



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

## **Ausarbeitung Seminar Ringvorlesung**

Gerrit Diederichs

Bausteine für Semantic Web Anwendungen

Betreuender Prüfer: Prof. Dr. Kai von Luck

**Gerrit Diederichs**

**Thema der Ausarbeitung Seminar Ringvorlesung**

Bausteine für Semantic Web Anwendungen

**Stichworte**

Semantic Web, RDF, Ontologien, OWL, SPARQL, SWRL, Inferenz

**Kurzzusammenfassung**

Das Semantic Web wird häufig als das "Web der nächsten Generation" bezeichnet. Die Idee ist Strukturen zu schaffen, in der Ressourcen des Internet um maschinenverwertbare Semantik angereichert werden. So soll erreicht werden Prozesse im Internet besser zu autonomisieren und zu automatisieren. Dieser Artikel liefert einen Überblick über die Technologien des Semantic Web. Sofern vorhanden werden die Standards vorgestellt, respektive aussichtsreiche Kandidaten für Standards. Schließlich wird ein Projekt das sich mit der Integration von externen Unternehmensdaten unter Verwendung von Semantic Web Technologien in einem fiktiven Ferienclub-Szenario befasst vorgestellt. Aus dem Projektszenario wird ein Vorschlag für eine Master-Thesis entwickelt.

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Einleitung</b>   | <b>1</b>  |
| <b>2</b> | <b>Technologien des Semantic Web</b>  | <b>2</b>  |
| 2.1      | RDF und Ontologiesprachen . . . . .   | 2         |
| 2.1.1    | Resource Description Framework RDF . . . . .  | 2         |
| 2.1.2    | Ontologiesprachen . . . . .   | 3         |
| 2.1.3    | Fazit . . . . .   | 3         |
| 2.2      | RDF-Querysprachen und Regelsprachen . . . . .   | 4         |
| 2.2.1    | RDF Anfragesprachen . . . . .   | 4         |
| 2.2.2    | Regelsprachen . . . . .   | 4         |
| 2.3      | Inferenz . . . . .  | 5         |
| 2.3.1    | Aufgaben von Inferenz . . . . .   | 5         |
| 2.3.2    | Inferenzmaschinen . . . . .   | 6         |
| 2.3.3    | DIG-Interface . . . . .   | 7         |
| 2.3.4    | Fazit . . . . .   | 8         |
| <b>3</b> | <b>Integration externer Dienste mit Semantic Web Technologien in einem fiktiven Ferienclub-Szenario</b> | <b>9</b>  |
| 3.1      | Beispielarchitektur für das Ferienclub Szenario . . . . .   | 9         |
| <b>4</b> | <b>Ausblick für eine Masterarbeit</b>   | <b>13</b> |

# 1 Einleitung

Die Information wie sie im heutigen Internet vorliegt ist vorrangig für die Verarbeitung durch den Menschen optimiert. Sie liegt unstrukturiert als Verlinkung von Ressourcen vor. Mit zunehmender Nutzung des Internet wird die zufriedenstellende Erschließung von Information - zumal sie unstrukturiert vorliegt - immer schwieriger. In seiner heutigen Form wird das Internet dieser Herausforderung nicht mehr gerecht. Die größten Problemfelder sind hierbei:

- Informationssuche  
Heutige Suchmaschinen basieren auf der Suche nach Schlagworten. Dieser Ansatz hat mehrere Schwachpunkte. Weder werden Schlagworte gleicher oder ähnlicher Bedeutung (Synonyme) ausreichend berücksichtigt, noch werden Mehrdeutigkeiten (Homonyme) erkannt. Somit ist das Suchergebnis häufig unbrauchbar.
- Informationsintegration  
Unternehmen nutzen das Internet um Dienstleistungen anzubieten und ihre Geschäftsprozesse abzubilden. Die Integration von Daten zwischen Unternehmen ist hierbei oft schwierig, da diese unterschiedlich strukturiert sind.
- Informationsextraktion  
Die Extraktion von Information aus unstrukturierten oder teilstrukturierten Daten ist aufwendig. Ein Beispielprojekt hierfür ist Web Fountain vom IBM Research Center in Almaden.

Das Semantic Web soll eine zusätzliche Komponente zum derzeitigen Web sein, die Internetressourcen strukturiert und mit einer wohldefinierten, formalen Semantik versieht, um so diese Ressourcen durch Maschinen verarbeitbar zu machen. Die Zusammenhänge zwischen den Ressourcen werden dabei logisch - in Form von Ontologien - beschrieben. Mittels Inferenzmaschinen können dann Schlussfolgerungen (Inferenz) bezüglich der Validität einer Ressource im Kontext einer bestimmten Anfrage beziehungsweise neue Information über eine Ressource gewonnen werden. Tim Berners-Lee hat seine Vision vom Semantic Web folgendermaßen beschrieben:

The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. (Timothy Berners-Lee, 2001)

Mithilfe des Semantic Web hofft man die oben genannten Probleme des heutigen Internet zu überwinden. Im Rahmen dieser Ausarbeitung soll ein Überblick über die Technologien des Semantic Web gegeben werden (Kapitel 2). Ausserdem wird die Entwicklung einer Semantic Web Anwendung im Rahmen eines Hochschulprojektes an der HAW Hamburg vorgestellt (Kapitel 3). Abschließend wird ein Ausblick für eine mögliche Masterarbeit in diesem Bereich gegeben (Kapitel 4).

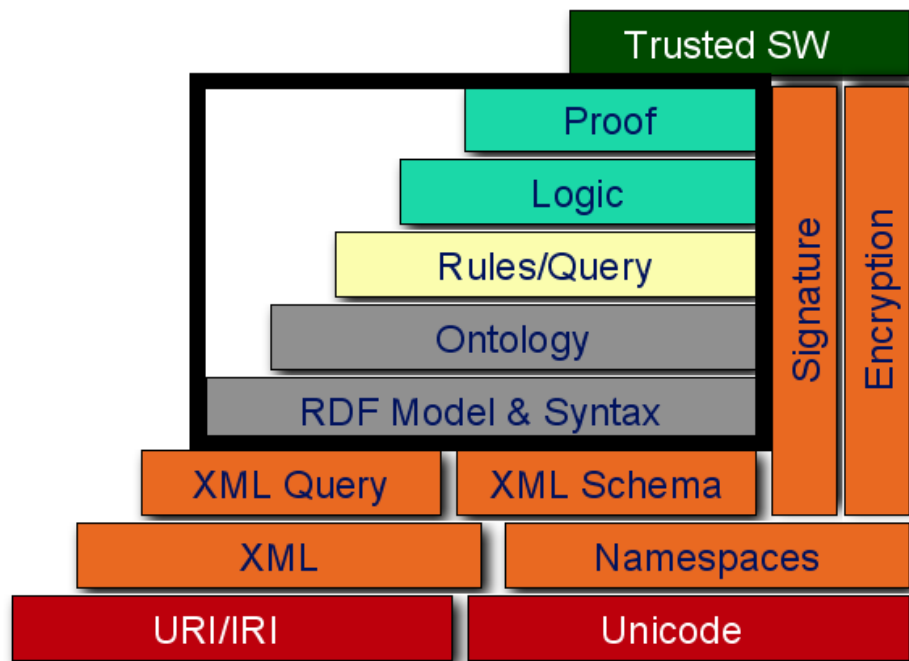


Abbildung 1: Semantic Web Schichtenmodell

## 2 Technologien des Semantic Web

Unter Führung des W3C ist eine Initiative entstanden, die sich mit der technischen Umsetzung des Semantic Web befasst. Ziel ist die Entwicklung von Standards mit denen sich semantische Zusammenhänge im Web repräsentieren lassen und maschinell verarbeitet werden können. Im Rahmen der Initiative ist ein Schichtenmodell entstanden, das den Zusammenhang der verschiedenen Technologien innerhalb des Semantic Web beschreibt. Dieses Kapitel stellt die in Abbildung 1 schwarz umrandeten Schichten respektive die Technologien vor.

### 2.1 RDF und Ontologiesprachen

Die in diesem Abschnitt beschriebenen Schichten des Semantic Web Modells bilden die Grundlage des Semantic Web, insofern sie die Technologie zur Verfügung stellen Internetressourcen zu beschreiben und Domänenwissen zu modellieren.

#### 2.1.1 Resource Description Framework RDF

Das Resource Description Framework (RDF) ist eine W3C Spezifikation ((RDFW3C)). Es beschreibt ein Metadatenmodell für beliebige Ressourcen. Das Modell besteht dabei aus Aussagen über Ressourcen. Eine RDF-Aussage ist aufgebaut als ein Triple mit Subjekt, Prädikat und Objekt. Dabei kann jeder Bestandteil des Triples eine Ressource sein. Eine RDF-Aussage lässt sich als gerichteter Graph interpretieren. Das Objekt einer Aussage kann wie-

derum eine Aussage sein. Man spricht hierbei von Reification. RDF-Graphen werden mittels einer XML-basierten Notation serialisiert. Mit RDF können Ressourcen und Beziehungen zwischen Ressourcen also beschrieben werden.

### 2.1.2 Ontologiesprachen

Ontologien dienen zur formalen Repräsentation von Wissen. Eine häufig zitierte Definition des Begriffes Ontologie im informationstechnischen Kontext ist die von Gruber:

An ontology is a formal, explicit specification of a shared conceptualization.  
(Gruber, 1993)

Ontologiesprachen sind die Werkzeuge um Ontologien zu modellieren, also die Wissensdomäne für einen bestimmten Kontext zu formalisieren und logisch zu beschreiben. Das Paradigma der Ontologiemodellierung ähnelt dem der objektorientierten Softwaremodellierung. Eine Ontologie besteht aus verschiedenen Konzepten (auch Klassen genannt), die man mit Klassen in der Objektorientierung vergleichen kann, und Individuen, die Instanzen dieser Konzepte sind. Konzepte haben Eigenschaften und darüber hinaus können sie wiederum in einer Beziehung zu anderen Konzepten der Ontologie stehen. Neben dem Erstellen eigener Ontologien können Ontologiesprachen auch dazu genutzt werden unterschiedliche Ontologien aufeinander zu mappen. Dies geschieht über Äquivalenzaussagen bezüglich der Konzepte der zu mappenden Ontologien. Die Ontologiesprache Web Ontology Language (OWL) ist eine Recommendation des W3C für das Semantic Web und stellt somit einen quasi Standard dar. OWL ist aus DAML+OIL hervorgegangen. OWL baut auf RDF beziehungsweise RDF-Schema auf. Ein Individuum eines Konzeptes in OWL wird beispielsweise über `<rdf:type>` erzeugt. OWL verfügt über drei unterschiedliche Ausdrucksstärken bezüglich der Modellierung von Ontologien. OWL Lite < OWL Description Logic < OWL Full. Hierbei gilt, dass die Mächtigkeit der Sprachkonstrukte von OWL Lite bis hin zu OWL Full ansteigt und der Sprachumfang der jeweils mächtigeren Sprache die der darunterliegenden voll enthält. Je mächtiger die verwendeten Sprachkonstrukte sind, desto aufwändiger ist ein späteres Reasoning (Inferenz) über die Ontologie. Hierbei gilt, dass OWL Description Logic noch entscheidbar ist, OWL Full jedoch Konstrukte enthalten kann die nicht mehr entscheidbar sind. Generell muss man bei der Modellierung von Ontologien stets einen Kompromiss zwischen Ausdrucksstärke und Entscheidbarkeit der Ontologie finden. Meist werden Ontologien höchstens mit den Konstrukten der Beschreibungslogik (Description Logic) - einem entscheidbaren Fragment der Prädikatenlogik - modelliert, um einen vernünftigen Kompromiss zu erzielen.

### 2.1.3 Fazit

RDF bildet zusammen mit einer Ontologiesprache - wie beispielsweise den Standardsprachen RDF-Schema oder OWL - die Möglichkeit eine Wissensbasis (Knowledgebase) über (Internet-)Ressourcen aufzubauen. Die Individuen - also die Ressourcen - werden dabei in RDF beschrieben. Das domänenspezifische Vokabular für diese RDF-Modelle wird mittels einer Ontologiesprache beschrieben in Form von Konzepten und ihren logischen Zusammenhängen. Auf dieser Basis lässt sich nun mittels Inferenzmaschinen neues Wissen

über die Ressourcen herleiten oder die Konsistenz der Wissensbasis sicherstellen. Ausführliche Einführungen in RDF bzw. RDFS und OWL lassen sich u.a. hier finden ((RDFW3C), (OWLW3C), (Michael C. Daconta, 2003), (Thallmair, 2004), (Wleklinski, 2003)).

## 2.2 RDF-Querysprachen und Regelsprachen

Im vorherigen Kapitel wurden Technologien vorgestellt um Wissensbasen für das Semantic Web zu erstellen. In diesem Kapitel werden nun Abfragesprachen vorgestellt, um diese Wissensbasen nach Information zu durchsuchen, und Regelsprachen, um Wissensbasen um zusätzliche Regeln zu erweitern.

### 2.2.1 RDF Anfragesprachen

Semantic Web Sprachen wie RDF, RDFS oder OWL werden überwiegend in XML repräsentiert, daher wäre es naheliegend die Abfrage von Wissensmodellen die auf diesen Sprachen basieren über XML-Abfragesprachen wie beispielsweise XQuery oder XPath zu realisieren. Diese Abfragesprachen haben jedoch einen entscheidenden Nachteil. Sie können lediglich Anfragen bezüglich der Struktur (Syntax) der in XML formulierten Daten stellen. Um die Möglichkeiten einer Wissensbasis aber wirklich nutzen zu können und das „Wissen“ abfragen zu können muss die Querysprache in der Lage sein das Datenmodell von RDF(S)/OWL und die semantischen Eigenschaften des Vokabulars zu verstehen. Daher werden momentan mehrere Vorschläge zur Entwicklung einer Standardsprache für die Abfrage von RDF-Modellen (und den jeweiligen Konzepten formuliert in einer Ontologiesprache) vorangetrieben. Einige Beispiele sind RQL, SPARQL, SeRQL, RDQL etc. Die Simple Protocol and RDF Query Language (SPARQL) ist aktuell ein W3C Working Draft und hat somit vielleicht die besten Aussichten als Sieger aus den verschiedenen Standardisierungsbemühungen hervorzugehen. Weiterführende Artikel zu Anfragesprachen für das Semantic Web gibt es u.a. hier (Joaachim Baumeister, 2005) oder hier (Peter Haase). Information zur Anfragesprache SPARQL findet man u.a. hier (SPARQLW3C) .

### 2.2.2 Regelsprachen

Im in Abbildung 1 dargestellten Schichtenmodell setzen die Regeln oberhalb der Schicht der Ontologie auf. Regeln bauen in diesem Kontext auf der darunterliegenden Ontologie auf die i.d.R. in OWL formuliert wird. In OWL werden implizit bereits Regeln formuliert, wie beispielsweise

Eine Frau ist Unterkonzept (respektive Unterklasse) eines Konzeptes Person  
(Subsumption).

Regelsprachen, die auf den modellierten Beziehungen einer Ontologie aufsetzen ermöglichen neue, komplexere Konstrukte und erhöhen damit die Ausdrucksstärke der Ontologie, sowie die möglichen Schlussfolgerungen. Regeln werden explizit deklariert und nicht implizit modelliert. Es gibt mehrere Regelsprachen die im Kontext des Semantic Web verwendet

werden können. Im folgenden wird die Semantic Web Rule Language (SWRL) kurz vorgestellt. SWRL ist eine W3C Member Submission (Member, 2004). Es ist eine Kombination von OWL Lite/DL mit Datalog RuleML. Eine SWRL Regel ist als Hornklausel bestehend aus atomaren Aussagen - die unäre OWL Konzepte, binäre OWL Prädikate oder spezielle OWL-Prädikate (*sameAs*(X,Y), *differentFrom*(X,Y), *builtIn*(X,Y)) enthalten können - aufgebaut. Sie besteht aus einem Regelkopf, der genau dann erfüllt ist wenn alle atomaren Aussagen des Regelkörpers erfüllt sind (Dies ergibt sich aus der Definition der Hornklausel als Disjunktion von Atomaren Aussagen mit genau einer nicht-negierten Aussage die sich umformulieren lässt zu einer Konjunktion von atomaren Aussagen, die eine atomare Aussage (den Regelkopf) implizieren). Eine beispielhafte SWRL Regel wäre:

Ein Onkel ist der Bruder eines Elternteils.

$\forall X,Y,Z : hasParent(X,Y) \wedge hasBrother(Y,Z) \rightarrow hasUncle(X,Z)$

Regelsprachen erhöhen die Ausdrucksmächtigkeit einer Ontologie. Damit wird aber die Berechenbarkeit bzw. Entscheidbarkeit verschlechtert. OWL DL mit SWRL ist nicht mehr entscheidbar.

## 2.3 Inferenz

Die Motivation Zusammenhänge im Semantic Web logisch zu beschreiben leitet sich aus der Möglichkeit ab über diese dann logische Schlüsse (Inferenz engl.:Reasoning) ziehen zu können. In den folgenden Kapiteln werden die Aufgaben von Inferenz, Inferenzmaschinen und eine weit verbreitete Schnittstelle zu einer bestimmten Art der Inferenzmaschinen (derer die auf Description Logic beruhen) vorgestellt.

### 2.3.1 Aufgaben von Inferenz

Die Inferenz deckt die Schichten „Logic“ und „Proof“ des in Abbildung 1 dargestellten Schichtenmodells ab. Im Einzelnen umfassen diese Aufgaben im Wesentlichen folgendes:

- Ermittlung abgeleiteter Bedingungen (Deduktion)  
Die Erschaffung „neuen Wissens“ aus einer gegebenen Ontologie. Eine neue Bedingung kann aus der gegebenen Ontologie abgeleitet werden, wenn sie in jeder legalen Beschreibung der Welt gilt.
- Ermittlung der Konsistenz von Konzepten  
Ist die logische Beschreibung der Konzepte einer Ontologie konsistent? Ein Konzept (eine Klasse) ist inkonsistent, wenn es in jeder legalen Beschreibung der Welt die leere Menge bezeichnet, d.h. es kann keine Individuen für dieses Konzept geben.
- Klassifikation  
Die Klassifikation von Individuen oder neuer Konzepte innerhalb einer Wissensbasis.
- Ermittlung der Äquivalenz von Konzepten  
Auf diese Weise können unterschiedliche Ontologien sofern sie Ähnlichkeiten haben, d.h. einen ähnlichen Sachverhalt beschreiben, aufeinander gemappt werden.



Inferenz wird also in erster Linie genutzt um Deduktion über Wissenbasen durchzuführen, ihre Konsistenz zu gewährleisten oder Äquivalenzen zu erkennen. Die „Engines“ die diese Inferenz durchführen sind die im nächsten Kapitel vorgestellten Inferenzmaschinen.

### 2.3.2 Inferenzmaschinen

Inferenzmaschinen sind salopp gesagt das Herz des Semantic Web, denn sie ermöglichen erst die Nutzung der Vorteile die eine semantische Beschreibung von Daten oder allgemeiner Ressourcen bringt. Inferenzmaschinen werden je nach Mächtigkeit der Sprachkonstrukte die sie verarbeiten können und der Kalküle die sie dafür benutzen in Komplexitätsklassen eingeteilt. Hierbei wird zwischen vier Komplexitätsklassen unterschieden:

- Higher Order Logic
- Full First Order Logic (Prädikatenlogik)
- Description Logic (Beschreibungslogik)
- Logic Programming (z.B. Horn Klauseln)

Die ersten beiden Komplexitätsklassen sind nicht mehr entscheidbar, daher werden Ontologien meist höchstens mit Beschreibungslogik (z.B. OWL DL) modelliert. Entsprechend werden meist Inferenzmaschinen verwendet, die auf Basis von Beschreibungslogik arbeiten. Beispiele für solche Inferenzmaschinen sind RACERPro, Pellet oder FaCT++.

Wenn eine Inferenzmaschine neues Wissen aus einer gegebenen Wissensbasis ableitet fügt sie dieses Wissen der Wissensbasis durch sogenannte „virtuelle Triple“ hinzu. Darüberhinaus gewährleistet die Inferenzmaschine die Konsistenz der Wissensbasis, z.B. durch den Beweis der Erfüllbarkeit von Konzepten.

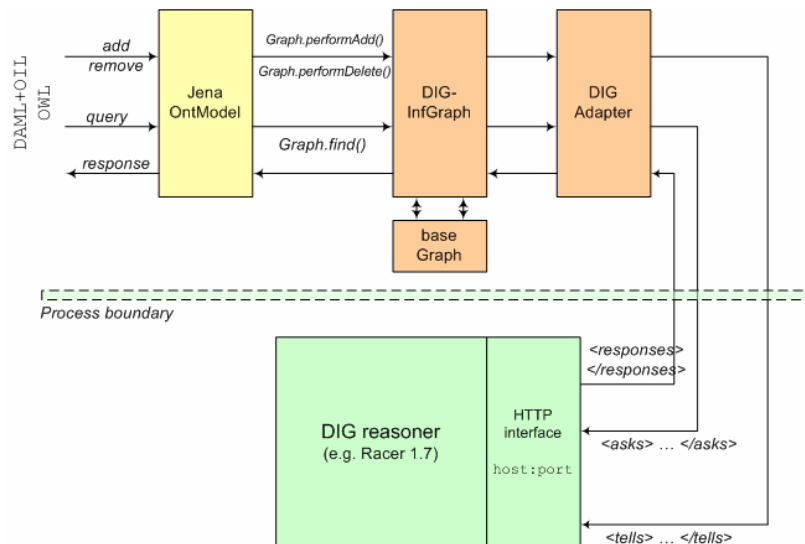


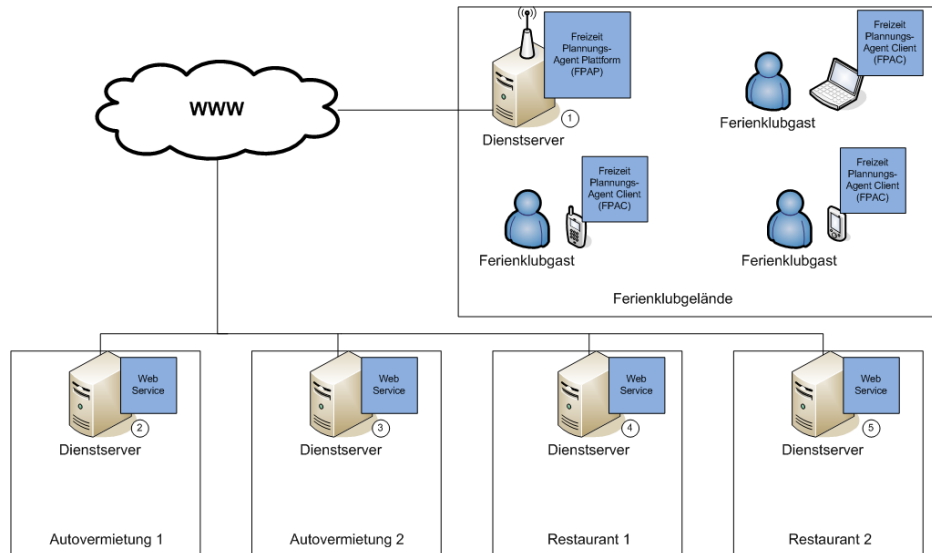
Abbildung 2: DIG Interface mit Jena

### 2.3.3 DIG-Interface

Das DIG-Interface ist aus den Bemühungen der DL Implementation Group (DIG) hervorgegangen. Es spezifiziert ein Interface für die Anbindung von Inferenzmaschinen an entsprechende Programme, die Inferenz nutzen wollen (z.B. Ontologieeditoren oder Agenten). Das DIG Interface basiert auf HTML und XML. Es implementiert dabei ein einfaches tell-ask-response-Protokoll. Mit der Funktion tell können Programme dem Reasoner von einer Ontologie „erzählen“, d.h. die Ontologie dem Reasoner beschreiben. Mit der Funktion ask können Anfragen bezüglich der Ontologie gestellt werden (Deduktion, Konsistenz). Die Antwort erfolgt jeweils mit einer Response. Um per DIG-Interface Ontologien gegenüber dem Reasoner beschreiben zu können, implementiert das Protokoll selbst wiederum eine Ontologiesprache, die auf Beschreibungslogik beruht. Folgende Abbildung gibt einen Überblick über die Kommunikation via DIG-Interface mit dem Semantic Web Framework Jena2: Das DIG-Interface stellt zwar keinen Standard dar wird aber von den meisten DL Inferenzmaschinen und Semantic Web Werkzeugen wie Jena 2 oder Protégé unterstützt. Ein Nachteil ist, dass die Ontologie zunächst wieder in die Ontologiesprache des DIG-Interface übersetzt werden muss. Hierbei gibt es nach ersten Erfahrungen des Autors beispielsweise keine vollständige Kompatibilität zu den Konstrukten von OWL DL.

#### 2.3.4 Fazit

Die Möglichkeit der Inferenz ist eine der zentralen Komponente des Semantic Web. Es gibt bereits mehrere Inferenzmaschinen, die Ontologiesprachen wie OWL unterstützen. Meist arbeiten sie auf Basis von Beschreibungslogik (Description Logic). Das DIG-Interface wird von den meisten DL-Inferenzmaschinen unterstützt. Weiterführende Information zum Thema Inferenz auf Basis von Description Logic gibt es hier ((DescriptionLogics), (Klusch, 1998), (Horrocks), (Beckstein)) .



### 3 Integration externer Dienste mit Semantic Web Technologien in einem fiktiven Ferienclub-Szenario

Im Rahmen eines Projektes an der HAW Hamburg das sich mit verschiedensten, verteilten Anwendungen im Rahmen eines fiktiven Ferienclub-Szenario beschäftigt arbeitet der Autor zum Zeitpunkt des Schreibens dieser Ausarbeitung an einer Architektur nebst Beispielanwendung, die die Integration mehrerer externer Dienstleistungen mittels Semantic Web Technologien realisiert. Abbildung 3 beschreibt das grundsätzliche Szenario: Es gibt mehrere externe Dienstleister wie Autovermietungen oder Restaurants, die ihr Angebot über Dienstserver zur Verfügung stellen. Dabei wird davon ausgegangen, dass die verfügbaren Angebote in Wissensbasen vorliegen. Sie also durch Ontologien beschrieben werden. Der Ferienclub hat eigene Ontologien, die ein eigenes Modell für die entsprechenden Dienste enthält. Ziel ist es zunächst die jeweiligen Ontologien zu modellieren und sie dann zu mappen. Auf Basis dieses Mappings werden die verschiedenen Dienstleister in eine Semantic Web basierte Anwendung integriert. Diese ist zunächst als einfache Webanwendung geplant, um Gästen des Ferienclub die entsprechenden Angebote die sie interessieren herauszusuchen.

### 3.1 Beispielarchitektur für das Ferienclub Szenario

Die grundsätzliche Architektur für die sich der Autor zur Lösung des Problems entschieden hat ist in Abbildung 4 zu sehen. Ein zentraler Punkt bei der Entwicklung von Semantic Web Anwendungen ist das Ontologie Engineering. Dieses umfasst sowohl das Erstellen der Ontologie sowie auch das Mapping mehrerer Ontologien und das Verifizieren der Konsistenz der gesamten Wissensbasis (Ontologie und Individuen). Im Projekt wird hierfür der Ontologieedi-

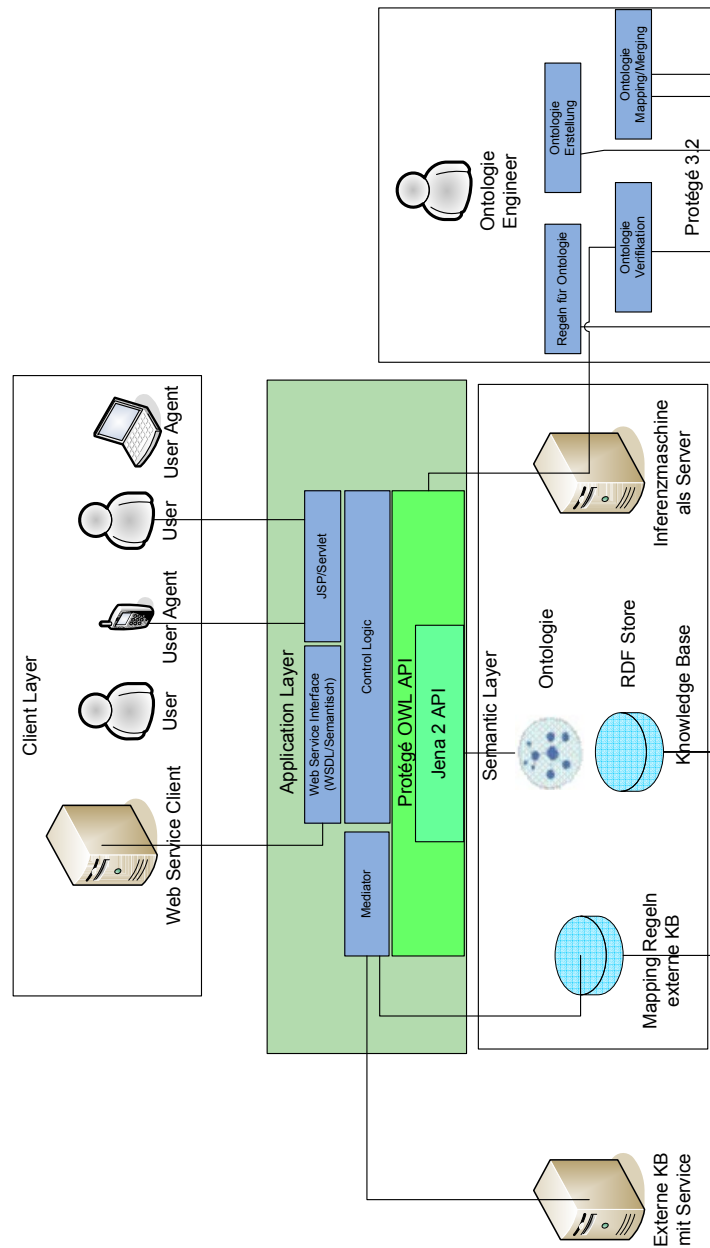


Abbildung 4: Architektur Projekt

tor Protégé in der Version 3.2 verwendet. Protégé ist ein Open Source Projekt der Stanford Universität (Protégé). Über Plugins ist es möglich einfach Funktionalitäten zu Protégé hinzuzufügen. Protégé bietet in der aktuellen Version 3.2 jegliche Funktionalität, die man für das Ontologie Engineering braucht. Es gibt Plugins für OWL (Protégé-OWL Plugin), für Ontologie Mapping bzw. Merging (Prompt), sowie eine DIG-Schnittstelle, um Inferenzmaschinen anzubinden. Es gibt noch viele weitere nützliche Plugins, die an dieser Stelle jedoch nicht in Gänze aufgezählt werden sollen. Mittels der Protégé-OWL API kann man auch Semantic Web Anwendungen entwickeln. Die Protégé-OWL API basiert auf der API des Semantic Web Framework Jena 2 ((Jena2)) . Als Inferenzmaschine wird im Projekt RACERPro (RACER) verwendet eine Inferenzmaschine auf Basis von Description Logic. Wie bereits eingangs von Kapitel 3 erwähnt liegt der Fokus der Anwendung auf der Integration externer Dienstleistungen mittels Semantic Web Technologien. Abbildung 5 zeigt eine detailliertere Sicht der Projektarchitektur. Es gibt zwei wesentliche Komponenten: den Search Service, der die Wissensbasis des Ferienclubs nach geeigneten Angeboten durchsucht und Anfragen zusätzlich an die Mediator Komponente delegiert. Diese verfügt über Mapping Regeln, die halbautomatisch mit Protégé-Prompt erstellt werden. Entsprechend der Mapping Regeln werden die Anfragen für die Wissensbasen der externen Dienste aufbereitet und die Wissensbasen abgefragt. Die Wissensbasen sind jeweils an die Inferenzmaschine RACERPro angebunden.

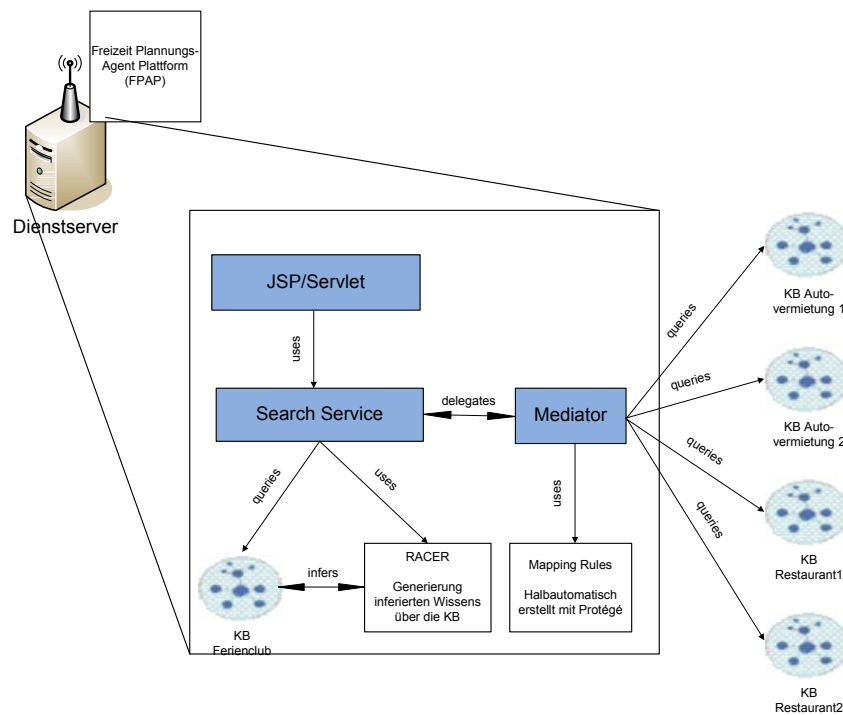


Abbildung 5: Detaillierte Sicht auf die Ferienclub Komponente

## 4 Ausblick für eine Masterarbeit

Der Autor hat die Absicht die ersten gesammelten Erfahrungen mit der Entwicklung von Semantic Web Anwendungen aus dem Projekt im Rahmen einer Masterarbeit zu vertiefen. Zu diesem Zweck soll das Projektszenario ausgebaut werden. Die Erweiterungen sind angelehnt an Granite Nights (GraniteNights) einem inzwischen abgeschlossenen Projekt der Universität von Aberdeen. Der Suchdienst soll zu einem Ausflugsplaner ausgebaut werden. Die semantische Infrastruktur wird um Benutzerprofile und einen Scheduler, der mögliche Tagesausflüge generiert, erweitert. Mit der Zeit wird das Benutzerprofil auf Basis der bisherigen Ausflüge eines Benutzers verfeinert. Abbildung 6 skizziert die geplanten Erweiterungen. Neben einer weiteren Vertiefung respektive Evaluierung der im Projekt verwendeten Semantic Web Technologien werden hierbei Erkenntnisse im Bereich der semantischen Modellierung von Benutzerprofilen sowie dem constraint-based Scheduling Ziele der Arbeit sein. Darüberhinaus wären Überlegungen sinnvoll Semantic Web Technologien zu nutzen, um Context-Awareness insbesondere ortsbezogene Information einzubauen und die Anwendung zu einem Push-Service für mobile Clients (z.B. PDAs) auszubauen. Diese Ausbaustufe könnte aber im Rahmen eines zu erstellenden Prototypen zu umfangreich sein.



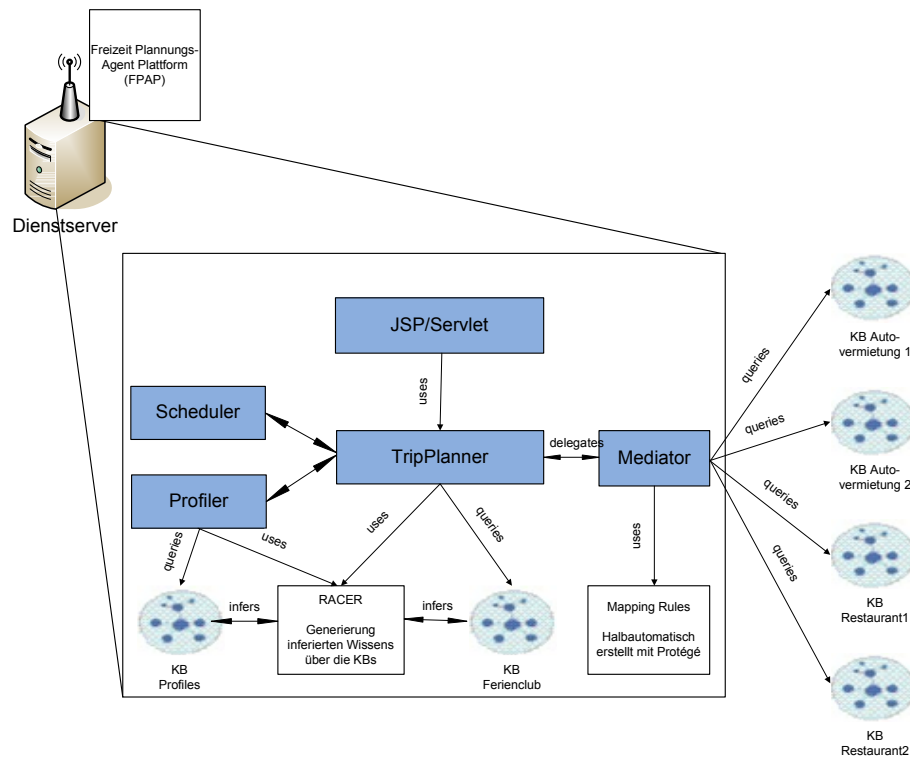


Abbildung 6: Geplante Erweiterungen für Master Thesis

## Literatur

- [Beckstein] BECKSTEIN, Clemens: *Einführung in die KI*. – URL <http://www.minet.uni-jena.de/www/fakultaet/beckstein/eki-ws0405/beschrlog.ps>
- [DescriptionLogics] DESCRIPTIONLOGICS, Homepage: *Description Logics*. – URL <http://dl.kr.org/>
- [DIG] DIG: *Description Logic Implementation Group*. – URL <http://dl.kr.org/dig/>
- [GraniteNights] GRANITENIGHTS, Universität A.: *Granite Nights*. – URL <http://www.csd.abdn.ac.uk/research/AgentCities/GNInfo/>
- [Gruber 1993] GRUBER, Thomas R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In: *International Journal Human-Computer Studies* 43 (1993), März, S. 907–928. – URL <http://tomgruber.org/writing/onto-design.pdf>
- [Horrocks] HORROCKS, Ian: *Logical Foundations for the Semantic Web*. – URL <http://www.cs.man.ac.uk/~horrocks/Slides/glasgow.pdf>
- [Jena2] JENA2: *Jena A Semantic Web Framework for Java*. – URL <http://jena.sourceforge.net/>
- [Joachim Baumeister 2005] JOACHIM BAUMEISTER, Dietmar S.: Anfragesprachen für das Semantic Web. In: *Informatik Spektrum* 28 (2005), Februar
- [Klusch 1998] KLUSCH, Matthias: *Kooperative Informationsagenten im Internet*. Kap. 4, Verlag Dr. Kovac, 1998
- [Member 2004] MEMBER, Submission W.: *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Mai 2004. – URL <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- [Michael C. Daconta 2003] MICHAEL C. DACONTA, Kevin T. S.: *The Semantic Web*. Kap. 5 & 8, Wiley Technology Publishing, 2003
- [OWLW3C] OWLW3C: *Web Ontology Language*. – URL <http://www.w3.org/2004/OWL/>
- [Peter Haase] PETER HAASE, Andreas Eberhart Raphael V.: *A Comparison of RDF Query Languages*. – URL <http://www.aifb.uni-karlsruhe.de/WBS/pha/rdf-query/>
- [Protégé] PROTÉGÉ: *The Protégé Ontology Editor and Knowledge Acquisition System*. – URL <http://protege.stanford.edu/>

- [RACER ] RACER: *Racer Systems GmbH & Co. KG.* – URL <http://www.racer-systems.com/de/index.phtml>
- [RDFW3C ] RDFW3C: *Resource Description Framework (RDF).* – URL <http://www.w3.org/RDF/>
- [SPARQLW3C ] SPARQLW3C: *SPARQL Query Language for RDF.* – URL <http://www.w3.org/TR/rdf-sparql-query/>
- [Thallmair 2004] THALLMAIR, Oliver: *Personalisierte Informationsagenten im Semantic Web*, Technische Universität München - Fakultät für Informatik, Diplomarbeit, August 2004
- [Timothy Berners-Lee 2001] TIMOTHY BERNERS-LEE, Ora L.: The Semantic Web. In: *Scientific American* (2001), Mai
- [Wleklinski 2003] WLEKLINSKI, Fabian: *Suche im Semantic Web*, Universität Frankfurt am Main, Diplomarbeit, November 2003