

## 3.5 OWL: WEB Ontology Language (1)

### 3.5.1 OWL-Syntax (Teil 1)

**A) Namensräume / RDF-Tag:** Die OWL-Syntax basiert auf XML, XML-Schema, RDF und RDFS. Daher sind die zugehörigen Namensräume am Anfang des Quelltextes anzugeben. Sie werden in einem RDF-Tag angegeben, das den OWL-Kode umschließt:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  .....>
  <owl:Ontology rdf:about="" />
  ..... OWL - KODE .....
</rdf:RDF>
```

## 3.5.1 OWL-Syntax

(2)

### B) Das Ontology-Element:

Hier wird zu Beginn der OWL – Quelltext mit allgemeinen Angaben versehen:

- **rdf:about** : Angabe von "" oder der URI-Name einer Resource, die durch die vorliegende Ontologie beschrieben werden soll.
- **rdfs:comment** : Angabe eines Kommentars.
- **owl:priorVersion** : Versionsverwaltung.
- **owl:imports** : Andere OWL-Ontologien können als Teil der vorliegenden Ontologie importiert werden. Der Import ist transitiv.
- **rdfs:label** : Der vorliegenden Ontologie kann eine natürlich sprachliche Bezeichnung zugeordnet werden.

BSP: 

```
<owl:Ontology rdf:about="" />
  <rdfs:comment>Beispiel in der DBW Vorlesung
</rdfs:comment>
  <rdfs:label>Verkehrsverbund Ontologie
</rdfs:label>
</owl:Ontology>
```

## 3.5.1 OWL-Syntax

(3)

### C) Klassen, Datentyp – Eigenschaften (DatatypeProperty)

#### Instanzen:

Wir gehen von der Beschreibung einer Ontologie **O** als 4-er Tupel aus: **O** = (**C**, **R**, **HC**, **A0**) . (vgl. Kap. 3.3)

- Die Begriffe **A** aus der Begriffsmenge **C** werden in OWL wie in einer OOP als **Klassen** dargestellt (`owl:Class`).
- Merkmale **w** eines Begriffs **A**, die einfache Attribute von **A** sind, werden als **Datentyp – Eigenschaft** (`owl:DatatypeProperty`) dargestellt. Wie eine RDFS – Property hat eine OWL – DatatypeProperty einen Definitionsbereich (domain) und einen Wertebereich (range). Die domain ist die Klasse **A**, der range ist der XML-Schema-Datentyp von **w**. **w** soll z.B. vom Typ `int` sein:

```
<owl:Class rdf:ID="A"/>
<owl:DatatypeProperty rdf:ID="w">
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#A"/>
</owl:DatatypeProperty>
```

## 3.5.1 OWL-Syntax

(4)

**BSP.1:** Die OWL-Klasse Bahnhof. Sie enthält folgende Attribute: bhfnr (Bahnhofsnummer; int), bname (Bahnhofsname; string), vvbtz (Tarifzone im Verkehrsverbund; string). Jedes Attribut wird durch eine OWL-DatatypeProperty abgebildet:

```
<owl:Class rdf:ID="Bahnhof" />
<owl:DatatypeProperty rdf:ID="bhfnr">
  <rdfs:domain rdf:resource="#Bahnhof" />
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#int" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="bname">
  <rdfs:domain rdf:resource="#Bahnhof" />
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="vvbtz">
  <rdfs:domain rdf:resource="#Bahnhof" />
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>
```

## 3.5.1 OWL-Syntax

(5)

**BSP.2:** Eine Instanz mit ID="Bahnhof\_3" der OWL-Klasse Bahnhof:

```
<Bahnhof rdf:ID="Bahnhof_3">
  <bhfnr
    rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
    3
  </bhfnr>
  <bname
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Bonn_HBF
  </bname>
  <vvbtz
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    vrs_60
  </vvbtz>
</Bahnhof>
```

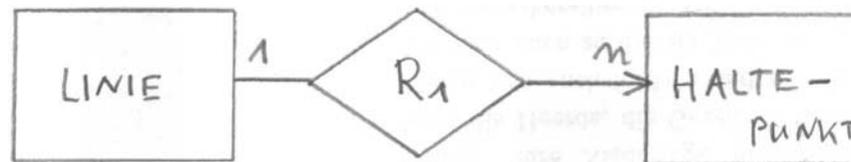
## 3.5.1 OWL-Syntax

(6)

### D) Relationen (ObjectProperty):

**Relationstypen**  $r$ , die der Relationenmenge  $R$  angehören, werden in OWL durch eine ObjectProperty (`owl:ObjectProperty`) abgebildet. Gilt:  $r \subseteq A \times B$ , so wird  $A$  als **domain** und  $B$  als **range** abgebildet.

**BSP.3:** Die Relation  $R_1$ ="haelt\_in", die zwischen den Entitätenmengen LINIE und HALTEPUNKT besteht und in folgendem ERD gegeben ist, wird als ObjectProperty der OWL-Klasse Linie dargestellt:



```
<owl:ObjectProperty rdf:ID="haelt_in">  
  <rdfs:domain rdf:resource="#Linie"/>  
  <rdfs:range rdf:resource="#Haltepunkt"/>  
</owl:ObjectProperty>
```

## 3.5.1 OWL-Syntax

(7)

Eine Instanz (ein Relationentupel)  $r(a,b)$  eines Relationentyps  $r$  wird innerhalb des OWL-Kodes der Instanz  $a$  eingefügt. Hierbei ist  $a$  Instanz der Domain – Klasse  $A$  der Relation  $r$ .

**BSP.4:** Der Relationstupel  $r(a,b)=\text{haelt\_in}(\text{Linie\_2},\text{Haltepunkt\_5})$  wird innerhalb des OWL-Kodes der Instanz  $\text{Linie\_2}$  eingefügt:

```
<Linie rdf:ID="Linie_2">
  <haelt_in rdf:resource="#Haltepunkt_5"/>
  <ltyp
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >IC</ltyp>
  <lnr
    rdf:datatype=" http://www.w3.org/2001/XMLSchema#int"
    >2</lnr>
  <lbez
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Rheingold</lbez>
</Linie>
```

## 3.5.1 OWL-Syntax

(8)

### **E) Klassenhierarchie (subClassOf):**

Um die Klassenhierarchie **HC** einer Ontologie **O = (C, R, HC, A0)** zu definieren, kann in OWL mit der RDFS-Angabe **rdfs:resource** eine Klasse B als Unterklasse einer Klasse A definiert werden:

```
<owl:Class rdf:ID="B">  
  <rdfs:subClassOf rdf:resource="#A" />  
  ...  
</owl:Class>
```

**Anm.a):** Ist B eine Unterklasse von A, dann gilt wie bei einer OOP die Vererbungseigenschaft: Ist b Instanz von B, dann ist b auch Instanz von A.

**b)** Weiterhin gilt die **Transitivität** der Unterklassenbeziehung: Ist C Unterklasse von B und ist B Unterklasse von A, dann ist auch C Unterklasse von A.

## 3.5.1 OWL-Syntax

(9)

**BSP.5:** Die Klasse **Gemuese** als Unterklasse der Klasse **Gericht** wird definiert durch:

```
<owl:Class rdf:ID=„Gemuese“>
  <rdfs:subClassOf rdf:resource="#Gericht"/>
</owl:Class>
```

Die Klasse **Gemuese** hat als Attribut die benötigte Zeit zum Garen (**garZeit**). Die Klasse **Gericht** hat als Attribut nur die Bezeichnung des Gerichts (**gBez**). Die OWL-Definition des Attributs **garZeit** lautet:

```
<owl:DatatypeProperty rdf:ID="garZeit">
  <rdfs:domain rdf:resource="#Gemuese"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

Alle Instanzen der Unterklasse **Gemuese** haben sowohl **garZeit**- als auch **gBez**-Werte.

## 3.5.1 OWL-Syntax

(10)

**Kardinalitäten:** Hat man in einem ERD eine Relation  $r$  zwischen einer Entitätenmenge  $A$  (domain) und  $B$  (range) mit einem Kardinalitätenpaar  $(a:b)$ , dann stellt sich die Frage, wie eine solche Bedingung in OWL modelliert werden kann. Kardinalitäten können in OWL als Einschränkungen der Domain-Klasse  $A$  der Relation formuliert werden. Diese Einschränkungen (Restriction) werden in einer Unterklassenklausel eingebunden:

```
<rdfs:subClassOf>  
  <owl:Restriction>  
    <owl:onProperty rdf:resource="#r"/>  
    <OWL-Restriktionsangabe .../>  
  </owl:Restriction>  
</rdfs:subClassOf>
```

OWL kennt folgende **Restriktionsangaben**:

- **allValuesFrom** (Wertebereichs ist Subklasse von B)
- **someValuesFrom** (mindestens eine Entität aus B)
- **cardinality** (genau  $b$  Entitäten aus B)
- **minCardinality** (mindestens  $b$  Entitäten aus B)
- **maxCardinality** (höchstens  $b$  Entitäten aus B)