

Idee für eine Software  
Piratenpartei Schweiz

# **KnowNet, eine Ontologie und Suchmaschine für Dokumente und Statements**

Lukas Zurschmiede  
Bachelor of Science in Information technology

Lommis, 21. Juni 2012

Lukas Zurschmiede  
Pirateparty Switzerland

# Inhaltsverzeichnis

<b>1</b>	<b>Ontologie für KnowNet</b>	<b>1</b>
1.1	Klassen	1
1.1.1	Klasse: Word	2
1.1.2	Klasse: Translation	2
1.1.3	Klasse: Synonym	2
1.1.4	Klasse: Sentence	3
1.1.5	Klasse: Document	3
1.1.6	Klasse: Paper	4
1.1.7	Klasse: Statement	4
1.1.8	Klasse: Journal	5
1.1.9	Klasse: Person	5
1.1.10	Klasse: Member	5
1.1.11	Klasse: Employee	6
1.1.12	Klasse: Blogger	6
1.1.13	Klasse: Writer	7
1.1.14	Klasse: Organization	7
1.1.15	Klasse: Association	7
1.2	Relationen	8
1.2.1	Relation: refers-to	9
1.2.2	Relation: has-sentence	9
1.2.3	Relation: sentence-of	10
1.2.4	Relation: has-word	10
1.2.5	Relation: word-of	10
1.2.6	Relation: has-writer	10
1.2.7	Relation: writer-of	11
1.2.8	Relation: has-subregion	11
1.2.9	Relation: subregion-of	11
1.2.10	Relation: has-employee	11
1.2.11	Relation: employee-of	12
1.2.12	Relation: has-member	12
1.2.13	Relation: member-of	12
1.2.14	Relation: has-document	12
1.2.15	Relation: document-of	13
1.2.16	Relation: is-about	13
1.2.17	Relation: mentioned-in	13

<b>Abkürzungsverzeichnis</b>	<b>D</b>
<b>Abbildungsverzeichnis</b>	<b>D</b>
<b>Codeverzeichnis</b>	<b>D</b>
<b>Literaturverzeichnis</b>	<b>F</b>

# 1 Ontologie für KnowNet

## 1.1 Klassen

Folgende Klassen und Spezialisierungen werden für die Ontologie gebraucht. In den nachfolgenden Kapiteln werden diese noch genauer beschrieben. Die Verlinkungen zwischen den einzelnen Klassen, also die ObjectProperties, werden im Kapitel ?? auf Seite ?? beschrieben. In den jeweiligen Detailbeschreibungen der Klassen wird jeweils darauf eingegangen und darauf verwiesen. Die Datenfelder, also die DataProperties, werden in den Detailbeschreibungen der Klassen beschrieben.

- Thing
  - Word
  - Sentence
  - Translation
  - Synonym
  - Document
    - \* Paper
    - \* Statement
    - \* Journal
  - Person
    - \* Employee
      - Blogger
      - Writer
    - \* Member
  - Organization
  - Association

### 1.1.1 Klasse: Word

Die Klasse *Word* wird verwendet, um ein einzelnes Wort, welches innerhalb eines Dokumentes gefunden werden soll, zu definieren. Einzelne Wörter können mit der Relation *wordcombo* (siehe ?? Seite ??) zu Wortkombinationen zusammengesetzt werden. Durch das Datenfeld *language* wird die Sprache des jeweiligen Wortes in iso639-1 (zwei Zeichen) definiert. Um Übersetzungen miteinander zu verknüpfen, wird die Klasse *Translation* (siehe Klasse: Translation Seite 2) verwendet. Durch die Relation *wordtype* wird die Art des Wortes definiert, also ob es sich um ein Nomen, ein Adjektiv oder ein Verb handelt.

#### Code

---

```
1: <owl:Class rdf:ID="#Word">
2: </owl:Class>
```

---

### 1.1.2 Klasse: Translation

Die Klasse *Translation* ist eine Vereinigung aller Wörter verschiedener Sprachen mit der selben Bedeutung. Dies geschieht durch das Konstrukt *owl:oneOf* und der Angabe aller Individuen innerhalb dieses Bereiches. Durch dieses Konstrukt können neue Wörter einfach und schnell eingepflegt und übersetzt werden und bei einer Abfrage kann jeweils eines aus der Liste gesucht werden.

#### Code

---

```
1: <owl:Class rdf:ID="#Translation">
2:     <owl:oneOf rdf:parseType="Collection">
3:         <Word rdf:about="#Word_Instance_IDI" />
4:         <Word rdf:about="#Word_Instance_IDI" />
5:         ...
6:     </owl:oneOf>
7: </owl:Class>
```

---

### 1.1.3 Klasse: Synonym

Die Klasse *Synonym* dient zur Verkettung von Wörtern mit der gleichen Bedeutung. Dies geschieht durch das Konstrukt *owl:oneOf* und der Angabe aller Individuen innerhalb

dieses Bereiches. Durch dieses Konstrukt können neue Wörter einfach und schnell eingepflegt und als Synonyme definiert werden, sowie bei einer Abfrage in der Liste gesucht werden.

## Code

---

```
1: <owl:Class rdf:ID="#Synonym">
2:     <owl:oneOf rdf:parseType="Collection">
3:         <Word rdf:about="#Word_Instance_IDI" />
4:         <Word rdf:about="#Word_Instance_IDI" />
5:         ...
6:     </owl:oneOf>
7: </owl:Class>
```

---

### 1.1.4 Klasse: Sentence

Ein Satz, welcher durch die Klasse *Sentence* definiert wird, ist eine Vereinigung von Wörtern zu einer Gruppe von solchen. Die Reihenfolge der Wörter ist hierbei egal, es geht nur um das Zusammenspiel und den Zusammenhang einzelner Wörter. Dieses Konstrukt wird bei der Instanz im Property *has-word* (siehe Relation: *has-word* Seite 10) durch eine anonyme Klasse mit `owl:unionOf` definiert werden.

## Code

---

```
1: <owl:Class rdf:ID="#Sentence">
2:     <rdfs:subClassOf>
3:         <owl:Restriction>
4:             <owl:onProperty rdf:resource="#has-word" />
5:             <owl:someValuesFrom ref:resource="#Word" />
6:         </owl:Restriction>
7:     </rdfs:subClassOf>
8: </owl:Class>
```

---

### 1.1.5 Klasse: Document

Die Klasse *Document* ist, ähnlich wie *Sentence*, eine Kombination von verschiedenen Sätzen. Die Reihenfolge dieser spielt keine Rolle, es geht rein um den Logischen Zusammenhang der einzelnen Sätze und somit der verschiedenen Wörter.

*Anmerkung:* Eventuell macht es hier Sinn, ein Dokument noch durch Kapitel zu strukturieren. Dies kann durch die Relation "ein Dokument aus Dokumenten" gemacht werden.

#### Code

---

```
1: <owl:Class rdf:ID="#Document">
2:   <rdfs:subClassOf>
3:     <owl:Restriction>
4:       <owl:onProperty rdf:resource="#has-sentence"
5:         />
6:       <owl:someValuesFrom ref:resource="#Sentence"
7:         />
8:     </owl:Restriction>
9:   </rdfs:subClassOf>
10: </owl:Class>
```

---

#### 1.1.6 Klasse: Paper

Die Klasse *Paper* definiert ein Dokument, welches im Zusammenhang zu einem Verein, also der Klasse *Association* steht. Durch diese Unterklasse kann am ende besser nach Dokumenten gesucht werden, welche zum Beispiel von einer bestimmten politischen Partei, oder einer Sektion von dieser, verfasst wurden.

#### Code

---

```
1: <owl:Class rdf:ID="#Paper">
2:   <rdfs:subClassOf rdf:resource="#Document" />
3: </owl:Class>
```

---

#### 1.1.7 Klasse: Statement

Die Klasse *Statement* wird verwendet, um ein Statement einer Person zu definieren. Bei einem Statement muss bei der Suche unterschieden werden, ob dieses von einer Person aus der Klasse *Association* stammt, oder von einer beliebigen anderen Person. Ein Statement von einer beliebigen Person kann genutzt werden, um sich ein Bild von Aussen zu machen, während eines von einer Person aus *Association* genutzt werden kann, um sich ein Bild von innen zu machen.

Durch das Property *from-text* kann ein Statement einem beliebigen anderen *Document* zugeordnet werden.

#### Code

---

```
1: <owl:Class rdf:ID="#Statement">
2:   <rdfs:subClassOf rdf:resource="#Document" />
3: </owl:Class>
```

---

### 1.1.8 Klasse: Journal

Ein *Journal* definiert einen Text, welcher von einem *Writer* oder *Blogger* geschrieben und Veröffentlicht worden ist. Er stellt nicht eine Meinung der *Association* dar, sondern diejenige eines Aussenstehenden. Ist der Author ebenfalls Mitglied der Klasse *Member*, kann der Text als Persönliche Meinung und somit indirekt als Parteimeinung gedeutet werden.

#### Code

---

```
1: <owl:Class rdf:ID="#Journal">
2:   <rdfs:subClassOf rdf:resource="#Document" />
3: </owl:Class>
```

---

### 1.1.9 Klasse: Person

Eine *Person* wird verwendet, um einem *Dokument* einen Autoren zuzuordnen.

#### Code

---

```
1: <owl:Class rdf:ID="#Person">
2: </owl:Class>
```

---

### 1.1.10 Klasse: Member

Ein *Member* ist eine *Person*, welche Mitglied einer *Association* ist. Dies zeichnet eine Person aus, ein *Document* - inkl. aller Unterklassen - als öffentliche Meinung zu schreiben und publizieren.



## Code

---

```
1: <owl:Class rdf:ID="#Member">
2:   <rdfs:subClassOf rdf:resource="#Person" />
3: </owl:Class>
```

---

### 1.1.11 Klasse: Employee

Die Klasse *Employee* wird verwendet, um einer *Person* einem Beruf/Anstellung zuzuweisen. Es können also nur *Person* Instanzen als *Employee* agieren.

## Code

---

```
1: <owl:Class rdf:ID="#Employee">
2:   <rdfs:subClassOf rdf:resource="#Person" />
3:   <rdfs:subClassOf>
4:     <owl:Restriction>
5:       <owl:onProperty rdf:resource="#writer-of" />
6:       <owl:someValuesFrom ref:resource="#Journal" /
7:     </owl:Restriction>
8:   </rdfs:subClassOf>
9: </owl:Class>
```

---

### 1.1.12 Klasse: Blogger

Ein *Blogger* ist ein *Employee*, welcher einen Blog schreibt oder Artikel für einen Blog verfasst. Die Klasse *Blogger* dient lediglich zur Kategorisierung einer Person. Ein *Blogger* ist eine Person, welche einen oder mehrere *Journal* verfasst hat.

## Code

---

```
1: <owl:Class rdf:ID="#Blogger">
2:   <rdfs:subClassOf rdf:resource="#Employee" />
3: </owl:Class>
```

---

### 1.1.13 Klasse: Writer

Ein *Writer* ist ein *Employee*, welcher Artikel für eine Zeitung schreib. Ein *Writer* steht also immer in Verbindung mit der Klasse *Organization*. Die Klasse *Writer* dient lediglich zur Kategorisierung einer Person. Ein *Writer* ist eine Person, welche einen oder mehrere *Journal* verfasst hat.

Ein *Writer* kann bei mehreren *Organization*'s angestellt sein. Diese Zuweisung bedarf der besonderen Beachtung bei bestimmten Abfragen und gestaltung von Meinungsbildern über oder von eine Person.

#### Code

---

```
1: <owl:Class rdf:ID="#Writer">
2:   <rdfs:subClassOf rdf:resource="#Employee" />
3:   <rdfs:subClassOf>
4:     <owl:Restriction>
5:       <owl:onProperty rdf:resource="#employee-of" />
6:       <owl:someValuesFrom ref:resource="#Newspaper" />
7:     </owl:Restriction>
8:   </rdfs:subClassOf>
9: </owl:Class>
```

---

### 1.1.14 Klasse: Organization

Die Klasse *Organization* definiert den Arbeitgeber von einem *Writer* respektive auch von jeder anderen beliebigen *Person*. Durch diese Zuweisung können Aussagen und Meinungen, welche eine Zeitung von der Partei hat, eruiert werden. Eine *Organization* kann mehrere Angestellte haben, welche nicht zwingendermassen nur bei der einen Organisation angestellt sind.

#### Code

### 1.1.15 Klasse: Association

Eine *Association* definiert eine Politische oder andere Partei, über welche man sich durch diese Ontologie ein Bild verschaffen können soll.

---

```

1: <owl:Class rdf:ID="#Organization">
2:     <rdfs:subClassOf>
3:         <owl:Restriction>
4:             <owl:onProperty rdf:resource="#has-employee"
5:                 />
6:             <owl:someValuesFrom ref:resource="#Employee"
7:                 />
8:         </owl:Restriction>
9:     </rdfs:subClassOf>
10: </owl:Class>

```

---

```

1: <owl:Class rdf:ID="#Association">
2:     <rdfs:subClassOf>
3:         <owl:Restriction>
4:             <owl:onProperty rdf:resource="#has-member" />
5:             <owl:someValuesFrom ref:resource="#Member" />
6:         </owl:Restriction>
7:     </rdfs:subClassOf>
8: </owl:Class>

```

---

## Code

## 1.2 Relationen

Die Relationen beschreiben die Abhängigkeiten wie auch die Verknüpfungen unter den Klassen. Relationen, welche klassenspezifisch sind, also direkt gebraucht werden um eine Klasse beschreiben, sind als anonyme Subklassen bereits in den Klassen selber definiert worden. Hier werden nur diejenigen Relationen beschrieben, welche verwendet werden zur direkten dynamischen und individuellen Verknüpfung verschiedener Instanzen.

### Mögliche Eigenschaften

Relationen können folgende Charakteristiken aufweisen:

**Funktional** Eine funktionale Relation  $P$  impliziert:

wenn  $P(u, v)$  und  $P(u, w)$  dann  $v == w$

**Invers Funktional** Eine inverse funktionale Relation  $P$  impliziert:

wenn  $P(v, u)$  und  $P(w, u)$  dann  $v == w$

**Transitiv** Wenn eine Relation  $P$  *transitiv* definiert ist für  $u, v, w$ :

wenn  $P(u, v)$  und  $P(v, w)$  impliziert  $P(u, w)$

**Symmetrisch** Wenn eine Relation  $P$  symmetrisch definiert ist:

wenn  $P(u, v)$  dann  $P(v, u)$

**InverseOf** Wenn bei der Relation  $P1$  eine Inverse Relation  $P2$  definiert ist:

wenn  $P1(u, v)$  dann  $P2(v, u)$

**Wichtig:** Die Eigenschaften *Reflexiv*, *Irreflexiv* und *Asymmetrisch* aus der OWL-2 Definition werden nicht verwendet.

### 1.2.1 Relation: refers-to

Die Relation *refers-to* beschreibt die Beziehungen zwischen den einzelnen Dokumenten. Wird zum Beispiel in einem Blog, was ja als *Document* interpretiert ist, auf einen Artikel aus einer Zeitung verwiesen, so kann dies durch diese Relation definiert werden. Die Relation ist sowohl *symmetrisch* als auch *transitiv* und das inverse von sich selbst.

#### Eigenschaften

- Transitiv
- Symmetrisch
- inverseOf *refers-to*

#### Code

---

```
1: <owl:ObjectProperty rdf:ID="#refers-to">
2:     <rdf:type rdf:resource="#owl:TransitiveProperty" />
3:     <rdf:type rdf:resource="#owl:SymmetricProperty" />
4:     <owl:inverseOf rdf:resource="#refers-to" />
5:     <rdfs:domain rdf:resource="#Document" />
6:     <rdfs:range rdf:resource="#Document" />
7: </owl:ObjectProperty>
```

---

### 1.2.2 Relation: has-sentence

Die Relation *has-sentence* ...

---

```
1: <owl:ObjectProperty rdf:ID="#has-sentence">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

## Code

### 1.2.3 Relation: sentence-of

Die Relation *sentence-of* ...

## Code

---

```
1: <owl:ObjectProperty rdf:ID="#sentence-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

### 1.2.4 Relation: has-word

Die Relation *has-word* ...

## Code

---

```
1: <owl:ObjectProperty rdf:ID="#has-word">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

### 1.2.5 Relation: word-of

Die Relation *word-of* ...

## Code

### 1.2.6 Relation: has-writer

Die Relation *has-writer* ...

---

```
1: <owl:ObjectProperty rdf:ID="#word-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

### Code

---

```
1: <owl:ObjectProperty rdf:ID="#has-writer">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

## 1.2.7 Relation: writer-of

Die Relation *writer-of* ...

### Code

---

```
1: <owl:ObjectProperty rdf:ID="#writer-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

## 1.2.8 Relation: has-subregion

Die Relation *has-subregion* ...

### Code

## 1.2.9 Relation: subregion-of

Die Relation *subregion-of* ...

### Code

## 1.2.10 Relation: has-employee

Die Relation *has-employee* ...

---

```
1: <owl:ObjectProperty rdf:ID="#has-subregion">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#subregion-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

### Code

---

```
1: <owl:ObjectProperty rdf:ID="#has-employee">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

### 1.2.11 Relation: employee-of

Die Relation *employee-of* ...

### Code

### 1.2.12 Relation: has-member

Die Relation *has-member* ...

### Code

### 1.2.13 Relation: member-of

Die Relation *member-of* ...

### Code

### 1.2.14 Relation: has-document

Die Relation *has-document* ...

---

```
1: <owl:ObjectProperty rdf:ID="#employee-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#has-member">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

## Code

### 1.2.15 Relation: document-of

Die Relation *document-of* ...

## Code

### 1.2.16 Relation: is-about

Die Relation *is-about* ...

## Code

### 1.2.17 Relation: mentioned-in

Die Relation *mentioned-in* ...

## Code



---

```
1: <owl:ObjectProperty rdf:ID="#member-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#has-document">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#document-of">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#is-about">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

---

```
1: <owl:ObjectProperty rdf:ID="#mentioned-in">
2:     <rdfs:domain rdf:resource="#" />
3:     <rdfs:range rdf:resource="#" />
4: </owl:ObjectProperty>
```

---

Class	Beschreibung	SubClass of	Details
<b>Word</b>	Beschreibt ein einzelnes Wort in einer Sprache, es kann auch aus mehreren wörtern zusammengesetzt sein; Anhand einer ID kann ein Link zu einer anderen Sprache oder einemmSynonym hergestellt werden;	Thing	Klasse: Word auf Seite 2
<b>Translation</b>	Gruppirt die gleichen Wörter aus verschiedenen Sprachen	Thing	Klasse: Translation auf Seite 2
<b>Synonym</b>	Gruppirt unterschiedliche Wörter der gleichen Sprache mit der gleichen Bedeutung	Thing	Klasse: Synonym auf Seite 2
<b>Sentence</b>	Ein Satz besteht aus verschiedenen Wörtern	Thing	Klasse: Sentence auf Seite 3
<b>Document</b>	Ein Dokument ist eine Sammlung von Sätzen inkl. einigen weiteren Attributen	Thing	Klasse: Document auf Seite 3
<b>Paper</b>	Ein Positionspapier oder eine andere Dokumentation einer Partei	Document	Klasse: Paper auf Seite 4
<b>Statement</b>	Eine Aussage, welche ein Mitglied einer Partei gemacht hat und welche so vertreten wird	Document	Klasse: Statement auf Seite 4
<b>Journal</b>	Ein Zeitungsbericht	Document	Klasse: Journal auf Seite 5
<b>Person</b>	Eine Person im Allgemeinen	Thing	Klasse: Person auf Seite 5
<b>Employee</b>	Ein Angestellter einer Organisation	Person	Klasse: Employee auf Seite 6
<b>Blogger</b>	Jemand der Blogs verfasst und dort seine Meinung offen Kund gibt	Employee	Klasse: Blogger auf Seite 6
<b>Writer</b>	Ein Angestellter eines Medienhauses welcher nicht immer seine eigene Meinung vertritt	Employee	Klasse: Writer auf Seite 7
<b>Member</b>	ein Angestellter oder Mitglied einer Partei, B welcher die Meinung der Partei vertritt	Person	Klasse: Member auf Seite 5
<b>Organization</b>	Eine Organisation/Medienhaus welches Zeitungen etc. herstellt, Arbeitgeber eines Journalisten	Thing	Klasse: Organization auf Seite 7
<b>Association</b>	Eine (politische) Partei, welche Mitglieder und	Thing	Klasse: Association auf Seite 7

Relation	Beschreibung	Domain	Range	Details
<b>refers-to</b>	Beziehung zwischen einzelnen Dokumenten	<i>Document</i>	<i>Document</i>	Relation: refers-to auf Seite 9
<b>has-sentence</b>	Gibt an, aus welchen Sätzen ein Dokument besteht	<i>Document</i>	<i>Sentence</i>	Relation: has-sentence auf Seite 9
<b>sentence-of</b>	Gibt an, in welchem Dokument dieser Satz vorhanden ist	<i>Sentence</i>	<i>Document</i>	Relation: sentence-of auf Seite 10
<b>has-word</b>	Gibt an, aus welchen Wörtern ein Satz besteht	<i>Sentence</i>	<i>Word</i>	Relation: has-word auf Seite 10
<b>word-of</b>	gibt an, in welchem Satz ein Wort vorhanden ist	<i>Word</i>	<i>Sentence</i>	Relation: word-of auf Seite 10
<b>has-writer</b>	Definiert einen oder mehrere Autoren eines Dokuments	<i>Document</i>	<i>Person</i>	Relation: has-writer auf Seite 10
<b>writer-of</b>	Definiert die Dokumente, an welchen eine Person mitgearbeitet hat	<i>Person</i>	<i>Document</i>	Relation: writer-of auf Seite 11
<b>has-subregion</b>	Definiert die Sub-Associations der aktuellen	<i>Association</i>	<i>Association</i>	Relation: has-subregion auf Seite 11
<b>subregion-of</b>	Definiert, dass diese Association eine Sub-Association der angegebenen ist	<i>Association</i>	<i>Association</i>	Relation: subregion-of auf Seite 11
<b>has-employee</b> <i>Association</i>	Gibt alle Employees der Organization an <i>Employee</i>	<i>Organization</i>  Relation: has-employee auf Seite 11		
<b>employee-of</b> <i>Association</i>	Gibt an, bei welchen Organizations der Employee angestellt ist Relation: employee-of auf Seite 12	<i>Employee</i>	<i>Organization</i>	
<b>has-member</b>	Definiert alle Mitglieder einer Association	<i>Association</i>	<i>Member</i>	Relation: has-member auf Seite 12
<b>member-of</b>	Definiert in welcher Association der Member Mitglied ist	<i>Member</i>	<i>Association</i>	Relation: member-of auf Seite 12
<b>has-document</b> <i>Organization</i>	gibt an, welche Association oder Organization ein dokument verfasst hat <i>Document</i>	<i>Association</i>  Relation: has-document auf Seite 12		
<b>document-of</b> <i>Organization</i>	gibt an, zu welcher Association oder Organization ein Dokument gehört Relation: document-of auf Seite 12	<i>Document</i>	<i>Association</i>	

# Abbildungsverzeichnis

## Codeblock-Verzeichnis

# Literaturverzeichnis

- [1] W3C, *OWL Web Ontology Language Reference*, 10. Februar 2004, <http://www.w3.org/TR/owl-ref/>
- [2] Natalya F. Noy Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, 2002, [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf)