Knowledge Representation
Part VIIc

OWL versus UML

Jan Pettersen Nytun, UiA
[1]: “The languages were devised to fulfill different purposes. While **OWL** supports the representation of knowledge about a system, **UML** was developed primarily to support the construction of a (software) system.”
Open-World versus Closed-World Interpretation Assumptions

[1]: “The UML is oriented towards data modeling and system construction ... knowledge is implicitly viewed as being complete. OWL, in contrast, interprets models as potentially representing partial knowledge.”
[1] Unique Name Assumption & Synonyms

• OWL allows the definition of synonyms for classes, properties and individual descriptions.

• UML assumes that every name has a unique interpretation, i.e., the unique name assumption.
[1] Sufficient Conditions & Defined vs. Primitive Concepts

• One of the purposes of OWL ontologies is to facilitate automatic classification.

• The UML does not provide native assistance in the definition of sufficient conditions or defined classes.
UML $\rightarrow$ OWL

Person

- name : String

Institution

- name : String

employee $\rightarrow$ employedAt

* $\rightarrow$ 1
In UML a class is a namespace by itself. I.e., property `name` in class `Person` and property `name` in class `Institution` are different entities.

Properties in OWL are globally scoped. Why is the following not a good idea?

```plaintext
Properties in OWL are globally scoped. Why is the following not a good idea?
```
Simulate the namespace of the class by using class name as prefix.
Multiplicity of `hasPersonName` is by default one.

```
:Person rdf:type owl:Class ;
rdfs:subClassOf [ rdf:type owl:Restriction ;
owl:onProperty :hasPersonName ;
owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
owl:onDataRange xsd:string ] .
```
Starting the reasoner – what is inferred?
Remove Person as Domain and run reasoner – what is inferred?
Nothing is inferred!
Make Person a defined class.
Run reasoner – what is inferred?
We have an individual with one `hasPersonName` property specified. But still this individual is not classified as a Person - why?
• OWL makes the open world assumption.

• In our example this means: There may be more hasPersonName properties for our individual.
It is possible to specify that the individual has only "Jan Pettersen"^^xsd:string as value for property hasPersonName.
Better solution:
Person

name : String

* employee

Institution

name : String

employedAt 1

**“unique naming”**

Person

name : String

* institutionHas-EmployedPerson

Institution

name : String

personIsEmployed-AtInstitution 1
[1] A Detailed Comparison of UML and OWL, Kilian Kiko & Colin Atkinson,
https://ub-madoc.bib.uni-mannheim.de/1898/1/TR2008_004.pdf