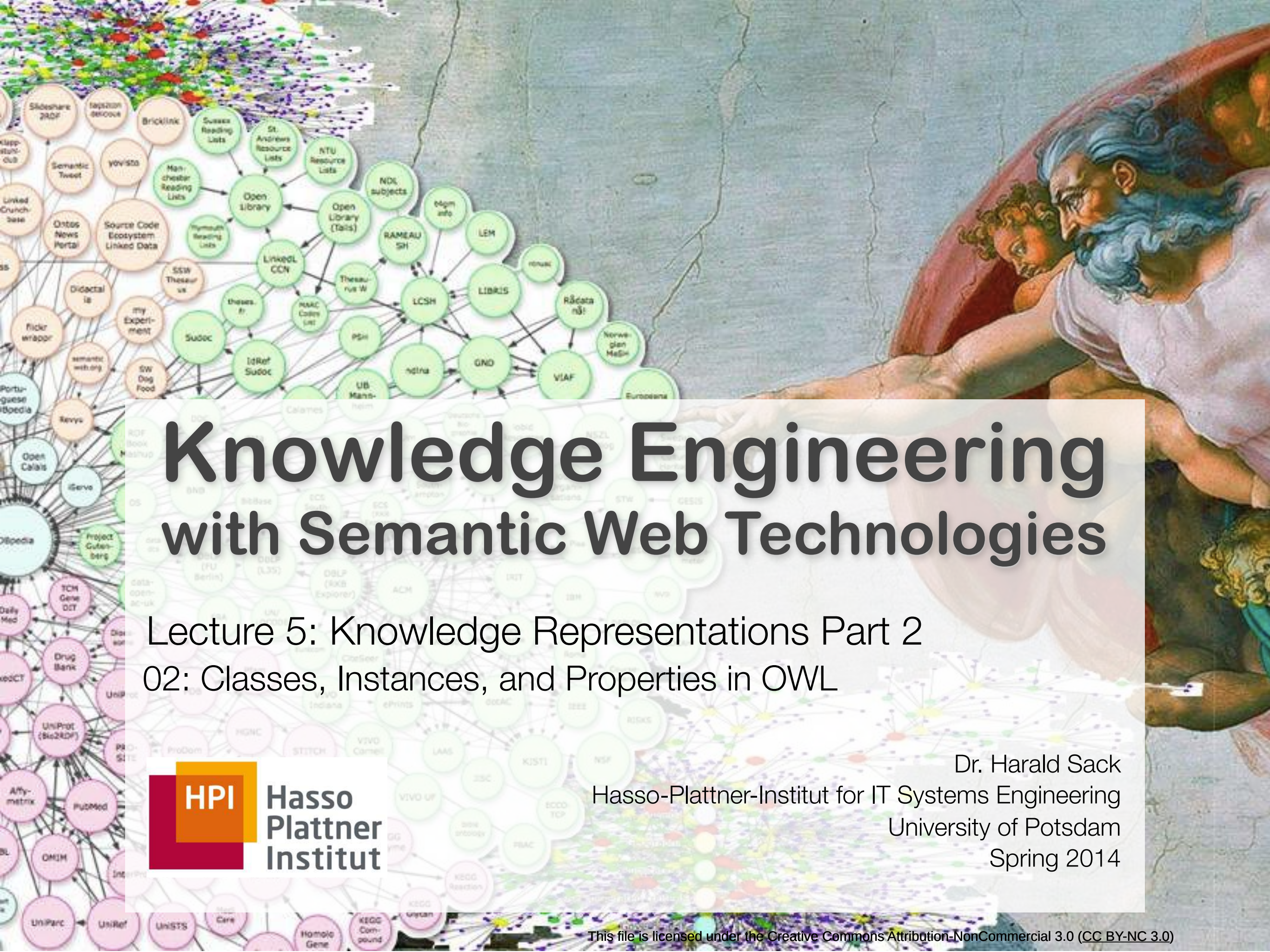




Knowledge Engineering with Semantic Web Technologies

Lecture 5: Knowledge Representations Part 2 02: Classes, Instances, and Properties in OWL



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 5: Knowledge Representations II

Open HPI Course: Knowledge Engineering with Semantic Web Technologies

02 Classes, Instances, and Properties in OWL

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 05: Knowledge Representations Pt.2

- Ontology axioms consist of the following three building blocks:
 - **Classes**
 - comparable with classes in RDFS
 - **Individuals**
 - comparable with objects in RDFS
 - **Properties**
 - comparable with properties in RDFS

OWL – Classes

- there exist two predefined classes
 - `owl:Thing` (class that contains all individuals) $\top$
 - `owl:Nothing` (empty class) $\perp$
- Definition of a class

```
:Book a owl:Class .
```

OWL – Individuals

- Definition of individuals via class membership

```
:NineteenEightyfour a :Book .
```

Book(NineteenEightyfour)

- Individuals can also be defined without direct class membership as **named entity**

```
:HaraldSack a owl:NamedIndividual .
```

OWL – Properties

- 7
- there exist two property variants:
 - object properties
 - datatype properties
 - **Object properties** are defined like classes

```
:author    a    owl:ObjectProperty .
```

- Domain and Range of object properties

```
:author    a    owl:ObjectProperty ;  
            rdfs:domain :Book ;  
            rdfs:range  :Writer .
```

OWL – ObjectProperties

8

- Domain and Range of object properties

```
:author    a owl:ObjectProperty ;  
            rdfs:domain :Book ;  
            rdfs:range  :Writer .
```

- in DL domain and range can be defined in the following way:
 - **Domain:** $\exists \text{author} . \top \sqsubseteq \text{Book}$
 - **Range:** $\top \sqsubseteq \forall \text{author} . \text{Writer}$

OWL – DatatypeProperties

- Datatype properties have datatypes as range

```
:publicationYear a owl:DatatypeProperty .
```

- Domain and range of datatype properties

```
:publicationYear a owl:DatatypeProperty ;  
    rdfs:domain :Book ;  
    rdfs:range xsd:integer .
```

- Many XML datatypes can be used.

$$\exists \text{publicationYear} . \top \sqsubseteq \text{Book}$$
$$\top \sqsubseteq \forall \text{publicationYear} . \text{Integer}$$

```
:Book a owl:Class .  
:Writer a owl:Class .  
:GeorgeOrwell a Writer .  
:author a owl:ObjectProperty ;  
        rdfs:domain :Book ;  
        rdfs:range :Writer .  
:publicationYear a owl:DatatypeProperty ;  
        rdfs:domain :Book ;  
        rdfs:range xsd:integer .  
:NineteenEightyFour a :Book ;  
        :author :GeorgeOrwell ;  
        :publicationYear 1948 .
```

- 11
- Properties in general are not functional

```
:Genre a owl:Class .  
:PoliticalFiction a :Genre .  
:ScienceFiction a :Genre .  
:DystopianFiction a :Genre .  
  
:literaryGenre a owl:ObjectProperty ;  
    rdfs:domain :Book ;  
    rdfs:range :Genre .  
  
:NineteenEightyFour a :Book ;  
    :literaryGenre :PoliticalFiction ;  
    :literaryGenre :ScienceFiction ;  
    :literaryGenre :DystopianFiction .
```

```
:Novel a owl:Class ;  
      rdfs:subClassOf :Book .  
:Book a owl:Class ;  
      rdfs:subClassOf :Work .  
:Work a owl:Class .
```

$\text{Novel} \sqsubseteq \text{Book}$

$\text{Book} \sqsubseteq \text{Work}$

- via inference it can be entailed that „Novel“ is also a subclass of „Work“

```
:Book a owl:Class .  
:Writer a owl:Class .  
:Novel a owl:Class ;  
        rdfs:subClassOf :Book .  
:Poet a owl:Class ;  
        rdfs:subClassOf :Writer .  
  
:Book owl:disjointWith :Writer .
```

$\text{Novel} \sqsubseteq \text{Book}$
 $\text{Poet} \sqsubseteq \text{Writer}$
 $\text{Book} \sqcap \text{Writer} \sqsubseteq \perp$

- via inference it can be entailed that „Novel“ and „Poet“ are also disjoint classes.

- OWL provides a **shortcut** to define several classes to be disjunctive

```
[ ] a owl:AllDisjointClasses ;  
    owl:members  
      ( :Book  
        :Writer  
        :Vegetable  
        :Furniture  
        :Car ) .
```

OWL – Class Hierarchies and Equivalence

15

```
:Writer a owl:Class .  
:Author a owl:Class .  
:Poet a owl:Class ;  
      rdfs:subClassOf :Writer .  
:Writer owl:equivalentClass :Author .
```

Poet $\sqsubseteq$ Writer
Writer $\equiv$ Author

- via inference it can be entailed that „Poet“ is also an „Author“

OWL – Individuals

16

```
:NineteenEightyfour a :Novel ;  
    :author :GeorgeOrwell ;  
    :publicationYear 1948 ;  
    owl:sameAs :ARX012345 .  
  
:Novel a owl:Class ;  
    rdfs:subClassOf :Book .  
  
:Book a owl:Class .
```

- via inference it can be entailed that „ARX012345“ is a „Book“
- Difference of Individuals via **owl:differentFrom**

```
:ARX012345 a :Novel ;  
    owl:differentFrom :ARX012346 .
```

OWL – Individuals

- OWL provides a shortcut to define several individuals to be different

```
[ ] a owl:AllDifferent ;  
    owl:distinctMembers  
      ( :NineteenEightyfour  
        :AnimalFarm  
        :BraveNewWorld  
        :Simulacron5 ) .
```



03 OWL Complex Classes and Properties