

UNIVERSITÄT LEIPZIG

Seminararbeit

**Einführung  
Ontologiemanagement**

vorgelegt von  
Johannes Schmidt

**Betreuer**

Dr. Dieter Sosna

Institut für Informatik  
Universität Leipzig

Leipzig, 03. Januar 2009



# Inhaltsverzeichnis

<b>1</b>	<b>Ontologie und Ontologien</b>	<b>5</b>
1.1	Begriffsdefinition . . . . .	5
1.1.1	Ontologie in der Philosophie . . . . .	5
1.1.2	Ontologie in der Informatik . . . . .	6
1.2	Einteilung von Ontologien . . . . .	8
1.2.1	Unterscheidung nach der Allgemeingültigkeit . . . . .	8
1.2.2	Unterscheidung nach dem Formalisierungsgrad . . . . .	8
1.2.3	Unterscheidung nach der Komplexität . . . . .	9
1.3	Verwendung von Ontologien . . . . .	10
1.4	Repräsentationssprachen . . . . .	11
1.5	Abfragesprachen . . . . .	14
<b>2</b>	<b>Ontologiemanagement</b>	<b>17</b>
2.1	Entwicklungsmethoden von Ontologien . . . . .	17
2.2	Integration von Ontologien . . . . .	18
2.2.1	Kombination verschiedener Ontologien . . . . .	18
2.2.2	Versionierung . . . . .	20
2.2.3	Weitere Herausforderungen . . . . .	20
<b>3</b>	<b>Fazit</b>	<b>23</b>
	<b>Literaturverzeichnis</b>	<b>25</b>



# 1 Ontologie und Ontologien

*Der Begriff der Ontologie wurde in der Philosophie geprägt. Er setzt sich aus den griechischen Wörtern „on“ (seiend) und „logos“ (Wort, Lehre) zusammen und beschreibt die Wissenschaft des Seins. In diesem Kapitel soll neben einem kurzen historischen Diskurs zur Begriffsentwicklung in der Philosophie ein Überblick über die Begriffsdefinition(en) aus dem Bereich der Informatik gegeben werden. In den Abschnitten 1.2 und 1.3 werden unterschiedliche Anwendungsmöglichkeiten und Klassifizierungskriterien von Ontologien beschrieben. Im Anschluss soll ein kurzer Überblick über die wichtigsten Repräsentations- und Abfragesprachen gegeben werden.*

## 1.1 Begriffsdefinition

### 1.1.1 Ontologie in der Philosophie

In der Philosophie wird die Ontologie als Teildisziplin der Metaphysik eingeordnet und untersucht „Das Sein als solches“. Sie beschreibt die allgemeinsten Strukturen und definiert Unterscheidungen im Wirklichen und Nichtwirklichen. Für diese Untersuchung bedient sich der Mensch den Mitteln der Sprache (vgl. Meixner 2004).

Der Begriff Ontologie wurde wahrscheinlich im 17. Jahrhundert durch den Philosophen Goclenius (1547–1628) geprägt. Bereits Aristotels (384–322 v. Chr.) untersuchte den Aufbau der Welt wissenschaftlich. Dabei standen nicht die Gründe oder die Funktion des Seienden im Vordergrund der Betrachtungen, sondern nur dessen Beschreibung und Einteilung. Dabei unterschied er zwischen *Substanzen* (Lebewesen, Steine, bewegliche Objekte) und *Akzidenzien* (Eigenschaften von Substanzen wie z.B. Laufen oder Wärme). Platon (427–347 v. Chr.) erweiterte diesen Ansatz und definierte das sog. *Reich der Ideen*. Eigenschaften sind demnach zeitlos und können auf Materie angewendet werden. Erst dadurch würden individuelle Dinge entstehen, die jedoch zeitlich sind. E. Husserl (1859–1938) unterschied im 20. Jahrhundert zwischen der „formalen“ und den „regionalen“ Ontologien.

Erstere untersucht allgemeine Eigenschaften des Seienden mit Hilfe der „reinen Logik“, wobei letztere sich auf die unterschiedlichen Arten des Seienden bezieht.

Ein Problem bei der Verwendung von Ontologien ist, dass man stets angeben muss, auf welche Ontologie man sich bezieht. Ohne eine solche Angabe können Missverständnisse bei der Beschreibung einer Theorie entstehen. Des Weiteren wies R. Carnap (1891–1970) darauf hin, dass Angaben zur Existenz nur innerhalb eines sprachlichen Rahmens möglich sind (vgl. Rehfuß 2003, Seite 514).

Weitere Informationen zur historischen Entwicklungsgeschichte des Ontologiebegriffs können (Meixner 2004), (Smith 2000) oder (Rehfuß 2003) entnommen werden.

### 1.1.2 Ontologie in der Informatik

Die wohl wichtigste Definition des Begriffes Ontologie wurde von Thomas Gruber geprägt:

*„An ontology is an explicit specification of a conceptualization.“* (Gruber 1993)

Der Begriff der „conceptualization“ ist als ein abstraktes Modell einer Domäne zu verstehen. Jedes Element des verwendeten Vokabulars der Ontologie muss eineindeutig, also exakt, definiert sein.

Das obige Zitat wird in (Breitmann u. a. 2007, Seite 1) jedoch leicht abgewandelt zitiert:

*„An ontology is a **formal**, explicit specification of a **shared** conceptualization.“*

Die Definition wurde um den Begriff „formal“ erweitert. Aufgrund der formalen Beschreibung ist sichergestellt, dass die Ontologie mit Hilfe eines Computers verstanden und verarbeitet werden kann. Der Begriff „shared“ bezieht sich auf einen gemeinsamen Konsens der abzubildenden Domäne. Eine Ontologiedefinition ist nicht in der Lage, die Realität exakt abzubilden, sondern stellt lediglich eine Annäherung bzw. einen Kompromiss dar (vgl. Hepp 2008, Seite 4). Die Anzahl der Unterstützer einer Ontologie spielt dabei eine wichtige Rolle. In der Literatur wird noch auf weitere Definitionen verwiesen (siehe Hepp 2008; Breitmann u. a. 2007). An dieser Stelle soll jedoch nicht auf deren Gemeinsamkeiten bzw. Unterschiede eingegangen werden.

Wie bereits angedeutet, existiert nicht *die eine* Ontologie. Für jede Anwendungsdomäne können verschiedene spezialisierte Ontologien definiert werden. Darum ist es auch sinnvoll, eine Unterscheidung zwischen der wissenschaftlichen Disziplin Ontologie und den spezifischen Ontologien zu treffen. Es lassen sich jedoch generelle Bestandteile für Ontologien herausarbeiten. Ein Begriff des verwendeten Vokabulars wird als Konzept (oder auch Klasse) bezeichnet. Die Konzepte werden typischerweise als Taxonomie (siehe Abschnitt 1.2.3) organisiert, d.h. sie können als Hierarchie angeordnet werden. Konkrete

Ausprägungen eines Konzeptes werden als Instanzen bezeichnet. Beispielsweise ist *Picasso* eine Instanz des Konzeptes *Maler*. Mit Hilfe von Relationen können nicht-hierarchische Beziehungen zwischen zwei oder mehreren Konzepten definiert werden. Dadurch können Netzwerke von Relationen entstehen. Jedes Konzept kann mit Hilfe von Attributen näher qualifiziert werden. Attribute stellen Relationen zwischen Konzepten und einem definierten Datentyp dar. Des Weiteren können Regeln und Einschränkungen innerhalb einer Ontologie definiert werden. Regeln, die immer gültig sind, werden Axiome genannt (vgl. Wichmann 2007, Seite 46f).

Der Ontologiebegriff kann auch streng formal definiert werden. In (Yildiz 2006) wird bspw. ein 7-Tupel zur Beschreibung verwendet. Diese Arbeit bezieht sich jedoch auf das Quintupel  $\mathcal{O} = \{\mathcal{C}, \mathcal{R}, \mathcal{CH}, rel, \mathcal{OA}\}$  nach A. Maedche (Breitmann u. a. 2007, Seite 19), wobei gilt:

- $\mathcal{C}$  ist die Menge aller Konzepte.
- $\mathcal{R}$  ist die Menge aller Relationen.
- $\mathcal{CH} \subseteq \mathcal{C} \times \mathcal{C}$  definiert ein Hierarchie von Konzepten oder eine Taxonomie, wobei  $\mathcal{CH}(\mathcal{C}_1, \mathcal{C}_2)$  angibt, dass  $\mathcal{C}_1$  ein Unterkonzept von  $\mathcal{C}_2$  ist.
- $rel: \mathcal{R} \rightarrow \mathcal{C} \times \mathcal{C}$  ist eine Funktion, die Relationen außerhalb der Hierarchie der Taxonomie zwischen Konzepten definiert.
- $\mathcal{OA}$  beschreibt eine Menge von Axiomen, die in Form der Aussagenlogik beschrieben werden.

Ontologien müssen nicht zwingend Relationen definieren. Laut (Lassila und McGuinness 2001) kann auch ein Glossar als Ontologie bezeichnet werden. In (Hepp 2008) wird darauf verwiesen, dass einige Wissenschaftler die Existenz von Relationen jedoch ausdrücklich vorsehen.

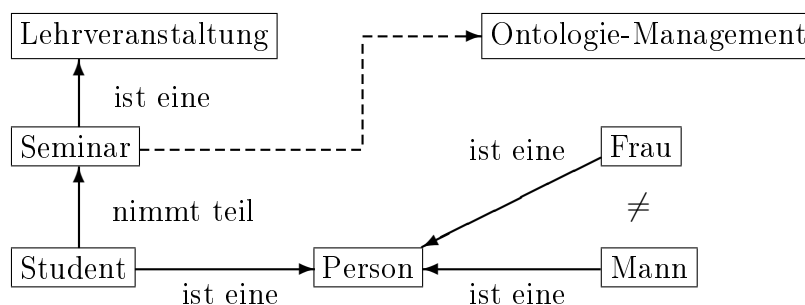


Abbildung 1.1: Seminar Ontologie

In Abbildung 1.1 wird eine sehr einfache Ontologie graphisch dargestellt. „Ontologie-Management“ ist eine Instanz des Konzeptes „Seminar“. An einem Seminar können Studenten teilnehmen. Ein Student kann männlich oder weiblich sein jedoch nicht beides gleichzeitig. Es wäre leicht möglich, die Ontologie zu konkretisieren, indem bspw. die Einschränkung definiert wird, dass mindestens fünf Studenten an einem Seminar teilnehmen müssen.

## 1.2 Einteilung von Ontologien

Ontologien können nach verschiedenen Gesichtspunkten eingeteilt werden. In der Literatur werden Unterscheidungen hinsichtlich ihrer Allgemeingültigkeit, ihrem Strukturierungsgrad und ihrer Komplexität vorgenommen.

### 1.2.1 Unterscheidung nach der Allgemeingültigkeit

Die sog. *Upper Level Ontologien* beschreiben ganz allgemeine Konzepte wie z.B. Zeit und Raum. Sie können als Basis für die Entwicklung domänenspezifischer Ontologien genutzt werden. Die *Domain Ontologien* müssen als Spezialisierung der Upper Level Ontologien verstanden werden. Sie beschreiben ebenfalls allgemeine Konzepte, jedoch werden diese auf eine allgemeine Anwendungsdomäne bezogen. Die *Task Ontologien* bauen ebenfalls auf den Upper Level Ontologien auf. Sie beschreiben Vokabularien, die zur Durchführung von allgemeinen Aufgaben notwendig sind. Die *Application Ontologien* können Domain und/oder Task Ontologien spezialisieren. Dabei wird das Vokabular auf die Verwendung innerhalb einer spezifischen Anwendungsdomäne reduziert (Guarino 1998).

### 1.2.2 Unterscheidung nach dem Formalisierungsgrad

Ontologien können mehr oder weniger formal beschrieben werden. Für die Kommunikation zwischen Menschen ist eine *informale* Definition in einer natürlichen Sprache ausreichend. Maschinen sind jedoch nicht in der Lage, informal definierte Strukturen zu verarbeiten. Aus diesem Grund muss die Sprache eingeschränkt und strukturiert werden. Man spricht dann von *semi-informalen* Sprachen. Die nächste Gruppe bilden die *semi-formalen* Sprachen. An Stelle der natürlichen Sprache wird eine formal definierte Sprache verwendet. Den höchsten Formalisierungsgrad bietet die Verwendung einer *streng formalen* Sprache (vgl. Yildiz 2006, Seite 4). Für formale Sprachen müssen bspw. Korrektheit und Vollständigkeit nachgewiesen werden.



### 1.2.3 Unterscheidung nach der Komplexität

In Abbildung 1.2 sind die einzelnen Ontologierepräsentationen dargestellt, wobei die Komplexität von links nach rechts zunimmt. Es wird zwischen leichtgewichtigen und schwergewichtigen Ontologien unterschieden. Die nachfolgenden Definitionen beziehen sich auf (Lassila und McGuinness 2001).

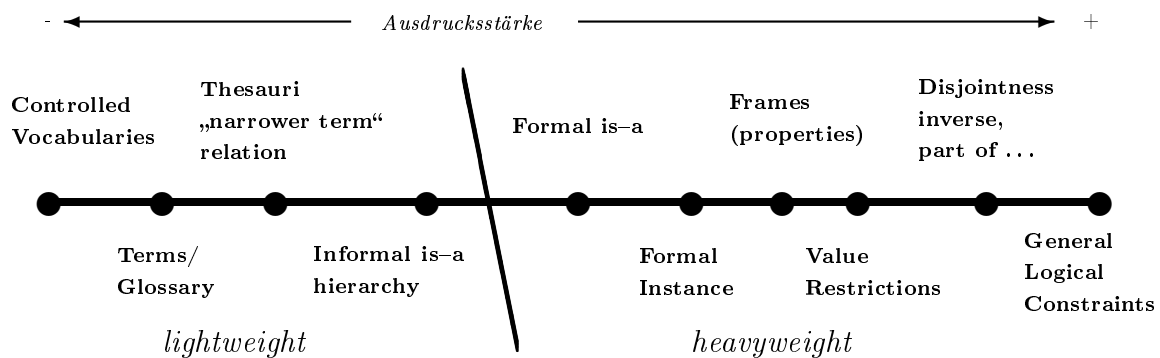


Abbildung 1.2: Ontologie Spektrum in Anlehnung an (Breitmann u. a. 2007, Fig 2.7, Seite 26)

[Original in (Lassila und McGuinness 2001)]

Die einfachste Form der Wissensstrukturierung stellt das *Vokabular* dar. Es werden endlich viele Begriffe als Liste angeordnet. Die Definition von Synonymen ist möglich, jedoch sollte ein Begriff als bevorzugt gekennzeichnet werden. Das *Glossar* erweitert die Auflistung der Begriffe um eine informelle Beschreibungen in natürlicher Sprache.

Ein *Thesaurus* kann als eine Erweiterung eines Vokabulars verstanden werden. Die Begriffe werden mit Hilfe von Relationen verknüpft. Häufig verwendete Relationstypen sind Äquivalenz, Homographie oder Assoziation.

Wenn ein Vokabular oder ein Glossar um eine hierarchische Struktur erweitert wird, so spricht man von einer *Taxonomie*. Die Begriffe werden mit Hilfe von Vererbungsbeziehungen (is-a) in Kategorien eingeordnet. Dabei wird zwischen informellen und formellen Taxonomien unterschieden. Im Unterschied zur informellen Taxonomie muss bei der formellen Taxonomie die Hierarchie explizit eingehalten werden (Transition). Wird eine Taxonomie um konkrete Ausprägungen eines Konzeptes erweitert, wird dies als *formal instance* bezeichnet.

In sog. *Frames* werden die Konzepte um vererbare Eigenschaften (Slots) erweitert. Ausgehend von Abbildung 1.1 könnte das Konzept Student um die Eigenschaft „Name“ erweitert werden. Sollen die Eigenschaften hinsichtlich ihres Datentyps oder ihrer Kardinalität eingeschränkt werden, spricht man von *value restrictions*.

Um komplexere Zusammenhänge zwischen den Konzepten darzustellen, können logische Einschränkungen definiert werden. Dies kann von einfachen Definitionen von disjunkten Konzepten bis hin zur Einschränkung eines Wertebereichs für eine Eigenschaft erfolgen.

## 1.3 Verwendung von Ontologien

Mit Hilfe einer Ontologie kann domänenspezifisches Wissen durch die Beschreibung von Konzepten und deren Beziehungen zueinander abgebildet werden. Daraus lassen sich drei Anwendungsbereiche ableiten:

1. **Kommunikation/Interaktion:** Dabei sind sowohl die Kommunikationswege Mensch–zu–Mensch, Mensch–zu–Maschine als auch Computer–zu–Maschine (Ontologie als Austauschformat) inbegriffen.
2. **Maschinelles Folgern:** Der Computer wird in die Lage versetzt, die Daten zu „verstehen“ und die Relationen zwischen den Instanzen auszuwerten.
3. **Wiederverwendung von Wissen**

Die Verwendung einer Ontologie kann auch bei der Wissensakquise von Vorteil sein. So ist es möglich, schnell und zuverlässig neues Wissen zu beschreiben. Durch den gezielten Einsatz von Softwarewerkzeugen können die Daten auf Inkonsistenzen gegen die Ontologiedefinition geprüft werden. Des Weiteren kann die Ontologie als Index einer Wissensbasis verstanden werden, wodurch die Suche unterstützt werden kann (vgl. Uschold und Jasper 1999, Seite 11–2f).

Ontologien finden in den verschiedensten Disziplinen der Informatik Anwendung z.B. in der Künstlichen Intelligenz, dem Software Engineering, dem Wissensmanagement oder dem Semantic Web. Es existieren bereits zahlreiche (z.T. frei verfügbare) Ontologiedefinitionen wie z.B. die Gene Ontology<sup>1</sup>, die Enterprise Ontology<sup>2</sup> oder Onto-Med<sup>3</sup>. In diesem Abschnitt sollen nur zwei exemplarische Anwendungen bzw. Definitionen von Ontologien beschrieben werden.

Das **Semantic MediaWiki**<sup>4</sup> ist eine Erweiterung für MediaWiki. Der Ersteller eines Wiki-Artikels kann mit Hilfe von Attributen Metainformationen anlegen oder sog. typisierte Verweise (Relationen) auf andere Wiki-Artikel anlegen. Für weitere Informationen sei auf die Webseite des Projektes verwiesen.

<sup>1</sup><http://www.geneontology.org/GO.downloads.ontology.shtml>

<sup>2</sup><http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>

<sup>3</sup><http://www.onto-med.de/index.jsp>

<sup>4</sup><http://semantic-mediawiki.org>

Das **Open Biomedical Ontologies**<sup>5</sup> Projekt hat sich zum Ziel gesetzt, verschiedene Ontologien im Bereich der Bio-Medizin zu entwickeln. Es existieren Ontologien für bspw. Anatomie oder Biochemie. Diese sind in verschiedenen Repräsentationssprachen (siehe nachfolgenden Abschnitt) als Download erhältlich.

## 1.4 Repräsentationssprachen

Für die Spezifikation einer Ontologie stehen eine Reihe von Beschreibungssprachen zur Auswahl. Es wird zwischen den traditionellen und den webbasierten — also für den Einsatz im Internet vorgesehenen — Repräsentationssprachen unterschieden. In (Dittmann 2007) wurden zahlreiche Sprachen zusammengetragen und ausführlich erläutert. In diesem Kapitel sollen nur die wichtigsten kurz vorgestellt werden.

### Traditionelle Sprachen

Die traditionellen Beschreibungssprachen basieren unter anderem auf der Prädikaten- bzw. Beschreibungslogik. Als Vertreter dieser Sprachgruppierung sollen LOOM<sup>6</sup> und CycL<sup>7</sup> genannt werden. LOOM wurde am Institut für Informationssysteme der Universität South California entwickelt. Mit Hilfe von Beschreibungslogiken und Produktionsregeln können unter anderem Relationen, Funktionen und Regeln definiert werden.

### Webbasierte Sprachen

Die Extensible Markup Language (XML) wurde vom World Wide Web Consortium (W3C) definiert<sup>8</sup> und wird in zahlreichen Internetanwendungen zum Datenaustausch von semi-strukturierten Daten verwendet. Mit Hilfe von XML-Schema-Definitionen (XSD) ist es möglich, die Struktur eines XML Dokumentes zu definieren. XSD Dokumente werden in XML beschrieben. Der Entwickler kann aus einer großen Anzahl von vordefinierten Datentypen wählen oder eigene Typen erzeugen. Des Weiteren können Einschränkungen auf Datentypen und Kardinalitäten definiert werden. Die Korrektheit eines XML Dokumentes wird gegen die XSD validiert. Weitere Informationen zu XML und XSD können auf die Webseite des W3C verwiesen. Die folgende XSD Definition soll beispielhaft die Konzepte Student und Seminar aus dem obigen Beispiel 1.1 beschreiben.

---

<sup>5</sup><http://www.obofoundry.org/>

<sup>6</sup><http://www.isi.edu/isd/LOOM/>

<sup>7</sup><http://www.opencyc.org/doc/>

<sup>8</sup><http://www.w3.org/XML/>

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="StudentTyp">
    <xs:sequence>
      <xs:element name="Vorname" type="xs:string"/>
      <xs:element name="Nachname" type="xs:string"/>
      <xs:element minOccurs="1" name="Seminar" type="SeminarTyp"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SeminarTyp">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Listing 1.1: XML Schema Definition

Auf Basis von XML können anwendungsspezifische Sprachen abgeleitet werden. Für die Modellierung von Ontologien wird häufig das Resource Description Framework<sup>9</sup> (RDF) eingesetzt. Ähnlich wie bei der Verwendung der XSD für XML wird mittels RDF-Schema (RDFS) die Struktur der in RDF modellierten Daten definiert. Dabei können Klassen und deren Eigenschaften definiert werden. RDF Klassen sind mit Hilfe von Uniform Resource Identifiers (URI) eindeutig identifizierbar. Das folgende RDF Schema<sup>10</sup> beschreibt die Konzepte Person und Seminar und deren Properties gemäß der Abbildung 1.1.

```

<rdf:RDF xml:lang="de"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Person">
    <rdfs:comment>Definition einer Person</rdfs:comment>
    <rdfs:subClassOf
      rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Student">
    <rdfs:comment>Ein Student an einer Universität</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Person"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Seminar">
    <rdfs:comment>Seminar</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
  </rdfs:Class>

  <rdf:Property rdf:ID="Name">
    <rdfs:comment>Name eines Seminars</rdfs:comment>
    <rdfs:domain rdf:resource="#Seminar"/>
    <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Literal"/>
  </rdf:Property>

```

<sup>9</sup><http://www.w3.org/RDF/>

<sup>10</sup>basierend auf <http://www.cs.rpi.edu/~puninj/XMLJ/classes/class8/slide35-1.html>

```

<rdf:Property rdf:ID="Nachname">
  <rdfs:comment>Nachname einer Person</rdfs:comment>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Literal"/>
</rdf:Property>
</rdf:RDF>

```

Listing 1.2: RDF Schema Definition

Als letzter Vertreter der webbasierten Beschreibungssprachen soll die Web Ontology Language<sup>11</sup> (OWL) genannt werden. Die Spezifikation wurde 2004 vom W3C freigegeben. OWL basiert auf RDFS und DAML+OIL, wobei sie sich durch ihre starke Ausdrucksmächtigkeit auszeichnet. Es wurden drei Versionen der OWL definiert: OWL Lite, OWL DL und OWL Full, wobei gilt  $\text{OWL Lite} \subseteq \text{OWL DL} \subseteq \text{OWL Full}$ . Die einzelnen Sprachvarianten unterscheiden sich hinsichtlich ihrer Ausdrucksmöglichkeiten. Weitere Informationen zum Aufbau der OWL kann der Spezifikation des W3C entnommen werden.

Das nachfolgende Beispiel zeigt eine OWL Definition der Konzepte Student, Person, Mann und Frau wie sie in Abbildung 1.1 dargestellt sind. Das Beispiel wurde mit Hilfe des Ontologie-Editors Protégé<sup>12</sup> erzeugt.

```

<?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 owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY example "http://www.uni-leipzig.de/ontology/example#" >
]>
<rdf:RDF xmlns="http://www.uni-leipzig.de/ontology/example#"
  xml:base="http://www.uni-leipzig.de/ontology/example#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:example="http://www.uni-leipzig.de/ontology/example#">
  <owl:Ontology rdf:about="" />

  <!-- http://www.uni-leipzig.de/ontology/example#Vorname -->
  <owl:DatatypeProperty rdf:about="#Vorname" />

  <!-- http://www.uni-leipzig.de/ontology/example#Frau -->
  <owl:Class rdf:about="#Frau">
    <owl:disjointWith rdf:resource="#Mann" />
  </owl:Class>

  <!-- http://www.uni-leipzig.de/ontology/example#Mann -->

```

<sup>11</sup><http://www.w3.org/TR/owl-semantic/>

<sup>12</sup><http://protege.stanford.edu/>

```

<owl:Class rdf:about="#Mann"/>

<!-- http://www.uni-leipzig.de/ontology/example#Person -->
<owl:Class rdf:about="#Person"/>

<!-- http://www.uni-leipzig.de/ontology/example#Student -->
<owl:Class rdf:about="#Student">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Vorname"/>
      <owl:someValuesFrom rdf:resource="&xsd:string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
</rdf:RDF>

```

Listing 1.3: OWL Beispiel

## 1.5 Abfragesprachen

Abfragesprachen für Ontologien müssen unterschiedliche Kriterien erfüllen. Sie müssen in der Lage sein, die komplette Datenmodelldefinition der Ontologie zu erfassen. Das heißt beispielsweise, dass bei einer Abfrage auf ein Konzept auch die von ihr abgeleiteten Konzepte berücksichtigt werden müssen. Weitere Anforderungen sind Filter, arithmetische Operationen, eingebettete Abfragen oder Aggregat- und Gruppierungsunterstützung. Eine detaillierte Liste von möglichen Anforderungen an Abfragesprachen kann (Magkanaraki u. a. 2002) entnommen werden.

Es existieren zahlreiche Implementierungen für Abfragesprachen. In den Veröffentlichungen (Magkanaraki u. a. 2002, Seite 3ff) und (Wichmann 2007, Seite 61ff) werden die wichtigsten Vertreter übersichtlich beschrieben. In dieser Arbeit soll nur auf eine Abfragesprache exemplarisch eingegangen werden.

Die RDF Query Language<sup>13</sup> (RQL) ist eine typisierte und deklarative Abfragesprache für RDF. Die Sprachdefinition enthält bereits zahlreiche vordefinierte Funktionen und Iteratoren. Die Syntax erinnert stark an die Structured Query Language (SQL) und bietet bspw. dem Anwender die Möglichkeit, neben den oben beschriebenen Funktionen bspw. Joins oder Unions zu verwenden. Eine Besonderheit an RQL ist, dass neben den eigentlichen RDF-Daten-Abfragen auch Abfragen zur Struktur der Ontologiedefinition möglich sind (Magkanaraki u. a. 2002, Seite 3).

<sup>13</sup><http://139.91.183.30:9090/RDF/RQL/>

Als Beispiel für eine RQL Abfrage wurde die Query 5 aus (Karvounarakis u. a. 2002) ausgewählt. Die Anfrage bezieht sich auf eine Beispielontologie für Museen. Es werden alle Museumsdatensätze zurückgegeben, die nach dem Jahre 2000 geändert wurden.

```
select X, Y  
from Museum{X}.last_modified {Y}  
where Y >= 2000-01-01
```

Listing 1.4: Eine einfache RQL Abfrage





## 2 Ontologiemanagement

*Das vorliegende Kapitel befasst sich mit der Erstellung und Verwendung von Ontologien. Dabei wird der Schwerpunkt der Betrachtungen auf die Herausforderungen des Managements gelegt.*

### 2.1 Entwicklungsmethoden von Ontologien

Die Entwicklung einer Ontologie ähnelt der Entwicklung von Software sehr. Dabei haben sich verschiedene Vorgehensmodelle etabliert, wobei sich bisher keines als Standard durchsetzen konnte. Es lassen sich jedoch allgemeine Phasen der Entwicklung herausarbeiten.

1. **Erfassen der Begriffe:** Dabei muss die Definition der Konzepte durch Domänenexperten festgelegt und überprüft werden.
2. **Implementierung**
3. **Evaluierung:** Das entwickelte System muss auf Fehler untersucht werden.

G. Wichmann beschreibt und vergleicht in (Wichmann 2007, Seite 39ff) verschiedene Entwicklungsansätze. Dabei wird besonders auf die fünf Entwicklungsphasen des On-To-Knowledge Projektes eingegangen (siehe auch Yildiz 2006).

Ontologien unterliegen einem Lebenszyklus. Das heißt, dass bei der Entwicklung bestimmte Entwicklungsphasen iteriert werden. Durch die nachträgliche Anpassung der Ontologiedefinition müssen die Anforderungen an das Versionsmanagement (siehe Abschnitt 2.2.2) beachtet werden.

Durch den Einsatz verschiedener Entwicklungswerkzeuge wie z.B. OntoEdit/OntoStudio<sup>1</sup> oder Protégé<sup>2</sup> kann die Entwicklung unterstützt und die Integration verschiedener Ontologien vereinfacht werden. Dabei müssen die Bedürfnisse von großen bzw. internationalen Teams berücksichtigt werden.

---

<sup>1</sup><http://www.ontoprise.de/deutsch/start/produkte/ontostudio/>

<sup>2</sup><http://protege.stanford.edu>

## 2.2 Integration von Ontologien

Bei der Verwendung von Ontologien treten zum Teil komplexe Probleme auf. Es ist beispielsweise zu klären, ob und wie eine Unterscheidung spezifischer Versionsstände einer Ontologie getroffen werden kann. Was muss beachtet werden, wenn verschiedene Ontologien innerhalb einer Anwendung verwendet werden sollen? Wie kann man Ontologien effizient speichern und abfragen? Was für Probleme können bei der Integration auftreten?

### 2.2.1 Kombination verschiedener Ontologien

Bei der Entwicklung von Anwendungen kann es vorkommen, dass verschiedene Ontologien verwendet werden sollen. In (Ding u. a. 2001) werden verschiedene Ansätze zur Integration mehrerer Ontologien beschrieben. Nachfolgend sollen nur die wichtigsten Unterscheidungen genannt werden.

**Vereinigung (merging):** Aus mehreren Ontologien, die sich in Teilen in ihrer Definition überlappen, wird eine neue Ontologie definiert.

**Zuordnung (mapping):** Definieren verschiedene Ontologien ähnliche Konzepte, so kann man diese mit Hilfe von (Gleichheits-)Relationen aufeinander abbilden.

Bei der Integration mehrerer Ontologien können verschiedene technische Probleme auftreten, die in der Literatur in die Kategorien Sprach- und Modellunterschiede (*ontology level mismatch* und *language level mismatch*) unterteilt werden. In Abbildung 2.1 sind die verschiedenen Unterteilungen als Grafik dargestellt. In (Passin 2004) wird das Beispiel eines Agenten für Buchungssysteme angeführt, der sowohl Flug- als auch Hotelreservierungen unterstützen soll. Die nachfolgenden Abschnitte greifen diesen Anwendungsfall auf, um die möglichen Integrationsprobleme zu verdeutlichen.

#### Sprachunterschiede

Bei der Implementierung des oben genannten Buchungsagenten muss auf die Implementierungssprache der zu verwendenden Ontologien geachtet werden. Wurden unterschiedliche Sprachen zur Definition der Ontologien verwendet (*syntax mismatch*), muss für eine Ontologiedefinition — sofern dies möglich ist — eine syntaktische Transformation durchgeführt werden. Es ist jedoch auch möglich, dass sich die Sprachen in Bezug auf ihre Ausdruckslogik unterscheiden (*logical mismatch*). In diesem Fall muss eine Transformation der Ausdrücke erfolgen. Sollten sich die Sprachen hinsichtlich ihrer Ausdrucksstärke (*expressivity*) unterscheiden, wird die Integration der Ontologien sehr viel komplexer.

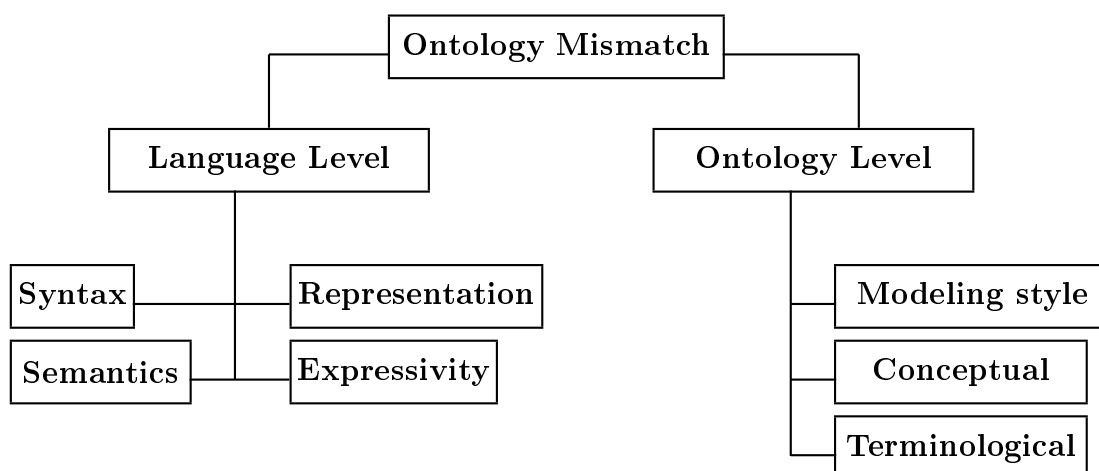


Abbildung 2.1: Verständnisprobleme für Ontologien frei nach (Ding u. a. 2001, Seite 6ff)

Wird beispielsweise nur in der Fluglinien-Ontologie die logische Negation unterstützt, ist eine Umwandlung in die Syntax der Sprache für die Hotelreservierung nur mit erhöhtem Aufwand möglich. Dies gilt genau dann, wenn sich die Beschreibungssprachen semantisch unterscheiden (*semantic mismatch*), wobei die Ausdruckslogiken sich nicht unterscheiden, d.h. die logische Ausdruckstärke der verwendeten Sprachen ist gleich hoch.

### Modellunterschiede

Für die Implementierung des Buchungsagenten wurde festgelegt, dass die beiden zu verwendenden Ontologien in der selben Sprache vorliegen müssen. Somit können Sprachunterschiede generell ausgeschlossen werden.

Werden mehrere Ontologien verwendet, kann es zu sog. Überlappungen der zu beschreibenden Domänen kommen. In der Ontologie der Fluglinien ist eine Beschränkung definiert, dass eine Person sich anhand der Personalausweisnummer identifizieren lassen kann. Die Hotel-Ontologie sieht dieses jedoch nicht vor. In beiden Ontologien wird eine Person definiert, jedoch unterscheidet sich das Konzept zur Person. Des Weiteren könnten unterschiedliche (sich widersprechende) Relationen zur Person definiert werden.

Ein weiteres Problem liegt vor, wenn innerhalb der Ontologiedefinitionen unterschiedliche Synonyme verwendet werden. Eine Person ist aus Sicht einer Fluggesellschaft ein Passagier jedoch für ein Hotel ein Gast. Die Komplexität dieses Problems nimmt zu, wenn die Ontologiedefinitionen in unterschiedlichen natürlichen Sprachen definiert werden. Diese Probleme lassen sich durch die Verwendung eines Thesaurus lösen, jedoch muss stets

sichergestellt sein, dass die Synonymzuordnungen durch einen Fachmann der spezifischen Domäne geprüft werden.

Neben den Synonymen gilt es, das Problem der Homonyme näher zu untersuchen. Beispielsweise ist das Wort Ente in den Domänen Tierwelt und Fahrzeuge unterschiedlich definiert.

Zum Schluss soll auf die Kodierung der Informationen in den Ontologien hingewiesen werden. Die Fluggesellschaften kodieren Zeitangaben immer in einer amerikanischen Notation in Universal Time Coordinated (UTC), wobei die deutschen Hotels ihre Zeiten in lokaler Zeit speichern. Die Lösung dieses Problems ist jedoch technisch leicht zu verwirklichen.

### 2.2.2 Versionierung

Ontologien müssen per Definition exakt und aktuell sein. Daraus folgt, dass sie regelmäßig angepasst werden müssen, um den aktuellen Stand des gemeinsamen Kompromisses darzustellen (siehe Kapitel 1.1.2). Folgende Gründe für eine Änderungen sind denkbar (vgl. Yildiz 2006, Seite 8):

- Änderungen von Relationen oder Konzepten
- Hinzufügen/Entfernen von Relationen oder Konzepten
- Änderung der Repräsentation der Ontologie (z.B. Implementierung in einer anderen Repräsentationssprache)

Änderungen der Ontologie können Inkompatibilitäten auslösen oder sogar das Ergebnis von Suchanfragen beeinflussen. Zur Lösung der Versionsprobleme könnten die verschiedenen Versionen der Ontologie archiviert und somit parallel verwendet werden. Es muss dokumentiert werden, welche Auswirkungen eine Änderung der Ontologie auf die Vorgängerversion hat. Nach (Ding u. a. 2001) sollen Versionen beispielsweise als kompatibel, rückwärtskompatibel bzw. inkompatibel gekennzeichnet und dokumentiert werden.

Des Weiteren wäre es denkbar, eine bestimmte Version einer Ontologie nur bis zu einem bestimmten Zeitpunkt zu unterstützen. Zur Zeit existiert kein ausgereiftes Framework bzw. kein allgemeines Vorgehen für das Management von Ontologieversionen.

### 2.2.3 Weitere Herausforderungen

Bei der (Wieder)Verwendung von Ontologien sind neben den bereits beschriebenen Herausforderungen noch weitere zu betrachten. In (Ding u. a. 2001) wird die Speicherung und Verwaltung von Ontologien in sog. *Ontology Library Systems* näher beschrieben. Neben

---

den Kernanforderungen wie Persistenz und die eindeutige Identifikation von Ontologien spielen weiche Anforderungen wie bspw. die effiziente Implementierung von Suchanfragen in großen Ontologien oder der einfache Zugriff auf eine solche Ontology Library eine wichtige Rolle.

Im Umfeld des Semantic Web ist eine zentrale Speicherung von Ontologien nicht möglich, wodurch sich andere Anforderungen an das Ontologiemanagement ergeben.



## 3 Fazit

Ontologien bilden einen großen Schwerpunkt in der wissenschaftlichen Forschung und finden verstärkt Anwendung in neueren Implementierungen. Die Möglichkeit mit Hilfe von Agenten oder Reasonern aus einem Datenbestand Schlüsse zu ziehen, bieten in zahlreichen Bereichen der Informatik neue Perspektiven. Es existieren bereits einige wirtschaftlich relevante Anwendungen (siehe Hepp u. a. 2008, Kapitel 8ff). Das Semantic MediaWiki zeigt, dass das Thema Ontologie auch einer großen Anwendergemeinde nahegebracht werden kann.

Ontologien dürfen jedoch nicht als Lösung für alle Probleme verstanden werden. Hepp weist in (Hepp 2008, Seite 10f) explizit darauf hin, dass für manche Probleme die Erstellung und Wartung einer Ontologie höhere Kosten verursachen kann, als eine Lösung ohne den Einsatz einer Ontologie. Des Weiteren ist der Einsatz einer ontologiebasierten Implementierung nicht sinnvoll, wenn Altsysteme integriert werden müssen, bei denen die Daten unstrukturiert gespeichert werden. Außerdem gibt es zur Zeit kaum Anwendungen, die in der Lage sind, große Ontologien bzw. eine große Anzahl an Instanzen performant zu verarbeiten.





# Literaturverzeichnis

- Breitmann u. a. 2007** BREITMANN, Karin K. ; CASANOVA, Marco A. ; TRUSZKOWSKI, Walter: *Semantic Web: Concepts, Technologies and Applications*. Springer-Verlag, 2007. – 18–34 S. – ISBN 978-1-84628-581-3
- Ding u. a. 2001** DING, Ying ; FENSEL, Dieter ; KLEIN, Michel ; OMELAYENKO, Borys: *Ontology Management: Survey, Requirements and Directions*. 2001. – URL <http://www.ontoknowledge.org/download/del4.pdf>. – Version 0.8, Draft
- Dittmann 2007** DITTMANN, Lars Uwe: *OntoFMEA : Ontologiebasierte Fehlermöglichkeits- und Einflussanalyse*. 1. Aufl. Wiesbaden : Dt. Univ.-Verl., 2007. – ISBN 978-3-8350-0749-9
- Gruber 1993** GRUBER, Thomas R.: A translation approach to portable ontology specifications. In: *Knowledge Acquisition* 5 (1993), jun, Nr. 2, S. 199–220. – URL <http://www-ksl.stanford.edu/knowledge-sharing/papers/ontolingua-intro.ps>
- Guarino 1998** GUARINO, Nicola: Formal Ontology and Information Systems: Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. In: *Proceedings of FOIS'98: Trento, Italy: 1998, June 06-08*, IOS Press, 1998, S. 3–15. – URL <http://www.loa-cnr.it/Papers/FOIS98.pdf>
- Hepp 2008** HEPP, Martin: Ontologies: State of the art, business potential, and grand challenges. In: HEPP, Martin (Hrsg.) ; LEENHEER, Pieter D. (Hrsg.) ; DE MOOR, Aldo (Hrsg.) ; SURE, York (Hrsg.): *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications*. Heidelberg : Springer-Verlag, 2008 (Semantic Web and Beyond), Kap. 1. – ISBN 978-0-387-69899-1
- Hepp u. a. 2008** HEPP, Martin ; LEENHEER, Pieter D. ; DE MOOR, Aldo ; SURE, York: *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications*. Heidelberg : Springer-Verlag, 2008 (Semantic Web and Beyond). – ISBN 978-0-387-69899-1
- Karvounarakis u. a. 2002** KARVOUNARAKIS, Gregory ; ALEXAKI, Sofia ; CHRISTOPHIDES, Vassilis ; PLEXOUSAKIS, Dimitris ; SCHOLL, Michel: RQL: A Declarative

- Query Language for RDF. In: *WWW '02: Proceedings of the 11th international conference on World Wide Web*. New York, NY, USA : ACM Press, 2002, S. 592–603. – URL <http://www.seas.upenn.edu/~gkarvoun/publications/www2002/www2002.pdf>
- Lassila und McGuinness 2001** LASSILA, Ora ; MCGUINNESS, Deborah L.: *The Role of Frame-Based Representation on the Semantic Web*. 2001. – URL <http://www.ida.liu.se/ext/epa/ej/etai/2001/018/01018-etaibody.pdf>
- Magkanaraki u. a. 2002** MAGKANARAKI, Aimilia ; KARVOUNARAKIS, Grigoris ; ANH, Ta Tuan ; CHRISTOPHIDES, Vassilis ; PLEXOUSAKIS, Dimitris: *Ontology storage and querying* / Foundation for Research and Technology Hellas Institute of Computer Science Information Systems Laboratory. URL <http://xml.coverpages.org/MagkanarakiOnt.pdf>, 2002. – Forschungsbericht. Technical Report No. 308
- Meixner 2004** MEIXNER, Uwe: *Einführung in die Ontologie*. Wissenschaftliche Buchgesellschaft, 2004. – ISBN 3-534-15458-4
- Passin 2004** PASSIN, Thomas B.: *Explorer's Guide to the Semantic Web*. Manning, 2004. – 141–169 S. – ISBN 1-932-39420-6
- Rehfuß 2003** REHFUSS, Wulff D. (Hrsg.): *Handwörterbuch Philosophie*. Vandenhoeck & Ruprecht, 2003. – ISBN 3-525-03323-0
- Smith 2000** SMITH, Barry: *Gegenstände und ihre Umwelten: Von Aristoteles zur ökologischen Ontologie*. In: *Einheit und Vielheit. Organologische Denkmodelle in der Moderne*. Passagen-Verlag, 2000, S. 35–64. – URL <http://ontology.buffalo.edu/smith/articles/Umwelten.pdf>. – ISBN 3-85165-410-2
- Uschold und Jasper 1999** USCHOLD, Mike ; JASPER, Robert: *A Framework for Understanding and Classifying Ontology Applications*. In: *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5): Stockholm, Sweden: 1999, August 2nd*, URL <http://www.cs.man.ac.uk/~horrocks/Teaching/cs646/Papers/uschold99.pdf>, 1999
- Wichmann 2007** WICHMANN, Gabriele: *Entwurf Semantic Web: Entwicklung, Werkzeuge, Sprachen*. VDM Verlag Dr. Müller, 2007. – ISBN 978-3-8364-0398-6
- Yildiz 2006** YILDIZ, Burcu: *Ontology Evolution and Versioning: The state of the art* / Vienna University of Technology, Institute of Software Technology & Interactive Systems (ISIS). Favoritenstrasse 9–11/188, A-1040 Vienna, 2006. – Forschungsbericht. – URL [http://publik.tuwien.ac.at/files/pub-inf\\_4603.pdf](http://publik.tuwien.ac.at/files/pub-inf_4603.pdf)