

Recherchebericht

Gruppe: swp6-11

Inhaltsverzeichnis

1	Begriffe	1
1.1	Plug-in	1
1.2	Templatesystem	1
1.3	Blogeintrag	1
1.4	Web-Blog	1
1.5	Profilseite	1
1.6	Import Schnittstelle	1
1.7	Editieren	1
1.8	Exportieren	2
1.9	Klappstuhl-Treffen und Klappstuhl-Mitglieder	2
1.10	Update	2
2	Konzepte	2
2.1	Linked Data	2
2.2	JSON (JavaScript Object Notation)	3
2.3	RDF (Ressource Description Framework)	3
2.4	SPARQL Protocol And RDF Query Language	5
2.5	WordPress	6
2.6	WordPress-Plugins	6
2.7	WordPress Plugin API	6
2.8	PHP	7
3	Beschreibung der zu studierenden Aspekte	9
3.1	Informationen über den Klappstuhlclub e.V.	9
3.2	Verwaltung der Daten von Infoboxen auf Wikipedia	10
3.3	Professorenkatalog der Universität Leipzig	10

Der Umfang des Dokuments ist etwas größer ausgefallen als gefordert, da wir eine modifizierte Version des Aufgabenblatts von Herrn Hellmann bekommen haben, auf welchem deutlich mehr Aufgaben zu bearbeiten waren.

1 Begriffe

1.1 Plug-in

Ein Plug-in ist eine Erweiterung einer Software um zusätzliche Funktionalität. Das Plug-in folgt dem Prinzip der Umkehrung der Steuerung. Nach diesem Paradigma steuert das Framework die Ausführung von Unterprogrammen anstatt der Anwendung.

1.2 Templatesystem

Als Templatesystem wird eine Software bezeichnet, die Templates verarbeitet, d.h. Dateien mit Platzhaltern für Informationen. Die Platzhalter werden durch aktuelle Informationen aus einer RDF Datenbank ersetzt. Die Templates sind von den Daten unabhängig und wiederverwendbar.

1.3 Blogeintrag

Ein Blogeintrag ist ein Eintrag in dem Web-Blog des Klappstuhl e.V. Ein Eintrag beschreibt in der Regel ein bestimmtes Treffen der Mitglieder. Zu der Beschreibung gehören mindestens der Ort, die Uhrzeit sowie ein individueller Text.

1.4 Web-Blog

Ein Web-Blog ist ein auf einer Webseite geführtes Journal mit chronologisch geordneten Einträgen. Der Web-Blog des Klappstuhl e.V. gibt Auskunft über Mitglieder und die wöchentlichen Treffen.

1.5 Profilseite

Eine Profilseite ist eine HTML Seite mit den persönlichen Informationen eines Mitgliedes des Klappstuhl e.V. Zu den Informationen können alle das Mitglied betreffenden Daten aus der RDF Datenbank gezählt werden.

1.6 Import Schnittstelle

Als Import Schnittstelle wird eine Sammlung der Importmodule im Wordpress bezeichnet. Der Import der Daten erfolgt aus einer oder mehreren Dateien, einer Datenbank oder mit Autorisierung direkt. Ein Importmodul ist für genau einen Datentyp zuständig.

1.7 Editieren

Als Editieren wird der Vorgang bezeichnet, bei dem eine Person die Daten in Form von Tripeln nach der Syntax des wp-semantic Plug-ins im WordPress eigenen Blogtextfeld verändert.

1.8 Exportieren

Exportieren bezeichnet einen http-request an das WordPress Plug-in, bei dem die URI zu einem bestimmten Blogeintrag übergeben wird. Als Rückgabe erfolgen die in dem Blogtextfeld gespeicherten Tripel in einem maschinenlesbaren Format, wie XML oder JSON. Die Methodik dazu beschreibt Linked Data.

1.9 Klappstuhl-Treffen und Klappstuhl-Mitglieder

Als Klappstuhl-Treffen wird die wöchentliche Versammlung der Mitglieder des Klappstuhl e.V. bezeichnet. Das Treffen findet in den Städten Leipzig und Berlin gleichzeitig statt. Zu jedem Treffen wird ein extra Blogeintrag angelegt, in welchem festgelegte Daten in das Blogtextfeld eingetragen werden.

Als Klappstuhl-Mitglied wird jeder bezeichnet, der zum Klappstuhl-Treffen kommt. Jedes Mitglied hat eine Profiseite auf der Webseite des Klappstuhl e.V. Ein Mitglied, welches zum Treffen mit einem eigenem Klappstuhl kommt und auch der tatsächliche Eigentümer des Klappstuhls ist, wird als Klappstuhl-Präsident bezeichnet.

1.10 Update

Update bezeichnet den Vorgang des automatisierten bzw. programmatischen Hinzufügens oder Löschens von Tripeln nach der Syntax des wp- semantics Plug-ins im WordPress eigenen Blogtextfeld mittels http-request. Dies soll als Schnittstelle zur Android app dienen.

2 Konzepte

2.1 Linked Data

Das Ziel des semantischen Webs ist es, die Bedeutung von Informationen für Maschinen verständlich zu machen. Als Teil dieses semantischen Webs konzentriert sich Linked Data darauf, Daten standardisiert zu verlinken. Wir kennen das Web als Netz von Webseiten. Linked Data hingegen versucht, ein Netz aus Daten zu schaffen, die ähnlich verlinkt sind, wie Webseiten aber durch Maschinen verwertbar sind. Dafür werden Daten mit einem "*Uniform Resource Identifier*" (URI) identifiziert und können so auch auf andere solche Daten verweisen. Der bekannteste Vertreter dieser URIs ist der URL (Uniform Resource Locator), welcher den Ort einer Ressource im Netz angibt, beispielweise http. Man stelle sich also nun eine ähnliche Vernetzung von RDF Daten vor, wie sie jetzt zwischen den Webseiten besteht. Das ermöglicht es, dass Maschinen diese Daten ähnlich einer Datenbank behandeln und durchsuchen. Der Schritt, die gesamten Daten des heutigen Webs automatisch durchsuchen zu können, wäre gewaltig. Um das zu erreichen, prägte *Tim Berners Lee* die vier Regeln:

1. Verwende zur Bezeichnung von Objekten URIs.
2. Verwende HTTP URIs, so dass sich die Bezeichnungen nachschlagen lassen.
3. Stelle zweckdienliche Informationen bereit, wenn jemand eine URI nachschlägt (mittels der Standards RDF und SPARQL).
4. Zu diesen Informationen gehören insbesondere Links auf andere URIs, über die weitere Objekte entdeckt werden können.

2.2 JSON (JavaScript Object Notation)

JSON ist ein Datenaustauschformat. Es basiert auf einer Untermenge der JavaScript Programmiersprache. Im Vergleich zu XML werden Daten bei JSON viel einfacher strukturiert. Dadurch ist es mit vielen Anwendungen kompatibel und relativ einfach zu erlernen. Derzeit gibt es grundsätzlich in allen gängigen Programmiersprachen eine Anbindung bzw. eine Schnittstelle zu JSON. Der Zusammenhang zwischen PHP und JSON wird durch die Funktionen `json_encode()` und `json_decode()` hergestellt. Für das Projekt RDF2WP ist insbesondere die Tatsache wichtig, dass die JSON- Objekte (Arrays) sich damit leicht in PHP-Objekte (Arrays) umwandeln lassen (und umgekehrt).

2.3 RDF (Ressource Description Framework)

RDF Graphen, URI und RDF Triple Stores und RDF- Syntax

Die Grundlage für RDF bildet XML. Hier werden Daten in einer Baumstruktur dargestellt. Dies ist allerdings nicht optimal für eine *intuitive Beschreibung* der Daten und die Integration von Daten aus anderen Quellen.

RDF (Ressource Description Framework) basiert aus diesem Grund auf einem *Graph-orientierten Datenschema*. Es dient zur Angabe von Metainformationen für Web- Ressourcen, kodiert strukturierte Informationen und ist ein universelles maschinenlesbares Austauschformat.

Hierbei sind die bereits in XML vorhandenen **URI's (Uniform Resource Identifier)** von großer Bedeutung. Sie dienen zur *weltweit eindeutigen Bezeichnung von Ressourcen*. Eine **Ressource** kann jedes Objekt sein, was (im Kontext der gegebenen Anwendung) eine klare Identität besitzt. Sowohl Knoten als auch Kanten eines RDF Graphen werden mit URI's beschrieben. Die einzige Ausnahme sind Literale und "Blank Nodes", welche keinerlei Bezeichnung tragen.

Ein einfacher RDF- Graph unterteilt sich in Objekt, Prädikat und Subjekt. Die zu beschreibenden Dinge (Objekte) haben also eine Eigenschaft (Property) mit einem Wert (Subjekt). Diese drei Bestimmungsstücke werden als **RDF Tripel** oder auch **RDF Statement** bezeichnet.

Diese Tripel und das Verständnis des Umgangs mit ihnen sind für das Projekt von elementarer Bedeutung. Die Zusammenfassung der Tripel wird in **RDF Tripel Stores** realisiert. Diese dienen zur *Verwaltung der Datenbasis* im Semantic Web und basieren oft auf relationalen DBMS. Im Projekt werden RDF- Tripel sowohl aus dem RDF- Tripel Store des Professorenkataloges (hier Virtuoso) der Universität Leipzig als auch des Klappstuhlclubs durch einen SPARQL- Endpunkt ausgelesen und schließlich neue Daten importiert. Für die Darstellung von Tripeln in RDF- Dokumenten haben sich die **XML- Syntax**, sowie die **Turtle Syntax** (Anfrageformulierung in SPARQL) durchgesetzt.

Vokabulare und Ontologien

Unter einem **Vokabular** versteht man eine Zusammenstellung von Bezeichnern mit klar definierter Bedeutung. Als Beispiel sei das RDF- Vokabular genannt, welches die RDF/ XML Syntax definiert. Vokabulare werden allerdings auch zur Darstellung von Informationen herangezogen. Ein Beispiel ist das für unser Projekt relevante *Vokabular zur Beschreibung geschichtswissenschaftlicher Informationen über Personen und Professoren des Professorenkatalogs (CPM- Vokabular)*. Eine **Ontologie** definiert eine *gemeinsame Sichtweise auf eine Domäne*, indem sie *Konzepte* und *Beziehungen* dieser Domäne auf konsistente Weise beschreibt und *logische Schlussfolgerungen* ermöglicht.

RDFS (RDF Schema)

RDF bietet Informationen über *einzelne Ressourcen und deren Beziehungen*. **RDF Schema** erweitert dieses Model, indem nun Aussagen über eine Menge gleichartiger Objekte sogenannter *Klassen* getroffen werden sowie *logische Zusammenhänge* zwischen diesen definiert werden können. Dabei ist RDF Schema ein spezielles RDF Vokabular. Im Gegensatz zu einem RDF- Vokabular, welches nur Termini für einen *bestimmten Anwendungsbereich* zulässt, können im RDF Schema Aussagen über die *semantische Beziehung* der Termini eines beliebigen *nutzerdefinierten Vokabulars* gemacht werden. Die Möglichkeiten, sogenanntes *Schemawissen* zu spezifizieren, machen aus RDFS eine *Ontologiesprache*. Sie eignet sich zur Beschreibung von *leichtgewichtigen Ontologien*. RDFS bildet die Grundlage für das Vokabular des Professorenkatalogs. Dieses ist in OWL verfasst, welches RDFS erweitert.

OWL (Web Ontology Language)

RDF ist nur beschränkt für die Abbildung von *komplexen Zusammenhängen* geeignet. Beispielsweise lässt sich nicht ausdrücken, dass etwas nicht gilt (Negation) oder dass zwei Klassen disjunkt sind. Um komplexe Zusammenhänge zu definieren, benutzt man ausdrucksstarke Repräsentationssprachen. Diese basieren auf *formaler Logik* (wie auch RDFS) und ermöglichen *logisches Schlussfolgern*, so auch die Ontologiesprache **OWL**. Diese basiert auf

der Prädikatenlogik erster Stufe. Eine OWL- Ontologie besteht wie bei RDFS hauptsächlich aus Klassen („owl: Class“) und Propertyts (Rollen). Außerdem gibt es Individuen, als RDF-Instanzen von Klassen.

2.4 SPARQL Protocol And RDF Query Language

SPARQL ist eine **Anfragesprache** zur Abfrage von Instanzen aus **RDF-Dokumenten**, die seit dem 15. Januar 2008 als W3C-Spezifikation veröffentlicht ist.

Die **Syntax** orientiert sich an der *Turtle-Notation* von RDF-Dokumenten und dem Gerüst von SQL. Die RDF-Tripel werden in der Reihenfolge *Subjekt Prädikat Objekt* notiert und durch Punkt abgeschlossen, wobei URIs in spitzen Klammern < > und Literale in Anführungszeichen zu setzen sind. Dabei sind Turtle-Abkürzungen zulässig. Variablen werden durch ? oder \$ gekennzeichnet und sind zulässig als Subjekt, Prädikat oder Objekt.

Es gibt verschiedene **Ausgabeformate** mit denen eine Anfrage beginnt:

SELECT ?var1 ?var2	liefert eine Tabelle mit den Variablenwerten zurück
CONSTRUCT <RDF-Schablone in Turtle>	liefert strukturiertes Turtle-Ergebnis nach Schablone
ASK	prüft nur, ob es Ergebnisse gibt

Hauptbestandteil der Anfrage ist das Graphmuster (**WHERE Anweisung**). Nach dieser können Tripel aufgezählt werden (Schnitt), die die Struktur der gesuchten Elemente beschreiben. **Verfeinerungen** der Anfrage bieten folgende Befehle:

UNION	erlaubt die Angabe alternativer Teile eines Musters (Vereinigung)
OPTIONAL	erlaubt die Angabe optionaler Teile eines Musters
{...}	gruppiert die Tripel
FILTER	Vergleich von Datenliteralen mit Operatoren <, =, >, <=, >=, != bietet zudem RDF spezifische Filterfunktionen wie isURI(A) oder isBLANK(A)
&&,	Filterbedingungen können mit Booleschen Operatoren verknüpft werden

Weiterhin gibt es die **Modifikatoren** ORDER BY, LIMIT, OFFSET DISTINCT die SELECT-Anfragen sortieren und einschränken können.

Die **Bedeutung** des Konzepts **für das Praktikum** leitet sich zum einen aus der Notwendigkeit für den Professorenkatalog ab, da dessen Daten mit SPARQL abgefragt werden müssen und zum anderen aus dem Fakt, dass SPARQL eine standardisierte Schnittstelle zwischen Triple-Stores, in welchem die RDF-Daten strukturiert gespeichert sind, und unserem Wordpress-Plugin, welches diese Daten präsentieren soll, ist. Somit ist die Unabhängigkeit

von einem speziellen Triple-Store gewährleistet.

Bei der Recherche ist aufgefallen, dass **keine Datenmanipulation** (d.h. Insert, Update, Delete), wie zunächst angenommen, möglich ist. Dahingehend muss evtl. in dieser Hinsicht (SPARUL) noch recherchiert werden.

2.5 WordPress

WordPress ist ein PHP/MySQL-basiertes, kostenloses System zur Verwaltung von Inhalten von Webseiten (Content Management System). Wegen seiner automatisierten Archivierungs- und Navigierungsfunktion eignet es sich insbesondere für Pflege von Weblogs. Der Benutzer kann dabei eigene Kategorien und Tags erstellen und seine Einträge diesen zuordnen, wodurch sie semantisch miteinander verbunden werden und später einfacher aufzufinden sind. Neben der direkten Präsentation von Inhalten können Leser auch Feeds mit neuesten Einträgen abonnieren. In WP können permanente Links (sog. Permalinks) zu den einzelnen Blogbeiträgen spezifiziert werden, die bei Verlinkung verwendet werden. Unter den seitens WP zur Verfügung gestellten Formaten sind z.B. „/Jahr/Monat/Tag/Titel_des_Eintrags“ sowie „/archive/Eintrags_ID“. Der Nutzer kann aber auch selbst Formate angeben. Die Verwaltung und Funktionalität von WordPress kann durch externe Plugins geändert und erweitert werden. Ein Plugin muss in den Ordner wp-content/plugins hochgeladen und danach über das Plugins-Menü aktiviert werden.

2.6 WordPress-Plugins

Die von unserer Gruppe installierten WordPress-Plugins sind wp-semantic und osm. Mittels wp-semantic können mit vorgegebener Syntax Tabellen in einen WP-Eintrag eingebettet werden. OSM ist die Abkürzung von OpenStreetMap. Mit dem Plugin können Karten und Geodaten in Weblogs dargestellt werden. Durch Spezifizierung von Angaben auf einer Karte wird Shortcode automatisch generiert der dann in eine Webseite eingefügt werden kann.

2.7 WordPress Plugin API

Anfangs ist es wichtig einen kurzen, einzigartigen Namen für das Plugin zu wählen, um Konflikte vorzubeugen. Ist z.B. der Name Ihres Plugins „RDF to WordPress“, wählen Sie die Kurzform „rdf2wp“. Die Plugin-Dateien sollen Sie dann im Ordner „rdf2wp“ speichern und die Hauptdatei „rdf2wp.php“ benennen. Es empfiehlt sich alle Funktionen und alle Datenbank-Parameter mit dem Präfix „rdf2wp_“ zu versehen.

2.8 PHP

Was ist PHP?

PHP (Akronym für "PHP: Hypertext Preprocessor") ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source Skriptsprache, welche speziell für die Webprogrammierung geeignet ist, und in HTML eingebettet werden kann.

```
<html>
  <head><title>Beispiel</title></head>
  <body><?php echo "Hallo, ich bin ein PHP-Skript!"; ?>
</body>
</html>
```

Beachten Sie den Unterschied zu einem Skript, welches in anderen Sprachen wie Perl oder C geschrieben wurde. Anstatt ein Programm mit vielen Anweisungen zur Ausgabe von HTML zu schreiben, schreibt man einen HTML-Code mit einigen eingebetteten Anweisungen, um etwas auszuführen. Der PHP-Code steht zwischen speziellen Anfangs- und Schlusstags, mit denen man in den PHP-Modus und zurück wechseln kann. Was PHP von clientseitigen Sprachen wie JavaScript unterscheidet, ist dass der Code auf dem Server ausgeführt wird. Sollten Sie ein Skript wie das obige auf ihrem Server ausführen, würde der Besucher nur das Ergebnis empfangen, ohne die Möglichkeit herauszufinden, wie der zugrundeliegende Code aussieht.

Formulare verarbeiten

Eine der mächtigsten Funktionen von PHP ist die Art, wie HTML-Formulare verarbeitet werden. Wenn der Benutzer das Formular ausfüllt und den Submit-Button anklickt, wird eine PHP-Seite aufgerufen. Die Variablen `$_POST[]` und `$_GET[]` werden für Sie automatisch von PHP gesetzt.

Variablen

Variablen werden in PHP dargestellt durch ein Dollar-Zeichen (\$) gefolgt vom Namen der Variablen. Bei Variablen-Namen wird zwischen Groß- und Kleinschreibung unterschieden (case-sensitive).

Plugin Header

Die Hauptdatei Ihres Plugins muss einen Header für die Standardinformationen beinhalten. Dieser wird von WordPress automatisch erkannt und in der Plugin-Verwaltung benutzt. Ohne den Header wird Ihr Plugin nie aktiviert.

WordPress Plugin Hooks

Die Hooks sind dafür da, um Ihr Plugin ins WordPress zu integrieren und in Bewegung zu setzen. Zurzeit gibt es über 1000 Hooks. Eine vollständige Liste finden Sie unter http://adambrown.info/p/wp_hooks. Es gibt zwei Arten von Hooks: Aktionen und Filter.

Aktionen

Aktionen sind Hooks, die an speziellen Stellen in WordPress ausgeführt werden. Ihr Plugin kann spezifizieren an welchen Stellen, es seine Funktionen aufruft.

```
add_action($tag, $function_to_add, $priority = 10, $accepted_args = 1)
```

Verankert Ihre Funktion mit einer Aktion.

```
remove_action($tag, $function_to_remove, $priority = 10, $accepted_args = 1)
```

Löst die Verankerung zwischen einer Funktion und einer Aktion auf. Alle Parameter die beim Verankern verwendet wurden, sollen übereinstimmen.

Filter

Filter sind Hooks, die ausgeführt werden bevor die Daten zu Datenbank gesendet werden oder bevor sie auf dem Schirm angezeigt werden. Ihr Plugin kann auf diesen Datenfluss zugreifen und es modifizieren.

WordPress Options Mechanism

WordPress besitzt einen Mechanismus um die individuellen Parameter Ihres Plugins in der WordPress-Datenbank zu speichern. Die Werte können elementare Variablen, Arrays oder PHP-Objekte sein und sind auf 232 Bytes begrenzt. Die Liste von den WordPress-Parametern finden Sie unter http://codex.wordpress.org/Option_Reference. Der Name soll einzigartig sein, damit kein Konflikt mit den Parametern des WordPress und der anderen Plugins entsteht, am besten setzt man einen Präfix `pluginname_` vor jedem Parameternamen. Benutzen Sie Unterstriche um die Wörter zu trennen und benutzen Sie keine Großbuchstaben.

<code>add_option(\$option, \$value = '')</code>	Erzeugt einen neuen Parameter in der Datenbank. Existiert der Parameter bereits, tut sich nichts
<code>update_option(\$option, \$newvalue)</code>	Verändert oder erzeugt einen Parameter in der Datenbank. Gibt eine Bestätigung zurück
<code>get_option(\$option, \$default = '')</code>	Gibt den Wert des Parameters zurück
<code>delete_option(\$option)</code>	Löscht den Parameter aus der Datenbank

3 Beschreibung der zu studierenden Aspekte

3.1 Informationen über den Klappstuhlclub e.V.

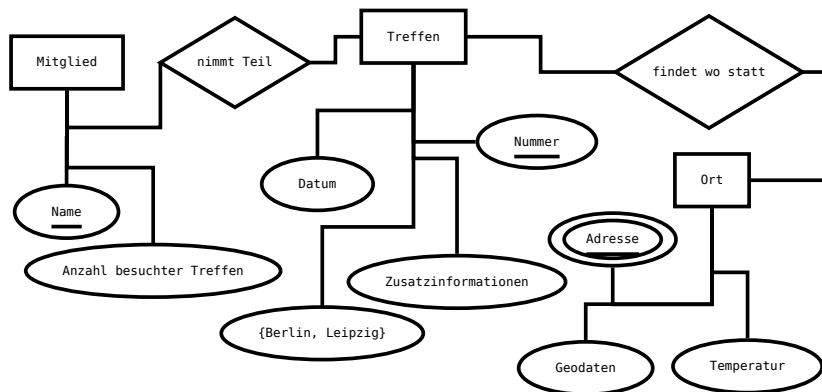
Auf der Homepage des Clubs sind bereits folgende Daten bereitgestellt:

- Darstellung von Nummer des Treffens, Stadt und Ort über den Titel des Blogeintrags
- Zusatzinformationen wie genaue Adresse, Anfahrt und Beschreibung des Treffens als Text (Darstellung variiert je nach Ersteller des Blogeintrags von XML-Listen bis zu einfachen HTML Aufzählungen)
- Datum des Treffens
- Kommentare zum Treffen
- Geotag des Treffpunkts

Die Mitglieder sind dem jeweiligem Treffen dadurch zugeordnet, dass jedes Treffen, an dem teilgenommen wurde, sich in einem Ordner mit dem Namen des Mitglieds befindet. Jedes Treffen ist außerdem noch im Ordner der jeweiligen Stadt zu finden.

Inwiefern die Datenstruktur für uns zu gebrauchen ist und wie der Import zu realisieren ist, muss noch besprochen werden, da sich die Strukturen wahrscheinlich nur teilweise zu unseren umwandeln lassen können.

Die Domäne lässt sich durch ein einfaches ER- Diagramm umreißen:



3.2 Verwaltung der Daten von Infoboxen auf Wikipedia

Infoboxen sind standardisierte Tabellen, die auf Wikipedia am Anfang eines Artikels angezeigt werden und grundlegende Daten enthalten. Für bestimmte Arten von Artikeln gibt es jeweils eine Vorlage für entsprechende Infoboxen, beispielsweise für Squash Spieler. Die Infobox für einen Squash Spieler soll nach dieser Vorlage dann Informationen wie Name, Geburtsort, Geburtsdatum, derzeitiger Wohnort oder Körpergröße enthalten. Um diese Daten maschinell verwertbar zu machen, nutzt Wikipedia Mikroformate. Die HTML-Tags, welche durch die Vorlage für die Infobox erzeugt werden, enthalten das hCard Mikroformat. Dieses ist speziell zur Darstellung von Kontaktdaten von Personen, Orten oder Firmen gedacht. Es ermöglicht die automatische Weiterverarbeitung der Daten des Spielers, beispielsweise in Katalogen auf Wikipedia.

In ähnlich einfacher Weise soll es mit dem WordPress-Plugin **DataPress** möglich sein, strukturierte Daten zu Blogs hinzuzufügen und darzustellen. Es soll dabei noch mehr darauf geachtet werden, dass die Blogger sich nicht um die Erstellung und die Verarbeitung der Daten kümmern müssen. Wesentlichen Eigenschaften des Blogs, wie “click and you see it” oder textähnliche Verarbeitung mit “copy-and-paste”, sollen erhalten bleiben. Die entstehenden Daten sollen dann über Formulare zugänglich sein, vergleichbar mit dem erstellen einer Slideshow.

3.3 Professorenkatalog der Universität Leipzig (Catalogus Professorum Lipsiensis)

Der Leipziger Professorenkatalog ist eine weltweit verfügbare Datenbank. Die Inhalte des Katalogs werden nach dem Standard des Semantic Web in RDF kodiert und sind als Linked-Data in diesem Format und über eine Anfrage Schnittstelle, einem sogenannten (*SPARQL-Endpunkt*), abrufbar.

Das Modell Catalogus Professorum Model stellt Vokabular für wesentliche Konzepte zu Professoren, Informationen über Professoren und deren Beziehung untereinander dar (kurz *CPM-Vokabular*). Das Vokabular ist in der Beschreibungssprache OWL repräsentiert.

Die Klassen des CPM- Vokabulars sind überwiegend in einer hierarchischen Struktur angeordnet. Da Professoren beispielsweise eine Teilmenge von Person sind, deren Informationen gesammelt werden, bildet “cpm: Professor” eine Unterklasse von “cpm:Person”. Hierbei werden Eigenschaften einer Klasse vererbt. Das Vokabular ist so konstruiert, dass es weltweit für alle Professorenkataloge genutzt werden kann. Es besteht aus den Konzepten *Personen und Professoren, Vornamen, Veröffentlichungen, Körperschaften und Geographische Gebiete*. Da zur Beschreibung historischer Personenkataloge noch keine Vokabulare existierten, wird in diesem Bereich ausschließlich auf das CPM- Vokabular zurückgegriffen. Lediglich für Geographische Gebiete wird das *Spatial Hierarchy Vocabulary* (shv) genutzt.

Ausgehend von den verschiedenen Nutzergruppen, dem Projektteam, den Experten und den interessierten Internetnutzern, wurden die drei Web- Applikationen *Semantisches Datenwiki*, die Website der Universität und der Projektserver etabliert. Das semantische Datenwiki stellt die Datenbasis für das Projekt. Hier läuft die derzeit stabile Version 0.85 des Applikationsframeworks *OntoWiki*. Das Einfügen und Pflegen von Professorendaten erfolgt dort Wiki- basiert durch das Projektteam. Die Website stellt die öffentliche Schnittstelle des Projektes dar. Alle Daten können ebenfalls durch Angabe der URI ihrer Ressource im RDF- Format geliefert werden. Damit bietet der Projektserver eine sehr leistungsstarke Schnittstelle für *Linked Open Data Web*. Dadurch können auch externe Applikationen Informationen des Kataloges nutzen.

Der Projektserver ist unter der URL des Vokabulars erreichbar. Auf ihm ist die neueste Version des OntoWiki installiert. Wichtig ist hierbei auch die verwendete Datenbank der *RDF-Store Virtuoso*. Dieser bietet einen leistungsfähigen *SPARQL-Endpunkt*, der für das zu erstellende Projekt genutzt werden muss. Mögliche Rückgabeformate der Anfragen sind hier HTML oder JSON.