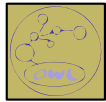


Überblick über OWL



Seminar/**SemanticWiki** (*Thomas Riechert, Sören Auer*)

<https://wiki.imise.uni-leipzig.de/SoerenAuer/SemSemanticWiki>

Universität Leipzig, Institut für Informatik

Wintersemester 2005/2006

Erstellt von: *Adrian Immanuel Kieß, Anonymous*

<http://www.kiess.onl>

Abstract

Im Rahmen des Seminars **SemantikWiki** (<https://wiki.imise.uni-leipzig.de/SoerenAuer/SemSemanticWiki>) wurde ein breites Themenspektrum behandelt: Community Management, Wikisysteme, Ontologien. Ein Teil des Seminars war der Darstellung von Formalismen zur Repräsentation von Metadaten gewidmet, wobei RDF, RDFS, OWL und deren Anwendungen vorgestellt wurden. Diese Ausarbeitung faßt die Fakten zur Ontologiesprache OWL zusammen, der als W3C-Standard eine wesentliche Rolle bei der Realisierung der *Semantic Web Vision* zukommt. Die Arbeit setzt OWL ins Verhältnis zu den anderen Komponenten des Semantic Web, stellt die Konstrukte der Sprache vor und gibt kurze Ausblicke und Grenzbeachtungen.

Keywords

Ontology Language, OWL, Semantic Web, Tutorial

Inhaltsverzeichnis

1	Intro: Semantic Web Vision	4
2	Hintergrund/Ontologie	4
2.1	Definitionen	4
2.2	Anatomie	6
2.3	Ontologiesprachen	6
2.4	Analogie: Objektorientierung/Logik	6
2.5	Typen von Ontologien	7
2.6	Anwendungen	7
3	Sprachbeschreibung	8
3.1	Design Goals	8
3.2	Verortung im Semantic Web Layer	8
3.3	Elementares	9
3.3.1	Klassendefinitionen	9
3.3.2	Axiome für Klassen	9
3.3.3	Individuals	10
3.3.4	Eigenschaften	10
3.3.5	Datentypen	10
3.4	OWL Dialekte	11
3.5	Syntax/Semantik/Reasoning	12
3.6	Implementierungen/Anwendungen	13
3.7	Grenzen von OWL	14
4	Beispiele¹	14
5	Anhang	18
5.1	Definitionen des Ontologiebegriffs	18
5.2	Liste der OWL Konstruktoren	19
6	Literatur	20

¹Weiterführende Kommentare zu diesen Beispielen finden sich unter: <http://www.w3.org/TR/owl-guide/>

1 Intro: Semantic Web Vision

"If HTML and the Web made all the online documents look like one huge book, RDF, schema and inference languages will make all the data in the world look like one huge database."

Tim Berners-Lee, Weaving the Web, 1999

Das Bedürfnis nach der Strukturierung von Wissen, Erkenntnissen und Fakten, ist alt und hat sich in verschiedensten Formen der Wissensrepräsentation niedergeschlagen. Die Menge der Daten, Wissen in Rohform, hat ab einem bestimmten Zeitpunkt die Möglichkeiten ihrer Ordnung überstiegen. Vannevar Bushs Aufsatz von 1945 gibt das unzweifelhaft zu verstehen, problematisiert dieses Feld und gibt die Skizze einer Lösung. Später wird der Terminus Wissensgesellschaft von den Gesellschaftswissenschaften eingeführt, welcher mit der Analyse der Gesellschaftsformen und deren Kernkategorien in Verbindung steht.

Die aktuelle Version der Vision (in der Informatik) nach inhaltlicher, maschineller Informationsstrukturierung - heißt *Semantic Web*, initiiert vom W3C.

Metadaten - Daten über Daten, bilden den Ausgangspunkt für die maschinenlesbare Darstellung von Inhalten (Webdokumente, Datenbanken); diese können in einem ersten Schritt über RDF eingearbeitet werden. Der Zusammenhang der Teile, die Struktur der Metadaten werden in weiteren Leveln modelliert: RDFS und OWL. Schließlich können Werkzeuge eingesetzt werden, um die vorhandenen Wissensbasen zu inferieren.

Der Ausgangspunkt des *Semantic Web* ist das Web. Ein Teil von Berners-Lee ursprünglicher Intention bei der Erfindung des Web ist durch dessen Popularisierung untergraben worden. Der Wissensschatz ist zunächst eine lose Ansammlung von Wissen. Ein in HTML verfasstes Webdokument ist von einer Maschine nicht thematisch oder inhaltlich in der Struktur zu erfassen. Die Einführung von XML hat einen Durchbruch geschaffen, insofern maschinenlesbare Metadaten ermöglicht wurden, welche jedoch eine große Applikationabhängigkeit aufweisen. Die Formulierung eines Faktes in XML beruht auf syntaktische Konvention und kann eingesetzt werden, solange zwischen den Endpunkten der Kommunikation Klarheit über die Interpretation besteht. Das Ziel – Dokumente, die ihre Struktur und die Bedeutung ihrer Elemente selbst offenlegen – kann erst durch weitere Konstruktionen erreicht werden.

Der Erfolg der *Semantic Web Vision* hängt stark von der rechtzeitigen Verfügbarkeit sinnvoller Werkzeuge und Applikationen ab. Eine Reihe von Werkzeugen und Betaversionen stehen zur Verfügung, die erste Schritte im Licht dieser Vision unternehmen. Es ist jedoch anzunehmen, daß der Impuls hinter dem Semantic Web (*Das Web als Buch/Datenbank*, Verknüpfung heterogener Daten und Datenquellen, mannigfache Sichtenkreation) selbst bei dessen Mißerfolg (Scheitern als Revolution) nicht aufgehoben sein wird.

2 Hintergrund/Ontologie

2.1 Definitionen

- **Philosophisch:** Ontologie ist eine philosophische Disziplin im Bereich der Metaphysik. Hergeleitet aus dem griechischen *on*, Seiendes, heißt es übersetzt *Lehre vom Sein*. Die im Mittelpunkt stehende Fragen sind: Was charakterisiert das Sein? Was ist Sein? Wie sind die Dinge, von denen gesagt werden kann, sie seien, geordnet? Erste Problematisierungen dieser Fragen finden sich in Aristoteles Schriften. Ontologische Fragestellungen beziehen sich auch auf die Frage nach der Existenz von Dingen, zu denen ein sinnlicher Zugang fehlt

und die gleichzeitig im Denken verankert sind, wie zunächst alle Abstrakta; Raum, Zeit, Gesellschaft, Gott. Neben den Fragen nach den Möglichkeiten und dem sinnvollen Gebrauch von Abstrakta, versuchen die Fragen nach der Ordnung eine Klassifikation der Dinge, von denen angenommen werden kann, sie sind existent. Das Perspektivische und das Allgemeine stehen hierbei im Widerstreit miteinander, da eine Seinskatalogisierung einen Rest an Pragmatik mit sich trägt², gleichzeitig kategorial umfassend Weltbeschreibung, gerade unabhängig von Positionierungen zu erreichen sucht.

- **Linguistisch:** Generell: Der Zugang zu den Dingen erfolgt über Sprache. Wörter stehen in mannigfachen Beziehung zueinander, die auch ontologischen Charakter haben können (Beispielsweise *Hyponym/Hypernym* Relation). Abbildung X verortet die Rolle der Ontologie im Rahmen des semiotischen Dreiecks.

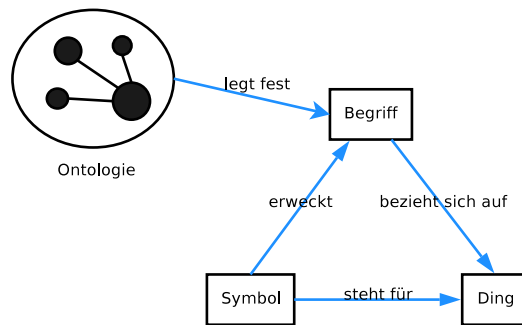


Abbildung 1: Semiotisches Dreieck (nach Ogden, Richards | 1923)

- **Informatik:** Ontologien sind Artefakte (*engineering artifacts*), die ein bestimmtes Vokabular, die einen Weltausschnitt beschreiben, und eine Reihe von Annahmen über die intendierte Bedeutung des Vokabulars enthalten. Gruber definiert Ontologie als "... a formal specification of a shared conceptualisation." (Gruber). Wissen über einen Bereich, **verteilt** zusammengetragen und **verwendbar**, **formal** niedergelegt, **maschinenlesbar**. Ein verwandter Begriff ist der der Konzeptualisierung, der nützlich ist, um von dem Ontologiebegriff der Philosophie, der sich Erfahrungsunabhängigkeit vornimmt, abzurücken, hin zu der Pragmatik (speziell) einer Gruppe, die *ihr* Wissen, *ihr* Vokabular oder auch einfach *ihren* Weltausschnitt zu erfassen versucht. Im Anhang (X.X) sind weitere Definitionen des Ontologiebegriffs zusammengetragen.

Formelhaft lassen sich die Fragestellungen der einzelnen Disziplinen in Bezug auf Ontologien wie folgt zusammenfassen:

- Philosophie: Was existiert?
- Linguistik: Worüber sprechen wir?
- Informatik: Was wird repräsentiert?
- Psychologie: Was sind die Basiskategorien unseres Denkens?

²Beispielsweise: Wieso ist die Idee des Guten die höchste (allgemeinste) Idee bei Platon?

2.2 Anatomie

Eine **Ontologie** besteht aus drei Teilen: Einer *Menge von Namen*, die Entitäten oder Konzepte bezeichnen, *Definitionen* der Konzepte und *Axiomen*. Namen und Konzeptdefinitionen werden nicht immer unterschieden. Werden neben einer Ontologie auch entsprechende Instanzen der Konzepte betrachtet, wird das Name **Wissensbasis** gebraucht. Die Grenze zwischen Ontologie und Wissensbasis kann fließend sein; die Verwechslung von Instanzen und Klassen bei der Modellierung gehört zu den typischen Fehlern des *Ontological Engineering*.

2.3 Ontologiesprachen

Es gibt eine Anzahl von Möglichkeiten zur expliziten Spezifikation von Wissen/Aussagen/Fakten:

Graphische Notationen

- Semantische Netzwerke (<http://www.jfsowa.com/pubs/semnet.htm>)
- Topic Maps (<http://www.topicmaps.org/>)
- UML (<http://www.uml.org/>)
- RDF (<http://www.w3.org/TR/rdf-primer/>)

Logikbasiert

- Beschreibungslogisch: OIL, DAML+OIL, OWL
- Regelbasiert: RuleML (<http://www.ruleml.org/>), Prolog
- First Order Logic (FOL): KIF (<http://logic.stanford.edu/kif/>)
- Konzeptuelle Graphen (<http://www.jfsowa.com/cg/>)
- Syntaktische Logiken höherer Stufe: LBase (<http://www.w3.org/TR/2003/NOTE-lbase-20031010/>)
- nichtklassische Logiken: FLogic, Non-Mon

Der Formalismus einer Sprache bestimmt die Möglichkeiten zur Maschinenverarbeitung.

2.4 Analogie: Objektorientierung/Logik

Die Objektorientierung teilt ein wesentliches Merkmal mit Arten von wissensbasierten Systemen: Die Trennung von Konzept und Objektebene. In der Beschreibungslogik werden dafür die Bezeichnungen *terminological box* und *assertional box* verwendet. Folgende Tabelle schematisiert diesen Fakt:

Objektorientierung	Ontologie	Logik
Objekte	Instanzen, Individuals	Konstanten
Klassen	Typen, Konzepte	unäre Prädikate
Attribute	Relationen, Eigenschaften, Rollen	binäre Prädikate

2.5 Typen von Ontologien

Ontologien können nach ihrem Einsatzbereich unterschieden werden (Guarino:98):

- *Top-Level*-Ontologien: Stellen übergeordnete Modelle zur Verfügung, die als Leitlinie bei der Entwicklung von weniger generellen Ontologien dienen können; dabei spielen Kategorien wie Zeit, Raum, Objekt und Prozess eine Rolle. Beispiele: Standard Upper Ontology (SUO/SUMO³), General Formal Ontology (GFO⁴)
- *Domain*-Ontologien: Wissen über einen konkreten und begrenzten Anwendungsbereich wird repräsentiert (z.B. Genom-Ontologie, Linguistische Kategorien)
- *Task*-Ontologien: Beschreibung von Vokabular bezogen auf Tätigkeiten und Aktivitäten.
- *Application*-Ontologien: Spezialisierung einer Domain- oder Task-Ontologie auf eine bestimmte Anwendung hin.

Aber auch durch die Intension der Ontologie lassen sich verschiedene Typen unterscheiden (SEOP:2004):

- repräsentational: Repräsentation einer Domäne, wobei echte ontologische Fragestellung nicht berücksichtigt werden müssen.
- deskriptiv: Wie repräsentationale Ontologie, jedoch mit dem Anspruch eine Domäne möglichst getreu abzubilden.
- systematisch: Domänenübergreifene Ontologien.

2.6 Anwendungen

Ontologien bilden aufgrund ihrer Eigenschaften (Weltbeschreibung, Ordnung von Dingen, Wissensrepräsentation, Formalisierung) einen Punkt, an dem sich viele Disziplinen schneiden. Das Gebiet der Ontologien ist interdisziplinär, sowohl in der Erforschung als auch in der Anwendung. Philosophie, Bibliothekswissenschaften, Informationssysteme, Künstliche Intelligenz, Kognitionswissenschaften, Sprachverarbeitung und Semantic Web liefern Material für theoretische Fragen und praktische Aufgaben.

- Philosophie: Grundlegende Kategorie der Erkenntnis, Fragen nach Dingen wie Zeit, Raum oder Sprache
- Bibliotheks/Informationswissenschaften: Bibliographisches Universum, Modellierung und Implementierung von Domänenwissen
- KI: Common sense reasoning (Cyc⁵)
- Sprachverarbeitung: Lexikalisches und linguistisches Wissen, maschinelle Übersetzung (WordNet, GOLD)
- Semantic Web: Das Web als heterogene, aber strukturierte Wissensbasis

Konkrete Anwendung (im Hintergrund) finden Ideen des Semantic Web bereits bei einer Reihe von Unternehmen und Produkten. Näheres dazu lässt sich unter <http://www.w3.org/2005/Talks/0524-Amsterdam-NSS-IH/> finden.

³Siehe: <http://www.ontologyportal.org/>

⁴Projekt der Universität Leipzig, Siehe: <http://www.ontology.uni-leipzig.de/>

⁵Umfängliches KI/Ontologieprojekt in den Neunziger Jahren. Siehe: <http://www.cyc.com/>

3 Sprachbeschreibung

OWL dient der Beschreibung von propositionalen Informationen, also Aussagen, die einen Wahrheitswert annehmen können.

3.1 Design Goals

Folgende Punkte nennen im Überblick die Ergebnisse des Prozesses der Entwicklung von OWL, seine Design Goals:

- Verteilte Entwicklung und Anwendung ontologisch niedergelegtem Wissen
OWL Ontologien trennen zum einen Objektebene und Konzeptebene, analog zu Beschreibungslogik und Objektorientierung. Gleichzeitig können Ontologien über eine Importfunktion beliebig viele vorhandene Dokumente einbinden. Somit lassen sich auch auf der Persistenzebene Schema und Fakten trennen, desweiteren können beide Teile selbst wieder beliebig zusammengestellt werden. Diese Modularität ermöglicht Flexibilität. Allerdings birgt die Modularität aufgrund der relative Freiheit des Designs einzelner Ontologien noch Probleme. Beispielsweise ist es in der aktuellen OWL Spezifikation nicht möglich *Teile* einer Ontologie zu importieren.
- Evolution
Ontological Engineering ist ein iterativer Prozess. Um diesem Fakt Rechnung zu tragen, sind Versionierungsinformationen in Ontologiedokumenten möglich. Desweiteren können Aussagen über Kompatibilität und Mißbilligung (Deprecation) von Ontologieelementen (Klassen, Eigenschaften) explizit formuliert und bei Verarbeitung ausgewertet werden.
- Möglichkeit zur Erkennung von Inkonsistenzen
OWL baut formal auf der Beschreibungslogik (SHIQ/D) auf. Reasoner können OWL Dokumente in eine logische Notation übersetzen und darüber inferieren. Eine Aufgabe von Reasoningtools besteht in der Aufdeckung von Inkonsistenz.
- Expressivität und Skalierbarkeit
OWL besitzt drei Dialekte. Dies ist der Tatsache geschuldet, daß nur OWL Full nahtlos auf RDF/S aufbaut und OWL Full als Sprache nicht entscheidbar ist, was allerdings eine Anforderung an die Sprache gewesen ist. Die Trennung in drei Subsprachen erlaubt es dem Anwender allerdings, den passenden - also in der Ausdrucksstärke ausreichenden - Dialekt zu wählen.
- XML-Syntax
Die OWL-Syntax ist eine RDF-Syntax. RDF besitzt eine Serialisierung als XML.
- Internationalisierung
Alle Literale werden über XML Schema Datentypen repräsentiert, wodurch das Sprachattribut zur Internationalisierung zur Verfügung steht.

3.2 Verortung im Sematic Web Layer

OWL bildet eine eigene Schicht im Semantic Web Layer. Die unter OWL liegenden Technologien sollen in ihrer Bedeutung für OWL skizziert werden.

URI Uniform Resource Identifier dienen zur Identifizierung von abstrakten oder physischen Ressourcen und werden im RFC 2396 beschrieben. Die URI ist fundamental. Sie ist *eindeutig*, dient nur zur *Identifikation* - nicht zur Lokation, sie ist *offen für alle* (jeder kann Metadaten im Web erstellen, verteilen, zusammenführen).

Unicode Unicode ermöglicht über 16-bit Codierung Internationalisierung von Dokumenten. Allerdings lassen sich URI noch nicht in Unicode ablegen.

XML XML bildet die Basis für die Maschinenlesbarkeit. XML Schema Datentypen können in OWL verwendet werden, allerdings eingeschränkt.

Namespaces Namespace ermögliche Scopes, in denen eigenes Vokabular verwendet und gegen Fehlnutzung gesichert werden kann.

RDF RDF dient der Beschreibung von Ressourcen, der Formulierung von Metadaten. RDF hat eine XML-Syntax und ein Graphenmodell. Klassen, Konzepte oder Eigenschaften können nicht explizit formuliert werden. RDF Statements können als $\langle \text{Subjekt, Prädikate, Objekt} \rangle$ Aussagen aufgefaßt werden, ähnlich natürlichsprachlichen Ausdrücken⁶. Allerdings besitzt RDF eine nicht-standardlogische Semantik.

RDFS RDFS verhält sich zu RDF nicht wie XMLS zu XML. Letzteres Verhältnis ist eines der Dokumentstruktur zu einem Dokument. RDFS stellt dagegen Vokabular für die Modellierung einfacher Ontologien bereit: Konzepte, Konzepthierarchie, Konzeptigenschaften, Wertebereiche und Instanzen.

3.3 Elementares

OWL erweitert das Vokabular von RDFS um Konstrukte, die eine weitreichendere und detailliertere Modellierung ermöglichen.

3.3.1 Klassendefinitionen

Es gibt sechs Möglichkeiten eine Klasse mit OWL zu definieren:

- Klassenidentifikation
- Aufzählung der Elemente (Enumerated Class)
- Einschränkung von Eigenschaften (Property Restrictions)
- Schnitt
- Vereinigung
- Komplement

3.3.2 Axiome für Klassen

Beziehungen zwischen Klassen können über drei Konstrukte angegeben werden:

- Subklassenbeziehung
- Äquivalente Klasse
- Disjunktheit von Klassen

⁶Zumindest in Sprachen mit starrer Satzstellung.

3.3.3 Individuals

Individuals bilden den Rand einer Ontologie und definieren die Fakten, bezeichnen im weiteren Sinn konkrete Entitäten. Die Frage ob ein Sachverhalt einer Klasse oder einem Individuum entspricht ist ein typische Modellierungsfehler während der Ontologieentwicklung. Eine Methode zur Vermeidung ontologische Unstimmigkeiten ist die Frage nach der Instanzierbarkeit während des Entwurfs der Klassen: Solange von einer Sache Instanzen möglich (wenn auch nicht unbedingt für die spezielle Ontologie nötig) sind, sollten sie als Klassen definiert werden.

Zwischen zwei Individuen kann in OWL Gleichheit oder Distinktheit explizit angegeben werden. Zu bemerken ist hierbei von OWL nicht geforderte Unique Name Assumption: Zwei Individuen mit unterschiedlichen Bezeichnern sind ohne weitere Angaben nicht notwendig verschieden. Eine Frage für das Reasoning.

3.3.4 Eigenschaften

Eigenschaften können im Gegensatz zu RDFS in OWL auch lokal an Klassen gekoppelt werden. Es wird zwischen Datatype Properties und Object Properties unterschieden. Erstere sind Relationen zwischen einem Individuum und einem Literal, ein XSD, letztere verbinden zwei Individuals. Im Anhang findet sich eine Tabelle mit den Eigenschaftensnamen, sowie Definitions- und Wertebereichen.

Elementare Eigenschaftskonstrukte/Hierarchie Eine Eigenschaft wird festgelegt, indem die Wertebereiche bestimmt werden und eine Entscheidung zwischen Object Property und Datatype Property getroffen wird. Zusätzlich kann eine Eigenschaftshierarchie festgelegt werden, Untereigenschaften schränken dabei den möglichen Wertebereich der übergeordneten Eigenschaft ein.

Eigenschaftscharakteristiken Eigenschaften können explizit als symmetrisch, transitiv, invers, funktional und inversfunktional beschrieben werden.

Eigenschaftsrestriktionen Wertebereichseinschränkung im Klassenbereich werden über Eigenschaftsrestriktionen realisiert, wodurch der Starrheit von RDFS – nur globale Eigenschaftsdefinition – überwunden wird. Der Mechanismus wird anhand eines Beispiels verdeutlicht.

Kardinalität Kardinalität können in OWL als Maximum, Minimum und exakte Kardinalität festgelegt werden.

Klassen, die an ein Individual gebunden sind Klassen können in ihrer Definition an eine spezielle Ausprägung einer anderen Klasse gebunden sein, was aber nur in OWL-DL möglich ist.

3.3.5 Datentypen

OWL erlaubt drei Arten von Datentypen:

- RDF Datentypen
- RDFS Klasse Literal
- Aufzählende Datentypen

3.4 OWL Dialekte

OWL besitzt drei Dialekte: OWL Full, OWL DL und OWL Lite. Nur OWL Full ist voll abwärtskompatibel zu RDF – jedes RDF Dokument ist ein OWL Full Dokument. Für OWL DL und OWL Lite gilt: Beide sind gültiges RDF, allerdings ist aufgrund der Mächtigkeit von RDF/S nicht jedes RDF Dokument ein OWL DL oder OWL Lite Dokument. Diese Aufspaltung läßt dem Ontologieentwickler die Wahl, markiert jedoch gleichzeitig einen Riß im Semantic Web Layer.

OWL Lite OWL Lite ist für die Implementierung einfacher Ontologien gedacht, in denen Konzepthierarchie, Eigenschaften, Eigenschaftscharakteristiken eine Rolle spielen. Mengenvereinigung von Klassen, sowie Kardinalitäten außer 0 und 1 sind in OWL Lite nicht möglich.

OWL DL OWL DL basiert auf der Beschreibungslogik (SHOIQ/D). Beschreibungslogiken werden in wissensbasierten Systemen eingesetzt, in denen neben der Aufgabe der Wissensrepräsentation, die der Wissensextraktion, der Inferenz, gestellt wird. Beschreibungslogiken sind entscheidbare Untermengen der Prädikatenlogik erster Stufe und stellen somit eine formale Semantik bereit, die älteren Ansätzen zur Wissensrepräsentation (Frames, Semantic Networks) fehlte. OWL DL und OWL Full besitzen den gleichen Umfang an Konstruktoren. Der Unterschied besteht darin, daß in OWL Einschränkung an die Verwendung der Konstrukte gestellt werden:

- Partitionierung der Vokabeln (*vocabulary partitioning*)
Jede Ressource ist gehört zu genau einem Typ. Die Menge der Klassen, Instanzen und Eigenschaften ist disjunkt.
- Explizite Typisierung (*explicit typing*)
Ressource, die über Inferenz als Klasse identifiziert werden *könnten*, müssen explizit definiert werden.
- Partitionierung der Eigenschaften (*property separation*)
Datatype und Object Properties sind disjunkt. Inverse, funktionale, invers-funktionale und symmetrische Eigenschaft können nur Object Properties besitzen.
- Keine Eigenschaftsrestriktionen bei transitive Eigenschaften (*no transitive property restrictions*)
- Einschränkung bei anonymen Klassen (*restrictions on anonymous classes*)
Anonyme Klassen sind nur erlaubt im Domain und Range von `owl:equivalentClass` und `owl:disjointWith`, und nur im Range von `rdfs:subClassOf`.

OWL Full OWL Full erlaubt alle OWL, RDF und RDFS Konstrukte; die Einschränkungen von OWL DL sind aufgehoben. Dies bedeutet unter anderem, daß es keine eindeutige Unterscheidung zwischen Klassen und Instanzen gibt. Eine Ressource kann gleichzeitig Instanz und Klasse sein. Durch die Anwendung von sprach-eigenen Konstrukten auf sich selbst⁷ ist das Kriterium der Entscheidbarkeit nicht mehr erfüllt, korrekte und terminierende Inferenz kann nicht garantiert werden.

⁷Beispielsweise läßt sich durch Kardinalitätsrestriktionen auf die Klasse `owl:Class` die Anzahl der in der Ontologie enthaltenen Klassen beschränken.

3.5 Syntax/Semantik/Reasoning

Syntax OWL besitzt eine RDF Syntax. Eine OWL Ontologie läßt sich auch als Menge von RDF Tripeln notieren. RDF selbst hat verschiedene Serialisierungen: RDF Graph, RDF/XML, N3. RDF/XML ist die offizielle sogenannte *exchange-syntax*. Da OWL Dokumente in RDF/XML *verbose* und unleserlich sind, werden Alternativen zur Notation angeboten: Eine reine XML Notation und eine sogenannte *abstract syntax*. OWL Dokumente sind nicht für die manuelle Erstellung gedacht - der Einsatz von Ontologieeditoren ist unabdingbar.

Semantik OWL besitzt zwei Semantiken, eine aufbauend auf Basis einer Standard-Modelltheorie⁸, eine weitere auf Basis der RDF Semantik⁹. Aufgrund der Entscheidbarkeit der Beschreibungslogik SHOIQ/D, welche mit OWL DL korrespondiert, können über OWL DL (und OWL Lite) Dokumente Schlüsse gezogen werden.

Reasoning Reasoning erfüllt mehrere Aufgaben:

- Konsistenzprüfung von Ontologien
Eine Ontologie wird inkonsistenz genannt, wenn sie ein Konzept enthält, für welche jede Interpretation leer sein muß. Diese Inkonsistenzen lassen sich über Inferenz aufdecken.
- Klassifikation
Individuen und Klassen können mittels Inferenz weiter klassifiziert, in die Ontologie eingeordnet, an die richtige Stelle gefügt werden. Fragen nach der Subsumtion von Konzepten und nach der Zugehörigkeit von Individuen zu einem Konzept sind dabei zu beantworten.
- Faktenschließung
Anhand der in einer Ontologie vorliegenden Fakten, Konzept- und Eigenschaftsdefinitionen können weitere Fakten über verschiedene Wege hergeleitet werden.

OWL besitzt keine Unique Name Assumption (UNA), während einige Reasoner *mit* einer UNA arbeiten.

Beispiele für Inferenz Im folgenden sollen lediglich natürlichsprachliche Aussagen zur Illustration dienen.

- Überkonzept ermitteln:
Prämissen: Ein Busfahrer ist eine Person, die einen Bus fährt. / Ein Bus ist ein Fahrzeug. / Ein Fahrzeug wird von einem Fahrer gefahren.
Also: Ein Busfahrer ist ein Fahrer.
- Über Eigenschaftshierarchie
Prämissen: Katzenbesitzer haben Katzen als Haustiere. / ein Haustier besitzen ist Untereigenschaft von mögen.
Ergo: Katzenbesitzer mögen ihre Katze.

⁸<http://www.w3.org/TR/2004/REC-owl-semantics-20040210/direct.html>

⁹<http://www.w3.org/TR/2004/REC-owl-semantics-20040210/rdfs.html>

- Vollständige Definition

Prämissen: Schafe essen ausschließlich Gras. / Gras ist eine Pflanze. / Pflanzen und Teile von Pflanzen sind disjunkt von Tieren und Teilen von Tieren. / Vegetarier essen ausschließlich Dinge, die nicht Tier sind oder Teil von einem Tier.

Es folgt: Schafe sind Vegetarier.

- Inverse und Domain/Range

Prämissen: Spike ist das Haustier von Pete. (angenommen ist *HaustierVon* ist *ObjectProperty* mit Domain 'Name des Tieres' und Range 'Name des Besitzers' und *InverseProperty*)

Daraus läßt sich schließen: Pete hat ein Haustier Spike. Pete ist ein Mensch. Spike ist ein Tier.

3.6 Implementierungen/Anwendungen

Es gibt eine Reihe von APIs für die Anbindung von OWL Ontologien an Applikationen:

- Jena

Von HP entwickeltes Semantic Web Framework mit Java API. Zur Verwendung soll der Verweis auf <http://jena.sourceforge.net/> genügen. Beispieldateien finden sich im Package `com.hp.hpl.jena.tutorial`.

- ProtégéOWL

Das OWL Plugin¹⁰ hat seinen Basisapplikation Protégé¹¹ gemessen an den LOC¹² eingeholt. Die API enthält alle Elemente, um OWL Ontologien von Anwendungen aus zu erstellen, manipulieren und zu sichern.

- Redland¹³

RDF API verfügbar für Ruby, Perl, C#, Java, Obj-C, PHP, Ruby und Tcl.

Für die Arbeit mit Ontologien stehen ebenso verschiedene Werkzeuge bereit:

- Validierung

OWL-Ontologie-Validator (Klassifiziert in OWL Lite, OWL DL und OWL Full) (<http://phoebus.cs.man.ac.uk:9999/OWL/Validator>)

- Inferenz

KAON2 (<http://kaon2.semanticweb.org>)

RACER (<http://www.sts.tu-hamburg.de/~r.f.moeller/racer/>)

FACT (<http://www.cs.man.ac.uk/~horrocks/FACT/>)

Pellet (<http://www.mindswap.org/2003/pellet/index.shtml>)

¹⁰Auffindbar unter: <http://protege.stanford.edu/plugins/owl/>

¹¹Ontologieeditor: <http://protege.stanford.edu>

¹²Metrik: Lines of Code.

¹³<http://librdf.org/>

3.7 Grenzen von OWL

An einigen Stellen stößt OWL an seine Grenzen:

- Temporalität. Relationen zwischen Dingen, die Zeitlichkeit besitzen, lassen sich nicht formulieren.
- Probleme bei Import. Es können bisher nur gesamte OWL Dokumente importiert werden, keine Teildokumente. Neben diesem Problem existiert das Problem der Kollision von Klassendefinition.

4 Beispiele¹⁴

Taxonomie Für die Klassenhierarchie werden die Elemente `owl:Class` und `rdfs:subClassOf` benötigt.

Beispiele:

```
<owl:Class rdf:ID="Wine" />
<owl:Class rdf:ID="WineGrape" />
```

Im Beispiel werden zwei Klassen über ihre ID definiert, *Wine* und *WineGrape*.

Individuen Individuen können als Elemente einer Klasse gesehen werden. Die Minimaleigenschaft eines Individuums ist die Zugehörigkeit zu einer Klasse.

Beispiele:

```
<Wine rdf:ID="ZinfandelCabernet">
```

oder

```
<owl:Thing rdf:ID="ZinfandelCabernet" />
<owl:Thing rdf:about="#ZinfandelCabernet">
  <rdf:type rdf:resource="#Wine">
</owl:Thing>
```

Die zwei Schreibweisen sind gleichgültig. Die zweite, ausführlichere Schreibweise zeigt die Verteilbarkeit von Definitionen, da die zwei Elemente nicht in einer Datei stehen müssen, sondern über eine URI aus verschiedenen Quellen kombiniert werden können.

Eigenschaften Eigenschaften legen generelle Besonderheiten einer Klasse und spezielle Wertausprägungen eines Individuums fest. Eigenschaften sind zweistellige Relationen, wobei es zwei Typen von Relationen gibt: Relationen zwischen zwei Instanzen (ObjectProperty) und Relationen zwischen einer Instanz und einem Literal (DatatypeProperty). Relationen können verschiedene Einschränkungen besitzen: Einschränkungen des Definitionsbereiches, des Wertebereiches, Spezialisierung einer übergeordneten Eigenschaft.

Beispiele:

```
<owl:ObjectProperty rdf:ID="madeFromGrape">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineGrape"/>
</owl:ObjectProperty>
```

¹⁴Weiterführende Kommentare zu diesen Beispielen finden sich unter: <http://www.w3.org/TR/owl-guide/>

In OWL ist eine Folge von Elementen ohne weiteren Operator eine Konjunktion. Im Beispiel besitzt die Eigenschaft *madeFromGrape* also die Domäne *Wine* und den Range *WineGrape*. Definitions- und Wertebereichsdefinitionen bieten eine erste Möglichkeiten des Schließens. Wenn also eine nicht näher spezifizierte Ressource die Eigenschaft *madeFromGrape* besitzt, dann kann daraus der Typ der Resource – in Fall des Beispiels die Domain *Wine* – abgeleitet werden:

```
<owl:Thing rdf:ID='CabernetShiraz'>
  <madeFromGrape rdf:resource='CabernetGrape'>
</owl:Thing>
```

Eigenschaften können – wie Klassen – eine Hierarchie bilden. Hierfür wird das *subPropertyOf* Element eingesetzt.

Beispiel:

```
<owl:ObjectProperty rdf:ID="hasWineDescriptor">
  <rdfs:domain rdf:resource="#Wine" />
  <rdfs:range rdf:resource="#WineDescriptor" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasColor">
  <rdfs:subPropertyOf rdf:resource="#hasWineDescriptor" />
  <rdfs:range rdf:resource="#WineColor" />
  ...
</owl:ObjectProperty>
```

Die Eigenschaft *hasColor* ist eine Untereigenschaft von *hasWineDescriptor*. Die Domain bleibt erhalten, während der Wertebereich weiter eingeschränkt wird (in diesem Fall auf *WineColor*, eine Klasse, die Unterklasse von *WineDescriptor* ist).

Weiterhin sind auch transitive Konstrukte möglich.

Beispiel:

```
<owl:ObjectProperty rdf:ID="locatedIn">
  ...
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

In diesem Fall wird eine Eigenschaft *locatedIn* für alle möglichen Domains definiert. Somit können auch Regionen selbst wieder Regionen enthalten.

Kardinalitätseinschränkungen können über das Element *owl:Restriction* realisiert werden.

Beispiel:

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
        1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Im diesem Beispiel wird (unter anderem) festgelegt, daß eine Instanz der Klasse Wein mindestens eine Rebsorte besitzen muß. Dies geschieht über die Definition einer anonymen Subklasse, der alle Individuen angehören, die genau diese Bedingung erfüllen.

Bei der Definition von DatatypeProperties werden die XML Schema Datentypen (XSD) verwendet.

Instanzen und Eigenschaften

Beispiel:

```
<CabernetSauvignon
  rdf:ID="SantaCruzMountainVineyardCabernetSauvignon" >
  <locatedIn rdf:resource="#SantaCruzMountainsRegion"/>
  <hasMaker rdf:resource="#SantaCruzMountainVineyard" />
</CabernetSauvignon>
```

Hier wird ein Individual CabernetSauvignon definiert, mit zwei Eigenschaften, *locatedIn* und *hasMaker*.

Spezielle Eigenschaften Die Relationen, die als Eigenschaften auftreten, können selbst bestimmte Eigenschaften haben. In OWL ist es möglich, Transitivität, Symmetrie, Eindeutigkeit und Inverse explicit zu formulieren.

Transitivität Beispiel für Transitivität:

```
<owl:ObjectProperty rdf:ID="locatedIn">
  <rdf:type rdf:resource="#owl:TransitiveProperty" />
  <rdfs:domain rdf:resource="#owl:Thing" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

Symmetrie Beispiel für eine transitive Relation ist zum Beispiel die Schachtelung von geographischen Regionen.

Beispiel für Symmetrie:

```
<owl:ObjectProperty rdf:ID="adjacentRegion">
  <rdf:type rdf:resource="#owl:SymmetricProperty" />
  <rdfs:domain rdf:resource="#Region" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

Beispiel für Symmetrie wäre zum Beispiel die Nachbarschaft zweier Regionen.

Eindeutigkeit (analog)

Inverse (analog)

Inversfunktional (analog)

Eigenschaftseinschränkungen Für Einschränkungen wurde bereits das Element owl:Restriction vorgestellt; um auf Eigenschaften Einschränkungen zu definieren wird das Element owl:onProperty verwendet.

Beispiel für *allValuesFrom*:

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Hier wird der Range von *hasMaker* auf Individuals der Klasse *Winery* eingeschränkt (da *hasMaker* als generelle Eigenschaft sich nicht nur auf Weine beziehen muß, sondern zum Beispiel auch auf Autos. Die Einschränkung gilt hier lokal, nur für die umschließende Klasse *Wine*).

Beispiel für *someValuesFrom*:

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:someValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Der Unterschied zwischen *allValuesFrom* und *someValuesFrom* ist der Unterschied zwischen Allquantor und Existenzquantor (Für alle/einige Individuals gilt...).

Kardinalität Für Kardinalitätsrestriktionen stehen drei Elemente zur Verfügung, minCardinality, maxCardinality, cardinality.

Beispiel:

```
<owl:Class rdf:ID="Vintage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasVintageYear"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Hier wird die Anzahl der Ausprägungsinstanzen der Eigenschaft *hasVintageYear* auf exakt eins beschränkt.

5 Anhang

5.1 Definitionen des Ontologiebegriffs

1. Bezeichnung für jenes Teilgebiet der Metaphysik, das sich mit dem Wesen der Existenz oder des Seins beschäftigt. (Fröhlich im Buch Wörterbuch Psychologie)
2. Ontologie erklärt die Beschaffenheit der Welt; Epistemologie erklärt die Beschaffenheit unserer Erfahrung von dieser Welt. (von Heinz von Foerster im Buch Kybernetik im Text Epistemologie und Kybernetik: Rückblick und Ausblick. Ein Fragment. (1985))
3. Ontologie ist also die Wissenschaft, die Theorie oder die Untersuchung des Seins, bzw. die Erforschung dessen, was ist, 'wie es ist' usw. (von Heinz von Foerster im Buch Wissen und Gewissen (1993) im Text Betrifft: Erkenntnistheorien)
4. A specification of a representational vocabulary for a shared domain of discourse – definitions of classes, relations, functions, and other objects – is called an ontology. (von Tom Gruber im Text A Translation Approach to Portable Ontology Specifications (1993))
5. Ontologie ist ein überlieferter Begriff aus der Philosophie und steht dort für die Lehre vom Sein – genauer: von den Möglichkeiten und Bedingungen des Seienden –, ist also eng verwandt mit der Erkenntnistheorie, die sich mit den Möglichkeiten und Grenzen menschlichen Wahrnehmens und Erkennens auseinandersetzt. (von Wolfgang Hesse im Text Ontologie(n) (2002))
6. A domain ontology is a systematic account – a list – of all the basic concepts (i.e., the objects, relations, and operations) that are needed in a particular domain. The primitives have to be defined for any system whatsoever, be it a database system, a communication system, an expert system, a system for understanding natural language, or a robot. (von Rolf Pfeifer, Christian Scheier im Buch Understanding Intelligence (1999) im Text Embodied Cognitive Science)
7. Eine Ontologie [beschreibt] also einen Wissensbereich (knowledge domain) mit Hilfe einer standardisierenden Terminologie sowie Beziehungen und ggf. Ableitungsregeln zwischen den dort definierten Begriffen. Das gemeinsame Vokabular ist in der Regel in Form einer Taxonomie gegeben, die als Ausgangselemente (modelling primitives) Klassen, Relationen, Funktionen und Axiome enthält. (von Wolfgang Hesse im Text Ontologie(n) (2002))
8. An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. For AI systems, what "exists" is that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, in the context of AI, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is

the statement of a logical theory. (von Tom Gruber, erfasst im Biblionetz am 09.10.2002)

5.2 Liste der OWL Konstruktoren

rdf:Property	rdfs:domain	rdfs:range
owl:allValuesFrom	owl:Restriction	rdfs:Class
owl:backwardCompatibleWith	owl:Ontology	owl:Ontology
owl:cardinality	owl:Restriction	xsd:nonNegativeInteger
owl:complementOf	owl:Class	owl:Class
owl:differentFrom	owl:Thing	owl:Thing
owl:disjointWith	owl:Class	owl:Class
owl:distinctMembers	owl:allDifferent	rdf:List
owl:equivalentClass	owl:Class	owl:Class
owl:equivalentProperty	rdf:Property	rdf:Property
owl:hasValue	owl:Restriction	
owl:imports	owl:Ontology	owl:Ontology
owl:incompatibleWith	owl:Ontology	owl:Ontology
owl:intersectionOf	owl:Class	rdf:List
owl:inverseOf	owl:ObjectProperty	owl:ObjectProperty
owl:maxCardinality	owl:Restriction	xsd:nonNegativeInteger
owl:minCardinality	owl:Restriction	xsd:nonNegativeInteger
owl:oneOf	owl:Class	rdf:List
owl:onProperty	owl:Restriction	rdf:Property
owl:priorVersion	owl:Ontology	owl:Ontology
owl:sameAs	owl:Thing	owl:Thing
owl:someValuesFrom	owl:Restriction	rdfs:Class
owl:unionOf	owl:Class	rdf:List
owl:versionInfo		

6 Literatur

Ausarbeitung

- Guarino, N., Formal Ontology and Information Systems. In: Guarino, N. (Hg.), Formal Ontology in Information Systems. Proceedings of the first international conference (FOIS/98), Amsterdam, 1998. S. 3-15.
- [SEOP:2004] Stanford Encyclopedia of Philosophy. Article: Logic and Ontology: <http://plato.stanford.edu/entries/logic-ontology/>
- Antoniou, G., van Harmelen, F.: A Semantic Web Primer, MIT Press, 2004.
- Passin, T.B.: Explorers Guide to the Semantic Web, Manning Publications Co., 2004.
- Powers, S.: Practical RDF, O'Reilly, 2003.
- W3C Specifications and Working Groups and Presentations:
 1. Semantic Web: <http://www.w3.org/2001/sw>
 2. Tim Berners-Lee Overview: <http://www.w3.org/2000/Talks/1206-xml2k-tbl>
 3. Ian Horrocks on RDF/OWL: <http://www.w3.org/2005/Talks/0524-Amsterdam-NSS-IH>

Definitionen

- Fröhlich, W. D., Wörterbuch der Psychologie. DTV, 2005.
- Foerster, H. v., KybernEthik. Merve, 1993.
- Foerster, H. v., Wissen und Gewissen. Suhrkamp, 1993.
- Gruber, T., A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2):199–220, 1993.
- Hesse, W., Ontologie(n). In: Informatik Spektrum Bd. 25, Heft 6, pp. 477-480, 2002. (http://www.mathematik.uni-marburg.de/~hesse/papers/Hes_02b.pdf)
- Pfeifer, R., Scheier, C., Understanding Intelligence. MIT Press, 1999.
- Gruber, T., What is an ontology? <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>)