

# **Recherchebericht**

**SWP12-10**

Paul Röwer

Peggy Lucke

J. Nathanael Philipp

Ha Tran

Alexander Richter

Felix Kühnl

Mark Hennig

18. Januar 2012

## Inhaltsverzeichnis

<b>1</b>	<b>Begriffe</b>	<b>3</b>
1.1	URI . . . . .	3
1.2	SPARQL . . . . .	3
1.3	OpenStreetMap (OSM) . . . . .	3
1.4	LinkedGeoData (LGD) . . . . .	3
1.4.1	Knoten (node) . . . . .	3
1.4.2	Wege (way) . . . . .	4
1.4.3	Relationen (relation) . . . . .	4
1.5	ADF . . . . .	4
1.6	Ontologie . . . . .	5
1.7	Mapping . . . . .	6
1.8	RDF . . . . .	6
1.9	Ressourcen . . . . .	6
<b>2</b>	<b>Konzepte (domänenspezifisches Was und Wie – Vorgegebenes)</b>	<b>7</b>
2.1	Semantic Web . . . . .	7
2.2	Client-Server . . . . .	7
2.3	Dynamische Webseiten Programmierung . . . . .	7
2.4	Entwicklungsumgebung . . . . .	8
2.5	Ablaufplan . . . . .	8
2.6	Modellierung . . . . .	9
2.7	Versionsverwaltungssystem . . . . .	9
<b>3</b>	<b>Beschreibung der zu studierenden Applikation</b>	<b>10</b>
3.1	Weboberfläche . . . . .	10
3.2	Autovervollständigung . . . . .	10
3.3	Edit-History . . . . .	10
3.4	Nice-to-haves . . . . .	10
3.4.1	Nice-to-haves vs. Must-haves . . . . .	10
3.4.2	URIs verlinken . . . . .	11
<b>4</b>	<b>Quellen</b>	<b>12</b>

# 1 Begriffe

## 1.1 URI

Ein **U**niform **R**esource **I**dentifier ist ein Bezeichner bzw. Identifikator für Ressourcen wie Webseiten, E-Mail-Empfänger oder sonstige Dateien im Internet. Sie besteht aus fünf Teilen, wobei mindestens „scheme“, z.B. „http“ und „path“, z.B. „www.example.com“ vorhanden sein müssen. URLs und URNs sind Unterarten von URIs.[12][30]

## 1.2 SPARQL

SPARQL ist eine Anfragesprache, welche sich im Semantic Web als Standard durchgesetzt hat, um aus RDF-Dokumenten einen RDF-Graphen bzw. Teilgraphen abzuleiten. Die Syntax besitzt die „SELECT–FROM–WHERE“-Grundsyntax von SQL und ähnelt dieser auch in weiteren Aspekten sehr stark. Mit SPARQL ist es möglich, komplexere Anfragen zu stellen als es reine Textsuchmaschinen ermöglichen. Das Hauptziel dahinter ist, die Inhalte des Internets nicht nur maschinenlesbar zu machen, was sie heute sind, sondern auch maschinenverständlich.[25]

## 1.3 OpenStreetMap (OSM)

OpenStreetMap (OSM) ist ein unter der CC-BY-SA 2.0-Lizenz stehendes Projekt. Es hat zum Ziel eine freie Weltkarte zu schaffen. Dazu werden weltweit Geo-Daten gesammelt. Dabei kann jeder selbst GPS-Daten sammeln und sie bei OSM eingeben.[9]

## 1.4 LinkedGeoData (LGD)

LinkedGeoData ist ein Projekt was vorhandenen Geodaten miteinander verknüpft. Die Basis bilden dabei die Daten des OpenStreetMap-Projektes. Durch das LinkedGeoData-Projekt werden die Geodaten des OpenStreetMap-Projektes aus dem xml in RDF umgewandelt, dies dient dazu die Informationen maschinenlesbar zu machen, da das RDF Format für Geodaten einheitlich definiert ist.[5][28][4]

Es gibt 3 Arten von RDF's im LinkedGeoData-Projekt: Knoten, Wege, Relationen

Der schematische Aufbau des RDF Format dafür sieht folgendermaßen aus:

<RDFtyp+id> <typ> <Information/Wert>

### 1.4.1 Knoten (node)

Ist genau ein Punkt auf dem Erdball, welcher durch verschiedene Informationen beschrieben wird. Zum einen braucht man Informationen um diesen Punkt genau identifizieren und positionieren zu können dazu dient eine ID, weiterhin Geopgrafische Lage durch Höhen-/Breitengrad sowie die relative Position zu Normal Null. Ein Knoten kann auch noch weitere Informationen die ihm betreffen enthalten, zB. Knoten wäre Universität Leipzig, so hätte dieser Knoten einen Namen "Uni-Leipzig" und auch eine Adresse.[6]

**Beispiel:**

<RDFtyp+id> = <http://linkedgeodata.org/triplify/node123456>

<Typ> = <http://www.w3.org/2003/01/geo/wgs84\_pos#lat>

<Wert> = "12.12345678"

Dies beschreibt den Breitengrad des Knoten mit der id 123456.

**1.4.2 Wege (way)**

Ist eine geordnete Liste von Knoten und ist bestandteil von Relationen.

Es gibt 3 typen von Wegen, zum einen ein normaler Weg mit Startknoten und Endknoten (open polyline), wenn nun Startknoten und Endknoten übereinstimmen ist der Weg ein Kreis (closed Polyline) und man kann auch die Fläche innerhalb eines Kreises beschreiben (area).

Ein Weg kann weitere Informationen Enthalten die Relavant zu diesem sind, zb. In welchen Land sich dieser befindet[7]

**Beispiel:**

<RDFtyp+id> = <http://linkedgeodata.org/triplify/way123456>

<Typ> = <http://www.georss.org/georss/polygon>

<Wert> = "12.1234567 34.9876543 12.1234578 34.9876532 12.1234589 34.9876521"

Der Weg mit der ID wird durch Punkte beschrieben welche aus Höhen-/Breitengrad bestehen.

**1.4.3 Relationen (relation)**

Verknüpft mehrere Wege und/oder Knoten unter einem Bestimmten Merkmal zu einer Gruppe. Es gibt eine vielzahl von Typen von Relationen Brücken, Region, Land etc.[8]

<RDFtyp+id> <beziehung> <RDFtyp+id>

**1.5 ADF**

Ist ein Framework von Oracle, dass zum Ziel hat, die Entwicklung von Java-Anwendungen einfach, visuell, deklarativ und effizient zu gestalten. ADF basiert auf dem MVC-Prinzip[13]

**View:** Ist die Benutzeroberfläche

**Controller:** Steuerungseinheit zur Navigation in der Anwendung

**Model:** Abstraktionsschicht auf Business Service um den View und Controller die Arbeit darauf zu ermöglichen.

**Business Services:** Zugriffsschicht auf die Daten der verschiedenen Quellen und die Data Services.

**Data Services:** besteht aus:

- Data Controls: abstrahieren die Implementierungsdetails vom Business Services.
- Data Bindings: dient zur besseren Trennung von View und Controller. Exponiert die Methoden des Data Controls und Attribute in den UI-Komponenten.

## 1.6 Ontologie

Ontologie beschreibt einen Wissensbereich mit Hilfe einer standardisierten Terminologie sowie Beziehungen und Ableitungsregeln zwischen den dort definierten Begriffen. D.h. Sie stellt einen formalisierten Ausschnitt der realen Welt dar. Sie muss auch Formal sein, damit sie ein wissensbasiertes System computergestützt verarbeiten kann.[2][24][14][27]

**Zweck einer Ontologie:**

- Zusammenfügen bestehender Wissensbestände
- in Wissensbeständen suchen und diese editieren
- Neue Instanzen aus bestehenden Wissensbeständen generieren

Ontologien in der Informatik besitzen eine formale Beschreibung der Daten und Regeln über den Zusammenhang der einzelnen Daten. Die Regeln erlauben Rückschlüsse (durch logisches Folgern) aus vorhandenen Daten und helfen Widersprüche zu erkennen, sowie fehlendes Wissen aus den Daten zu ergänzen.

**Komponenten:**

- Klassen: Beschreiben eine Menge von Eigenschaften in Form von Attributen
- Relationen: Beschreibt Beziehungen zwischen Konzepten einer Domäne (eines Wissensbereiches). Meist binäre Relationen in Ontologien.
- Funktionen: Spezialfall von Relationen. Es wird jedem Element des Wertebereichs genau ein Element des Definitionsbereiches zugeordnet
- Formale Axiome: dienen zur Modellierung prädikatenlogischer Sätze, die immer wahr sind. Sie finden Verwendung, wenn das zu repräsentierende Wissen nicht durch andere Komponente darstellbar ist.
- Objekte: Sind einzelne Instanzen einer Klasse.

## 1.7 Mapping

Mapping ist ein Prozess der zwischen 2 verschiedenen Datenmodellen eine Verbindung herstellt. Die Datenmodelle können sowohl atomare Daten und auch Metadaten enthalten. Mapping funktioniert wie ein abstraktes Modell, um Beziehungen innerhalb einer bestimmten Domäne zu ermitteln. Data Mapping ist der oft erste Schritt zur Daten-Integration in einer bestimmten Domäne. Diese Datentransformation wird verwendet um die Beziehung zwischen einer ursprünglichen Quelle und einer Zielquelle, in welcher die Daten benutzt werden, zu vermitteln.[3][10]

## 1.8 RDF

Das Resource Description Framework bezeichnet eine Familie von Standards des W3C zur formalen Beschreibung von Informationen über Objekte, sogenannte Ressourcen, die durch eindeutige Bezeichner (URIs) identifiziert werden.

Es ist ursprünglich als Datenmodell zur Beschreibung von Metadaten im World Wide Web entwickelt worden. Durch eine Ausdehnung des Begriffs Ressource aus dem ursprünglichen WWW-Kontext heraus auf jegliche durch URIs identifizierbaren Entitäten wird RDF für eine Reihe von unterschiedlichen Anwendungen eingesetzt, z.B. Katalogdienste oder allgemeine Wissensrepräsentation.

Durch die formale Repräsentation in RDF sind Informationen von Maschinen auswertbar und maschinell durchsuchbar (z. B. mit der Anfragesprache SPARQL).

RDF bietet ein einfaches Datenmodell mit einer wohldefinierten formalen Semantik welches auf gerichteten Graphen basiert. Daten in RDF werden als Tripel modelliert, also als eine Sequenz aus 3 Elementen, dem sogenannten Subjekt (die Ressource), dem Prädikat (Eigenschaft) und dem Objekt (das Argument des Prädikats). Die Menge der Tripel bilden einen Graphen.

RDF Daten können zum Beispiel im XML Format gespeichert werden oder auch in speziellen Datenbanken, den sogenannten Triplestores.[31][15]

## 1.9 Ressourcen

Ressourcen können jegliche Arten von Daten sein, am bekanntesten sind URL's die zur Bezeichnung von Websites verwendet werden. Ihre einheitliche Syntax wurde vom W3C-Konsortium als Standard festgelegt.

In der Web-Architektur bezeichnet eine Ressource allgemein alles, was Identität hat (dargestellt z. B. als URL), in dem Sinne, dass es eine Quelle für Beschreibungen über sich sein kann. Dabei sind URIs die Identifikatoren für die Ressourcen.

Als Teil davon werden Informationsressourcen angesehen, welche die Eigenschaft haben, dass alle ihre wesentlichen Eigenschaften in einer Nachricht übermittelt werden können.  
[16]

## 2 Konzepte (domänenspezifisches Was und Wie – Vorgegebenes)

### 2.1 Semantic Web

Das Semantic Web ist eine Erweiterung des heutigen World Wide Webs, in dem zusätzlich zu den gespeicherten Daten Informationen über deren Bedeutung (Semantik) maschinenlesbar vorliegen.

- Das heutige Internet stellt eine enorme Menge von Daten und Informationen zu den verschiedensten Themen zur Verfügung, welche aber nicht zusammenhängend, sondern lose verstreut sind. Außerdem liegen viele Informationen in Form von natürlicher Sprache vor, die nur unzureichend von Computern verarbeitet werden kann.
- Das Semantic Web erweitert das Internet um Technologien, mit denen es möglich ist, Daten maschinenlesbar miteinander zu verknüpfen und bietet damit die Möglichkeit, auch komplexe Anfragen umfassend und exakt zu bearbeiten. Außerdem können zu bereits gefundenen Daten schnell zugehörige und daher ebenfalls relevante Informationen ermittelt werden. Aufgrund der bereits vorliegenden Daten können neue Informationen automatisch generiert werden. Das Semantic Web ebnet somit den Weg zur Überwindung der Datenisolation auf einzelnen Plattformen (“walled gardens”) und der Datenintegration aller im Web gespeicherten Informationen.

[17]

### 2.2 Client-Server

Client und Server stellen eine Beziehung zwischen zwei Rechnertypen da. Der Client tätigt Anfragen an einen bestimmten Dienst des Servers. Dieser empfängt und bearbeitet dieselben und sendet dem Client eine gewünschte Antwort zurück. Die Internetdienste, wie das World Wide Web, nutzen das Client-Server Prinzip bei jeder Aktion, bei der Daten zwischen zwei Rechnern im Internet ausgetauscht werden. Zum Beispiel die Übermittlung von Webseiten, die Bereitstellung von Datenbankzugriffen oder auch die Vermittlung von E-Mails.[29]

### 2.3 Dynamische Webseiten Programmierung

Im Gegensatz zu statischen Webseiten, deren Inhalt nur durch den Webseitenbetreiber geändert werden, werden die Inhalte dynamische Webseiten erst in dem Moment des Abrufens zusammengestellt. Dazu wird ein Programm auf dem Webserver ausgeführt, dass die Inhalte zusammensetzt. In der Regel wird dies über PHP, J2EE oder .NET erreicht.

Des weiteren ist es oft auch erforderlich auf Eingaben des Benutzers zu reagieren und Inhalte dem entsprechend anzupassen. Besonders ist dies der Fall bei einer Autovervollständigung, bei der die Änderungen „on the fly“ geschehen müssen.[18]

## 2.4 Entwicklungsumgebung

Eine Anwendungssoftware mit welcher Aufgaben der Softwareentwicklung möglich ist nennt man Entwicklungsumgebung. Diese verfügt in Allgemeinen über folgende Komponenten:

- Texteditor mit Syntaxhighlighting
- Compiler bzw. Interpreter
- Quelltextformatierungsfunktion

Darüber hinaus sind auch oft weitere hilfreiche Komponenten wie:

- Debugger
- Versionsverwaltung
- Projektmanagement
- UM-Modellierung

vorhanden. In der Regel wird nur eine Programmiersprache unterstützt, aber es gibt aber auch Anwendungen, welche mehrere unterstützen. Entwicklungsumgebungen sind geschaffen worden um:

- häufig wiederkehrende Aufgaben abzunehmen,
- einen schnellen Zugriff auf wichtige Funktionen zu bieten,
- Arbeits(zwischen)ergebnisse zu verwaltet,
- in spätere Bearbeitungsfunktionen direkt zu überführen,
- formalen Arbeiten abzunehmen.

Dies trägt zur Entlastung des Software-Entwickler bei, damit sich dieser auf seine eigentliche Aufgabe, die Softwareentwicklung/Programmierung zu konzentrieren.[19]

## 2.5 Ablaufplan

Ein Ablaufplan ist für das Projektmanagement die Dokumentation des geplanten, sachlichen und zeitlichen Ablaufes des Projektes. Zuerst werden mit Hilfe der Anforderungen im Projekt die Hauptaufgaben definiert und dann diese in Arbeitsschritten (sog. Vorgänge oder Meilensteine) geteilt, sodass die zeitliche Dauer sowie Aufwand an fachlichen Kenntnissen des einzelnen Vorgangs leichter einzuschätzen sind. Im Ablaufplan ist dann die Reihenfolge, in welcher die einzelnen Vorgänge unter Berücksichtigung der sachlogischen Zusammenhänge, der Randbedingung und des Projektziels zu erreichen sind, angeordnet. Hier werden Eingangswerte, Prozessabläufe und Ausgangswerte eines oder mehrerer Prozesse innerhalb des Systems in Diagrammform wie z.B ein Netzplan



oder ein Gantt-Diagramm graphisch dargestellt und beschrieben. Ablaufpläne helfen dem Projektmanagement wenn Probleme auftauchen, diese zu analysieren. Somit ist die Projektmanagement in der Lage zeitnah Gegenmaßnahmen oder unterstützende Maßnahmen einzuleiten. Es gibt unterschiedliche Formen, aber alle Prozessablaufpläne stellen Vorgänge, Entscheidungspunkt und die Reihenfolge der Bearbeitung dar.[26][1][11]

## 2.6 Modellierung

Ein Modell ist eine Repräsentation eines Systems von Objekten, Beziehungen und/oder Abläufen. Ein Modell vereinfacht und abstrahiert dabei im allgemeinen das repräsentierte System. Dabei kann der Begriff System entweder einen Teil der Realität (Abbild der Realität) oder ein noch nicht bestehendes Gebilde, das noch erstellt werden muss, bezeichnen. Modellierung ist ein Prozess, bei dem ein Modell zu einem bestimmten System erstellt werden. Ein Modell hilft uns, das System besser zu verstehen, zu simulieren und zu prognostizieren. Es gibt unterschiedliche Modelle in der Informatik, die verschiedenen Eigenschaften besitzen und somit in unterschiedlichen Bereichen Verwendung finden. Die wichtigsten Modelle in der Informatik sind E-R-Modell (Entity-Relationship-Modell) und UML (Unified Modeling Language). Es gibt verschiedene Programme, die das Modellieren unterstützt. Diese unterstützen jedoch nur das grafische Zeichnen der Modelle.[20][21]

## 2.7 Versionsverwaltungssystem

Bei einem Versionsverwaltungssystem handelt sich um eine Software, die Änderungen zum Beispiel an einer Datei mitprotokolliert, dabei werden sowohl Zeitstempel als auch eine Benutzerkennung gespeichert.

Die Hauptaufgaben eines Versionsverwaltungssystems sind die Protokollierung der Änderungen, die Wiederherstellung von alten Ständen einzelner Dateien, die Archivierung der einzelnen Stände eines Projektes, die Koordinierung des gemeinsamen Zugriffs von mehreren Entwicklern auf die Dateien und die gleichzeitige Entwicklung mehrerer Entwicklungszweige (engl. Branches) eines Projektes.

Man unterscheidet grundsätzlich zwei verschiedene Gruppen von Versionsverwaltungssystemen. Zum einem gibt es zentrale Systeme, diese sind nach dem Client-Server-Prinzip aufgebaut. Die Versionsgeschichte wird dabei lediglich auf dem Server gespeichert. Zum anderen gibt es verteilte Versionsverwaltungssysteme, bei dieser Art hat jeder, der an einem Projekt arbeitet, alles Lokal auf seinem Rechner, um zum Beispiel die Versionsgeschichte anzuschauen braucht er keine Verbindung zu einem Server herzustellen.

Wir benutzen in diesem Projekt Mercurial, ein verteiltes Open-Source-Versionsverwaltungssystem.[22]

## 3 Beschreibung der zu studierenden Applikation

### 3.1 Weboberfläche

Im Rahmen des Softwareprojektes soll eine graphische Oberfläche entwickelt werden, die das komfortable Editieren der OSM-LGD-Mappings des Linked GeoData-Projektes, welche in einer XML-Datei vorliegen, durch Nutzer über den Webbrowser ermöglicht. Das GUI wird voraussichtlich in PHP programmiert werden. Ein Mock-Up dieser, erstellt durch den Betreuer/Auftraggeber, liegt vor und dient als Vorlage.

### 3.2 Autovervollständigung

Benutzereingaben werden von der genutzten Anwendung sinnvoll ergänzt. Dabei muss ein Pool an Möglichkeiten zur Verfügung stehen, der mit fortschreitender Benutzereingabe stetig abnimmt. Mögliche Realisierungen einer Autovervollständigung:[23]

- Drop Down Liste
- anzeigen des möglichen Textes
- sofortiges Einsetzen des Vorlagentextes
- wiederholtes Verzweigen mit eindeutiger Vervollständigung

### 3.3 Edit-History

Bei einer Edit-History werden Änderungen, zum Beispiel an einer Datenbank, gespeichert, so dass man jederzeit Änderungen wieder zurücknehmen kann. Man kann neben den vorgenommenen Änderungen auch speichern, wer diese Änderungen vorgenommen hat, sodass man denjenigen kontaktieren kann, um über die Änderung(en) gegebenenfalls zu diskutieren.

### 3.4 Nice-to-haves

#### 3.4.1 Nice-to-haves vs. Must-haves

Als Nice-to-haves bezeichnet man Programmfunktionen, die vom Auftraggeber zwar gewünscht, für den produktiven Einsatz der Software aber nicht zwingend erforderlich sind. Im Gegensatz dazu sind Must-haves Features, die unbedingt im fertigen Produkt vorliegen müssen. Ob Nice-to-haves implementiert werden, entscheidet sich unter Umständen erst während des Projektes; droht die Zeit zu knapp zu werden, so sollten die Ressourcen auf die Entwicklung der Must-haves konzentriert werden.

- Die Einteilung der Programmfunktionen in Must-haves und Nice-to-haves muss in Zusammenarbeit mit dem Auftraggeber vorgenommen werden und erfolgt im Rahmen der Anforderungsanalyse.

*Anm: Meines Erachtens lieber bei den Begriffen einzuordnen, da der Begriff „Nice-to-have“ erklärt wurde. Was genau jetzt in unserer Anwendung ein Must-have und was ein Nice-to-have ist, können wir jetzt noch nicht sagen.*

### **3.4.2 URIs verlinken**

Die URIs, die vom GUI dargestellt werden, sollen klickbare Links darstellen. Klickt ein Benutzer auf einen solchen, so werden ihm weitere Mappings angezeigt, die diesen URI enthalten.

## 4 Quellen

- [1] Tom Alby. *Projektmanagement: Definitionen, Einführungen und Vorlagen*. 16. Jan. 2012. URL: <http://projektmanagement-definitionen.de/glossar/ablauf/>.
- [2] Marcel Bien. *Ontologie*. 16. Jan. 2012. URL: <http://www2.informatik.hu-berlin.de/~bien/ontologie.html>.
- [3] Conjecture Corporation. *wiseGEEK*. 16. Jan. 2012. URL: <http://www.wisegeek.com/what-is-data-mapping.htm>.
- [4] Kuratorium für Technik und Bauwesen in der Landwirtschaft e. V. (KTBL) Daniel Martini. *Semantische Technologien: Potenzial für die Landwirtschaft und Herausforderungen im Geodatenbereich*. 16. Jan. 2012. URL: <http://www.ktbl.de/fileadmin/PDFs/GI-Tagung/10Martini.pdf>.
- [5] Claus Stadler Dr. Sören Auer Jens Lehmann. *LinkedGeoData.org*. 16. Jan. 2012. URL: <http://linkedgeodata.org/files/paper.pdf>.
- [6] OpenStreetMap Foundation. *OpenStreetMap*. 16. Jan. 2012. URL: <http://wiki.openstreetmap.org/wiki/Node>.
- [7] OpenStreetMap Foundation. *OpenStreetMap*. 16. Jan. 2012. URL: <http://wiki.openstreetmap.org/wiki/Way>.
- [8] OpenStreetMap Foundation. *OpenStreetMap*. 16. Jan. 2012. URL: <http://wiki.openstreetmap.org/wiki/Relation>.
- [9] OpenStreetMap Foundation. *OpenStreetMap*. 16. Jan. 2012. URL: <http://www.openstreetmap.de/>.
- [10] GeekInterview.com. *Geek Interview*. 16. Jan. 2012. URL: <http://www.learn.geekinterview.com/data-warehouse/dw-basics/what-is-data-mapping.html>.
- [11] InLoox GmbH. *Ablaufplan*. 16. Jan. 2012. URL: <http://www.inloox.de/projektmanagement-glossar/themen/ablaufplan/>.
- [12] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier).
- [13] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Oracle\\_ADF](http://de.wikipedia.org/wiki/Oracle_ADF).
- [14] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Ontologie\\_\(Informatik\)](http://de.wikipedia.org/wiki/Ontologie_(Informatik)).
- [15] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://de.wikipedia.org/wiki/Resource_Description_Framework).
- [16] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <http://de.wikipedia.org/wiki/Ressource#Informatik>.
- [17] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Semantic\\_web](http://de.wikipedia.org/wiki/Semantic_web).

- [18] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: [http://de.wikipedia.org/wiki/Webseite#Dynamische\\_Webseiten](http://de.wikipedia.org/wiki/Webseite#Dynamische_Webseiten).
- [19] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <http://de.wikipedia.org/wiki/Entwicklungsumgebung>.
- [20] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <http://de.wikipedia.org/wiki/Modell#Informatik>.
- [21] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <http://de.wikipedia.org/wiki/Modellierung>.
- [22] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <https://de.wikipedia.org/wiki/Versionsverwaltung>.
- [23] Wikimedia Foundation Inc. *Wikipedia*. 16. Jan. 2012. URL: <http://de.wikipedia.org/wiki/Autovervollst%C3%A4ndigen>.
- [24] Gesellschaft für Informatik e.V. *Ontologie(n)*. 16. Jan. 2012. URL: [http://www.gi.de/no\\_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/ontologien-57.html](http://www.gi.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/ontologien-57.html).
- [25] Jean-Philippe Lang. *jQuery extended Autocompletion - SPARQL*. 16. Jan. 2012. URL: <https://www.hostedredmine.com/projects/swp10-7/wiki/SPARQL>.
- [26] Projekt Magazin. *Ablaufplan*. 16. Jan. 2012. URL: <http://www.projektmagazin.de/glossarterm/ablaufplan>.
- [27] Heiko Peter. *Ontologien*. 16. Jan. 2012. URL: [https://docs.google.com/viewer?a=v&q=cache:YOMxjMh9FWsJ:www.db-thueringen.de/servlets/DerivateServlet/Derivate-9518%3FXSL.ContextID%3D+ontologie+informatik+Typen&hl=de&gl=de&pid=bl&srcid=ADGEESiJoj08\\_klqqDv3LFDvcqhAKUsShsknmdAXus4FfKRpSLsVuz-3wAqu826r08ScCCQuzLtubhP6nXqwyDd5hKVAY814Mc-WCM7hfFHXPV0kW\\_Kb\\_5hki1sRxzTRZi558T4jmvbY&sig=AHIEtbTVVUoZgEfJNeKYbw6R3UtI8yUw3A](https://docs.google.com/viewer?a=v&q=cache:YOMxjMh9FWsJ:www.db-thueringen.de/servlets/DerivateServlet/Derivate-9518%3FXSL.ContextID%3D+ontologie+informatik+Typen&hl=de&gl=de&pid=bl&srcid=ADGEESiJoj08_klqqDv3LFDvcqhAKUsShsknmdAXus4FfKRpSLsVuz-3wAqu826r08ScCCQuzLtubhP6nXqwyDd5hKVAY814Mc-WCM7hfFHXPV0kW_Kb_5hki1sRxzTRZi558T4jmvbY&sig=AHIEtbTVVUoZgEfJNeKYbw6R3UtI8yUw3A).
- [28] Claus Stadler. *Agile Knowledge Engineering and Semantic Web (AKSW)*. 16. Jan. 2012. URL: [http://aksw.org/files/clus\\_stadler\\_\\_linkedgedata\\_\\_semantisch\\_vernetzte\\_geodaten.pdf](http://aksw.org/files/clus_stadler__linkedgedata__semantisch_vernetzte_geodaten.pdf).
- [29] e teaching.org. *Client-Server*. 16. Jan. 2012. URL: <http://www.e-teaching.org/technik/vernetzung/architektur/client-server/>.
- [30] e teaching.org. *URI/Uniform Resource Identifier*. 16. Jan. 2012. URL: <http://www.e-teaching.org/glossar/uri>.
- [31] W3C. *Resource Description Framework*. 16. Jan. 2012. URL: <http://www.w3.org/RDF/>.