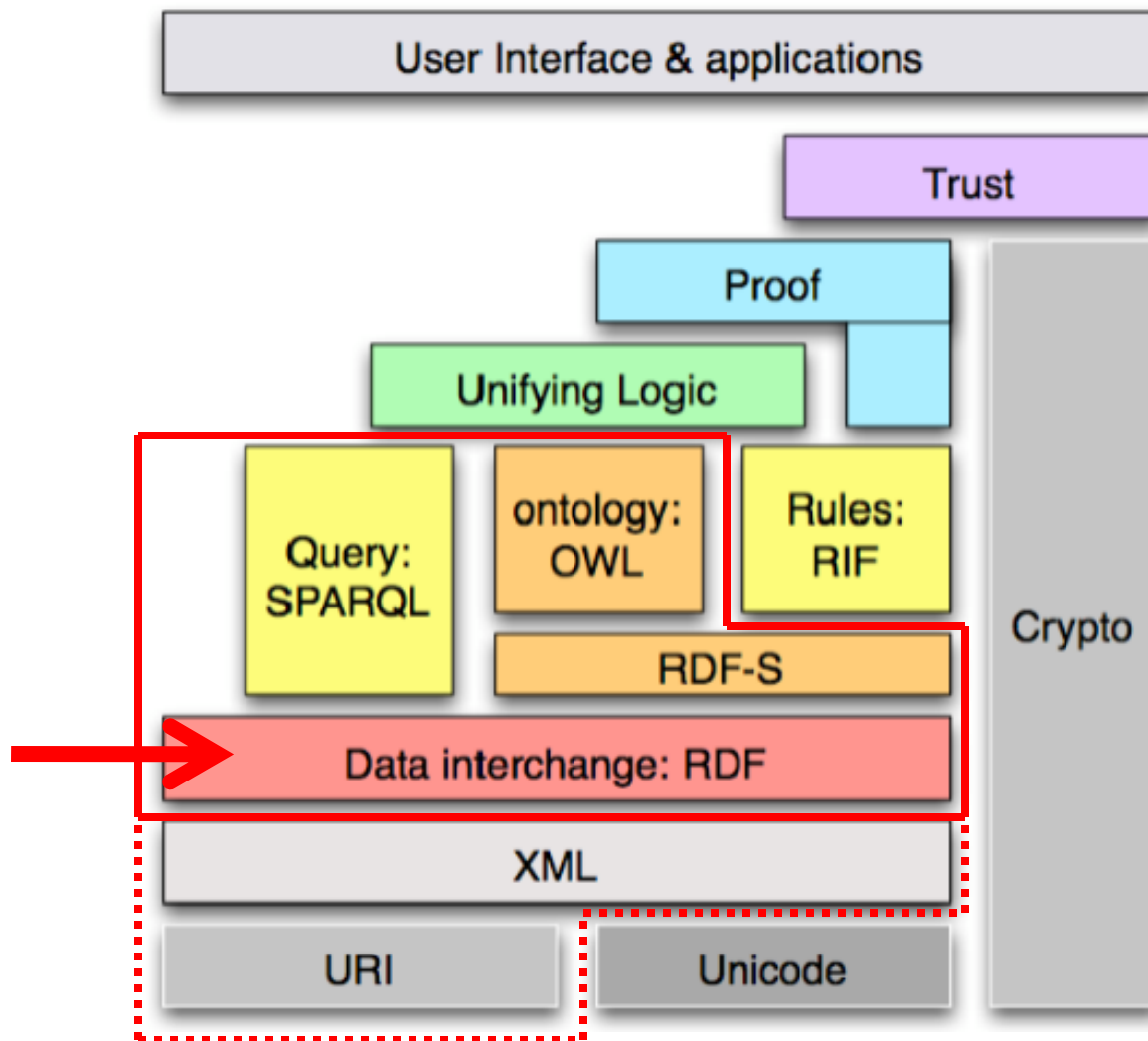
**Chapman & Hall/CRC, 2010**

**Flyer with special offer is available.**

**<http://www.semantic-web-book.org>**



# Today: RDF syntax



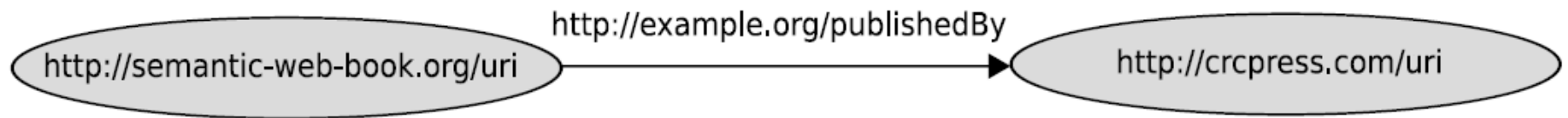
**+ conjunctive queries for OWL**

- 1. Motivation**
- 2. Triples and Graphs**
- 3. RDF syntaxes: Turtle and RDF/XML**
- 4. Datatypes**
- 5. n-ary relationships**
- 6. Empty nodes**
- 7. Lists**
- 8. Class project**
- 9. Class presentation**

- How do you encode the piece of knowledge “The book FOST is published by CRC Press”
- `<book>`  
`<title>FOST</title>`  
`<publisher>CRC Press</publisher>`  
`</book>`
- `<publisher>`  
`<name>CRC Press</name>`  
`<book><title>FOST</title><book>`  
`</publisher>`
- etc.

- Merging trees is rather cumbersome and the result isn't always clear.
  - `<publisher>`  
`<name>CRC Press</name>`  
`<book><title>FOST</title><book>`  
`</publisher>`
  - `<book>`  
`<title>Semantic Web</title>`  
`<publisher>Springer</publisher>`  
`</book>`

- Use (directed) graphs as data model



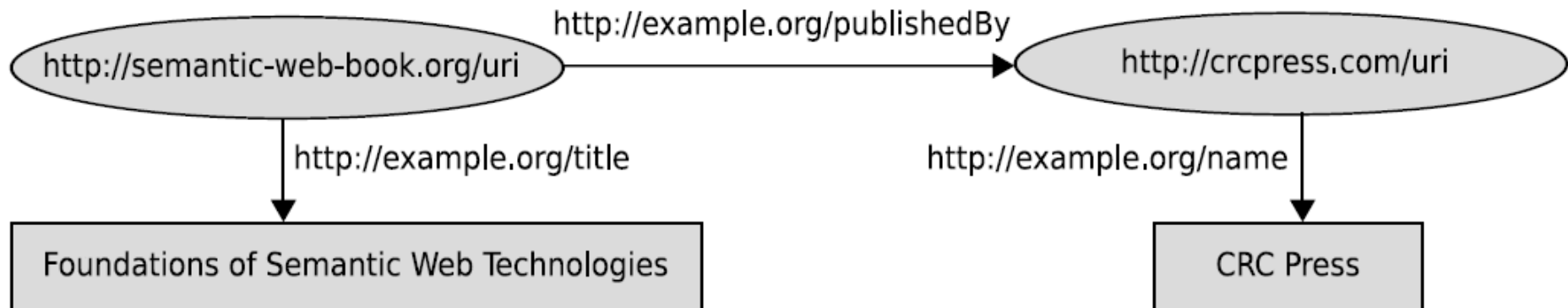
- **“Resource Description Framework”**
- **W3C Recommendation 2004**  
**<http://www.w3.org/RDF/>**
- **RDF is a data model**
  - **originally for describing metadata for web pages, but has grown beyond that**
  - **structured information**
  - **universal, machine-readable data exchange format**
  - **main syntax uses XML for serialization**



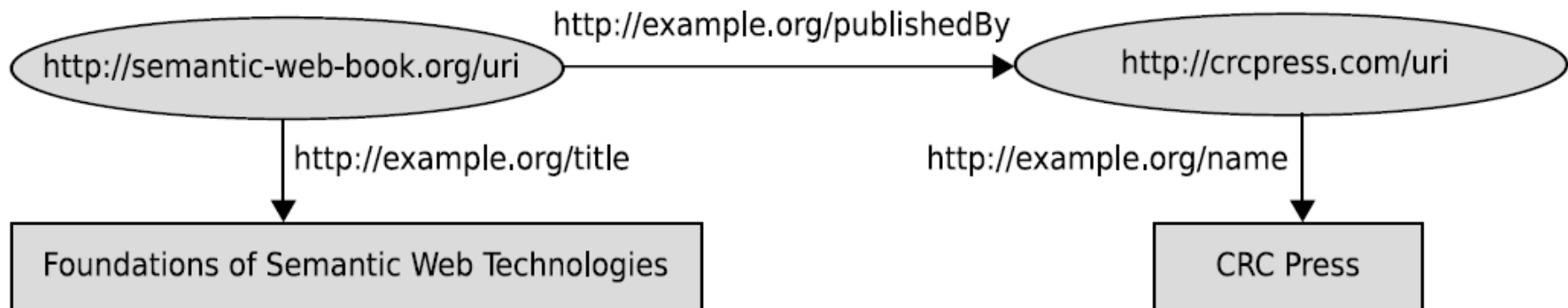
1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. n-ary relationships
6. Empty nodes
7. Lists
8. Class project
9. Class presentation

- **URIs**
  - for referencing resources
- **Literals**
  - data values
- **Empty nodes**
  - talking about something which doesn't have a name  
(or the name of which isn't known)

- for representing data values
- encoded as strings
- interpreted by means of datatypes
- literals without datatype are treated the same as strings



- there are several possibilities for representing graphs
- we use: graph as list of (node-edge-node) triples



- An RDF triple consists of



(borrowed from linguistics)

- allowed are:
  - In the subject : URIs and empty nodes
  - In the predicate: URIs (usually called *properties*)
  - In the object: URIs and empty nodes and literals
- Note that the graph can be reconstructed from the list of triples.

1. Motivation
2. Triples and Graphs
3. **RDF syntaxes: Turtle and RDF/XML**
4. Datatypes
5. n-ary relationships
6. Empty nodes
7. Lists
8. Class project
9. Class presentation

- simple syntax for RDF
- triples are directly listed as such
  - URIs are in <angle brackets>
  - Literals are "enclosed in quotes"
  - triples end with a full-stop .
  - whitespace (blanks, line feeds) is ignored

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

- shortcuts for prefixes

```
@prefix book: <http://semantic-web-book.org/> .
```

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

```
@prefix crc: <http://crcpress.com/> .
```

```
book:uri    ex:publishedBy    crc:uri .
```

```
book:uri    ex:title          "Foundations of Semantic Web Technologies" .
```

```
crc:uri     ex:name           "CRC Press" .
```



```
@prefix book: <http://semantic-web-book.org/> .  
@prefix ex: <http://example.org/> .  
@prefix crc: <http://crcpress.com/> .
```

```
book:uri    ex:publishedBy    crc:uri .  
book:uri    ex:title          "Foundations of Semantic Web Technologies" .  
crc:uri     ex:name           "CRC Press" .
```

- **grouping of triples with the same subject**
- **grouping of triples with same subject and predicate**

```
@prefix book: <http://semantic-web-book.org/> .  
@prefix ex: <http://example.org/> .  
@prefix crc: <http://crcpress.com/> .
```

```
book:uri    ex:publishedBy    crc:uri ;  
            ex:title          "Foundations of Semantic Web Technologies" .  
crc:uri     ex:name           "CRC Press", "CRC" .
```

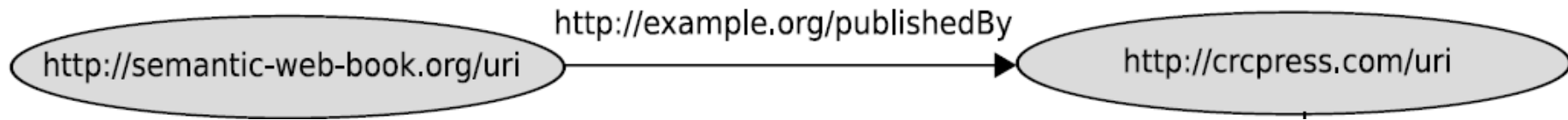
- **Turtle is easy to read and write**
- **But XML is the basis for data transfer on the web**
- **There's a lot of tool (and programming library) support for XML**
- **Hence, the main syntax for RDF is XML-based.**
- **Turtle is not a W3C recommendation**
- **The normative syntax for RDF is it's XML syntax**

- namespaces are used for disambiguating tags
- tags belonging to the RDF language come with a fixed namespace, usually abbreviated 'rdf'

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

  <rdf:Description rdf:about="http://semantic-web-book.org/uri">
    <ex:publishedBy>
      <rdf:Description rdf:about="http://crcpress.com/uri">
        </rdf:Description>
      </ex:publishedBy>
    </rdf:Description>

</rdf:RDF>
```



**subject node**

**URI of the subject**

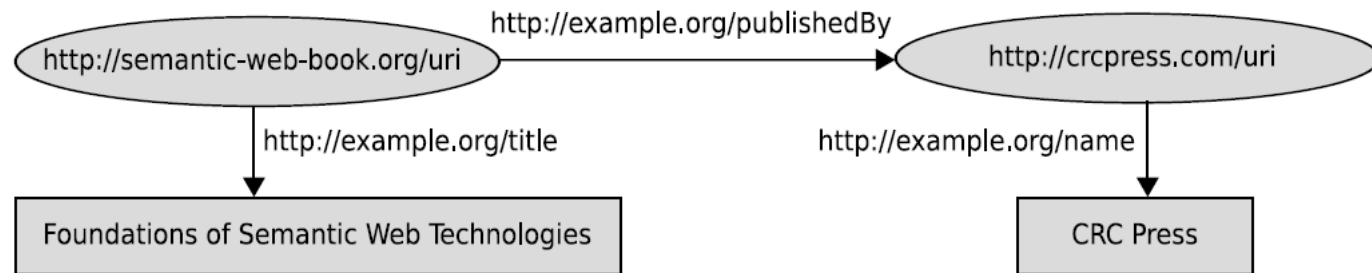
**property**

**object node**

**URI of the object**

```
<rdf:Description rdf:about="http://semantic-web-book.org/uri">
  <ex:publishedBy>
    <rdf:Description rdf:about="http://crcpress.com/uri">
    </rdf:Description>
  </ex:publishedBy>
</rdf:Description>
```

- Untyped literals can be left as free text
- 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>
```

- **Equivalent representation of literals using XML attributes**
  - the attribute-name is then the property-URI
- **Equivalent representation of objects by giving their URIs as value of a `rdf:resource` attribute within a property tag.**

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

- The use of namespaces is essential since the use of the colon ':' in XML attributes is not allowed unless it is used with a namespace.
- Problem: namespaces cannot be used in values of XML attributes: `rdf:about="book:uri"` is **wrong** since 'book' would be interpreted in the sense of a URI schema.
- Solution: use XML ENTITYs.

```
<?xml version="1.0" encoding="utf-8"?> <!DOCTYPE rdf:RDF [  
    <!ENTITY book 'http://semantic-web-book.org/'>  
]>  
  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
    xmlns:ex ="http://example.org/">  
  
    <rdf:Description rdf:about="&book;uri">  
        <ex:title>Foundations of Semantic Web Technologies</ex:title>  
    </rdf:Description>  
  
</rdf:RDF>
```

- Use of the base namespace

```
<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>
```



1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. **Datatypes**
5. n-ary relationships
6. Empty nodes
7. Lists
8. Class project
9. Class presentation

# Datatypes in RDF

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

http://example.org/title

http://example.org/publicationDate

"RDF Primer"^^http://www.w3.org/2001/XMLSchema#string

"2004-02-10"^^http://www.w3.org/2001/XMLSchema#date

```
@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 .
```

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-primer">
  <ex:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    RDF Primer
  </ex:title>
  <ex:publicationDate
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    2004-02-10
  </ex:publicationDate>
</rdf:Description>
```

- usually use of XML Schema datatype
- Note that the same data value can have different representations:
  - "3.14"^^xsd:decimal is the same as "+03.14"^^xsd:decimal but
  - "3.14"^^xsd:string is **not** the same as "+03.14"^^xsd:string
- there is only one required datatype in RDF, called rdf:XMLLiteral
  - arbitrary (balanced) XML fragments
  - special syntax:

```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:title rdf:parseType="Literal">
    Foundations of
    <br />
    <b>Semantic Web Technologies</b>
  </ex:title>
</rdf:Description>
```

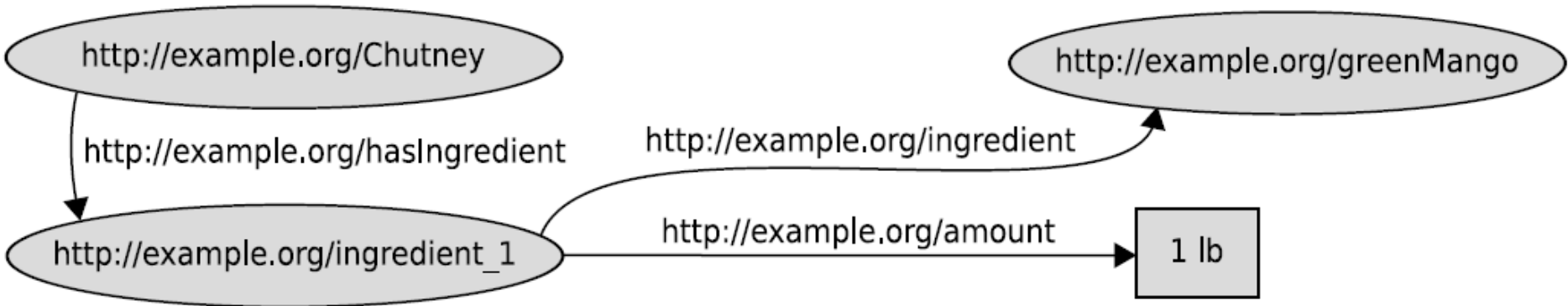
1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. **n-ary relationships**
6. Empty nodes
7. Lists
8. Class project
9. Class presentation

# What is wrong with these?

```
@prefix ex: <http://example.org/> .  
ex:Chutney ex:hasIngredient "1lb green mango",  
                                "1tsp. Cayenne pepper" .
```

```
@prefix ex: <http://example.org/> .  
ex:Chutney ex:ingredient ex:greenMango; ex:amount "1lb" ;  
            ex:ingredient ex:CayennePepper; ex:amount "1tsp." .
```

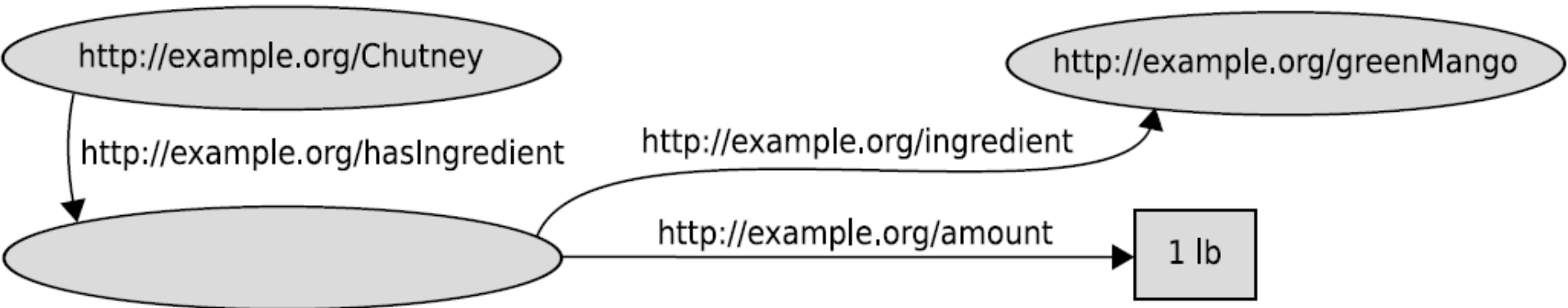
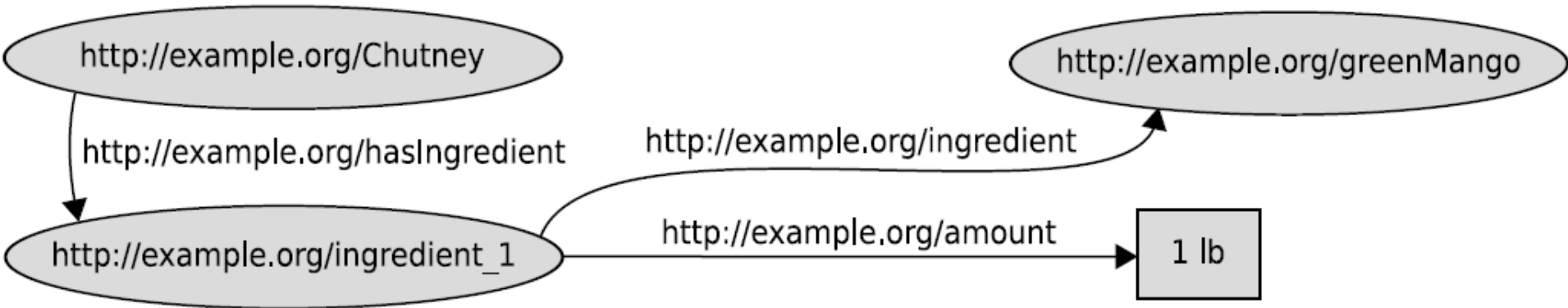
# It's a ternary relationship!



```
@prefix ex: <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Chutney      ex:hasIngredient  ex:ingredient1 .
ex:ingredient1  rdf:value        ex:greenMango;
ex:ingredient1  ex:amount         "1lb" .
```

1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. n-ary relationships
6. **Empty nodes**
7. Lists
8. Class project
9. Class presentation

# It doesn't need a name :)





```
<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>
```

## shortcut:

```
<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>
```

```
<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>
```

## Turtle:

```
@prefix ex: <http://example.org/> .
ex:Chutney    ex:hasIngredient    _:id1 .
_:id1        ex:ingredient    ex:greenMango;    ex:amount    "1lb" .
```

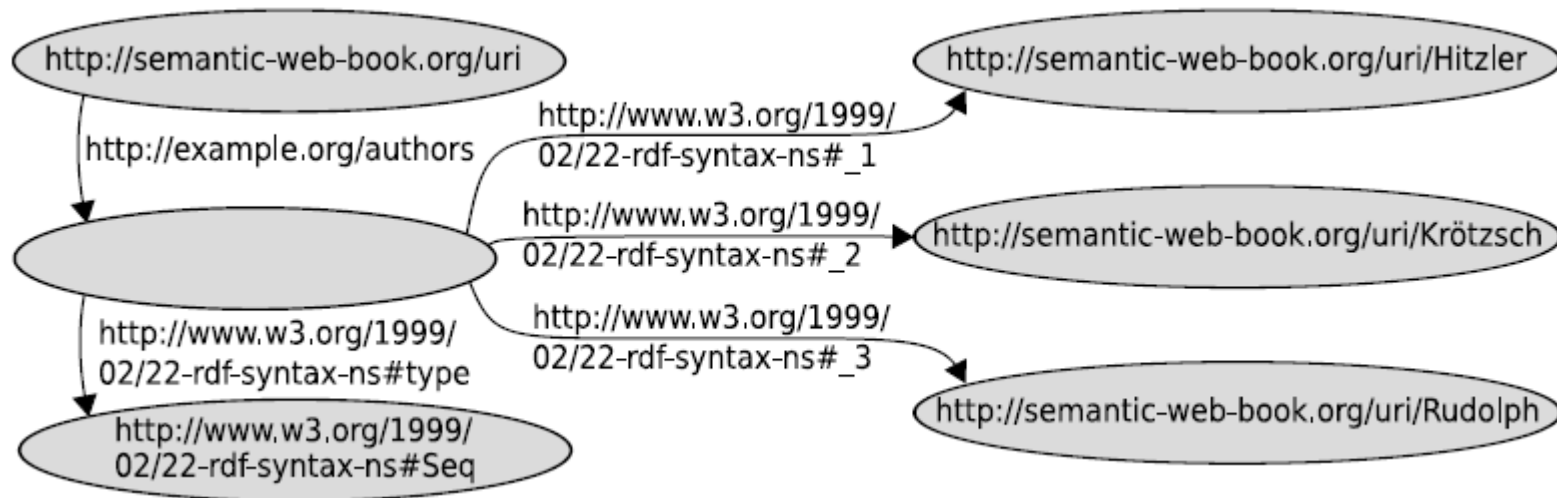
```
@prefix ex: <http://example.org/> .  
ex:Chutney    ex:hasIngredient  _:id1 .  
_:id1        ex:ingredient  ex:greenMango;    ex:amount  "1lb" .
```

## shortcut:

```
@prefix ex: <http://example.org/> .  
ex:Chutney    ex:hasIngredient  
               [ ex:ingredient  ex:greenMango;    ex:amount  "1lb" ] .
```

1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. n-ary relationships
6. Empty nodes
- 7. Lists**
8. Class project
9. Class presentation

# Open lists (containers)



```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:authors>
    <rdf:Seq>
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Hitzler" />
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Kröttsch" />
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Rudolph" />
    </rdf:Seq>
  </ex:authors>
</rdf:Description>
```

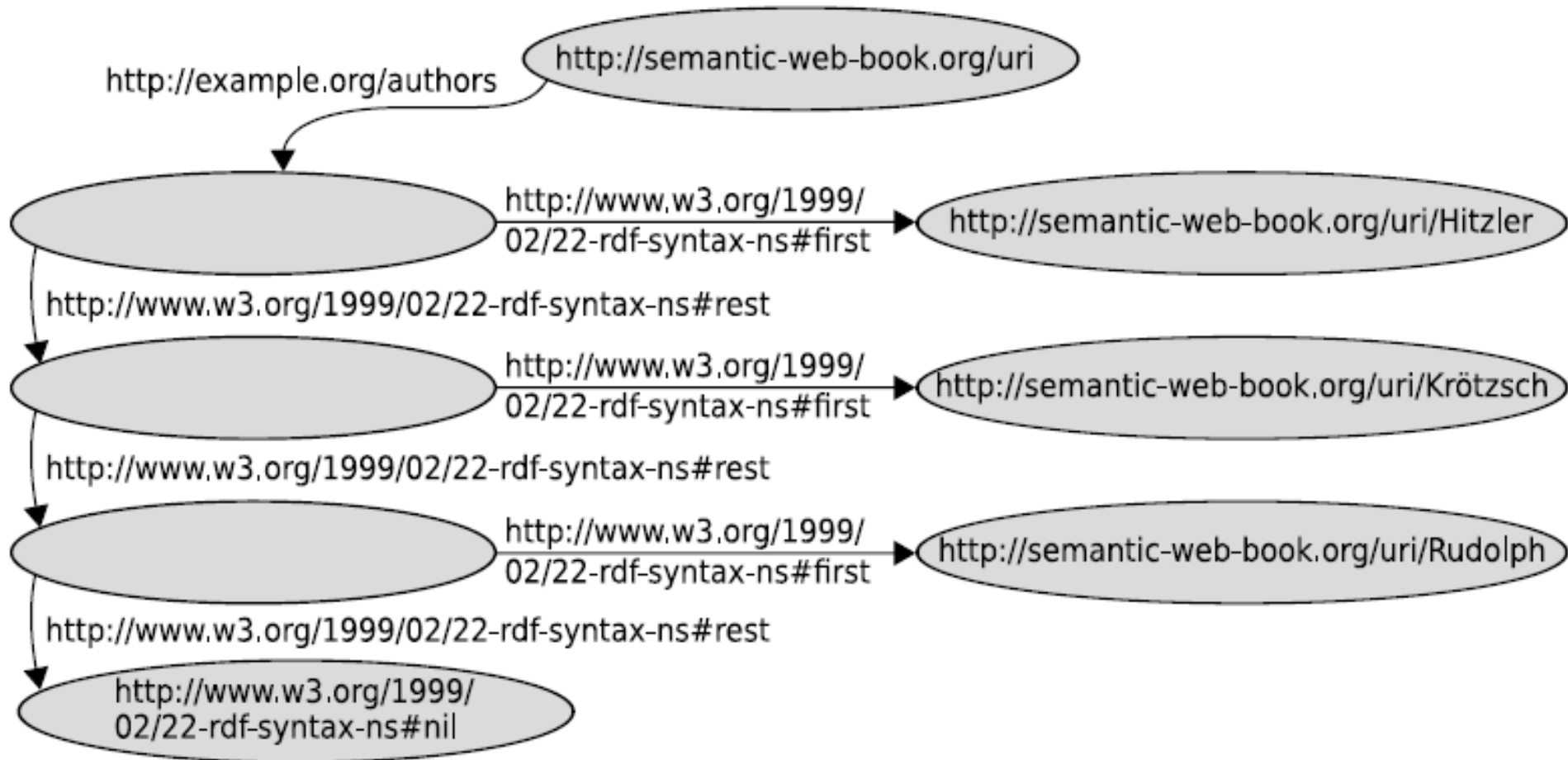
- “open”: new elements can be added.
- **rdf:Seq** – ordered list
- **rdf:Bag** – unordered set
- **rdf:Alt** – set of alternatives
- **Lists are actually hardly reflected in the formal semantics (more about this later)**

# Closed lists (collections)

```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:authors rdf:parseType="Collection">
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Hitzler" />
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Krötzsch" />
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Rudolph" />
  </ex:authors>
</rdf:Description>
```

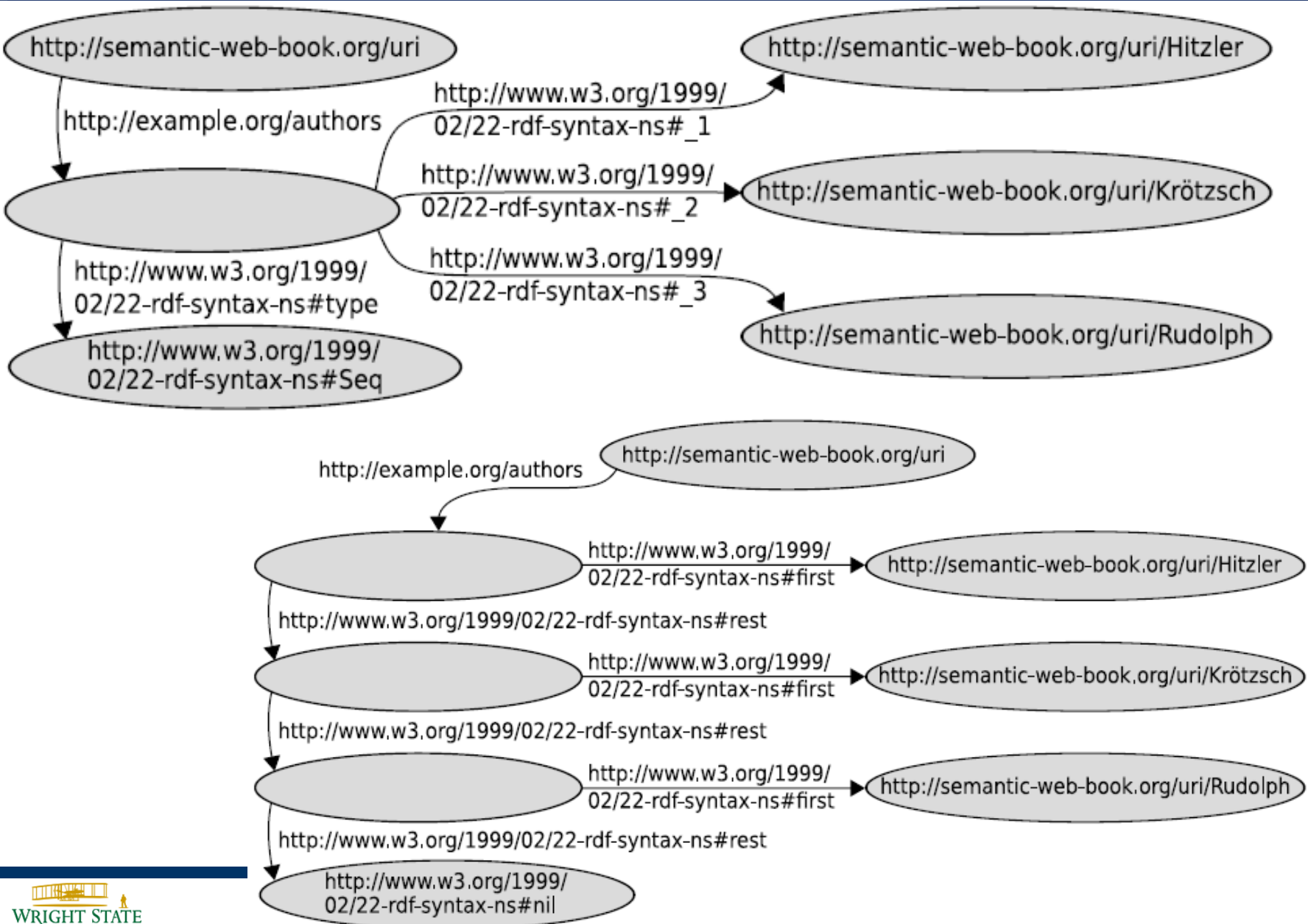
```
@prefix book: <http://semantic-web-book.org/> .
book:uri <http://example.org/authors>
  ( book:uri/Hitzler book:uri/Krötzsch book:uri/Rudolph ) .
```

# Closed lists (collections)





# Comparison

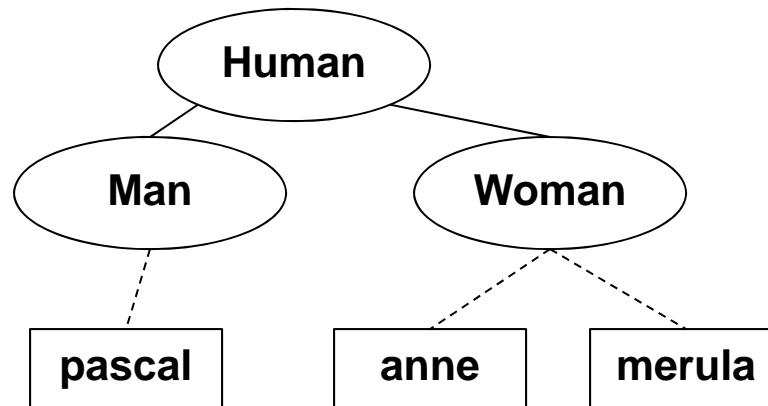


1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. n-ary relationships
6. Empty nodes
7. Lists
8. **Class project**
9. Class presentation

- Aircrafts
- Alcoholic Beverages
- American Football
- Cars and Vehicles
- Computer Science
- Cuisine
- Networks
- Parasite lifecycle
- People at University
- Trust
  
- **Be punctual!**
- **Send me readable input!**

# Class project: next step

- re-check your taxonomy for correctness!
- add ca. 10 instances to your taxonomy



- add ca. 8 subject-predicate-object triples to your taxonomy, reusing the instances you created, and inventing suitable predicates (RDF properties). Use Turtle syntax for these.  
`anne motherOf merula .`
- introduce changes to your ontology in whatever way needed
- document briefly what you have done and why (in particular if you find bugs!)
- **send to me by Thursday noon**

1. Motivation
2. Triples and Graphs
3. RDF syntaxes: Turtle and RDF/XML
4. Datatypes
5. n-ary relationships
6. Empty nodes
7. Lists
8. Class project
9. **Class presentation**

- **RDFa – embedding RDF in HTML (W3C standard)**  
**Pavan, Thursday 28<sup>th</sup> of January**
- **Scalable Distributed Reasoning using MapReduce (Urbani, Kotoulas, Oren, van Harmelen, ISWC2009)**  
**Wenbo, Thursday 28<sup>th</sup> of January**
- **Semantic MediaWiki**  
**Vinh, to be scheduled**

## Applications:

- **Linked Open Data (paper by Heath, Bizer, Berners-Lee, IJSWIS 2009)**
- **FOAF (friend of a friend social networking vocabulary)**
- **The SNOMED ontology (major biomedical ontology)**
- **Yahoo! Search Monkey (enhancing web search)**

## Standards:

- **SKOS – data model for sharing and linking knowledge organization systems via the Web (W3C standard)**

## Research papers:

- **Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples (Weaver, Hendler, ISWC2009)**

## Tools:

- **Protege – Ontology editing tool**
- **Jena – Java framework for Semantic Web by HP**
- **RDF triple stores (Virtuoso, Redland, Sesame, AllegroGraph)**



**Thursday 21<sup>st</sup> of January: RDFS Part I**  
**Tuesday 26<sup>th</sup> of January: RDF and RDFS Semantics**  
**Thursday 28<sup>st</sup> of January: 2 class presentations**  
**Tuesday 2<sup>nd</sup> of February: Exercise session**

**Estimated breakdown of sessions:**

**Intro + XML: 2**

**RDF: 3**

**OWL and Logic: 5**

**SPARQL and Querying: 2**

**Class Presentations: 3**

**Exercise sessions: 3**