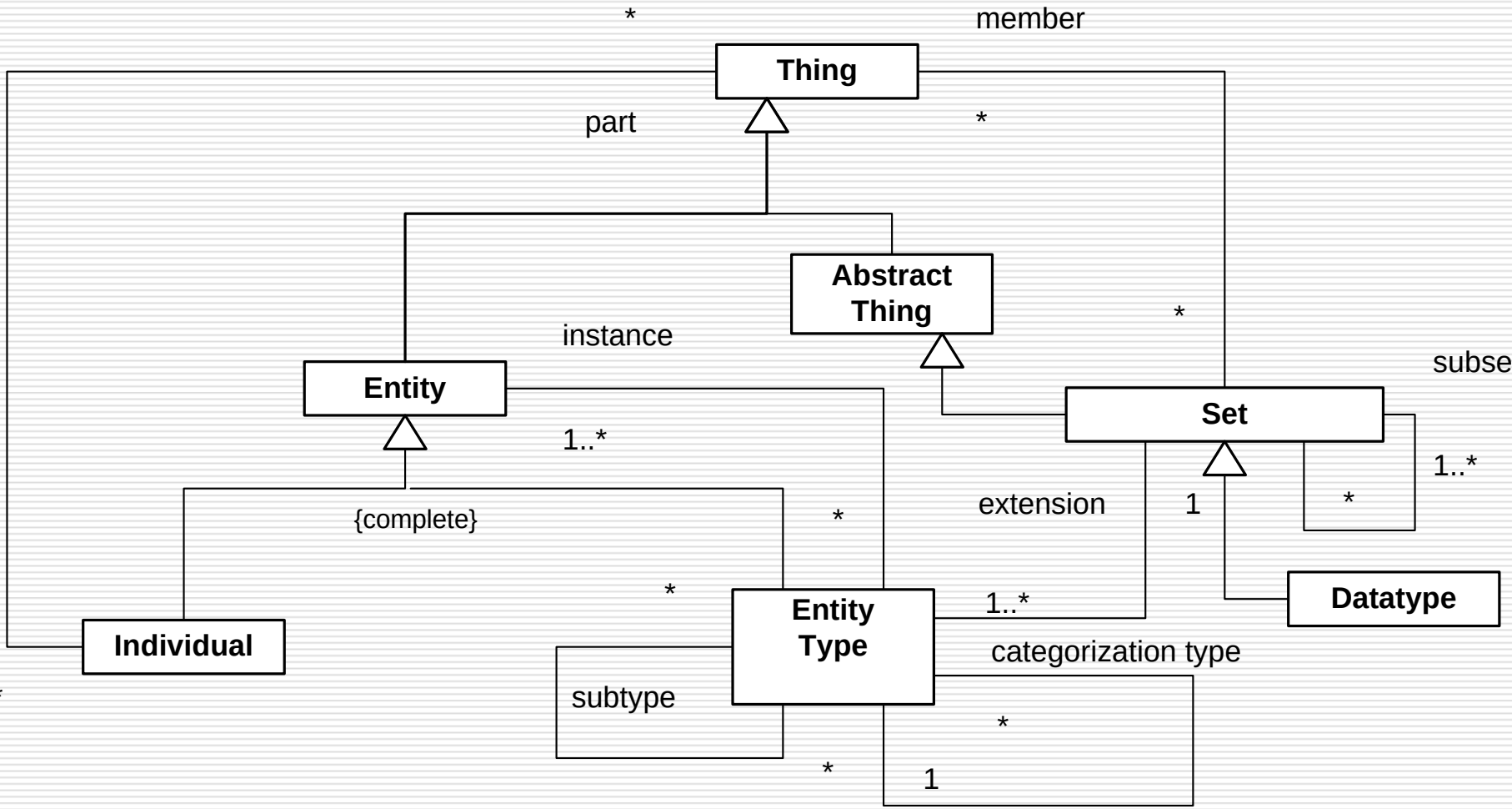
Rule Modeling and Markup Part II

Gerd Wagner
Brandenburg University of Technology
at Cottbus, Germany

Contents

- What are things, entities, individuals, classes (entity types)? Is it just set theory?
- UML language models for RDF, OWL and SWRL

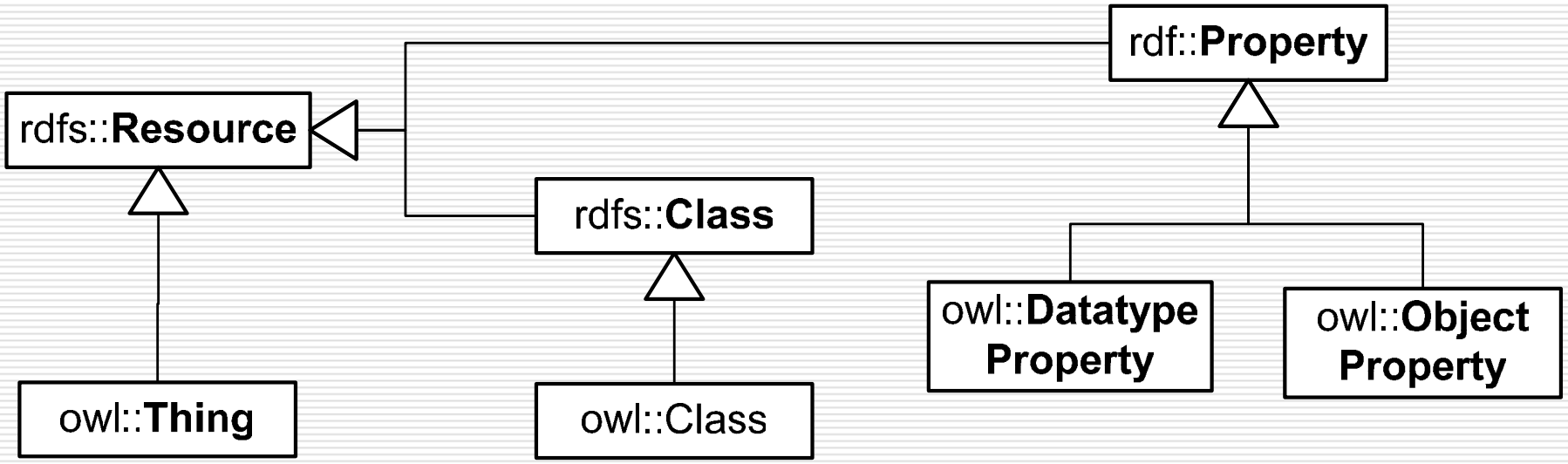
Foundational Ontology (from UFO)



Non-extensional classes

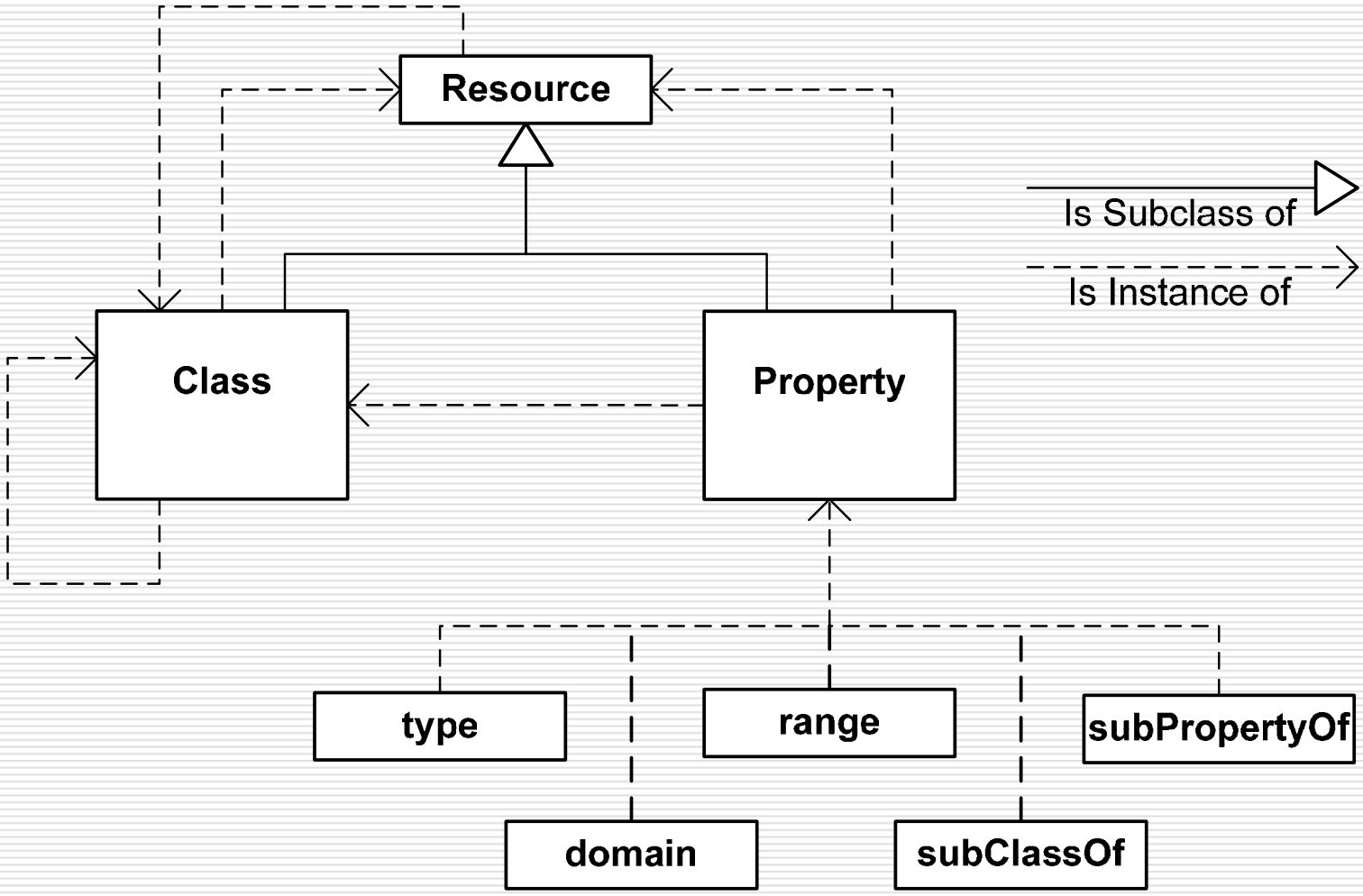
- Datatypes are sets, so they are extensional: if two datatypes, **`short`** and **`int`**, have the same data values, then they are equal: **`short`** = **`int`**
- Entity types are not sets, they are not extensional: if two entity types, **`Lecturer`** and **`Researcher`**, have the same instances, they are still not the same (since they have a different *intension*)

Differences between OWL and RDF



- RDF allows higher-order classes and non-extensional classes

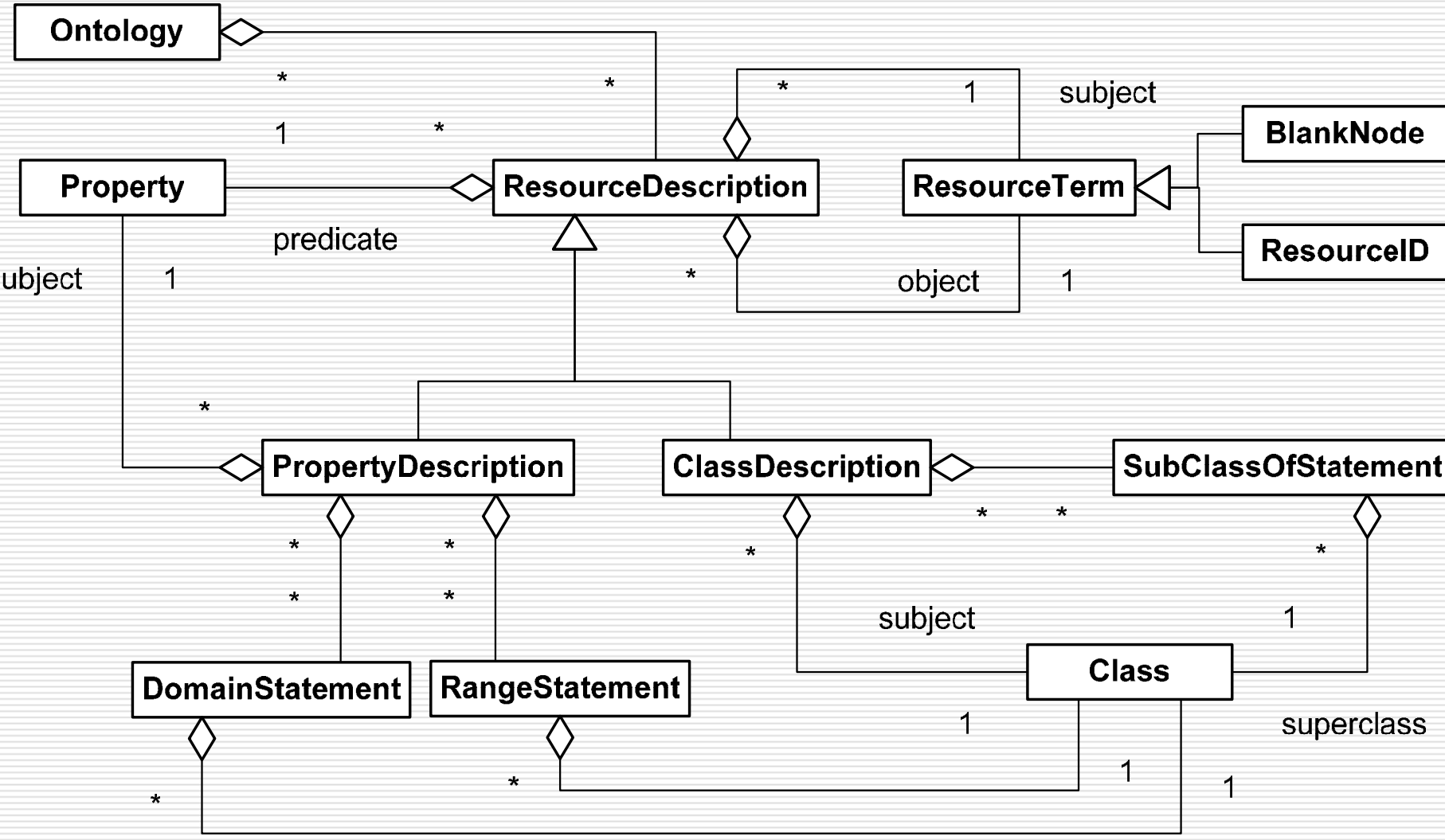
The most important RDF concepts



Language Models

- A fragment of UML, called "MOF", can be used to define the abstract syntax of a language in the form of a language model
- The MOF representation can be directly processed by modeling tools

What is an RDFS ontology?



Attribution Facts

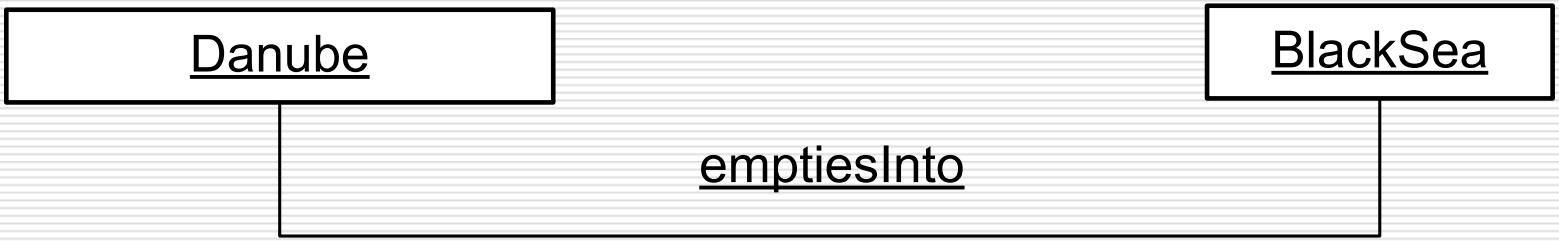
"Danube has length 2858"

Danube
length : Integer = 2858

```
<rdf:RDF xmlns:rdf="..." xmlns:rdfs="..."
xmlns:geo="http://geology.org/">
<rdf:Description rdf:about="geo:Danube">
   <geo:length rdf:datatype="xsd:integer">
   2858
    </geo:length>
</rdf:Description>
<rdf:RDF>
```

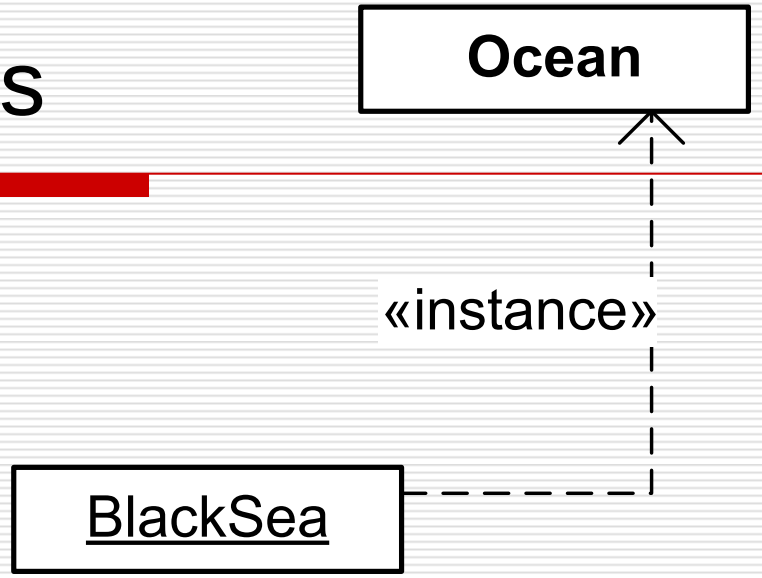
Association Facts

"Danube empties into Black Sea"




```
<rdf:Description rdf:about="geo:Danube">
<geo:emptiesInto rdf:resource="geo:BlackSea"/>
</rdf:Description>
```

Classification Facts



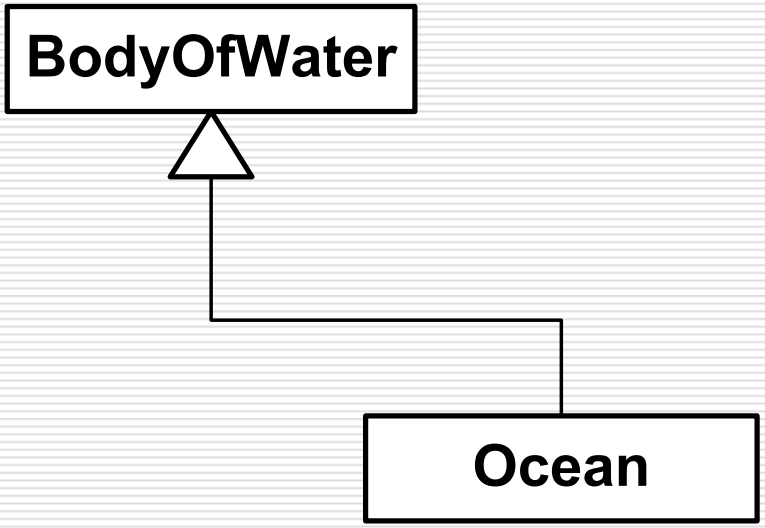
"Black Sea is classified as Ocean"

```
<rdf:Description rdf:about="geo:BlackSea">
<rdf:type rdf:resource="geo:Ocean"/>
</rdf:Description>
```

or, alternatively:

```
<geo:Ocean rdf:about="geo:BlackSea"/>
```

Generalization Statement

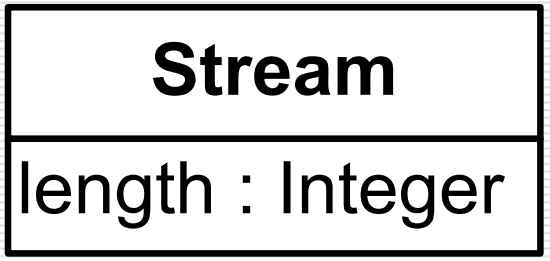


"Ocean is a category of body of water"

"category" = "subclass"

```
<rdf:Description rdf:about="geo:Ocean">
<rdfs:subClassOf rdf:resource="geo:BodyOfWater"/>
</rdf:Description>
```

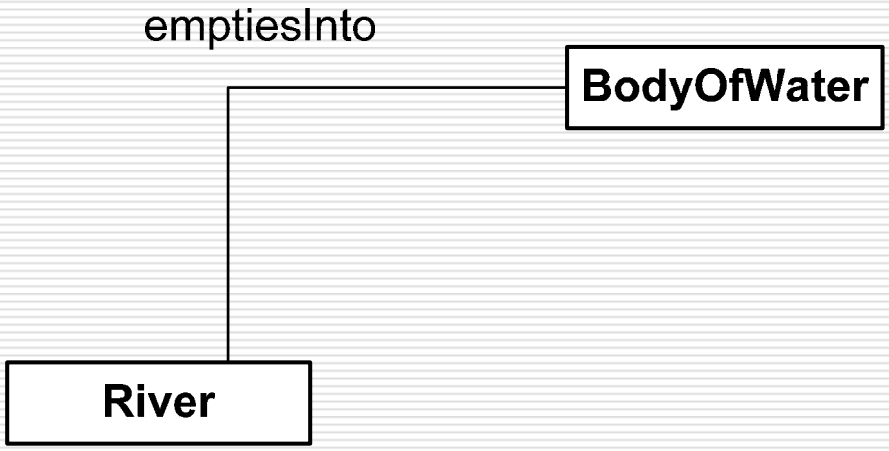
Attribution Fact Type



"Stream has Integer as length"

```
<rdf:Property rdf:ID="length">
        <rdfs:domain rdf:resource="#Stream"/>
        <rdfs:range rdf:resource="rdfs:Literal"/>
</rdf:Property>
```

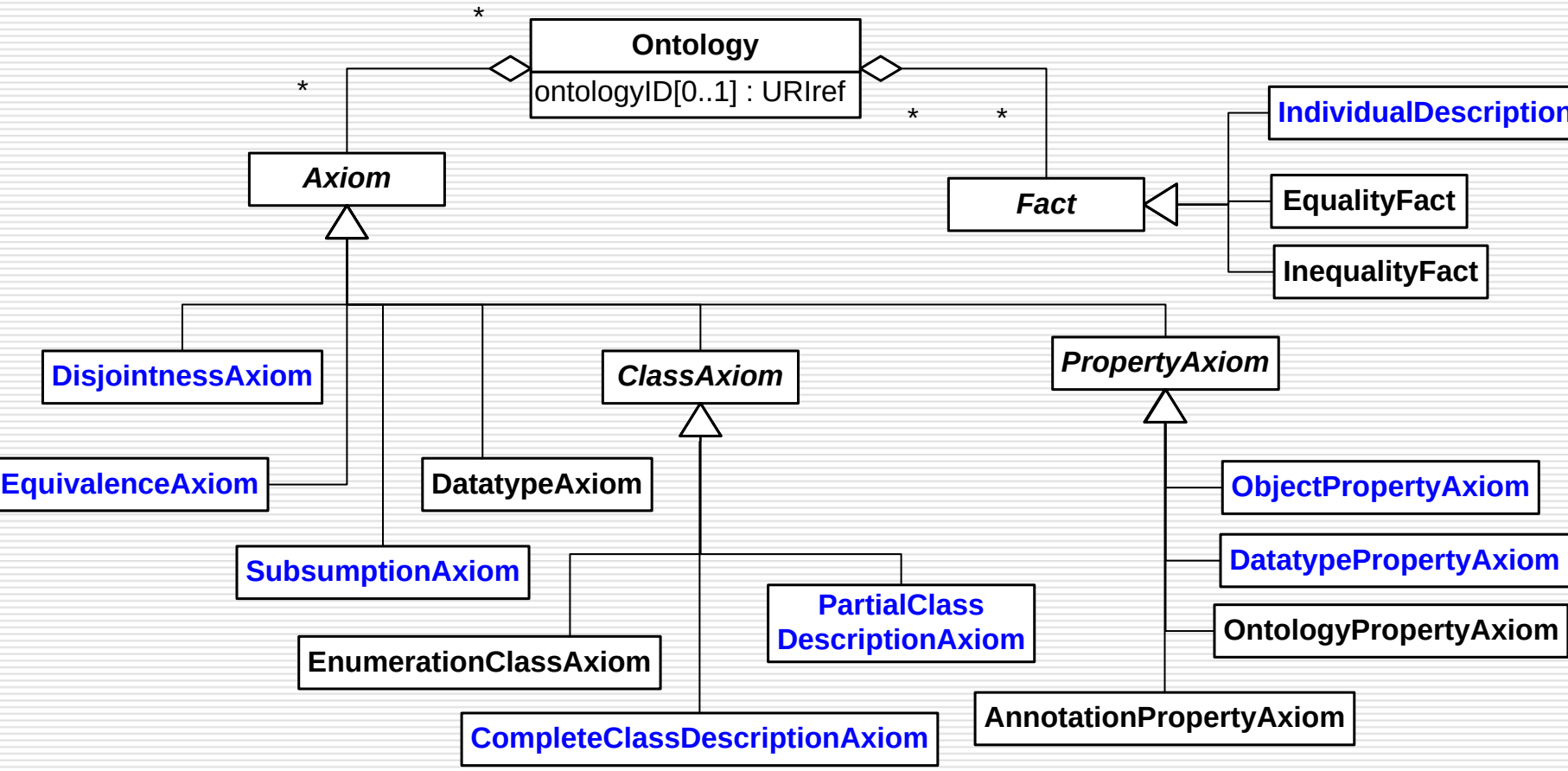
Association Fact Type



"River empties into BodyOfWater"

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

What is an OWL ontology?



Defining Datatype Properties (Attributes)

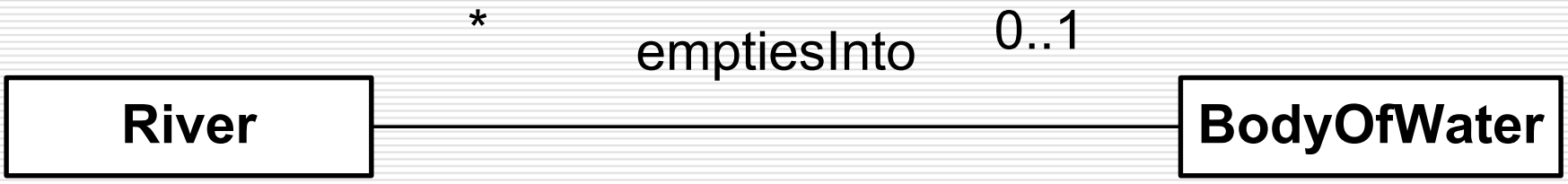
Untyped:

```
<owl:DatatypeProperty rdf:ID="length"/>
```

Typed (with functionality constraint):

```
<owl:DatatypeProperty rdf:ID="length">
   <rdfs:domain rdf:resource="#Stream"/>
   <rdfs:range rdf:resource="xsd:integer"/>
   <rdf:type rdf:resource="owl:FunctionalProperty"/>
</owl:DatatypeProperty>
```

Defining Object Properties (Binary Associations)



Untyped:

```
<owl:ObjectProperty rdf:ID="emptiesInto"/>
```

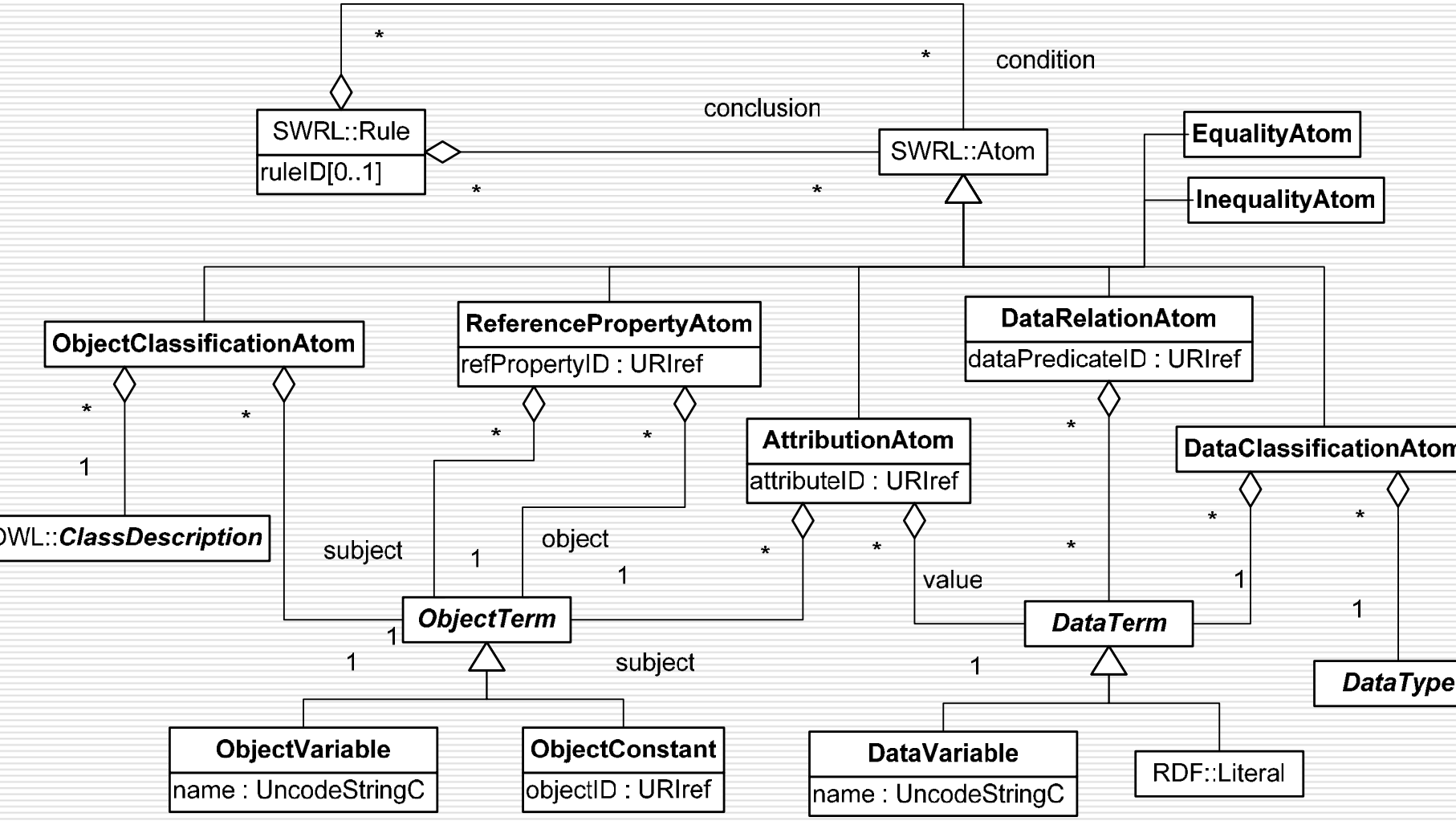
Typed (with functionality constraint):

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

Rule Language Standards

- The OMG recommends ***OCL*** as a PIM language for expressing integrity rules and derivatuion rules in conjunction with UML models
- The W3C has not yet recommended a general rule language for the (Semantic) Web
- There is a W3C member submission, the Semantic Web Rule Language (***SWRL***), which is based on OWL
- Also, the ***RuleML*** Initiative has defined a general markup language, RuleML 0.89, for expressing various kinds of derivation rules

SWRL


SWRL::Rule
ruleID[0..1]
conclusion
condition
SWRL::Atom
EqualityAtom
InequalityAtom
ObjectClassificationAtom
ReferencePropertyAtom
refPropertyID : URIref
DataRelationAtom
dataPredicateID : URIref
AttributionAtom
attributeID : URIref
DataClassificationAtom
OWL::ClassDescription
subject
object
value
ObjectTerm
DataTerm
DataType
ObjectVariable
name : UncodeStringC
ObjectConstant
objectID : URIref
DataVariable
name : UncodeStringC
RDF::Literal

Example 1 in OWL

```
<owl:Class rdf:ID="Rental"/>
<owl:Class rdf:ID="Person">
    <owl:disjointWith rdf:resource="#Rental"/>
</owl:Class>
<owl:Class rdf:ID="QualifiedDriver">
    <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="startDate">
    <rdfs:domain rdf:resource="#Rental"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="additionalDriver">
    <rdfs:domain rdf:resource="#Rental"/>
    <rdfs:range rdf:resource="#Person"/>
</owl:ObjectProperty>
...
```

Example 1 in OWL/SWRL (1)

```
<swrl:variable rdf:ID="r1"/> <swrl:variable rdf:ID="p1"/>
<ruleml:Imp>
<ruleml:body>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="#Rental"/>
    <swrl:argument1 rdf:resource="#r1"/>
  </swrl:ClassAtom>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="#Person"/>
    <swrl:argument1 rdf:resource="#p1"/>
  </swrl:ClassAtom>
  <swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate rdf:resource="#additionalDriver"/>
    <swrl:argument1 rdf:resource="#r1"/>
    <swrl:argument2 rdf:resource="#p1"/>
  </swrl:IndividualPropertyAtom>
</ruleml:body>
```

The rule condition ("body")

Example 1 in OWL/SWRL (2)

The rule conclusion ("head") :

```
<ruleml:head>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource=”#QualifiedDriver”/>
    <swrl:argument1 rdf:resource=”#p1”/>
  </swrl:ClassAtom>
</ruleml:head>
</ruleml:Imp>
```

Exercise 2

1. Read the handout and try to understand the vocabulary used.
2. Do the rules:
   a) Identify the vocabulary fragment needed for the rule and draw just this fragment as a class diagram
   b) Express the rule in OCL
   c) Express the rule in SWRL

Strengths of OWL/SWRL

- OWL and SWRL have a rigorous logical semantics
- OWL inference engines can be used
  - to check the consistency of a UML class model that represents a business vocabulary
  - to discover derived model elements
- Combined OWL/SWRL tools may even do a better job on verifying UML, because they can also take OCL invariants and derivation expressions into consideration

Weaknesses of OWL/SWRL

- No n-ary association predicates
- Too much mathematical logic instead of computational logic
  - classical (2-valued) instead of partial (3-valued), note that SQL and OCL are 3-valued
  - no negation-as-failure concept
- Since there is no distinction between a definition (derivation rule) and an integrity constraint (they are all "axioms"), this may have undesirable consequences in applications

What's in the pipeline?

- The W3C plans to establish a working group for defining a rule language recommendation: will it be based on OWL (like SWRL) or rather on RDF?
- The OMG has started ongoing efforts to define
  - a CIM business rules language (BSBR-CFP), and
  - a PIM production rule language (PRR-CFP)

The Business Rules Proposal SBVR

- Semantics of Business Vocabularies and Rules (SBVR) is the major OMG submission for a business rules language
- SBVR is based on a "Structured English" vocabulary specification method
- Pros: It takes business semantics serious and attempts to combine it with formal logic
- Cons:
  - Its current version is too large and seems to contain many meta-theoretical concepts that should better not be included in a standard
  - it's not enough tailored towards PIM

Conclusion

- In the current and planned efforts of the OMG and W3C
  - there is not enough communication between the different groups
  - there is not enough pressure to be concerned with integration
- Integration of the proposed languages (OCL, SBVR, PRR, RDF, OWL, SWRL?, ...) is key to success!