

# Projekt Semantic Web

## Abschlussbericht

15.08.2004

Frederik Bäßmann <fbaess@gmx.de>  
Richard Cyganiak <richard@cyganiak.de>  
Anja Jentsch <anja@anjeve.de>  
Franziska Liebsch <flibsch@inf.fu-berlin.de>  
Konstantin Zacharov <zacharov@inf.fu-berlin.de>

Freie Universität Berlin Institut für Informatik  
Fachbereich Mathematik und Informatik  
AG Netzbasierte Informationssysteme - Prof. Dr.-Ing. Robert Tolksdorf

# 1 Abstract

Das Projekt Semantic Web war eine Lehrveranstaltung am Institut für Informatik der Freien Universität Berlin im Sommersemester 2004. Ziel war die Erstellung eines Online-Informationssystems auf Grundlage von Semantic Web-Technologien und die Entwicklung einer Ontologie zur Beschreibung von Hochschulen und Studiengängen. Der vorliegende Bericht dokumentiert Planung, Verlauf und Ergebnisse des Projekts und diskutiert die Frage nach dem Nutzen von Semantic Web-Technologien für ein solches Vorhaben.

## Inhaltsverzeichnis

<b>1 Abstract</b>	<b>1</b>
<b>2 Einleitung</b>	<b>1</b>
<b>3 Projektplan und Organisation</b>	<b>2</b>
3.1 Projektplan . . . . .	2
3.2 Organisation . . . . .	3
<b>4 Umsetzung</b>	<b>4</b>
4.1 Projektstart . . . . .	4
4.2 Entwicklung des Vokabulars durch die AG Modellierung . .	5
4.3 FOAF-Informationssystem . . . . .	7
4.4 Organisation innerhalb der AG Programmierung . . . . .	7
4.5 Die Teilaufgaben der AG Programmierung thematisch . . . .	7
4.6 Architektur des Online-Informationssystems . . . . .	8
4.7 UseCases . . . . .	9
4.8 Graphische Darstellung des Online-Informationssystems . .	10
4.9 Umsetzung des Online-Informationssystems . . . . .	11
<b>5 Ergebnisse</b>	<b>11</b>
5.1 Beschreibung des Vokabulars . . . . .	12
5.2 Fazit: Was nutzen die Semantic Web-Technologien? . . . . .	14

## 2 Einleitung

Das Projekt Semantic Web war eine Lehrveranstaltung am Institut für Informatik der Freien Universität Berlin im Sommersemester 2004, veranstaltet von Prof. Robert Tolksdorf und der AG Netzbaasierte Informationssysteme. Die Lehrveranstaltung wurde als Projekt durchgeführt. Aufgabe der Teilnehmer war die Erstellung eines Online-Informationssystems, das die Studiengänge der Berliner Hochschullandschaft unter besonderer Berücksichtigung der Informatik erfasst. Das System sollte mit Hilfe von Technologien

des Semantic Web erstellt werden. Ein ähnlich angelegtes Projekt<sup>1</sup> wurde bereits im Jahr 2003 von Prof. Tolksdorf durchgeführt.

Anders als bei den meisten Projektlehrveranstaltungen sollten die Teilnehmer selbst die Organisation übernehmen, während Prof. Tolksdorf sich auf die Rolle des Auftraggebers beschränkte. Fünf der 21 Teilnehmer bildeten eine Projektleitungsgruppe. Ihre Aufgaben umfassten die Entwicklung eines Projektplans, die Verteilung von Arbeitspaketen und Kontrolle deren Erfüllung. Weiterhin sollte der Projektverlauf dokumentiert werden. Dazu entstand unter anderem ein Zwischenbericht<sup>2</sup>. Der vorliegende Bericht beschreibt Verlauf und Ergebnisse des gesamten Projekts und schließt die Arbeit dieser Gruppe ab.

Der Bericht beginnt mit einer Darstellung des Projektauftrags und des Projektplans. Es folgt ein Abschnitt zur Umsetzung, der einige der getroffenen Designentscheidungen motiviert. Der Bericht schließt mit einer Darstellung der Projektergebnisse. Dabei gehen wir besonders auf die viel diskutierte Frage nach dem „Mehrwert“ von Semantic Web-Technologien ein.

### 3 Projektplan und Organisation

Das Projekt sollte ein Online-Informationssystem (im Folgenden „OIS“ abgekürzt) auf Grundlage von Technologien des Semantic Web erstellen. Im System sollten die Studiengänge der Berliner Hochschullandschaft unter besonderer Berücksichtigung der Informatik erfasst sein. Es sollte die Studiengänge inhaltlich erschließ- und vergleichbar machen und als „Berliner Hochschulkompass Informatik“ geeignet sein.

Der gesamte Projektauftrag ist auf der Webseite der Lehrveranstaltung<sup>3</sup> nachzulesen.

Das Projekt wurde extern begutachtet. Als Gutachter fungierten die Mitglieder einer Arbeitsgruppe des Seminars XML und Semantic Web, das zeitgleich an der Technischen Universität Berlin stattfand. Die Gutachter verfassten einen Zwischenbericht<sup>4</sup>, der den Stand vom 30. Juni 2004 (drei Wochen vor Projektende) widerspiegelt.

#### 3.1 Projektplan

Die Arbeitsgruppe Projektmanagement entwickelte einen Projektplan<sup>5</sup>, der die Entwicklung folgender zentraler Komponenten vorsah:

---

<sup>1</sup>[http://projects.mi.fu-berlin.de/semweb/docs/Semweb\\_03\\_Projekttdokumentation.pdf](http://projects.mi.fu-berlin.de/semweb/docs/Semweb_03_Projekttdokumentation.pdf)

<sup>2</sup>[http://projects.mi.fu-berlin.de/semweb/docs/Meilensteinbericht\\_2004-06-09.html](http://projects.mi.fu-berlin.de/semweb/docs/Meilensteinbericht_2004-06-09.html)

<sup>3</sup>[http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/04/P\\_SW/](http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/04/P_SW/)

<sup>4</sup><http://insel.cs.tu-berlin.de/pub/bscw.cgi/d1637/zwischenbericht.pdf>

<sup>5</sup><http://projects.mi.fu-berlin.de/semweb/wiki/?page=ProjektPlan>

- Ein Vokabular zur Beschreibung von Hochschulen und Studiengängen, formuliert in RDFS und/oder OWL.
- RDF-Daten, welche die Landschaft der Informatik an Berliner Hochschulen beschreiben.
- Ein Informationssystem, das die übersichtliche Navigation und Suche innerhalb der Daten mittels Webbrowser ermöglicht.

Zusätzlich sollen folgende Hilfskomponenten erstellt werden:

- Eine Referenzdokumentation dieses Vokabulars als HTML-Seite(n).
- Ein kurzes Tutorial, das die Erfassung typischer Daten mit diesem Vokabular als RDF anhand von Beispielen erklärt. Tutorial und Referenzdokumentation dokumentieren das erstellte Vokabular, erleichtern die Kommunikation zwischen Modellierern und Programmierern, und machen die Daten leichter für Dritte nutzbar.
- Eine Webschnittstelle zum Erstellen von RDF-Daten; Profil-Daten werden in HTML-Formularen eingegeben und als Ausgabe erhält man eine entsprechende RDF-Datei. Wir nennen diese Komponente „Hochschul-a-Matic“ (nach dem Vorbild der FOAF-a-matic<sup>6</sup>).
- Einen Verzeichnisdienst, bei dem die URLs publizierter RDF-Daten erfasst werden. Er ermöglicht die Verwendung der Daten durch Dritte.
- Ein Prototyp-Informationssystem, das Daten im FoaF-Vokabular verarbeiten und darstellen kann („FOAF-OIS“). Seine Entwicklung überbrückt die Zeit vor der Fertigstellung des Hochschulvokabulars. So können bereits Erfahrungen im Umgang mit den notwendigen Technologien gesammelt werden.

### 3.2 Organisation

Das Projekt hat 21 Teilnehmer<sup>7</sup>. Die Mehrzahl sind Studenten des Fachbereichs; drei kommen von der Technischen Universität Berlin. Gemäß den Vorgaben des Auftraggebers wurden drei Arbeitsgruppen gebildet:

**AG Modellierung** (8 Personen): Aufgaben waren die Erstellung von Modellen für die Beschreibung von Hochschulen, Studiengängen und Informatikthemen, wenn möglich aufbauend auf vorhandenen Ansätzen; die Erfassung der Daten über die Berliner Hochschullandschaft mit Hilfe dieses Modells; und die Dokumentation des Modells und der anderen Arbeiten

---

<sup>6</sup><http://www.ldodds.com/foaf/foaf-a-matic.html>

<sup>7</sup>Teilnehmerliste:<http://projects.mi.fu-berlin.de/semweb/wiki/?page=TeilnehmerListe>

**AG Programmierung** (8 Personen): Hauptaufgabe der Gruppe war die Umsetzung des Online-Informationssystems auf Grundlage der Modelle und Daten der AG Modellierung. Dazu notwendig waren die Auswahl und Einrichtung geeigneter Werkzeuge und Komponenten, die Entwicklung eines geeigneten Navigations- und Interaktionskonzepts, und die Dokumentation der Arbeiten.

**AG Projektmanagement** (5 Personen): Aufgaben dieser Gruppe waren die Organisation und Dokumentation der Projektarbeit und die Außenvertretung des Projekts. Jeweils zwei Mitglieder dieser Gruppe organisierten die Arbeit einer der beiden anderen Gruppen.

Der Termin der Lehrveranstaltung wurde zu einer wöchentlichen zweistündigen Projektsitzung genutzt, an der nach Möglichkeit alle Projektmitglieder teilnahmen. Weitere Treffen einzelner Gruppen und Mitglieder wurden nach Bedarf vereinbart. Zur Kommunikation und Koordination der Arbeiten werden intensiv elektronische Medien benutzt, insbesondere eine Mailingliste, ein Wiki<sup>8</sup> und Instant Messaging. Der Zwischenbericht<sup>9</sup> enthält weitere Informationen über die verwendete elektronische Infrastruktur.

Trotz intensiver Nutzung elektronischer Kommunikationsmedien stand zu wenig Zeit für Besprechungen zur Verfügung. Zwei Stunden pro Woche reichten nicht aus, um die Arbeit von 21 Projektmitgliedern zu koordinieren. Missverständnisse und doppelt gemachte Arbeit waren die Folge.

Die Mehrheit der Projektteilnehmer verfügten über wenig oder keine Vorkenntnisse im Bereich Semantic Web, dafür aber über breit gefächerte Vorkenntnisse in relevanten Bereichen wie XML, Webserver-Technologien und Softwareentwicklung. Leider wurde nicht wie beim Vorgängerprojekt ein paralleles Seminar mit Schwerpunkt Semantic Web angeboten. Aus Zeitgründen konnten auch die Projektsitzung nicht für die Vermittlung von Grundwissen genutzt werden. Die Teilnehmer mussten sich relevante Kenntnisse selbst erarbeiten.

## 4 Umsetzung

### 4.1 Projektstart

Die Lehrveranstaltung begann mit der 0. Sitzung am 21. April 2004. Prof. Tolksdorf stellte die Arbeitsziele des Projekts vor, es wurden die drei Arbeitsgruppen gebildet und verschiedene organisatorische Dinge besprochen.

Ab der 1. Sitzung am 28. April leitete die Arbeitsgruppe Projektmanagement die Sitzungen. In der ersten Sitzung bekamen alle Teilnehmer die

---

<sup>8</sup><http://projects.mi.fu-berlin.de/semweb/wiki/>

<sup>9</sup>[http://projects.mi.fu-berlin.de/semweb/docs/Meilensteinbericht\\_2004-06-09.html](http://projects.mi.fu-berlin.de/semweb/docs/Meilensteinbericht_2004-06-09.html)

Aufgabe, ein Profil ihrer selbst mit dem FoaF-Vokabular zu erstellen und zu publizieren. Dadurch kamen die Teilnehmer mit ersten Konzepten und Technologien des Semantic Web in Kontakt, wurden ihre Kontaktdaten gesammelt und entstand ein Grundstock an RDF-Daten für das FoaF-OIS.

In der 2. Sitzung am 5. Mai stellte das Projektmanagement den zuvor erarbeiteten Projekt- und Zeitplan vor. Arbeitspakete wurden aufgeteilt und es begann die Arbeit in den AGs Modellierung und Programmierung.

## 4.2 Entwicklung des Vokabulars durch die AG Modellierung

Für die Entwicklung des Vokabulars und die Erfassung von Beispieldaten wurde auf Ontologie-Editoren wie Protégé oder OilEd verzichtet. Statt dessen wurde das Vokabular als Textdatei in der Syntax RDF/N3 aufgeschrieben. Dadurch wurden verschiedene Interoperabilitätsprobleme vermieden, die sonst im Umgang mit Semantic-Web-Tools an der Tagesordnung sind. Die Projektmitglieder konnten sich auf RDF konzentrieren und mussten sich nicht mit den Eigenheiten verschiedener Editoren auseinandersetzen. Die gängige Syntax RDF/XML wurde nicht benutzt, da sie für Menschen schwerer les- und schreibbar ist. Die Verwendung von CVS funktioniert mit RDF/N3 problemloser als mit XML-Formaten.

Als problematisch haben sich Umlaute in den RDF/N3-Dateien herausgestellt, da sie UTF-8-codiert sein müssen. Das wird von vielen Texteditoren und CVS unzureichend unterstützt.

Dank der CVS-Webschnittstelle und verschiedener Online-Dienste steht jederzeit eine nach RDF/XML konvertierte Version des aktuellen Vokabulars bereit. Dabei gehen allerdings Kommentare im N3 verloren.

Um eine gemeinsame Arbeit am Vokabular mit acht Personen zu ermöglichen, wurde dieses in der ersten Sitzung am 28. April in sechs Konzepte unterteilt. Das Konzept *Hochschule* umfasst die verschiedenen Hochschularten, sowie deren Eckdaten. Die *Organisationsstruktur* von Hochschulen modelliert die einzelnen Einrichtungen an Hochschulen wie Fachbereiche und Fakultäten sowie deren Beziehung zueinander. Das *Personen*konzept umfasst die für unsere Anwendung wichtigsten Personen wie Professoren und Wissenschaftliche Mitarbeiter etc.. Die verschiedenen Arten von *Lehrveranstaltungen* sind in einem weiteren Konzept modelliert. Die *Studiengänge* mit ihren Zugangsvoraussetzungen, Sachgebieten und Schwerpunkten bilden ein weiteres Konzept. Das Konzept *Abschlüsse* umfasst alle Arten von Abschlüssen und deren Zugehörigkeit zu den verschiedenen Studiengängen.

Die Konzepte *Organisationsstruktur* sowie *Studiengänge* wurden wegen ihres Umfangs von zwei Personen modelliert, die anderen jeweils von einer.

Zu Beginn wurden alle Konzepte umgangssprachlich entworfen und eine grobe Klassenhierarchie erstellt, sowie Verbindungen zu anderen Konzepten notiert. Dadurch gewannen wir Zeit, um uns mit RDF/N3 zu beschäftigen

und konnten besser über die entstehenden Konzepte diskutieren.

Hierbei wurde unter anderem beschlossen, das Konzept *Lehrveranstaltungen* zwar zu modellieren, jedoch aufgrund des hohen Aufwandes und der im Vergleich dazu sehr niedrigen Wiederverwendbarkeit von einer umfangreichen Instanz-Erfassung abzusehen.

Ab der 3. Sitzung begann die Modellierung mit RDF/N3.

Zur Validierung der RDF/N3 Dateien wurde CWM<sup>10</sup> verwendet.

Nachdem der Grundstock des Vokabulars feststand, wurde mit der Erstellung konkreter Beispiele zu jedem Konzept begonnen. So konnten wir Modellierungsmängel finden und den Programmierern die Verwendung des Vokabulars erklären. Im weiteren Verlauf des Projektes wurden Exemplardaten durch unser Partnerseminar an der TU Berlin erfasst.

In der ersten Projektwoche haben wir einige schon bestehende Vokabulare und Ontologien auf ihre Verwendbarkeit in unserem Kontext untersucht:

**Dublin Core**<sup>11</sup> und **FOAF**<sup>12</sup> werden verwendet, da sie weit verbreitete Standards sind.

**EduOnto**<sup>13</sup> ist eine Ontologie zur Beschreibung von Ressourcen, mit denen man Schulumrichtungen modellieren kann. Wir verwenden EduOnto auch zur Modellierung von Einrichtungsadressen.

**Die Dokumente zu ISO/IEC SC36**<sup>14</sup> sind nicht frei verfügbar und wurden deshalb nicht genauer evaluiert.

**Die ACM-Klassifikation**<sup>15</sup> verwenden wir zur Beschreibung der Sachgebiete von Studiengängen und Lehrveranstaltungen im Bereich der Informatik.

**WordNet**<sup>16</sup>, **GermaNet**<sup>17</sup> und **EuroWordNet**<sup>18</sup> könnten noch verwendet werden, um zusätzliche semantische Bezüge herzustellen, weisen aber angesichts des momentanen Standes noch keine Vorteile auf.

Es wurden zwei alternative Möglichkeiten diskutiert, um diese existierenden Vokabulare einzubinden:

1. Direkte Verwendung dieser Vokabulare zum Ausdrücken der Hochschuldaten
2. Definition eigener Klassen und Properties für diese Konzepte, und Herstellung von Äquivalenzen mit Hilfe von RDFS und OWL (rdfs:subClassOf, rdfs:subPropertyOf, owl:sameAs)

Es wurde beschlossen, Äquivalenzen zu anderen Vokabularen zu verwenden, da dies ein sprachlich einheitliches Vokabular ergibt.

---

<sup>10</sup><http://www.w3.org/2000/10/swap/doc/cwm.html>

### **4.3 FOAF-Informationssystem**

Um sich mit den Werkzeugen, die für die Umsetzung der Programmieraufgaben benötigt werden, vertraut zu machen, und da zu Anfang noch nicht auf ein fertiggestelltes Hochschulmodell der Modellierungs-Gruppe zugegriffen werden konnte, wurde beschlossen, zunächst ein Servlet auf Basis des FOAF (Friend Of A Friend)-Vokabulars zu schreiben. Dieses erste Servlet diente der Auswertung von in RDF-Dateien repräsentierten FOAF-Profilen. Die zuvor erstellten Profile aller Projektmitglieder dienten als Testdaten für dieses System.

Ferner wurden ein User-Frontend für den Zugriff auf das Onlineinformationssystem über das WWW erstellt. Allerdings wurde dieser Dienst bisher nur lokal zu Testzwecken gestartet. Das FOAF-Onlineinformationssystem wurde als Servlet für den Apache-Tomcat 5 implementiert.

### **4.4 Organisation innerhalb der AG Programmierung**

Da wir in diesem Semester das Glück hatten, eine verhältnismäßig große Teilnehmergruppe im gesamten Projekt „Semantic Web“ und somit auch in der Programmiergruppe zu haben, ließen sich die Teilaufgaben der Implementierung unseres Online-Informationssystems (im folgenden abgekürzt als OIS) gut aufteilen. Wir hatten später auch die Möglichkeit, Teilnehmer aus der Modellierungsgruppe als Unterstützung für die Programmiergruppe zu gewinnen, die ferner als Vermittler zwischen beiden Gruppen fungierten, um eine entsprechende Kommunikation hinsichtlich der Arbeit mit dem Modell und seinen Nutzdaten zu gewährleisten.

### **4.5 Die Teilaufgaben der AG Programmierung thematisch**

1. Implementierung der Datenbankanbindung
2. Absprache des Apache-Tomcat-Servers mit dem Rechnerbetrieb des Fachbereichs Mathematik und Informatik abn der Freien Universität Berlin, es wurde ein Tomcat 4.0.4 gewählt
3. Implementierung des sog. „Hochschul-A-Matic“, ein Browser-Formular auf JSP-Basis zur Erstellung von RDF-Teilmodellen, das der Modellierungsgruppe zur Arbeitserleichterung zur Verfügung stehen sollte
4. Implementierung eines URL-Verzeichnisdienstes; in diesem werden die URLs der Modell-Dateien (RDF/XML) festgehalten, eine Grundidee des Systems, die Daten nicht zentral, sondern im Sinne eines Verteilten Systems an beliebigen Standpunkten ablegen zu können. Ferner muss der Verzeichnisdienst in regelmäßigen Zeitabständen überprüfen, ob die Modell-Dateien modifiziert oder entfernt wurden und dieses zu Aktualisierungszwecken an Jena melden.

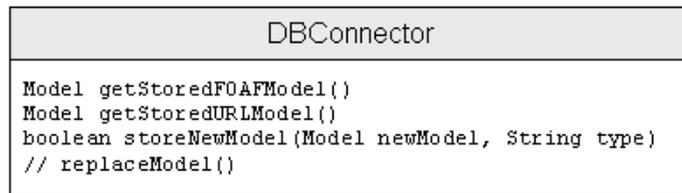
5. Implementierung von „Beans“, kleinen Klassen zur Entgegennahme von direkten Anfragen auf bestimmte Modelldaten mithilfe von speziellen `get()/set()`-Methoden
6. Implementierung einer `FactoryClass` zur Datenspeisung der Beans
7. Implementierung individueller, in der Gruppe mit dem Management ausgearbeiteter „UseCases“, direkter semantischer Benutzeranfragen an das System. Diese werden in der implementierten Form als „Queries“ bezeichnet.
8. Aufsetzen eines Designs für die Browser-Benutzerebene auf JSP-Basis

#### 4.6 Architektur des Online-Informationssystems

Es wurde für das Online-Informationssystem eine Architektur gewählt, bei der die von Jena erfassten Modelldaten über entsprechende Beans zur internen Kommunikation und mithilfe von Servlets bzw. JSP an ein Benutzer-View-Frontend weitergeleitet und dort ausgegeben werden. Die eigentliche Suche erfolgt durch die in den Beans implementierten und auf das jeweilige Suchkriterium ausgelegten (`get()/set()`-) Methoden. Die Interaktion zwischen den Beans und Jena erfolgt teilweise mithilfe von RDQL. Ferner wird eine Datenbank (auf den privaten Rechnern der Programmier-Gruppe SQL, später auf dem Universitäts-Rechner Oracle) zur Speicherung der Modelldaten eingesetzt. Für jeden UseCase soll es eine Anfrageklasse in der Package `queries` geben. Innerhalb dieser Klassen gibt es die bereits erwähnten `set()`-Methoden, mit denen die speziellen Parameter der Anfrage festgelegt werden. Sie bekommen das `model` übergeben und führen in der Methode `doQuery()` die passende RDQL Anfragen aus. Die gefundenen URIs werden an eine Factoryklasse (Package `factories`) übergeben, die dann entsprechende Beans (Package `beans`) erzeugt. Der Aufrufer der Anfrageklasse bekommt dann eine Liste mit den Beans geliefert. Die Anfrageklassen (`queries`) verwalten also die speziellen UseCases. Es wird je Anfrage eine Klasse genutzt, um so die Fehleranfälligkeit einer Universalanfrage zu vermeiden und die spätere Erweiterbarkeit zu gewährleisten. Die Zuteilung des `models` soll von „ausen“ (also vom Servlet) vorgenommen werden, um eine Unabhängigkeit von der Art der Datenhaltung (Datenbank oder im Speicher) zu bekommen. Die `Factories` werden erstellt, um das Befüllen der Beans zu zentralisieren. Da dieser Vorgang nur vom Ergebnistyp, und nicht von der Anfrage abhängig ist, wurde diese Arbeit auf eigene Klassen aufgeteilt. Die Beans enthalten jeweils eine Entität.

Zusätzlich wurde eine von außen zugängliche RDQL-Schnittstelle geplant, so dass Anfragen von Maschinen (Agenten) am Frontend vorbei an die Jena-Instanzen gestellt werden können.

Für die Kommunikation mit der Datenbank gibt es die Klasse `DBConnector.java` (Package `ois`).



Mit dieser Klasse wird die Kommunikation zur Datenbank gekapselt. Hier werden auch Verknüpfungen von äquivalenten Einträgen vorgenommen.

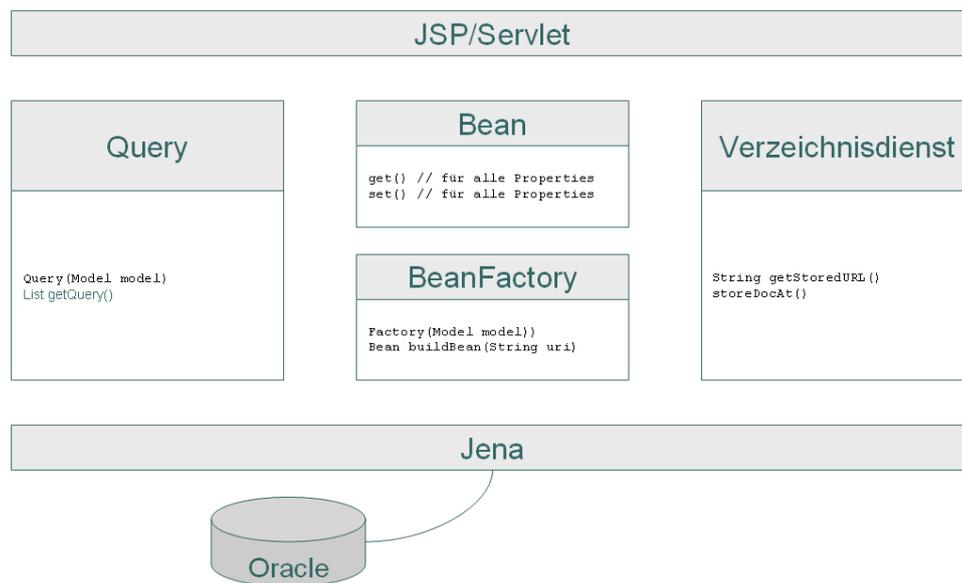
#### 4.7 UseCases

Die Gruppe entschied sich in Absprache mit der Management-Gruppe für die Umsetzung folgender UseCases, hier als Beispielanfragen:

1. „Ich will etwas über Hardware lernen.“
2. „Ich will in Berlin studieren.“
3. „Ich möchte eine eher [technische, angewandte, theoretische, praktische, medien, bio, wirtschaftliche ...] orientierte Informatikausbildung.“
4. „An welchem Institut werde ich am ehesten mit meiner Abinote angenommen?“
5. „Ich will Informatik mit dem Nebenfach xyz studieren.“
6. „Ich möchte nächsten April beginnen.“
7. „Wo wurden in den letzten Y Semestern die meisten Veranstaltungen zum Thema X angeboten?“
8. „Ich möchte auf der Basis meines Abschluss in xyz-Informatik FH möglichst direkt einen Diplom in Informatik an einer Universität erwerben. Wo werden mir viele meiner bisherigen Leistungen direkt anerkannt?“
9. „Ich möchte auf der Basis meines Abschluss in xyz-Informatik FH möglichst direkt einen Diplom in Informatik an einer Universität erwerben. Welche dort notwendigen Leistungen fehlen mir?“
10. „An welchem Institut laufen Projekte in meinem Interessensgebiet, an denen ich mich beteiligen kann?“
11. „Wo wird im Fachgebiet x geforscht?“

Pro UseCases sollte nun eine spezielle `Query`-Klasse implementiert werden. Zunächst wurde darüber diskutiert, universelle Anfrage-Möglichkeiten zu implementieren, um so eine möglichst große Anzahl an möglichen Anfragen abdecken und gegebenenfalls Anfragen (UseCases) kombinieren zu können, was jedoch aus Komplexitätsgründen nicht beibehalten wurde, da der Zeitaufwand wahrscheinlich den Zeitrahmen des Projekts überlastet hätte. Zu den entsprechenden Queries sollten dann später auf der Design-Ebene passende JSPs für das Benutzer-Frontend implementiert werden.

#### 4.8 Graphische Darstellung des Online-Information-Systems



Der Ablauf einer Anfrage anhand der Grafik ist wie folgt:

1. Die im Browser gewählte Anfrage wird im Servlet instanziiert und das entsprechende `Query` wird aufgerufen.
2. Das `Query` führt die Suche via RDQL und den zusätzlichen Jena-Methoden auf dem Jena-Modell durch und übergibt die Suchergebnisse in Form von URIs der `BeanFactory`
3. Die `BeanFactory` füllt die Beans mit den Daten aus dem Jena-Modell und liefert diese zurück an das Servlet.
4. Das Servlet formatiert die Suchergebnisse nach Bedarf und stellt diese auf dem anfragenden Browser dar.

## 4.9 Umsetzung des Online-Informationssystems

Die Umsetzung des OIS verlief generell planmäßig. Der Verzeichnisdienst und die Datenbankanbindung waren bereits sehr früh fertig, der Code konnte teilweise aus der Implementierung der FOAF-Übungen übernommen werden; auch die Bean- und die BeanFactory- Klassen stellten keinen großen Aufwand dar.

Der Apache-Tomcat-Server stand nach etwas mühseliger Absprache mit der Technik funktionstüchtig zur Verfügung, hier war das Hauptproblem die Einigung auf eine bestimmte Tomcat-Version. Das Hochschul-A-Matic-Teilsystem stand nach ein paar Verzögerungen ebenfalls funktionstüchtig und zufriedenstellend bereit.

An manchen Punkten bestand erhöhter Diskussionsbedarf, wie sich immer wieder bei den wöchentlichen Besprechungen herausstellte. Die Implementierung der Queries für die UseCases musste letztendlich „UseCases für UseCase“ erfolgen, nicht, wie ursprünglich favorisiert, universell durch eine übergeordnete Query-Klasse, aus der spezielle Anfragen hätten abgeleitet werden können; somit wurde leider auch keine Überschneidung der Kriterien verschiedener UseCases auf der Anfrageebene möglich.

Etwas schleppend verlief leider die Aufsetzung des Designs für das User-Frontend. Dies ist rückblickend betrachtet etwas schade, da man bereits recht frühzeitig erkannte, dass dies ein Aspekt ist, der nicht hätte vernachlässigt werden sollen, da er einen zentralen Punkt bei der Abschlusspräsentation darstellen würde. Leider blieben bis zuletzt ein paar Unschönheiten, wie unangebrachte Fehlerausgaben des Servers oder unpassende Überschriften auf der Benutzeroberfläche sichtbar, was aber nicht implizieren soll, das gesamte Design sei misslungen gewesen, vielmehr war es durchaus übersichtlich und optisch ansprechend, wenn auch leider stellenweise noch etwas unausgereift.

## 5 Ergebnisse

Das wichtigste Ergebnis der Arbeitsgruppe Modellierung ist ein umfassendes Vokabular zur Beschreibung der Informatikstudiengänge in RDF/N3<sup>19</sup>.

Es ist mit RDFS und teilweise OWL beschrieben. Eine automatische Konvertierung zu RDF/XML ist unter folgender URL verfügbar: <http://projects.mi.fu-berlin.de/semweb/ns/hochschulen#>. Dies ist gleichzeitig der Namespace für alle Klassen und Properties des Vokabulars.

Weitere Ergebnisse sind ein dazugehöriges Tutorial<sup>20</sup>, eine HTML-Referenz<sup>21</sup>

---

<sup>19</sup><http://projects.mi.fu-berlin.de/semweb/webcvs/co.php/psw/vocab/vokabular.n3?r=HEAD&p=1>

<sup>20</sup><http://projects.mi.fu-berlin.de/semwebis/tutorial>

<sup>21</sup>[http://projects.mi.fu-berlin.de/semwebis/reference/vocabular\\_reference.html](http://projects.mi.fu-berlin.de/semwebis/reference/vocabular_reference.html)

aller Klassen und Properties und Daten der Freien Universität Berlin, der Humboldt Universität zu Berlin, der Universität Potsdam, der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin. Außerdem hat die Seminargruppe der Technischen Universität Berlin Daten zu ihrer Universität erstellt.

Unsere Software ist funktionsfähig und konnte in der Abschlusspräsentation angemessen demonstriert werden, ist aber nicht bereit für den Produktivbetrieb. Dazu wären folgende Nacharbeiten notwendig:

- gründlicher Test aller Funktionen
- Überarbeitung der Benutzerschnittstelle mit Hinblick auf Usability (zu diesem Punkt enthält der Bericht der TU-Gutachtergruppe wertvolle Hinweise)
- Möglichkeit zur Aktualisierung eingelesener RDF-Dateien im Online-Informationssystem (zur Zeit nur durch Löschung der Datenbank und Neueinlesung aller Profile möglich).

Damit haben wir das Projektziel eines Systems, das „qualitativ als Berliner Hochschulkompass Informatik geeignet“ sein soll, nicht ganz erreicht. Dafür hätten wir etwas mehr Zeit gebraucht.

## 5.1 Beschreibung des Vokabulars

Um eine Hochschule und ihre jeweilige Organisationsstruktur beschreiben zu können, wurde die Klasse `Einrichtung` sowie die Property `istUntereinrichtungVon` erstellt.

Neben der Klasse `Hochschule`, welche Unterklasse von `Einrichtung` ist, gibt es auch Klassen für die verschiedenen Arten von Hochschulen: `Universitaet`, `Fachhochschule` und `KunstOderMusikhochschule`.

Auf der Klasse `Hochschule` sind folgende Properties definiert: `hatGruendungsjahr`, `hatAnzahlStudierende` und `hatLogo`.

Die Organisationsstruktur kann durch eine Hierarchie von Untereinrichtungen dargestellt werden. Dazu sind die Unterklassen `Fachbereich`, `Fakultaet`, `Institut`, `LehrUndForschungsgebiet`, `Immatrikulationsamt`, `Pruefungsamt`, `Studienberatung` und `Sekretariat` nützlich.

Auf der Klasse `Einrichtung` wurden folgende Properties definiert: `hatLeiter`, `hatSekretariat`, `hatOeffnungszeit`, `hatTelefonsprechzeit`, `beschaeftigtSichMitSachgebiet`.

Die in einer Einrichtung arbeitenden Personen wurden als Unterklasse von `foaf:Person` definiert. Eine Person hat einen Vor- und Nachnamen und einen Titel. In unserem Fall wurden `Professor`, `WissenschaftlicherMitarbeiter`, `StudentischerMitarbeiter`

und Tutor als Personen modelliert. Die genannten Personen können mit den Properties `hatPruefungsberechtigung`, `hatAnstellungAnEinrichtung`, `hatSprechzeit` beschrieben werden. Die Properties `giltFuerLehrveranstaltung`, `giltFuerStudiengang`, `giltFuerSachgebiet` wurden für die Klasse `Pruefungsberechtigung` modelliert.

Hochschulen können Studiengänge anbieten. Studiengänge beschäftigen sich mit Sachgebieten und haben einen oder mehrere Schwerpunkte innerhalb dieser Sschgebiete. Weiterhin haben Studiengänge Eckdaten wie die durchschnittliche Abbruchquote und die Regelstudienzeit. Ein Studiengang kann auf einem anderen aufbauen oder auch nur als Nebenfach angeboten werden.

Die Klasse `StudiemsSemester` fasst Informationen über die Einschreib- und Zulassungsbedingungen für einen Studiengang in einem bestimmten Semester zusammen.

Auf jedem Studiengang kann es einen oder mehrere Abschlüsse geben. Diese unterteilen sich in `Bachelor`, `Diplom`, `Magister`, `Master`, `Staatsexamen`. Um (Informatik-)Studiengänge vergleichbar zu machen, können die Mindestanforderungen in Semesterwochenstunden für die jeweiligen (Informatik-)Teilgebiete festgehalten werden. So sind auf Abschluss folgende Properties definiert: `hatMindestSWSGesamt`, `hatMindestSWSInAngewandterInformatik`, `hatMindestSWSInMathematik`, `hatMindestSWSInNebenfach`, `hatMindestSWSInPraktischerInformatik`, `hatMindestSWSInTechnischerInformatik`, `hatMindestSWSInTheoretischerInformatik`.

Auf Einrichtungen angebotene Lehrveranstaltungen werden wie Studiengänge über Sachgebiete vergleichbar gemacht. Zudem wird über die Property `gehörtZuStudiengang` bestimmt, in welchem Studiengang jede Lehrveranstaltung anrechenbar ist.

Mit Hilfe des bestehenden Vokabulars lassen sie Hochschulen mit ihren Untereinrichtungen sowie deren Beziehungen zueinander modellieren. Hierbei ist es dem Benutzer des Vokabulars überlassen, ob er Einrichtungen weiter spezifizieren möchte, z.B. als Fachbereich. Jede Einrichtungen kann einen Leiter, ein Sekretariat sowie Öffnungs- und Telefonische Sprechzeiten haben. Die an Hochschulen tätigen Personen wurden in die Klassen `Professor`, `WissenschaftlicherMitarbeiter`, `StudentischerMitarbeiter` und `Tutor` geteilt. Jede dieser Klassen ist eine Unterklasse von `foaf:Person`. Personen können eine Prüfungsberechtigung haben. Studiengänge an Hochschulen kennzeichnen sich durch Sachgebiete und zugehörige Abschlüsse aus. Zudem können auch einzelne Semester erfasst werden, um Zugangsvoraussetzungen für Studiengänge zu einem bestimmten Zeitpunkt festzuhalten. Lehrveranstaltungen beschäftigen sich wie Studiengänge mit bestimmten Sachgebieten und werden von Einrichtungen angeboten. Weiterhin haben sie Dozenten, welche nicht zwingend Teil der den genannten Personenklassen sein müssen,

sondern auch nur foaf:Person.

Grenzen des Vokabulars sind unter anderem die Erfassung von persönlichen Daten und einzelnen Studiumsabschlüssen. Wir haben uns bewusst entschieden, nicht alles bis hin zum einzelnen Studenten zu modellieren. Da jedoch z.B. Einschreibefristen oder NCs für unsere Anwendung interessant sind, können Instanzen für konkrete Semester angelegt werden.

Vor der Benutzung für über die reine Informatik hinausreichende Studiengänge oder Lehrveranstaltungen bedarf das Konzept der Sachgebiete eine weitere Überarbeitung. Bisher ist nur die ACM-Klassifikation als ein Sachgebiet angegeben.

## 5.2 Fazit: Was nutzen die Semantic Web-Technologien?

Keine Frage wurde im Laufe des Projekts so oft diskutiert wie diese: Welchen Mehrwert bringt die Verwendung von Semantic Web-Technologien für das Projekt? Letztendlich hängt von dieser Frage die endgültige Beurteilung des Projekts ab. Haben wir die Möglichkeiten dieser Technologien in einer Art genutzt, die einen konkreten Vorteil gegenüber üblichen Technologien hat? Bringt das Semantic Web unmittelbare Erleichterungen für „echte“ Anwendungen? Die Gutachter der TU Berlin bringen es in ihrem Zwischenbericht mit wenig Nachsicht auf den Punkt:

„Wenn diese Frage durch das Projekt nicht geklärt wird, man also nicht erkennen kann, warum die Daten ausgerechnet in RDF gegeben sein sollten und nicht in einem proprietären Format, das eventuell Suchen und Abfragen viel schneller macht, dann muss das Projekt trotz des ganzen Aufwands als gescheitert angesehen werden.“

### Offenes System zur Erfassung, zum Austausch und zur Nutzung von Daten:

Ein Schwerpunkt unserer Arbeit war die Entwicklung eines RDFS/OWL-Vokabulars zur Beschreibung von Hochschulen und Studiengängen. Wir benutzen es als gemeinsames Modell und gemeinsame Sprache aller Bestandteile unseres Softwaresystems. Das Vokabular kann aber über unsere eigene Anwendung hinaus nützlich sein. Unsere eigenen Ein- und Ausgabemechanismen (die Hochschul-A-Matics und das Online-Informationssystem) sind in diesem Sinne nur Beispiele für die Verwendungsmöglichkeiten. Der echte Nutzen entsteht, wenn Dritte das Vokabular nutzen, um den Inhalt ihrer Datenbanken ins Semantic Web zu bringen, oder um eigene Informations- oder Suchdienste auf Grundlage der veröffentlichten Daten zu implementieren.

Dieses Szenario ist ohne RDF schwer vorstellbar. RDF senkt die Kosten für die Nutzung fremdpublizierter Daten, da es die Verwendung vieler bestehender Editoren, Viewer, Konverter, Parser und Frameworks ermöglicht (was auch für uns in der Entwicklungsphase sehr hilfreich war).

Semantic Web-Technologien machen also unser Vokabular, unsere Software und von uns erfasste Daten nutzbar für Dritte. Doch auch wir profitieren von Daten und Diensten, die schon im Semantic Web verfügbar sind. Beispielsweise verwenden wir eine Repräsentation der ACM-Klassifikation<sup>22</sup>, die von Tim Finin (University of Maryland Baltimore County) nach DAML konvertiert wurde. Der FoaF Explorer<sup>23</sup> „versteht“ den Teil unseres Vokabulars, den wir in Relation mit FOAF gesetzt haben.

Magnus Niemann (TU Berlin) hat in der Abschlusspräsentation bemerkt, dass man das Vokabular als Hauptergebnis eines solchen Projekts sehen könnte, und die Entwicklung darauf aufbauender Anwendungen als Aufgabe eines zweiten Projekts.

#### **Semantisches Grundgerüst für weitere Semantic-Web-Anwendungen:**

Unser Vokabular enthält Klassen und Properties zur Beschreibung vieler zentraler Entitäten einer Hochschule und ihrer Beziehungen zueinander: Personen, Einrichtungen, Untereinrichtungen, Lehrveranstaltungen, Studiengänge. Dieses Vokabular bietet ein semantisches Grundgerüst, an das sich weitere Semantic-Web-Anwendungen anhängen können. Das wird am Beispiel des FOAF-Vokabulars deutlich. Es dient zur einfachen Beschreibung von Personen. Diverse andere Vokabulare bauen auf FOAF auf, um spezielle Informationen (Lebenslauf, geographische Angaben, detaillierte Beziehungen zu anderen Personen, Web of Trust) im Semantic Web unterzubringen. Diese Vokabulare sind meist hochspezialisiert und müssen sich nicht mehr mit der Definition grundlegender Konzepte wie Personen und Organisationen herumschlagen. Ein kompaktes und gut durchdachtes Hochschul-Vokabular kann eine ähnliche Rolle spielen.

#### **Einheitliches Modell für alle Teile und Schichten einer Anwendung:**

Wir haben alle Teile der Anwendung um das gemeinsame RDF-Vokabular herum aufgebaut. Dadurch verwenden wir überall einheitliche Begriffe und Konzepte, von der Persistenz bis zur Benutzungsschnittstelle. Das ist eine gute Eigenschaft für jede Art von Software. Das Verständnis des zu Grunde liegenden Modells wird einfacher, und die Kommunikation unter den Beteiligten wird verbessert. Die Fachkonzepte der Anwendung wurden von der Implementierung entkoppelt. Die Verwendung von RDF fördert diese Entkopplung. Das ist ein Vorteil von RDF-basierten Anwendungen.

**Inferenz und Reasoning:** In RDF verfasste Daten sind mit reicher Semantik und einer wohldefinierten Modelltheorie unterlegt. Das eröffnet viele interessante Möglichkeiten im Umgang mit diesen Daten.

---

<sup>22</sup><http://www.daml.org/ontologies/67>

<sup>23</sup><http://xml.mfd-consult.dk/foaf/explorer/>

Sie können gegen ein RDFS- oder OWL-Vokabular validiert werden. In verschiedenen Vokabularen erfasste Daten können mit Hilfe einer OWL-Ontologie integriert werden. OWL und höhere Schichten der „Semantic-Web-Torte“ wie RuleML ermöglichen das Ziehen neuer Schlüsse aus bekannten Daten und erlauben komplexere Anfragen. Für jeden dieser Prozesse gibt es vorgefertigte Werkzeuge. Sie sind generisch und müssen nicht für jede Fachdomäne neu entwickelt werden.

Diese Stärken des Semantic-Web-Stapels haben wir kaum genutzt. Das hat verschiedene Gründe. Viel davon steckt noch im Stadium akademischer Forschung. Implementierungen sind nicht ausgereift, schwer zu integrieren und wenig praxistauglich. Die große Anzahl aufeinander aufbauender Technologien und Konzepte war ein Problem für uns. Zu Beginn wussten wir noch nicht genug über unsere Aufgabe und diese Technologien. Wir wollten später auf Gelegenheiten achten, bei denen wir sie sinnvoll einsetzen können. Doch schon die Beherrschung der Grundlagen (RDF und Modellierung) lastete uns weitgehend aus. Sicher haben wir so einige Chancen versäumt, die den Nutzen der Semantic-Web-Technologien deutlicher gemacht hätten.

Wir haben demonstriert, dass ein Online-Informationssystem auf Grundlage von RDF realisierbar ist. Es wäre in seiner abschließenden Form auch mit gängigen Technologien umsetzbar gewesen. Doch im Gegensatz zu üblichen webgestützten Informationssystemen ist es keine „Insel“, sondern als Bestandteil eines offenen semantischen Netzwerks zur Beschreibung von Hochschulen konzipiert. Das ist der potentielle Mehrwert, der natürlich erst dann real wird, wenn Dritte oder Folgeprojekte auf unserer Arbeit aufbauen.

RDF ist eine „enabling technology“. Die Verwendung von RDF allein bringt kaum unmittelbare Vorteile. Interessant ist vielmehr, was für neue Möglichkeiten sich eröffnen, wenn man konsequent auf RDF setzt.