

# Recherchebericht

## Begriffe

---

### Ontologie:

Eine Ontologie ist eine sprachlich gefasste und formal geordnete Darstellung einer Menge von Begrifflichkeiten und deren Beziehungen in einem bestimmten Bereich. Sie enthält Regeln zu Schlussfolgerungen und zur Gewährleistung ihrer Gültigkeit. Sie wird dazu genutzt, Wissen in digitalisierter und formaler Form zwischen Anwendungsprogrammen und Diensten auszutauschen. Sie steht eng mit der Idee des semantischen Webs im Zusammenhang. Im Unterschied zu einer Taxonomie, die nur eine hierarchische Untergliederung bildet, stellt eine Ontologie ein Netzwerk von Informationen mit logischen Relationen dar.

### SPARQL:

SPARQL (SPARQL Protocol And RDF Query Language) ist eine Graph-basierte Anfragesprache für RDF. SPARQL besteht dabei aus drei Teilen: SPARQL Query Language for RDF, der Anfragesprache selbst, SPARQL Query Results XML Format, wobei das Format der Anfrageergebnisse in XML festgelegt wird und SPARQL Protocol for RDF, einem Protokoll für den Austausch von Abfragen und Ergebnissen. Eine SPARQL-Anfrage beschreibt eine Schablone für RDF-Fakten, eine Art „Steckbrief“ für Teilgraphen aus dem Semantischen Netz. Innerhalb dieser Anfrage können Bedingungen für Subjekt, Prädikat und Objekt spezifiziert werden. Mit dem dadurch entstandenen Graph-Muster lassen sich ein oder mehrere Teilgraphen mit diesen Bedingungen finden. Die Ergebnisse der Anfrage sind gebundene Variablen und deren Werte. Sie könne als Liste oder in RDF ausgegeben werden.

### Ressource (URI):

(Uniform Resource Identifier Abk. URI; engl für „einheitlicher Bezeichner für Ressourcen“) URIs bestehen aus Zeichenfolgen, die zur Identifizierung einer abstrakten oder physischen Ressource dienen. Vor allem werden URIs im WWW eingesetzt dabei bezeichnen Ressourcen etwa Webseiten, sonstige Dateien, Video/Audio Daten oder auch Email Adressen. URIs sind aus folgenden fünf Teilen aufgebaut: *scheme* (Schema), *authority* (Anbieter), *path* (Pfad), *query* (Abfrage) und *fragment* (Teil) wobei *scheme* sowie *path* immer vorhanden sein muss. Mit Hilfe von URIs können Weltweit Pfade zu Ressourcen bezeichnet werden.

### RDF:

(Resource Description Framework) bezeichnet eine Formulierung logischer Aussagen über beliebige Dinge (z.b. Ressourcen). RDF wurde vom W3C als Standard zur Beschreibung von Metadaten konzipiert. Mittlerweile gilt es als grundlegender Baustein des Semantischen Web's. Im RDF Modell besteht jede Aussage aus den drei Einheiten Subjekt, Prädikat und Objekt, wobei eine Ressource als Subjekt mit einer anderen Ressource als Prädikat näher beschrieben wird. Mit einer weiteren Ressource oder nur einem Wert als Objekt bilden diese drei Einheiten ein Tripel. Die formale Semantik basiert auf einem gerichteten Graphen. Die Daten im RDF Model sind lediglich Aussagen über Ressourcen.

## **RDF – Schema:**

Zur Interpretation von in RDF formulierten Aussagen bedarf es einen gemeinsamen Vokabulars wie zum Beispiel Dublin Core (ist eine Sammlung einfacher und standardisierter Konventionen zur Beschreibung von Dokumenten und anderen Objekten im Internet). RDFS stellt ein Vokabular zur Verfügung, mit dem Eigenschaften und Relationen zwischen Ressourcen repräsentiert werden können. Man kann also mit RDFS einfache Ontologien formalisieren.

Grundlegend basiert RDFS auf einem mengentheoretischen Klassenmodells mit dem es möglich ist eine formale Beschreibung der Semantik der im RDF verwendeten Elemente festzulegen.

Neben RDFS existieren weitere Ontologie Beschreibungssprachen wie z.B. OWL

## **OWL:**

OWL (Web Ontology Language) ist eine vom W3C (World Wide Web Consortium) formalisierte Beschreibung, mit deren Hilfe Ontologien mit einer formalen Semantik erstellt werden können. Ziel ist es Begriffe zusammen mit ihren Beziehungen formal zu beschreiben, so dass Software in der Lage ist den Inhalt der Aussagen (also die Bedeutung) zu verarbeiten. Daher ist OWL ein wichtiger Bestandteil des SemanticWeb und basiert auf der RDF-Syntax, erweitert diese aber um weitere Sprachkonstrukte, um die Ausdrucksmächtigkeit zu steigern, wodurch es möglich wird Ausdrücke ähnlich der Prädikatenlogik zu erzeugen.

## **Wrapper:**

Bietet die Möglichkeit durch Einbettung eines Teils einer Software in eine andere Kompatibilität zu erreichen. Anschaulich kann man sich einen Wrapper wie einen Reisestecker vorstellen, der Geräte in einem Land mit anderen Steckdosen nutzbar macht.

Es ist als eine Konvertierung der Schnittstellen zu verstehen und kann nötig sein um einzelne Komponenten einer Software in ein neues Framework zu integrieren. Er stellt die Funktionsfähigkeit der Komponente mit den neu definierten Schnittstellen sicher.

In Ontowiki werden Wrapper genutzt um Daten Ressourcen aus Altsystemen in RDF zu konvertieren.

Ein Beispiel in einem dem Thema nahestehenden Anwendungsfall ist Triplify. Das ist ein kleines Modul, das es ermöglicht Daten, die in einer klassischen relationalen Datenbank gespeichert sind als RDF oder Linked Data zugänglich zu machen.

## **Web Application:**

Eine Web Application ist eine Software, die über einen Internetbrowser bedient wird. Das eigentliche Programm wird größtenteils auf einem Webserver ausgeführt. Die Eingaben im Browser und die angezeigten Ergebnisse werden über das Internet mit dem Protokoll http übertragen.

Die Vorteile einer solchen Anwendung sind die flexible Nutzungsmöglichkeit über jedes Gerät mit geeignetem Browser und Internetzugang und die Mehrbenutzerfähigkeit.

Zum Nachteil kann die zentrale Verwaltung auf einem Webserver werden, wenn dieser nicht angemessen gegen Ausfall geschützt ist.

Einige Beispiele sind Programme zur Verwaltung von Websides, Online-Banking Zugänge oder auch Zugänge zu Bibliotheken.

## **Framework:**

Ein Framework stellt das Grundgerüst einer komponentenbasierten Software dar.

Ein Framework stellt für sich keine funktionierende Software dar. Erst durch das Programmieren einzelner Komponenten entsteht eine lauffähige Anwendung. Definiert werden im Framework hauptsächlich der Kontrollfluss und Schnittstellen, mit denen einzelne Komponenten arbeiten.

Durch die im Framework definierten Entwurfsmuster wird die Struktur der einzelnen Komponenten beeinflusst.

Ein spezieller Typ ist das Web Application Framework. Ein solches bietet den Rahmen für die Entwicklung einer Web Application und bietet die meisten dafür notwendigen Techniken und Entwurfsmuster. Sie bieten meistens eine breite, generische, Nutzbarkeit für verschiedene Anwendungsfälle, unterscheiden sich aber in speziellen unterstützten Techniken.

### **Vokabular:**

Das Vokabular bildet eine Menge von gewählten Bezeichnungen, die die Datenstruktur in einer Ontologie beschreiben. Da aber das Semantic Web nicht eindeutig ist, dh. dass auch synonymische Begriffe auftauchen können, verwendet man z.B. im RDF (eine Ontologiesprache) so genannte URIs (Ein Uniform Resource Identifier). Sie sind eine Form von Identifikatoren, die die Ambiguität auflösen. Dank dessen ist dann die Verarbeitung (bzw. Suche in) einer solchen Struktur viel einfacher. Man versucht auch das Vokabular-Alignment zu bilden, dh. es werden best matches von zwei unterschiedlichen Vokabularen gesucht und als Ergebnis zeigt man die Relation zwischen denen, ohne ein weiteres Vokabular zu bilden.

### **Datenbank:**

Eine Datenbank ist ein elektronisches System zur dauerhaften Speicherung und Verwaltung von großen Datenmengen. Sie stellt die Daten sowohl Benutzern als auch Programmen auf Anfrage hin zur Verfügung. Hierbei gliedert sie sich in zwei Bestandteile: Ein Datenbank-Management-System (DBMS) und die eigentliche Datenmenge. DBMS verwalten die Daten und kontrollieren den Zugriff auf sie, daher ist die Auswahl eines passenden DBMS eine kritische Entscheidung.

## **Konzepte**

---

### **Zend Framework:**

Das Zend Framework ist ein Open-Source Framework für die Entwicklung von Web-Anwendungen und Services mit PHP 5. Es beinhaltet bereits Komponenten für Datenbankabfragen, Verarbeitung von Templates, Formularerstellung, Datenvalidierung und Versand von E-Mails. Jede Komponente wurde mit wenigen Abhängigkeiten zu anderen Komponenten entworfen, wodurch die Komponenten unabhängig von einander nutzbar sind.

Durch den Einsatz objektorientierter Programmierung und ausgewählter Entwurfsmuster wie MVC sind Komponenten einfach erweiterbar, so können z.B. Plug-ins und Helferklassen leicht integriert werden. Das Zend Framework wird bereits seit dem ersten Release mit einem Referenzhandbuch ausgeliefert, außerdem gibt es eine vollständige Dokumentation der Schnittstellen. Durch diese gute Dokumentation kann man sich leicht in Zend einarbeiten, was in einer geringeren Einarbeitungszeit resultiert.

Zend ist heute sehr weit verbreitet und wird von vielen Webanwendungen verwendet. Durch eine große Community, die sich um die Entwicklung von Zend kümmert, ist es heute sehr ausgereift und hat sich etabliert. Durch die hohe Verbreitung sollten auftretende Probleme schnell und einfach gelöst werden können.

### **Semantic Web:**

Semantic Web ist ein Konzept zur Weiterentwicklung/Erweiterung des Internets.

Ein Problem des World Wide Webs ist eine unvorstellbare Menge an unstrukturierten Daten, wodurch es so nicht möglich ist die Informationen auf ihre Bedeutung hin von Computern zu verarbeiten. Es besteht zwar die Möglichkeit die Daten zu durchsuchen, aber um wirklich nützliche Informationen heraus zu filtern müssen sie vorerst von Menschen gelesen und interpretiert werden. Alle Informationen müssen daher mit einer eindeutigen Beschreibung versehen werden, die von Computer „verstanden“ werden kann. Damit werden die Informationen verwertbar, interpretierbar und können automatisch weiterverarbeitet werden. Das ermöglicht es Informationen auf Basis der Inhalte miteinander in Beziehung zu setzen, wodurch z.B. die

Suche nach gewissen Informationen gezielter erfolgen kann. Grundlagen für das Semantic Web bilden Technologien wie RDF, OWL oder SPARQL.

Beispiel: Dresden<Stadt> liegt an der Elbe<Fluss>. Paul Schuster<Familiennamen> wurde 1950<Geburtsdatum> in Dresden<Geburtsort> geboren.

Bei einer Suchanfrage nach Flüssen könnte eine Suchmaschine aus einem solchen Text herausfiltern, dass „Elbe“ ein Fluss ist. Zudem ist nun auch bekannt, dass die Elbe in Beziehung zur Stadt Dresden steht.

## **Linked Data:**

Linked Data bezeichnet im Internet freie Datenvorkommen, die per URI (Uniform Resource Identifier) identifiziert sind. Diese Daten können über URIs direkt abgerufen werden und verweisen zusätzlich auf andere Daten. Zur Kodierung und Verlinkung der Daten wird das Resource Description Framework und darauf aufbauende Standards wie SPARQL und die Web Ontology Language verwendet. Linked Data ist damit Teil des semantischen Webs. Die verknüpften Daten stammen aus verschiedenen Quellen und können automatisch weiterverwendet werden. Sie ergeben ein weltweites Netz, das Linked Data Cloud oder Giant Global Graph genannt wird.

Nach Tim Berners Lee gelten folgende Regeln für *Linked Data*:

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.
4. Links auf andere *URIs*, über die weitere Objekte entdeckt werden können sind sehr wichtig.

## **OntoWiki:**

Das OntoWiki ist ein Wikisystem, in dem alle Daten als Tripel gespeichert werden, was ihre Verknüpfungen ermöglicht. Das Werkzeug ist also ein generischer Browser, der der kollaborativen Arbeit mehrerer Nutzer dient. Unterstützt werden u.A. sowohl Versionsverfolgung nach dem Editieren verschiedener Inhalte als auch Kommentare und Annotationen der User.

Das OntoWiki basiert auf dem RDF-Modell, das ein System zur Beschreibung der Ressourcen ist. Da das OntoWiki zu den OpenSource-Projekten zählt, bietet es viele Möglichkeiten der Erweiterbarkeit. Z.B. weitere Module um GUI Fragmente zu erstellen oder Plugins um die Daten zu modifizieren.

Es wird in der Skriptsprache PHP entwickelt, dazu kann eine MySQL- oder Virtuoso-Datenbank verwendet sein. Einen technischen Ansatz finden hier auch solche Servern wie Apache oder NGINX, was die schnellere Alternative zu Apache ist.

## **MySQL:**

MySQL ist ein Freeware – DBMS, das auf dem relationalen Datenbankmodell aufbaut. Aufgrund seiner leichten Erlernbarkeit, hohen Zuverlässigkeit und Schnelligkeit sowie System- und Plattformunabhängigkeit ist es sehr weit verbreitet.

Außerdem bieten zahlreiche Software-Schnittstellen die Möglichkeit zur Modifikation und Anpassung. Jedoch ist es häufig nötig zusätzliche Plug-ins von Drittanbietern zu verwenden, damit MySQL allen Anforderungen gerecht werden kann. Diese sind jedoch aufgrund der Beliebtheit von MySQL in großer Zahl vorhanden.

Diese Beliebtheit bringt noch weitere Vorteile mit sich, so ist es einfach bei Problemen fachkundige Hilfe zu finden. Allgemein stellt MySQL eine gute und universelle Möglichkeit zur Abspeicherung von Daten jeder Art dar, die durch Erweiterungen in jedes Framework eingefügt werden kann.

## **Virtuoso:**

Virtuoso ist eine universelle Serversoftware, die die Funktionalitäten verschiedener Serverkonzepte in einem System vereint. Sie bietet mehrere mögliche Datenbanksysteme zur Auswahl und bietet Möglichkeiten zur

einfachen Verknüpfung mit Anwendungsprogrammen. Virtuoso unterstützt SPARQL für die Abfrage von in Datenbanken abgelegten RDF-Daten, womit sie besonders für Anwendungen im Semantic Web interessant wird. Auch bietet sie eine eigene Sprache zur Entwicklung von Web-Anwendungen und hohe Unabhängigkeit vom Betriebssystem. Insgesamt betrachtet ist Virtuoso eine exzellente Alternative zu jeder normalen Datenbank, da sie eine Vielzahl von Funktionen und Erweiterungen von Haus aus bereitstellt. Damit verbunden ist allerdings ein hoher Recherche- und Einarbeitungsaufwand, der sich jedoch lohnen könnte.

## **Aspekte**

---

### **Einleitung:**

Das Projekt "Wissenschaftsbeziehungen im 19. Jahrhundert zwischen Deutschland und Russland auf den Gebieten Chemie, Pharmazie und Medizin" der Sächsischen Akademie der Wissenschaften zu Leipzig möchte seine Personendatenbank überarbeiten. Diese enthält bio- und bibliographischen Daten über Wissenschaftler, die sowohl in Russland als auch in Deutschland tätig waren.

Der folgende Abschnitt soll den jetzigen Zustand der Datenbank darlegen und eine andeutungsweise Vorstellung eines möglichen neuen Systems liefern. Danach werden drei Problemfelder erörtert, die die Aufgabe der Überarbeitung grob untergliedern und zum Abschluss soll anhand von ähnlichen Projekte gezeigt werden, welche Realisierungen einer solchen Datenbank schon durchgeführt wurden. Dies soll dazu dienen, schon im Vorfeld eine Vorstellung der möglichen konkreten Umsetzung zu bekommen.

### **Ist-Zustand:**

Das Projekt befindet sich in der Aufbauphase. Es wurde ein stabiles einfach strukturiertes System geschaffen in dem bereits jetzt eine Übersicht über die Personen, deren Daten einmal abgerufen werden sollen, zugänglich ist. Ca. 85 Personen sind auch bereits mit allen Daten näher beschrieben.

Realisiert ist dies durch eine einfache HTML-Webseite (Vgl. <http://drw.saw-leipzig.de/personendatenbank.html>). Alle Informationen liegen als HTML-Dokumente vor.

Die Grenzen dieser Umsetzung werden schnell deutlich und die ersten Anforderungen an ein überarbeitetes System bewusst.

1. Das Einpflegen der Daten kann nur durch eine Person geschehen, ansonsten droht die Gefahr von Kollisionen beim Bearbeiten.
2. Es wird gesondertes Personal gebraucht um die Daten, die bisher nach der Recherche als Textdatei vorliegen, in die Webseite einzutragen.
3. Die Darstellung ist nicht sehr flexibel.
  - Listen lassen sich nicht sortieren
  - Es gibt keine Suchfunktion
  - Links zu weiteren Informationen müssen aufwendig eingefügt werden
4. Neue Techniken in der Darstellung (JavaScript, HTML5) oder semantische Strukturierung sind nicht implementiert
5. Das System ist sehr aufwendig zu erweitern, da es keine Schnittstellen bietet.

### **Grobe Annäherung:**

Die Grundlage des Systems sollte eine stabile, von mehreren Benutzern leicht zu verstehende Plattform darstellen.

Anpassungen und neue Funktionen sollten leicht implementierbar sein. Da davon auszugehen ist, dass eine weitere Entwicklung auf technischer Seite Projektcharakter besitzt und von verschiedenen Personen durchgeführt werden wird, sollte man gängige Techniken benutzen. Ein ausgereiftes Framework bietet dafür die beste Lösung. Im optimalen Fall sollte das Framework schon in dieser Anwendungsdomäne erprobt sein. (→ Framework, → Web Application, → OntoWiki, → Zend Framework)

Als Grundlage der Datenhaltung muss eine zentrale Datenbank dienen, die Mehrbenutzerfähigkeit

unterstützt.

Zur Bedienung sollte ein langfristig nutzbarer plattformunabhängiger Zugang gewährleistet werden.

Um eine Verzahnung mit existierenden Wissensdatenbanken während des Softwareprojektes oder in Zukunft zu ermöglichen und für die Daten eine weitere Schnittstelle zur Verfügung zu stellen liegt es nahe die Plattform von vornherein als semantische Plattform zu planen.

(→ Ontologie, → Semantic Web, → Linked Data, → Ressource)

## **Problemfelder:**

Aus diesen ersten Überlegungen leiten sich mehrere Problemfelder ab, die die Aufgabe in voneinander abgrenzbare Bereiche gliedert:

### 1. Datengewinnung und Datenhaltung

Es müssen bestehende Daten erfasst und strukturiert werden und geeignete Techniken gefunden und implementiert werden um die Daten semantisch aufzubereiten.

(→ RDF, → RDFS, → OWL, → Wrapper, → Vokabular)

Es muss eine geeignete Datenhaltung implementiert werden, die dem Umfang und der Art der Daten gerecht wird.

(→ Datenbank, → MySQL, → Virtuoso)

### 2. Dateneingabe, Datenanreicherung und Datenverwaltung

Es muss eine geeignete, angepasste Eingabemöglichkeiten für die Eingabe von neuen Daten erstellt werden. Diese muss robust und flexibel genug sein, um die tägliche Arbeit der WissenschaftlerInnen zu ermöglichen.

Es ist möglich Techniken einzusetzen um die Daten automatisch mit anderen Datenquellen zu verknüpfen. (→ Linked Data)

### 3. Datendarstellung und Schnittstellen

Es muss eine geeignete Datendarstellung erarbeitet werden, die auf die Anforderungen wissenschaftlichen Arbeitens ausgelegt ist. Sie muss insbesondere abgestimmt sein auf mögliche wissenschaftliche Fragestellungen der Fachdomäne.

Außerdem muss eine geeignete Möglichkeit gefunden werden die Daten semantisch zugänglich zu machen. (→ SPARQL)

## **Vergleichbare Projekte:**

Im Folgenden sollen einige thematisch vergleichbar aufgestellte Projekte vorgestellt werden. Bei einigen davon wird erläuternd gezeigt, wie eine konkrete ähnliche Umsetzung unseres Projektes aussehen könnte.

### **Catalogus-Professorum.org**

Catalogus Professorum ist ein Projekt der Universität Leipzig, das zur Bearbeitung von Daten von über 1300 Professoren entstanden ist. Es beruht aber auf keiner einfachen Datenbank, sondern wurde aufbauend auf der Idee des Semantic Web erstellt. Die AKSW-Research Group aus der Universität Leipzig verwendete bei diesem Projekt Wiki-Konzepte, um den Anwendern die Erstellung bzw. Editierung von Datensätzen auf Grundlage eines einfachen Prinzips zu ermöglichen.

Wie war es möglich den komplexen Prozess zu vereinfachen? Nach der Bestimmung unterschiedlicher Vokabulare wurden die Ressourcen des Catalogus Professorum in den Tripeln mit Hilfe von RDF gespeichert, deren Verarbeitung die SPARQL-Anfragesprache ermöglicht. Um die benutzerfreundliche Sicht zu erzeugen, hat man im Catalogus Professorum eine Extension *Visual Query Builder* entwickelt, die in JavaScript implementiert ist und mit dem Triple Store korrespondiert (dank der SPARQL). Das Auffinden unterschiedlicher Relationen bereitet auch keine Probleme, dazu wird RelFinder verwendet. Alle Daten werden in der Virtuoso-Datenbank gespeichert.

### Mögliche Ähnlichkeiten zur Umsetzung unseres Projekts

In unserem Projekt haben wir im Moment mit den HTML-Daten zu tun. Wir müssen die bereits gesammelten Ressourcen in ähnlicher Weise im RDF-Format speichern, damit man sich auf das Prinzip des Semantischen

Webs beziehen kann. Dazu wäre die Benutzung von OntoWiki sinnvoll, weil es nicht nur die Idee des Semantic Web unterstützt, sondern auch ein generischer Browser der RDF-Tripel, wodurch die Arbeit mehrerer Nutzer problemlos parallel erfolgen kann. Die Anfragen wären auch mit SPARQL möglich. Zu der Auswahl der Datenbank könnten man sich überlegen, ob bei dieser (relativ kleinen) Datenmenge die leichter einsetzbare Variante eher in Frage kommt, da sie einen kleineren Einarbeitungsaufwand erfordert.

### **olgdw.de/biographisches-lexikon-der-oberlausitz**

Hierbei handelt es sich um eine Datenbank von Wissenschaftlern aus der Oberlausitz, bereitgestellt von der Oberlausitzer Gesellschaft der Wissenschaften. Die Daten werden, ähnlich wie das bekannte System Wikipedia, in Form von Artikeln gespeichert. Wichtige Informationen wie Lebensdaten, Geburtsort oder Beruf sind jedoch zu Beginn jedes Artikels gesondert und maschinenlesbar aufgeführt. Dies hat den Vorteil, dass auch automatisierte Programme auf die einzelnen Seiten zugreifen und Verbindungen zwischen Personen mit gleichem Geburtsort etc. herstellen können. Die eben erwähnten Daten haben dabei eine Form die RDF ähnelt, jedoch fehlt jede Art von Möglichkeit, Personen direkt zu verbinden, es ist also kein ontologisches System. Zusätzlich gibt es Raum für Bemerkungen, in denen jeder Autor Hinweise zu den Daten bzw. ihrer Verlässlichkeit geben kann. Um die Datenbank vor möglichem Vandalismus oder unqualifizierten Änderungen zu schützen wurde hierbei ein System verwendet, das jedem Internetanwender erlaubt in der Datenbank zu recherchieren, doch nur angemeldeten und durch einen Administrator freigeschalteten Benutzern Schreibrechte gibt.

#### Mögliche Ähnlichkeiten zur Umsetzung unseres Projekts

Das Anmeldungssystem ist sehr effektiv und sollte möglicherweise ähnlich übernommen werden. Ansonsten ist diese Datenhaltung eher ungeeignet für unser Projekt, da die Recherche erheblich mehr Aufwand kosten würde als vorgesehen.

### **www.webgenealogie.de**

Diese Website bietet eine sehr umfangreiche semantische Personendatenbank der Stammbäume sächsischer Familien über mehrere Jahrhunderte hinweg. Dabei werden jeder Person zwar nur eine sehr kleine Anzahl von Orten, Jahreszahlen und Familienmitgliedern direkt zugeordnet, über das semantische Konstrukt jedoch bilden sich sehr schnell extrem komplexe Netzwerke heraus. Dies eignet sich hervorragend für die dort betriebene Ahnenforschung. Zu den Personen selbst findet man aber nur sehr wenige Informationen, Angaben zu Berufen oder Lebensläufen fehlen.

Die Daten werden bei jeder Anfrage neu aus den eingetragenen Netzwerken zusammengefügt und in einer Dokumentenform ausgegeben, die nach der Benutzung wieder gelöscht wird.

Auch hier existiert eine Form von Anmeldungssystem, nicht registrierte Benutzer können nur grundlegende Daten wie den Namen, Geburts- und Sterbedaten auslesen. Für tiefer gehende Recherchen muss wiederum ein Konto angelegt und freigeschaltet werden, Änderungen sind nur von Administratoren durchführbar.

Die Daten wurden zunächst aus alten Dokumenten wie Steuerlisten oder Gerichtsbüchern zusammengetragen, in einheitliche Form gebracht und dann von einem Programm automatisch verarbeitet. Dies erleichterte die Erweiterung der Datenbank, da es nicht mehr nötig war, jede einzelne Information per Hand einzufügen.

#### Mögliche Ähnlichkeiten zur Umsetzung unseres Projekts

Drei wesentliche Punkte verdienen Beachtung:

Erstens sollte man möglicherweise erwägen ein ähnlich restriktives Anmeldungssystem einzuführen, um wirklich jede Art von Störung auszuschliessen. Zweitens gibt das System gute Denkanstöße für den Umgang mit unexakten Daten (z.B. nur ungefähre Jahreszahlen). Hierbei wird die Jahreszahl als Objekt angegeben und davor ein Kommentar wie „etwa“ oder „vor“ eingefügt. Und drittens ist die Ausgabe als sich während der Bearbeitung selbstständig aktualisierendes Dokument eine Erwähnung wert. Eine ähnliche Form wäre für unser Projekt in Erwägung zu ziehen.

## **Quellen:**

### Ontologie:

1. [http://de.wikipedia.org/wiki/Ontologie\\_\(Informatik\)](http://de.wikipedia.org/wiki/Ontologie_(Informatik))
2. <http://www.gi.de/service/informatiklexikon/detailansicht/article/ontologien.html>
3. [http://www.uni-protokolle.de/Lexikon/Ontologie\\_\(Informatik\).html](http://www.uni-protokolle.de/Lexikon/Ontologie_(Informatik).html)

### SPARQL:

4. <http://de.wikipedia.org/wiki/SPARQL>
5. [http://www.w3c-tag.de/2008/Tikwinski\\_SPARQL.pdf](http://www.w3c-tag.de/2008/Tikwinski_SPARQL.pdf)
6. <http://michaels-website.de/wp-content/uploads/2012/05/sparql.pdf>

### URI:

7. [http://de.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier)

### RDF:

8. [http://de.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://de.wikipedia.org/wiki/Resource_Description_Framework)
9. <http://www.w3.org/RDF/>

### RDFS:

10. <http://de.wikipedia.org/wiki/RDF-Schema>
11. [http://en.wikipedia.org/wiki/RDF\\_Schema](http://en.wikipedia.org/wiki/RDF_Schema)
12. <http://www.w3.org/TR/rdf-schema/>

### OWL:

13. <http://www.w3.org/TR/owl-features/>
14. [http://de.wikipedia.org/wiki/Web\\_Ontology\\_Language](http://de.wikipedia.org/wiki/Web_Ontology_Language)

### Wrapper:

15. [http://de.wikipedia.org/wiki/Wrapper\\_\(Software\)](http://de.wikipedia.org/wiki/Wrapper_(Software))
16. <http://semanticweb.org/wiki/Triplify>
17. <http://t3n.de/magazin/semantische-webapplikationen-entwickeln-neue-dienste-ebnen-221228/>

### Web Application:

18. <http://de.wikipedia.org/wiki/Webanwendung>

### Framework:

19. <http://de.wikipedia.org/wiki/Framework>
20. [http://de.wikipedia.org/wiki/Web\\_Application\\_Framework](http://de.wikipedia.org/wiki/Web_Application_Framework)

### Vokabular:

21. [http://de.wikipedia.org/wiki/Kontrolliertes\\_Vokabular](http://de.wikipedia.org/wiki/Kontrolliertes_Vokabular)
22. [http://en.wikipedia.org/wiki/Controlled\\_vocabulary](http://en.wikipedia.org/wiki/Controlled_vocabulary)
23. <http://www.w3.org/standards/semanticweb/ontology>

### Datenbank:

24. <http://dbs.uni-leipzig.de/file/dbs1-ws12-kap1.pdf>
25. <http://de.wikipedia.org/wiki/Datenbank>

### Zend:

26. <http://framework.zend.com/manual/1.12/de/manual.html>
27. [http://books.google.de/books?id=Tw5\\_Xe\\_MQQ8C&printsec=frontcover&hl=de&source=gbs\\_ge\\_summy\\_r&cad=0#v=onepage&q&f=false](http://books.google.de/books?id=Tw5_Xe_MQQ8C&printsec=frontcover&hl=de&source=gbs_ge_summy_r&cad=0#v=onepage&q&f=false)



Semantic Web:

28. <http://www.w3.org/standards/semanticweb/>
29. [http://de.wikipedia.org/wiki/Semantisches\\_Web](http://de.wikipedia.org/wiki/Semantisches_Web)

Linked Data:

30. <http://linkeddatabook.com/editions/1.0/>
31. [http://de.wikipedia.org/wiki/Linked\\_Open\\_Data](http://de.wikipedia.org/wiki/Linked_Open_Data)
32. <http://www.w3.org/standards/semanticweb/data>

OntoWiki:

33. [www.informatik.uni-leipzig.de/~auer/publication/WissensarbeitMitOntoWiki.pdf](http://www.informatik.uni-leipzig.de/~auer/publication/WissensarbeitMitOntoWiki.pdf)
34. <http://en.wikipedia.org/wiki/OntoWiki>
35. <http://aksw.org/Projects/OntoWiki.html>

MySQL:

36. <http://dev.mysql.com/doc/refman/5.1/de/what-is.html>
37. <http://de.wikipedia.org/wiki/Mysql>

Virtuoso:

38. <http://packages.debian.org/de/squeeze/virtuoso-opensource>
39. <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtuosoFAQ>
40. [http://en.wikipedia.org/wiki/Virtuoso\\_Universal\\_Server](http://en.wikipedia.org/wiki/Virtuoso_Universal_Server)