2013 DCMI-AsiaPac Workshop on RDA, DC and Linked Data

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**

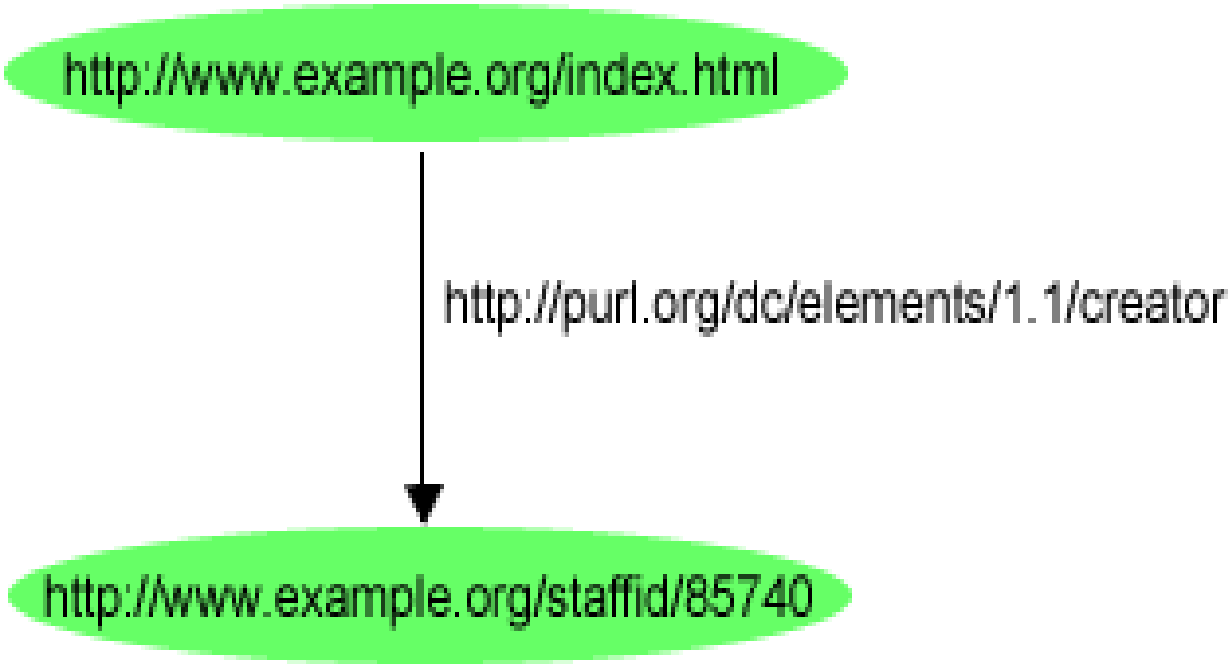
A Simple Example

- **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"

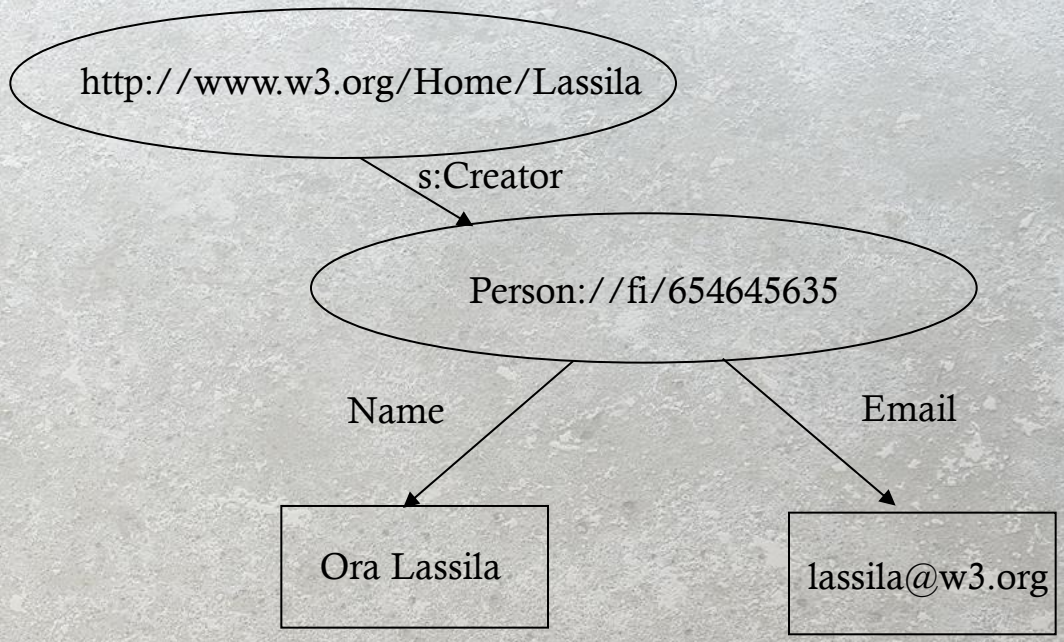
- **Directed graph**

A Simple Example

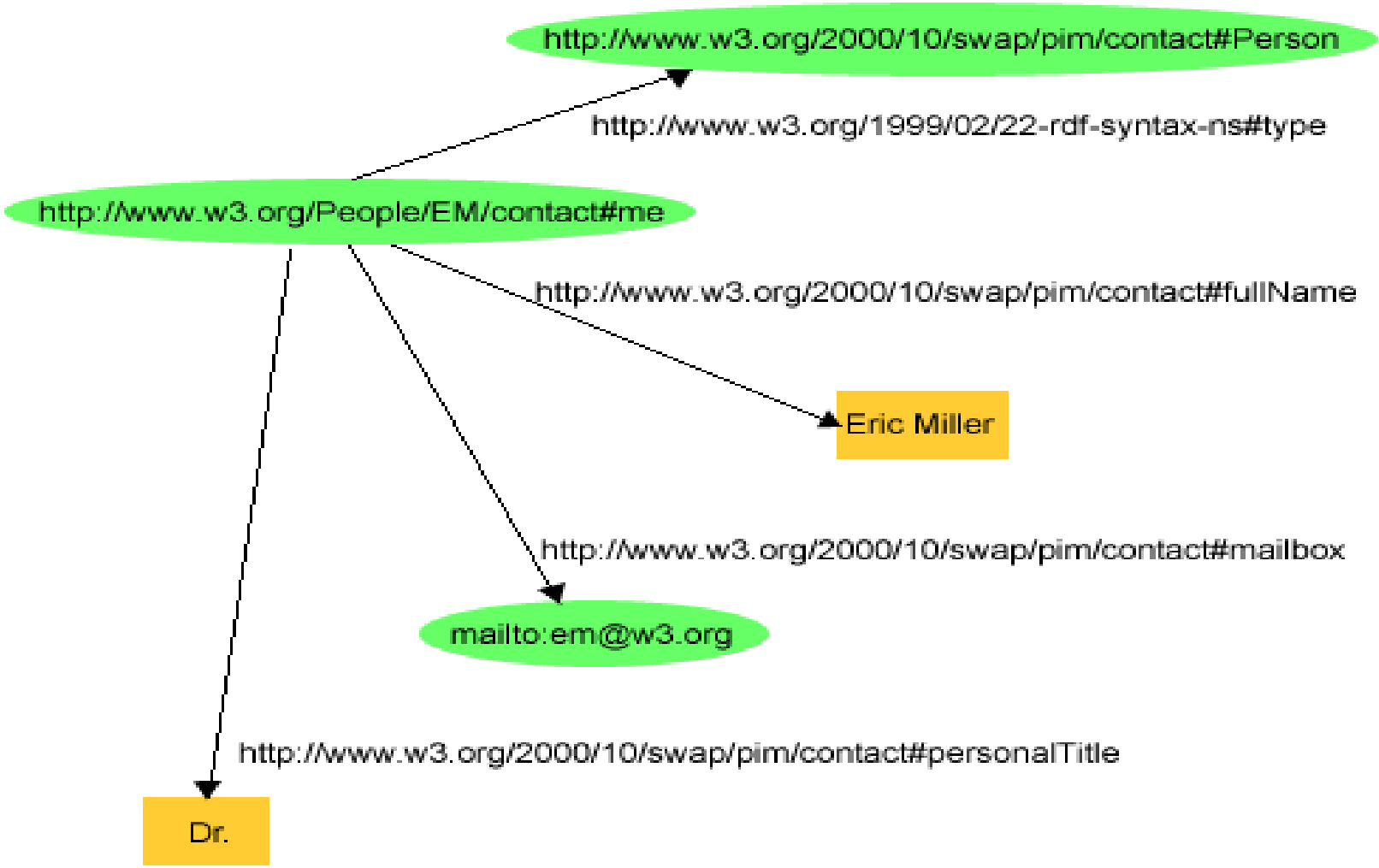


Another Example

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



RDF for Eric Miller



Specifying Classes

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

```
<rdfs:Class id="MyClass">
        <rdfs:label>My Class</rdfs:label>
        <rdfs:comment>John Cowan's demonstration  Class</rdfs:comment>
</rdfs:Class>
```

Vehicle Class Hierarchy

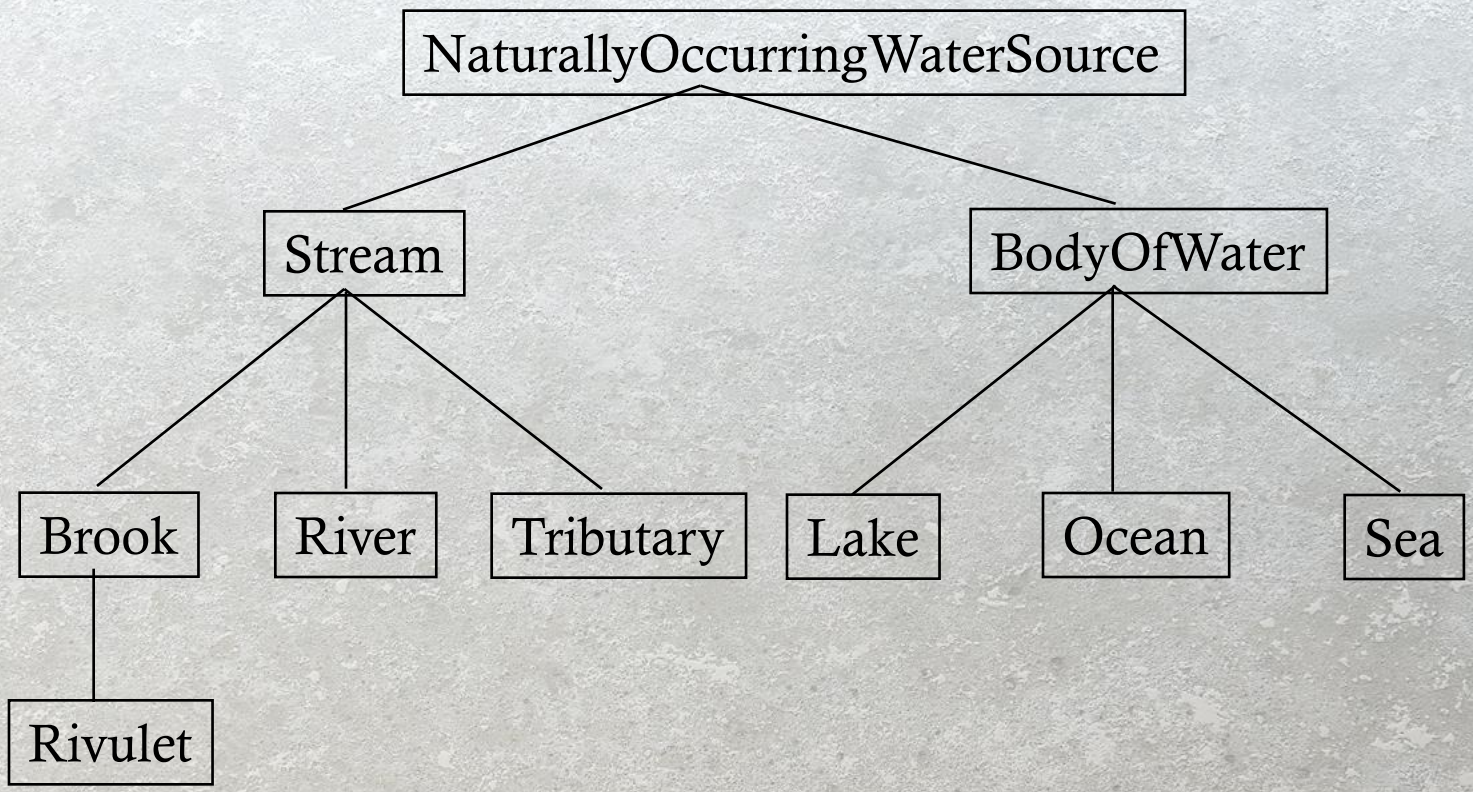
Specifying Properties

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

```
<rdfs:Property id="myProperty">
        <rdfs:comment>
                John Cowan's demo property
        </rdfs:comment>
        <rdfs:domain resource="#MyClass"/>
        <rdfs:range resource="..#Literal"/>
<rdfs:Property>
```

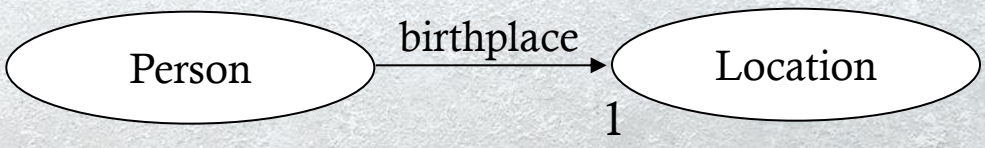
2.RDF/OWL

“Water Taxonomy" to explain OWL

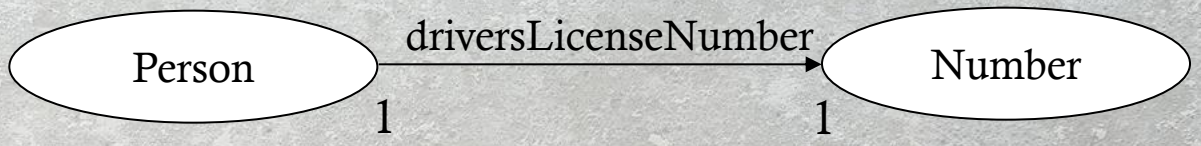


Notations used in this Lecture

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

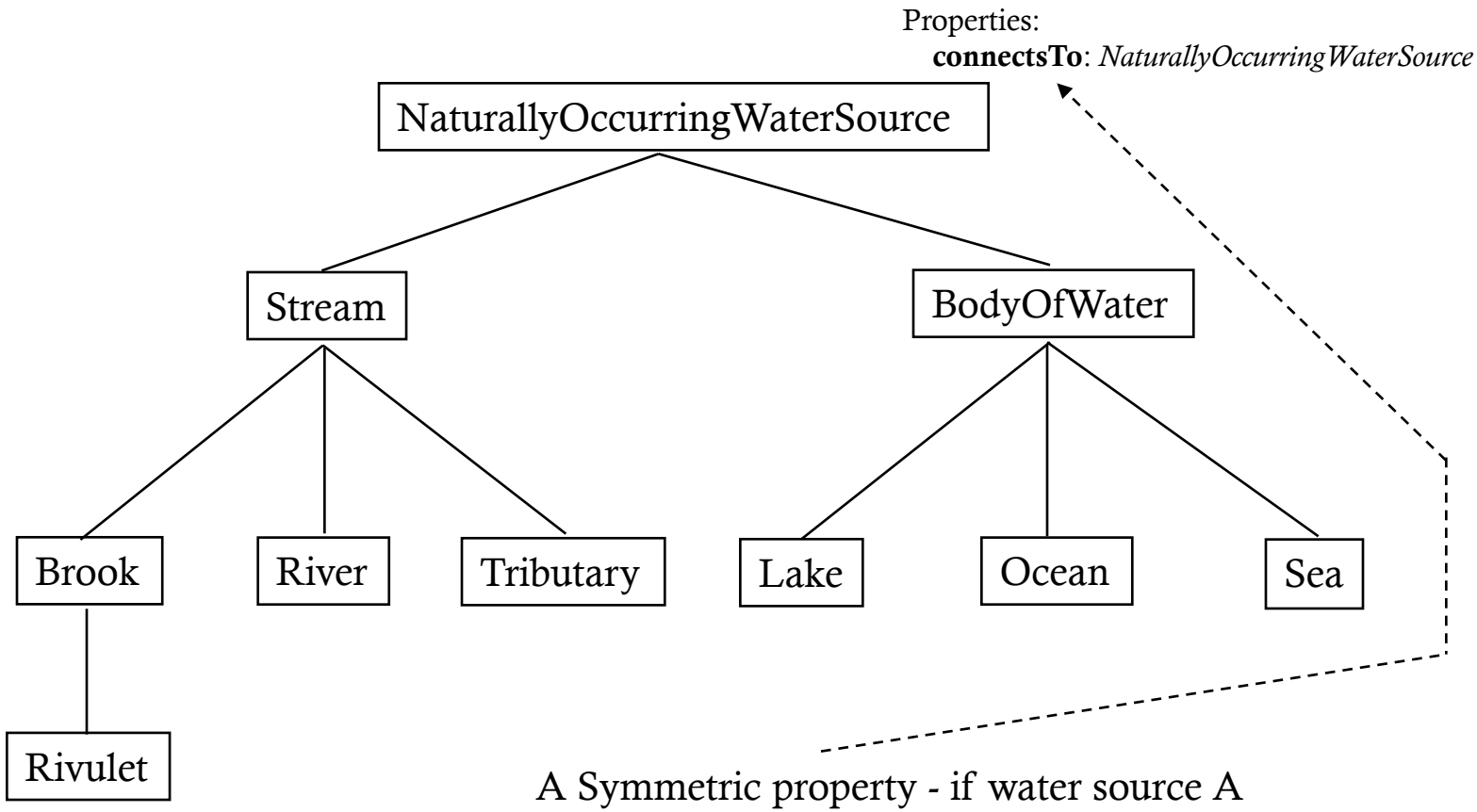


This notation is used to indicate that a person has only one driver's license number. Further, a driver's license number is associated with only one person:



Using OWL to Define Properties

Symmetric Properties



A Symmetric property - if water source A connectsTo water source B then water source B connects to water source A.

Symmetric Property

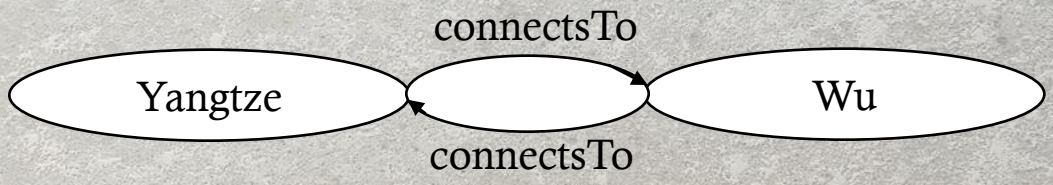
Assume that connectsTo has been *defined, in an OWL document, to be a Symmetric property*:

```
<?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>
```

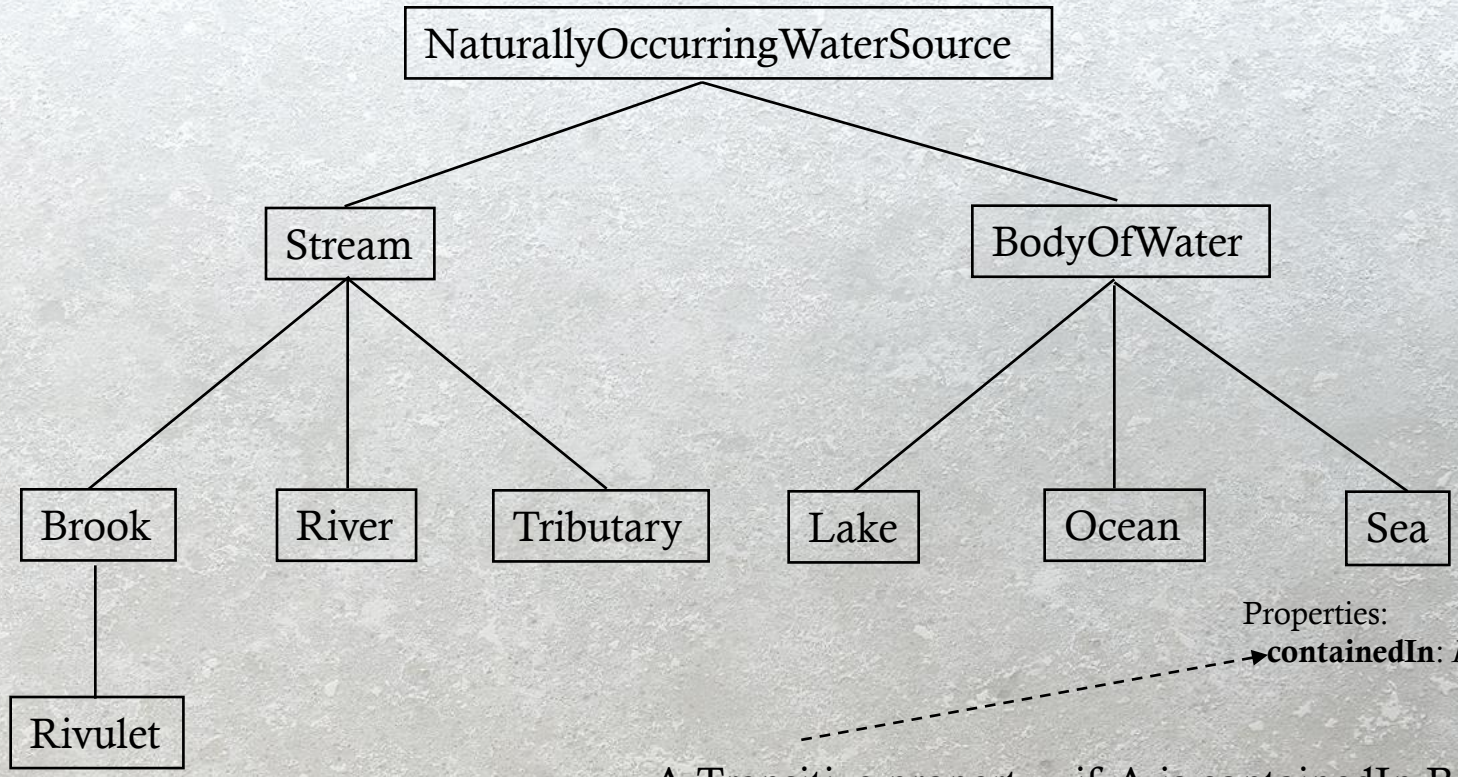
Yangtze.rdf

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

The Wu River connectsTo the Yangtze River.



Transitive Properties



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 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>
```

EastChinaSea.rdf

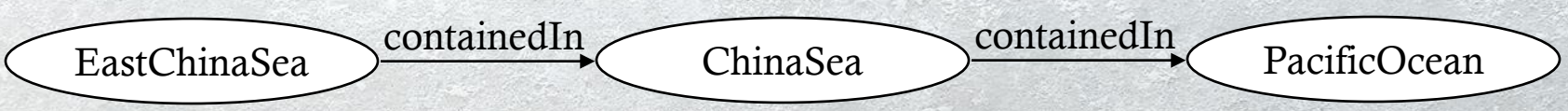
```
<?xml version="1.0"?>
<Sea rdf:about="http://www.china.gov#ChinaSea"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <containedIn>
        <Ocean rdf:about="http://www.geodesy.org#PacificOcean"/>
    </containedIn>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

ChinaSea.rdf

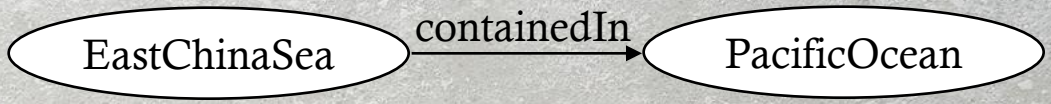
If containedIn is defined to be a Transitive property then we can infer that:

The EastChinaSea is containedIn the PacificOcean.

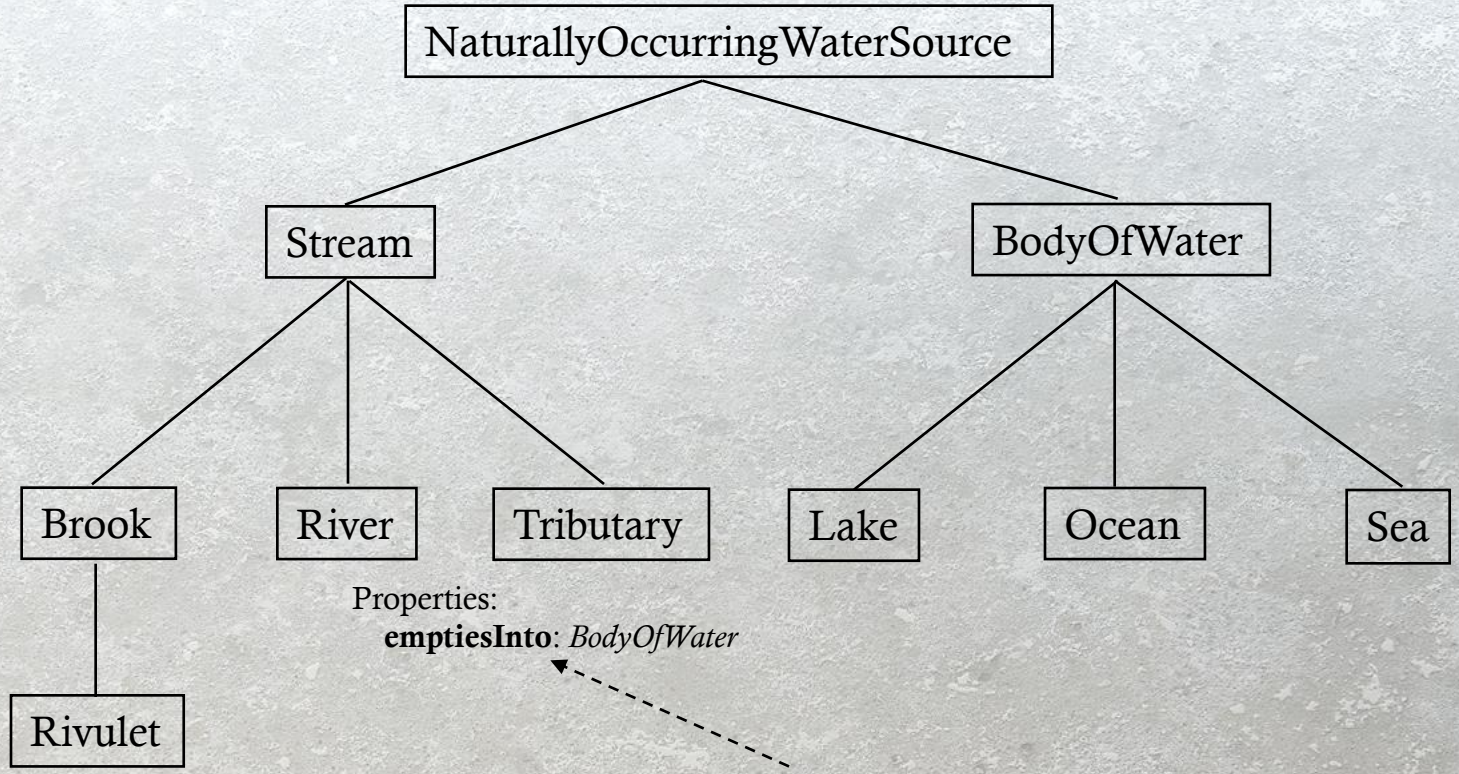
Transitive Property



If containedIn is defined to be Transitive, we can infer that:



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 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>
```

Yangtze-doc1.rdf

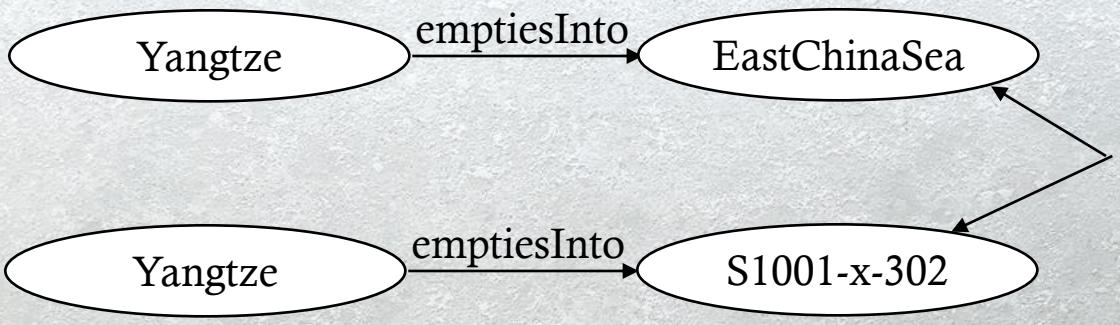
```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/rivers#Yangtze"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <emptiesInto rdf:resource="http://www.national-geographic.org#S1001-x-302"/>
</River>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

Yangtze-doc2.rdf

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

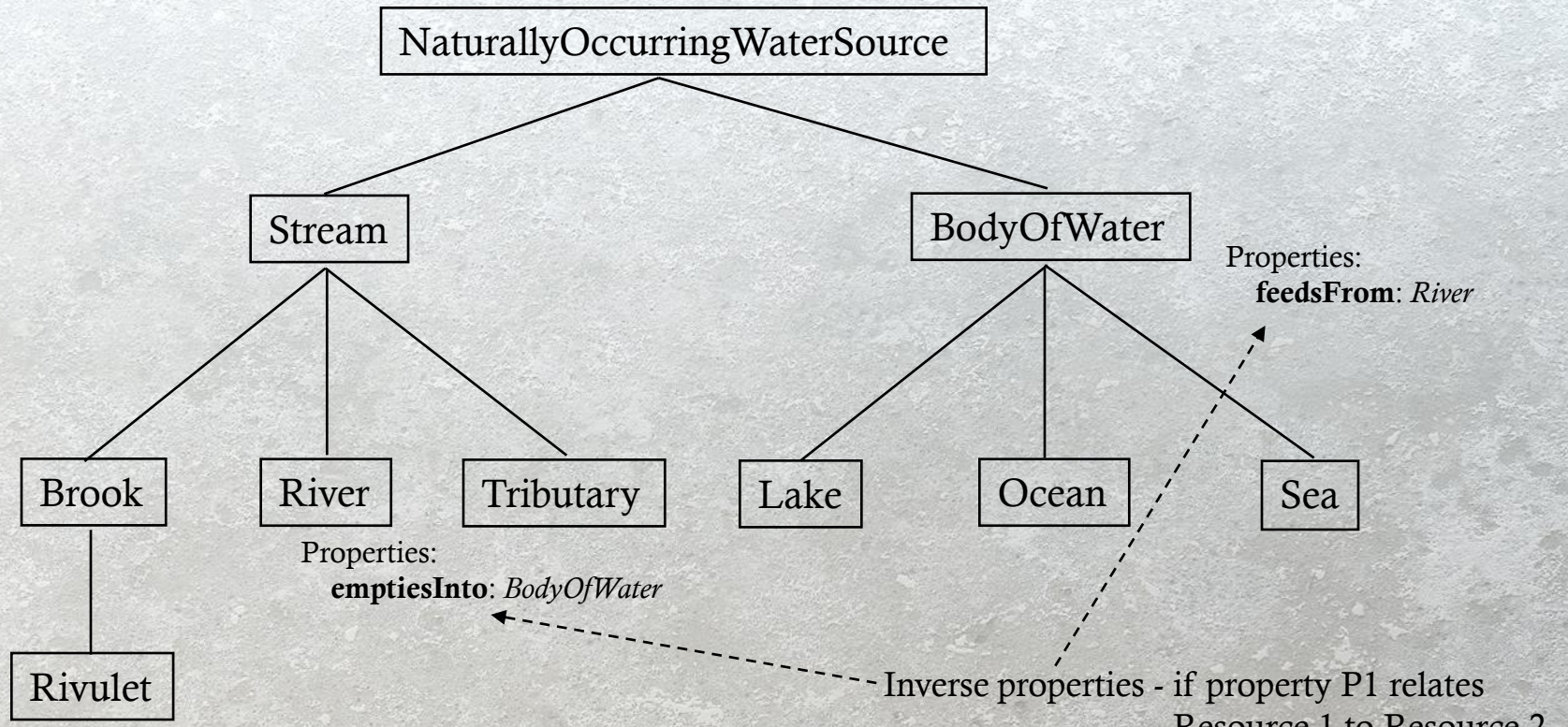
http://www.china.org/geography#**EastChinaSea** = http://www.national-geographic.org#**S1001-x-302**

Functional Property (cont.)



If emptiesInto has been defined to be Functional then we can infer that these two values must refer to the same thing.

Inverse Properties



Inverse Properties

Consider this document:

```
<?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>
```

Yangtze.rdf

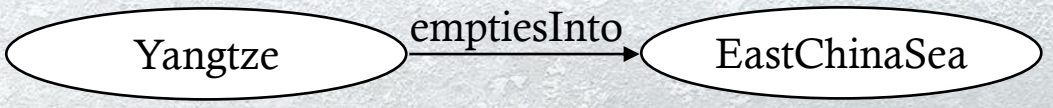
The above states that:

The Yangtze emptiesInto the EastChinaSea.

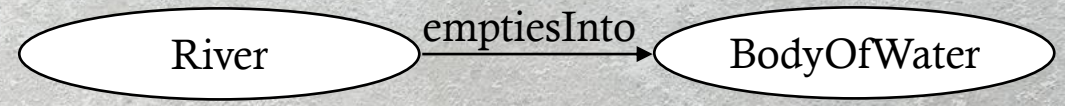
If emptiesInto and feedsFrom are defined to be Inverse properties then we can infer that:

The EastChinaSea feedsFrom the Yangtze.

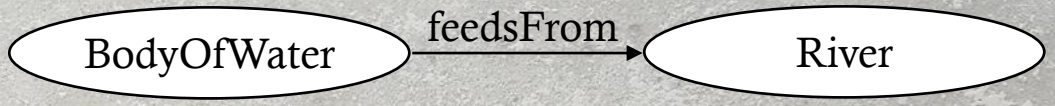
emptiesInto <---> feedsFrom (Inverse Properties)



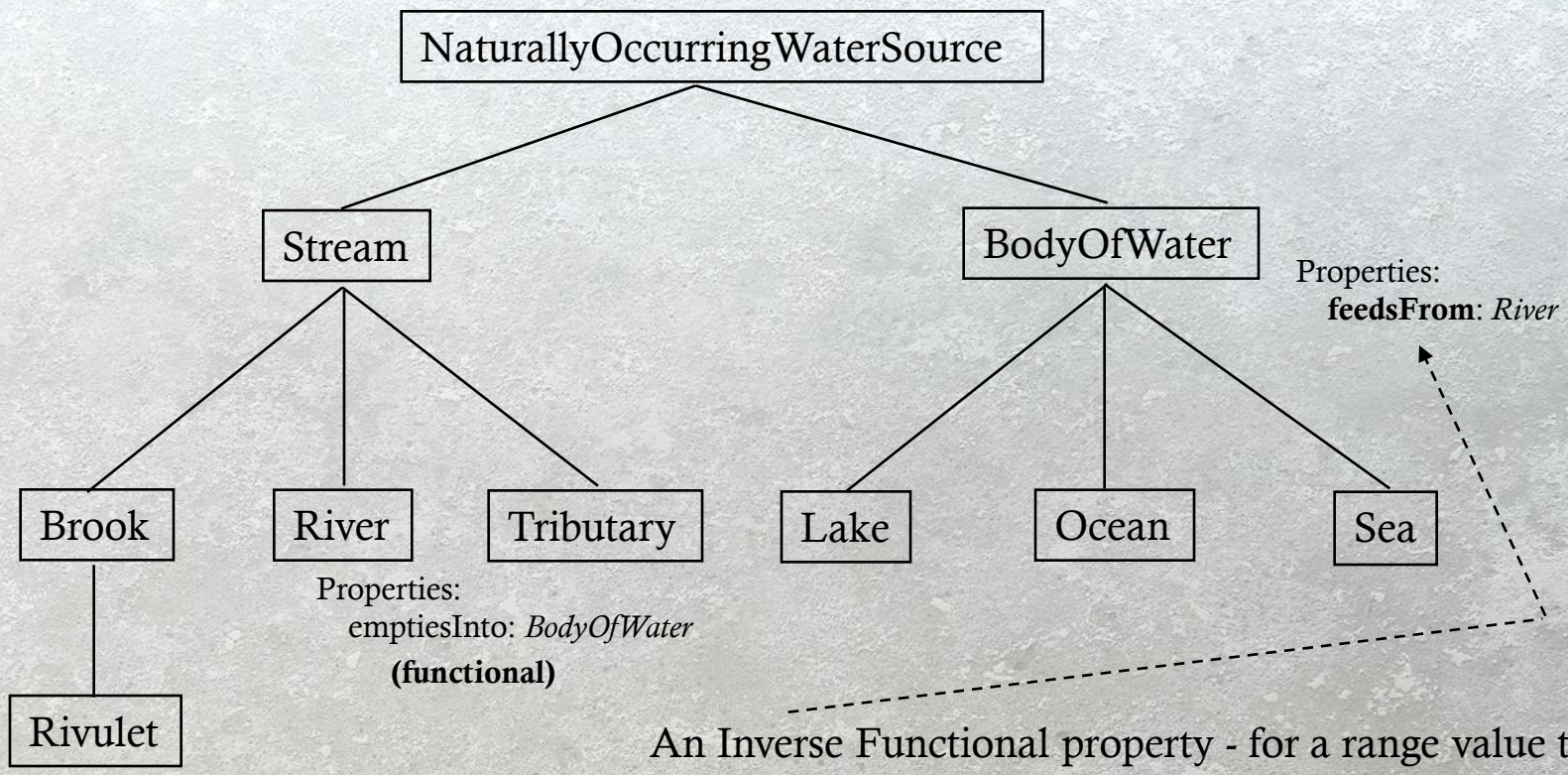
A specific instance:



The general case:



Inverse Functional Properties



Inverse Functional Properties

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>
```

EastChinaSea.rdf

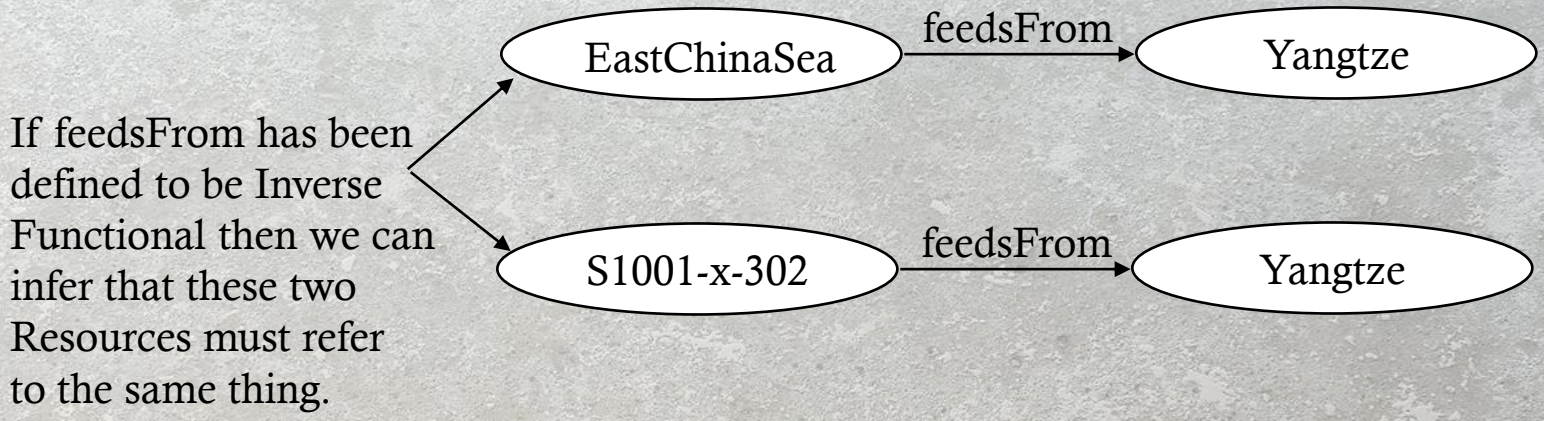
```
<?xml version="1.0"?>
<Sea rdf:ID="S1001-x-302"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/water/naturally-occurring#">
    <feedsFrom>
        <River rdf:about="http://www.china.org/rivers#Yangtze"/>
    </feedsFrom>
</Sea>
```

S1001-x-302.rdf

Inverse Functional Property (cont.)

If feedsFrom has been defined to be InverseFunctional then we can infer that:

EastChinaSea = S1001-x-302.



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.

Example of an Inverse Functional Property

Consider the email property in this document:

```
<Person rdf:ID="SallyJane">
    <email>sally-jane@yahoo.com</email>
</Person>
```

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.

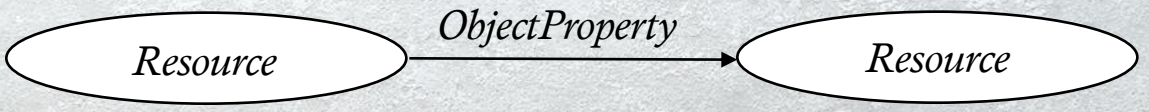
Defining Properties in OWL

- 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.

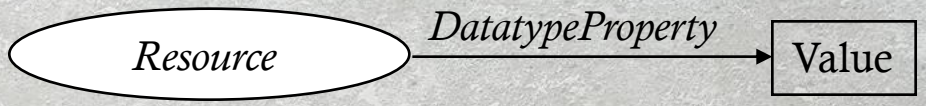
- OWL decided that these are two classes of properties, and thus each should have its own class:
  - owl:ObjectProperty is used to relate a Resource to another Resource
  - owl:DatatypeProperty is used to relate a Resource to an rdfs:Literal or an XML Schema built-in datatype

ObjectProperty vs. DatatypeProperty

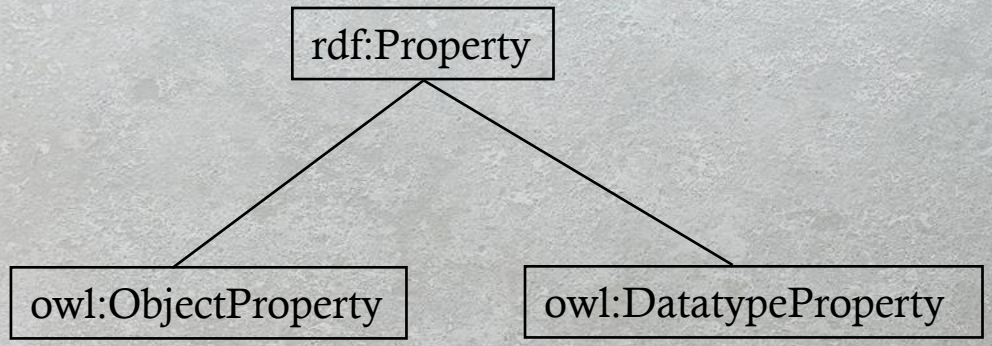
An ObjectProperty relates one Resource to another Resource:

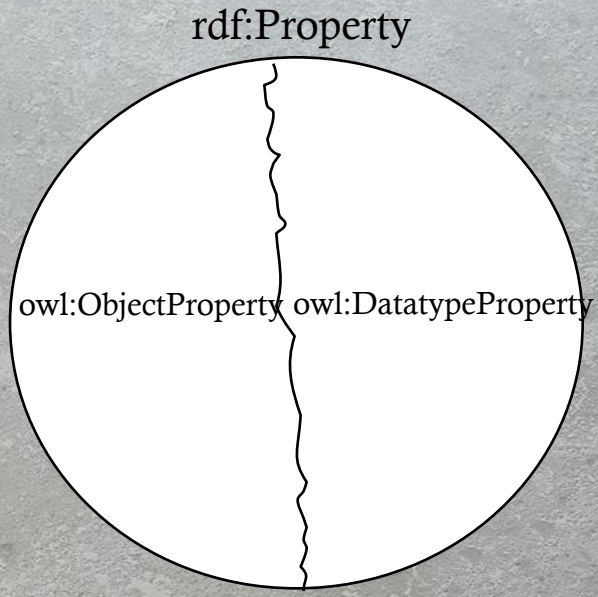


A DatatypeProperty relates a Resource to a Literal or an XML Schema datatype:



owl:ObjectProperty and owl:DatatypeProperty are subclasses of rdf:Property





Defining Properties in OWL vs. RDF Schema

RDFS

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

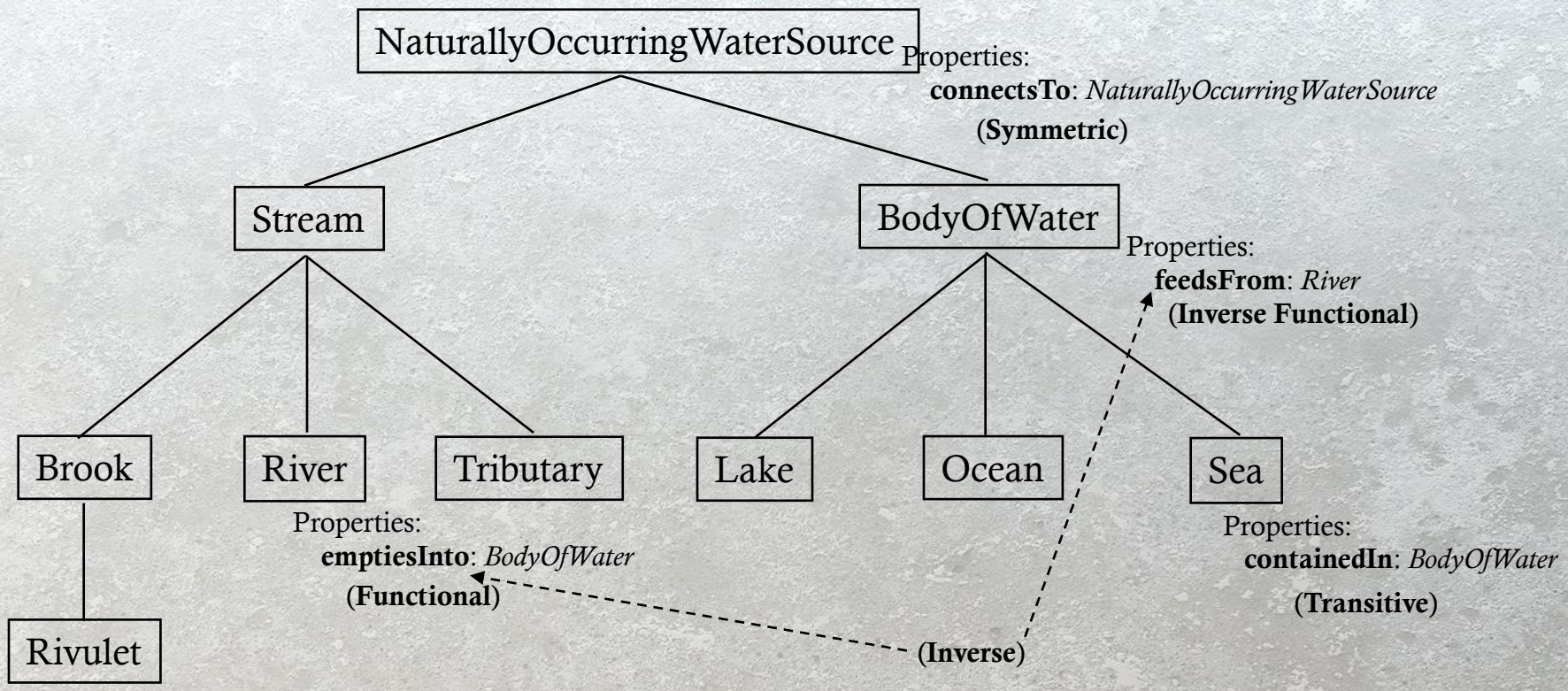
```
<rdf:Property rdf:ID="length">
    <rdfs:domain rdf:resource="#River"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</rdf:Property>
```

OWL

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

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

Summary of Properties for the Water Taxonomy



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 …

owl:sameAs

```
<?xml version="1.0"?>
<Sea rdf:ID="EastChinaSea"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns="http://www.geodesy.org/water/naturally-occurring#">
   <owl:sameAs rdf:resource="http://www.national-geographic.org#S1001-x-302"/>
    …
</Sea>
```

We are clearly indicating that this instance is describing the same thing as the S100-x-302 instance.