

1 Begriffe

- Nutzer** - Ein angemeldeter User des Ontowiki, der das PKM-Plugin installiert hat; nicht notwendigerweise mit den Konzepten des Semantic Web vertraut.
- Wissen** - Informationen und Daten in einem Kontext, die einer Vernetzung und Interpretation unterzogen werden. Es wird sich auf explizit gegebenes Wissen bezogen, und solches, welches man daraus ableiten kann.
- PKM** - Übersicht über das persönliche → Wissen eines → Nutzers sowie seine Sichtweise auf dieses Wissen (repräsentiert durch Tags und Notizen). Aufgrund des Semantic Web Hintergrunds sind diese Daten nicht vollständig isoliert, sondern teilweise mit anderen vernetzt.
- PIM** - Personal Information Manager/Management, Software zur Verwaltung persönlicher Daten wie Kontakte, Aufgaben, Notizen und Termine und somit wichtiger Bestandteil von → PKM. Einfachheit und Softwareergonomie haben eine sehr hohe Priorität.
- URI/URL** - Uniform Resource Identifier sind Zeichenfolgen, die beliebige Ressourcen, auch Webadressen, mittels eines beliebigen eindeutigen Namens identifizieren. URLs (Uniform Resource Locators) adressieren immer eine Webadresse.
- Bookmarks** - Sowohl Links auf interne Daten im Ontowiki (deren semantische Annotation genutzt werden kann) als auch auf externe Adressen, die zumindest als Literale gespeichert werden.
- Kontakt** - Eintrag im Adressbuch; anderer → Nutzer oder eine beliebige Person, die Ontowiki oder das PKM-Plugin nicht verwendet.
- Ontologie** - Die Gesamtheit unseres PKM-Modells, im Unterschied zu bestehenden Schemata wie FOAF, die nur Bestandteile davon darstellen.
- Gruppen** - Menge von → Nutzern, die gleiche Rechte (zum Beispiel zum Datenzugriff zwecks Informationsvernetzung) haben. Nutzer können Gruppen mit Erlaubnis eines Leiters beitreten und jederzeit austreten.

2 Konzepte

2.1 Bestehende Schemata und Ontologien

vCard Ontology

Allgemeine Kontaktinformationen lassen sich mit der vCard Ontologie einheitlich in RDF speichern. Zu dieser virtuellen Visitenkarte gehören unter anderem Name, Adresse und Telefonnummer.

RDF Calendar

Mit RDF Calendar lassen sich standardisierte *iCalendar*-Kalenderdaten wie zum Beispiel Events, Aufgaben oder Erinnerungen nach OWL überführen, so dass die Möglichkeit gegeben wird, diese Daten mit anderen Ressourcen zu verknüpfen.

Dublin Core

Dieser internationale Standard stellt Konventionen bereit, um Dokumente und Objekte im Internet zu beschreiben und zu annotieren. Metadaten wie Sprache, Autor oder Titel können neben weiteren Formaten auch in RDF/XML dargestellt werden.

Annotea Bookmark

Lesezeichen lassen sich mit diesem Schema verwalten und auch hierarchisch thematisieren. Auch Dublin Core Metadaten wie Beschreibung oder Datum können neben dem Thema mit einer URL verlinkt werden.

Tag Ontology

Zur einfachen Implementierung und Nutzung können Relationen zwischen Tags und Ressourcen angelegt werden. Mit der Nutzung von Tagging-Klassen ist es denkbar, eine Ressource mit verschiedenen Tags zu verbinden und zusätzliche Metadaten zu speichern.

2.2 Relevante Konzepte

Das Semantic Web und dessen formale Sprache

von RICK HEGEWALD

Die *Semantik* (griechisch *sémantikós* „bezeichnend“ - KLUGE Ethymologie) als die Lehre von den Bedeutungen bildet das Fundament des am Anfang dieses Rechercheberichtes noch unklaren Begriffes „Semantic Web“. Text, für das menschliche Individuum zwar meist sinnvoll – bedingt durch Kognition und damit der Fähigkeit zur Semantisierung (z.B. Apfel und Birne sind beides Früchte) – stellt sich einer Maschine nur als Zeichenkette dar. D.h. ein Computer versteht nicht, er kann diese Art von Information nicht interpretieren. In der Anwendung, etwa der Recherche bei diversen Suchmaschinen, besteht das Problem darin, die tatsächlich gesuchte Information in der Fülle vorhandener (zudem dezentraler und heterogener) Informationen zu finden. Außerdem lässt sich implizites Wissen (z.B. dass Leipzig und Dresden sächsische Städte sind) i.A. nur schwer recherchieren.

Nun besteht die Möglichkeit die Zeichenkette mit Hilfe bestimmter Verfahren um Annotationen zu erweitern. Doch wie wäre es, schon von vornherein andere Wege der Verarbeitung und Speicherung von Informationen zu beschreiten, nämlich den der Festlegung der Bedeutung von maschinenlesbaren Daten? Auf jene Weise kann

die Information von einer Maschine verarbeitet werden. Damit wurde der noch wenig erkundete Pfad des *Semantic Web* beschritten.

RDF

von RICK HEGEWALD

Das RDF (engl. „Resource Description Framework“) stellt grundsätzlich eine formale Sprache zur Beschreibung von Metadaten oder strukturierter Information im Internet dar. Jene wurde vom W3C als Fundament für das Semantische Web entwickelt und gründet hauptsächlich auf der Idee Eigenschaften von Ressourcen in durch Computer verarbeitbare Form zu bringen. Dabei „geht es also nicht nur um korrekte Darstellung von Dokumenten, sondern auch um die Kombination und Weiterverarbeitung der enthaltenen Informationen“ (P. HITZLER (u.a.): *Semantic Web*, Berlin (2008) Springer. S.11).

Das *RDF-Modell* kann, wie meist zur Veranschaulichung von Modellen, als Graph dargestellt werden. Im Gegensatz zur Taxonomie von XML, sprich der Darstellung von Informationen in einer Baumstruktur, setzt RDF auf eine ontologische Strukturierung.

Hierarchie spielt also eine untergeordnete Rolle. Vielmehr geht es um Vernetzung – Zyklen sind dabei wohlmöglich gerade von Interesse – und liegt auf der einen Seite begründet im Versuch einer Beschreibung von allgemeinen Beziehungen zwischen Ressourcen und andererseits im problemlosen Umgang mit dezentral verwalteten und gespeicherten Informationen im World Wide Web.

Das *RDF-Modell* setzt sich zusammen aus einem *Tripel* von Objekttypen – Ressource/ Subjekt, Eigenschaftselement/Prädikat und Objekt. Ressourcen sind dabei u.a. Webseiten oder Objekte. Das Prädikat erläutert das Subjekt und bezieht es auf das Objekt, welches den Wert des Prädikates beschreibt. Zur eindeutigen Identifikation der jeweiligen Objekttypen dient der URI. Eine besondere Art des Objektes stellt das *Literal* dar. Es dient zur eindeutigen Interpretation eines Bezeichners und ist eine Darstellung für Datenwerte in RDF.

„Das *RDF-Modell* ist unabhängig von einer speziellen Darstellungsform.“

(http://de.wikipedia.org/wiki/Resource_Description_Framework)

D.h. der *RDF-Graph* wird serialisiert, z.B. in N3, N-Triples oder Turtle. Die Syntax dieser verschiedenen Methoden zur Umwandlung von komplexen Objekten und ihrer Beziehungen in lineare Zeichenketten erscheint mir in diesem groben Überblick nicht sinnvoll ausführlich zu beschreiben. Es sei soviel gesagt, dass es im Wesentlichen um eine „Ansammlung von Tripeln [...] [.] in beliebiger Reihenfolge hintereinander geschrieben“ (SW) geht. Die oben genannten Methoden unterscheiden sich hierbei in der Art der Darstellung dieser *Tripel*.

Eine Besonderheit ist die XML-Serialisierung, welche auf Baumstrukturen beruht. Die sogenannte *RDF/XML-Darstellung* kodiert dementsprechend *RDF-Tripel* in einer Hierarchie.

Was bedeutet das nun für unser PKM in *OntoWiki*? Da unser „Projekt“ noch in den Kinderschuhen steckt, lassen sich nur einige bereits recherchierte Ideen dazu formulieren:

Grundlegende Informationen über einen Nutzer, wie dessen Name, Geburtsdatum, Geschlecht, Notizen etc., werden nicht nur in bloßen Datenbankspalten gespeichert,

sondern mittels eines RDF-Modells und Verknüpfung bestimmter Knoten (quasi als Schnittstelle) mit anderen Modellen des OntoWikis zu einem großen ontologischen Netz von Informationen zusammengefügt.

Dabei lässt sich bereits implizites Wissen, wie z.B. Herr Mustermann hat am gleichen Tag Geburtstag wie Frau Unbekannt, extrahieren.

Erweitert man nun diese Grundfunktionen durch die Möglichkeit Notizen zu schreiben, wichtige Termine und Veranstaltungen in einen Kalender einzutragen oder seinen Wohnort in einer Landkarte wieder zu finden und vieles mehr, so entwickelt sich ein schier grenzenloses Netz an durch den Computer verarbeitbarer Informationen. Herr Mustermann wüsste nicht nur, dass Frau Unbekannt mit ihm gemeinsam Geburtstag hat, sondern könnte sie zu Kaffee und Kuchen einladen, denn durch Zufall befinden sich beide gerade auf einer Konferenz zum „Gesundheitsbewusstsein im Alter“ in Leipzig. Der Kalender liefert die nötige Verknüpfung. Über die MAP-Funktion findet Frau Unbekannt zudem heraus, dass die Konferenz nicht weit von der Arena Leipzig entfernt ist und entschließt sich kurzfristig, da aktuelle Konzerte angezeigt werden (siehe Mashup-Gruppe), zu Howard C. zu gehen.

RDF-Schema (RDF-S)

von ANTHONY VOLKMANN

RDF-Schema ist eine von W3C spezifizierte Erweiterung von RDF. Das RDF-Datenmodell bietet keinerlei Möglichkeiten, Ressourcen und Eigenschaften in Beziehung zueinander zu setzen. Zu diesem Zwecke bedient man sich des RDF-Schemas. Allerdings ist RDF-Schema kein Schema im eigentlichen Sinne, sondern erweitert RDF um Konstrukte, mit denen man Schemata erstellen kann. Man kann also sagen, dass RDF-S durch RDF beschrieben wird, und es gleichzeitig erweitert.

Zu den genannten Erweiterungen gehört vor allem das Klassenkonzept und die damit verbundenen Möglichkeiten der Vererbung, Mehrfachvererbung und Spezialisierung. Weiterhin ist es möglich, Wertebereiche und Definitionsbereiche von Eigenschaften festzulegen. In unserem Projekt könnte vor allem die Vererbung und die Arbeit mit Unterklassen, aber auch die Möglichkeit von Eigenschaften interessant sein. Zum Beispiel könnte man bei Notizen oder auch Terminen die Eigenschaft *Sichtbarkeit* festlegen, welche dann angibt, ob diese privater oder öffentlicher Natur, und somit für andere Benutzer sichtbar sind.

Zusammen mit RDF stellt RDF-Schema ein mächtiges Werkzeug im Bezug auf das Semantic Web dar. Jedoch sollte man sich nicht allein auf RDF-Schema verlassen. Vor allem bei umfangreichen Projekten kann es schnell sehr unübersichtlich werden.

OWL

von ANDREAS WEISE

Die *Web Ontology Language* (OWL) ist eine vom W3C standardisierte Sprache zur Beschreibung von Ontologien. Sie bietet umfassendere Ausdrucksmittel als RDF-S, um auch komplexes Wissen gut modellieren zu können. Dazu verwendet OWL genau wie RDF-S vor allem Klassen und Properties, erlaubt jedoch erweiterte Beziehungen dieser untereinander.

So kann man nicht nur mittels einfacher Klassen und Subklassen Hierarchien aufbauen, sondern auch komplexe Klassen aus diesen zusammensetzen. Dafür werden sog. logische Konstruktoren verwendet, die eine neue Klasse als Durchschnitt (`owl:intersectionOf`), Vereinigung (`owl:unionOf`) oder Komplement (`owl:complementOf`) bestehender Klassen beschreiben.

Eine weitere Möglichkeit zur Definition komplexer Klassen stellen Rolleneinschränkungen (`owl:Restriction`) dar. Eine Klasse ist in diesem Fall eine Menge von Objekten, die bezüglich bestimmter Rollen vorgegebene Bedingungen erfüllt. Sie kann z.B. dadurch definiert werden, dass sie aus allen Objekten besteht, für die eine vorgegebene Rolle nur Werte einer Klasse oder auch genau einen bestimmten Wert annimmt. Man könnte beispielsweise definieren, dass die Klasse der Männer aus allen Personen besteht, die den Wert „maennlich“ bezüglich der Eigenschaft „Geschlecht“ haben.

Es ist auch möglich, Kardinalitätseinschränkungen festzulegen (z.B.: „ein Projekt hat mindestens einen Mitarbeiter“).

Generell unterscheidet OWL abstrakte Rollen (`owl:ObjectProperty`) zwischen Individuen und konkrete Rollen (`owl:DatatypeProperty`), die Individuen Datenwerte (Literele) zuordnen. Außerdem können Rollen – wie schon in RDF-S – verfeinert werden („MutterVon“ als `rdfs:subPropertyOf` von „ElternteilVon“). Neu dagegen ist, dass man ihnen Eigenschaften wie Transitivität, Symmetrie oder Funktionalität zuordnen oder eine Rolle als invers zu einer anderen definieren kann.

Mit erweiterten Ausdrucksmitteln wird natürlich das Schlussfolgern komplexer und ineffizienter – bis hin zur Unentscheidbarkeit. Deshalb bietet OWL drei Teilsprachen mit verschiedener Ausdrucksstärke: *OWL Lite* ist als Fragment konzipiert, das nur grundlegende Elemente bietet, um zumindest Taxonomien beschreiben zu können. So dürfen beispielsweise Zahlenrestriktionen nur mit den Werten 0 und 1 verwendet werden. Diese Teilsprache ist in der Praxis jedoch wenig relevant. *OWL DL* zielt auf größtmögliche Ausdrucksstärke bei gleichzeitig gewährter Entscheidbarkeit. Es ist wesentlich weniger restriktiv und wird von den meisten Werkzeugen unterstützt. *OWL Full* enthält als einzige Teilsprache RDF-S komplett. Es ist dadurch mitunter unentscheidbar und wird meist nicht vollständig unterstützt.

OWL ist für unser Projekt in verschiedener Hinsicht relevant:

Einerseits entwickeln wir unser Modell auf Grundlage bestehender Ontologien, die (wie zum Beispiel RDF Calendar oder FOAF) auch OWL verwenden, und müssen somit in der Lage sein, vorgegebene OWL-Serialisierungen zu interpretieren, um diese Ontologien in unsere einbinden zu können.

Andererseits kann es natürlich auch vorkommen, dass wir mit RDF-S an Grenzen stoßen und selbst OWL verwenden möchten. Genau lässt sich dies allerdings momentan noch nicht absehen.

FOAF

von CHRISTIAN PETZOLD

FOAF steht für *Friend Of A Friend*, zu deutsch „Freund eines Freundes“. Dies erlaubt auch gleich eine grundsätzliche Vorstellung der Zielstellung des FOAF-Projektes

(<http://www.foaf-project.org/>), das im Jahre 2000 ins Leben gerufen wurde. Ein Freund eines Freundes könnte auch ein Freund von mir sein, nur kenne ich ihn noch nicht. Es existiert nur eine Verbindung über meinen Freund. FOAF will nun in solchen Situationen unter anderem der Mittler sein, der diese Verbindung findet und darstellt. Anders gesagt: FOAF ist sozusagen ein „social network“, das jedoch keine zentrale Datenbank benötigt.

Um diese Vorstellung zu verwirklichen, muss ein Computer Daten der Personen vergleichen können. Er sollte also in der Lage sein, einen Sinn hinter veröffentlichten Informationen zu erkennen. Dafür wird bei FOAF das sogenannte RDF-Schema aufbauend auf XML-Vokabular genutzt. Dieses ist in der Lage, Klassen (resources) und Eigenschaften (properties) zu definieren, die dann in einem RDF-Dokument genutzt werden können. Ein Beispiel für eine Klasse ist die Klasse `Person`. Diese „Person“ wird durch ihre Eigenschaften näher beschrieben: Es gibt also z.B. Eigenschaften wie Name, Homepage, E-Mail-Adresse usw.

Es ist klar, dass es zur Klasse `Person` noch einige weitere Eigenschaften gibt, die die jeweilige Person beschreiben. Auf der *FOAF Vocabulary Specification*-Seite (<http://xmlns.com/foaf/spec/>) werden jedoch auch weitere Klassen wie zum Beispiel `Document`, `Group` oder `Image` beschrieben, so dass sich sehr viele weitere Möglichkeiten der Beschreibung sowie Vernetzung von Daten ergeben. Es sollte jedoch erwähnt werden, dass sich manche dieser Klassen noch im Teststatus befinden oder teilweise instabil arbeiten.

Eine wichtige Eigenschaft von `Person` ist `knows`. Diese verlinkt Personen untereinander und stellt damit einen wichtigen Bestandteil des FOAF-Projektes dar, denn nur so ist es möglich, ein soziales Netz zu errichten und zu traversieren.

Eine nicht ganz unwichtige Fragestellung ist die nach der eindeutigen Identifizierung der Klasseninstanzen. FOAF versucht, allgemeine Eigenschaften wie zum Beispiel `foaf:person_uri` zu vermeiden und die Eindeutigkeit über sogenannte *unambiguous properties* (unzweideutige Eigenschaften) sicherzustellen.

FOAF ist in dieser Hinsicht sehr flexibel ausgelegt und nutzt diese Flexibilität beim sogenannten *Mergen* von RDF-FOAF-Files. Einfach ausgedrückt werden zwei oder mehrere FOAF-Files zusammengemischt um ihren Informationsgehalt zu erhöhen und somit für Software auswertbar zu werden.

Im Internet gibt es einige Seiten die die Erstellung von FOAF-Dateien (meist `foaf.rdf` benannt) erleichtern, indem Formulare bereitgestellt werden. Es ist jedoch auch prinzipiell möglich, eine einfache Text-Datei zu benutzen.

Nun steht die Frage der möglichen Anwendungen von FOAF im Raum. Meist ist dafür spezielle Software nötig, die mit RDF-Dateien umgehen kann. Eine offensichtliche Möglichkeit der Nutzung ist die soziale Vernetzung. Verschiedene Personen können auf andere Personen mit gleichen Interessen verweisen, wie oben schon beschrieben wurde. Eine Software könnte diese Information dann extrahieren und herausstellen. Denkbar wäre es auch, die Anmeldung bei Internet-Services zu erleichtern, indem man einfach auf seine FOAF-Datei verweist. Weiterhin existieren etwas speziellere Anwendungen wie z.B. Visualisierung von Beziehungen oder jegliche Art von automatisierter Datenauswertung.

Für PKM stellt FOAF somit einen möglichen Ansatzpunkt zur Bildung von Gruppen (basierend auf gleichen Interessen), Visualisierung von Freundesbeziehungen bzw. Informations-Management (mithilfe speziellerer Klassen, wie oben dargestellt) dar. Inwiefern sich das FOAF-Konzept im PKM-Kontext als praktikabel erweist, wird sich zeigen. Jedenfalls stellt es theoretisch viele Möglichkeiten bereit, um bestimmte Aspekte von PKM zu realisieren.

SPARQL

von ANTONIA SIEGERT

SPARQL (*SPARQL Protocol and RDF Query Language*) ist eine Anfragesprache für RDF, die vom Aufbau der SQL ähnelt. Es werden Webseiten nach RDF-Graphen durchsucht und Ergebnisse formatiert wiedergegeben. Als Anfragemuster dienen die RDF-Tripel, in denen durch Anfragevariablen nach den passenden Inhalten gesucht werden kann.

```
PREFIX ex: <http://example.org/>
SELECT ?titel ?autor
WHERE
{
  ?buch ex:VerlegtBei <http://springer.com/Verlag> .
  ?buch ex:Titel ?titel .
  ?buch ex:Autor ?autor .
}
```

Mit PREFIX können am Beginn Abkürzungen erstellt werden, um die URIs nicht jedes Mal ausschreiben zu müssen und die Übersichtlichkeit zu verbessern. SELECT bestimmt die Ausgabeformatierung. Anfragevariablen wird immer ein '?' oder ein '\$' vorangestellt. In der WHERE-Klausel werden die RDF-Tripel angegeben, welche auch optional sein können (durch OPTIONAL{<Anfrage>}). Durch den UNION-Operator lassen sich dort Anfrage-Mengen vereinigen (mittels {<Anfrage>} UNION {<Anfrage>}).

Anfragen lassen sich durch FILTER einschränken:

```
[..]
?buch ex:Preis ?preis
FILTER (?preis < 35) .
[..]
```

Auf RDF-Literale lassen sich auch spezielle Operationen anwenden, wie zum Beispiel isURI(L), isLITERAL(L) oder isBLANK(L), welche dann in den Filter-Operand eingesetzt werden können.

Anschließend an die WHERE-Klausel lassen sich wie in SQL Darstellungseinstellungen vornehmen, wie:

- ORDER BY ?preis - oder DESC(?preis)
- LIMIT 5 - maximale Anzahl der Ergebnisse
- OFFSET 25 - Abfrageergebnisse beginnen an Stelle 25

Wenn die Quelle nicht bekannt ist, kann sie in Form von URIs angegeben werden, welche nach der SELECT-Klausel zu platzieren sind.

```
FROM <URI>  
FROM <URI>  
[...]
```

3 Beschreibung der Applikation

Schlussendlich geht es darum, den Einsatz und Nutzen unseres Moduls im OntoWiki zu beschreiben. Grundsätzlich vereinfacht das OntoWiki eine Wissensbasis – quasi als Landkarte von Informationen – zu visualisieren. Dabei setzt es u.a. auf verschiedene Sichtweisen der selben Objekte bzw. Daten.

Im Moment besteht die Möglichkeit neue Nutzer mit Passwort anzulegen. Wir verstehen unsere Aufgabe nun zu erst darin, die Registrierung zu erweitern, d.h. zusätzliche Personendaten abzufragen. Ein neues Fenster im linken Bereich der OntoWiki-Ansicht erweitert die bestehende Navigation um das PKM. Dort findet sich eine klare und selbsterklärende Struktur. Der Inhaltsbereich in der Mitte der Seite wird je nach Benutzereingabe verändert. Am rechten Rand erscheinen zusätzliche Informationen und Tools der jeweiligen Auswahl.

Im PKM-Fenster implementieren wir natürlich ein gewöhnliches PIM. Das Ziel ist es nun diese Basis um Funktionen zu erweitern. Modular gestaltet, sollte es auch später möglich sein, unsere gegebenen Schnittstellen des PKM-„Kerns“ zu nutzen und neue Features einzubinden.

Wir werden versuchen die folgenden Funktionen zu integrieren:

- Bookmarks - Speicherung von Verweisen zu interessanten Seiten und Wiki-Texten (interne und externe)
- Adressbuch - Einfache Speicherung von Benutzerdaten des OntoWikis (sofern public) und externer Adress- und Personendaten
- Notizbuch - Soll ein Gedanke oder eine Textstelle nicht verloren gehen, schnell eine Notiz hinterlassen
- Aufgabenliste - Was in mittelfristiger Zeit als zu erledigen gilt
- Tagging - Annotation von Wiki-Texten um zusätzliche Kommentare oder Informationen
- Kalender - Eintragen von Terminen, Veranstaltungen, Geburtstagen, usw.
- Map - Darstellung von Wohn-, Arbeitsstelle, Veranstaltungsort, usw.

Zuletzt sei noch erwähnt, dass es unsere Aufgabe und unser Grundsatz sein muss, die Inhalte einfach, benutzerfreundlich und ohne zusätzliche Kenntnisse der Syntax (z.B. RDF-S oder OWL) les- und veränderbar zu gestalten. Erfahrenen Benutzern sollte die Möglichkeit gegeben werden, auf die gewöhnlichen Funktionen zugreifen zu können. Denn eine Einschränkung des Grundgedankens des Semantic Web stellt ein solches PKM in gewisser Hinsicht dar. Doch um etwas Neues zu erreichen muss man üblicherweise Anderes aufgeben. So gewinnt man Einfachheit und Usability, verliert jedoch an Umfang und Komplexität.