



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

## ***Diplomarbeit***

# ***Semantic Web: Konzept einer OWL-Ontologie für einen Chatbot***

vorgelegt von

***Andre Eisenmenger***

am 15.04.2008

1. Prüfer: Prof. Dr. Franziskus Geeb

2. Prüfer: Prof. Dr. Ulrike Spree

Fakultät Design, Medien und Information  
*Studiendepartment Information*

## **Abstract**

Die vorliegende Diplomarbeit beschäftigt sich mit der Frage, inwieweit Ontologien dazu geeignet sind, Frage-Antwort-Systeme (so genannte Chatbots), zu unterstützen. Hierzu wird ein bestehendes Online-Lexikon in verschiedenen Ansätzen in einer Ontologie dargestellt und untersucht, wie dieses Lexikonwissen mit Hilfe von Ontologien von einem Chatbot genutzt werden kann.

### **Schlagworte:**

Web 2.0, Chatbot, OWL, Ontologien, Wissenspräsentation, Wissensmodelle

# Inhaltsverzeichnis

<b>Abstract .....</b>	<b>ii</b>
<b>Abbildungsverzeichnis.....</b>	<b>v</b>
<b>Abkürzungsverzeichnis .....</b>	<b>vii</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Inhaltlicher Aufbau .....	2
1.2 Zielsetzung.....	2
<b>2 Wege zum Semantic Web (Grundlagen) .....</b>	<b>4</b>
2.1 Semiotik und Semantik.....	4
2.2 Begriffe.....	5
2.3 Benennungen, Bezeichnungen.....	6
2.4 Terminologische Kontrolle .....	7
2.5 Wissenspräsentation mit Ontologien und Taxonomien .....	8
2.5.1 Taxonomie .....	8
2.5.2 Thesaurus .....	9
2.5.3 Topic Map.....	10
2.5.4 Ontologien.....	11
<b>3 Technische Umsetzung des Semantic Web.....</b>	<b>14</b>
3.1 Struktur und Syntax durch XML und RDF.....	14
3.1.1 XML.....	14
3.1.2 Namensräume.....	16
3.1.3 XML und das Semantic Web .....	16
3.2 RDF .....	17
3.2.1 RDF-Schema.....	19
3.3 Web Ontology Language (OWL) .....	21
3.3.1 Möglichkeiten von OWL .....	22
3.3.2 Lite, DL und Full .....	22
3.4 Ontologierstellung mit Protégé.....	23
<b>4 Internetauftritt „lookedup4You“ .....</b>	<b>24</b>
4.1 Was ist ein „Chatbot“ .....	24
4.1.1 AIML .....	25
4.1.2 Chatbot „chat4you“ .....	26
4.1.3 Suche.....	26
<b>5 Konzept einer OWL-Ontologie .....</b>	<b>29</b>
5.1 Strukturelle Darstellung des Lexikons.....	29
5.1.1 Erstellung der Klassenhierarchie.....	30
5.1.2 Beziehungen.....	32
5.1.3 Verwendung von Stichwort und Synonym .....	34
5.1.4 Umsetzung in AIML .....	36
5.1.5 Nutzen des strukturellen Ansatzes .....	38
5.2 Inhaltliche Darstellung der Lexikonartikel .....	39

5.2.1	Erstellung der Klassenhierarchie für die inhaltlichen Darstellung .....	40
5.2.2	Instanzen .....	42
5.2.3	Beziehungen.....	42
5.2.4	Inhalt in AIML.....	43
5.2.5	Nutzen des inhaltlichen Ansatzes.....	45
5.3	Verknüpfung von Inhalt und Struktur.....	45
5.3.1	Aufbau der Ontologie.....	45
5.3.2	Nutzen des vermischten Ontologieansatzes .....	48
5.4	Weitere Ansätze und Probleme.....	48
<b>6</b>	<b>Zusammenfassung und Ergebnisbetrachtung .....</b>	<b>51</b>
	<b>Literatur-/Quellenverzeichnis.....</b>	<b>53</b>
	<b>Anhang A : Ontologie der Lexikonstruktur .....</b>	<b>A-1</b>
	<b>Anhang B : Ontologie des Inhalts.....</b>	<b>B-1</b>
	<b>Anhang C : Ontologie von Struktur und Inhalt.....</b>	<b>C-1</b>
	<b>Eidesstattliche Versicherung.....</b>	<b>viii</b>

## Abbildungsverzeichnis

Abbildung 1: Semantic Layer Cake.....	1
Abbildung 2: Semiotische Dreieck nach Aristoteles.....	5
Abbildung 3: Semiotisches Dreieck nach Wüster.....	7
Abbildung 4: Semantische Treppe.....	8
Abbildung 5: Beispiel einer Taxonomie.....	9
Abbildung 6: Beispiel eines Thesaurus.....	10
Abbildung 7: Beispiel einer Topic Map.....	11
Abbildung 8: Einfache Darstellung eines XML-Elements.....	15
Abbildung 9: Beispiel einer Verschachtelung in XML.....	15
Abbildung 10: Verwendung von Namensräumen in XML.....	16
Abbildung 11: einfacher RDF Graph.....	18
Abbildung 12: RDF Graph in XML.....	18
Abbildung 13: RDF-Graph mit Literal.....	18
Abbildung 14: leerer Knoten als RDF Graph.....	19
Abbildung 15: hierarchische Struktur in XML.....	20
Abbildung 16: einfache taxonomische Hierarchie in OWL.....	21
Abbildung 17: Webaufttritt „lookeup4you“.....	24
Abbildung 18: einfaches Frage-Antwort Paar in A.I.M.L. ....	25
Abbildung 19: Beispiel für Wildcard.....	26
Abbildung 20: Verweis in A.I.M.L.....	26
Abbildung 21: Zugehörigkeit eines Lemma zu einer Kategorie.....	27
Abbildung 22: Eingabemaske für Synonyme und Stichworte.....	27
Abbildung 23: Klassenhierarchie der Lexikonstruktur.....	30
Abbildung 24: einfache Hierarchie in OWL/XML.....	31
Abbildung 25: Disjunktheit zwischen zwei Klassen.....	31
Abbildung 26: Restriktion der Klasse Synonym.....	32
Abbildung 27: Properties der Klasse Lemma.....	32
Abbildung 28: Lemma <i>Geldkarte</i> in Protege.....	33
Abbildung 29: Range und Domain des Property <i>Stichwort_verweist_auf</i> .....	34
Abbildung 30: Definition des Property <i>Stichwort_verweist_auf</i> in XML Code.....	34
Abbildung 31: Definition des Property <i>ist_verwandt_mit</i> in XML Code .....	34
Abbildung 32: Beziehungen des Lemma <i>Geldkarte</i> in XML Code....	36
Abbildung 33: Reaktion auf Synonymeingabe.....	36
Abbildung 34: Reaktion auf Stichworteingabe.....	37
Abbildung 35: Frage-Antwortpaar für das Lemma <i>Geldkarte</i> .....	37
Abbildung 36: Einsatz von Wildcards.....	38
Abbildung 37: Klassenhierarchie der Lexikonkategorie <i>Reisen und Fahren</i> .....	41
Abbildung 38: Individual <i>Au Pair</i> (Protégé).....	43

Abbildung 39: Properties der Klasse Person (Protégé).....	43
Abbildung 40: Beziehungen der Instanz <i>Au_Pair</i> .....	43
Abbildung 41: einfacher Dialog aus Triple .....	44
Abbildung 42: Zufallsantwort auf eine Frage.....	44
Abbildung 43: Triples des Individual <i>Au_Pair</i> .....	44
Abbildung 44: Klassenhierarchie in Protégé.....	46
Abbildung 45: Properties der Klasse Lemma (Protégé) .....	47
Abbildung 46: Triples des Individuals <i>Au_Pair</i> .....	48

## **Abkürzungsverzeichnis**

AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
FAS	Frage-Antwort-System
OWL	Web Ontology Language
RDF	Resource Description Framework
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

# 1 Einleitung

Das World Wide Web (kurz Web) bietet eine schier unüberschaubare Menge ständig präsenter Informationen. Die Bereitstellung von Informationen im Web ist für nahezu jeden handhabbar. Das ständige Wachsen von Informationen im Web hat jedoch einen entscheidenden Nachteil. Die Suche nach Dokumenten im Internet gleicht zur Zeit einer einzigen Stichwortwortsuche. Während es dem Menschen Möglich ist, die Information eines Dokumentes zu erfassen und mit anderen Informationen in Beziehung zu setzen, ist eine Maschine dazu nicht in der Lage. Unsere Computer haben vom Inhalt der gesuchten Dokumente keine Ahnung. Das bedeutet, dass eine Information, die zwar prinzipiell verfügbar ist, nicht oder nur schwer gefunden werden kann. Bei der Suche nach dem Begriff „Fischer“, werden Ergebnisse geliefert, die sich mit der Fischerei und mit dem gleichnamigen Politiker befassen.

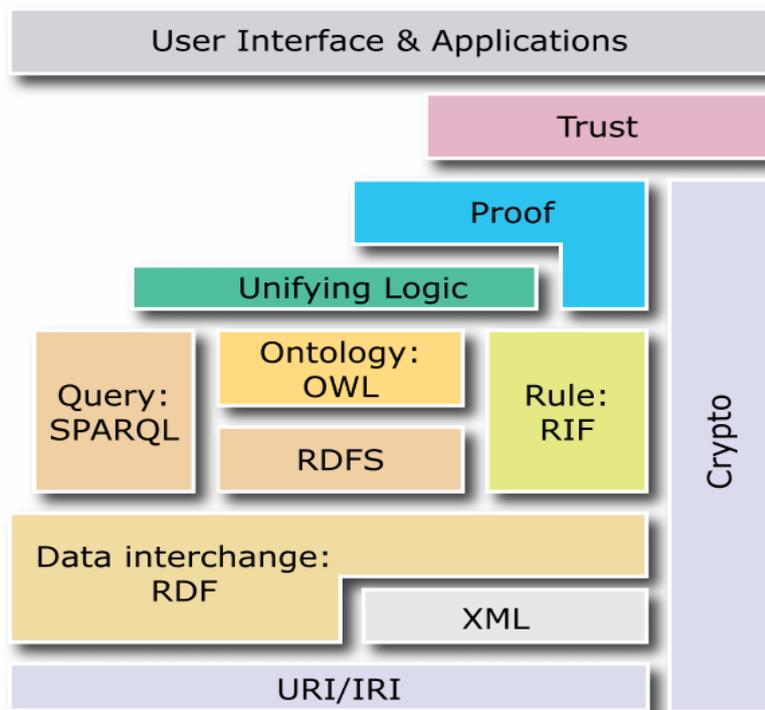


Abbildung 1: Semantic Layer Cake

Um Webinhalte für Maschinen „lesbar“ zu machen, hat das W3C<sup>1</sup> eine Roadmap veröffentlicht, in der beschrieben wird, wie das Semantic Web entstehen soll. Das Semantic Web stellt eine Erweiterung zum World Wide Web dar. Es soll das bestehende Netz nicht ersetzen, sondern erweitern. Mit dem Semantic Web sollen Informationen mit Hilfe von Metadaten verknüpft und miteinander in Beziehung gesetzt werden. Obwohl bereits mehrere Anwendungen für das Semantic Web existieren, blieb der große Erfolg jedoch bis jetzt aus. Die Web Ontology Language (OWL) soll dem Semantic Web nun zum Durchbruch verhelfen. Technisch ist OWL eine Erweiterung von RDF.

## **1.1 Inhaltlicher Aufbau**

Die vorliegende Arbeit gliedert sich in sechs Kapitel. Im Anschluss an die Einleitung werden die formalen Grundlagen des Semantic Web erläutert. Anschließend gibt es einen Überblick über die technische Umsetzung des Semantic Web. Danach werden Frage-Antwort-Systeme erklärt und deren Nutzung und Anwendungen, Stärken und Schwächen hervorgehoben. Im Folgenden wird anhand eines Beispiels aus einem bestehenden Lexikonwissen eine Ontologie erstellt. Abschließend wird ermittelt, ob eine solche Ontologie von einem Rechner für einen Chatbot nutzbar gemacht werden kann bzw., ob die beispielhaft entwickelte Ontologie als Gesprächsgrundlage für einen Chatbot dienen kann.

## **1.2 Zielsetzung**

Die Web Ontology Language (OWL) ist die neue Ontologiesprache für das Web. In dieser Arbeit soll geklärt werden, ob sich diese Sprache für die Entwicklung einer Ontologie eines Chatbot eignet. Als Grundlage für diese Ontologie dient ein bereits bestehendes, Datenbank-basierendes Lexikon, dass in Form einer Ontologie dargestellt

---

<sup>1</sup> World Wide Web Consortium

werden soll. Diese Ontologie soll für einen Chatbot nutzbar gemacht werden. Ziel ist es, ein verbessertes Gesprächsverhalten des Chatbot zu erreichen, also die Kommunikation zwischen Mensch und Maschine zu optimieren.

## **2 Wege zum Semantic Web (Grundlagen)**

Wie bereits erwähnt, geht es bei der Semantic Web Initiative des W3C um den Umgang mit Inhalten im World Wide Web. Um zu verstehen, was das Semantic Web überhaupt ist, ist es notwendig, im Vorwege einige Begriffe zu erläutern. Zunächst wird ausgeführt, was sich hinter dem Begriff Semantik verbirgt.

### **2.1 Semiotik und Semantik**

Semiotik ist die „allgemeine Lehre von den Zeichen, Zeichensystemen und Zeichenprozessen. Sie wird üblicherweise in die drei Teilbereiche Syntax, Semantik und Pragmatik untergliedert, wobei diese Bereiche durch ihre Beziehungen zwischen Zeichen, der Zeichenbedeutung und den Benutzern der Zeichen in einer bestimmten Situation definiert werden.“ (GALINSKI 2006, S. 54)

- Syntax: die Beziehung von Zeichen untereinander
- Semantik: die Beziehung von Zeichen zu deren Bedeutung
- Pragmatik: Beziehung der Zeichen zu deren Erzeugern und Interpretieren (vgl. GALINSKI 2006, S. 54)

Nach Nöth ist die Semantik neben der Syntax und der Pragmatik ein Zweig der allgemeinen Semiotik (NÖTH 2000, S. 159). Sie beschäftigt sich mit der Bedeutung von Begriffen.

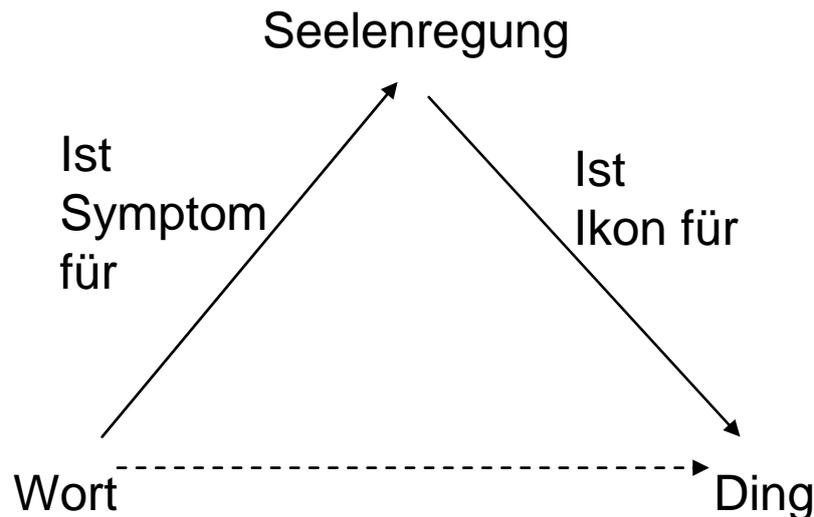


Abbildung 2: Semiotische Dreieck nach Aristoteles

Bei Aristoteles ist ein Zeichen ein Symptom für eine Seelenregung, d.h. für etwas, das der Sprecher sich vorstellt. Diese Vorstellung des Sprechers ist dann ein Ikon für ein Ding. Dies sind für ihn die primären Zeichenrelationen (Schenkel in der Figur). Davon abgeleitet ist die sekundäre Zeichenrelationen (gestrichelte Basis in der Figur). Spätere Interpretationen halten daran fest, dass zwischen dem Wort (Zeichenform) und dem Bezeichneten (Ding) im Allgemeinen keine direkte Beziehung besteht. Die Beziehung wurde irgendwann willkürlich festgelegt. Diese Beziehung bezeichnet man als Bedeutung (oder Begriff) (vgl. GALINSKI 2006, S. 55-57).

## 2.2 Begriffe

*Begriffe sind geistig-gedankliche Abbildungen von Gegenständen, Sachen, Vorgängen, Ideen usw. Ein Begriff enthält das Typische einer Gruppe gleichartiger Gegenstände, gleichartiger Sachen, gleichartiger Vorgänge, gleichartiger Ideen usw. Die Begriffe entstehen durch die Abstraktionsleistung menschlichen Denkens*

(GAUS 2005, S. 57).

*Für jeden Gegenstand gibt fast immer fast unerschöpflich viele und durchaus sinnvolle Aussagen oder Eigenschaften, und wir sind immer gezwungen, aus dieser Fülle von denkbaren Aussagen eine Auswahl dessen zu treffen, was für uns wichtig und wesentlich ist*

(FUGMANN 1992, S. 39).

Das Problem besteht darin, dass jeder Person einen bestimmten Gegenstand anders definiert. Für den Einen hat der „Mensch“ die Eigenschaften „belebt“ und „46 Chromosomen“. Diese Eigenschaften können von einem Anderen jedoch anders bezeichnet werden. Anstatt „Mensch“ könnte man auch „homo sapiens“ sagen oder statt „belebt“ „lebendig“ und statt Chromosomen „Träger von mendelnden Erbanlagen“. Das heißt, man kann dasselbe meinen, es aber unterschiedlich ausdrücken. Diesen Zusammenhang zwischen Gegenstand und Eigenschaften kann man als Begriff bezeichnen (vgl. KIEL 2002, S. 27).

### **2.3 Benennungen, Bezeichnungen**

Da Begriffe nicht direkt zwischen Personen ausgetauscht werden können, wird jedem Begriff eine Benennung zugeordnet. Eine Benennung ist also ein Name, eine Bezeichnung, eine Kennzeichnung eines Begriffs. Anstatt Benennung oder Bezeichnung sagt man in der Dokumentation oft auch einfach Wörter (vgl. GAUS 2005, S. 57).

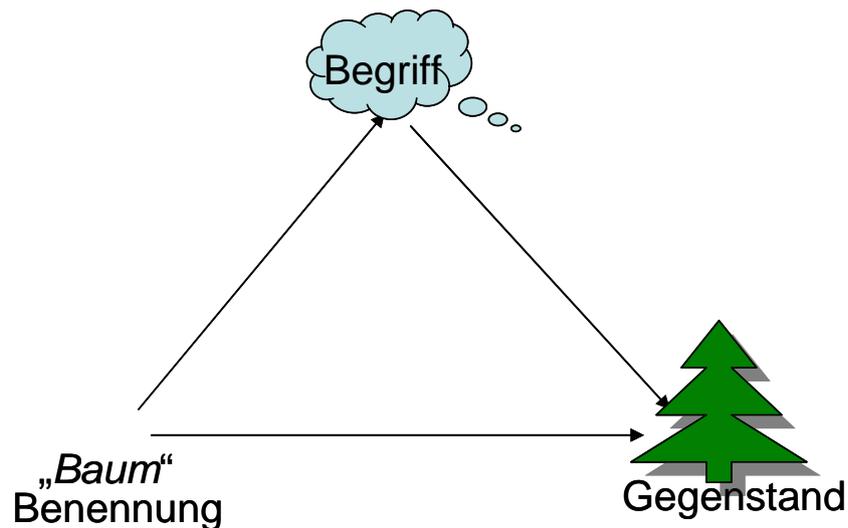


Abbildung 3: Semiotisches Dreieck nach Wüster

## 2.4 Terminologische Kontrolle

*Um die Eindeutigkeit der Beziehungen zwischen Begriffen und Benennungen herzustellen, ist es unerlässlich, dass terminologische Kontrolle ausgeübt wird. Hierunter sollen alle Regeln und Aktivitäten zusammengefasst werden, die dazu geeignet sind, Mehrdeutigkeiten auf der Bezeichnungsebene auszuschalten*

(WERSIG 1985, S. 47).

Mit Hilfe kontrollierter Vokabulare lässt sich eine solche terminologische Kontrolle ausüben. Durch sie sollen vor allem das Homonymproblem vermieden werden, also Ausdrücke, die mehrere unterschiedliche Begriffe repräsentieren können. Desweiteren vermeiden diese Deskriptoren das Auftreten von Synonymen, indem jedem Begriff genau ein, oder ein bevorzugter Deskriptor zugewiesen wird. Für komplexere Wissenspräsentationen greift das kontrollierte Vokabular zu kurz, da hier keine Strukturen gebildet werden können (vgl. KIENREICH 2006, S. 362).

## 2.5 Wissenspräsentation mit Ontologien und Taxonomien

Im folgendem Abschnitt wird gezeigt, wie Ontologien entstehen und auf welchen semantischen Modellen sie basieren. Gleichzeitig wird anhand von Beispielen der Begriff der terminologischen Kontrolle weiter verdeutlicht. Abbildung 4 zeigt die Evolution semantischer Modelle als semantische Treppe. Von den hier gezeigten semantischen Modellen sollen im folgenden Taxonomie, Thesaurus, Topic Map und Ontologie erläutert werden.

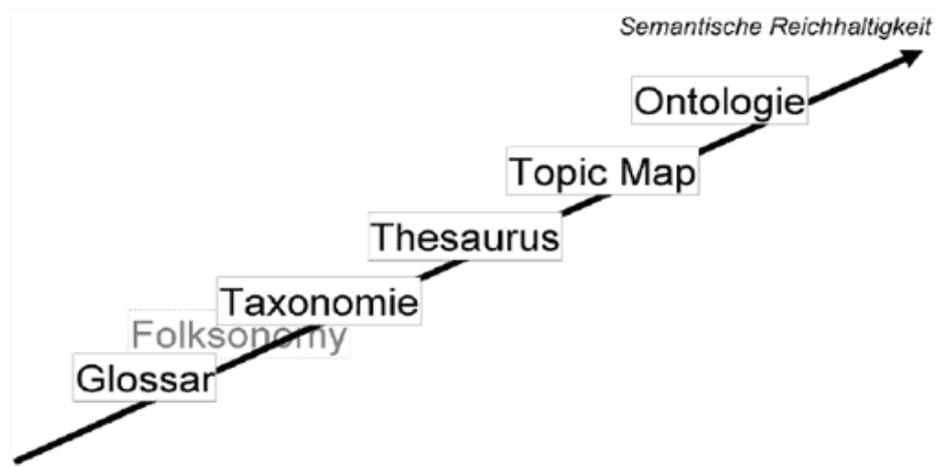


Abbildung 4: Semantische Treppe

### 2.5.1 Taxonomie

Eine Taxonomie ist eine hierarchische Klassifikation (üblicherweise in einer Baumstruktur) von Objekten. Sie wird in der Informationstechnologie dazu benutzt, um Informationen zu klassifizieren, die mit diesen Objekten in Zusammenhang stehen (vgl. DACONTA 2003, S. 180). Ihren Ursprung hat die Taxonomie unter anderem in der Naturwissenschaft. In der Biologie dient sie z.B. der Darstellung hierarchischer Beziehungen von Pflanzen und Tieren. Hier wird also versucht, Begriffe eines Themengebietes zu definieren und miteinander in Beziehung zu setzen, um dieses möglichst präzise und umfassend zu beschreiben.

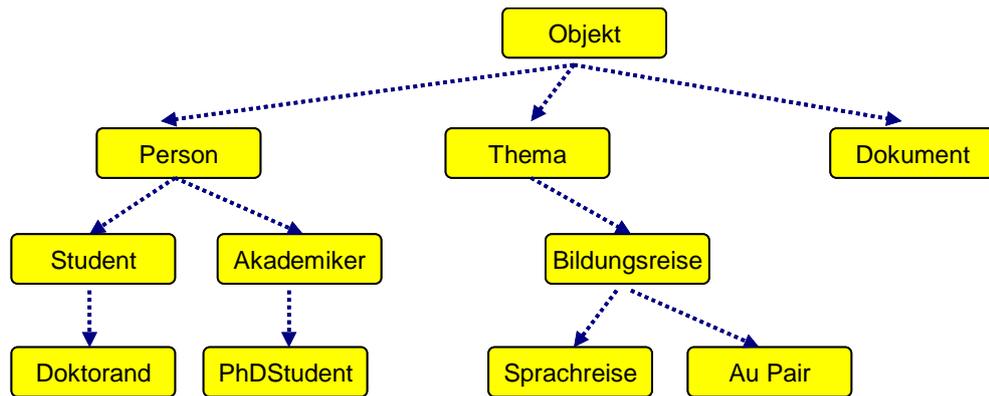


Abbildung 5: Beispiel einer Taxonomie

Ein Nachteil der Taxonomie ist, dass durch die lineare Struktur (Abbildung 5), synonyme Begriffe nicht ausgedrückt werden können. Des Weiteren können Redundanzen (Polyseme) entstehen, etwa dann, wenn eine Bezeichnung unterschiedliche Bedeutungen aufweist.

## 2.5.2 Thesaurus

Die genannten Nachteile einer Taxonomie lassen sich mit Hilfe von Relationen einschränken. Hierzu werden Beziehungen zwischen Begriffen dargestellt, die es erlauben, die feste Hierarchie der Taxonomie zu durchbrechen. Durch die Relationen „ist Synonym für“ und „ist ähnlich zu“ lässt sich die Struktur stark erweitern.

Ein solches Konstrukt bezeichnet man als Thesaurus. In einem Thesaurus werden Begriffe und Bezeichnungen eindeutig aufeinander bezogen, indem Synonyme möglichst vollständig erfasst und Polyseme besonders gekennzeichnet werden (vgl. BURKART 2004, S. 141).

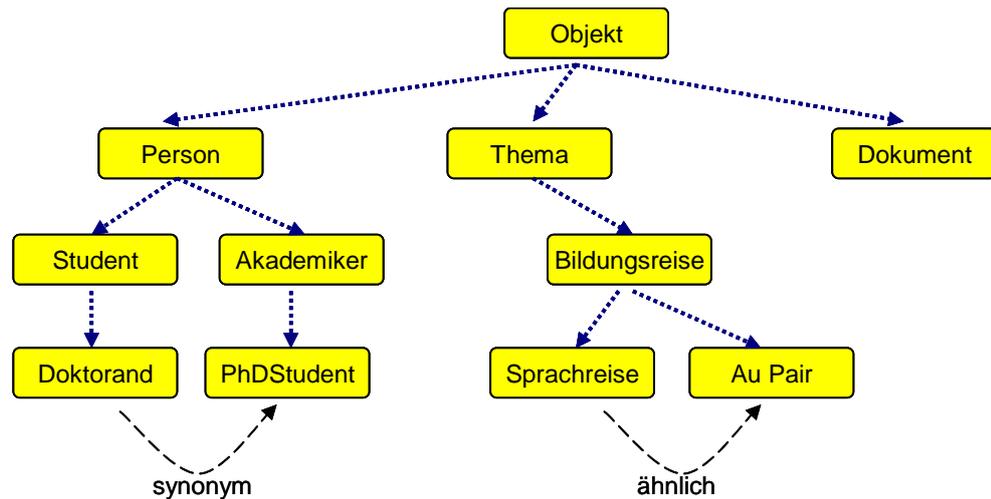


Abbildung 6: Beispiel eines Thesaurus

Erweiterung man unser Beispiel mit den genannten Relationen, lässt sich nun Aussagen, dass „Doktorand“ ein Synonym für „PhDStudent“ ist und „Sprachreise“ ein verwandter Begriff zu „Au Pair“.

### 2.5.3 Topic Map

Bei Topic Maps handelt es sich um einen ISO-Standard, der unter anderem von Steve Pepper entwickelt wurde. Mit Hilfe von Topic Maps können Wissensstrukturen und Assoziationen in Informationsbeständen auf standardisierte Weise beschrieben werden.

Die Idee von Topic Maps ist, „ähnlich wie bei Lexika, Glossaren oder Indexen, externe Dokumente zu referenzieren und die Grundbausteine dieses Wissens, die Topics (etwa, worüber ein Artikel in diesem Lexikon handelt), miteinander in Beziehung zu bringen und so eine Wissensbasis aufzubauen. (WIDHALM 2002, S. 6)

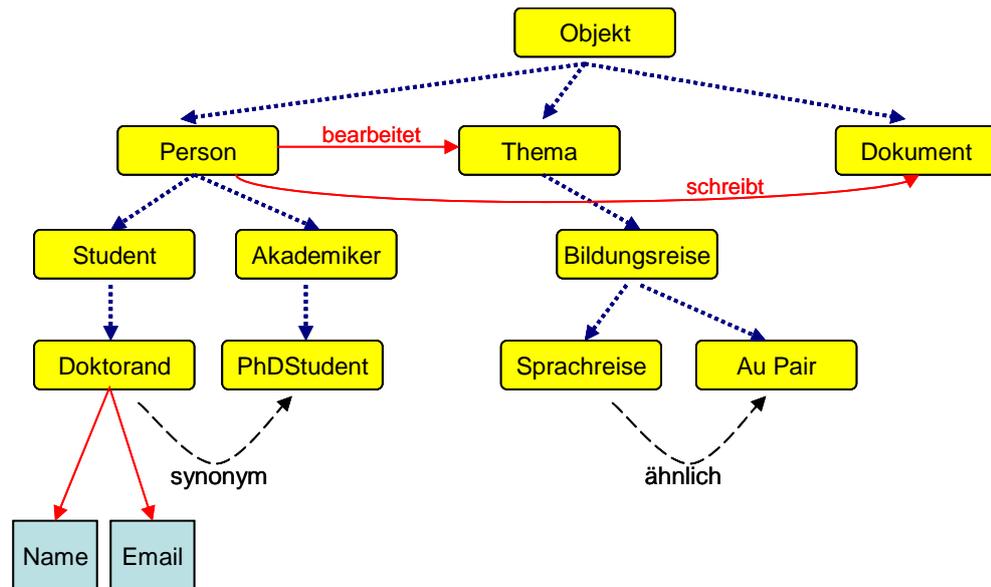


Abbildung 7: Beispiel einer Topic Map

Topic Maps gehen in ihrer Ausdrucksstärke über den Thesaurus hinaus. Eine Topic Map besteht aus:

- Topics
- Assoziationen
- Occurrences
- Scopes

Topics beschreiben einen Aussagegegenstände, die mit Hilfe von Assoziationen miteinander verknüpft werden. Occurrences verbinden die Topics mit einer Informationsressource. Ein Topic kann mehrere Namen haben, z.B. für verschiedene Sprachen. Mit Scopes lässt sich der Kontext bestimmen, in dem ein Name gültig ist. Ein Topic, das für den *Strand* steht kann z.B. drei Namen haben: *Strand*, *playa* und *beach*. Ist nun der Scope *Deutsch* gültig, so ist nur der Name *Strand* gültig (vgl. WIKIPEDIA 2008 a).

## 2.5.4 Ontologien

Historisch stammt der Begriff Ontologie aus der Philosophie und ist die Bezeichnung für die Lehre vom Wesen des Seins. Im Bezug auf das Semantic Web greift diese Definition jedoch zu kurz. Tom Gruber definiert eine Ontologie als eine explizite Spezifikation einer Konzep-

tualisierung (vgl. GRUBER 1992). An dieser Stelle sei noch einmal an das „semiotische Dreieck“ erinnert, wonach sich Wörter und Symbole auf Konzepte beziehen, die durch reale Gegenstände oder Dinge repräsentiert werden.

Mit Ontologien kann das Wissen einer Domäne formal repräsentiert werden. Sie beschreiben Konzepte und ihre Beziehungen innerhalb einer Wissensdomäne und unterstützen Maschinen Inhalte zu interpretieren, anstatt sie einfach nur darzustellen (vgl. BLUMBAUER, S. 12). In einer Ontologie werden Klassen gebildet. Diese Klassen werden dann in einer Taxonomie angeordnet, es werden Eigenschaften definiert, Einschränkungen und erlaubte Werte festgelegt, sowie Instanzen generiert. Alle Instanzen und Unterklassen eines solchen Konzepts erben die Eigenschaften ihrer Oberklasse. Der Unterschied zur Taxonomie ist der, dass die Ontologie ein Netzwerk von Informationen mit logischen Relationen darstellt, während die Taxonomie eine rein hierarchische Untergliederung bildet, die zu Redundanzen führen kann.

Ein Vorteil von Ontologien ist die Wiederverwendbarkeit von Wissen. Ist das grundlegende Konzept einer Wissensdomäne erst einmal beschrieben, kann es in anderen Ontologien wieder verwendet werden und muss nicht erneut beschrieben werden.

Bestandteile einer Ontologie:

- Begriffe → (engl. „concepts“) Die Beschreibung gemeinsamer Eigenschaften wird als Begriff definiert. Begriffe werden auch als Klassen bezeichnet. Diese können in der Klassenstruktur mit Über- und Unterklassen angeordnet werden (z.B. die Begriffe „Stadt“ oder „Land“)
- Instanzen → Repräsentieren Objekte in der Ontologie. Sie stellen das zur Verfügung stehende Wissen dar. Diese werden anhand vorher definierter Begriffe erzeugt und auch als Individuals bezeichnet (z.B. München als Instanz des Begriffs Stadt).

- Relationen → Instanzen gleichen Typs müssen an verschiedene Gegebenheiten angepasst werden. Dazu werden Relationen verwendet, die beschreiben, welche Beziehungen zwischen den Instanzen bestehen (z.B. Stadt München *liegt in* Land Deutschland). Relationen werden auch als Eigenschaften bezeichnet.
- Axiome → Axiome sind Aussagen innerhalb der Ontologie, die immer wahr sind. Diese werden normalerweise dazu verwendet, Wissen zu repräsentieren, das nicht aus den Begriffen abgeleitet werden kann (z.B. zwischen Amerika und Europa existiert keine Zugverbindung) (vgl. WIKIPEDIA 2008 b).

### 3 Technische Umsetzung des Semantic Web

Informationen in der Weise zur Verfügung zu stellen, dass diese in maschinenlesbarer Form zur Verfügung stehen, dafür steht das Semantic Web. Um dieses Ziel umzusetzen sind einheitliche Standards nötig. Für die Entwicklung und Akzeptanz dieser Standards setzt sich das World Wide Web Consortium ein. Die wichtigsten Spezifikations-sprachen sollen hier kurz vorgestellt werden (vgl. HITZLER 2008, S.11).

#### 3.1 Struktur und Syntax durch XML und RDF

##### 3.1.1 XML

XML bildet die syntaktische Grundlage für das Semantic Web, auf der auch die Syntax der im späteren Verlauf beschriebenen Beschreibungssprachen beruht. XML steht für „Extensible Markup Language“ und ist eine Empfehlung des W3C (vgl. HITZLER 2008, S. 17).

Markup-Sprachen dienen dazu Textdokumente mit zusätzlichen Informationen anzureichern. Man spricht auch vom *auszeichnen* oder *annotieren*. Solche Annotationen werden auch als Metadaten bezeichnet. Die Markup-Sprache HTML<sup>2</sup> nutzt sogenannte Tags, um Teile eines Text Dokuments auszuzeichnen. Diese Tags werden von einem Web-Browser interpretiert und entsprechend am Bildschirm ausgegeben. Die auszuzeichnenden Textteile werden von einem Start-Tag und einem End-Tag eingeschlossen. Je nach Tag Art werden Textteile mit dem Tag `<i>` kursiv oder mit `<b>` fett ausgegeben. Das folgende Beispiel zeigt für den Satz „Der Autor des Buches heißt Hans Gustav“ die Auszeichnung durch HTML.

`<i>Der Autor des Buches</i> heißt <b>Hans Gustav</b>.`

Ein Web-Browser stellt diesen Satz wie folgt dar:

---

<sup>2</sup> Hypertext Markup Language

*Der Autor des Buches* heißt **Hans Gustav**.

Während HTML eine festgelegte Menge von Tags besitzt, hat XML diese Restriktion nicht. Zwar wurde das Konzept vollständig geklammerter Textdokumente für XML übernommen. Die Erweiterbarkeit wird hier dadurch erreicht, dass Tagnamen frei gebildet werden können. Bei der Strukturierung eines Dokuments mit XML ist man also frei in der Wahl der so genannten Elementnamen.

Elemente bestehen aus einem Start-Tag und einem End-Tag, die jeweils von spitzen Klammern eingeschlossen sind. Die Daten zwischen diesen Klammern bezeichnet man als Inhalt eines Elements. Im Beispiel ist *Autor* der Elementname und *Hans Gustav* der Inhalt.

```
<Autor>Hans Gustav</Autor>
```

Abbildung 8: Einfache Darstellung eines XML-Elements

Grundsätzlich hat jedes XML-Dokument ein Wurzelement. Alle weiteren Elemente befinden sich in verschachtelter Form innerhalb dieses Wurzelements. Dieses Prinzip verdeutlicht folgendes Beispiel:

```
<Buchautoren>  
  <Autor>  
    ... <Name>Hans Gustav</Name>  
  </Autor>  
</Buchautoren>
```

Abbildung 9: Beispiel einer Verschachtelung in XML

Zusätzlich können Elemente mit Attributen versehen werden. Attribute weisen den XML-Elementen Werte zu. So kann zum Beispiel dem Tag *Name* das Attribut *Geschlecht* zugewiesen werden. Der Attributwert wird hinter dem Attributnamen nach einem Gleichheitszeichen in Anführungszeichen gesetzt, wie das folgende Beispiel zeigt.

```
<Name Geschlecht=„männlich“>Hans Gustav</Name>
```

### 3.1.2 Namensräume

Die freie Wählbarkeit der Elementnamen wirft natürlich im Web die Frage nach der Eindeutigkeit auf. Mehrdeutigkeiten in den Namen würden die gemeinsame Nutzung von Dokumenten aus unterschiedlichen Quellen sehr erschweren. Die nahe liegende Lösung für das Problem ist, URIs zum optionalen Bestandteil von Tagnamen zu machen (vgl. BIRKENBIHL 2006, S. 78). Namenskonflikte entstehen durch die Mehrdeutigkeiten von Begriffen. So kann das Element *Titel* zum einen den Titel eines Buches meinen und zum anderen den Titel einer Person. Namensräume werden deshalb in das Start-Tag eines XML-Elements integriert und sind dann innerhalb dieses Elements gültig. Den Namensraum bildet ein Attribut mit der Bezeichnung `xmlns` mit der dazugehörigen URI. Optional kann hinter dem Attributnamen, durch Doppelpunkt getrennt, ein Präfix gesetzt werden, welches dann als Abkürzung für die URI dient, da diese häufig sehr lang sind. Folgendes Beispiel zeigt die Verwendung eines Namensraumes (vgl. HITZLER 2008, S. 25):

```
<Buchautoren xmlns="http://Beispiel.de/" xmlns:xyz="http://xyz.www.Beispiel.de/">
  <Titel>Semantic Web</Titel>
  <xyz:Autor>
    <xyz:Name>Hans Gustav</xyz:Name>
    <xyz:Titel>Professor</xyz:Titel>
  </xyz:Autor>
</Buchautoren>
```

Abbildung 10: Verwendung von Namensräumen in XML

### 3.1.3 XML und das Semantic Web

XML stellt durch die hierarchische Baumstruktur eine maschinenlesbare Meta-Sprache dar. Für den Menschen können die Elementnamen eine Bedeutung haben. So erzeugt der Elementname *Buch* beim Menschen sofort das geistige Abbild eines solchen. Für eine Maschine ist es jedoch unerheblich ob ein Element *Buch* oder *XYZ* heißt. Maßgeblich für den Computer ist nur die einheitliche Struktur der Elemente. Deshalb reicht XML für die Verarbeitung und Weiter-

gabe expliziten Wissens nicht aus, sondern bildet nur die Grundlage, um die Sprachen des Semantic Web, OWL und RDF, zu definieren.

### 3.2 RDF

Während es bei XML darum geht, Daten in einer einheitlichen Sprache und streng hierarchisch zu strukturieren, dient das Resource Description Framework (RDF) dazu, Daten im Web auszutauschen, ohne dass deren Informationen verloren gehen. Es geht darum, Informationen über Daten selbst anzureichern und miteinander in Beziehung zu setzen. Diese so genannten Metadaten können zum Beispiel Angaben zu Titel, Name oder Copyright einer Website enthalten. Die Beschreibung von Aussagen in RDF erfolgt meistens in Form eines Graphen, bestehend aus benannten Knoten und Kanten oder in der RDF/XML-Syntax.

*Das so genannte RDF-Modell besteht aus drei Objekttypen: Ressourcen, Eigenschaften und Werten, die man zusammen auch als Triple bezeichnet. Der Wert einer Aussage kann wiederum die Ressource einer anderen Aussage sein. Durch die Verknüpfung dieser Aussagen „ergibt sich ein gerichteter Graph aus Ressourcen und ihren Eigenschaften. Zur eindeutigen Bezeichnung von Ressourcen und Eigenschaften werden URIs<sup>3</sup> verwendet. URIs verankern RDF im Web*

(BIRKENBIHL 2006, S.80).

Die RDF-Spezifikation schlägt vor, Aussagen als Sätze in der Form Subjekt, Prädikat und Objekt (so genannte Triple) zu formulieren, wobei die Ressource das Subjekt, die Eigenschaften das Prädikat und die Werte der Eigenschaften als Objekte zu interpretieren sind (vgl. ECKSTEIN 2004, S. 239ff).

---

<sup>3</sup> Uniform Resource Identifier

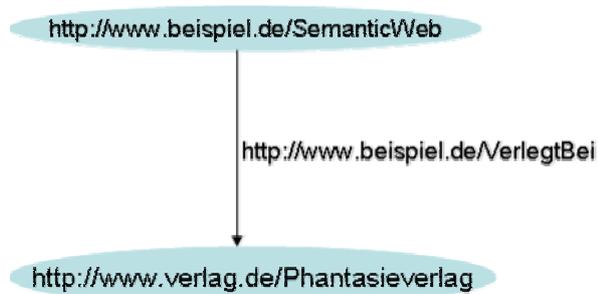


Abbildung 11: einfacher RDF Graph

Abbildung 11 zeigt einen einfachen RDF-Graphen. Er Beschreibt die Beziehung zwischen einem Buch und dem Verlag bei dem es erschienen ist. Man kann sagen, das Buch *Semantic Web* ist verlegt beim *Phantasieverlag*. Auch hier ist die Satzform mit Subjekt, Prädikat und Objekt zu erkennen und durch URIs repräsentiert. Die Darstellung der obigen Abbildung sieht in der RDF/XML-Syntax wie folgt aus (Abbildung 12):

```
<rdf:Description rdf:about="http://www.beispiel.de/SemanticWeb">
  <bs:VerlegtBei>
    <rdf:Description rdf:about="http://www.verlag.de/Phantasieverlag">
    </rdf:Description>
  </bs:VerlegtBei>
</rdf:Description>
```

Abbildung 12: RDF Graph in XML

Das Tripel wird in der RDF/XML-Syntax nach dem Subjekt gruppiert. Die deklarierten Namensräume *rdf:* und *bs:* sind beliebig. Subjekt und Objekt werden durch die Elemente des Typs *rdf:Description* beschrieben, das Attribut *rdf:about* gibt den Bezeichner des Elements an. Das Objekt wird direkt als Element *bs:VerlegtBei* angegeben.

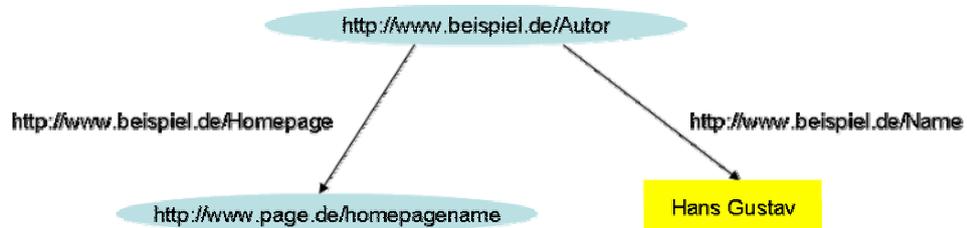


Abbildung 13: RDF-Graph mit Literal

Im nächsten Beispiel (Abbildung 13) sieht man, dass die durch die Prädikate beschriebenen Objekte unterschiedlicher Art sein können. *Hans Gustav* ist ein Literal und stellt lediglich den Wert der Eigenschaft dar. Literale werden im RDF-Graph als Rechteck dargestellt. Während „<http://www.homepage.de>“ in elliptischer Form dargestellt wird und somit eine Ressource ist. Ressourcen können auch als leere Knoten dargestellt werden (Abbildung 15).

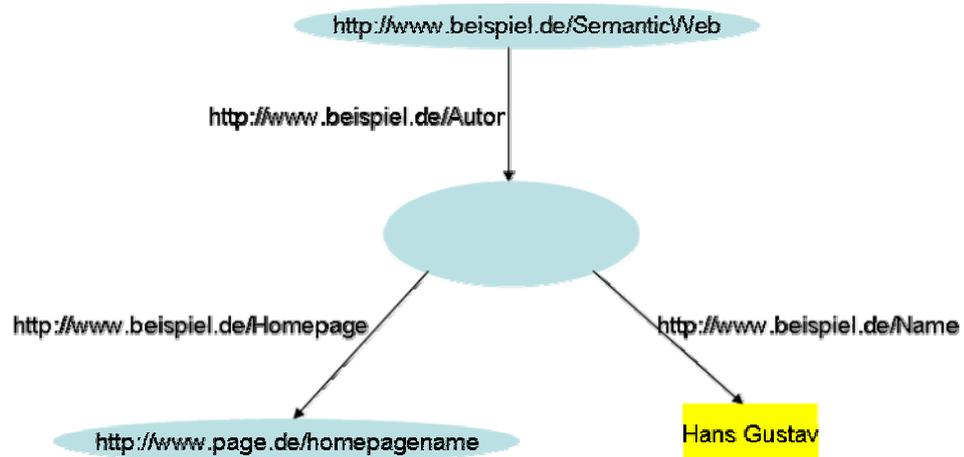


Abbildung 14: leerer Knoten als RDF Graph

Abbildung 14 zeigt einen RDF-Graph mit einem leeren Knoten. Diese leeren Knoten, auch Black Nodes genannt, dienen dazu, mehrwertige Beziehungen zu beschreiben. Das Buch *Semantic Web* könnte zum Beispiel mehrere Autoren mit Namen und Homepage haben.

### 3.2.1 RDF-Schema

Mit RDF kann man ausdrücken, dass Hans Gustav ein Autor ist. Für den Menschen ist es kein Problem daraus zu schließen, dass Hans Gustav auch ein Mensch ist. RDF fehlt die Möglichkeit dies auszudrücken. RDF lediglich dient dazu Aussagen über einzelne Ressourcen zu treffen. Um mit solchen Aussagen Ontologien erstellen zu können ist es nötig, diese Aussagen miteinander in Beziehung zu setzen, eine Hierarchie zu bilden. RDFS ist eine Erweiterung von RDF und wird dazu verwendet terminologisches Wissen in Form von Hierarchien zu spezifizieren. Dies geschieht in RDFS hauptsächlich

durch die Bildung von Ober- und Unterklassen, sowie Ober- und Unterproperties.

Der Namensraum von RDFS lautet <http://www.w3.org/2000/01/rdf-schema#> und wird üblicherweise mit `rdfs:` abgekürzt. „RDF-Schema erlaubt es, in RDF Klassen von Objekten zu definieren und Subklassen von Klassen zu definieren sowie Eigenschaften Werte- und Gültigkeitsbereiche zuzuordnen. Dies geschieht durch die Definition von Klassen wie `rdfs:Resource`, `rdfs:Class` und Prädikaten wie `rdfs:subClassOf`, `rdfs:domain`, `rdfs:range`, `rdfs:type`. In RDF-Schema sind auch Eigenschaften Ressourcen und gehören zur Klasse `rdfs:Properties`“ (BIRKENBIHL 2006, S.82).

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:bs="http://www.xyz.de/beispiele#">
  <rdfs:Class rdf:about="&bs;mensch" rdfs:label="mensch">
  </rdfs:Class>
  <rdfs:Class rdf:about="&bs;autor" rdfs:label="autor">
  <rdfs:subClassOf rdf:resource="&bs;mensch"/>
  </rdfs:Class>
  <bs:autor rdf:about="&bs;HansGustav"/>
</rdf:RDF>
```

Abbildung 15: hierarchische Struktur in XML

Obiges Beispiel sagt aus, dass die Klasse Autor eine Unterklasse der Klasse Mensch ist.

Weiters lassen sich mit Hilfe von RDF-Schema Eigenschaften definieren und einschränken. `Rdfs:domain` und `rdfs:range` legen zum Beispiel die Domäne und den Wertebereich fest, der für eine Eigenschaft gültig ist. Auf diese Weise lässt sich festlegen, dass das Subjekt einer Eigenschaft *istAutorVon* eine Person sein muss und das Objekt ein Buch.

### 3.3 Web Ontology Language (OWL)

Mit RDF-Schema lassen sich einfache Ontologien darstellen. Jedoch hat RDF-Schema nur begrenzte Ausdrucksmittel und ist für die Darstellung komplexer Zusammenhänge nicht ausreichend. Um eben solche komplexen Konstrukte darstellen zu können, wurde OWL entwickelt. OWL steht für „Web Ontology Language“ und ist eine Empfehlung des W3C aus dem Jahr 2004. OWL verfügt über drei Teilsprachen mit unterschiedlicher Ausdrucksstärke. Je nach Anforderung an die Praxis kann zwischen OWL Full, OWL DL und OWL Lite gewählt werden. Die für den Datenaustausch verwendete Syntax für OWL-Ontologien ist RDF. Wie auch schon in RDF(S) besteht eine Ontologie hauptsächlich aus Klassen und Properties. In OWL können diese jedoch in komplexere Beziehung zueinander gesetzt werden (vgl. HITZLER 2008, S. 125-126).

Die meisten Elemente in OWL betreffen Klassen, Properties und Instanzen von Klassen und Beziehungen zwischen diesen Instanzen (vgl. SMITH 2004, S. 12)

```
<owl:Class rdf:ID="Mensch" />
<owl:Class rdf:ID="Frau">
  <rdfs:subClassOf rdf:resource="#Mensch" />
</owl:Class>
<Frau rdf:ID="Marta" />
```

Abbildung 16: einfache taxonomische Hierarchie in OWL

Abbildung 16 zeigt die Deklaration der Klasse *Mensch* (*owl:Class*) und der Unterklasse *Frau* (*rdfs:subClassOf*). Außerdem ist Marta eine Instanz der Klasse *Frau*.

Der bisher rein taxonomische Ansatz von OWL wird durch Properties erweitert. Properties lassen es zu, Fakten über Klassen und Individuals anzuführen (vgl. SMITH 2004, S. 12). Die Verwendung von Properties wird in Kapitel 5 ausgeführt.

### 3.3.1 Möglichkeiten von OWL

Mit Hilfe bestimmter neuer Eigenschaften und Ressourcen lassen sich in OWL neue Klassen erzeugen. Erreicht wird dies durch:

- Aufzählung der Instanzen
- Durchschnitt mit anderen Klassen (das was in Klasse A und B ist)
- Vereinigung mit anderen Klassen (das was in Klasse A oder B ist)
- Kardinalitäten (n Instanzen aus, mehr/weniger als n Instanzen aus A)
- Komplementäroperationen (alles was nicht in A ist)

(vgl. BIRKENBIHL 2006, S. 84)

Die Einschränkung von Domänen und Wertebereichen von Eigenschaften, die letzten Kapitel erläutert wurden, lassen sich mit Hilfe von OWL weiter spezifizieren. Für Eigenschaften (p) können folgende Bedingungen gelten:

- p ist transitiv
- p ist funktional
- p is symmetrisch
- p ist invers zu
- p ist invers funktional

### 3.3.2 Lite, DL und Full

OWL-Lite ist die syntaktisch einfachste Subsprache. Mit ihr können einfache Klassenhierarchien und einfache Begrenzungen dargestellt werden. Sie bietet z.B. die Möglichkeit existierende Thesauri oder andere konzeptionell simple Hierarchien zu migrieren.

OWL-DL besitzt eine wesentlich höhere Ausdrucksstärke und basiert auf der Description Logic. Mit OWL-DL lässt sich ein „automated reasoning“ durchführen, mit dem es möglich ist, die Ontologie computergestützt nach Inkonsistenzen zu prüfen.

OWL-Full ist die Ausdrucksstärkste OWL Subsprache. Sie wird eingesetzt, wenn ein hohes Maß an Ausdrucksstärke wichtiger ist als eine computergestützte Möglichkeit der Verarbeitung.

Unterschieden wird zwischen den drei OWL Subsprachen OWL-Lite, OWL-DL und OWL-Full. OWL-Lite besitzt die geringste Ausdrucksstärke und OWL-Full die Stärkste. OWL-DL kann als Erweiterung von OWL-Lite angesehen werden und OWL-Full wiederum als Erweiterung zu OWL-DL (vgl. SMITH 2004, S. 6-7).

### **3.4 Ontologierstellung mit Protégé**

Da formale Sprachen wie RDFS oder DAML+OIL einen hohen Einarbeitungsaufwand erfordern, gibt es so genannte Ontologieeditoren, die die Modellierung einer Ontologie wesentlich erleichtern (HILGENFELD 2003, S. 5). Die bekanntesten Editoren sind:

- OntoEdit (Ontoprise GmbH Karlsruhe)
- OilEd (Manchester)
- Protégé (Stanford)

(HILGENFELD 2003, S. 6)

Protege ist einer der bekanntesten Ontologieeditoren. Mit Protege lassen sich Wissensdatenbanken anlegen. Protege ist frame-basiert, das bedeutet, dass das Domänenwissen mittels einer hierarchischen Struktur von Konzepten, Attributen von Konzepten (slots) und Instanzen (Individuals) dargestellt wird.

Neben der hohen Benutzerfreundlichkeit ist die Erweiterbarkeit von Protégé ein weiteres wichtiges Kriterium, das für den Editor spricht. Mit ihm lassen sich, dank eines Plugins auch OWL-Ontologien erstellen, die das ursprüngliche Konzept um logische Bestandteile ergänzen.

## 4 Internetauftritt „lookedup4You“

Das Online-Lexikon „lookedup4you“ hat sich zum Ziel gesetzt, Verbraucherschutzrelevante Themen für Kinder und Jugendliche aufzubereiten und zu gestalten. Ziel war die Erlangung von Konsumkompetenz die es der Zielgruppe ermöglicht, ihre Rechte und Pflichten wahrnehmen zu können.



Abbildung 17: Webauftritt „lookeup4you“

Auftraggeber des Projektes „lookedup4you“ sowie des Folgeprojektes „chat4you“ war die Verbraucherzentrale Nordrhein-Westfalen e.V. Im Zuge des Folgeprojekts „chat4you“ wurde die Homepage verbessert und um ein weiteres Feature ergänzt, einen Chatbot. Der Chatbot sollte das bestehende Online-Lexikon interaktiver und somit attraktiver gestalten. In diesem Kapitel soll allgemein die Funktionsweise von Chatbots erläutert werden und im speziellen die Verknüpfung des Bots mit dem Online-Lexikon „lookedup4you“.

### 4.1 Was ist ein „Chatbot“

Der Begriff Chatbot, auch Chatterbot, ist eine Kurzform von „chat robot“. Es handelt sich um ein Computerprogramm, das menschliche Gespräche simuliert (vgl. MÖBUS 2005, S. 73). Der Chatbot kann lediglich aus einer Texteingabe- und Ausgabemaske bestehen und durch eine personalisierte Darstellungsform ergänzt werden.

Die Kommunikation mit dem System erfolgt über natürlichsprachliche Eingaben. Der Chatbot greift auf eine angelegte Wissensbasis zu, in

der ein entsprechender Fragenbestand hinterlegt ist und gleicht diese mit der Eingabe ab. Findet eine Übereinstimmung mit der Wissensbasis statt, generiert der Chatbot eine Antwort und gibt diese aus oder führt eine andere Aktion aus, z.B. die Ausgabe eines Dokuments (vgl. BRAUN 2003, S. 21). Ein solches System bezeichnet man im Allgemeinen als Frage-Antwort-System (FAS).

Ein solches Frage-Antwort-System mit natürlichsprachlichen Eingaben (Fragen), kann zusätzlich durch eine Visualisierung ergänzt werden. Diese sogenannten Avatare können dem Abbild eines Menschen nachempfunden sein und neben den gegebenen Antworten in Form von Text auch menschliche Mimik oder Gestik simulieren, um natürlicher zu wirken (vgl. MÖBUS 2006, S. 73).

#### 4.1.1 AIML

ALICE steht für „Artificial Linguistic Internet Computer Entity“ und ist ein Chatbot für natürliche Sprache. Der Chatbot basiert auf einer XML-Spezifikation, der AIML Technologie. Die Wissensbasis eines ALICE Chatbots basiert auf Frage-Antwort Modulen, Kategorien genannt. Diese werden in der AIML-Sprache (artificial intelligence markup language) dargestellt. Die Kategorie wird als *category* Tag dargestellt. Die Frage repräsentiert das *pattern* Tag und die dazugehörige Antwort das *template* Tag.

```
<category>
| <pattern>WAS WEISST DU UEBER AUTOS</pattern>
| <template>Autos sind Fortbewegungsmittel.</template>
| </category>
```

Abbildung 18: einfaches Frage-Antwort Paar in A.I.M.L.

Die Frage wird durch die chatbot engine mit den patterns in der Wissensbasis abgeglichen. Bei einem Treffer antwortet der Chatbot mit dem dazugehörigen Template aus der Wissensbasis.

```

<category>
| <pattern>WIE GROSS *</pattern>
| <template>Grösser als ein Stecknadelkopf.</template>
</category>

```

Abbildung 19: Beispiel für Wildcard

Mit Hilfe von Wildcards (Abbildung 19) innerhalb der pattern ist es dem Chatbot möglich, auch auf Fragen zu antworten, für die kein übereinstimmender pattern vorhanden ist.

```

<category>
| <pattern>WER HAT DICH ENTWICKELT</pattern>
| <template>
| | <srai>WER HAT DICH PROGRAMMIERT</srai>
| </template>
</category>

<category>
| <pattern>WER HAT DICH PROGRAMMIERT</pattern>
| <template>Hans Gustav hat mich programmiert</template>
</category>

```

Abbildung 20: Verweis in A.I.M.L

Andere AIML Tags ermöglichen es, die Dialoge natürlicher erscheinen zu lassen. Der Tag *srai* erlaubt es beispielsweise, auf andere Kategorien zu verweisen (Abbildung 20), die Tags *set* und *get* erlauben es, Werte von Variablen zu übernehmen. Der Tag *topic* definiert ein bestimmtes Thema im Dialog und der Tag *condition* erlaubt konditionierte Antworten.

#### 4.1.2 Chatbot „chat4you“

Der Chatbot übernimmt neben der Small-Talk Funktion auch noch die Aufgabe einer Suchfunktion. Dazu verfügt er über eine Eingabemaske die mit der Datenbank des Online-Lexikons verbunden ist.

#### 4.1.3 Suche

Der Themenbereich des Online-Lexikon umfasst wie bereits erwähnt den Verbraucherschutz für Kinder und Jugendliche. Das Lexikon ist

thematisch in mehrere Kategorien unterteilt, die wiederum Unterkategorien enthalten, die dann entsprechende Artikel alphabetisch geordnet zur Verfügung stellen. Die Kategorien stehen auf der Homepage in Menüform zur Verfügung und können so durchbrowsed werden. Bei der Informationssuche hat der Nutzer ebenfalls die Möglichkeit, gesuchte Begriffe über die Eingabemaske des Chatbot zu erfragen. Er erhält dann entweder eine Antwort vom Chatbot oder einen Link zum entsprechenden Artikel.



The screenshot shows a light green interface with a header box containing the text "Lemma 'Abonnement' anzeigen:". Below this, there is a label "Kategorie (Subkategorie):" followed by a dropdown menu. The dropdown menu is open, showing the selected option "Trends + Shopping (Allgemeines)". Below the dropdown, there is a label "Lemma:" followed by a text input field containing the word "Abonnement".

Abbildung 21: Zugehörigkeit eines Lemma zu einer Kategorie

Die Artikel bestehen unter anderem aus Titel, Kurzdefinition, Langtext und Siehe-auch-Hinweise. Diese Bausteine sind in einer Datenbank abgelegt. Des Weiteren wird jeder Artikel durch ein Lemma repräsentiert und durch Synonyme ergänzt. Diese sind ebenfalls in der Datenbank abgelegt. Wird in das Eingabefeld ein Begriff eingegeben startet eine Datenbankabfrage. Hier wird zunächst nach Lemma, Synonym und Stichwort gesucht. Bringt das keinen Treffer, findet eine Volltextsuche in den Artikelbausteinen statt.



The screenshot shows a light green interface with two sections. The first section is labeled "Synonym:" and contains a text input field with the number "1:" followed by the text "Abo". The second section is labeled "Stichwort:" and contains a text input field with the number "1:" followed by the text "Abos". There is a small square button with an upward-pointing arrow between the two input fields.

Abbildung 22: Eingabemaske für Synonyme und Stichworte

Außerdem greift der Chatbot bei der Bearbeitung von Suchanfragen auf ein Set von vorformulierten Fragen und Antworten zurück. Zusätzlich zur Überarbeitung der Suchfunktion wurden deshalb verschiedene ‚Standardantworten‘ entwickelt, die jeweils in Abhängigkeit von der Art der Suchergebnisse zusammen mit diesen ausgegeben werden sollten. Je nach dem, woher die Ergebnisse der Datenbankabfrage stammten, wurde dabei zwischen „Lemma-“, „Synonym-“, „Stichwort“ und „Volltexttreffern“ unterschieden.

## 5 Konzept einer OWL-Ontologie

Das im vorigen Kapitel beschriebene Lexikon soll im folgenden Abschnitt in eine Ontologie überführt werden. Es soll untersucht werden, ob sich durch die Darstellung in einer Ontologie ein Mehrwert erreichen lässt, den man auf den Chatbot übertragen kann. Zunächst wird die Struktur des Online-Lexikons in eine Ontologie überführt. Ausgangspunkt hierfür sind die Lemmata, die mit Synonymen, Stichworten, Texten und anderem mehr, in Beziehung gesetzt werden sollen. Danach wird der Inhalt einer einzelnen Kategorie des Lexikons in einer Ontologie dargestellt und der Versuch unternommen, dadurch einen Nutzen für den Chatbot zu erreichen.

### 5.1 Strukturelle Darstellung des Lexikons

Das Lexikon besteht aus Lemmata die durch Kategorien und Unterkategorien hierarchisch gegliedert sind. Jedes Lemma kann ein oder mehrere Synonyme und Stichworte haben, die auf ein bestimmtes Lemma verweisen. Die Definition der Lemmata erfolgt mittels Kurz- und Langtexten. Außerdem kann jedes Lemma auf ein Quiz oder einen Filmbeitrag verweisen.

In der folgenden Ontologie werden die Unterkategorien weggelassen, da diese ursprünglich das Browsen erleichtern sollten und in der Ontologie die Zuordnung der Lemmata aufgrund von Redundanzen erschweren. Folgende Fragen soll die Ontologie unter anderem beantworten:

- Welche Stichworte hat ein Lemma
- Welche Synonyme hat ein Lemma
- Wie ist die Definition eines Lemma
- Zu welcher Kategorie gehört ein Lemma

Für die Erstellung einer Ontologie müssen zunächst die Wissensdomäne repräsentierende Begriffe gesammelt werden. Die Begriffe, die

für die Modellierung dieser Ontologie benötigt werden, stammen aus dem bestehenden Lexikon, da sie den zu beschreibenden Wissensbereich bereits auseichend repräsentieren. So kommen u. a. die Begriffe *Synonym*, *Stichwort*, *Lemma*, *Kategorie* zum Einsatz.

### 5.1.1 Erstellung der Klassenhierarchie

Abbildung 23 zeigt die Hierarchische Klassenstruktur der in Protege erstellten Ontologie.



Abbildung 23: Klassenhierarchie der Lexikonstruktur

Die Klasse *Lemmaverknuepfungen* hat die direkte Unterklasse *Kategorie*, die wiederum die einzelnen Kategoriebezeichnungen als Instanzen enthält. Des Weiteren hat die Klasse *Lemmaverknuepfungen* die Unterklasse *Woerter* mit den Unterklassen *Stichwort* und *Synonym*. Die Klasse *Lemma* enthält die Lemmata, ebenfalls als Instanzen, die mit den Instanzen der Klasse *Lemmaverknuepfungen* und deren Unterklassen in Beziehung gesetzt werden.

#### Transitivität

Hierbei gilt das Prinzip der Transitivität. Wenn *B* eine Unterklasse von *A* ist und *C* eine Unterklasse von *B*, dann ist *B* eine Unterklasse von *A*.

A (vgl. NOY 2005, S.10). Wenn *Woerter* eine Unterklasse von *Lemmaverknuepfungen* ist und *Stichwort* eine Unterklasse von *Woerter*, dann ist *Stichwort* eine Unterklasse von *Lemmaverknuepfungen*.

### Disjunktheit

Ist ein Wort als *Stichwort* deklariert, kann es nicht gleichzeitig ein *Synonym* sein. Deshalb müssen die Klassen *Stichwort* und *Synonym* disjunkt gesetzt werden. Abbildung 24 definiert die Klasse *Synonym* als Unterklasse von *Woerter*. Mit dem Tag *owl:disjointWith* wird erreicht, dass eine Instanz der Klasse *Stichwort* nicht gleichzeitig eine Instanz der Klasse *Synonym* sein kann.

```
<owl:Class rdf:ID="Synonym">
  <rdfs:subClassOf rdf:resource="#Woerter"/>
  <owl:disjointWith rdf:resource="#Stichwort"/>
</owl:Class>
```

Abbildung 24: einfache Hierarchie in OWL/XML

Die Klasse *Stichwort* ist ebenso als Unterklasse von *Woerter* definiert und Instanzen dürfen nicht in beiden Klassen vorkommen (s. Abbildung 25).

```
<owl:Class rdf:ID="Stichwort">
  <rdfs:subClassOf rdf:resource="#Woerter"/>
  <owl:disjointWith rdf:resource="#Synonym"/>
</owl:Class>
```

Abbildung 25: Disjunktheit zwischen zwei Klassen

### Restriktionen

Restriktionen oder Rolleneinschränkungen definieren „eine Klasse als die Menge aller Objekte, für die eine bestimmte Rolle immer einen Wert aus einer vorgegebenen Klasse annimmt“ (HITZLER 2008, S.137). In diesem Fall darf ein *Synonym* nur existieren kann, wenn es mindestens einer Instanz der Klasse *Lemma* zugehörig ist. Realisiert wird dies durch die Restriktion *owl:someValuesFrom*.

```

<owl:Class rdf:ID="Synonym">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Synonym_verweist_auf"/>
      <owl:someValuesFrom rdf:resource="#Lemma"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Abbildung 26: Restriktion der Klasse Synonym

Solche Restriktionen existieren ebenfalls für die Beziehung zwischen Stichwort und Lemma, sowie zwischen Lemma und Kategorie.

### 5.1.2 Beziehungen

Als nächstes sollen die im letzten Abschnitt erwähnten Beziehungen hergestellt werden. Beziehungen werden durch Properties auch Rollen genannt realisiert. In OWL gibt es zwei Arten von Rollen: abstrakte und konkrete. Während abstrakte Rollen Individuen mit Individuen verbinden, verknüpfen konkrete Rollen Individuen mit Datenwerten (vgl. HITZLER 2008, S.130). Die Verknüpfung der Objekte wird über die Tags *ObjectProperty* für abstrakte Rollen und *DatatypeProperty* für konkrete Rollen realisiert.

- ▼ ■ `gehört_zu_Kategorie` (someValuesFrom Kategorie)
  - ☰ Kategorie
  - `hat_Kurztext` (single string)
  - `hat_Langtext` (single string)
  - `hat_Quiz` (multiple string)
  - `hat_Video` (multiple string)
- ▼ ■ `ist_äquivalent_mit` (multiple Synonym) (someValuesFrom Synonym)
  - ☰ Synonym
- ▼ ■ `ist_verwandtschaftlich_mit` (multiple Stichwort) (someValuesFrom Stichwort)
  - ☰ Stichwort

Abbildung 27: Properties der Klasse Lemma

In Abbildung 27 sind alle Properties, dazu gehören Datatype- (grün) und Objectproperties (blau), aufgeführt. In den Klammern hinter den

einzelnen Properties ist der Wertebereich angegeben. So kann das Property *ist\_aequivalent\_mit* mehrere Synonyme haben.

Abbildung 28 zeigt zunächst die Beziehungen des Lemma *Geldkarte* in Protégé. Über die Maske können die Eigenschaften von Instanzen hinzugefügt oder gelöscht werden. Wie man sieht, hat das Lemma mehrere Synonyme und Stichworte, sowie einen Kurztext. Außerdem gehört *Geldkarte* der Kategorie *Geld und Job* an. Ein Langtext ist nicht vorhanden. Mit Hilfe der Maske kann das Lemma ebenso mit einem Video und einem Quiz verbunden werden.

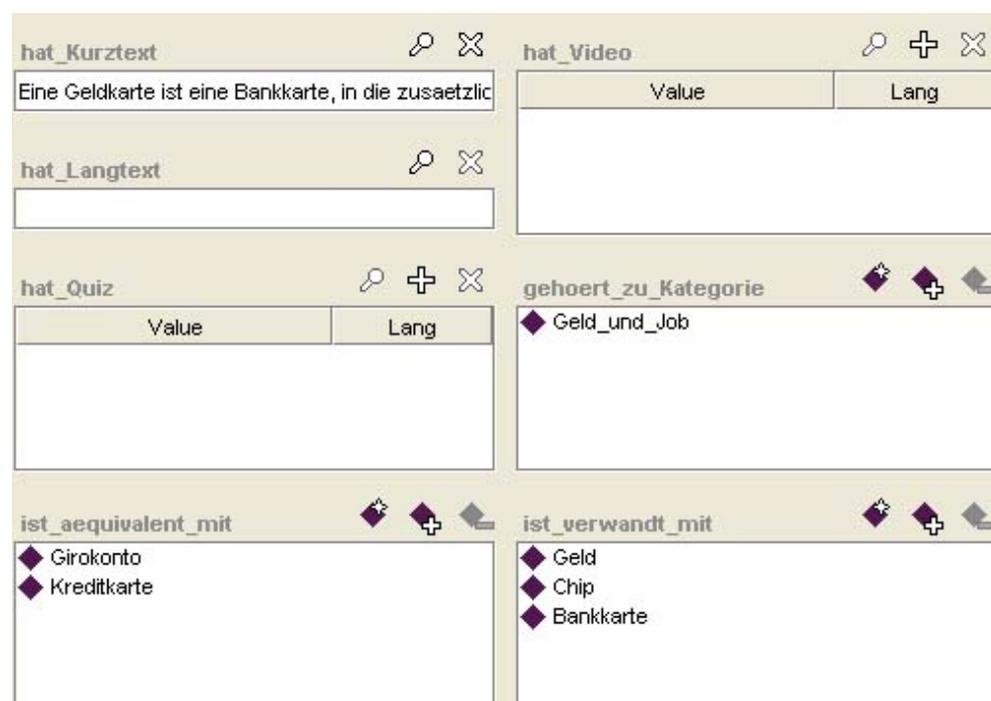


Abbildung 28: Lemma *Geldkarte* in Protege

Die nachfolgende Abbildung 29 zeigt die Definition des ObjectProperty *Stichwort\_verweist\_auf*. Über die Tags domain und range wird der Wertebereich des Property festgelegt. Das bedeutet, dass die Domäne des Property *Stichwort\_verweist\_auf* ein Stichwort sein muss und der Wertebereich ein Lemma.



Abbildung 29: Range und Domain des Property *Stichwort\_verweist\_auf*

Der XML-Code des Property *Stichwort verweist auf beinhaltet beschreibt neben Range und Domain ebenfalls eine genauere Typisierung*. Durch *FunctionalProperty* (Abbildung 30) wird sichergestellt, dass ein Stichwort nur einem Lemma zugeordnet werden kann. Außerdem wird mit dem Tag *owl:inverseOf* die Inverse Beziehung des Properties festgelegt.

```
<owl:ObjectProperty rdf:ID="Stichwort_verweist_auf">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Stichwort"/>
  <rdfs:range rdf:resource="#Lemma"/>
  <owl:inverseOf rdf:resource="#ist_verwandt_mit"/>
</owl:ObjectProperty>
```

Abbildung 30: Definition des Property *Stichwort\_verweist\_auf* in XML Code

Wenn das eine Property funktional ist, dann muss die inverse Beziehung invers funktional sein. Das bedeutet hier, dass einem Lemma mehrere Stichworte zugeordnet werden können (Abbildung 31).

```
<owl:ObjectProperty rdf:ID="ist_verwandt_mit">
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="#Stichwort"/>
  <owl:inverseOf rdf:resource="#Stichwort_verweist_auf"/>
</owl:ObjectProperty>
```

Abbildung 31: Definition des Property *ist\_verwandt\_mit* in XML Code

### 5.1.3 Verwendung von Stichwort und Synonym

Durch die Verwendung von Stichworten und Synonymen ist es möglich, dass der Chatbot auf Eingaben die entweder einem Synonym oder einem Stichwort entsprechen, unterschiedlich reagieren kann. Mit Hilfe dieser Struktur könnte der Chatbot auf die folgenden Eingaben entsprechende Antworten ausgeben:

Frage:

*Was ist eine Geldkarte (Lemma)?*

Antwort:

*Eine Geldkarte ist eine Bankkarte, in die zusätzlich ein Mikro-Chip integriert ist und die von Banken und Kreditinstituten ausgegeben wird. Die Geldkarte dient als bargeldloses Zahlungsmittel.*

Frage:

*Was ist eine Kreditkarte (Synonym)?*

Antwort:

*Den genauen Suchbegriff „Kreditkarte“ habe ich nicht gefunden. Der Begriff „Geldkarte“ ist jedoch gleichbedeutend. Möchtest du die Definition sehen?*

Frage:

*Was ist ein Chip (Stichwort)?*

Antwort:

*Den genauen Suchbegriff „Chip“ habe ich leider nicht gefunden. Der Begriff „Geldkarte“ ist jedoch mit deinem Suchwort verwandt. Möchtest du die Definition sehen?*

Frage:

*Zu welchem Themengebiet gehört der Begriff Geldkarte?*

Antwort:

Der Begriff Geldkarte gehört zum Themengebiet Geld und Job.

Die Verwendung von Stichwort und Synonym ist sinnvoll, da die Wahrscheinlichkeit, dass der Nutzer genau den Begriff eingibt der einem Lemma entspricht relativ gering ist. Aus diesem Grund müssen Synonyme und Stichworte mit dem Lemma verknüpft werden, um auf diesen zu verweisen und damit die Trefferwahrscheinlichkeit zu erhöhen. Der Chatbot soll auf diese Weise direkt auf Begriffe rea-

gieren können, die keinem Lemma entsprechen. Des Weiteren ist jedes Lemma mit einer Kategorie, einem Kurz- und Langtext verknüpft.

#### 5.1.4 Umsetzung in AIML

Hier folgt nun eine mögliche Darstellung des beschriebenen Ansatzes in AIML. Aus dem XML/RDF Code des Lemma Geldkarte (Abbildung 32) lassen sich einige Frage-Antwort Paare ableiten, die dann in AIML ausgedrückt werden können.

```
<Lemma rdf:ID="Geldkarte">
  <chat_Kurztext rdf:datatype="xsd:string"
    >Eine Geldkarte ist eine Bankkarte, in die zusaetzlich
  <gehoeert_zu_Kategorie rdf:resource="#Geld_und_Job"/>
  <ist_aequivalent_mit rdf:resource="#Kreditkarte"/>
  <ist_aequivalent_mit rdf:resource="#Girokonto"/>
  <ist_verwandt_mit rdf:resource="#Bankkarte"/>
  <ist_verwandt_mit rdf:resource="#Chip"/>
  <ist_verwandt_mit rdf:resource="#Geld"/>
</Lemma>
```

Abbildung 32: Beziehungen des Lemma Geldkarte in XML Code

Der Chatbot soll zwischen Synonym- und Stichworteingaben unterscheiden können.

```
<category>
<pattern>Was ist eine Kreditkarte</pattern>
  <template>
    | | | | Den genauen Suchbegriff „Kreditkarte“ habe ich nicht gefunden.
    | | | | Der Begriff „Geldkarte“ ist jedoch gleichbedeutend. Möchtest du
    | | | | die Definition ansehen?
  </template>
</category>
```

Abbildung 33: Reaktion auf Synonymeingabe

Um das zu erreichen, muss ein „vorgeschaltetes“ Programm die Eingabe mit der Ontologie abgleichen und erkennen, ob es sich bei der Eingabe um ein Stichwort, ein Synonym oder ein Lemma handelt.

```

<category>
<pattern>Was ist ein Chip</pattern>
  <template>
    | | | |
    | | | | Den genauen Suchbegriff „Chip“ habe ich leider nicht gefunden.
    | | | | Der Begriff „Geldkarte“ ist jedoch mit deinem Suchwort verwandt.
    | | | | Möchtest du die Definition sehen?
  </template>
</category>

```

Abbildung 34: Reaktion auf Stichworteingabe

Anschließend gibt der Chatbot die entsprechende Antwort aus. In Abbildung 34 handelt es sich um eine Stichworteingabe. Abbildung 35 zeigt ein direktes Matching mit einem Lemma. Der im Triple hinterlegte Text wird daraufhin ausgegeben.

```

<category>
<pattern>Was ist eine Geldkarte</pattern>
  <template>
    | | | | Eine Geldkarte ist eine Bankkarte, in die zusätzlich
    | | | | ein Mikro-Chip integriert ist und die von Banken und
    | | | | Kreditinstituten ausgegeben wird. Die Geldkarte dient
    | | | | als bargeldloses Zahlungsmittel.
  </template>
</category>

```

Abbildung 35: Frage-Antwortpaar für das Lemma *Geldkarte*

Eine andere Möglichkeit ist, den eigentlichen Begriff durch eine Wildcard vom Rest der Eingabe zu trennen (Abbildung 36). Das hat allerdings den Nachteil, dass dann im Extremfall alle Substantive mit einer Wildcard versehen werden und dann praktisch kein Smalltalk mehr möglich ist, da sämtliche Substantive belegt sind.

```

<category>
  <pattern>* Kreditkarte</pattern>
  <template>Den genauen Begriff Kreditkarte habe ich nicht gefunden.
  | | | | Der Begriff Geldkarte ist jedoch gleichbedeutend.
  | | | | <srai>* Geldkarte</srai>
  </template>
</category>

<category>
  <pattern>* Geldkarte</pattern>
  <template>
  | | | | Eine Geldkarte ist eine Bankkarte, ...
  </template>
</category>

```

Abbildung 36: Einsatz von Wildcards

Die obige Abbildung sieht im Dialog wie folgt aus:

Frage:

*Was ist eine Kreditkarte?*

Antwort:

*Den genauen Begriff Kreditkarte habe ich nicht gefunden. Der Begriff Geldkarte ist jedoch gleichbedeutend. Eine Geldkarte ist eine Bankkarte,....*

### 5.1.5 Nutzen des strukturellen Ansatzes

Durch die Wiedergabe der Struktur des Lexikons in einer Ontologie konnte sich hier kein bedeutender Mehrwert erzielen lassen. Der Verweis von Synonymen und Stichworten auf ein Lemma wird bereits mit der bestehenden Form des Lexikon abgedeckt. Jedoch ist es nun möglich, dass der Chatbot bei der Beantwortung einer Suchanfrage zwischen Synonymeingaben und Stichworteingaben unterscheiden kann und somit differenziertere Antworten ausgeben kann. Durch die Verwendung von Synonymen und Stichworten kann eine Verbesserte Trefferwahrscheinlichkeit erreicht werden. Jedoch müssen diese Hilfsörter händisch hinzugefügt werden, was einen zusätzlichen Arbeitsaufwand bedeutet, was den Begriff aber immer noch nicht besser beschreibt.

Mit Hilfe von OWL kann die hier erstellte Ontologie zwar dargestellt werden, die Vielseitigkeit von OWL kommt jedoch, aufgrund der einfachen Struktur der Ontologie, nicht zum Tragen.

## 5.2 Inhaltliche Darstellung der Lexikonartikel

Nachdem die Struktur des Lexikons in einer Ontologie beschrieben ist, soll in diesem Kapitel der Inhalt des Lexikons in einer Ontologie dargestellt werden. Die Kategorie *Reisen und Fahren* ist der Ausgangspunkt der Konzeptualisierung. Zur Beschreibung des Inhalts werden einige der Lemmata aus der genannten Kategorie des „lookedup4you“-Lexikons verwendet. Neben den Lemmata selbst werden die Artikelinhalte der verwendeten Lemmata als Grundlage für die Bildung des Domänenwissens benutzt.

Mit der Ontologie wird beschrieben, wohin verschiedene Arten von Personen reisen, welche Unterkunft sie wählen oder ob sie eine Reiseversicherung abschließen.

Nach der Beschreibung des Inhalts in diesem Kapitel, folgt im nächsten Kapitel der Versuch die Konzepte der beiden Ontologien miteinander zu verknüpfen, also Inhalt und Struktur miteinander zu verknüpfen.

Bisher kann der Bot ein Lemma nur anhand der festgelegten Kurz- oder Langtexte beschreiben. Mit Hilfe einer Ontologie die den Inhalt einer Kategorie repräsentiert, lassen sich diverse zusätzliche Fragen beantworten. Für das Lexikonlemma *Au Pair* lassen sich zum Beispiel auf eine Frage folgende Antwortsätze bilden:

Frage:

*Was kannst du mir noch über das Wort Au Pair sagen?*

Mögliche Antworten:

*Ein Au Pair arbeitet als Haushaltshilfe.*

*Ein Au Pair wohnt bei einer Gastfamilie.*

*Ein Au Pair macht eine Bildungsreise.*

Diese Aussagen sind soweit sinnvoll, da sie den Begriff Au Pair näher spezifizieren und definieren. Ebenso sind jedoch Aussagen möglich die zwar inhaltlich richtig sein können, jedoch im Kontext der Begriffsdefinition des Wortes Au Pair nichts zu Verständnis beitragen:

Frage:

*Was kannst du mir noch über das Wort Au Pair sagen?*

Mögliche Antwort:

*Ein Au Pair schließt eine Reiserücktrittskostenversicherung ab.*

Der Satz *Au Pair schließt eine Reiserücktrittskostenversicherung ab* ist sicherlich korrekt, da ein Au Pair, wie jeder andere Reisende auch, eine solche Versicherung abschließen kann. Im Sinne der Definition des Begriffs hat er jedoch keinen Nutzen, da jeder der eine Reise unternimmt eine solche Versicherung abschließen kann und die Information nichts zum Verständnis des Begriffs beiträgt.

### **5.2.1 Erstellung der Klassenhierarchie für die inhaltlichen Darstellung**

Die Wiedergabe des Inhalts einer bestimmten Kategorie in einer Ontologie, bietet die Möglichkeit, dieses Wissen in strukturierter Form darzustellen und Begriffe miteinander in Beziehung zu setzen. Um eine entsprechende inhaltliche Struktur darzustellen reicht es jedoch nicht aus, nur auf die Lemmata des Lexikons zurück zu greifen. Zusätzlich werden deshalb markante Begriffe aus den Artikeln selbst zur Spezifikation der Ontologie verwendet. Dadurch werden jedoch Lemmata und Artikelbegriffe miteinander vermengt. Dies ist aber notwendig, um zum Beispiel den Wissensbereich Reisen und Fahren sinnvoll darstellen zu können.

So sind die Begriffe *Kurs* und *Unterkunft* den Artikeln entliehen, *Bildungsreise* entspricht einem Lemma und *Versicherung* einer *Unterkategorie* des Lexikons.



Abbildung 37: Klassenhierarchie der Lexikonkategorie *Reisen und Fahren*

Es gab die Überlegung, die vom Lexikon vorgegebenen Unterkategorien auch als solche für die Ontologie zu übernehmen. Dabei traten jedoch zunehmend Schwierigkeiten auf. Zum einen sind die Begriffe *Essstörung* und *Krankheiten* im Lexikon beide Unterkategorien der Kategorie *Körper und Gesundheit*. Das bedeutet, sie befinden sich hierarchisch auf einer Ebene. In einer Ontologie erscheint es jedoch sinnvoller Essstörung als eine Form von Krankheit zu definieren. Also ist *Essstörung* eine Unterklasse oder Instanz von *Krankheiten*.

Ein anderer Grund die Unterkategorienstruktur nicht zu übernehmen, ist die Redundanz verschiedener Unterkategorien. Ein krasses Beispiel hierfür stellt die Unterkategorie *Allgemeines* dar. Sie ist in fast jeder Kategorie vorhanden, wobei einige Artikel in mehreren Unterkategorien

tegorien *Allgemeines* vorhanden sind. Der Sinn einer Ontologie besteht aber darin, einen bestimmten Wissensbereich möglichst genau zu beschreiben und eine Unterkategorie *Allgemeines* wirkt diesem Grundsatz entgegen. Eine Möglichkeit wäre, die in den diversen Unterkategorien *Allgemeines* vorhandenen Artikel besser zuzuordnen und auf diese Unterkategorie gänzlich zu verzichten.

### **5.2.2 Instanzen**

Als Instanz ist nicht das spezifische Hotel „Holiday In“ als Unterkunft interessant, sondern ganz allgemein die Tatsache, dass eine reisende Person eine Unterkunft hat. Deshalb ist Hotel eine Instanz der Klasse Unterkunft und nicht etwa das konkrete Hotel „Holiday In“.

Die aus diesem Zusammenhang gewonnene Information lautet: *Ein Au Pair wohnt bei einer Gastfamilie* oder *Ein Au Pair macht eine Bildungsreise*.

Ein Nachteil dieser Festlegung einer Instanz ist, dass eine weitere Differenzierung des Begriffs Bildungsreise nun nicht mehr möglich ist. Eine Bildungsreise könnte zum Beispiel eine Sprachreise oder aber auch eine Studienfahrt sein. Diese Differenzierung ist durch die Festlegung des Begriffs Bildungsreise als Instanz nun nicht mehr möglich. Legt man die Begriffe Sprachreise und Studienfahrt als Instanzen einer Klasse Bildungsreise fest, so ist eine Verwendung dieses Begriffs auf Instanzenebene nicht mehr möglich. Im Sinne des Lexikons ist aber eine Verwendung des Begriffs Bildungsreise auf Instanzenebene vorzuziehen, da es beim Lexikon um allgemeine Informationen geht.

### **5.2.3 Beziehungen**

Wie auch schon beim vorhergehenden Konzept werden die Instanzen der Klassen über Properties miteinander verknüpft. Aus diesen Konstrukten werden anschließend die Antworten der Frage-

Antwortpaare für den Chatbot generiert. Abbildung 38 zeigt die Beziehungen des Lemmata *Au\_Pair*.

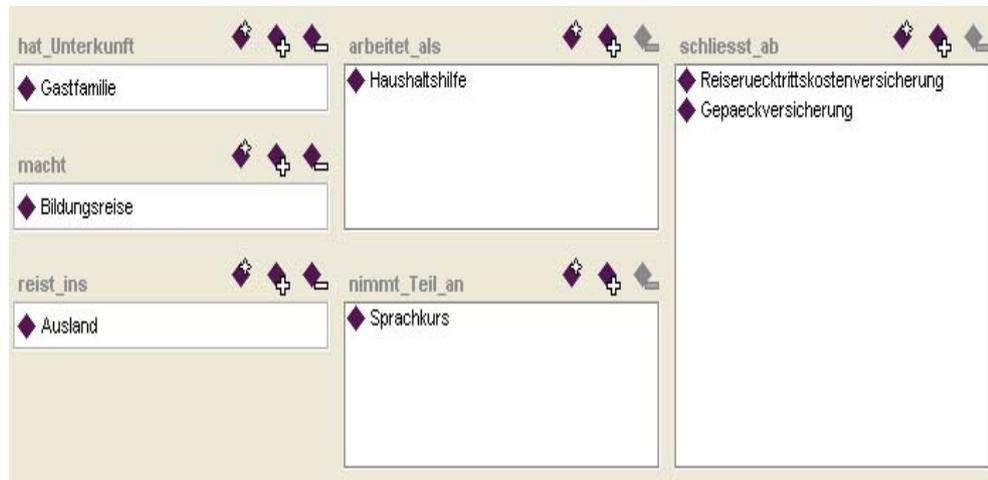


Abbildung 38: Individual *Au Pair* (Protégé)

Abbildung 39 zeigt die Properties der Instanz *Au\_Pair*.

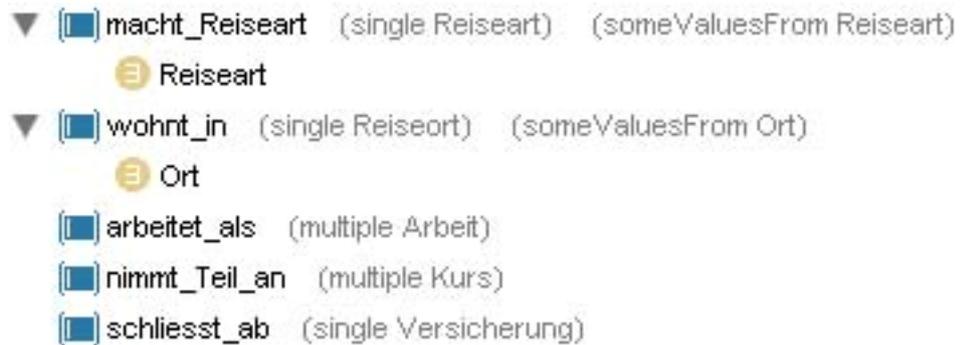


Abbildung 39: Properties der Klasse Person (Protégé)

## 5.2.4 Inhalt in AIML

Der RDF/OWL Code der Instanz *Au\_Pair* zeigt die Beziehungen über die Properties zu anderen Instanzen (Abbildung 40).

```
<Person rdf:ID="Au_Pair">
  <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
  <schliesst_ab rdf:resource="#Gepaeckversicherung"/>
  <arbeitet_als rdf:resource="#Haushaltshilfe"/>
  <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
  <reist_ins rdf:resource="#Ausland"/>
  <hat_Unterkunft rdf:resource="#Gastfamilie"/>
  <macht rdf:resource="#Bildungsreise"/>
</Person>
```

Abbildung 40: Beziehungen der Instanz *Au\_Pair*

Aus dem Code-Triples lassen sich einfache Dialoge generieren (Abbildung 41). Auf die Frage, wohin ein Au Pair reist, kann geantwortet werden, dass ein Au Pair ins Ausland reist.

```
<category>
  <pattern>Wohin reist ein Au Pair</pattern>
  <template>
    | | | | Ein Au Pair reist ins Ausland.|
  </template>
</category>
```

Abbildung 41: einfacher Dialog aus Triple

Im nachfolgenden Beispiel (Abbildung 42) kann auf die Frage, was noch über ein Au Pair zu sagen ist, eine Zufallsantwort gegeben werden.

```
<category>
<pattern>Was kannst du mir noch über ein Au Pair sagen?</pattern>
  <template>
    <random>
      <li>Ein Au Pair arbeitet als Haushaltshilfe.</li>
      <li>Ein Au Pair wohnt bei einer Gastfamilie.</li>
      <li>Ein Au Pair macht eine Sprachreise.</li>
    </random>
  </template>
</category>
```

Abbildung 42: Zufallsantwort auf eine Frage

In dem Beispiel wird über das Tag *random* eine zufällige Auswahl als Antwort auf eine Frage gegeben.

For Individual:  Au\_Pair



Property	Value	Type
arbeitet_als	 Haushaltshilfe	 Arbeit
hat_Unterkunft	 Gastfamilie	 Unterkunft
macht	 Bildungsreise	 Reiseart
nimmt_Teil_an	 Sprachkurs	 Kurs
reist_ins	 Ausland	 Reiseort
schliesst_ab	 Reiseruecktrittskostenversicherung	 Versicherung
schliesst_ab	 Gepaeckversicherung	 Versicherung

Abbildung 43: Triples des Individual *Au\_Pair*

Die Antworten werden mit Hilfe von Triples generiert die in Abbildung 43 zu sehen sind. Das Subjekt stellt hierbei z.B. *Au\_Pair* dar, das Objekt wird durch *Haushaltshilfe* repräsentiert, beide werden durch das Prädikat (Property) *arbeitet\_als* miteinander verknüpft.

### **5.2.5 Nutzen des inhaltlichen Ansatzes**

Der Ansatz bietet die Möglichkeit, inhaltliche Informationen zu den Kategorien zu extrahieren. Diese Informationen sind in den Triples enthalten. Weiters wird die Struktur einer Lexikonkategorie übersichtlich dargestellt und lässt sich gut erweitern.

Durch die Verwendung der Triples sind jedoch auch Aussagen möglich, die zwar inhaltlich korrekt sind, die jedoch keinen spezifischen Informationsnutzen haben.

## **5.3 Verknüpfung von Inhalt und Struktur**

Die Darstellung der Lexikonstruktur in einer Ontologie, bezieht sich lediglich auf die Zugehörigkeiten der einzelnen Lemmata zu ihren Stichworten, Synonymen, hinterlegten Definitionstexten (Kurztext und Langtext) usw. Die inhaltliche Darstellung, die im letzten Abschnitt diskutiert wurde, definiert einen Wissensbereich, dem jedoch die Anbindung an die Lexikonstruktur fehlt. Beide Ansätze für sich genommen, bringen also keinen Mehrwert. In diesem Abschnitt soll deshalb untersucht werden, ob man die Ontologien für Inhalt und Struktur miteinander verknüpfen kann und welchen Nutzen das für einen Chatbot bedeuten kann.

### **5.3.1 Aufbau der Ontologie**

Die Ontologie besteht im Wesentlichen aus der Zusammenführung der vorausgegangenen Ontologieansätze von Inhalt und Struktur. Die Lexikonkategorie *Reisen und Fahren* repräsentiert den inhaltlichen Teil, in dem unter anderem die Lemmata *und* *Gepäckversicherung* als Instanzen dargestellt sind.



Abbildung 44: Klassenhierarchie in Protégé

Da die Kategorie Reisen und Fahren als Klasse, wie auch als Instanz Verwendung findet, wird sie in der Klassenstruktur mit K\_Reisen\_und\_Fahren benannt.

- arbeitet\_als (multiple Arbeit)
- hat\_Unterkunft (single Unterkunft)
- ▼ ■ macht (single Reiseart) (someValuesFrom Reiseart)
  - ⊖ Reiseart
- nimmt\_Teil\_an (multiple Kurs)
- ▼ ■ reist\_ins (single Reiseort) (someValuesFrom Ort)
  - ⊖ Ort
- schliesst\_ab (multiple Versicherung)
- ▼ ■ ist\_aequivalent\_mit (multiple Synonym) (someValuesFrom Synonym)
  - ⊖ Synonym
- ▼ ■ ist\_verwandt\_mit (multiple Stichwort) (someValuesFrom Stichwort)
  - ⊖ Stichwort
- gehoert\_zu\_Kategorie
- hat\_Kurztext (multiple string)
- hat\_Langtext
- hat\_Quiz
- hat\_Video

Abbildung 45: Properties der Klasse Lemma (Protégé)

Die Zusammenführung der beiden Ontologieansätze hat eine Vermischung der Properties zur Folge (Abbildung 44). Einerseits hat man inhaltliche Properties, wie *arbeitet\_als* und andererseits strukturgebende Properties, wie *ist\_verwandt mit*.

For Individual:  Au\_Pair



Property	Value	Type
 arbeitet_als	 Haushaltshilfe	 Arbeit
 gehoert_zu_Kategorie	 Reisen_und_Fahren	 K_Reisen_und_Fahren
 hat_Unterkunft	 Gastfamilie	 Unterkunft
 ist_verwandt_mit	 Kinderbetreuung_im_Ausland	 Stichwort
 ist_verwandt_mit	 Auslandserfahrung	 Stichwort
 ist_verwandt_mit	 Auslandsaufenthalt	 Stichwort
 macht	 Bildungsreise	 Reiseart
 nimmt_Teil_an	 Sprachkurs	 Kurs
 reist_ins	 Ausland	 Reiseort
 schliesst_ab	 Gepaeckversicherung	 Versicherung
 schliesst_ab	 Reiseruecktrittskostenversicherung	 Versicherung

Abbildung 46: Triples des Individuals *Au\_Pair*

Die Aussagen die in diesem Fall mit Hilfe von Triples über ein Au Pair gemacht werden können, beinhalten deshalb ebenfalls inhaltliche und strukturelle Informationen. So kann man sagen, dass ein Au Pair ins Ausland reist aber auch, dass es der Kategorie Reisen und Fahren zuzuordnen ist (Abbildung 45).

### 5.3.2 Nutzen des vermischten Ontologieansatzes

Die Darstellungsform bietet zum einen durch die übersichtliche Klassenstruktur die Möglichkeit, neue Lemmata zu identifizieren und sie in die Inhaltsstruktur zu integrieren. Durch die Verknüpfung mit Stichworten und Synonymen wird die Trefferwahrscheinlichkeit erhöht.

Auf der anderen Seite ist die Vermengung von inhaltlichen und strukturellen Properties ein Nachteil, da diese Darstellungsweise unübersichtlich ist.

## 5.4 Weitere Ansätze und Probleme

### Nutzung von Substantiven

Die Lemmata in dem hier als Grundlage dienenden Lexikon sind ausschließlich Substantive. Substantive sind also besonders geeig-

nete Suchwörter. Da Substantive immer groß geschrieben werden, könnte dies genutzt werden, um bei der Sucheingabe auf Groß-/Kleinschreibung zu achten. Dieser Ansatz setzt aber die Beschränkung des Lexikons auf die deutsche Sprache voraus, da in anderen vielleicht relevanten Sprachen (z.B. Englisch) Substantive im Allgemeinen klein geschrieben werden.

### **Online-Slang**

Ebenso ist der Online-Slang der Nutzer ein Problem. Jegliche Schreibweisen, Synonyme und Stichworte müssten erfasst werden. Ein aufwändiges Verfahren, dass viel Zeit in Anspruch nimmt und den eigentlichen Vorteil einer Ontologie entgegensteht.

### **Konnotation und Denotation**

Begriffe können mehrere Ebenen der Bedeutung haben, auch Konnotationen genannt, die von einer Ontologie nur schwer erfasst werden können. Konnotation ist vom aktuellen Redner und der aktuellen Situation abhängig. Unter konnotativer Bedeutung versteht man „eine sekundäre Bedeutung, die ein Sprachzeichen neben einer primären Standard- oder Kernbedeutung hat, welche als Denotation definiert wird.“ (NÖTH 2000, S.151)

Als Beispiel soll hier das Wort Schatten angeführt sein, dass in sonnenreichen Regionen positive und in sonnenarmen Gebieten negative Assoziationen hervorrufen kann. In der Sahara spendet der Schatten Schutz vor der brennenden Sonne, während Schatten im deutschen Sprachraum eher negativ besetzt ist.

Für den Chatbot bedeutet dies, dass er im Zweifelsfall nicht weiß, auf welcher Bedeutungsebene der Nutzer über einen Begriff mit ihm kommunizieren will.

Um das Problem in einer Ontologie zu lösen, müsste man persönliche, soziale, lokale oder technische Bedeutungsebenen erfassen.

Das Problem liegt jedoch in der Erkennung der Bedeutungsebene durch den Chatbot. Das ist deshalb schwierig, da sich die Intention des Nutzers nur aus dem Satzzusammenhang ableiten lässt.

## 6 Zusammenfassung und Ergebnisbetrachtung

In dieser Arbeit wurden verschiedene Ontologieansätze entwickelt, um sie als Grundlage für einen Chatbot zu nutzen. Es wurde untersucht, ob diese Konzepte dazu beitragen, das Verhalten des Chatbots zu verbessern,

Die Struktur des Lexikons *Lookedup4You* lässt sich zwar in einer Ontologie darstellen. Der Ansatz bringt aber wenig Nutzen, da die Struktur des Lexikons bereits als Datenbankversion im Einsatz ist. Neben der Anbindung an Synonyme und Stichwörtern ist nun auch der Verweis auf eine Kategorie möglich. Dennoch kann man sagen, dass eine grundsätzliche Nutzung dieses Ansatzes möglich ist. Außerdem bietet er, dank der Benutzeroberfläche, eine übersichtliche Darstellung und ermöglicht so eine leichte Pflege und Editierung der strukturellen Inhalte.

Der inhaltliche Ansatz lässt differenziertere Antworten zu, dank auswertbarer Triple. Durch die Verwendung der Triples sind jedoch auch Aussagen möglich, die zwar inhaltlich korrekt sind, die jedoch keinen spezifischen Informationsnutzen haben.

Der Ansatz bringt Probleme mit sich, da hier einige Lemmata des Lexikons als Klassen definiert sind, andere wiederum als Individuals. Diese Vorgehensweise erschien bei der Erstellung der Wissensdomäne sinnvoller, da sie den zu beschreibenden Wissensbereich besser beschreibt.

Die Klassenstrukturbildung der inhaltlichen Ontologie ist unproblematisch, solange nur eine Lexikonkategorie erfasst wird. Erfasst man jedoch weitere Kategorien des Lexikons in einer Ontologie, lässt es sich nicht vermeiden, dass bestimmte Klassen, wie zum Beispiel die Klasse Person, in mehreren Kategorien auftauchen, um die entsprechende Kategorie ausreichend modellieren zu können. Ein möglicher

Ausweg wäre die Erstellung einer globalen Ontologie, die sämtliche Informationen aller Kategorien in sich vereint. Auf diese Weise lassen sich Redundanzen vermeiden, jedoch gibt es dann keine Trennung der Kategorien mehr.

Auch hätte eine totale Loslösung von den Lexikonkategorien möglicherweise zu einem anderen Ergebnis geführt. Eine allgemeine Ontologie zum Thema Verbraucherschutz ohne genaue inhaltliche Vorgaben durch Lemmata, Kategorien und Artikelinhalte hätte die Entwicklung einer entsprechenden Ontologie erleichtert. Die ursprüngliche Idee, dass sämtliche Lemmata als Individuals oder als Klassen dargestellt werden, ließ sich aus strukturellen Gründen nicht verwirklichen.

Der vermischte Ansatz aus Inhalt und Struktur bietet die Möglichkeit einen Wissensbereich (eine Kategorie) durch neue Klassen und Instanzen zu erweitern und diese Instanzen in die Stichwort- und Synonymstruktur zu integrieren.

Vom technischen Standpunkt aus, hätten die Ontologien auch mit RDFS realisiert werden können. Mit OWL ist es möglich, logische Schlussfolgerungen zu ziehen und Aussagen zu treffen, die mit RDFS nicht möglich sind. Diese Fähigkeit von OWL kommt jedoch nicht zum tragen.

Insgesamt kann man sagen, dass die in dieser Arbeit entwickelten Konzepte das Verhalten eines Chatbots nur Ansatzweise verbessern. Die Möglichkeit, im Dialog, zwischen Synonym und Stichwort zu unterscheiden bietet eine kleine Verhaltensverbesserung. Auch der Verweis auf die zugehörige Kategorie bietet einen kleinen Mehrwert. Auch die Verwendung von Triples, mit deren Hilfe die Begriffe näher beschreiben werden können, trägt zu einer Verbesserung bei.

## Literatur-/Quellenverzeichnis

### **BIRKENBIHL 2006**

BIRKENBIHL, Klaus: Standards für das Semantic Web. In: Pelligrini, Tassilo (Hrsg.) ; Blumbauer, Andreas (Hrsg.): *Semantic Web : Wege zur vernetzten Wissensgesellschaft*. Berlin : Springer-Verl., 2006, S. 73-88. – ISBN 3-540-25719-5

### **BLUMBAUER 2006**

BLUMBAUER, Andreas ; Pelligrini, Tassilo: Semantic Web und semantische Technologien: Zentrale Begriffe und Unterscheidungen. In: Pelligrini, Tassilo (Hrsg.) ; Blumbauer, Andreas (Hrsg.): *Semantic Web : Wege zur vernetzten Wissensgesellschaft*. Berlin : Springer-Verl., 2006, S. 9-26. – ISBN 3-540-25719-5

### **BRAUN 2003**

BRAUN, Alexander: *Chatbots in der Kundenkommunikation*. Berlin (u.a.) : Springer, 2003. - ISBN 3-540-00019-4

### **BURKART 2004**

BURKART, Margarete: Thesaurus. In: Kuhlen, Rainer (Hrsg.) ; Seeger, Thomas (HRSG.) ; Strauch, Dietmar (Hrsg.): *Grundlagen der praktischen Information und Dokumentation : Handbuch zur Einführung in die Informationswissenschaft und – praxis*. 5., völlig neu gefasste Ausg. München : Saur, 2004, S.141-154

### **DACONTA 2003**

DACONTA, Michael C. ; Obrst, Leo J. ; Smith, Kevin T.: *The semantic web : A guideline to the future of XML, Web Services, and knowledge management*. Indianapolis : Wiley Publishing, Inc., 2003. – ISBN 0-471-43257-1

### **ECKSTEIN 2004**

ECKSTEIN, Rainer ; Eckstein, Silke: *XML und Datenmodellierung : XML-Schema und RDF zur Modellierung von Daten und Metadaten*. 1. Aufl. Heidelberg : dpunkt.Verl., 2004. – ISBN 3-89864-222-4

### **FUGMANN 1992**

FUGMANN, Robert: *Theoretische Grundlagen der Indexierungspraxis*. Frankfurt : Indeks-Verl., 1992. – ISBN 3-88672-400-x

**GALINSKI 2006**

GALINSKI, Christian: Wozu Normen? Wozu semantische Interoperabilität?. In: Pelligrini, Tassilo (Hrsg.) ; Blumbauer, Andreas (Hrsg.): *Semantic Web : Wege zur vernetzten Wissensgesellschaft*. Berlin : Springer-Verl., 2006, S 47-72. – ISBN 3-540-25719-5

**GAUS 2005**

GAUS, Wilhelm: *Dokumentations- und Ordnungslehre : Theorie und Praxis des Information Retrieval*. 4., überarb. Aufl. Berlin : Springer, 2005. – ISBN 978-3-540-23818-8

**GRUBER 1993**

GRUBER, Tom: *What is an Ontology?*. URL:<<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>> (letztes Zugriffsdatum: 2008-03-20)

**HILGENFELD 2003**

HILGENFELD, Inga; Schmidt, Dirk: *Möglichkeiten und Grenzen des Ontologiebasierten Editors Protégé. Seminararbeit im Rahmen des Seminars Management Support und Wirtschaftsinformatik Wintersemester2003/2004*.

URL:<[http://www.oec.uni-osnabrueck.de/fachgeb/winf2/Seminare/wi2\\_03/Seminararbeiten/Thema07.pdf](http://www.oec.uni-osnabrueck.de/fachgeb/winf2/Seminare/wi2_03/Seminararbeiten/Thema07.pdf)> (letztes Zugriffsdatum: 2008-04-07)

**HITZLER 2008**

HITZLER, Pascal ; Krötzsch, Markus ; Rudolph, Sebastian ; Sure, York: *Semantic Web – Grundlagen*. 1. Aufl. Berlin : Springer-Verl., 2008. – ISBN 978-3-540-33993-9

**KIENREICH 2006**

KIENREICH, Wolfgang ; Strohmaier, Markus: Wissensmodellierung – Basis für die Anwendung. In: Pelligrini, Tassilo (Hrsg.) ; Blumbauer, Andreas (Hrsg.): *Semantic Web : Wege zur vernetzten Wissensgesellschaft*. Berlin : Springer-Verl., 2006, S. 359-372. – ISBN 3-540-25719-5

**MÖBUS 2006**

MÖBUS, Claus: *Web-Kommunikation mit OpenSource : Chatbots, Virtuelle Messen, Rich-Media-Content*. Berlin : Springer-Verl., 2006. – ISBN 10 3-540-23286-9

**NÖTH 2000**

NÖTH, Winfried: *Handbuch der Semiotik*. 2., vollständig neu bearb. Aufl. Stuttgart : Metzler, 2000. – ISBN 3-476-01226-3

**NOY 2001**

NOY, Natalya F.; Sintek, Michael; Decker, Stefan; Crubézy, Monica; Fergerson, Ray W.; Musen, Mark A.: *Creating Semantic Web Contents with Protégé-2000*. URL: <<http://icc.mpei.ru/documents/00000829.pdf>>. (letztes Zugriffsdatum: 2008-04-08).

**SMITH 2004**

SMITH, Michael K. ; Welty, Chris ; McGuinness, Deborah L.: *OWL Web Ontology Language Guide*. URL: <<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>>. [Stand: 10.02.2004] (letztes Zugriffsdatum: 2008-04-02)

**WERSIG 1985**

WERSIG, Gernot: *Thesaurus-Leitfaden : Eine Einführung in das Thesaurus-Prinzip in Theorie und Praxis*. 2. erg. Aufl. München : Saur, 1985. – ISBN 3-598-21252-6

**WIDHALM 2002**

WIDHALM, Richard ; Mück, Thomas: *Topic Maps : Semantische Suche im Internet*. Berlin : Springer-Verl., 2002. - ISBN 3-54041719-2

**WIKIPEDIA 2008 a**

WIKIPEDIA: *Topic Maps*. URL: <[http://de.wikipedia.org/wiki/Topic\\_Maps](http://de.wikipedia.org/wiki/Topic_Maps)>. [Stand: 03.2008] (letztes Zugriffsdatum: 2008-03-23)

**WIKIPEDIA 2008 b**

WIKIPEDIA: *Otologie*. URL: <<http://de.wikipedia.org/wiki/Ontologie>>. [Stand: 03.2008] (letztes Zugriffsdatum: 2008-03-02)

## Anhang A : Ontologie der Lexikonstruktur

```
<?xml version="1.0"?>
```

```
<!DOCTYPE rdf:RDF [  
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >  
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >  
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >  
>
```

```
<rdf:RDF xmlns="http://www.owl-  
ontologies.com/Ontology1205855785.owl#"  
  xml:base="http://www.owl-ontologies.com/Ontology1205855785.owl"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:owl="http://www.w3.org/2002/07/owl#">  
  <owl:Ontology rdf:about=""/>  
  <Synonym rdf:ID="Abo">  
    <Synonym_verweist_auf rdf:resource="#Abonnement"/>  
  </Synonym>  
  <Lemma rdf:ID="Abonnement">  
    <hat_Kurztext rdf:datatype="&xsd:string"  
      >Ein Abonnement ist der regelmae&#223;ige Bezug einer Leis-  
tung</hat_Kurztext>  
    <gehoeert_zu_Kategorie rdf:resource="#Geld_und_Job"/>  
    <ist_aequivalent_mit rdf:resource="#Abo"/>  
    <ist_verwandt_mit rdf:resource="#Abos"/>  
  </Lemma>  
  <Stichwort rdf:ID="Abos">  
    <Synonym_verweist_auf rdf:resource="#Abonnement"/>  
    <Stichwort_verweist_auf rdf:resource="#Abonnement"/>  
  </Stichwort>  
  <Stichwort rdf:ID="Adrenalin">  
    <Stichwort_verweist_auf rdf:resource="#Extremsportart"/>  
  </Stichwort>  
  <Stichwort rdf:ID="Adresse">  
    <Stichwort_verweist_auf rdf:resource="#Freemail"/>  
  </Stichwort>  
  <Lemma rdf:ID="Allergie">  
    <hat_Kurztext rdf:datatype="&xsd:string"  
      >Allergien sind eine ueberempfindliche Reaktion des Koerpers auf  
verschiedene Inhaltsstoffe, so genannte Allergene.</hat_Kurztext>
```

```

    <ist_verwandt_mit rdf:resource="#Stress"/>
</Lemma>
<Stichwort rdf:ID="Arbeit">
    <Stichwort_verweist_auf rdf:resource="#Arbeitsvertrag"/>
</Stichwort>
<Lemma rdf:ID="Arbeitsvertrag">
    <hat_Kurztext rdf:datatype="&xsd:string"
        >Ein Arbeitsvertrag ist ein Vertrag zwischen einem Arbeitgeber und
        einem Arbeitnehmer, in dem die Rechte und Pflichten beider Vertragspar-
        teien bez&#252;glich des Arbeitsverhaeltnisses festgelegt
        sind.</hat_Kurztext>
    <ist_verwandt_mit rdf:resource="#Job"/>
    <ist_verwandt_mit rdf:resource="#Arbeit"/>
</Lemma>
<Lemma rdf:ID="Au_Pair"/>
<Stichwort rdf:ID="Bank">
    <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<Stichwort rdf:ID="Bankkarte">
    <Stichwort_verweist_auf rdf:resource="#Geldkarte"/>
</Stichwort>
<Stichwort rdf:ID="Bungee">
    <Stichwort_verweist_auf rdf:resource="#Extremsportart"/>
</Stichwort>
<Stichwort rdf:ID="Chip">
    <Stichwort_verweist_auf rdf:resource="#Geldkarte"/>
</Stichwort>
<Kategorie rdf:ID="Computer_und_Internet"/>
<Stichwort rdf:ID="Diebstahl">
    <Stichwort_verweist_auf rdf:resource="#Plagiat"/>
</Stichwort>
<Stichwort rdf:ID="E-Mail">
    <Stichwort_verweist_auf rdf:resource="#Freemail"/>
</Stichwort>
<Stichwort rdf:ID="Email">
    <Stichwort_verweist_auf rdf:resource="#Freemail"/>
</Stichwort>
<Lemma rdf:ID="Extremsportart">
    <hat_Kurztext rdf:datatype="&xsd:string"
        >Unter Extremsportarten versteht man sportliche Aktivitaeten, bei
        denen der Koerper einer extremen Belastung ausgesetzt
        wird.</hat_Kurztext>
    <ist_verwandt_mit rdf:resource="#Bungee"/>
    <ist_verwandt_mit rdf:resource="#Rafting"/>
    <ist_verwandt_mit rdf:resource="#Fallschirmspringen"/>
    <ist_verwandt_mit rdf:resource="#Adrenalin"/>
</Lemma>

```

```

<Stichwort rdf:ID="Fallschirmspringen">
  <Stichwort_verweist_auf rdf:resource="#Extremsportart"/>
</Stichwort>
<Synonym rdf:ID="free_E-Mail">
  <Synonym_verweist_auf rdf:resource="#Freemail"/>
</Synonym>
<Lemma rdf:ID="Freemail">
  <hat_Kurztext rdf:datatype="&xsd:string"
    >Freemail ist die Bezeichnung f&#252;r kostenlose E-Mails im In-
ternet.</hat_Kurztext>
  <ist_aequivalent_mit rdf:resource="#free_E-Mail"/>
  <ist_verwandt_mit rdf:resource="#Internet"/>
  <ist_verwandt_mit rdf:resource="#Email"/>
  <ist_verwandt_mit rdf:resource="#Adresse"/>
  <ist_verwandt_mit rdf:resource="#E-Mail"/>
  <ist_verwandt_mit rdf:resource="#kostenlos"/>
  <ist_verwandt_mit rdf:resource="#Mail"/>
</Lemma>
<Stichwort rdf:ID="Gebuehr">
  <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<owl:ObjectProperty rdf:ID="gehoeert_zu_Kategorie">
  <rdfs:domain rdf:resource="#Lemma"/>
  <owl:inverseOf rdf:resource="#gehoeert_zu_Lemma"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="gehoeert_zu_Lemma">
  <rdfs:range rdf:resource="#Lemma"/>
  <owl:inverseOf rdf:resource="#gehoeert_zu_Kategorie"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Geld">
  <Stichwort_verweist_auf rdf:resource="#Geldkarte"/>
</Stichwort>
<Kategorie rdf:ID="Geld_und_Job">
  <gehoeert_zu_Lemma rdf:resource="#Abonnement"/>
  <gehoeert_zu_Lemma rdf:resource="#Geldkarte"/>
</Kategorie>
<Lemma rdf:ID="Geldkarte">
  <hat_Kurztext rdf:datatype="&xsd:string"
    >Eine Geldkarte ist eine Bankkarte, in die zusaetzlich ein Mikro-
Chip integriert ist und die von Banken und Kreditinstituten ausgegeben
wird. Die Geldkarte dient als bargeldloses Zahlungsmittel.</hat_Kurztext>
  <gehoeert_zu_Kategorie rdf:resource="#Geld_und_Job"/>
  <ist_aequivalent_mit rdf:resource="#Girokonto"/>
  <ist_aequivalent_mit rdf:resource="#Kreditkarte"/>
  <ist_verwandt_mit rdf:resource="#Geld"/>
  <ist_verwandt_mit rdf:resource="#Chip"/>
  <ist_verwandt_mit rdf:resource="#Bankkarte"/>

```

```

</Lemma>
<Synonym rdf:ID="Girokonto">
  <Synonym_verweist_auf rdf:resource="#Geldkarte"/>
</Synonym>
<Stichwort rdf:ID="Glaebiger">
  <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<Kategorie rdf:ID="Handy"/>
<owl:DatatypeProperty rdf:ID="hat_Kurztext">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hat_Langtext">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hat_Quiz">
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hat_Video">
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<Stichwort rdf:ID="Internet">
  <Stichwort_verweist_auf rdf:resource="#Freemail"/>
</Stichwort>
<owl:ObjectProperty rdf:ID="ist_aequivalent_mit">
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="#Synonym"/>
  <owl:inverseOf rdf:resource="#Synonym_verweist_auf"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ist_verwandt_mit">
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lemma"/>
  <rdfs:range rdf:resource="#Stichwort"/>
  <owl:inverseOf rdf:resource="#Stichwort_verweist_auf"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Job">
  <Stichwort_verweist_auf rdf:resource="#Arbeitsvertrag"/>
</Stichwort>
<Stichwort rdf:ID="Jod">
  <Stichwort_verweist_auf rdf:resource="#Spurenelement"/>
</Stichwort>

```

```

<owl:Class rdf:ID="Kategorie">
  <rdfs:subClassOf rdf:resource="#Lemmaverkneuepfungen"/>
</owl:Class>
<Kategorie rdf:ID="Koerper_und_Gesundheit"/>
<Stichwort rdf:ID="kostenlos">
  <Stichwort_verweist_auf rdf:resource="#Freemail"/>
</Stichwort>
<Stichwort rdf:ID="Kredit">
  <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<Synonym rdf:ID="Kreditkarte">
  <Synonym_verweist_auf rdf:resource="#Geldkarte"/>
</Synonym>
<Stichwort rdf:ID="Kreditschulden">
  <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<owl:Class rdf:ID="Lemma">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#gehört_zu_Kategorie"/>
      <owl:someValuesFrom rdf:resource="#Kategorie"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#ist_aequivalent_mit"/>
      <owl:someValuesFrom rdf:resource="#Synonym"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#ist_verwandtschaftlich"/>
      <owl:someValuesFrom rdf:resource="#Stichwort"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:label xml:lang="de">Lemma</rdfs:label>
  <rdfs:comment xml:lang="de"></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Lemmaverkneuepfungen">
<Stichwort rdf:ID="Mail">
  <Stichwort_verweist_auf rdf:resource="#Freemail"/>
</Stichwort>
<Lemma rdf:ID="Plagiat">
  <hat_Kurztext rdf:datatype="&xsd:string">
    >Der Diebstahl von geistigem Eigentum wird auch als Plagiat be-
    zeichnet.</hat_Kurztext>

```

```

    <ist_verwandt_mit rdf:resource="#Urheberrecht"/>
    <ist_verwandt_mit rdf:resource="#Diebstahl"/>
</Lemma>
<Stichwort rdf:ID="Rafting">
    <Stichwort_verweist_auf rdf:resource="#Extremsportart"/>
</Stichwort>
<Stichwort rdf:ID="Rauchen">
    <Stichwort_verweist_auf rdf:resource="#Zigaretten"/>
</Stichwort>
<Lemma rdf:ID="Reisegepaeckversicherung"/>
<Kategorie rdf:ID="Reisen_und_Fahren"/>
<Lemma rdf:ID="Reiseruecktrittskostenversicherung"/>
<Lemma rdf:ID="Reiseunfallversicherung"/>
<Lemma rdf:ID="Schulden">
    <hat_Kurztext rdf:datatype="&xsd:string"
        >Schulden hat man, wenn man eine Rechnung, geliehenes Geld oder
einen zur&#252;ckzahlen muss.</hat_Kurztext>
    <ist_verwandt_mit rdf:resource="#Kredit"/>
    <ist_verwandt_mit rdf:resource="#Vertrag"/>
    <ist_verwandt_mit rdf:resource="#Schuldnerberatung"/>
    <ist_verwandt_mit rdf:resource="#Kreditschulden"/>
    <ist_verwandt_mit rdf:resource="#Gebuehr"/>
    <ist_verwandt_mit rdf:resource="#Glaebiger"/>
    <ist_verwandt_mit rdf:resource="#Bank"/>
</Lemma>
<Stichwort rdf:ID="Schuldnerberatung">
    <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<Kategorie rdf:ID="Sport_und_Fitness"/>
<Lemma rdf:ID="Spurenelement">
    <hat_Kurztext rdf:datatype="&xsd:string"
        >purenelemente sind chemische Stoffe in Miniaturformat, die
f&#252;r lebenswichtige Stoffwechsel-Funktionen im menschlichen Koer-
per gebraucht werden.</hat_Kurztext>
    <ist_verwandt_mit rdf:resource="#Jod"/>
    <ist_verwandt_mit rdf:resource="#Spurenelemente"/>
    <ist_verwandt_mit rdf:resource="#Stoffe"/>
</Lemma>
<Stichwort rdf:ID="Spurenelemente">
    <Stichwort_verweist_auf rdf:resource="#Spurenelement"/>
</Stichwort>
<owl:Class rdf:ID="Stichwort">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#Stichwort_verweist_auf"/>
            <owl:someValuesFrom rdf:resource="#Lemma"/>
        </owl:Restriction>

```

```

    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Woerter"/>
    <owl:disjointWith rdf:resource="#Synonym"/>
</owl:Class>
<Stichwort rdf:ID="Stichwort_3"/>
<owl:ObjectProperty rdf:ID="Stichwort_verweist_auf">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Stichwort"/>
  <rdfs:range rdf:resource="#Lemma"/>
  <owl:inverseOf rdf:resource="#ist_verwandt_mit"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Stoffe">
  <Stichwort_verweist_auf rdf:resource="#Spurenelement"/>
</Stichwort>
<Stichwort rdf:ID="Stress">
  <Stichwort_verweist_auf rdf:resource="#Allergie"/>
</Stichwort>
<owl:Class rdf:ID="Synonym">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Synonym_verweist_auf"/>
      <owl:someValuesFrom rdf:resource="#Lemma"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Woerter"/>
  <owl:disjointWith rdf:resource="#Stichwort"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="Synonym_verweist_auf">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Synonym"/>
  <rdfs:range rdf:resource="#Lemma"/>
  <owl:inverseOf rdf:resource="#ist_aequivalent_mit"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Tabak">
  <Stichwort_verweist_auf rdf:resource="#Zigaretten"/>
</Stichwort>
<Kategorie rdf:ID="Tiere_und_Umwelt"/>
<Kategorie rdf:ID="Trends_und_Shopping"/>
<Kategorie rdf:ID="TV_und_Musik"/>
<Stichwort rdf:ID="Urheberrecht">
  <Stichwort_verweist_auf rdf:resource="#Plagiat"/>
</Stichwort>
<Kategorie rdf:ID="Verbraucherrecht"/>
<Stichwort rdf:ID="Vertrag">
  <Stichwort_verweist_auf rdf:resource="#Schulden"/>
</Stichwort>
<owl:Class rdf:ID="Woerter">

```

```
<rdfs:subClassOf rdf:resource="#Lemmaverknuepfungen"/>
</owl:Class>
<Stichwort rdf:ID="Zigarette">
  <Stichwort_verweist_auf rdf:resource="#Zigaretten"/>
</Stichwort>
<Lemma rdf:ID="Zigaretten">
  <hat_Kurztext rdf:datatype="&xsd:string"
    >Zigaretten sind Genussmittel, die aus den getrockneten und feinge-
    schnittenen Blaettern der Tabakpflanze hergestellt werden.</hat_Kurztext>
  <ist_verwandt_mit rdf:resource="#Zigarette"/>
  <ist_verwandt_mit rdf:resource="#Rauchen"/>
  <ist_verwandt_mit rdf:resource="#Tabak"/>
</Lemma>
</rdf:RDF>
```

## Anhang B : Ontologie des Inhalts

```
<?xml version="1.0"?>
```

```
<!DOCTYPE rdf:RDF [  
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >  
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >  
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >  
>  
>
```

```
<rdf:RDF xmlns="http://www.owl-  
ontologies.com/Ontology1206735041.owl#"  
  xml:base="http://www.owl-ontologies.com/Ontology1206735041.owl"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:owl="http://www.w3.org/2002/07/owl#">  
  <owl:Ontology rdf:about=""/>  
  <owl:AllDifferent>  
    <owl:distinctMembers rdf:parseType="Collection"/>  
  </owl:AllDifferent>  
  <owl:Class rdf:ID="Arbeit">  
    <rdfs:subClassOf rdf:resource="#Taetigkeit"/>  
  </owl:Class>  
  <owl:ObjectProperty rdf:ID="arbeitet_als">  
    <rdfs:domain rdf:resource="#Person"/>  
    <rdfs:range rdf:resource="#Arbeit"/>  
  </owl:ObjectProperty>  
  <Person rdf:ID="Au_Pair">  
    <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>  
    <schliesst_ab rdf:resource="#Gepaeckversicherung"/>  
    <arbeitet_als rdf:resource="#Haushaltshilfe"/>  
    <nimmt_Teil_an rdf:resource="#Sprachkurs"/>  
    <reist_ins rdf:resource="#Ausland"/>  
    <hat_Unterkunft rdf:resource="#Gastfamilie"/>  
    <macht rdf:resource="#Bildungsreise"/>  
  </Person>  
  <Reiseort rdf:ID="Ausland"/>  
  <Reiseart rdf:ID="Bildungsreise"/>  
  <Unterkunft rdf:ID="Gastfamilie"/>  
  <Versicherung rdf:ID="Gepaeckversicherung"/>  
  <owl:ObjectProperty rdf:ID="hat_Dauer"/>  
  <owl:ObjectProperty rdf:ID="hat_Unterkunft">
```

```

    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Unterkunft"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hat_Zielort"/>
<Arbeit rdf:ID="Haushaltshilfe"/>
<Unterkunft rdf:ID="Herberge"/>
<Unterkunft rdf:ID="Hotel"/>
<Reiseort rdf:ID="Inland"/>
<owl:ObjectProperty rdf:ID="ist_aehnlich"/>
<owl:ObjectProperty rdf:ID="ist_aequivalent"/>
<owl:Class rdf:ID="Kurs">
  <rdfs:subClassOf rdf:resource="#Taetigkeit"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="Kurztext">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="macht">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Reiseart"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="nimmt_Teil_an">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Kurs"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Ort">
  <rdfs:subClassOf rdf:resource="#Reisen_und_Fahren"/>
</owl:Class>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#macht"/>
      <owl:someValuesFrom rdf:resource="#Reiseart"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#reist_ins"/>
      <owl:someValuesFrom rdf:resource="#Ort"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Reisen_und_Fahren"/>
</owl:Class>
<owl:Class rdf:ID="Reiseart">
  <rdfs:subClassOf rdf:resource="#Reisen_und_Fahren"/>
</owl:Class>

```

```

<owl:Class rdf:ID="Reisen_und_Fahren"/>
<owl:Class rdf:ID="Reiseort">
  <rdfs:subClassOf rdf:resource="#Ort"/>
</owl:Class>
<Versicherung rdf:ID="Reiseruecktrittskostenversicherung"/>
<Versicherung rdf:ID="Reiseunfallversicherung"/>
<owl:ObjectProperty rdf:ID="reist_ins">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Reiseort"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="schliesst_ab">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Versicherung"/>
</owl:ObjectProperty>
<Person rdf:ID="Schueler">
  <schliesst_ab rdf:resource="#Reiseunfallversicherung"/>
  <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
  <reist_ins rdf:resource="#Inland"/>
  <hat_Unterkunft rdf:resource="#Gastfamilie"/>
  <macht rdf:resource="#Bildungsreise"/>
</Person>
<Kurs rdf:ID="Sprachkurs"/>
<Person rdf:ID="Student">
  <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
  <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
  <reist_ins rdf:resource="#Ausland"/>
  <hat_Unterkunft rdf:resource="#Hotel"/>
  <macht rdf:resource="#Bildungsreise"/>
</Person>
<owl:Class rdf:ID="Taetigkeit">
  <rdfs:subClassOf rdf:resource="#Reisen_und_Fahren"/>
</owl:Class>
<Kurs rdf:ID="Tauchkurs"/>
<Person rdf:ID="Tourist">
  <schliesst_ab rdf:resource="#Reiseunfallversicherung"/>
  <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
  <schliesst_ab rdf:resource="#Gepaeckversicherung"/>
  <nimmt_Teil_an rdf:resource="#Tauchkurs"/>
  <reist_ins rdf:resource="#Ausland"/>
  <hat_Unterkunft rdf:resource="#Hotel"/>
  <macht rdf:resource="#Urlaubsreise"/>
</Person>
<owl:Class rdf:ID="Unterkunft">
  <rdfs:subClassOf rdf:resource="#Ort"/>
</owl:Class>
<Reiseart rdf:ID="Urlaubsreise"/>

```

```
<owl:Class rdf:ID="Versicherung">  
  <rdfs:subClassOf rdf:resource="#Reisen_und_Fahren"/>  
</owl:Class>  
</rdf:RDF>
```

## Anhang C : Ontologie von Struktur und Inhalt

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.owl-
ontologies.com/Ontology1206735041.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1206735041.owl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>
  <owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection"/>
  </owl:AllDifferent>
  <Synonym rdf:ID="Abo"/>
  <Stichwort rdf:ID="Abos"/>
  <Stichwort rdf:ID="Adrenalin"/>
  <Stichwort rdf:ID="Adresse"/>
  <owl:Class rdf:ID="Arbeit">
    <rdfs:subClassOf rdf:resource="#Taetigkeit"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="arbeitet_als">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Arbeit"/>
  </owl:ObjectProperty>
  <Person rdf:ID="Au_Pair">
    <schliesst_ab rdf:resource="#Gepaeckversicherung"/>
    <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
    <gehört_zu_Kategorie rdf:resource="#Reisen_und_Fahren"/>
    <arbeitet_als rdf:resource="#Haushaltshilfe"/>
    <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
    <reist_ins rdf:resource="#Ausland"/>
    <hat_Unterkunft rdf:resource="#Gastfamilie"/>
    <macht rdf:resource="#Bildungsreise"/>
    <ist_verwandt_mit rdf:resource="#Kinderbetreuung_im_Ausland"/>
    <ist_verwandt_mit rdf:resource="#Auslandserfahrung"/>
  </Person>
</rdf:RDF>
```

```

    <ist_verwandt_mit rdf:resource="#Auslandsaufenthalt"/>
</Person>
<Reiseort rdf:ID="Ausland"/>
<Stichwort rdf:ID="Auslandsaufenthalt">
    <Stichwort_verweist_auf rdf:resource="#Au_Pair"/>
</Stichwort>
<Stichwort rdf:ID="Auslandserfahrung">
    <Stichwort_verweist_auf rdf:resource="#Au_Pair"/>
</Stichwort>
<Stichwort rdf:ID="Bank"/>
<Stichwort rdf:ID="Bankkarte"/>
<Reiseart rdf:ID="Bildungsreise"/>
<Stichwort rdf:ID="Bungee"/>
<Stichwort rdf:ID="Chip"/>
<Stichwort rdf:ID="Diebstahl"/>
<Stichwort rdf:ID="E-Mail"/>
<Stichwort rdf:ID="Email"/>
<Stichwort rdf:ID="Fallschirmspringen"/>
<Synonym rdf:ID="free_E-Mail"/>
<Unterkunft rdf:ID="Gastfamilie"/>
<Stichwort rdf:ID="Gebuehr"/>
<owl:ObjectProperty rdf:ID="gehört_zu_Kategorie">
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Geld"/>
<Versicherung rdf:ID="Gepaeckversicherung">
    <ist_verwandt_mit rdf:resource="#Gep&#228;eckverlust"/>
</Versicherung>
<Stichwort rdf:ID="Gep&#228;eckverlust"/>
<Synonym rdf:ID="Girokonto"/>
<Stichwort rdf:ID="Glaebiger"/>
<owl:DatatypeProperty rdf:ID="hat_Kurztext">
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hat_Langtext">
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hat_Quiz">
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="hat_Unterkunft">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Unterkunft"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hat_Video">

```

```

    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="hat_Zielort"/>
  <Arbeit rdf:ID="Haushaltshilfe"/>
  <Unterkunft rdf:ID="Herberge"/>
  <Unterkunft rdf:ID="Hotel"/>
  <Reiseort rdf:ID="Inland"/>
  <Stichwort rdf:ID="Internet"/>
  <owl:ObjectProperty rdf:ID="ist_aequivalent_mit">
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
    <rdfs:range rdf:resource="#Synonym"/>
    <owl:inverseOf rdf:resource="#Synonym_verweist_auf"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="ist_verwandt_mit">
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:domain rdf:resource="#K_Reisen_und_Fahren"/>
    <rdfs:range rdf:resource="#Stichwort"/>
    <owl:inverseOf rdf:resource="#Stichwort_verweist_auf"/>
  </owl:ObjectProperty>
  <Stichwort rdf:ID="Job"/>
  <Stichwort rdf:ID="Jod"/>
  <owl:Class rdf:ID="K_Reisen_und_Fahren">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#ist_aequivalent_mit"/>
        <owl:someValuesFrom rdf:resource="#Synonym"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#ist_verwandt_mit"/>
        <owl:someValuesFrom rdf:resource="#Stichwort"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <Stichwort rdf:ID="Kinderbetreuung_im_Ausland">
    <Stichwort_verweist_auf rdf:resource="#Au_Pair"/>
  </Stichwort>
  <Stichwort rdf:ID="kostenlos"/>
  <Stichwort rdf:ID="Kredit"/>
  <Synonym rdf:ID="Kreditkarte"/>
  <Stichwort rdf:ID="Kreditschulden"/>
  <owl:Class rdf:ID="Kurs">
    <rdfs:subClassOf rdf:resource="#Taetigkeit"/>
  </owl:Class>

```

```

<owl:Class rdf:ID="Lemmaverknuepfungen"/>
<owl:ObjectProperty rdf:ID="macht">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Reiseart"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Mail"/>
<owl:ObjectProperty rdf:ID="nimmt_Teil_an">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Kurs"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Ort">
  <rdfs:subClassOf rdf:resource="#K_Reisen_und_Fahren"/>
</owl:Class>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#K_Reisen_und_Fahren"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#reist_ins"/>
      <owl:someValuesFrom rdf:resource="#Ort"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#macht"/>
      <owl:someValuesFrom rdf:resource="#Reiseart"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<Stichwort rdf:ID="Rafting"/>
<Stichwort rdf:ID="Rauchen"/>
<owl:Class rdf:ID="Reiseart">
  <rdfs:subClassOf rdf:resource="#K_Reisen_und_Fahren"/>
</owl:Class>
<K_Reisen_und_Fahren rdf:ID="Reisen_und_Fahren"/>
<owl:Class rdf:ID="Reiseort">
  <rdfs:subClassOf rdf:resource="#Ort"/>
</owl:Class>
<Versicherung rdf:ID="Reiseruecktrittskostenversicherung"/>
<Stichwort rdf:ID="Reiseunfall"/>
<Versicherung rdf:ID="Reiseunfallversicherung">
  <ist_verwandt_mit rdf:resource="#Reiseunfall"/>
  <ist_verwandt_mit rdf:resource="#Urlaubsunfall"/>
</Versicherung>
<owl:ObjectProperty rdf:ID="reist_ins">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Person"/>

```

```

    <rdfs:range rdf:resource="#Reiseort"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="schliesst_ab">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Versicherung"/>
</owl:ObjectProperty>
<Person rdf:ID="Schueler">
    <schliesst_ab rdf:resource="#Reiseunfallversicherung"/>
    <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
    <reist_ins rdf:resource="#Inland"/>
    <hat_Unterkunft rdf:resource="#Gastfamilie"/>
    <macht rdf:resource="#Bildungsreise"/>
</Person>
<Stichwort rdf:ID="Schuldnerberatung"/>
<Kurs rdf:ID="Sprachkurs"/>
<Stichwort rdf:ID="Spurenelemente"/>
<owl:Class rdf:ID="Stichwort">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#Stichwort_verweist_auf"/>
            <owl:someValuesFrom rdf:resource="#K_Reisen_und_Fahren"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Woerter"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="Stichwort_verweist_auf">
    <rdf:type rdf:resource="&owl;SymmetricProperty"/>
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Stichwort"/>
    <rdfs:range rdf:resource="#K_Reisen_und_Fahren"/>
    <owl:inverseOf rdf:resource="#ist_verwandt_mit"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Stoffe"/>
<Stichwort rdf:ID="Stress"/>
<Person rdf:ID="Student">
    <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
    <nimmt_Teil_an rdf:resource="#Sprachkurs"/>
    <reist_ins rdf:resource="#Ausland"/>
    <hat_Unterkunft rdf:resource="#Hotel"/>
    <macht rdf:resource="#Bildungsreise"/>
</Person>
<owl:Class rdf:ID="Synonym">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#Synonym_verweist_auf"/>
            <owl:someValuesFrom rdf:resource="#K_Reisen_und_Fahren"/>
        </owl:Restriction>
    </rdfs:subClassOf>

```

```

    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Woerter"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="Synonym_verweist_auf">
  <rdf:type rdf:resource="#owl:FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Synonym"/>
  <rdfs:range rdf:resource="#K_Reisen_und_Fahren"/>
  <owl:inverseOf rdf:resource="#ist_aequivalent_mit"/>
</owl:ObjectProperty>
<Stichwort rdf:ID="Tabak"/>
<owl:Class rdf:ID="Taetigkeit">
  <rdfs:subClassOf rdf:resource="#K_Reisen_und_Fahren"/>
</owl:Class>
<Kurs rdf:ID="Tauchkurs"/>
<Person rdf:ID="Tourist">
  <schliesst_ab rdf:resource="#Gepaeckversicherung"/>
  <schliesst_ab rdf:resource="#Reiseruecktrittskostenversicherung"/>
  <schliesst_ab rdf:resource="#Reiseunfallversicherung"/>
  <nimmt_Teil_an rdf:resource="#Tauchkurs"/>
  <reist_ins rdf:resource="#Ausland"/>
  <hat_Unterkunft rdf:resource="#Hotel"/>
  <macht rdf:resource="#Urlaubsreise"/>
</Person>
<owl:Class rdf:ID="Unterkunft">
  <rdfs:subClassOf rdf:resource="#Ort"/>
</owl:Class>
<Stichwort rdf:ID="Urheberrecht"/>
<Reiseart rdf:ID="Urlaubsreise"/>
<Stichwort rdf:ID="Urlaubsunfall"/>
<owl:Class rdf:ID="Versicherung">
  <rdfs:subClassOf rdf:resource="#K_Reisen_und_Fahren"/>
</owl:Class>
<Stichwort rdf:ID="Vertrag"/>
<owl:Class rdf:ID="Woerter">
  <rdfs:subClassOf rdf:resource="#Lemmaverknuepfungen"/>
</owl:Class>
<Stichwort rdf:ID="Zigarette"/>
</rdf:RDF>

```

### **Eidesstattliche Versicherung**

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangabe kenntlich gemacht.

Ort, Datum

Unterschrift