

The image is a composite. On the left, there is a large, intricate network graph with numerous nodes (circles) of various colors (blue, green, orange, pink) and many connecting lines, representing a semantic web or knowledge graph. On the right, there is a classical painting of a man with a long white beard and hair, wearing a pink robe, looking down at a woman who is reclining. The central white text box contains the following text:

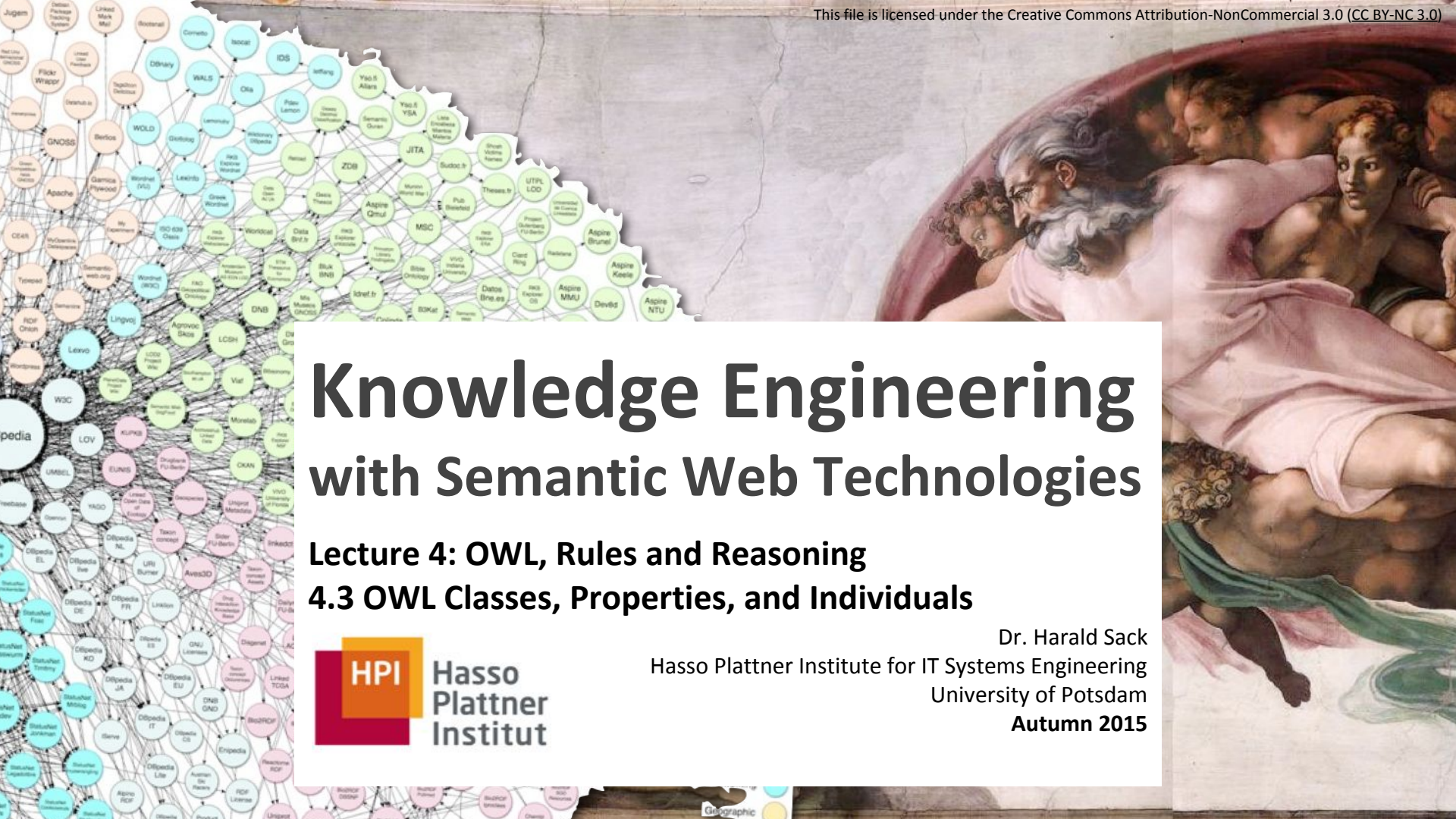
Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules and Reasoning

4.3 OWL Classes, Properties, and Individuals

HPI Hasso Plattner Institut

Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015



The image is a composite. On the left, there is a large, intricate network graph with numerous nodes (circles) of various colors (blue, green, orange, pink) and many connecting lines, representing a semantic web or knowledge graph. On the right, there is a classical painting of a man with a long white beard, wearing a pink robe, looking down at a woman who is reclining. The central white text box contains the following text:

Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules and Reasoning

4.3 OWL Classes, Properties, and Individuals

HPI Hasso Plattner Institut

Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015

The image is a composite. On the left, there is a large, intricate network graph with numerous nodes (circles) of various colors (blue, green, orange, pink) and many connecting lines, representing a semantic web or knowledge graph. On the right, there is a classical painting of a man with a long white beard, wearing a pink robe, looking down at a woman who is reclining. The central white text box contains the following text:

Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules and Reasoning

4.3 OWL Classes, Properties, and Individuals

HPI Hasso Plattner Institut

Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015

OWL – Classes, Properties and Individuals

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

- there exist two predefined classes
 - `owl:Thing` (class that contains all individuals) $\top \equiv C \sqcup \neg C$
 - `owl:Nothing` (empty class) $\perp \equiv C \sqcap \neg C$
- Definition of a class
 - `:Book a owl:Class .`

- Definition of individuals via class membership

- `:NineteenEightyfour a :Book .`

`Book(NineteenEightyfour)`

- Individuals can also be defined without class membership
as named individuals

- `:HaraldSack a owl:NamedIndividual .`

- there exist two property variants:
 - object properties
 - datatype properties
- **Object properties** have classes as range
 - `:author` a `owl:ObjectProperty` .
- Domain and Range of object properties
 - `:author` a `owl:ObjectProperty` ;
 `rdfs:domain` `:Book` ;
 `rdfs:range` `:Writer` .

$\exists \text{author} . \top \sqsubseteq \text{Book}$

$\top \sqsubseteq \forall \text{author} . \text{Writer}$

OWL Datatype Properties

- **Datatype properties** have datatypes as range
 - `:publicationYear` a `owl:DatatypeProperty` .
- Domain and Range of datatype properties
 - `:publicationYear` a `owl:DatatypeProperty` ;
`rdfs:domain` `:Book` ; $\exists \text{publicationYear}.\top \sqsubseteq \text{Book}$
`rdfs:range` `xsd:integer` . $\top \sqsubseteq \forall \text{publicationYear}.\text{Integer}$

OWL Properties and Individuals

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

OWL Properties and Individuals

- 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 .
```

OWL Class Hierarchies

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

Novel $\sqsubseteq$ Book
Book $\sqsubseteq$ Work

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

OWL Class Hierarchies and Disjunctiveness

```
: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 .
```

Novel $\sqsubseteq$ Book
Poet $\sqsubseteq$ Writer
Book $\sqcap$ Writer $\sqsubseteq \perp$

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

OWL Class Hierarchies and Disjunctiveness

- 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

```
: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

```
: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 Class Hierarchies and Disjunctiveness

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

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



04 OWL Complex Classes

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules, and Reasoning