Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit - Master Informatik

Artem Khvat

Ontologien und Repräsentation des Wissens

Fakultät Technik und Informatik
Studiendepartment Informatik
Betreuer: Prof. Dr. von Luck
Datum: 31. Dezember 2005

Artem Khvat

Thema der Seminararbeit - Master Informatik

Ontologien und Repräsentation des Wissens

Stichworte

Semantic Web, Ontologien, OntoWeb, CS AKTive Space, MOnTo, XML, URL, OWL, Jena, Protege

Kurzzusammenfassung

In der vorliegenden Ausarbeitung geht es um die Repräsentation von Wissen anhand von Ontologien im Gebiet des Semantic Web. Sie basiert auf dem Vortrag 'Ontologien und Repräsentation des Wissens', welcher im Rahmen eines Seminars an der Hochschule für Angewandte Wissenschaften Hamburg stattgefunden hat. Die Betreuung dieses Seminars wurde von Prof. Dr. Kai von Luck und Prof. Dr. Bernd Schwarz durchgeführt. Diese Ausarbeitung teilt sich in vier Abschnitte auf. Zu Beginn wird ein Problem des heutigen Webs erläutert und dieses anhand von Beispielen deutlich gemacht. Am Ende des ersten Abschnittes wird eine Technologie vorgeschlagen, die als mögliche Lösung für diese Probleme dienen kann und auch schon heute eingesetzt wird. Im zweiten Abschnitt werden mehrere erfolgreich durchgeführte Projekte im Bereich Semantic Web präsentiert. Das Hauptmerkmal dieser Projekte ist, dass man sie als kleine Musterlösungen für die Probleme des heutigen Webs betrachten kann. In Abschnitt Drei werden Werkzeuge präsentiert, die im Bereich der Wissensrepräsentation schon relativ weit verbreitet sind. Der vierte Abschnitt gibt einen Ausblick auf eine bevorstehende Master Thesis, welche sich im Wesentlichen mit Ontologien und dem automatisierten Datenaustausch in heterogenen und semantisch annotierten Umgebungen befassen wird. Dabei wird eine Risikoabschätzung vorgenommen und Maßnahmen zur Abschwächung der Risiken diskutiert.

Inhaltsverzeichnis

1 Das Problem

In den letzten Jahren ist das World Wide Web zu einem gigantischen Datenspeicher angewachsen. Der Trend der letzten Jahre zeigt, dass es sich in Bezug auf die Datenmengen ständig vergrößern wird. Parallel dazu, kommen immer größere Bandbreiten im Büro und Haushalt durch die ständige Weiterentwicklung der Netzwerktechnologien, und die Kapazitäten der Desktop-Rechner bleiben auch nicht stehen und vergrößern sich in rasantem Tempo. Diese Entwicklungsprozesse stellen neue Qualitätsanforderungen an die Interaktionen mit dem World Wide Web. Immer mehr Herausforderungen des täglichen Lebens finden jetzt ihre Lösung im Internet. Aber nicht immer wird die eigentliche Lösung für den Benutzer dadurch einfacher und übersichtlicher. Als Beispiel dafür, betrachten wird folgendes Szenario über die Interaktion zwischen Mensch und Internet. Dieses Szenario ist aus verschiedenen Sichten ein Klassiker geworden, da es sich jeden Tag Millionen Mal ereignet. Es geht um die Suche nach Informationen im World Wide Web. Es gibt heute eine Reihe von Suchmaschinen im Internet, die ihre Dienste dem Benutzer frei zur Verfügung stellen. Die am stärksten bekannten davon sind Google (`http://www.google.de`), Yahoo (`http://www.yahoo.com`) und Altavista (`http://www.altavista.com`). Für unser Beispiel wurde die Suchmaschine Google benutzt.

Nehmen wir an, dass wir auf der Suche nach Informationen über ein Auto sind. In unserem Falle handelt es sich um Fahrzeuge des Baujahres 1967. Nachdem der Suchauftrag erledigt wurde, besteht das Ergebnis aus mehreren hundert Seiten, die in vielen Fällen ein unbrauchbares Ergebnis darstellen oder nur sehr wage mit dem ursprünglichen Anliegen in Verbindung stehen. Das Einzige gemeinsame Merkmal aller Seiten sind das Vorhandensein der Wörter 'Baujahr','Fahrzeug' und '1967'. Dabei spielt z.B. die letzte Zahl nicht immer die Rolle eines Baujahres. Am Ende ist es dem Benutzer überlassen aus den vielen Hundert Seiten die für ihn brauchbaren herauszufiltern.

Wie man sieht, beschränkt sich das Suchen der heutigen Suchmaschinen im Internet nur auf das Auffinden von Seiten durch den Vergleich von Zeichenketten. Die Bedeutung des eigentlichen Inhalts bleibt der Maschine verschlossen und kann nicht als Suchkriterium verwendet werden. Die gefundenen Links auszuwerten, ist die oft nicht immer kleine Aufgabe des Menschen. Eine erhebliche Verbesserung der Suche kann also dadurch erreicht werden, wenn man den Suchmaschinen die semantische Bedeutung der maschinenlesbaren Informationen im World Wide Web verständlich machen könnte.

Um den Maschinen das selbständige Herausfiltern und Verwerten von Informationen zu ermöglichen, wurde eine hierarchische Struktur von Sematic Web Sprachen und verschiedenen Mechanismen entwickelt. Manche davon befinden sich noch in den Prototypstadien, andere dagegen haben schon heute eine weite Verbreitung und Nutzung. Das Semantic Web baut auf dem schon bestehenden Web von heute auf. Dadurch ist die Kompatibilität mit den bestehenden Technologien im Web eine Anforderung. Der geistige Vater und Erfinder des Sematic Webs -

Berners-Lee - beschreibt dessen Architektur mit Hilfe des so genannten 'Layer-cake'. Unicode ist ein weltweit universeller Zeichensatz. Die ISO Norm 10646

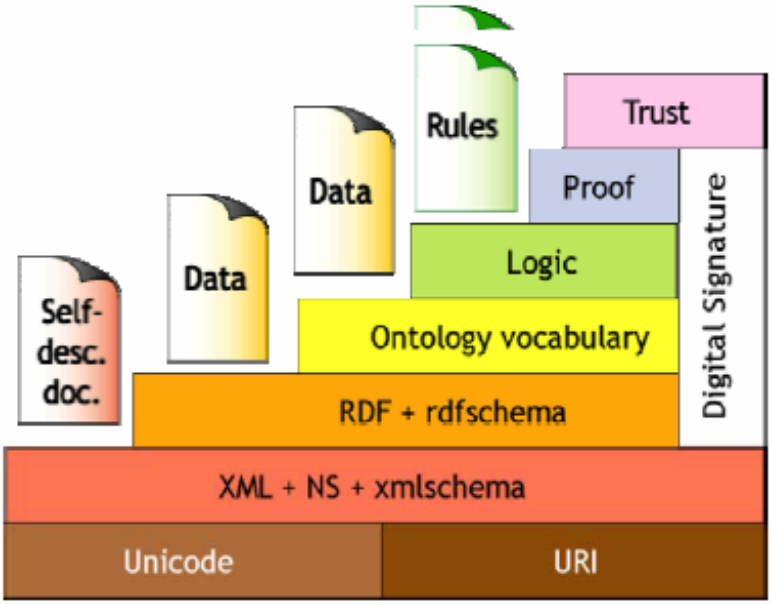

Abbildung 1: Layer-cake des Semantic Web

garantiert die Verwendung des international einheitlichen Standards für Bezeichnungen im World Wide Web.

Der 'Uniform Resource Identifier' (URI) hat seine Einführung bereits bei der Entwicklung des World Wide Web gefunden. Er wird zur eindeutigen Bezeichnung von Ressourcen im Netz benutzt und wurde von der W3C eingeführt. Damit man über ein und dasselbe Objekt in der Welt mit anderen Kommunikationspartnern sprechen kann, braucht man eine eindeutige Identifizierung dieses Objektes. Genau diese Aufgabe wird durch einen URI erledigt. Im heutigen Internet hat daher jede vorhandene Ressource einen nur für sie bestimmten URI. Dieser URI beinhaltet einen eindeutigen Namen und Pfad. Eine Ableitung des URI ist der so genannte URL ('Uniform Resource Locator'). Der Hauptunterschied zwischen den beiden besteht darin, dass der URL zusätzlich zu der eindeutigen Namenssicherung auch gleichzeitig die Lokalisierung des Objektes sichert, wie z.B. bei der URL der Webseite (`http://www.google.de`). Im Gegensatz dazu muss der URI keinen Pfad des Objektes enthalten.

Während die visuelle Wiedergabe der Daten auf einer Webseite mit Hilfe von HTML (Hyper Text Markup Language) ausgeführt wird, übernimmt XML (Extensible Markup Language) den Austausch von Daten über das Netz. Hiermit werden dem Rechner viel mehr Angaben über die Strukturierung der Daten und teilweise Informationen über den Inhalt zur Verfügung gestellt. Z.B:

<sentence>
<fahrzeug href=http://www.seriouswheels.com/pics-1960-1969/1968-Pontiac-Firebird-Red-Blower-s-sy.jpg>'Fierbird' Baujahr
<baujahr>1968</baujahr>
</fahrzeug>, ist das beste was, dem Man, passieren kann.
</sentence>

Der Informationsgewinn wäre in diesem Fall die Möglichkeit der Maschine folgende Dinge auch ohne das Eingreifen des Anwenders zu bestimmen.

- Es handelt sich um einen Satz
- 'Firebirde' eine Ressource vom Type <fahrzeug> ist
- die eindeutig durch den URL (http://www.seriouswheels.com/pics-1960-1969/1968-Pontiac-Firebird-Red-Blower-s-sy.jpg) referenziert und lokalisiert wird
- die Zahl 1968 ist eine Instanz vom Type <baujahr>, und nicht nur einfach eine Zahlenfolge
- Die Zahl 1968 vom Typ <baujahr> ist eine Teilstruktur der Instanz vom Type <fahrzeug>

Um die möglichen Probleme der Mehrdeutigkeit zu lösen, werden sog. Namespace-Präfixe (Namensräume) benutzt. Im Folgenden Beispiel kommt <auto> doppelt vor:

<bmw:auto>...</bmw:auto> und <mercedes:auto>...</mercedes:auto>

Die Namespaces mit URI sehen wie folgt aus:

<bmw:auto xmlns:bmw='http://www.bmw.com/xmlns/'>...</bmw:auto>

<mercedes:auto xmlns:mercedes='http://www.mercedes.de/'>...</mercedes:auto>

Sehr wichtig ist dabei, dass XML lizenzfrei als Technologie des W3C zur Verfügung steht und Unterstützung von Seiten der Industrie findet.

Ontologien beschreiben nun die Gegenstände formal. Der Schwerpunkt liegt dabei auf der Beschreibung ihrer Beziehungen zueinander. Die Basis von Ontologien bildet die RDF-Syntax und das RDF-Schema. Zusätzlich enthalten sie Konstrukte, die der Sprache Prolog äußerst ähnlich sind. Ein möglicher Einsatz von Ontologien wird durch folgendes Beispiel veranschaulicht. Nehmen wir an, der Benutzer will eine Suche über verschiedene Autoanbieter starten. Dabei hat dieser den Wunsch ein bestimmtes Auto zu finden. Für die Beschreibung der eigenen Bestände benutzen die Autoanbieter XML - Vorlagen. Des Weiteren nehmen wir an, dass unsere Vorstellung von dem gesuchten Auto wie folgt aussieht:

<?xml version='1.0' standalone='yes' encoding='UTF-8'?> <Auto>
<hersteller>Opel </hersteller>
<model>
<name> Vectra</name>

```
    <farbe> Blau</farbe>
    <jahr> 1998</jahr>
  </model>
  <motor>
    <leistung> 75 KW</leistung>
    <hub>1600</hub>
  </motor>
</Auto>
```

Während der Suche stoßen wir auf zwei Anbieter, die das gesuchte Auto in ihren Beständen haben. Allerdings weicht deren Vorstellung von dem uns bekannten Auto ab.

Anbieter 1:

```
<?xml version='1.0' standalone='yes' encoding='UTF-8'?>
<Auto>
  <hersteller>Opel</hersteller>
  <name>Vectra </name>
  <farbe>blau</farbe>
  <jahr>1998</jahr>
  <leistung>75KW</leistung>
  <hub>1600</hub>
</Auto>
```

Wie leicht ersichtlich ist, handelt es sich hier um das gleiche Auto. Nur durch die andere Strukturierung des XML - Dokumentes schlägt die Suche fehl, da durch direkten Vergleich der Tags keine Erkennung der semantisch gleichen Autos möglich ist.

Anbieter 2:

```
<?xml version='1.0' standalone='yes' encoding='UTF-8'?>
<Auto>
  <manufacturer>Opel</manufacturer >
  <model>
    <name>Vectra </name>
    <color>Blau</color>
    <year>1998</year>
  </model>
  <motor>
    <power>75KW</power>
    <cylindercapacity>1600</cylindercapacity>
  </motor>
</Auto>
```

Auch bei diesem Anbieter, handelt es sich wieder um das von uns gesuchte Auto. Das Problem ist hier allerdings die unterschiedlichen Bezeichungen der Tags. Auch dadurch ist ein Informationsgewinn durch einen direkten Vergleich der Tags nicht möglich und die Suche schlägt abermals fehl.

Wie das Beispiel zeigt, kann ohne das explizite Eingreifen des Anwenders die Suche in heterogenen Umgebungen fehlschlagen. Die Suchmaschine ist nicht in der Lage Zusammenhänge zu erkennen, die für uns Menschen offensichtlich erscheinen. Das Problem hierbei ist nicht der Mangel an Rechenkapazität von heutigen Rechnern, sondern vielmehr das Fehlen der Möglichkeit Schlussfolgerungen aus den gegebenen Repräsentationen zu ziehen. Dazu bedarf es parallel zur maschinenlesbaren Beschreibung der Daten durch XML eine maschinenlesbare semantische Beschreibung. Und genau diese Lücke sollen die Ontologien schließen. Die Ontologien sollen den Maschinen ermöglichen genau diese Schlussfolgerungen aus den Informationen zu ziehen, mit denen sie arbeiten. Die ersten brauchbaren Ontologien, die genau für diese Aufgabe erstellt wurden, sind schon im Einsatz.

Momentan kann man nicht von 'der einen' Ontologie sprechen. Es existieren verschiedene Stellvertreter davon. Zum Beispiel kam es durch die Initiative der Europäer zu einem Produkt namens OIL. Ein weiteres Beispiel ist die amerikanische Variante DAML. Später haben die beiden Forschungsgruppen ihre Ergebnisse zusammengefügt. Das Ergebnis dieser Vereinigung war die Ontologiesprache OWL (Web Ontologie Language). Allerdings existieren, wie oben schon erwähnt, neben OWL noch weitere Ontologiesprachen. Hier wurden nur die bedeutendsten genannt.

Die Logik, Proof -und Truststufen des 'Layer-cake' sind noch in einem Entwicklungsstadium. Das endgültige Ziel ist der Aufbau eines 'Web des Vertrauens'.

2 Projekte

In diesem Kapitel werden zwei erfolgreich durchgeführte Projekte, aus dem Bereich des Computer Science, vorgestellt. Ziel ist an Hand der funktionierenden Lösungen zu zeigen, welches Potenzial der Einsatz von Ontologien in Kombination mit anderen Techniken in sich verbirgt und das mögliche Einsatzgebiet dieser Technologie zu präsentieren.

2.1 CS AKTive Space

Als erstes wird das Project CS AKTive Space vorgestellt [9]. Zweck dieses Projektes ist es, einen Überblick über die Universitäten in Großbritannien zu verschaffen, welche im Bereich Informatik wissenschaftliche Forschungen durchführen. Es werden Informationen bezüglich der Forschungsinstitute, wissenschaftlicher Mitarbeiter und Forschungsgruppen erfasst. Auf Grund der gesammelten Informationen bittet die Applikation diverse Dienste und Anwendungsmöglichkeiten. Zum Beispiel :

- Anzeigen von einzelnen oder regional zusammengefassten Forschungsgebieten der Universitäten
- Anzeigen der relevanten Mitarbeiter in einem bestimmten Forschungsgebiet

- Auflisten von zusammenarbeitenden Personen
- Anzeigen von Kontaktinformationen und Verweisen auf wissenschaftliche Arbeiten der Mitarbeiter

Alles zusammen verwaltet das System 2000 Fakultäten, 24000 Projekte, riesige Mengen von wissenschaftlichen Papieren und mehrere hundert Forschungsgruppen. Die Generierung des gesamten Inhaltes, der auf der Webseite präsentiert wird, passiert völlig automatisch. Die Informationen werden aus unterschiedlichsten Quellen (RDF Sources, DB, persönliche Webseiten) des Sematic Web bezogen. Durch so einen Ansatz wird die Aktualität der Datenhaltung garantiert. Außerdem wird die Vollständigkeit der Informationen so weit wie möglich beibehalten. Parallel zu den eigentlichen Zielen der Informationsverwaltung von Forschungsprojekten bei Universitäten, wurden im Laufe des Projektes folgende Fragen der Wissensrepräsentation behandelt:

- Probleme des Umfangs und der Skalierbarkeit der Daten
- Die besten Methoden für die Unterstützung des Erwerbs und des Sammelns von Wissen
- Beseitigung der Informationsredundanz
- Die automatische Überwachung der Möglichkeit neue Dienste, in Abhängigkeit von neu erworbenem Wissensvolumen, anzubieten
- Grundprinzipien der Repräsentation einer Webseite in einer Semantic Web Umgebung

Grundsätzlich sammelt und verarbeitet CS AKTive Space Informationen aus diversen, weit verteilten Quellen. Unterstützung findet die Applikation durch einen Server, der es ermöglicht die RDF - Tripel zu speichern und zu verarbeiten. Der Server trägt den Namen 3Stone. 3Stone ist unter SourceForge frei verfügbar und kann etwa 25 Millionen RDF - Tripel speichern. Das Anzeigen der abgespeicherten Inhalte der Datenbank wird durch den 3Stone Browser ermöglicht. Die Erweiterung der Wissensbasis wird mit Hilfe von dem integrierten Tool Armadillo realisiert. Armadillo interpretiert, auf Basis des vorhandenen Wissens und der Fähigkeit natürlicher Sprachverarbeitung, unterschiedliche Ressourcen im Internet. Interessante Fakten werden extrahiert und automatisch in RDF - Tripel umgewandelt. Das ermöglicht die entsprechenden Webinhalte mit den richtigen Annotationen zu versehen. So kann zum Beispiel aufgrund des Wissens über einen Autor, auf die von ihm verfassten Texte geschlossen werden. Damit man alle Mitarbeiter finden könnte, die mit einer Person eng zusammenarbeiten, benutzt CS AKTive Space das Ontocopi Applet. Ontocopi ermöglicht es Beziehungen innerhalb einer Wissensbasis aufzuspüren, indem es eine Methode der ontologischen Netzwerkanalyse benutzt. Diese Methode erlaubt Beziehungen aufzudecken, die in der Wissensbasis nur implizit vorhanden sind. Es werden die Beziehungen der einzelnen

Instanzen aufgrund des Typs und Dichte untersucht, und anhand der Gewichtungen der Relevanz überprüft. Es wird zum Beispiel der Beziehung 'ist Leiter von' eine höhere Gewichtung gegeben als der Beziehung 'ist Mitarbeiter von'. Durch Akkumulierung der einzelnen Gewichtungen entlang der Suchpfade werden die relevanten Instanzen erkannt.

Zum Schluss muss gesagt werden, dass kein großer Wert darauf gelegt wurde, die technischen Aspekte dieses Projektes perfekt zu realisieren. Es war vielmehr ein Versuch eine mögliche Lösung der Probleme im heutigen World Wide Web zu präsentieren. Es wurde versucht die grundlegenden Probleme zu untersuchen und eventuell die ersten Lösungen dafür zu implementieren. Die ausführliche Beschreibung der Ergebnisse des Projektes und die letzte Version der Implementierung findet man auf der Webseite von CS AKTive Space unter `http://cs.aktivespace.org/`. Um den Erfolg und Wichtigkeit dieses Projektes zu unterstreichen, sei hier erwähnt, dass CS AKTIve Space Semantic den Web Challenge 2003 gewonnen hat.

2.2 OntoWeb

Als Nächstes wird ein Projekt präsentiert, welches ein geschlossenes thematisches Netzwerk darstellt [6]. Das Hauptmerkmal von OntoWeb ist, dass das Wissen über Ontologien und Wissensverwaltung über ein Onlineportal nicht nur angeboten wird, sondern die Plattform selbst Ontologien zum Sammeln und Verwalten des Wissens verwendet. Bei so einem Einsatz von Ontologien, wird ihre Mächtigkeit besonders deutlich. Heute sind schon mehr als 120 Partner aus der Industrie und Forschung aus den Bereichen Wissensmanagement und eCommerce in diesem Projekt involviert. Ähnlich wie bei CS AKTive Space wird durch eine einheitliche semantische Form auf heterogenes Wissens zugegriffen. Das Ziel von OntoWeb ist es, die Kommunikation unter diversen Forschungsgruppen zu fördern und somit Faktoren wie Zeit und Qualität zu verbessern.

OntoWeb baut auf drei freiverfügbaren Produkten auf:

- Einem objektorientierten Content Management System namens Zope
- Dem DOGMA Framework für die Erstellung von Ontologien
- Dem Produkt SEAL für die Erstellung und Verwaltung des Onlineportals

Zope spielt parallel die Rolle des zentralen Web- und Contentservers. Um die Inhalte zu speichern wird die Zope Datenbank (ZODB) verwendet. Dem Benutzer ist es möglich die Daten direkt über das Portal abzuspeichern.

Das DOGMA - Framework spielt eine zentrale Rolle beim Aufbau der Ontologien in OntoWeb. Man kann es als Schnittstelle zwischen den beteiligten Ressourcen und der zugrunde liegenden Ontologien verwenden. Der Hauptgedanke des Ansatzes ist die Ontologien in zwei Schichten, die Ontologie Basis und den Commitment Layer, aufzuteilen. Bei dieser Art der Aufteilung von Ontologien spricht

man auch von der Double Articulation einer Ontologie. Es werden die atomaren formalen Konzepte von dem Kontext, in dem sie ihre Verwendung finden, getrennt. Die Basis enthält dann lediglich die Konzepte, für die im Commitment Layer die Regeln für deren Verwendung in verschiedenen Domänen festgehalten werden. Der wesentliche Vorteil so eines Aufbaus der Ontologien ist die Unabhängigkeit der Konzepte von ihrem Kontext. Ähnliche Konzepte sollen nicht redundant wiedererstellt werden, nur weil sie in verschiedenen Szenarios unterschiedliche Ziele erfüllen. Um dies zu veranschaulichen, wird folgendes Beispiel vorgestellt[1].

Die eindeutige Bestimmung eines Buches in einer Ontologie für Buchläden wird mit Hilfe seiner ISBN realisiert. In diesem Fall müssen alle beteiligten Partner die solche Interpretationen akzeptieren, diese Identifizierungsregel einhalten. Bei den Bibliotheken, die ihre Bücher anhand der Zusammenstellung von Autor und Titel identifizieren, ist diese Ontologie unbrauchbar. Bei der Double Articulation dagegen ist genau dies möglich. Die Konzepte Buch, ISBN, Autor, Titel usw. und ihre Beziehungen untereinander sind von den Regeln der beiden Domänen, in unserem Fall Buchladen und Bibliothek, unabhängig. Die beiden können auf die gleiche Ontologienbasis zugreifen, dabei aber unterschiedliche Regeln befolgen.

Die Bibliothek würde dann zum Beispiel folgende Regel formulieren:

- Buch hat Titel
- Buch geschrieben von Autor
- Buch hat ISBN
- Buch entspricht Titel+Autor

Die Buchläden dagegen :

- Buch hat Titel
- Buch geschrieben von Autor
- Buch hat ISBN
- Buch kostet Preis
- Buch entspricht ISBN

SEAL benutzt die Technologien des Sematic Web zum Sammeln, Strukturieren und Integrieren von Informationen.Die Aufgabe von SEAL ist es den Umgang mit großen Mengen von Informationen zu erleichtern. Die Erstellung von Webseiten und spätere Integration der neuen Inhalte soll komplett automatisiert passieren. Unter Berücksichtigung dieser Anforderungen spielen die Ontologien eine sehr wichtige Rolle, weil es nur durch die semantische Annotation der Daten externer Anbieter ermöglicht wird, den gesamten Ablauf zu automatisieren. Für die Sicherung der Qualität der angebotenen Information, wurde in SEAL ein Workflow

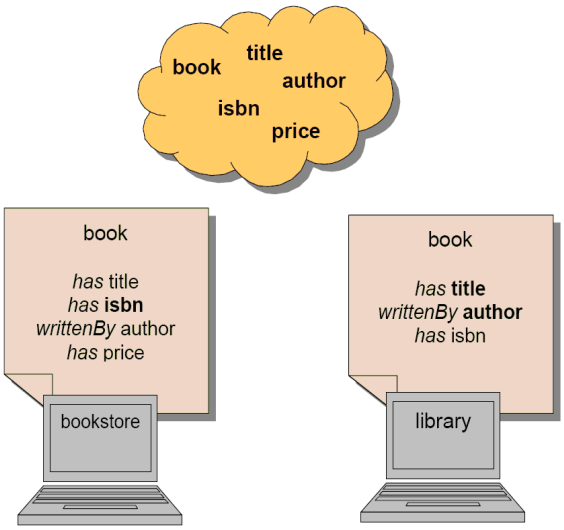

Abbildung 2: Buch-Ontologie in Bibliothek- und Buchladen-Domain

realisiert. In diesem Workflow muss ein Dokument drei Zustände durchlaufen, bevor es veröffentlicht wird. Sofort nach der Erstellung befindet sich ein Dokument in dem Zustand 'Private'. In dem Zustand 'Pending' bleibt ein Dokument solange es von anderen Mitgliedern der Webseite auf Korrektheit überprüft wird. Sobald es von dem Manager akzeptiert wird, landet das Dokument in dem Zustand 'Public' wo es für die restliche Welt frei verfügbar gemacht wird. Dieser Prozess ermöglicht es die Qualität der veröffentlichten Informationen zu garantieren.

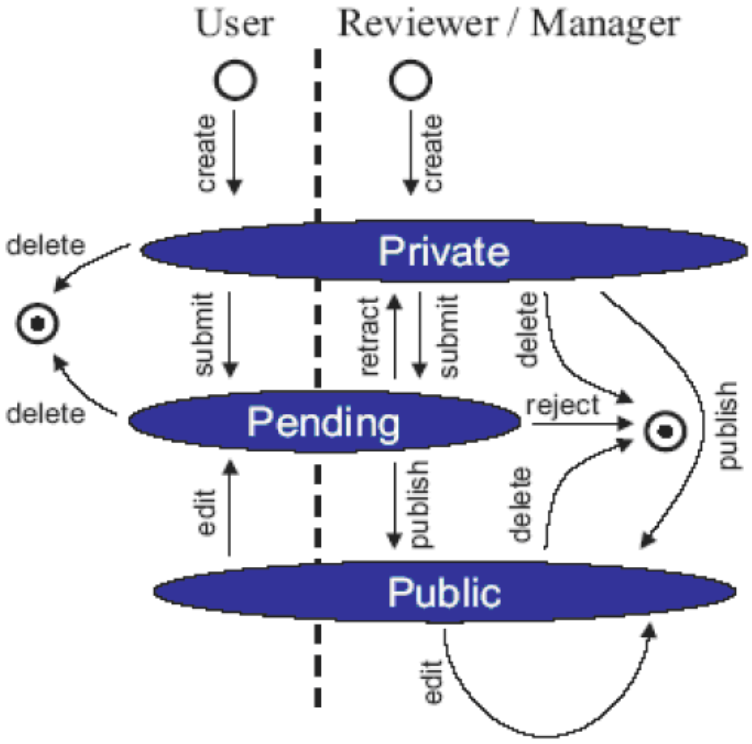

Abbildung 3: SEAL Workflow

3 Werkzeuge

Im diesem Kapitel werden kurz Werkzeuge präsentiert, die heute für die Lösung der Probleme des Semantic Web vorhanden sind. Es geht hier nicht darum die ganze Dokumentation zu präsentieren, sondern die Hauptmerkmale zu erläutern, damit dem Leser ein Überblick über ihre mögliche Anwendung gegeben wird.

3.1 Jena

Jena [11] ist ein Java Framework für die Erstellung von Semantic Web Applikationen. Es hat folgende Funktionen:

- RDF API für die Generierung und Verarbeitung von RDF-Dokumenten
- Unterstützung von diversen DB-Systemen z.B. Oracle
- Reasoning Subsystem für RDFS, OWL/Lite und Teile von OWL/Full
- Ontology Subsystem für die Unterstützung bei der Arbeit mit RDFS, OWL, DAML+OIL

- Unterstützung der Sprache RDQL, für die Erstellung von Anfragen an Ontologien

Eine ausführliche Dokumentation für die Jena API ist unter http://jena.sourceforge.net/ zu finden.

3.2 Protégé

Protégé [8] ist:

- ein Werkzeug um Ontologien zu erstellen und Daten in diese einzutragen
- OpenSource
- eine java-basierte Plattform, die um Plug-Ins erweitert werden kann
- eine Bibliothek die von anderen Anwendungen verwendet werden kann um Daten einzusehen und diese zu manipulieren
- ursprünglich zur Wissensrepräsentation in der medizinischen Informatik von der University of Stanford entwickelt worden

Protégé bietet:

- Erstellung und Manipulation von Ontologien
- Erstellung von Informationen zur weiteren Verwertung
- Lesen, Schreiben und Interkonvertierung folgender Formate:

  Relationale Datenbanken (ODBC)

  CLIPS

  UML / XMI

  XML / XML Schema

  RDF

  Topic Maps

  DAML+OIL

  OWL
- Eine benutzerfreundliche GUI
- Mehrbenutzerfähigkeit
- Ein Protege-Server, mehrere Clients, Wissensdatenbank liegt auf Server, Abgleich mit Clients in Echtzeit
- Echtes verteiltes Arbeiten an einer Wissensdatenbank

4 Master Thesis

In diesem Kapitel wird ein prototypisches System MOnTo 0.3.b zur Realisierung des Zugriffs auf eine heterogene Umgebung präsentiert und die Erweiterungen dieser Applikation durch eine Master Thesis vorgestellt.

Die Applikation ist im Rahmen des Masterprojektes 'Ferienclub' an der Hochschule für Angewandte Wissenschaften Hamburg entstanden. Ziel war es einen Prototypen zu bauen, der die Probleme der Heterogenität beim Datenaustausch in einer semantisch annotierten Umgebung löst. Als Anwendungsfall war die Metapher des Ferienclubs ausgewählt worden. Es sollte gewährleistet werden, dass unterschiedliche Autoanbieter ihr Angebot dem Besucher des Clubs zur Verfügung stellen können, ohne ihre Ontologien dafür extra anpassen zu müssen. Die ganze Anpassung und der Ontologiemerge soll von der Applikation durchgeführt werden. Dabei soll die Beteiligung des Menschen an diesem Prozess so klein wie möglich bleiben. Die folgende Abbildung macht die Gesamtarchitektur deutlich:

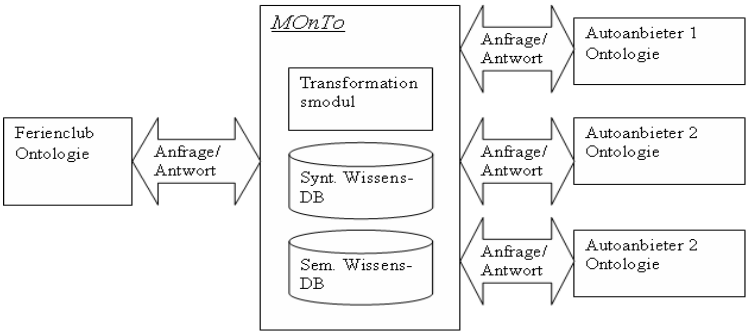

Abbildung 4: MOnTo Architektur

Die heutige Version von MOnTo 0.3b wurde mit Hilfe der Jena2 API realisiert und ist in der Lage die Teilaufgabe der angestrebten Ziele zu lösen. Sie verfügt über die folgenden Funktionalitäten:

- Einlesen und Darstellung der Ontologien in OWL und RDFS
- Halbautomatischer Merge zweier Ontologien
- Einfügen, Löschen, Umbenennen der einzelnen Slots einer Ontologie
- Erstellung der und Zugriff auf die eigene Wissensbasis
- Automatische Übersetzung von Anfragen unter Verwendung von RDQL
- Topologische Überprüfung auf Gleichheit von zwei Ontologien

- Erzeugung und Speicherung der neuen Ontologien

Die Arbeit in diesem Gebiet hat gezeigt, dass die automatische Anpassung von neuen semantisch annotierten Informationsartefakten in ein bestehendes System sehr schwer und fast unmöglich zu realisieren ist. Das Thema der Master Thesis ist es nun, die Grenzen des Automatisierens herauszufinden. Zusätzlich soll herausgefunden werden, welchen Voraussetzungen die Umgebung und die zu integrierenden Informationsartefakte genügen müssen, um die Rolle des Menschen möglichst zu minimieren. Dabei wird ein methodischer Schwerpunkt des Ansatzes auf der Graphunifikation liegen. Es wird versucht, nicht nur durch ein rein syntaktisches Vergleichen die Bedeutung der Instanz aus einer fremden Ontologie zu bestimmen. Dies soll durch semantische Regeln und topologische Eigenschaften der zuständigen Ontologie erreicht werden. Dadurch soll die Trefferquote bei Suchanfragen deutlich erhöht werden und die möglichen Probleme der Mehrdeutigkeit beseitigt werden. Ein Beispiel dafür kann wie folgt aussehen. Ein Autoanbieter hat in seiner Ontologie eine Instanz 'Name'. Bei der Suche findet die Applikation durch syntaktisches Vergleichen zwei Einträge in ihrer Wissensdatenbank, die mit 'Name' bezeichnet werden. Zum einen handelt es sich um den Namen eines Menschen und zum anderen handelt es sich um den Namen eines Autos. Durch den topologischen Vergleich der Ontologien, die für die Beschreibung dieser Instanzen zuständig sind, und die Untersuchung der semantischen Regeln wird es möglich, die richtige Zuordnung zu treffen.

Das größte Risiko bei dieser Master Thesis ist der Zeitaspekt in Zusammenhang mit der Komplexität des Themas. Es ist offensichtlich, dass die angestrebten Aufgaben nicht in der dafür vorgesehenen Zeit von 6 Monaten komplett zu schaffen sind. Aus diesem Grund ist es sehr wichtig die Ziele klar abzugrenzen und bestimmte Bereiche eventuell offen zu lassen. Der Plan ist, diese Aufgabe am Ende des Semesters erledigt zu haben.

Literatur

[1] http://isdis.cs.uga.edu/SemNSF/SIGDMOD-Record-Dec02/Meersman.pdf

[2] http://www.semantic-web.at

[3] http://www.w3.org/DesignIssues/Semantic.html

[4] http://www.semanticweb.org

[5] http://www.w3.org/TR/rdf-schema/

[6] http://www.w3.org/2001/sw/WebOnt/

[7] http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/rdf_paper_final.pdf

[8] http://protege.stanford.edu/

[9] http://triplestore.aktors.org/demo/AKTiveSpace/

[10] http://www.w3.org/

[11] http://www.hpl.hp.com/semweb/jena.htm

[12] Asucion Gomez-Perez Mariano Fernadez-Lopez, Oscar Corcho Ontological Engineering Springer Verlag 2003