Publishing Linked Data

Sam Oh, Professor, Sungkyunkwan University LIS
Affiliate Professor, University of Washington, iSchool
ISO/IEC JTC1/SC34 Chair, ISO TC46/SC9 Chair, DCMI Oversight Committee Member

Jinho Park, NLK Senior Researcher
Wonghon Jang, LG Sangnam Digital Library, Manager
Sangeun Han, Sungkyunkwan University, Graduate Student
Presentation Outline

• Intro to RDF/OWL
• Ontology Design using TopBraid Composer (TBC)
• Publishing Linked Data using Open Refine and BibFramework
• NLK LD Conversion Case
1. RDF
RDF Data Model

• Resources
  – A resource is a thing you talk about (can reference)
  – Resources have URI’s
  – RDF definitions are themselves Resources

• Properties
  – slots, define relationships to other resources or atomic values

• Statements
  – “Resource has Property with Value”
  – (Values can be resources or atomic XML data)
A Simple Example

- **Statement**
  - “Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”

- **Structure**
  - Resource (subject) http://www.w3.org/Home/Lassila
  - Property (predicate) http://www.schema.org/#Creator
  - Value (object) "Ora Lassila"

- **Directed graph**

```plaintext
http://www.w3.org/Home/Lassila -> s:Creator -> Ora Lassila
```
A Simple Example

- **Statement**
  - “Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”

- **Structure**
  - Resource (subject) http://www.w3.org/Home/Lassila
  - Property (predicate) http://www.schema.org/#Creator
  - Value (object) "Ora Lassila"

- **Directed graph**
  
  ![Directed graph diagram](http://www.w3.org/Home/Lassila)
A Simple Example

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

http://purl.org/dc/elements/1.1/creator

http://www.example.org/staffid/85740
Another Example

- To add properties to **Creator**, point through an intermediate **Resource**.

```
http://www.w3.org/Home/Lassila

s:Creator

Person://fi/654645635

Name
Ora Lassila

Email
lassila@w3.org
```
RDF for Eric Miller

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.
Specifying Classes

- To specify a class, create an RDF resource of type rdfs:Class

```xml
<rdfs:Class id="MyClass">
  <rdfs:label>My Class</rdfs:label>
  <rdfs:comment>John Cowan’s demonstration Class</rdfs:comment>
</rdfs:Class>
```
1. Intro to RDF/OWL

Vehicle Class Hierarchy

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.example.org/schemas/vehicles#Van

http://www.w3.org/2000/01/rdf-schema#subClassOf

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
Specifying Properties

- To specify a property, create an RDF resource of type rdfs:Property

```
<rdf:Property id="myProperty">
  <rdfs:comment>
    John Cowan’s demo property
  </rdfs:comment>
  <rdfs:domain resource="#MyClass"/>
  <rdfs:range resource="..#Literal"/>
</rdf:Property>
```
2. RDF/OWL
"Water Taxonomy" to explain OWL

NaturallyOccurringWaterSource

Stream
- Brook
- Rivulet

BodyOfWater
- River
- Tributary
- Lake
- Ocean
- Sea

Ocean
This notation is used to indicate that a person has only one birthplace location:

![Diagram showing the relationship between Person and Location]

This notation is used to indicate that a person has only one driver's license number. Further, a driver's license number is associated with only one person:

![Diagram showing the relationship between Person and Number]
Using OWL to Define Properties
A Symmetric property - if water source A connectsTo water source B then water source B connects to water source A.
Assume that connectsTo has been defined, in an OWL document, to be a Symmetric property:

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <connectsTo>
    <River rdf:about="http://www.china.org/rivers#Wu"/>
  </connectsTo>
</River>
```

Since connectsTo has been defined to be a Symmetric property we can infer that:

The Wu River connectsTo the Yangtze River.
A Transitive property - if A is containedIn B, and B is containedIn C then A is containedIn C.
Transitive Property

Suppose that you retrieve these two documents from two different Web sites. One describes the EastChinaSea and the other describes the ChinaSea:

```xml
<?xml version="1.0"?>
<Sea rdf:ID="EastChinaSea"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <containedIn>
    <Sea rdf:about="http://www.china.gov#ChinaSea"/>
  </containedIn>
</Sea>
```

```xml
<?xml version="1.0"?>
<Sea rdf:about="http://www.china.gov#ChinaSea"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <containedIn>
    <Ocean rdf:about="http://www.geodesy.org#PacificOcean"/>
  </containedIn>
</Sea>
```

If containedIn is defined to be a Transitive property then we can infer that: The EastChinaSea is containedIn the PacificOcean.
If containedIn is defined to be Transitive, we can infer that:

EastChinaSea containedIn ChinaSea containedIn PacificOcean

EastChinaSea containedIn PacificOcean
2. RDF/OWL

**Functional Properties**

A Functional property - for each instance there is at most one value for the property.
Functional Properties

Suppose that there are two independent documents describing the Yangtze River:

```xml
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/rivers#Yangtze"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <emptiesInto rdf:resource="http://www.china.org/geography#EastChinaSea"/>
</River>
```

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/rivers#Yangtze"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <emptiesInto rdf:resource="http://www.national-geographic.org#S1001-x-302"/>
</River>
```

If emptiesInto is defined to be functional then we can infer that:

\[
\]
If `emptiesInto` has been defined to be Functional then we can infer that these two values must refer to the same thing.
Inverse Properties

In a database or knowledge graph, inverse properties are used to represent the relationship from Resource 2 to Resource 1 when a property P1 relates Resource 1 to Resource 2. For example, if a river empties into a body of water, then the inverse property would relate the body of water to the river.
2. RDF/OWL

Inverse Properties

Consider this document:

```xml
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <emptiesInto rdf:resource="http://www.china.org/geography#EastChinaSea"/>
</River>
```

The above states that:

The Yangtze emptiesInto the EastChinaSea.

If emptiesInto and feedsFrom are defined to be Inverse properties then we can infer that:

The EastChinaSea feedsFrom the Yangtze.
emptiesInto <---> feedsFrom (Inverse Properties)

The general case:

River emptiesInto BodyOfWater

A specific instance:

EastChinaSea feedsFrom Yangtze

Yangtze emptiesInto EastChinaSea
Inverse Functional Properties

Properties:
- emptiesInto: BodyOfWater (functional)
- feedsFrom: River

An Inverse Functional property - for a range value the domain is unique.
These two independent documents discuss "feeding from" the Yangtze:

```
<?xml version="1.0"?>
<Sea rdf:ID="EastChinaSea"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <feedsFrom>
    <River rdf:about="http://www.china.org/rivers#Yangtze"/>
  </feedsFrom>
</Sea>
```

```
<?xml version="1.0"?>
<Sea rdf:ID="S1001-x-302"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <feedsFrom>
    <River rdf:about="http://www.china.org/rivers#Yangtze"/>
  </feedsFrom>
</Sea>
```
If feedsFrom has been defined to be InverseFunctional then we can infer that:

EastChinaSea = S1001-x-302.
2. RDF/OWL

Confused about the difference between a Functional property and an Inverse Functional property

Consider the birthdate property in this document:

```
<Person rdf:ID="JohnDoe">
  <birthdate>March 24, 1971</birthdate>
</Person>
```

A Person has a single birthdate. Therefore, birthdate is a Functional Property.

Question: Is birthdate an Inverse Functional Property? 
Answer: No. If birthdate was an Inverse Functional property then only one person could have a birthdate. There are many people that have the same birthdate. Thus, birthdate is not an Inverse Functional property.
An email address applies to only one person. Therefore, email is an Inverse Functional Property.

Question: Is email a Functional Property? Answer: No. If email was a Functional Property then a person could have only one email. Many people have multiple email addresses. Thus, email is not a Functional Property.
2. RDF/OWL

Defining Properties in OWL

- Recall that with RDF Schema the rdf:Property was used for both:
  - relating a Resource to another Resource
    - Example: The emptiesInto property relates a River to a BodyOfWater.
  - relating a Resource to an rdfs:Literal or a datatype
    - Example: The length property relates a River to a xsd:nonNegativeInteger.

- OWL decided that these are two classes of properties, and thus each should have its own class:
  - owl:ObjectProperty is used to relate a Resource to another Resource
  - owl:DatatypeProperty is used to relate a Resource to an rdfs:Literal or an XML Schema built-in datatype
An ObjectProperty relates one Resource to another Resource:

\[
\text{Resource} \xrightarrow{\text{ObjectProperty}} \text{Resource}
\]

A DatatypeProperty relates a Resource to a Literal or an XML Schema datatype:

\[
\text{Resource} \xrightarrow{\text{DatatypeProperty}} \text{Value}
\]
owl:ObjectProperty and owl:DatatypeProperty are subclasses of rdf:Property
2. RDF/OWL

Defining Properties in OWL vs. RDF Schema

```xml
<rdf:Property rdf:ID="emptiesInto">
  <rdfs:domain rdf:resource="#River"/>
  <rdfs:range rdf:resource="#BodyOfWater"/>
</rdf:Property>
```

```xml
<owl:ObjectProperty rdf:ID="emptiesInto">
  <rdfs:domain rdf:resource="#River"/>
  <rdfs:range rdf:resource="#BodyOfWater"/>
</owl:ObjectProperty>
```

```xml
<owl:DatatypeProperty rdf:ID="length">
  <rdfs:domain rdf:resource="#River"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
```

```xml
<owl:DatatypeProperty rdf:ID="length">
  <rdfs:domain rdf:resource="#River"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
```
2. RDF/OWL

Summary of Properties for the Water Taxonomy

- **NaturallyOccurringWaterSource**
  - Properties:
    - `connectsTo`: NaturallyOccurringWaterSource
    - (Symmetric)

- **Stream**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Functional)

- **River**
  - Properties:
    - `feedsFrom`: Stream
    - (Inverse Functional)

- **Tributary**
  - Properties:
    - `feedsFrom`: River
    - (Inverse Functional)

- **Lake**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Transitive)

- **Ocean**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Transitive)

- **Sea**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Transitive)

- **Brook**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Functional)

- **Rivulet**
  - Properties:
    - `emptiesInto`: BodyOfWater
    - (Inverse Functional)
2. RDF/OWL

Indicating that two instances are the same

Consider these two instance documents:

```
<?xml version="1.0"?>
<Sea rdf:ID="EastChinaSea"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.geodesy.org/water/naturally-occurring#">
    ...
</Sea>
```

```
<?xml version="1.0"?>
<Sea rdf:ID="S100-x-302"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.geodesy.org/water/naturally-occurring#">
    ...
</Sea>
```

Are they referring to the same Sea? In fact, S100-x-302 is the catalog number for the East China Sea. So, these two instances do refer to the same Sea. It would be useful if we could state in an instance document that it is describing the same thing as another instance document. We use owl:sameIndividualAs to express this sameness ...
We are clearly indicating that this instance is describing the same thing as the S100-x-302 instance.