Masterarbeit am Institut für Informatik der Freien Universität Berlin,

Arbeitsgruppe Software Engineering

Visualisierung von Linked Open Data mit Orts- und Zeitbezug

Hernando Sáenz-Sánchez
Matrikelnummer: 4342493
h.saenz@fu-berlin.de

Erstgutachterin: Prof. Dr. Ina Schieferdecker
Zweitgutachterin: Prof. Dr. Margarita Esponda-Argüero
Betreuer: Dr. Edzard Höfig

Berlin, den 14. Mai 2012

Zusammenfassung

Linked Open Data (LOD) ermöglicht den Zugriff auf eine große Menge von über das Internet offen verfügbare Daten . Eine gute Möglichkeit zur Analyse dieser Daten ist die Visualisierung. Nichtsdestotrotz ist die Auswahl einer richtigen Visualisierung nicht immer einfach. Eine unpassende Darstellung kann Daten verstecken und damit gehen wichtige Informationen verloren. Das Ziel der Masterarbeit ist die Untersuchung und Auswertung verschiedener Darstellungsarten von großen zeit- und ortsbezogenen LOD-Datensammlungen. Dazu werden grundlegende Konzepte der Visualisierung und der raum- und zeitbezogenen Daten beschrieben, Techniken des semantischen Webs und der Visualisierung erläutert, eine Ontologie zur Klassifizierung der Visualisierungen definiert und ein Modell zur Beschreibung der Komponenten der Visualisierung präsentiert. Anschließend werden Visualisierungen durch die Verwendung dieser Ontologie ausgewertet.

Inhaltsverzeichnis

1 Einleitung **9**
1.1 Motivation . . . 9
1.2 Problemstellung . . . 11
1.3 Aufbau der Arbeit . . . 11

2 Grundlagen **13**
2.1 Daten und Information . . . 13
2.1.1 Datentypen . . . 14
2.1.2 Skalen . . . 14
2.2 Ontologien . . . 16
2.2.1 Typen von Ontologien . . . 17
2.2.2 Komponenten von Ontologien . . . 17
2.2.3 Entwurfskriterien laut Grubber . . . 17
2.2.4 Erstellung von Ontologien . . . 18
2.3 Das Web . . . 20
2.3.1 Probleme mit der Informationssuche . . . 21
2.3.2 Semantisches Web . . . 22
2.3.3 URI . . . 23
2.3.4 RDF . . . 24
2.3.5 RDF-Schema . . . 27
2.3.6 SPARQL . . . 28
2.3.7 OWL . . . 30
2.4 Visualisierung . . . 31
2.4.1 Informationsvisualisierung . . . 32
2.4.2 Visuelle Wahrnehmung . . . 32
2.4.3 Ziel der Visualisierung . . . 34
2.4.4 Visualisierungsaufgaben und Interaktionstechniken . . . 35
2.4.5 Darstellung nach Datentyp . . . 36
2.5 Geographische Information . . . 37
2.5.1 WGS84 Geo Positioning . . . 38
2.5.2 RDF Vokabular WGS84 Geo Positioning . . . 39
2.5.3 INSPIRE-Initiative . . . 39
2.6 Open Data . . . 39
2.6.1 Ziele von Open Data . . . 41
2.6.2 Linked Open Data . . . 42
2.6.3 Grundprinzipien . . . 42

3 Stand der Technik **44**
3.1 Techniken zur Visualisierung von räumlich-zeitlichen Daten . . . 44
3.1.1 Die drei Komponenten der räumlich-zeitlichen Daten und das „pyramid“ Model . . . 44
3.1.2 Zeitleisten und Karten . . . 45
3.1.3 Zeit-Welle . . . 46
3.1.4 Hitzekarte . . . 48

3.1.5 Choroplethenkarte 48
3.2 Eine gemeinsame Sprache für Visualisierung 49
3.3 Visualisierungsmodelle 50
3.3.1 Klassische Modelle 50
3.3.2 VISO (Visualization Ontology) 54
3.3.3 Linked Data Visualization Model 56
3.4 Automatischen Zuordnung und Erzeugung 58
3.5 Anforderungen der Visualisierung von LOD 58
3.5.1 Arten von Konsumenten von LOD 59
3.5.2 Entwurfsrichtlinien 59
3.6 Werkzeuge zur Visualisierung von LOD im Browser 60
3.6.1 LOD Strukturen und Abfragen 60
3.6.2 SPARQL Ergebnisse 61

4 Ontologie zur Visualisierung von Linked Open Data mit Zeit- und Ortsbezug 62
4.1 Grafik-Ontologie 62
4.1.1 Feststellung der Domäne, Bereich und Anwendung der Ontologie 62
4.1.2 Wiederverwendung der vorhandenen Ontologie 63
4.1.3 Identifikation relevanter Begriffen 63
4.1.4 Erstellung der Konzepthierarchie 65
4.1.5 Definition der Beziehungen 66
4.1.6 Formalisierung von Klassen 66
4.1.7 Beispiel 69
4.2 GeoZeit-Ontologie 69
4.2.1 Feststellung der Domäne, des Bereichs und die Anwendung der Ontologie 70
4.2.2 Wiederverwendung der vorhandenen Ontologie 71
4.2.3 Identifikation relevanter Begriffen 71
4.2.4 Erstellung der Konzepthierarchie 72
4.2.5 Definition der Beziehungen 72
4.2.6 Formalisierung von Klassen 73

5 Entwicklung 75
5.1 Entwurf 75
5.1.1 Ein Modell zur Visualisierung von LOD 75
5.1.2 Interaktion des Benutzers 77
5.1.3 Visualisierungszweckabstraktion 79
5.2 Implementierungstechniken 80
5.2.1 Das Webframework 80
5.2.2 Die SPARQL-Abfrage 82
5.2.3 Die Javascript-APIs 83
5.2.4 Die Karten 84
5.3 Anwendungsfälle 85
5.3.1 LOD-Abfrage 85
5.3.2 Visualisierungszweckabstraktion 86

5.3.3 Grafische Darstellung von Ergebnissen 86

6 Ergebnisse **90**
6.1 Auswertung der implementierten Visualisierungen 90
6.1.1 Vorgehensweise 90
6.1.2 Karte mit Markierungen (Auswendungsfall 3) 90
6.1.3 Hitzekarte (Auswendungsfall 4) 91
6.1.4 Karte mit gruppierten Markierungen(Anwendungsfall 5) . . . 92
6.1.5 Choroplethenkarte (Anwendungsfall 8) 93
6.2 Abwägungen 97
6.3 Weiterführende Arbeit 98

Appendices **99**

A Ontologien **100**

B Implementierungsdetails **115**

Abbildungsverzeichnis

1 Informationsaustausch wird durch Daten gemacht [1] 13
2 Charakterisierung von Daten und Skalenniveau 15
3 Ebenen der Strukturierung von Informationen im Web. 20
4 Webseite einer Suchmaschine 21
5 Repräsentation von RDF durch einen Graphen 24
6 RDF vereinfachte Darstellung für die erste Aussage 25
7 RDF Darstellung für die zweite Aussage 26
8 Drei Komponenten der zeitlich-räumlichen Visualisierung 44
9 Zeitleisten 45
10 Ort-Zeit Würfel von Kraak [2] 46
11 Die Zeitrad -Technik [3] 47
12 Die Zeit-Wellen-Methode [4] 47
13 Time Wave beantwortete zeitliche Fragen [4] 47
14 Hitzekarte 48
15 Choroplethenkarte 49
16 Visualisierungszyklus [5] 49
17 The pipeline model [6] 51
18 Daten -Zustandsmodell von Chi [7] 51
19 Referenzmodell für die Visualisierung von Shneiderman [8] 53
20 Wilkinson - Aspekte eines Visualisierungssystems [9] 54
21 VISO Ontologie [10] 55
22 VISO Semantics-based Information Visualization Workflow [11] . . . 55
23 Gruppen von Komponenten von VISO 56
24 Information Visualization Data Stage Reference Model von Chi [12] . 57
25 Linked Data Visualization Model [13] 57
26 OWL Klasse Kontinuität der Daten 67
27 OWL Klasse Zweck (Purpose) 67
28 OWL Klasse Dimension und Unterklassen 68
29 OWL Klassenhierarchie von Grafik (Graphic) 69
30 OWL Ausprägung - Balkendiagramm 70
31 OWL Klassenhierarchie von Grafik (Graphic) 74
32 Interaktionsdiagramm der Komponenten 76
33 Interaktionsdiagramm des Benutzers 78
34 Abfrage GWT-EndPoint 82
35 Abfrage GWT-Jena-EndPoint 83
36 Karten mit Markierungen 91
37 Hitzekarten mit 50 Punkten 91
38 Hitzekarte 10000 Punkte ohne Gewichtung 92
39 Hitzekarte 10000 Punkte mit Gewichtung 93
40 Karte mit gruppierten Markierungen 93
41 Choroplethenkarte 94
42 Choroplethenkarte 94
43 Mögliche Gruppierung räumlich-zeitlicher Daten 95
44 OWL Klasse Form der Zeit (TimeForm) 107

45 OWL Klasse Form der Zeit (TimeScale) 108
46 OWL Klasse Dauer (Duration) 108
47 OWL Klasse Zeitdaten(TimeData) 108
48 OWL Klasse Geodaten (Geodata) und Unterklassen 109
49 OWL Klasse Ort(place) 109

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 14. Mai 2013

Hernando Sáenz-Sánchez

1 Einleitung

1.1 Motivation

Vor nicht so vielen Jahren war es nahezu unmöglich, sich viele der Dinge vorzustellen, die wir heute haben und die ein Teil unseres täglichen Lebens geworden sind. Die neuen Entwicklungen haben die Art und Weise, wie wir leben und kommunizieren, geändert. Es ist praktisch unmöglich, erwachsene Menschen zu finden, die nicht einen Computer, ein Mobiltelefon oder ein E-Mail-Konto besitzen und täglich das Web nutzen, um Information zu recherchieren, zu chatten oder einfach zu navigieren.

Das World Wide Web (Abk. Web) könnte als ein großes Informationssystem betrachtet werden, das sich kontinuierlich, ungeordnet und sehr schnell entwickelt und die Verfügbarkeit von Daten enorm steigert. Deshalb sind die Extraktion und die Informationsverwaltung immer noch ein aktuelles Problem des Internets.

Im Prinzip ist das Web eine riesige Menge von Seiten mit Texten, Videos, Bildern, usw., die durch Links miteinander verknüpft sein können. Es gibt keine definierten Regeln für die Veröffentlichung von Informationen. Deswegen kann jeder einen Inhalt veröffentlichen. Diese Vereinfachung ist positiv, ein Schlüssel zum Erfolg und macht das Web frei, birgt jedoch auch die Schwierigkeit zu erfahren, ob die Informationen zuverlässig, aktuell und wahrheitsgemäß sind oder nicht.

Wegen der unterschiedlichen Formate jeder Webseite können die Menschen und auch die Maschinen nicht immer einfach die wichtige Information in einer Webseite finden. Lösungen wie die Suchmaschinen sollten dabei helfen, aber es ist nicht immer möglich, weil die Suchkriterien nicht mit den Stichworten einer Seite übereinstimmen, diese Stichworte mehrere Bedeutungen haben oder ein Synonym verwendet wurde (In Kapitel zwei wird diese Problematik im Detail beschrieben).

Im Jahr 2001 wurde von Tim-Berners-Lee [14] eine mögliche Lösung für dieses ungeordnete Wachstum des Webs vorgeschlagen. Der Vorschlag war das sog. semantische Web (Abschnitt 2.3.2), das die Weiterentwicklung des Internets und den Mechanismus der Extraktion von Informationen verbessern sollte. Diese Änderungen sollten außerdem das Suchen erleichtern und eine Zuordnung ermöglichen. Die Grundidee ist, die Struktur der Daten der Webseiten zu formalisieren, sodass die Inhalte nicht nur maschinenlesbar, sondern auch maschinenverstehbar sind. Dank dieser Formalisierung wird es möglich sein, dass neue Anwendungen automatisch kombiniert und genutzt werden können, um Informationen herauszufinden (In Kapitel zwei wird das semantische Web im Detail beschrieben).

> „The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.“[14]

Die Implementierung des semantischen Webs hat in den letzten Jahren neue Dienste

und Ansätze zur Veröffentlichung von Daten im Web gezeigt. Ein gutes Beispiel dafür ist Linked Open Data (LOD) (Abschnitt 2.6.2), die eine Änderung in der Art ist, wie die Daten im Internet veröffentlicht werden.

Im Gegensatz zum normalen Web, in welchem man durch Dokumente navigieren muss, schlägt LOD vor, durch die Daten zu navigieren. Dazu werden bereits existierende Techniken verwendet, damit die Daten untereinander verbunden werden können. (Ein großer Unterschied zu den traditionellen Hyperlinks ist, dass die Verbindungen nicht mehr fest sind. Sie werden auf Anfrage generiert).

In dem Kontext von LOD kann man von zwei verschiedenen Gruppen sprechen, die mit den Daten zu tun haben. Einerseits stehen die Leute, die die Daten produzieren, die Produzenten, anderseits die Leute, die die Daten nutzen, die Konsumenten.

Für diese zweite Gruppe, die Konsumenten, sollten theoretisch die Implementierungen des semantischen Webs die Suche nach Informationen vereinfachen. Aber es ist leider nicht immer so, weil das neue semantische Web einige technische Kenntnisse erfordert, um die semantische Information durchsuchen zu können. (RDF Abschnitt 2.3.4, Ontologie Abschnitt 2.2, SPARQL Abschnitt 2.3.6, OWL Abschnitt 2.3.7).

Deswegen ist es wichtig und jeden Tag notwendiger, den Konsumenten ohne technische Kenntnisse Möglichkeiten anzubieten, die ihm erlauben, die Daten auf einfache Art und Weise zu verwenden. Das semantische Web soll den normalen Benutzer ansprechen.

Diverses Werkzeug wurde vorgeschlagen, damit die normalen Benutzer vom semantischen Web profitieren können. Die meisten davon versuchen in einem Browser durch graphenbasierte Visualisierungen die Ontologien und RDF-Tripeln darzustellen, aber diese Art von Darstellungen sind beschränkt [11] und verwenden nicht die volle Kapazität der Visualisierung von Daten (generische Diagramme, mehrere Ansichten, Verfeinerung oder die Empfehlung von geeigneten Visualisierungen).

Die Bedeutung (Wichtigkeit) der Visualisierung liegt darin, wie die Information vom Menschen wahrgenommen wird in Abhängigkeit davon, wie die Information auf die eine oder andere Art und Weise dargestellt wird. Je nach Darstellungsart der Daten wird es einfacher oder schwieriger, die Informationen zu vergleichen, auszuwerten, zu bewerten oder zu verstehen. Eine gute Visualisierung kann helfen, die Analyse der Daten zu vereinfachen. Deshalb ist es wichtig, herauszufinden, welche Möglichkeiten es gibt, Daten darzustellen und so die beste Option auszuwählen.

Diese Masterarbeit konzentriert sich auf die Datenvisualisierung. Insbesondere werden die bestehenden Mechanismen zur Visualisierung von Daten mit Zeit- und Ortsbezug untersucht, um eine Klassifizierung vorzuschlagen, die dazu dient, einen Zusammenhang zwischen LOD und Visualisierungen zu finden.

Die Thematik der Visualisierung von Daten ist jedoch kein einfaches Thema. Es gibt viele Disziplinen mit unterschiedlichen Ansätzen, die sich bemüht haben zu erforschen, welche grafische Darstellung am besten zu einer bestimmten Art von Information passt.

In der Informatik gibt es viele verschiedene API zur Generierung dynamischer Bilder, die die Programmierer verwenden können. Es werden einfach Parameter geschickt, und danach wird ein Bild zurückgegeben. Es ist jedoch notwendig, über technische Kenntnisse von Programmierung zu verfügen, was die Verwendung nicht leicht macht. Diese Problematik könnte gelöst werden, wenn die API zur Generierung von Bilder oder Karten integriert würden, mit denen die LOD Abfrage erfolgt. So könnte ein Forscher ohne Programmierungskenntnisse beispielsweise seine Daten graphisch darstellen.

Diese Integration ist nicht immer einfach, weil die Daten sehr unterschiedlich sind. Die Verwendung eines Balkendiagramms ist beispielsweise nicht empfehlenswert, wenn die Anzahl von Daten, die dargestellt wird, mehr als 25 beträgt. Das bewirkt, dass das Auswahlverfahren einige Bedingungen erfüllen muss, damit die Bedeutung der grafischen Darstellung nicht verloren geht.

1.2 Problemstellung

Es gibt viele verschiedene Arten der Visualisierung von Daten. Ein Zusammenhang zwischen einer visuellen Repräsentation und deren Daten ist nicht immer einfach und deutlich. Deswegen ist es wichtig, eine mögliche Beziehung zu finden. Ein unpassendes Bild kann wichtige Daten verstecken, und damit gehen wichtige Informationen verloren.

Eine richtige Auswahl der Visualisierung soll die zusammenhängende Darstellung zwischen komplexen Datensätzen und deren visuellen Artefakten (Grafiken, Bilder) ermöglichen. Aber diese Auswahl ist nicht immer einfach. Man muss diverse Faktoren, abhängig von den Datentypen, berücksichtigen, um das Treffen der Entscheidung zu vereinfachen. Um diese Auswahl zu erleichtern, ist eine Klassifizierung von Visualisierungen notwendig, die ermöglicht, eine geeignete Visualisierungsart nach den verfügbaren Daten zu finden.

Das Ziel der Masterarbeit ist die Untersuchung und Auswertung verschiedenen Darstellungsarten von größeren RDF-Datensätzen (Linked Open Data) mit Zeit- und Ortsbezug. Dazu ist es nötig, entscheidende Kriterien (Eigenschaften, Einschränkungen) für die Auswertung einer Darstellung herauszufinden. Diese Kriterien sollen ermöglichen, Datentypen und ihre entsprechende, mögliche grafische Darstellung zuzuordnen. Es wird angenommen, dass die RDF-Datensätze ein Datum und geografische Koordinaten besitzen. Die Auswertung beschränkt sich auf die Untersuchung von implementierbaren Visualisierungen.

1.3 Aufbau der Arbeit

Im folgenden Abschnitt werden die Begriffe Semantik Web (Abschnitt 2.3.2) Linked Open Data (Abschnitt 2.6.2) und Web Ontologie (Abschnitt 2.2) erläutert und ihre Bedeutung für diese Arbeit erklärt. Auch andere notwendige Grundlagen werden in diesem Kapitel geschildert.

In Kapitel 3 wird der Stand der Technik in verschiedenen Bereichen der Visualisierung beschrieben. Am Anfang werden zwei Methoden zur Visualisierung von räumlich-zeitlichen Daten erläutert. Danach werden diverse Modelle zum Verständnis des Visualisierungsprozesses präsentiert. Es werden außerdem wichtige Anforderungen für die Visualisierung von LOD und bestehende Lösungen kurz beschrieben.

Kapitel 4 beschreibt die Lösung der Problemstellung, dazu wird eine Ontologie vorgeschlagen, die hilft, die möglichen Visualisierungen zu klassifizieren. Die Ontologie wird nach der beschreibenden Methodologie im Abschnitt 2.2.4 erstellt und zur Formalisierung wurde RDF-S/OWL als Ontologie-Sprache verwendet. Diese Methodologie besteht aus folgenden sechs Schritten:

- Festlegung der Domäne und des Bereichs.
- Analyse der vorhandenen Ontologien für die mögliche Wiederverwendung.
- Identifikation relevanter Begriffe.
- Erstellung der Konzepthierarchie.
- Definition von Beziehungen.
- Formalisierung von Klassen.

In Kapitel 5 geht es um die Implementierung im Browser von Komponenten, um die grafische Darstellung mit den Kriterien, die durch die Ontologie modelliert wurden, auswerten und klassifizieren zu können. Die Komponenten werden im Abschnitt 5.1.1 mithilfe eines Modells beschrieben. Die Implementierung der Komponenten ermöglicht, die grafische Darstellung von Daten mit Zeit- und Ortsbezug auszuwerten. Es werden außerdem kurz die notwendigen Techniken und deren Einsatz in der Entwicklung beschrieben.

Im Kapitel 6 werden die Grafiken ausgewertet sowie andere Ergebnisse beschrieben.

2 Grundlagen

2.1 Daten und Information

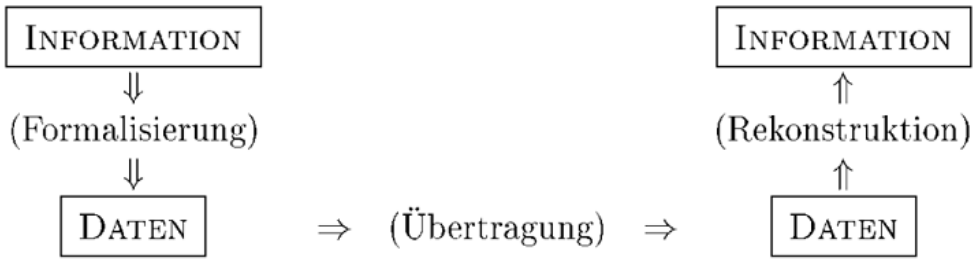
Information und Daten sind ein grundlegender Teil des aktuellen Lebens, und manchmal werden sie als synonym verwendet. Deswegen ist es wichtig, beide Konzepte im Bereich der Informatik zu definieren, damit es beispielsweise möglich wird, Konzepte wie Generierung von Information und Generierung von Daten zu unterscheiden.

Daten sind eine formale und vordefinierte Darstellung von Informationen, die normalerweise durch Zeichnen kodiert wird, einer Syntax folgt und eine Kommunikation und Interpretation erlaubt. Beispiele für Daten sind die gesprochene oder geschriebene Sprache. Nach der Struktur der Daten werden in der Regel drei Typen von Daten differenziert: strukturierte, semistrukturierte und unstrukturierte Daten.

Die Daten selbst sind keine Informationen. Die Verarbeitung von Daten ist eigentlich die Quelle, woher die Information kommt. Information ist das, was man kommunizieren möchte (der Inhalt). Der Wert der Information ist infolgedessen abhängig von der Datenqualität[1], da die Qualität direkt mit dem Prozess der Transformation der Daten zu tun hat. Je besser die Daten sind, desto einfacher, sinnvoller, möglicher, wichtiger, gültiger und zuverlässiger ist die Verarbeitung.

Es gibt zwei wichtigen Teilprozesse in der Transformation von Daten, wenn man über Informationsaustausch spricht: die Formalisierung und die Rekonstruktion. Der erste erfolgt am Anfang zur Formalisierung der Information, der zweite nach der Übertragung der Daten, damit die Information rekonstruiert werden kann [1].

Abbildung 1: Informationsaustausch wird durch Daten gemacht [1]



Es ist notwendig, die Daten zu speichern, damit die Information persistieren kann. Dies wurde im Laufe der Zeit auf verschiedene Arten durchgeführt. Dank der Existenz von Computern, magnetischen Speichermedien und Internet, können heutzutage die Daten digital gespeichert werden. Diese Digitalisierung der Information und die aktuellen Netzwerke haben die sofortige Verfügbarkeit großer

[1] „Datenqualität (im erweiterten Sinn oft auch Informationsqualität) bezieht sich auf die Relevanz und Korrektheit von Informationen und beschreibt, wie gut diese geeignet sind, die Realität abzubilden und in einem bestimmten Kontext einen Nutzen zu stiften „ [15]

Mengen von Daten ermöglicht. Das wichtigste Beispiel dafür ist das World Wide Web.

2.1.1 Datentypen

Nach der Definition der Begriffe Daten und Information, wird beschrieben, wie die Daten, abhängig von der Information, die diese beinhalten, zugeordnet werden können.

Im Prinzip lassen sich die Daten auf zwei verschiedene Arten, abhängig vom Informationsgehalts ihrer Merkmale, unterscheiden [16]:

- **Qualitative Daten:** Sie nutzen beschreibende Werte, um Information zu messen oder klassifizieren, und lassen sich nicht zahlenmäßig erfassen. Beispiele sind Wohnorte, Namen oder Geschlecht.
- **Quantitative Daten:** Dies sind normalerweise Zahlenwerte (numerische Werte), die verwendet werden, um etwas zu messen, wägen oder zählen. Es ist möglich, mit diesen Werten zu rechnen. Beispiele sind Alter, Gewicht oder Große.

2.1.2 Skalen

Diese Unterteilung ist eigentlich sehr grob, deswegen wird in der Regel das sogenannte Skalenniveau verwendet, das die Daten nach ihren Typen klassifiziert. Man unterscheidet folgende Skalen [16]:

- **Nominalskala:** Die nominal skalierten Werte können nicht geordnet werden. Ein einfaches Beispiel sind Farben (grün, blau, rot, ...) oder Blutgruppen. Es ist möglich, die Werte zu vergleichen, um sie einer Kategorie zuzuordnen. Beispiele in dem Bereich der Geoinformation[2] sind Wohnorte, Postleitzahlen oder Länder.

$$Berlin \neq Bayern$$

- **Ordinal- oder Rangskala:** Die Werte können geordnet werden. Ein Beispiel ist das deutsche Schulnotensystem, eins ist besser als zwei, zwei ist besser als drei, aber der Unterschied zwischen den Noten ist nicht gleich und steht in keinem Verhältnis.

$$Ungengend < Befriedigend < Gut < SehrGut$$

  Ein weiteres Beispiel ist die Bewertung der Produkte, die Seiten wie Amazon oder Ebay anwenden.

[2] Geoinformation wird in Abteilung 2.5 erläutert

- **Metrische Skala:** Neben der Rangordnung ist es auch möglich, die Abstände zwischen den Werten zu messen und zu interpretieren.
  - **Intervallskala:** Differenzbildungen zwischen den Werten sind zulässig, d.h. die Ordnung und die Größe des Abstandes zwischen zwei Werten sind festgelegt. Ein Beispiel dafür sind Jahreszahlen (Multiplikation oder Division hingegen sind nicht sinnvoll). Beispiele im Bereich der Geoinformation sind Meereshöhe, Jahreszahl oder Temperatur.

    $$Temperaturnderung = TemperaturvoreinerStunde - aktuelleTemperatur$$

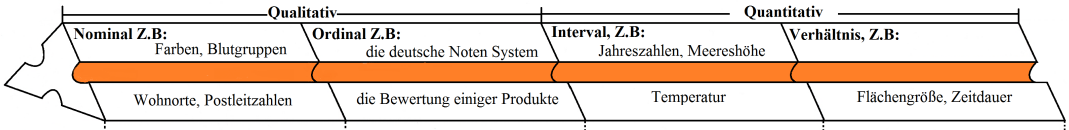
  - **Verhältnisskala(auch Ratioskala)**: Die Werte lassen sich auf Identität und Größe vergleichen, subtrahieren, multiplizieren und dividieren. Beispiele im Bereich der Geoinformation sind Flächengrößen in

    $$m^2$$

    oder Zeitdauer. Eine mögliche mathematische Operation mit den Daten wäre

    $$Durchschnittsnote = (Note1 * Note2)/2$$

Die folgende Abbildung zeigt das Verhältnis zwischen den zwei oben genannten Unterteilungen (Skalenniveau sowie qualitative und quantitative Daten).



Ordnung relevant	nein	ja	ja	ja
Abstand relevant	nein	nein	ja	ja
=/±	ja	ja	ja	ja
</>	nein	ja	ja	ja
+/-	nein	nein	ja	ja
*/÷	nein	nein	nein	ja

Abbildung 2: Charakterisierung von Daten und Skalenniveau

Die Daten werden häufig auch, abhängig von der Kontinuität der möglichen Werte der Daten, in zwei Gruppen unterteilt. Diese sind diskrete und stetige Daten.

- **Diskrete Daten:** Nur abzählbare Werte sind möglich, d.h. die Werte machen nur Sinn, wenn sie ganze Zahlen sind, wie z.B. Anzahl von Kindern.
- **Stetige Daten:** Es gibt beliebig viele Zwischenwerte.

Wenn über Datenorganisation gesprochen wird, werden bestimmte Begriffe benutzt, um die minimalen Einheiten der Daten oder Typen zu definieren:

- **Datenfeld (Datenelement):** ist die minimale Einheit von Daten. Es besteht aus einem oder mehreren Zeichen. Beispiele sind Matrikelnummer oder Staatsangehörigkeit.
- **Datensatz:** Datensätze sind inhaltlich zusammenhängende Datenfelder. Beispiel für einen Datensatz eines Artikels wäre Artikelnummer, Artikelname, Warengruppe und Artikelpreis.

Abhängig von den Datensatztypen werden die Daten auch geordnet, z.B. es ist möglich zwischen Daten mit und ohne räumlichen Bezug zu differenzieren. Folgendes sind Beispiele für diese Unterteilung:

- Ohne räumlichen Bezug: Tabelle, Graphen, Bäume, Text
- Mit räumlichen Bezug: punktbasiert, zellbasiert.

Viele andere Unterteilungen sind möglich und können je nach Kriterien ausgewählt werden.

2.2 Ontologien

Zur Formalisierung der Information wird der Begriff Ontologie verwendet, dessen Bedeutung anders ist als in der Philosophie (Studie der vorhandenen Dinge und ihre Beziehungen). Die Ontologien sind von großem Wert für die Informatik und die modernen Anwendungen. Entitäten werden in einer formale Art durch die Ontologien beschrieben, sowie ihre Beziehungen, Eigenschaften, Einschränkungen und ihr Verhalten. Diese formale Darstellung einer Entität erlaubt die Verarbeitung der Information durch Maschinen, was ein Vorteil für das semantische Web ist.

Es gibt nicht eine einzige Definition von Ontologien in der Informatik. Folgende Definitionen sollen einen Überblick über diesen Begriff geben.

> „An ontology is an explicit specification of a conceptualization[3]“[17]

> „An ontology is a formal specification of a shared conceptualization.“ [18]

> „Ontologies such as this one, in effect, lay the ground rules for modeling a domain by defining the basic terms and relations that make up the vocabulary of this topic area. These ground rules serve to guide system builders in fleshing out knowledge bases, building services that operate on knowledge bases, and combining knowledge bases and services to create larger systems. For one system to make use of either the knowledge or reasoners of another system, the two must have consistent ontologies. “[19]

[3] Die Konzeptualisierung ist hier eine abstrakte und einfache Darstellung einer Entität. Die Berücksichtigung der Objekte, der Begriffe und deren Beziehungen untereinander ist wichtig für diese Darstellung.

Angesichts dieser Definitionen kann eine Ontologie als ein abstraktes Modell definiert werden, das ermöglicht, Objekte einer bestimmten Domäne darzustellen. Das Modell soll die Objekte, die zugehörigen Begriffe und deren Beziehungen untereinander berücksichtigen.

2.2.1 Typen von Ontologien

Die Ontologien werden in zwei Übergruppen eingeteilt. Der erste Typ der Ontologie ist abhängig von der Art des Wissens, das sie darstellt. Diese Gruppe hat zwei Untergruppen, allgemeine Ontologien und Domänen-Ontologien [20]:

- Die **allgemeinen Ontologien** beschreiben grundlegende Konzepte der Welt.
- Die **Domänen-Ontologien** beschreiben spezielle Wissensdomänen.

Der zweite Typ der Ontologie ist abhängig von der Rolle, die sie in einem System spielt. Diese Gruppe hat auch zwei Untergruppen, Anwendungs-Ontologien und Aufgaben-Ontologien [20]:

- Die **Anwendungs-Ontologien** beschreiben spezielle Aspekte der Anwendung.
- Die **Aufgaben-Ontologien** beschreiben eine angegebene Aufgabestellung

2.2.2 Komponenten von Ontologien

Die Ontologien bestehen aus folgenden Komponenten:

- **Begriffe oder Klasse:** Darstellung von Objekten aus der Realität.
- **Beziehungen:** Interaktionen zwischen Begriffen, die zu einer bestimmten Domäne gehören, sind besondere Eigenschaften, deren Werte Objekte anderer Klassen sind. Vererbung ist eine wichtigste Beziehung zwischen Konzepten.
- **Eigenschaften:** charakterisieren Begriffe. Sie dienen dazu, Begriffe zu beschreiben und zu unterscheiden.
- **Beschränkungen:** bestimmen Werte für Eigenschaften und Beziehungen.
- **Kategoriensystemen:** sind Objekte und anhand charakteristischer Eigenschaften zu klassifizieren.
- **Ausprägung:** Verwirklichung einer Klasse.
- **Subsumption:** Logische Implikation zwischen zwei Konzeptdefinitionen

2.2.3 Entwurfskriterien laut Grubber

Zur Erstellung einer Ontologie sind Entscheidungen über deren Entwurf zu treffen. Zur Vereinfachung der Entscheidung wurden von Gruber [17] fünf Kriterien

beschrieben:

- **Klarheit:** Eine Ontologie sollte eindeutig ihre Bedeutung beschreiben und alle Definitionen sollten mit natürlicher Sprache dokumentiert werden.
- **Kohärenz** Es soll Kohärenz zwischen Definitionen und Inferenzen bestehen
- **Erweiterbarkeit** Es soll möglich sein, neue Begriffe aus alten zu definieren.
- **„Minimal encoding bias“** Die Beschreibung soll unabhängig von der Syntax sein.
- **Minimale Einschränkungen:** Eine Ontologie sollte so wenig Einschränkungen wie möglich haben, um die Sachen (Anm.: welche Sachen?) zu modellieren.

2.2.4 Erstellung von Ontologien

Zur Erstellung von Ontologien gibt es keine Standard-Vorgehensweisen, aber die Erstellung ist ein zyklischer Prozess, der aus folgenden Schritten besteht [21]:

1. **Die Domäne, der Bereich und die Anwendung der Ontologie werden festgelegt.** Um diese Fokussierung zu erstellen, wird die sogenannte Competency Question (relevante Fragen) verwendet, um das System zu überprüfen. Es wird vermutet, dass die Ontologie diese relevanten und prüfenden Fragen beantworten soll.

   Einige grundlegende Fragen sind:

   - Welche Domäne deckt die Ontologie ab?
   - Welche Anwendung hat die Ontologie?
   - Welche Fragen müsste die Ontologie beantworten?
   - Wer wird die Ontologie handhaben?

   Für eine Ontologie zur Visualisierung von Daten könnten folgende Fragen gestellt werden:

   - Wie müssten die Daten dargestellt werden?
   - Welchen Unterschied gibt es zwischen einem Punktdiagramm und einem Balkendiagramm?
   - Wie viel Merkmale können dargestellt werden?

   „Eine Liste solcher Fragen bildet den Ausgangspunkt für die Erstellung einer Ontologie. Im Laufe der Entwicklung können Fragen hinzukommen, aber auch Fragen entfernt werden, die sich als nicht relevant oder in der gegebene Sprache als nicht umsetzbar herausstellen“[21]

2. **Wiederverwendung der vorhandenen Ontologien.** In bestimmten Fällen können vorhandene Ontologien wiederverwendet werden. Nicht nur die

gesamte Ontologie, sondern auch die Teile einer Ontologie können nützlich sein. Einerseits hat die Wiederverwendung Vorteile wie die folgende:

- Größe Reduzierung der Aufwand der Erstellung.
- Vermeidung von Fehlern bei der Modellierung.
- Einfacher Austausch der Information zwischen Anwendungen aufgrund der Einheitlichkeit und Interoperabilität der Beschreibungen.

Andererseits gibt es Probleme bei der Wiederverwendung der Ontologien, wie zum Beispiel:

- Die Erweiterung und die Anpassung der Ontologien sind manchmal schwieriger als eine neue Ontologie zu erstellen.

3. **Identifikation relevanter Begriffe.** Die Begriffe haben eine Beziehung zu der Domäne, die dargestellt wird. Es gibt verschiedene Quellen zur Identifizierung relevanter Begriffe. Beispiele sind Interviews mit Experten, Dokumente oder Datenbanken. Die Begriffe können auch aus Compentency Questions oder der Analyse relevanter Dokumente stammen. Für die Analyse sind Substantive, Verben und Adjektive wichtig.

   - Die Substantive könnten relevante Konzepte beschreiben.
   - Die Verben können Beziehungen zwischen Konzepten ausdrücken.
   - Die Adjektive können auf mögliche Werte der Eigenschaften aufmerksam machen.

   Für die Ontologie zur Visualisierung von Daten können folgende Begriffe eingesetzt werden:

   - Position, kartesianische Achse, geografische Koordinaten, Längengrad, Breitengrad
   - Balkendiagramm, Punktediagramm, Liniendiagramm, Säulendiagramm
   - Lage, Daten, Statistik, Tendenz

4. **Erstellung der Konzepthierarchie.** Welche Begriffe stellen Klassen von Objekten dar? Welche Beziehungen gibt es zwischen Klassen? Diese Fragen müssen beantworten werden. Es ist wichtig, die verschiedenen Begriffe zusammenzustellen, da die Eigenschaften der Begriffe eine wichtige Rolle spielen. Kategoriensysteme sollten erstellt werden.

5. **Definition von Beziehungen**. Zur Identifizierung der Beziehungen ist es ratsam, folgende Fragen zu beantworten [21]:

   - Welche Eigenschaften haben die Objekte einer Klasse gemeinsam?
   - Welche Eigenschaften unterscheiden die Objekte einer Klasse von denen anderer Klassen?
   - Welche Eigenschaften unterscheiden die Objekte einer Klasse?

Es gibt zwei Typen von Eigenschaften, die Intrinsischen Eigenschaften und die Extrinsischen Eigenschaften:

- Die Intrinsischen Eigenschaften sind untrennbar mit einem Objekt verbunden und definieren dieses.
- Die Extrinsischen Eigenschaften werden dem Objekt verliehen und können sich mit der Zeit ändern.

6. **Formalisierung von Klassen.** Die Klasse muss Konsistenz sein. Das heißt, die Klasse und die Oberklasse müssen kompatibel sein.

2.3 Das Web

Das Web ist im Prinzip eine riesige Menge von Seiten mit Texten, Videos, Bildern, usw., die durch Links miteinander verknüpft sein können. Das Web wie andere Wichtige Dienstleistungen (Mail, Telnet, VoIP, usw.) funktioniert über Internet. Es wurde im Jahr 1990 für (Anm.: von?) Tim Berners-Lee mit Hilfe von Robert Cailliau offiziell vorgeschlagen [22] und basiert auf drei Komponenten: HTML (Hyper Text Markup Language), HTTP (Hyper Text Transfer Protocol) und URL (Uniform Resource Locators).

Die Menge von Information, die im Web existiert, ist riesig und vielleicht unvorstellbar, aber unabhängig von Größe und Verwendung wird über Arten von Informationen im Web gesprochen. Es werden normalerweise drei verfügbare Ebenen definiert: informal, semiformal und formal [23].

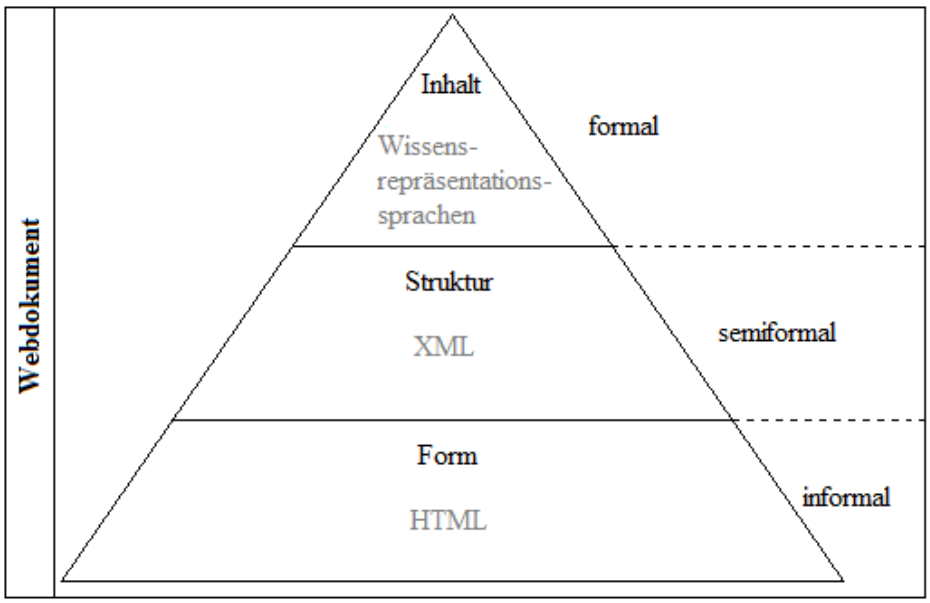


Abbildung 3: Ebenen der Strukturierung von Informationen im Web.

Auf der ersten Ebene wird die Information informal präsentiert. Es hat nur mit der Darstellung zu tun. Es gibt keine definierte Regel für die Veröffentlichung von

Abbildung 4: Webseite einer Suchmaschine

Informationen oder das Format. Deswegen kann jeder seine eigene Seite entwerfen, seine Kreativität zeigen und seine Meinung ausdrücken. Das Web hat außerdem viele weitere gute Eigenschaften wie zum Beispiel die Verfügbarkeit, die Aktualität oder die Vielfältigkeit der Information. Aber leider können die Menschen und auch die Maschinen wegen der unterschiedlichen Formate jeder Webseite nicht immer auf einfachem Weg die wirklich wichtige Information in einer Webseite finden. Noch problematischer ist es, wenn man beispielsweise wissen möchte, ob die Information zuverlässig, aktuell, richtig oder wahrheitsgemäß ist oder nicht. Es gibt heutzutage keinen Mechanismus, um die Quellen einer Webseite zu überprüfen.

Die semiformale Ebene wird mit XML beschrieben und ermöglicht, Daten auszutauschen. In der Regel werden DTDs[4] und XML-Schemata[5] benutzt, um festzulegen, was nur auf der syntaktischen Ebene ausgedrückt werden kann [23].

In der dritten Ebene befinden sich die Wissensrepräsentationssprachen [23], die später (Abschnitt 2.3.2) beschrieben werden.

2.3.1 Probleme mit der Informationssuche

Normalerweise wird eine Suchmaschine verwendet, wenn man Informationen sucht. Das Verfahren ist sehr einfach. Man schreibt Stichworte in ein Feld und schickt die Anfrage an die Suchmaschine. Wenn man die richtigen Stichwörter geschrieben hat, erhält man eine Liste von Ergebnissen, die theoretisch mit den Suchkriterien

[4] DTD ist ein Satz an Regeln, um Dokumente eines Typs zu deklarieren [24].
[5] „Ein XML-Schema ist ein maschinenlesbarer Bauplan für XML-Dokumente. “ [25]

übereinstimmen sollten. Die Suche hat jedoch keinen Erfolg, wenn die Suchtkriterien nicht mit den Stichworten einer Seite übereinstimmen, diese Stichworte mehrere Bedeutungen haben oder ein Synonym verwendet wurde.

Das wirkliche Problem ist nicht, dass die Suchmaschine schlecht programmiert wurde. Das große Problem ist, dass die Suchmachine ohne Kontext und mit den sehr heterogenen Formaten der Webseiten die wichtige Information herausfinden soll. Ohne Kontext ist es praktisch unmöglich für eine Suchmaschine, einige der folgenden Probleme zu lösen:

- Mehrdeutigkeit: Viele Wörter können, abhängig vom Kontext, unterschiedliche Bedeutungen haben.
- Traditionelle Kenntnisse: Wenn man in einer bestimmten Kultur aufgewachsen ist, nimmt man einige Definitionen für wahr, obwohl diese Definitionen keine absolute Wahrheit sind, sondern Darstellungen der realen Welt, die auf der Tradition basieren. So ist z.B. der Nachname einer Person nur ein Word. Die Aussage ist normalerweise wahr in Deutschland, aber in anderen Ländern nicht.
- Dialekte: Lokale oder regionale Sprachvarietät.

Es ist auch kompliziert, die verschiedenen Dateiformate (z.B. ASCII vs. Unicode) und die verwendeten natürlichen Sprachen zu unterscheiden [27]. Deswegen kann die Verwendung einer Suchmaschine nicht immer helfen, die richtige Information zu finden.

Ein weiteres Problem ist die Zuverlässigkeit der Information, da eine große Menge des Inhalts im Web täglich geändert wird. Das beispielsweise zur Folge, dass die Ergebnisse einer Abfrage ständig aktualisiert werden müssen.

2.3.2 Semantisches Web

2001 wurde von Tim-Berners-Lee [14] eine mögliche Lösung für das ungeordnete Wachstum des Webs vorgeschlagen. Sein Vorschlag war das semantische Web, das die Weiterentwicklung des Internets und den Mechanismus der Extraktion von Informationen verbessern sollte. Diese Änderungen sollten außerdem die Suche erleichtern und eine Zuordnung ermöglichen. Mithilfe des semantischen Webs sollte Information eine Bedeutung erreichen.

> „The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.“[14]

Die Idee des semantischen Webs ist, dass Maschinen und Menschen Informationen verarbeiten können und, basierend auf die Inhalte, Beziehungen gesetzt werden können. Damit die Informationen verknüpft werden können, ist es notwendig, dass die Informationen eine feste, vordefinierte Struktur haben. Mit Hilfe dieser Struktur

wird die entsprechende Bedeutung festgelegt. Tim-Berners-Lee [14] erläuterte das semantische Web wie folgt:

> „The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing Web are already under way. In the near future, these developments will usher in significant new functionality as machines become much better able to process and ünderstandthe data that they merely display at present.“

Das semantische Web basiert auf den offenen Standards URI (Abschnitt 2.3.3), RDF (Abschnitt 2.3.4), RDFS (Abschnitt 2.3.5), SPARQL (Abschnitt 2.3.6) und OWL (Abschnitt 2.3.7), für die Beschreibung von Informationen. Dank dieser Standardisierung soll es möglich sein, Daten zwischen verschiedenen Plattformen auszutauschen. Burkhand (Burkhard, Burkhardt?) sieht das semantische Web als eine „Vision, die den globalen Informationsaustausch zwischen unterschiedlichen Menschen, Funktionen, Sprachen, Medien, Kulturen zu Ziel hat“ [26].

2.3.3 URI

Ein Uniform Resource Identifier (URI) ist eine Folge von Zeichen zur Identifizierung von abstrakten oder physischen Ressourcen [27]. Eine Ressource ist jedes Objekt, das man identifizieren und beschreiben möchte. Die URIs haben folgendes Schema.

Listing 1: Schema von URIs [28]

```
URI = scheme":" authority [path]["?"query]["#"fragment]
```

- schema: z.B. http, ftp, mailto
- authority: Der Host.
- path: Der Pfad in dem Host
- query: Kette zur Identifizierung von nicht hierarchischen Ressourcen
- fragment: Referenz eines Teiles innerhalb einer Ressource.

Folgende sind gültige URI nach der Spezifikation RFC3986.

Listing 2: Beispiele von gültigen URIs [28]

```
ftp://ftp.is.co.za/rfc/rfc1808.txt
http://www.ietf.org/rfc/rfc2396.txt
ldap://[2001:db8::7]/c=GB?objectClass?one
mailto:John.Doe@example.com
news:comp.infosystems.www.servers.unix
tel:+1-816-555-1212
telnet://192.0.2.16:80/
```

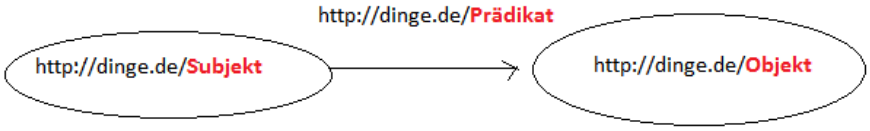


Abbildung 5: Repräsentation von RDF durch einen Graphen

```
urn:oasis:names:specification:docbook:dtd:xml:4.1.2
```

Ein wichtiger Punkt ist, dass ein URI nicht existieren muss, d.h. man kann sein eigenes URI gemäß der Spezifikation bilden.

2.3.4 RDF

Das Resource Description Framework (RDF) ist eine grundlegende Sprache für die Beschreibung strukturierter Informationen im Web und im Informationsaustausch [29, 23]. RDF wurde zur Repräsentierung von Metadaten[6] entwickelt. Dazu wurden folgende, wichtige Aspekte bei seinem Entwurf berücksichtigt:

- Die Information ist (im Web) verteilt. Es sind geeignete Identifizierungs- und Zugriffsmechanismen notwendig.
- Die Information (im Web) ist subjektiv. Aussagen müssen diskutierbar und ableitbar sein.
- Die Repräsentationen der Information müssen erweiterbar und einfach zu benutzen sein.

1997 wurde von W3C[7] die erste Spezifikation vorgeschlagen und 1999 wurde die erste offizielle Spezifikation veröffentlicht. Im Jahr 2004 wurde eine neue Überarbeitung [29] veröffentlicht, die die alte Definition ersetzt hat.

In RDF werden logische Aussagen über Ressourcen formuliert. Diese Aussagen werden als Statements bezeichnet. Hierfür wird zu jeder Ressource ein URI zugewiesen. Es gibt jedoch eine Ausnahme für die sogenannten „Blank Nodes“, die keine Bezeichnung tragen: das sogenannte **RDF Metamodel**.

RDF hat ein einfaches Metamodell, das durch eine Menge von Tripel (RDF Graph) beschrieben wird. Jedes Tripel beschreibt eine Beziehung (häufig als Pfeile dargestellt). Man kann RDF [31, 29] auch als einen gerichteten Graphen[8] sehen. Die Form eines Tripels ist in Abbildung 5 zu sehen. Ein Tripel hat drei Komponenten:

[6] „Metadata is machine understandable information about web resources or other things“ [30]
[7] http://www.w3.org/
[8] Menge von Knoten(Subjekten und Prädikaten), die durch gerichtete Kanten (Beziehungen) verbunden sind

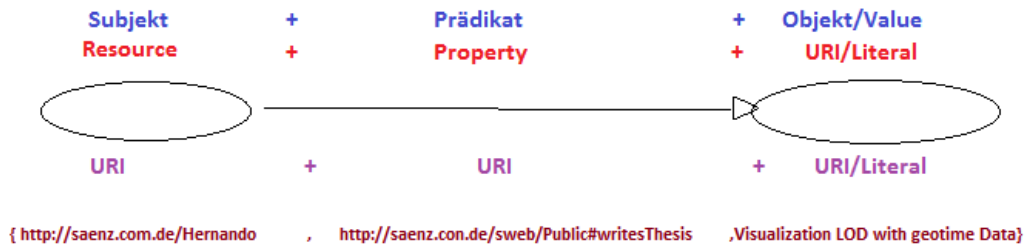


Abbildung 6: RDF vereinfachte Darstellung für die erste Aussage

Ein Tripel hat drei Komponenten:

- das **Subjekt** kann entweder ein URI oder eine „Blank Node“ sein.
- das **Prädikat** muss ein URI sein.
- das **Objekt** kann entweder ein URI, ein „Literal“ oder eine „Blank Node“ sein.

Nach der Spezifikation sollte ein RDF-Tripel in der Reihenfolge Subjekt, Prädikat, Objekt geschrieben werden. Literale werden verwendet, um einfache Datentypen wie Zahlen oder Daten zu identifizieren, aber es wäre auch möglich, diese durch ein URI darzustellen.

RDF Notationen

Als die erste Version des RDF von Tim Berners-Lees [30] vorgeschlagen wurde, wurde Lees Notation 3 (N3) als Syntax definiert. Später wurde in der RDF-Recommendation 2004[29] eine vereinfachte Version davon (N3) mit dem Namen N-Triples empfohlen. Danach wurde es erneut erweitert zu der RDF-Syntax Turtle. Im Jahr 2004 [29] wurde auch die RDF/XML-Syntax vorgeschlagen. Für die Beschreibung der Notationen (Turtle und RDF/XML) werden die folgenden zwei Aussagen verwendet:

1. Hernando schreibt eine Masterarbeit mit dem Titel Visualisierung von LOD mit Orts- und Zeitbezug (siehe Abbildung 6).
2. Hernando schreibt seine Masterarbeit mit dem Titel Visualisierung von LOD mit Orts- und Zeitbezug auf Deutsch (siehe Abbildung 7).

Turtle (Terse RDF Tripel Language)

Es ist eine Teilmenge von N3, die von Dave Beckett entwickelt wurde. Wichtige Charakteristiken von Turtle sind:

- URIs zwischen spitzen Klammern.
- RDF-Tripel (Subjekt, Prädikat und Objekt) endet mit einem Punkt.
- Literale unter Hochkomma.
- Leerzeichen ignoriert.

Die erste Aussage wird in Turtle wie folgt geschrieben.

Listing 3: N3 Notation für die Abbildung 6

```
< http://saenz.com.de/Hernando>
   <http://saenz.con.de/sweb/Public#writesThesis> Visualization
   LOD with geotime data.
```

Namensräume könnten, wie im folgenden Beispiel, Präfixe voranstellen, um Platz zu sparen.

Listing 4: Präfix

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl:  <http://www.w3.org/2002/07/owl#>.
```

Dank der Verwendung von Präfixen wird die Notation einfach. Das folgende Turtel-Beispiel zeigt die zweite Aussage.

Listing 5: N3 Notation für Abbildung 7

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix hss: <http://saenz.com.de/Public#>.

hss:Hernando hss: writesThesis [ dc:title Visualization LOD with
   geotime data;
       dc:language de];
```

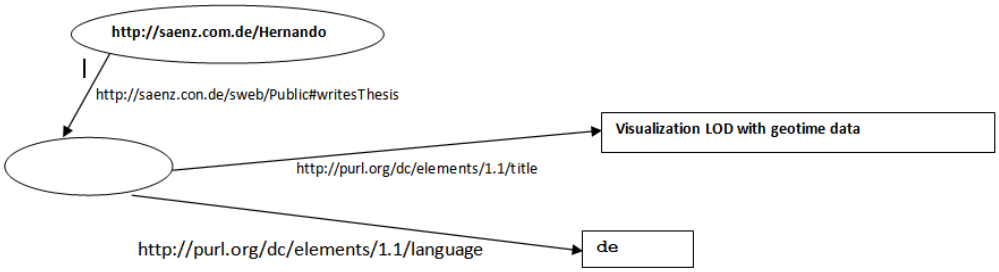


Abbildung 7: RDF Darstellung für die zweite Aussage

XML-Serialization

Die Darstellung von Turtle ist relativ einfach zu verarbeiten, sie ist jedoch nicht die am häufigsten verwendete RDF-Syntax. Da viele Programmiersprachen die Verarbeitung von XML unterstützen, wird diese Syntax häufiger benutzt. Das hat den Vorteil, dass man keine neue Bibliotheken oder ähnliches zur Verarbeitung von RDF installieren muss. Da XML verwendet wird, werden die Triples in hierarchischer Form geschrieben. Das Subjekt gruppiert die Prädikate und die zugehörigen Objekte.

Listing 6: XML Notation für die erste Aussage

```
<?xml version="1.0" encoding=utf8?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:hss=" http://saenz.con.de/sweb/Public#">
     <rdf:Description rdf:about=" http://saenz.com.de/Hernando">
          <hss:writesThesis> Visualization LOD with geotime data
             </ hss:writesThesis >
     </rdf:Description>
</rdf:RDF>
```

Die erste Zeile, die die XML-Version und Kodierung definiert, ist optional. Danach folgen die Knoten vom Typ rdf:RDF, der normalerweise als Wurzel des Dokuments benutzt wird. Dann werden die anderen Namensräume und Elemente geschrieben.

Listing 7: XML Notation für die zweite Aussage

```
<?xml version="1.0" encoding=utf8?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:hss="http://saenz.con.de/Public#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
     <rdf:Description rdf:about="
        http://saenz.com.de/Hernando">
          <hss:writesThesis rdf:parseType="Resource">
            < dc:title >
              Visualization LOD with geotime data
            </ dc:title >
            < dc:language >
              de
            </ dc:language >
        </hss:writesThesis >
     </rdf:Description>
</rdf:RDF>
```

2.3.5 RDF-Schema

RDFS ist ein Bestandteil von RDF, das die Möglichkeit bietet, Aussagen über Ressourcen und Beziehung zu schreiben. 1998 wurde von W3C die erste Spezifikation vorgeschlagen und 2004 wurde die erste offizielle Spezifikation veröffentlicht. Es handelt sich um ein spezielles RDF-Vokabular, deswegen können alle Anwendungen, die RDF unterstützen, RDFS lesen [32].

Das Ziel von RDFS ist, ein universelles Ausdrucksmittel zu werden, mit dem man innerhalb eines RDFS-Dokuments Aussagen über die semantischen Beziehungen eines nutzerdefinierten Vokabulars machen kann, um ein neues Vokabular definieren zu können. Dies ermöglicht, in einer Domäne vorkommende semantische Abhängigkeiten zu beschreiben [31].

Folgendes sind wichtige Möglichkeiten, die RDFS bietet:

- Definition von Klassen

  Ausprägung von Klassen durch <rdf:type>

- Definition von Hierarchien (Unterklassen, Superklassen)
- Definition von Eigenschaften

Die Klassen und Eigenschaften von RDF sind vergleichbar mit Programmiersprachen wie Java. Der Unterschied besteht darin, dass die Definition einer Klasse in den normalen Programmiersprachen auf die Eigenschaften seiner Ausprägungen basiert, im Gegensatz zu RDFS, in dem die Eigenschaften durch die Klassenzugehörigkeit definiert werden [32].

Listing 8: Zusammenfassung der RDF(S)-Sprachkonstrukte [31]

```
# RDF(S) Klassen
rdfs:Class rdf:Property rdfs:Resource rdfs:Literal rdfs:Datatype
   rdf:XMLLiteral
# RDF(S) Eigenschaften
rdfs:range rdfs:domain rdf:type  rdfs:subClassOf
   rdfs:subPropertyOf rdfs:label rdfs:comment
# RDF Listen
rdfs:Container rdf:Bag rdf:Seq rdf:Alt rdf:li rdf:_1rdf:_2 ...
rdfs:ContainerMembershipProperty  rdfs:member rdf:List
   rdf:first rdf:rest rdf:nil
# RDF Reikation
rdf:Statement   rdf:subject rdf:predicate    rdf:object
#RDF Attribute
rdf:about rdf:ID rdf:resource rdf:nodeID rdf:datatype
```

2.3.6 SPARQL

In den vorhergehenden Abschnitten wurde die Art und Weise definiert, wie die Semantik der Information durch RDF und RDFS beschrieben werden kann. In diesem Abschnitt wird erläutert, wie auf die mit RDF und RDFS gespeicherte Information zugegriffen werden kann.

„SPARQL Protocol and RDF Query Language“ (SPARQL) ist eine Abfragesprache für RDF. Im April 2006 wurde von W3C die erste Spezifikation („Recommendation“) vorgeschlagen und im Januar 2008 wurde die erste offizielle Spezifikation veröffentlicht [33]. Deren Name sieht der sehr bekannten Abfragesprache SQL[9] ähnlich, von der sie inspiriert wurde und mit der Ähnlichkeiten in der Syntax (z.B. SELECT) bestehen.

In SPARQL werden Variablen mit ? (oder $) markiert. Sie können an allen Positionen in den Tripeln verwendet werden. Das Ergebnis einer SPARQL-Abfrage ist eine Liste von Lösungen, die alle Bedingungen erfüllen [34].

[9] Structured Query Language (SQL) ist die standardisierte Abfragesprache der relationalen Datenbanken, die auf der relationalen Algebra basiert.

Das folgende Beispiel aus [34] zeigt, wie eine Abfrage und das entsprechende Ergebnis aussieht.

Listing 9: Ergebnis einer Abfrage in SPARQL [34]

```
Wissensbasis:

@prefix ex : <http://www.dfki.de/example#> .

ex:baispilov ex:name "Rudi Baispilov" .
ex:baispilov ex:company "CarFS AG " .
ex:baispilov ex:email <mailto:rudi.baispilov@carfs.de>.
ex:mustermann ex:name "Gesine Mustermann" .
ex:mustermann ex:company "DFKI GmbH " .
ex:mustermann ex:email <mailto:gesine.mustermann@dfki.de>.

SPARQL-Anfrage:

PREFIX ex : <http://www.dfki.de/example#>
SELECT ?person ?name ?company ?email
WHERE
{
? person ex:name ?name .
? person ex:company ?company .
? person ex:email ?email .
}

Ergebnis der Anfrage:

person name company email
ex:baispilov "Rudi Baispilov" "CarFS AG" <mailto:...>
ex:mustermann "Gesine Mustermann" "DFKI GmbH" <mailto:...>
```

Einschränkungen der Ergebnisse könnten mit FILTER gemacht werden.

Listing 10: Nutzung von Filter zur Einschränkung eines Ergebnisses [34]

```
SELECT ?person ?name ?company ?email
WHERE
{
FILTER regex(?name, "^Rudi")
?person ex:name ?name .
?person ex:company ?company .
?person ex:email ?email .
}
```

Es besteht die Möglichkeit, optionale Einschränkungen zu spezifizieren. Der Operator heißt OPTIONAL[10] und ist ein linkassoziativer Operator. Es existiert auch die Möglichkeit, Alternativen mit dem Operator UNION auszudrücken [34].

Listing 11: Nutzung von OPTIONAL und UNION [34]

[10] Analog zu LEFT OUTER JOIN in SQL

```
Wissensbasis:

@prefix ex: <http://www.dfki.de/example#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
ex:baispilov ex:name "Rudi Baispilov" .
ex:baispilov ex:email <mailto:rudi.baispilov@carfs.de> .
ex:mustermann foaf:name "Gesine Mustermann" .

SPARQL-Abfrage:

PREFIX  ex : <http://www.dfki.de/example#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT   ?person   ?name     ?email
WHERE    {
          {?person ex:name ?name }
    UNION {?person foaf:name ?name }.
    OPTIONAL{ ?person ex:email ?email }
    }

Ergebnis der Abfrage:

ex:baispilov  "Rudi Baispilov"  <mailto:rudi...@carfs.de>
ex:mustermann "Gesine Mustermann"
```

Es gibt viele andere zur Verfügung stehende SPARQL-Operatoren.

Die Spezifikation ist groß und enthält viele weitere Operatoren.

2.3.7 OWL

Im Abschnitt 2.3.5 wurde beschrieben, wie RDFS sich dafür eignet, einfache Aussagen über Ressourcen und Beziehungen zu modellieren. Im folgenden Abschnitt wird OWL zur Modellierung von Ontologien beschrieben. Die Grundlagen von Ontologien wurden in Abschnitt 2.2 erläutert.

Zur Erstellung von Ontologien ist es erforderlich, kompliziertes Wissen auszudrucken und komplexere Zusammenhänge darzustellen. Dazu ist RDFS nicht ausreichend, weil es nur beschränkte Ausdruckmittel besitzt [35].

Deshalb wurde OWL (Web Ontology Language) im Februar 2004 von W3C als Ontologiesprache vorgeschlagen, die auf der Prädikatenlogik basiert. Bei der Entwicklung waren zwei Charakteristiken (die Ausdrucksstärke der Sprache und effiziente Schlussfolgerungen) gleichermaßen wichtig [35]. Im Jahr 2009 wurde eine neue Version [36], OWL 2, veröffentlicht, die die Version 1 erweiterte. Der aktuelle, normative Standard (W3C Recommendation) für Version 2 wurde im Dezember 2012 veröffentlicht [37].

In OWL 1 gibt es drei verschiedene Teilsprachen von OWL: OWL Full , OWL DL, OWL Lite. Folgende Eigenschaften wurden von Hiztler et al. [35] beschrieben und weisen Unterschiede untereinander auf:

- **OWL Full** enthält OWL DL und OWL Lite. Es ist sehr ausdrucksstark, unentscheidbar, und seine Semantik enthält einige Aspekte, die aus logischem Blickwinkel problematisch sind. Es wird durch aktuelle Softwarewerkzeuge nur bedingt unterstützt.
- **OWL DL** enthält OWL Lite und ist Teilsprache von OWL Full. Es ist entscheidbar und wird von aktuellen Softwarewerkzeugen fast vollständig unterstützt.
- **OWL Lite** ist Teilsprache von OWL DL und OWL Full, entscheidbar, weniger ausdrucksstark.

In OWL 1 werden drei mögliche Syntaxen für die Beschreibung der Ontologie definiert: RDF(S) Syntax, abstrakte Syntax, DL Syntax.

OWL 2 hat fünf Teilsprachen: OWL EL, OWL RL, OWL QL, OWL DL, OWL FULL. Die ersten drei sind neue Teilsprachen. OWL Lite (OWL1) ist veraltet und wird nicht mehr verwendet.

In OWL 2 werden fünf mögliche Syntaxen für die Beschreibung der Ontologie mit einem spezifischen Ziel definiert: Funktionale Syntax, RDF-Syntax, XML-Syntax, Manchester-Syntax, Turtle.

2.4 Visualisierung

Die Datensätzen haben Beziehungen untereinander, die nicht immer einfach zu erkennen oder zu entdecken sind. Dafür gibt es ein gutes Werkzeug zur Analyse, Bewertung und Erkundung von Datensätzen. Dieses Werkzeug ist die Visualisierung von Daten [38], die normalerweise auf Computer generierten Graphik-Systemen basiert. Diese Graphik-Systeme sollen uns mit einer schnelleren Auffassungsgabe der Daten helfen.

> „Computer-based visualization systems provide visual representations of datasets intended to help people carry out some task better. These visualization systems are often but not always interactive. Resource limitations include the capacity of computers, of humans, and of displays. The space of possible visualization system designs is huge and full of tradeoffs.„ [39]

Es wurde das Wort Visualisierung verwendet ohne bisher eine Definition dafür zu geben. Nun ist es Zeit dafür. Für diese Masterarbeit werden folgende Definitionen von Preim et al. [40] verwendet:

> „Visualisierung ist der Prozess der Transformation von Informationen in eine visuelle Form, die es dem Benutzer auf visuelle Weise gestattet, verborgene Aspekte in den Daten zu entdecken, die für Exploration und Analyse wesentlich sind.„

Für die Visualisierung ist es daher wichtig nachzudenken, wie die Menschen die Bedeutung von Bildern extrahieren (Wahrnehmung), was die Leute von einem

Visualisierungstyp verstehen (Kognition) oder welche Bedeutung ein Bild in einem spezifischen, sozialen und / oder kulturellen Kontext hat [5].

2.4.1 Informationsvisualisierung

Zur Erstellung grafischer Darstellungen der Daten gibt es heutzutage, dank der Entwicklung des Computers und des De-facto-Standards verwendeter Software wie Excel, viele Werkzeuge. Es ist gleichzeitig gut und schlecht, da manchmal die Visualisierungsarten falsch verwendet werden. Damit die Benutzer mit den zahlreichen Möglichkeiten nicht überfordert wird, muss die Information so gefiltert, verdichtet und aufbereitet werden, dass sie weitestgehend effizient und effektiv genutzt werden kann. Dazu ist die Integration verschiedener Wissenschaften, wie die der Wahrnehmungspsychologie oder die der Linguistik, notwendig [41].

Gebiete wie das Informationsdesign[11] nutzen diese Integration nicht nur für die Informationsvisualisierung, sondern auch für technische Schreiben (z.B. Fehlermeldungen). Moser [41] definiert die Aufgabe der Informationsvisualisierung folgendermaßen:

> „Die Informationsvisualisierung beschäftigt sich damit, numerische Daten so aufzubereiten, dass ein Betrachter daraus möglichst einfach die gewünschten Erkenntnisse ableiten kann. Dabei kommen oft Interaktionstechniken zum Einsatz, die ein zusätzliches Filtern und Hervorheben gewisser Daten ermöglichen.„

Eine klare Abgrenzung zwischen den Visualisierungsdisziplinen ist nicht einfach, da das Specktrum der Datenvisualisierung sehr groß ist. Deswegen ist es zum Beispiel schwierig, die Grenze zwischen der Informationsvisualisierung und der wissenschaftlich-technischen Visualisierung zu finden [40].

2.4.2 Visuelle Wahrnehmung

Wie oben beschrieben, spielen diverse Faktoren eine wichtige Rolle in der richtigen, grafischen Darstellung von Daten. Einer davon, die Wahrnehmung, ist wichtig, nicht nur, weil sie die Interpretation der Daten, sondern auch die Aufmerksamkeit, die der Empfänger der Information schenkt, beeinflussen kann. Die Wahrnehmung ist nicht nur relevant für die Visualisierung oder die Psychologie, sondern auch für die Entwicklung des Menschen.

> „Die Entwicklung des Menschen wurde und wird entscheidend von seiner Fähigkeit geprägt, seine Wahrnehmungen, Empfindungen und Schlussfolgerungen zu abstrahieren und weiterzuvermitteln. Er ist imstande, seine Erfahrungen zu formalisieren; er kann zwischen verschiedenen Bereichen Beziehungen herstellen und Querverbindungen bewusst ausnutzen; das wesentliche daran ist, dass

[11] „Information design is defined as the art and science of preparing information so that it can be used by human beings with efficiency and effectiveness„[42]

er dieses abstrakte Wissen auch weitergeben kann. Einzelerfahrungen können somit kollektiv nutzbar werden.„[1].

Bosch et al. [43] beschreiben die Wahrnehmung als die Verbindung zwischen der Außerwelt und der Innenwelt eines Individuums und sagen, dass die Wahrnehmung die visuelle Aufnahme von Reizen, Hören, Tasten, Schmecken und Riechen beinhaltet. Kroeber-Riel et al. [44] definieren Wahrnehmung als einen Informationsverarbeitungsprozess, durch den ein Individuum Kenntnis von sich selbst und von seiner Umwelt erhält. Sie definieren drei wichtige beeinflussende Merkmale für diesen Verarbeitungsprozess der Informationen. Diese sind Subjektivität, Selektivität und Aktivität [45].

- **Subjektivität:** Jede Person nimmt seine Umwelt individuell wahr.
- **Selektivität:** Nur Reize, die Aufmerksamkeit erregen, werden wahrgenommen. Die Aufmerksamkeit ist dann der Filter, der nur einigen Reize die tatsächliche Verarbeitung erlaubt.
- **Aktivität:** Aktivität beschreibt die aktive Aufnahme und Verarbeitung von Reizen und Informationen.

Die Psychologie zerlegt die Wahrnehmung in elementare Komponenten. Sie definiert Gesetze zur Ordnung der verschiedenen Wahrnehmungskomponenten. Diese Thematik wird unter der Überschrift Gestaltpsychologie behandelt[45].

Folgende Gesetze sind die am häufigsten verwendeten Gesetze, die helfen können zu verstehen, wie räumliche und zeitliche Anordnungen die Aufmerksamkeit und die Wahrnehmung der Leute beeinflussen [39].

- Gesetz der Ähnlichkeit: Objekte, die ähnliche visuelle Eigenschaften wie Farbe, Form oder Größe haben, werden von den Augen gruppiert.
- Gesetz von Figur und Grund Trennung: Es wird zwischen Figur und Hintergrund differenziert, wenn ein Bild wahrgenommen wird.
- Gesetz der Nähe: Wenn Objekte nahe beieinander liegen, werden Sie als eine Gruppe wahrgenommen.
- Gesetz der Geschlossenheit: Die Augen versuchen fehlende Teile einer Figur automatisch zu verkomplettieren.
- Gesetz der Kontinuität: Wenn die Augen eine Richtung entdecken, verfolgen sie diese instinktiv weiter.
- Gesetz der Bewegung: Objekte, die sich in die gleiche Richtung bewegen, werden als zusammengehörig wahrgenommen.
- Gesetz der Symmetrie: Objekte, die symmetrisch zueinander angeordnet sind, ziehen mehr Aufmerksamkeit auf sich als Objekte, die zufällig angeordnet sind.
- Gesetz der Verbundenheit: Objekte, die sich vom Rest unterscheiden, werden zuerst wahrgenommen.

- Gesetz der Einfachheit: Einfache Objekte werden besser als komplizierte Objekte aufgenommen.

2.4.3 Ziel der Visualisierung

Die Kenntnis von der Art und Weise, wie der Verarbeitungsprozess der grafisch dargestellten Information (visuelle Wahrnehmung) funktioniert, soll helfen zu verstehen, dass viele Faktoren für die Auswahl des Zieles der Informationsvisualisierung eine Rolle spielen. Abhängig von der Fragestellung ist es möglich, verschiedene Visualisierungsarten für einen einzigen Datensatz zu verwenden. Deswegen ist es eine gute und richtige Darstellung entscheidend für den Erfolg der Visualisierung.

Die erste Frage, die man sich stellen sollte, ist die nach dem **Zielpublikum**. Hier sollte man sich über das Problem der Heterogenität des Zielpublikums Gedanken machen. Die Selektivität[12] bewirkt, dass die Information mit sehr unterschiedlichen Personen kommuniziert werden soll, die verschiedene Lebensläufe, Werte oder kulturelle Hintergründe haben können [46]. Aufgrund dessen ist es denkbar, je nach Zielgruppe verschiedene Ansichten anzubieten, wenn eine Visualisierung gleichzeitig mehrere Gruppen unterstützen soll.

Die zweite Frage ist, welche **relevanten Merkmale der Daten** der grafischen Darstellung betont werden sollen. Dies trägt zur Reduzierung der Information bei, die dargestellt werden soll [40].

Der dritte Faktor ist die **Nützlichkeit**[13] der Darstellung. Das heißt, wie nützlich eine Darstellung zum Verständnis der Daten sein kann [47]. Dazu sollte man sich auch Gedanken über die Aufgaben machen, die unterstützt werden sollen.

Es ist wichtig, sich die **Darstellung der Daten** zu überlegen. Dazu muss entschieden werden, welche Visualisierungsform angewandt wird, um die Anzahl der Datenelemente, die dargestellt werden können, zu erfahren, damit das Ergebnis der Visualisierung eindeutig und fehlerfrei interpretiert werden kann. Die Auswahl einer Visualisierungsform könnte mit Hilfe der Eigenschaften der Merkmale und Zusammenhänge einzelner Variablen einfacher herausgefunden werden. Ein Beispiel für die Verwendung dieser Eigenschaften ist die Einschränkung, die die Verwendung einer bestimmten Skala bietet. So könnten beispielweise nur bestimmte, zur Verfügung stehende Diagramme oder Visualisierungsformen ausgewählt werden, die mit dem Skalenniveau übereinstimmen [41, 40]. Ein Überblick über die Eigenschaften einiger Diagramme wird in Abschnitt 2.4.5 angezeigt. In Kapitel 4 wird eine Klassifizierung möglicher Darstellungen gemäß der Eigenschaften der Merkmale vorgestellt.

Ebenso wichtig ist es, darüber nachzudenken, was man mit den dargestellten Daten machen möchte. Das heißt, wenn sowohl die Daten als auch die Ergebnisse bekannt sind, ist es lediglich erforderlich, die Daten zu präsentieren, in Gegensatz zu dem Fall,

[12] Siehe 2.4.2

[13] Nützlichkeit entspricht in diesem Kontext auch der Funktionalität, Benutzerfreundlichkeit [47]

dass die Ergebnisse unbekannt sind und die Daten analysiert werden. Die Analyse der Daten benötigt normalerweise visuelle Mittel bzw. **Interaktionstechniken** (siehe Abschnitt 5.1.2).

Der letzte Aspekt, der beachtet werden sollte, ist die Herkunft der Daten (Datenquellen), die dargestellt werden sollen [40].

Es werden von Moser [41] außerdem sieben Punkte genannt, die beachtet werden sollten, damit die Visualisierung verständlich und lesbar wird:

- Die einfachsten Darstellungen sind oft die besten.
- Die Achsausschnitte sind so zu wählen, dass die Daten die Visualisierung ausfüllen.
- Nicht datenbezogene Elemente auf ein Minimum reduzieren.
- Perspektivische Darstellungen vermeiden.
- Einen passenden Ausschnitt für eine korrekte Interpretation der Daten wählen.
- Verschiedene Ansichten anstelle verschiedener Designs anbieten.
- Für zusammenhängende Graphiken stets dieselbe Skala verwenden.

2.4.4 Visualisierungsaufgaben und Interaktionstechniken

Diverse Visualisierungen erlauben die Interaktion des Benutzers. Dies ermöglicht, je nach Bedürfnis, die Ansicht zu zoomen (oder zu wechseln), nicht-relevante Daten auszublenden, die Daten nach einem spezifischen Kriterium zu ordnen oder gewisse Daten hervorzuheben. Die Interaktion wird durch die Anwendung einiger bekannter Techniken erreicht, die das Grundprinzip der Informationssuche (**„Overview first, zoom and filter, then details-on-demand“** [48]) unterstützen [41].

1996 wurden von Schneidermann[48] grundlegende Aufgaben beschrieben, die alle Visualisierungen unterstützen sollen. Dazu wurde eine Taxonomie von Informationsvisualisierungslösungen vorgeschlagen (Task by Data Type Taxonomy). Die sieben Aufgaben sind folgende:

1. **Overview:** Der Überblick über den gesamten Informationsraum hilft globalen Mustern, Trends zu erkennen und ein Verständnis dafür zu gewinnen.
2. **Filter:** Mit Hilfe von Filtern wird die Reduzierung der Datenmenge vereinfacht.
3. **Zoom:** Zur Betrachtung anderer (globaler) Stufen von Details.
4. **Details-on-demand:** Nach Auswahl eines bestimmten Datenobjekts werden die Details dieses Objekts angezeigt.
5. **Relate:** Beziehungen zwischen den Daten sollen betrachtet und Verknüpfen von Ansichten angeboten werden.

6. **History:** Unterstützung von „undo“, „redo“ und „progressive Verfeinerung“
7. **Extract:** Extraktion von Untermengen der Information

Außerdem gibt es noch andere Interaktions-Techniken, die wichtige Interaktionen anbieten[41]:

1. Markieren: Es ermöglicht, Datenwerte, eine Detailansicht oder einen Filter auszuwählen. Die markierten Werte sollen visuell hervorgehoben werden.
2. Drill-down: „Die Drill-down Funktion erlaubt es dem Benutzer, auf einer aggregierten Sicht zu beginnen, die ihm eine schnelle Übersicht bietet.“
3. Perspektivische Verzerrung: Die Daten werden in der Nähe des Fokus vergrößert und alle andere verkleinert. Ein sehr bekanntes Beispiel ist das Dock von OS X[14]

2.4.5 Darstellung nach Datentyp

Nachdem die wichtigsten Visualisierungsaufgaben und das Visualisierungsziel in den vorherigen Abschnitten beschrieben wurden, werden nun einige, mögliche Kriterien zur Auswahl einer Darstellung je nach Datentypen beschrieben.

Es gibt viele Taxonomien zur Klassifizierung der Visualisierungsarten. Mehrere Autoren nutzen den Datentyp zur Klassifizierung von Visualisierungslösungen und zur Erstellung von Taxonomien, weil mit Hilfe der Datentypen ein Datenmodell gebaut werden kann, das als Basis für sämtliche Visualisierungen verwendet wird. Dieses Datenmodell repräsentiert typischerweise einen Ausschnitt der realen Welt und besteht aus Informationsobjekten mit ihren Merkmalen (Attributen) und Beziehungen untereinander [40].

Nach Keim [49] müssen folgende Datentypen beachtet werden.

- **Dimensionalität von Datenobjekten:** Abhängig von der Anzahl von Attributen, die dargestellt werden möchten.
  - **Eindimensionale Daten:** Daten, die nur ein Attribut haben, von dem sie abhängig sind, sind diesem Begriff zugeordnet. Ein Beispiel dafür kann die Geschwindigkeit sein, die durch ein Tachometer dargestellt werden kann. Die Darstellung von eindimensionalen Daten wird häufiger zum Vergleich der Daten untereinander verwendet als zur Anzeige eines einzigen Wertes. Ein Beispiel ist der Vergleich von Geschwindigkeiten mehrerer Autos. Dafür gibt es verschiedene Visualisierungstechniken, wie z.B. Balkendiagramme, Kreisdiagramme, Histogramme oder Tag-Clouds.
  - **Zwei dimensionale Daten:** Daten, die zwei abhängige Attribute haben, sind diesem Begriff zuzuordnen. Sie werden normalerweise verwendet, um Daten darzustellen, die eine Korrelation haben. Ein Beispiel ist

[14] „The Dock is the bar of icons that sits at the bottom or side of your screen. It provides easy access to some of the applications“ https://support.apple.com/kb/ht2474 (Stand 26.02.2013)

der Vergleich zwischen Einkommen und Ausgaben einer Person. Eine Visualisierungstechnik dieses Datentyps ist z.B. Scatterplot.

  - **Multidimensionale Daten:** In der Tat sind zweidimensionale Daten eine Sonderform von multidimensionalen Daten. Abhängig vom Autor der Klassifikation spricht man von multidimensionalen Daten, wenn die Anzahl der Attribute größer oder gleich drei [49] oder vier [40] ist. Ein bekanntes Beispiel multivariater Visualisierung sind 2DScatterplots, bei denen neben den zwei Dimensionen weitere visuelle Eigenschaften wie Form, Größe, Textur oder Transparenz benutzt werden.

- **Relationen:** Informationen, die nur Sinn machen, wenn die Relationen angezeigt werden. Hier werden zwei Unterteilungen definiert:
  - **Hierarchien (Bäume)** Wenn die Relation zwischen den Daten hierarchisch ist, werden beispielsweise Visualisierungen wie Bäume oder das Venn-Diagramm verwendet.
  - **Netzwerke (Graphen)** Wenn aber die Relationen zwischen den Daten keine Zuordnung haben oder es nicht wichtig ist zu zeigen, können diese Visualisierungen verwendet werden. Beispiele sind Mentionmap, Node-Link-Diagramm oder Tree-Ring.
- **Text und Hypertext**
- **Programmcode und Softwaresystem:** Darstellung durch Diagramme wie UML

2.5 Geographische Information

Heutzutage sind im Web häufig Karten zu finden, die uns helfen, Wohnungen, Häuser, Geschäfte, Sehenswürdigkeiten usw. zu lokalisieren. Die Karten sind nicht mehr etwas, das man selten nutzt, sondern ein Teil unseres Lebens.

Digitale Karten stellen viele geografische Informationen graphisch dar und vermitteln ein Bild unserer Umwelt. Die Bilder beinhalten topographische Merkmale wie z.B Gebäude oder Verkehrswege, die uns die Orientierung erleichtern [1].

Im Bereich der Geoinformation existiert wie im Web (W3.org) eine Organisation, die sich um die Standardisierung kümmert. Spezifischerweise beschäftigt sie sich mit der raumbezogen Informationsverarbeitung und Interoperabilität zwischen Diensten. Die Open GeoSpatial Consortium (OGC) ist eine internationale, Non-Profit- und freiwillige Organisation von mehr als 400 Mitgliedern (Unternehmen, Behörden und Universitäten). Alle Spezifikationen sind kostenlos und herunterladbar[15] [50]. Im Folgenden werden einige, wichtige Begriffe der Geoinformation, die relevant für diese Masterarbeit sind, erläutert.

[15] www.opengeospatial.org

Als geografische Information (Geoinformation) wird diejenige Information bezeichnet, die direkt oder indirekt einen Ort-, Lage-, Raum- und Zeitbezug hat. Der Raumbezug (auch Georeferenz) beschreibt die Lage in einem Bezugssystem, das die Geoinformation verarbeiten kann [1].

Es werden drei Datentypen zur Modellierung der geographischen Information verwendet, die Vektor-, Raster- und Metadaten [50]:

- Vektordaten (Features[16] [50]) dienen dazu, diskrete Phänomene zu repräsentieren.
- Rasterdaten (Coverages) sind Funktionen von einem raum-zeitlichen Definitionsbereich (coverage domain) in einem Wertebereich (coverage range).
- Metadaten stellen semantische Netze dar und nutzen die verfügbare Methode der semantischen/objektorientierten Modellierung.

Computer oder Systeme, die Geodaten austauschen, brauchen eine klare Definition der Position (location) durch ein räumliches Bezugssystem (engl. SRS). Im Prinzip gibt es drei Typen von SRS [1]:

1. Systeme, die städtische Standorte oder eindeutige Kennungen verwenden.
2. Systeme, die Koordinaten als numerische Werte in einem Koordinatensystem verwenden.
3. Systeme, die lineare Referenzierungen verwenden.

Eine eindeutige Kennung kann z.B. eine Postleitzahl sein. Ortsnamen benötigen zusätzliche Informationen, wenn man diese allein durch den Namen nicht lösen kann. Ein einfaches Beispiel wäre „Berlin“, das mehr als 50 Treffer bei Geonames[17] hat. Zur Konvertierung von Namen in Koordinaten gibt es Dienstleistungen, die „Gezetters“ (auch „Geocoding“) genannt werden. Koordinaten sind eine Folge von N Zahlen, die die Position eines Punktes im n-dimensionalen Raum bezeichnet. Sie werden immer mit einem Koordinatenreferenzsystem (CRS) ausgedruckt. Ein CRS ist ein Koordinatensystem, das auf die Erde bezogen ist [1].

Rechtwinklige Koordinatensysteme spielen eine wichtige Rolle bei der Geoinformation, da diese normalerweise bei der Berechnung der Koordinatengeometrie und der graphischen Datenverarbeitung verwendet werden [51].

2.5.1 WGS84 Geo Positioning

Für Lagebestimmungen und Kartennetzentwürfe ist die Form der Erde einem Ellipsoiden anzunähern, der durch Drehung um die kleinere Achse gebildet wird. Weltweit werden in den Landesvermessungen und Kartenprojektionen verschiedene Ellipsoiden-Systeme benutzt. Eines davon, das World-Geodetic-System 84(WGS84),

[16] „Ein Feature ist ein identifizierbares Objekt mit raumzeitlichen und beliebigen weiteren Attributen.“

[17] http://www.geonames.org/search.html?q=berlin. (Stand 02.02.2013)

hat eine wichtige Bedeutung, da die Satelliten dieses nutzen, um die Position zu bestimmen (GPS). WGS84 basiert auf einem geozentrischen Koordinatensystem mit einem einzigen Referenzellipsoid [51].

2.5.2 RDF Vokabular WGS84 Geo Positioning

WGS84_pos ist ein RDF-Vokabular zur Darstellung der Breitengrad-, Längengrad- und Höheninformationen im WGS84 geodätischer Bezugspunkte. Es wurde im Jahr 2003 von W3.org veröffentlicht. Es wird eine Klasse „Point„ definiert, die zu der Domäne „SpatialThing„ gehört und drei Eigeschaften hat: „latitude“, „longitude“ und „altitude“ [52]. Folgendes Beispiel zeigt die Darstellung eine Position mit WGS84.

Listing 12: Beispiel WGS84_pos mit RDF-XML [52]

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
  <geo:Point>
    <geo:lat> 55.701  </geo:lat>
    <geo:long> 12.552 </geo:long>
  </geo:Point>
</rdf:RDF>
```

Listing 13: Beispiel WGS84_pos mit RDF-XML [52]

```
@prefix rdf:
   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>

geo:Point geo:lat 55.701; geo:long 12.552
```

2.5.3 INSPIRE-Initiative

Die Interoperabilität zwischen raumbezogenen Informationsdiensten ist auch ein wichtiges Thema für die Europäische Union. Daher wurde im Jahr 2007 die Initiative INSPIRE (Infrastructure for Spatial Information in the European Community) gegründet. Das Ziel der Richtlinien ist die Schaffung eines europaweit verbindlichen, regulatorischen Rahmenwerks zur Kooperation und Interoperabilität auf Basis der offenen Standards des OGC [50].

2.6 Open Data

Das Internet ermöglicht die Verfügbarkeit von Daten. Das bedeutet jedoch nicht, dass man uneingeschränkten Zugriff auf alle Informationen haben kann. Vor sechs Jahren wurde eine neue Initiative ins Leben gerufen, die die freie Verfügbarkeit von Daten zum Ziel hat. Diese Initiative ist unter dem Namen „Open Data“ bekannt. Open Knowledge Foundation definiert Open Data wie folgt:

„Open data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike.“[53]

Open Data ist anderen Bewegungen wie Open Content oder Open Source ähnlich. Von letzterer wurden mehrere Prinzipien, die für die Definition von “Open Definition“ (Offenes Wissen) verwendet wurden, auch als Basis für die Definition von Open Data angewandt.

Open Definition [54] definiert elf Kriterien, die die Daten erfüllen müssen, um Offene Daten zu sein. Folgendes sind die Definitionen von Lucke und Geiger[55] für diese Kriterien.

1. **Zugang**: „Das Werk soll als Ganzes und zu nicht höheren als den Reproduktionskosten, vorzugsweise zum gebührenfreien Download im Internet verfügbar sein. Das Werk soll ebenso in einer praktischen und modifizierbaren Form verfügbar sein. “

2. **Weiterverbreitung**: „Die Lizenz darf niemanden davon abhalten, das Werk entweder eigenständig oder als Teil einer Sammlung aus verschiedenen Quellen zu verteilen oder zu verkaufen. Die Lizenz darf keine Lizenzzahlungen oder andere Gebühren für Verkauf oder Verteilung erforderlich machen. “

3. **Folgenutzung**: „Die Lizenz muss Modifikationen oder Derivate erlauben, ebenso wie deren Distribution unter den Bedingungen des ursprünglichen Werkes. Die Lizenz darf zu bestimmten Zuordnungen und Namensgebungen verpflichten. “ [55]

4. **Keine technischen Einschränkungen**: „Das Werk muss in einer Form zur Verfügung gestellt werden, die keine technischen Hindernisse für die Ausführung obiger Aktivitäten beinhaltet. Dies kann dadurch erreicht werden, dass ein offenes Datenformat verwendet wird, zum Beispiel eines, dessen Spezifikationen veröffentlicht sind, das frei verfügbar ist und das keine finanziellen oder anderen Hindernisse bezüglich der Nutzung aufweist. “

5. **Namensnennung**: „Die Lizenz kann als Bedingung für Weiterverbreitung und Folgenutzung die Zuordnung des Werkes zu seinen Schöpfern und Mitwirkenden erfordern. Diese Bedingung sollte nicht beschwerlich wirken. Zum Beispiel sollte, sofern Zuordnung verlangt wird, dem Werk eine Liste der jenigen beigefügt sein, die eine Zuordnung verlangen. “

6. **Integrität**: „Die Lizenz kann als Bedingung für die Distribution in modifizierter Form verlangen, dass das Derivat einen anderen Namen oder eine andere Versionsnummer als das ursprüngliche Werk erhält. “

7. **Keine Diskriminierung von Personen oder Gruppen**: „Die Lizenz darf keine Personen oder Gruppen von Personen diskriminieren. “

8. **Keine Einschränkung der Einsatzzwecke**:„ Die Lizenz darf keine Restriktionen gegen die Verwendung des Werkes in speziellen Einsatzzwecken enthalten. Zum Beispiel darf die Nutzung des Werkes für kommerzielle oder militärische Zwecke nicht ausgeschlossen werden. “

9. **Vererbung der Lizenzbestimmungen**: „Die Rechte, mit denen ein Werk ausgestattet wird, müssen bei der Weiterverteilung an alle Empfangenden übergehen, ohne dass diese verpflichtet sind, eine zusätzliche Lizenzierung zu erlangen. “
10. **Die Lizenz darf nicht einer spezifischen Sammlung zugehörig sein**: „Die Rechte, mit denen ein Werk ausgestattet wird, dürfen nicht davon abhängig gemacht werden, dass das Werk als Teil einer spezifischen Sammlung angehört. Falls das Werk aus der Sammlung extrahiert und innerhalb der Lizenzbestimmungen verwendet oder verbreitet wird, müssen die Empfangenden sämtliche Rechte erhalten, mit denen auch die ursprüngliche Sammlung ausgestattet war. “
11. **Die Lizenz darf die Verbreitung anderer Werke nicht einschränken**: „Die Lizenz darf keine Bedingungen an andere Werke stellen, die mit dem lizenzierten Werk gemeinsam vertrieben werden. Die Lizenz darf beispielsweise nicht verlangen, dass alle Werke, die auf demselben Medium enthalten sind, offen sind. “

2.6.1 Ziele von Open Data

Es gibt viele öffentliche Quellen von Informationen, wie z.B. wissenschaftliche Daten oder Regierungsdaten. Die Regierungsdaten sind beispielsweise eine große Informationsquelle, die beinahe ungenutzt geblieben ist, weil der Mechanismus zur Veröffentlichung prinzipiell Dateien sind, mit denen die Bearbeitung kaum möglich ist.

Die Implementierung von Open Data soll durch die Nutzung von standardisierten Dateiformaten bei der Bearbeitung helfen und unter anderem folgende Szenarien [53] ermöglichen:

- Transparenz und demokratische Kontrolle
- Partizipation
- Selbstbestimmung und Entwicklung
- Verbesserte oder neue Produkte und Dienstleistungen
- Innovation
- Verbesserte Effizienz staatlicher Dienstleistungen
- Verbesserte Effektivität staatlicher Dienstleistungen
- Verbesserte Evaluation von Politika
- Neues Wissen aus einer Kombination von Datenquellen und Mustern aus großen Datensammlungen

All diese Szenarien sollten mit Hilfe von Open Data möglich sein, weil die Verwendung von standardisierten Dateiformaten die Bearbeitung und die Kompatibilität

ermöglicht. Leider kann jedoch eine zuverlässige Interoperabilität nicht gesichert werden, da die Inhalte der Dateien ganz anders aussehen können. Dies ist relativ einfach herauszufinden, indem man beispielweise in zwei oder mehreren Datensätzen das Attribut (Spalte) vergleicht, das für die geografische Position verwendet wurde. Folgendes Beispiel zeigt das.

Es werden die Datensätze „Veranstaltungsangebote der Seniorenbegegnungsstätten“[18] und „Berliner Wochen- und Trödelmärkte“[19], die im Portal Open Data Berlin[20] veröffentlicht wurden, genommen und die Spalte(n), die die geografische Position beschreiben, verglichen. Das Ergebnis ist, dass der erste Datensatz die Spalte „Marktort“ und der zweite die Kombination „Straße“ + „PLZ / Ort“ verwendet, was nicht kompatibel ist.

2.6.2 Linked Open Data

Das Problem der Interoperabilität[21] zwischen offenen Informationsquellen führt zur Notwendigkeit der Integration von Daten. Diese Notwendigkeit wurde 2006 von Berners-Lee[56, 57] beschrieben und die Lösung „Linked Data„ durch die Nutzung der Technologien (RDF, RDFS, SPARQL, usw.) des semantischen Webs vorgeschlagen.

Später wurde das Konzept von Linked Data um Linked Open Data erweitert. Die folgende Definition beinhaltet die aktuelle Bedeutung von Linked Open Data (LOD). Die Definition stammt von Lucke und Geiger [55]:

> „Offene vernetzte Daten sind sämtliche Datenbestände, die im Interesse der Allgemeinheit der Gesellschaft ohne jedwede Einschränkung zur freien Nutzung, zur Weiterverbreitung und zur freien Weiterverwendung frei zugänglich gemacht und über das World Wide Web miteinander vernetzt sind.„

Durch die Vernetzung offener Daten über das Internet, das LOD ermöglicht, wurden zahlreiche Möglichkeiten eröffnet wie z.B. die Nutzung der Daten für Statistiken, Auswertungen, Karten oder Publikationen und das Verständnis von Zusammenhängen zwischen Daten innerhalb kurzer Zeit [55].

2.6.3 Grundprinzipien

Es wurden vier Grundprinzipien[22] von Berners-Lee [57] zur Veröffentlichung von Daten im Web definiert, damit alle veröffentlichten Daten Teil eines einzigen globalen

[18] http://daten.berlin.de/datensaetze/veranstaltungsangebote-der-seniorenbegegnungsst%C3%A4tten (Stand 07.03.2013)

[19] http://daten.berlin.de/datensaetze/berliner-wochen-und-tr%C3%B6delm%C3%A4rkte (Stand 07.03.2013)

[20] http://daten.berlin.de (Stand 07.03.2013)

[21] im Abschnitt 2.6.1 wurde ein Szenario (Open Data) erläutert, in dem dieses Problem aufgetreten wird.

[22] bekannt als „Linked Data principles“[58].

Datenraums werden:

- Verwendung von URIs zur Identifizierung von Dingen
- Verwendung von HTTP URIs , damit auf die Dinge zugegriffen und verwiesen werden kann.
- Wenn ein URI befragt wird, sollten nützliche Informationen durch die Verwendung von Standards (RDF, SPARQL) angeboten werden.
- Es sollten Links zu anderen „URLs“ hinzugefügt werden, damit neue Dinge entdeckt werden können.

3 Stand der Technik

Im diesem Kapitel wird der Stand der Technik in verschiedenen Bereichen der Visualisierung beschrieben. Am Anfang werden zwei Methoden zur Visualisierung von räumlich-zeitlichen Daten erläutert. Danach werden diverse Modelle zum Verständnis des Visualisierungsprozesses präsentiert. Es werden außerdem wichtige Anforderungen für die Visualisierung von LOD und bestehende Lösungen kurz beschrieben.

3.1 Techniken zur Visualisierung von räumlich-zeitlichen Daten

3.1.1 Die drei Komponenten der räumlich-zeitlichen Daten und das „pyramid“ Model

Im Jahr 1984 wurden von Peuquet [59] drei grundlegende Komponenten beschrieben, auf die räumlich-zeitliche Visualisierungen achten sollten. Jede Komponente soll eine Art von Frage beantworten. Die erste Komponente ist der **Ort**, an dem das Attribut liegt. Normalerweise werden Karten zur Darstellung verwendet. Die Zweite ist das **Attribut**, das dargestellt werden möchte. Das kann durch Diagramme visualisiert werden. Zuletzt wurde die **Zeit**, also wann etwas passiert ist, festgelegt. Zeitleisten werden häufig zur Darstellung verwendet. Abbildung 8 zeigt diese Komponenten der zeitlich-räumlichen Visualisierung.

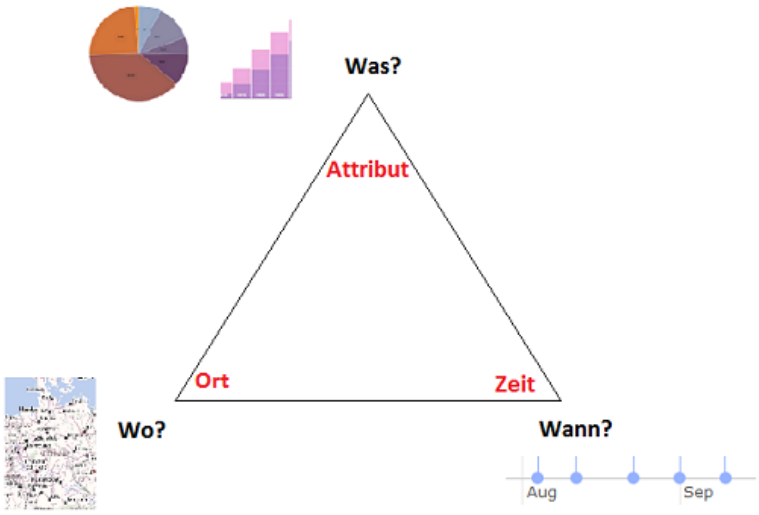


Abbildung 8: Drei Komponenten der zeitlich-räumlichen Visualisierung

Im Jahr 1994 wurde dieses „triad“ Modell [60] erweitert und als Grundlage für das „triad“ Framework verwendet. Das Framework wurde vorgeschlagen, um drei Arten von Fragen beantworten zu können, die zwei von diesen Attributen kombinieren. Die möglichen Kombinationen sind folgende:

- **Wann + Wo -> Was**
- **Wann + Was -> Wo**
- **Was + Wo -> Wann**

Im Jahr 2000 wurde das „pyramid“ Framework [61] vorgeschlagen. Das Framework ist eine zusätzliche Erweiterung des „triad“ Frameworks, das die Ebene der Information unterscheiden kann. Die Ebene von Mennis et al. [61] wurden wie folgt definiert:

- Daten: sind gespeicherte Beobachtungen.
- Informationen: sind Daten, die bearbeitet und konzeptualisiert wurden, um eine Bedeutung zu geben.
- Wissen: ist ein kumulatives Verständnis von Informationen

Dank dieser Trennung der Ebenen ist es möglich, Beobachtungsdaten (Daten-Ebene) mit abgeleitetem, menschlichem Wissen (Wissen-Ebene) zu integrieren, um die Analyse zu unterstützen. Die Integration solle eine semantische Repräsentation der drei grundlegenden Komponenten (Ort, Zeit, Attribute) erlauben.

3.1.2 Zeitleisten und Karten

Eine Zeitleiste ist eine grafische Darstellung von Zeiträumen, die sehr häufig mit anderen grafischen Darstellungen verbunden wird. Ihr Ziel ist ein besseres Verständnis des zeitlichen Verhaltens der Daten. Folgende Abbildung zeigt, wie eine Zeit-Leiste aussehen kann, die eine Anzahl von Events zwischen Juli und Dezember darstellt.

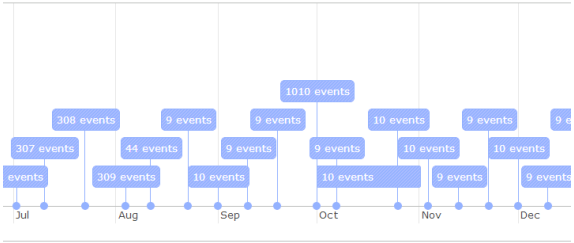


Abbildung 9: Zeitleisten

Kraak [2] beschreibt, wie eine Zeitleiste in Verbindung mit einer Karte eine gute und einfache Lösung für die Interaktionen zwischen Daten mit Orts- und Zeitbezug sein kann. Die Idee ist nicht wirklich neu und wurde bereits beispielsweise in dem Projekt „Tempest“ von Edsall und Peuquet in 1997 [62] verwendet.

Nach Kraak soll die Zeitleiste die grundlegende Interaktionstechnik sein, um zeitbezogene Fragen zu beantworten, weil sie den Zugriff auf die verfügbaren Daten in verschiedenen zeitlichen Räumen und zeitlichen Zoom-Operationen ermöglicht. Außerdem ermöglicht sie auch, die drei grundlegenden Fragen dieser Art von

Daten „was“, „wo“ und „wann“ zu beantworten. In Abbildung 10 wird ein interessanter Würfel[23] zur Darstellung der drei Dimensionen (Ziel, Ort, Zeit) von Kraak gezeigt.

- Ziel (was): wird durch ein Objekt angezeigt.
- Ort (wo): wird durch die Karte angezeigt.
- Zeit(wann): wird durch die Zeitleiste angezeigt.

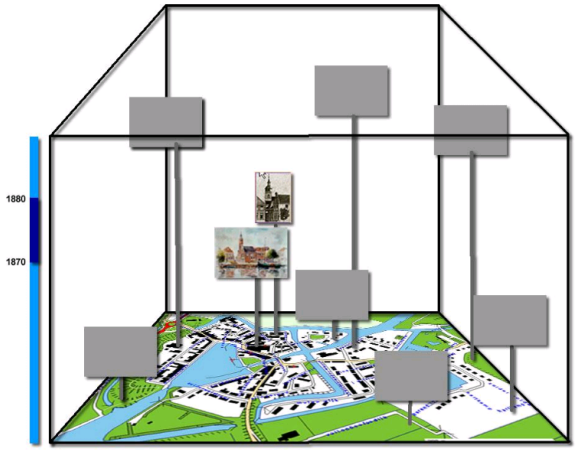


Abbildung 10: Ort-Zeit Würfel von Kraak [2]

3.1.3 Zeit-Welle

Im Jahr 2003 wurde von Tominski et al. [63] eine Visualisierungstechnik vorgeschlagen, um mehrere zeitabhängige Merkmale visualisieren zu können. Die Technik, die Zeit-Rad genannt wurde, wird in Abbildung 11 gezeigt und wurde wie folgt beschrieben:

> „Es werden um eine zentrale Zeitachse acht Merkmalsachsen angeordnet. Der visuelle Zusammenhang von Zeit und Merkmalsausprägungen wird durch Linien zwischen den Zeitobjekten auf der Zeitachse und korrespondieren den Werten auf der Merkmalsachse hergestellt.“ [63]

Nach Li und Kraak [4] führt die Darstellung der Zeit durch eine Zeitleiste (lineare Zeit) oder ein Zeit-Rad (zyklische Zeit) nicht immer zu befriedigenden Lösungen. Deswegen haben sie im Jahr 2008 die Zeit-Wellen-(„Time Wave“) Methode vorgeschlagen. Das Ziel ist eine Verbesserung der visuellen Erforschung und Analyse von Geodaten mit Zeit- und Raumbezug. Die Visualisierung kombiniert die Stärken von Zeitleiste und Zeit-Rad, damit lineare und zyklische Zeit gleichzeitig dargestellt werden können.

[23] 1970 wurde von Hägerstrand zum ersten Mal die Idee des Raum-Zeit-Würfels vorgeschlagen, der ermöglicht, sowohl räumliche als auch zeitliche Informationen in einer intuitiven Weise zu visualisieren. Dieser Würfel beinhaltete nicht eine Karte, sondern geographische Koordinaten.

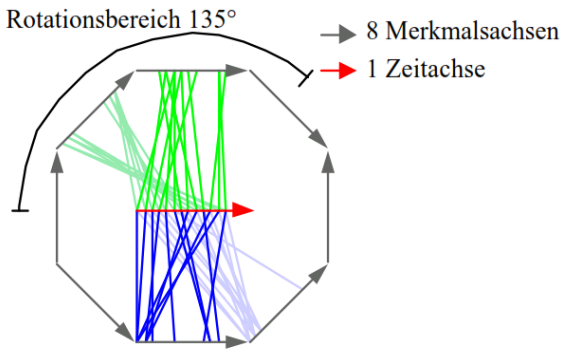


Abbildung 11: Die Zeitrad -Technik [3]

Die Wellenlänge stellt eine Zeitspanne dar, die ein Zyklus des Rads ist. Ein Zeitraum entspricht einer bestimmten Zeiteinheit (Jahr, Tag, Monat). Die Amplitude der Welle entlang der vertikalen Achse entspricht einem Viertel der Zeiteinheit und der Radius entspricht einem Zeit-Rad (siehe Abbildung 12).

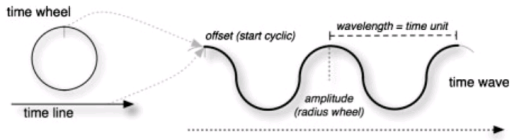


Abbildung 12: Die Zeit-Wellen-Methode [4]

Die Visualisierung kann nach Li und Kraak auf folgende Arten von Fragen antworten:

- Wann(Zeit).
- Wie lange (Zeitdauer).
- Wie oft (Häufigkeit).
- Welche Reihenfolge (Reihenfolge von Ereignissen).

Folgende Abbildung nach Li und Kraak zeigt, wie die Fragen durch die Visualisierung beantwortet werden können.

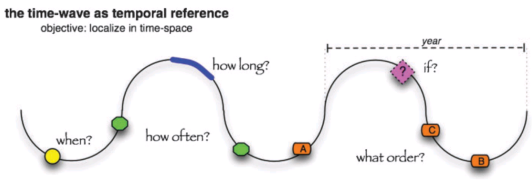


Abbildung 13: Time Wave beantwortete zeitliche Fragen [4]

3.1.4 Hitzekarte

Hitzekarten (Heat maps) sind Diagramme, die zur Visualisierung von Daten besonders geeignet sind um großen Datenmenge intuitiv und schnell zu erfassen. Die Daten werden mit einer zweidimensionalen Farbe dargestellt. Diese farbige Kodierung ermöglicht Abweichungen in den Daten, die durch die Farbvariationen zu erkennen sind. Der Begriff Hitzekarte ist leider sehr allgemein definiert. Deswegen können viele verschiedene Darstellungen dieses Begriffs wie z.B. Treemaps oder Choroplethenkarte passen.

Für die Masterarbeit wird folgende Definition des Begriffs Hitzekarte verwendet:

Eine Hitzekarte ist ein Diagramm, die eine Temperaturverteilung auf einer Landkarte darstellt.

Basierend auf der Definition hat eine Hitzekarte drei Dimensionen. Die ersten zwei Dimensionen sind die geographische Länge (x-Achse) und die geographische Breite (y-Achse). Die dritte Dimension ist die Farbe [9].

Die folgende Abbildung zeigt eine Hitzekarte, die mit Google Maps generiert wurde.

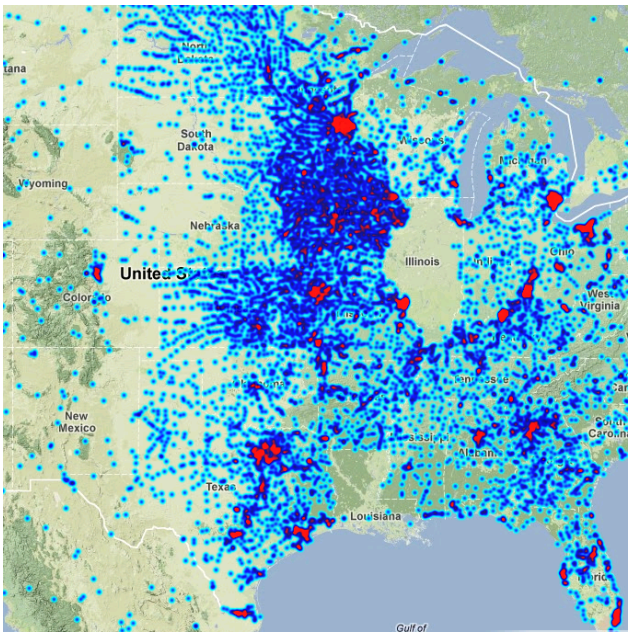


Abbildung 14: Hitzekarte

3.1.5 Choroplethenkarte

Eine Choroplethenkarte ist eine thematische Karte, in der Flächen dargestellt werden. Dazu werden Farbe oder Punkte verwendet. Die Flächen, die den gleichen Wert darstellen, werden mit der gleichen Farbe dargestellt und von den anderen

Flächen differenziert. Die Choroplethenkarte zeigt die Variabilität innerhalb einer Region.

Die folgende Abbildung zeigt eine Choroplethenkarte, die mit Google Maps generiert wurde.

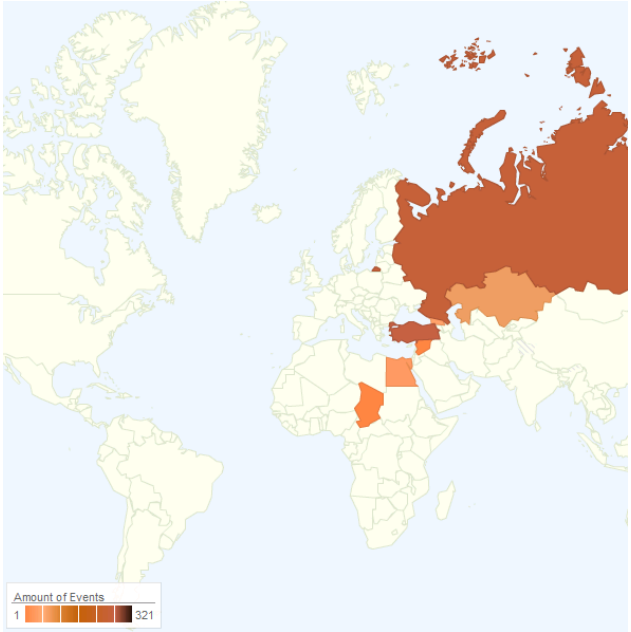


Abbildung 15: Choroplethenkarte

3.2 Eine gemeinsame Sprache für Visualisierung

Duke et al. [5] beschreiben einen zyklischen Interaktionsprozess (siehe Abbildung 16) zwischen verschiedenen Komponenten der Visualisierung, der aus vier Teilen besteht (Mensch zu Mensch, Mensch zu System, System zu System, System zu Mensch).

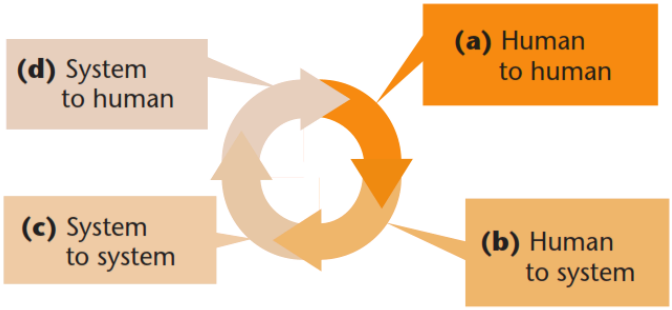


Abbildung 16: Visualisierungszyklus [5]

In jedem Unterprozess gibt es zwei Komponenten, die miteinander kommunizieren sollen. Deshalb schlagen sie eine gemeinsame Sprache zur Beschreibung der Visualisierungskomponenten vor, damit der Sinn der Visualisierung nicht verloren geht. Die

gemeinsame Sprache kann in drei verschiedene Stufen gemäß Formalisierungsgrad festgestellt werden.

	$Terminologie \longrightarrow$	$Taxonomie \longrightarrow$	$Ontologie$
$Formalisierungsgrad$	$+$	$++$	$+++$

Sie nennen und erläutern außerdem vier Gründe für die Formalisierung der Visualisierung durch eine semantische Ontologie (RDF und OWL), die helfen, die Visualisierung dem semantischen Web anzunähern: Zusammenarbeit, Zusammensetzung, Wiederholbarkeit (durch die Speicherung der Schritte zur Erstellung der Visualisierung) und Bildung.

3.3 Visualisierungsmodelle

Die Visualisierung ist mehr als ein einfaches Bild. Deswegen gibt es in der Visualisierung, wie in vielen anderen Bereichen, viele verschiedene Arten von Komponenten oder Teilen zu klassifizieren oder zu modellieren. Der gesamte Prozess einer Visualisierung impliziert die Zusammenarbeit und Interaktion vieler Elemente mit dem Ziel, verständliche Informationen anzuzeigen. Damit der Transformationsprozess verständlich wird, werden nun einige der wichtigsten klassischen Visualisierungsmodelle beschrieben.

3.3.1 Klassische Modelle

„The pipeline model“

Anfang der 90er Jahre wurde von Haber und McNabb [6] ein konzeptioneller Visualisierungsprozess beschrieben, der drei große Transformationen zwischen den Daten vollzieht, um die Visualisierung zu ermöglichen. Die Transformationen konvertieren Rohdaten zu einem darstellbaren Bild, ohne die Integrität der Informationen zu verlieren.

Das Modell besteht aus folgenden vier Komponenten und drei Transformationen, wie in Abbildung 17 dargestellt:

- Komponenten
  - Rohdaten: die Quelle der Grafik.
  - Fokus-Daten: ausgewählte Daten.
  - Geometrischen Daten: visuelle Variablen, die die Eigenschaften einer Grafik beschreiben.
  - Bilddaten: das Ergebnis.

- Transformationen
  - Filter: Verarbeitung von Daten, damit die anderen Komponenten des Modells die Daten bearbeiten können
  - Mapping: Transformierung von Datentabellen in visuellen Variablen
  - Rendern: Erzeugung von Bildern aus den geometrischen- und visuellen Attributen.

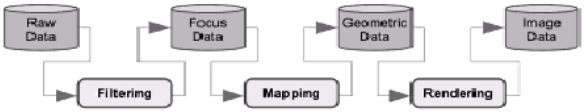


Abbildung 17: The pipeline model [6]

„The data state model “

Im Jahr 1998 wurde von Chi [7] das Daten-Zustandsmodell vorgestellt, das eine spezialisierte, modifizierte Version des „Information visualization pipeline“ Modells ist. Beide Modelle besitzen ähnliche Komponenten, die relevant für die Modellierung von Visualisierungssystemen sind und können als Basis zum Verständnis des Visualisierungsprozesses genommen werden.

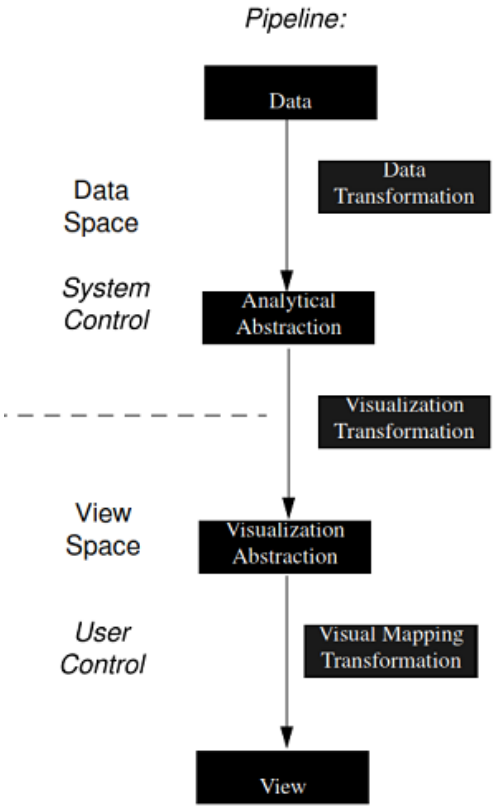


Abbildung 18: Daten -Zustandsmodell von Chi [7]

Das Modell besteht aus sieben Komponenten, die in zwei Gruppen, Transformationen und Stufen, unterteilt werden. In Abbildung 18 sind diese Komponenten des Modells dargestellt.

- Daten: die Quelle der Grafik (Stufe)
- Datentransformation: Erzeugung eines Datensatzes aus einer Menge von Rohdaten
- Analytische Abstraktion: indizierte Sätze von Daten (Stufe)
- Visualisierungstransformation: Zuordnung des Datensatzes zu einem Visualisierungssatz
- Visualisierungsabstraktion: Abstraktionsmodell der Visualisierung (Stufe)
- Visuelle Mapping-Transformation: das Hervorheben von relevanten Elementen
- Ansicht: das Ergebnis (Stufe)

„Information Visualization Data Stage Reference Model“

Im Jahr 2000 wurde von Chi [12] das Modell erweitert und als Informationsvisualisierungs-Daten-Zustandsmodell umbenannt (siehe Abbildung 24). Mit Hilfe dieses Modells klassifiziert Chi sechsunddreißig Visualisierungstechniken in unterschiedliche Bereiche wie wissenschaftliche Visualisierung, geographisch basierte Visualisierung oder die Visualisierung von Bäumen.

„Readings in information visualization: using vision to think“

Im Jahr 1999 wurde von Card et al. [8] das Buch „Readings in information visualization: using vision to think“ veröffentlicht. Im diesem Buch wurde das Referenzmodell für die Visualisierung von Shneiderman beschrieben (Shneiderman ist einer der drei Autoren des Buches).

Das Referenzmodell besteht auch aus drei Transformationen und vier Stufen, aber die Begriffe und die Transformationen sind nicht zu 100 Prozent gleich. Die sieben Komponenten sind folgende:

- Stufen
  1. Rohdaten: die Quelle der Grafik
  2. Datentabellen: Kombination von relationalen Daten und Metadaten (Die Metadaten beschreiben die Beziehungen zwischen den Daten)
  3. Visuelle Strukturen: Kodierung der Informationen durch die Beschreibung von grafischen Eigenschaften
  4. Ansichten: die Ergebnisse.
- Transformationen
  1. Datentransformation: Umwandlung von Rohdaten in Datentabellen

2. Visuelles Mapping: Transformierung von Datentabellen in visuelle Strukturen
3. View-Transformation: Erzeugung der Ansichten durch das Spezifizieren von grafischen Eigenschaften wie Position, Skalierung und Ausschnitte.

Im Gegensatz zum Daten-Zustandsmodell von Chi beschreibt das Referenzmodell, wie der Visualisierungsprozess von der Aufgabe des Benutzers ausgelöst und kontrolliert wird. Die Interaktion des Benutzers ermöglicht beispielsweise Datenbereiche zu begrenzen oder die Art der Transformation zu ändern.

Abbildung 19 zeigt die oben beschriebenen Komponenten und die Interaktion des Benutzers.

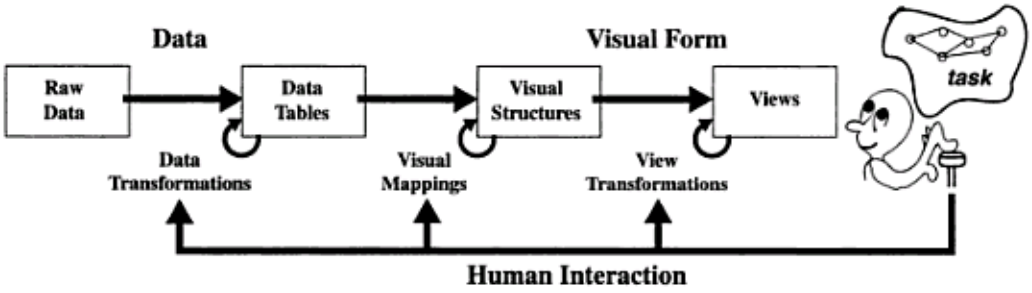


Abbildung 19: Referenzmodell für die Visualisierung von Shneiderman [8]

„The Grammar of Graphics“ - GOG

Im selben Jahr (1999) wurde von Wilkinson [64] ein anderes Modell aus der Perspektive eines Software-Systems vorgeschlagen. Das Modell wurde im Buch „The Grammar of Graphics“ vorgestellt. Das Buch basiert auf einer mathematischen und statistischen Perspektive der Visualisierung.

Wilkinson unterscheidet zwischen zwei Arten von Komponenten: einerseits die mathematischen Komponenten, anderseits die Komponenten, die mit Ästhetik der Grafik zu tun haben. Das erste Modell (1999) besteht aus elf Komponenten. Diese sind folgende:

- Daten: die Quelle der Grafik
- Daten-View: Erzeugung von Datensätzen aus einer Menge von Rohdaten
- Dataset: Index-Sätze von Daten
- Verweise: Mechanismus zur Unterstützung der Objektlokalisierung
- Variable-Mapping: Zuordnen eines Datensatzes zu einem variablen Satz
- Algebra: Sammlung von Operatoren und Regeln
- Variablen-Satz: Satz von Variablen (diskrete oder kontinuierliche Variablen)
- Grapher: Erzeugung eines Diagramms aus einer bestimmten grafischen Funktion

- Graph: Ein Graph ist eine Teilmenge von Datensätzen
- Lage: ist ein Schema für die Lokalisierung eines Punkts in einem gegeben Raum
- Ästhetik: ästhetische Eigenschaften eines Graphen
- Grafik: Ergebnis eines Graphen unter einer oder mehreren Funktionen

Sechs Jahre später wurde die zweite Auflage des Buches [9] veröffentlicht. Dort wurde die Definition des ersten Modells geändert. Das neue GOG-Modell besteht aus sieben Komponenten, die orthogonal miteinander verbunden sind. Abbildung 20 zeigt die Elemente des neuen Modells.

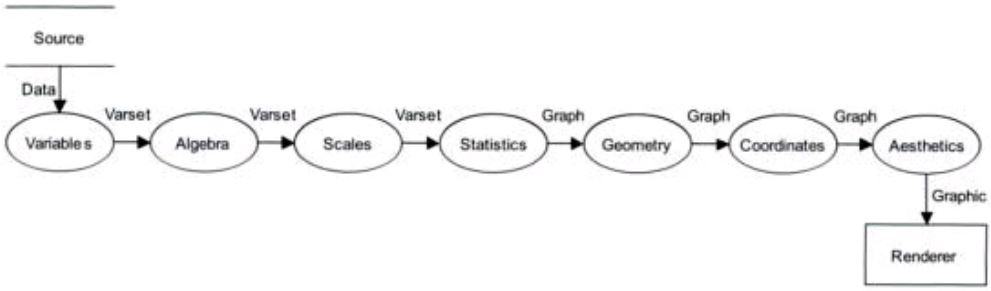


Abbildung 20: Wilkinson - Aspekte eines Visualisierungssystems [9]

3.3.2 VISO (Visualization Ontology)

Im Jahr 2011 wurde von Voigt und Polowinski [65] eine semantische Ontologie zur Beschreibung von Visualisierungssystemen vorgeschlagen. VISO ist eine modulare Ontologie zur Formalisierung der Visualisierung, die ein RDF-S/OWL Vokabular bietet, um die Komponenten der Visualisierung und Datenquellen zu beschreiben. Die Ontologie enthält Faktenwissen der Visualisierungsdomäne und soll als ein semantisches Framework zum Speichern von Kontextinformationen dienen[66].

VISO besteht aus sieben Modulen (Daten, Grafik, Aktivität, Benutzer, System, Domäne und Fakten), die verschiedene Komponenten der Datenvisualisierungsdomäne darstellen. Abbildung 21 zeigt diese sieben Komponenten.

VISO Semantics-based Information Visualization Workflow

Voigt et al. [11] schlagen einen semantisch basierten Visualisierungsprozess vor, indem alle Beteiligte interagieren, damit eine mögliche Darstellung einfacher vorgeschlagen werden kann. Der gesamte Prozess basiert auf der Ontologie VISO, die Konzepte und Beziehungen zwischen den diversen Faktoren der Visualisierung bietet. Abbildung 22 zeigt, wie die Interaktion zwischen den Komponenten durch die Verwendung von VISO funktioniert.

Der Entwurf des Visualisierungsprozesses orientiert sich an der Interaktion mit den normalen Benutzern und besteht aus fünf Stufen:

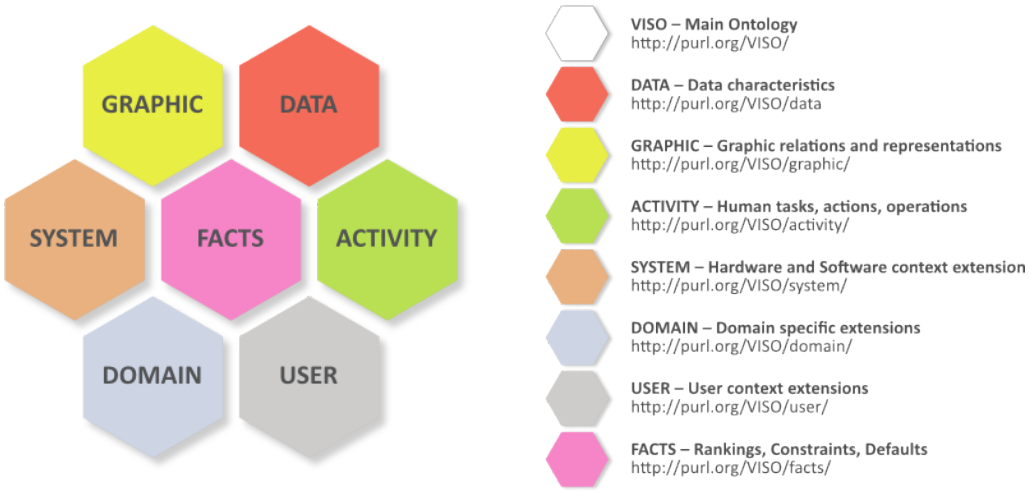


Abbildung 21: VISO Ontologie [10]

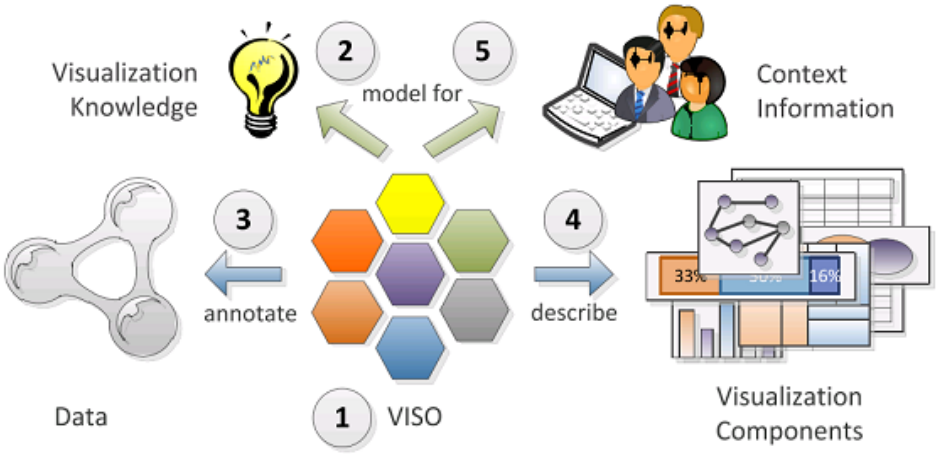


Abbildung 22: VISO Semantics-based Information Visualization Workflow [11]

1. Beschaffung des Datensatzes
2. Überblick über die ausgewählten Daten
3. Auswahl relevanter Datenvariablen und geeigneter Visualisierungskomponenten
4. Konfiguration visueller Komponenten
5. Interaktion mit der Visualisierung

Zur Formalisierung des Visualisierungswissens wurden die Komponenten der Visualisierung in vier Gruppen unterteilt: Daten, Grafik, menschliche Tätigkeiten und die Integration Benutzer-, System- und Domain-Ontologien [65]. Abbildung 23 zeigt die Elemente, die jede Gruppe beinhaltet.

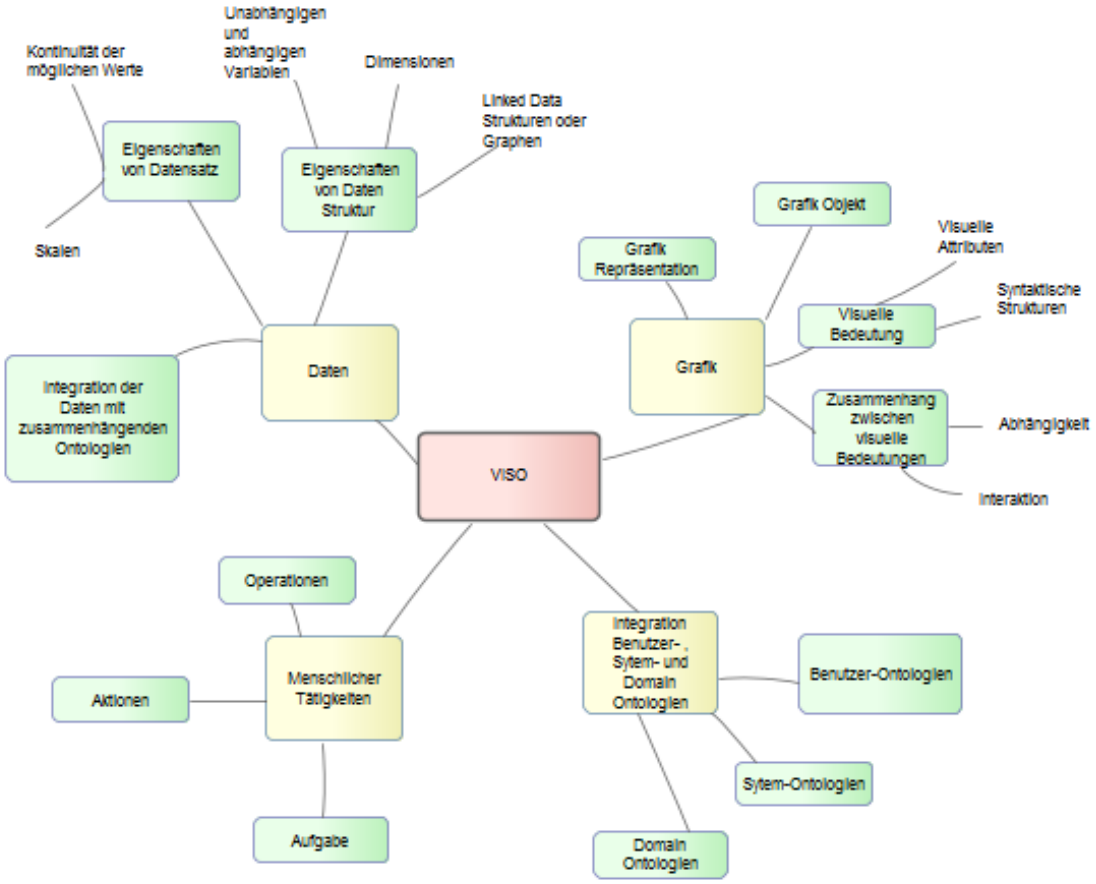


Abbildung 23: Gruppen von Komponenten von VISO

3.3.3 Linked Data Visualization Model

Das „Linked Data Visualization Model“ (LDVM) wurde von Brunetti et al. [13] vorgeschlagen. Das Modell sollte ermöglichen, verschiedene Datensätze mit unterschiedlichen Visualisierungen in einer dynamischen Weise zu verbinden. Um flexibel zu sein, unterteilt das LDVM den Visualisierungsprozess in Schritte. Jeder Schritt besteht aus einer deklarativ definierten Reihe von Transformationen.

Dieses Modell basiert auf dem Data-Stage-Reference-Modell von Chi [12] (siehe Abschnitt 3.3.1) und einer generischen Bibliothek zur Extraktion und Visualisierung von Daten. Die folgenden Abbildungen zeigen das Modell von Chi (links) und das LDVM (rechts). Beide Modelle sehen sehr ähnlich aus und sind kaum zu unterscheiden.

Die Komponenten sind eine Anpassung an das Modell von Chi für LOD. Brunetti et al. [13] beschreib die Komponenten wie folgt:

- RDF-Daten: Rohdaten (Taxonomien, Vokabulare, Ontologien, etc..)
- Analytische Extraktion: Erhalten der Datenextraktionen aus Rohdaten, z.B.

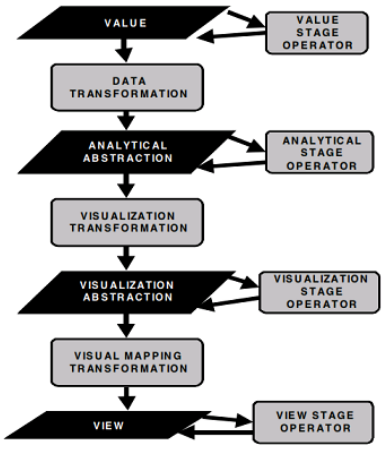


Abbildung 24: Information Visualization Data Stage Reference Model von Chi [12]

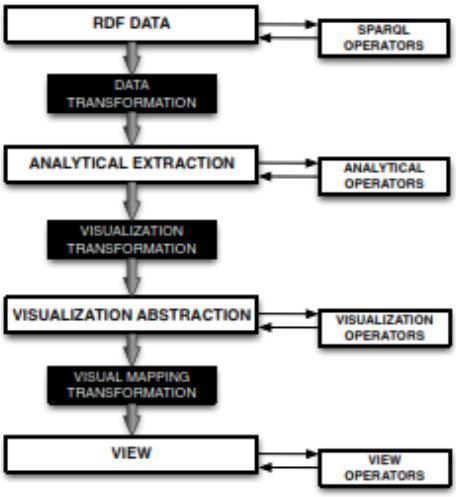


Abbildung 25: Linked Data Visualization Model [13]

Berechnung aggregierter Werte

- Visualisierungsabstraktion: Sichtbare Informationen mit einer Visualisierungstechnik
- Ansicht: das Ergebnis des Visualisierungsprozesses
- Datentransformation: verwandelt Rohdaten mit SPARQL Query Templates
- Visualisierungstransformation: Bearbeitung der Daten, damit sie geeignet für die Visualisierung sind
- Visuelle Mapping-Transformation: Visuelle Darstellung der Visualisierungsabstraktionen, um eine zu erhalten
- SPARQL Operatoren: SPARQL-Funktionen (SELECT, FILTER, COUNT, GROUP BY, etc.)
- Analytische Operatoren: Funktionen, angewandt auf die Datenextraktion
- Visualisierungsoperatoren: visuelle Variablen (Größen, Farben, etc.)
- Ansichtsoperatoren: Angewandt auf die Ansichtsaktionen, z. B. drehen, skalieren, zoomen, etc.

Basierend auf ihrem Modell wurde ein Prototyp „LODVisualization“ implementiert, die bei App Engine[24] läuft.

Im Abschnitt 5.1.1 wird ein Modell vorgeschlagen, das zur Visualisierung von Daten verwendet werden kann. Das Modell basiert auf den Modellen von Chi [12] und Brunetti [13]. Das Modell fügt eine neu Komponente hinzu, damit das Ziel der Visualisierung einfacher übertragen wird.

[24] http://lodvisualization.appspot.com

3.4 Automatischen Zuordnung und Erzeugung

Einige Autoren versuchen durch verschiedene Annäherungen grafische Darstellungen von Daten automatisch zu generieren. Voigt et al. [66] unterscheiden zwischen drei verschiedenen Ansätzen der automatischen Generierung von Visualisierungen. Diese sind daten-, task- und interaktionsgetriebene Ansätze.

- **Datengetriebene Ansätze** analysieren die Meta-Modelle der Daten, um potenzielle, entsprechende Visualisierungen zu erzeugen. Zu dieser Gruppe gehört der Ansatz von Gilson et al. [67], der die automatische Generierung von Visualisierungen durch verfügbare, domänenspezifische Daten vorschlägt. Es werden drei verschiede Ontologien verwendet, um die Daten zuzuordnen. Die erste ist eine Domäne-Ontologie eines spezifischen Bereichs, die zweite Ontologie beschreibt visuelle Darstellungen und die dritte Ontologie speichert Informationen über die Beziehungen der anderen Ontologien. Die Quelle der Daten ist nicht LOD, sondern eine Webseite, die zu dieser spezifischen Domäne gehört.
- **Taskgetriebene Ansätze** basieren auf der Tatsache, dass die Wirksamkeit einer Visualisierung von der Genauigkeit der Beschreibung ihrer Aufgabe abhängig ist.
- **Interaktionsgetriebene Ansätze** nutzen die früheren und/oder aktuellen Interaktionen des Benutzers, um neue Visualisierungen zu generieren, die die Exploration der Daten unterstützen.

Visualization recommendation algorithm

Der Visualisierungs-Empfehlungs-Algorithmus von Voigt et al. [66] schlägt eine Liste von möglichen Visualisierungen für ausgewählte Daten vor. Er besteht aus zwei Komponenten:

- „Discovery of Mappings“: Der Algorithmus erzeugt eine Liste von ausgewählten Daten und Interaktionen der Benutzer. Die verwendbaren Visualisierungen werden identifiziert und die nicht verwendbaren ausgeschlossen.
- „Ranking of Mappings“: Der Algorithmus sortiert die zugeordneten Visualisierungen, die aus dem früheren Entdeckungsschritt generiert wurden, nach einigen vordefinierten Regeln.

3.5 Anforderungen der Visualisierung von LOD

Dadzie und Rowe [68] veröffentlichten im Jahr 2011 eine Untersuchung über die Annäherungen zur Visualisierung von LOD. Dort wurden verschiedene Herausforderungen, die die Verwendung von LOD hat, aufgezeigt. Sie beschrieben einige grundlegenden Anforderungen, die die Implementierung von Anwendungen zur Visualisierung von Linked Data beachten sollte.

3.5.1 Arten von Konsumenten von LOD

Sie definieren hauptsächlich zwei Arten von Benutzer von LOD: die Laien und die technischen Benutzer.

- Die **Laien** sind Benutzer, die nicht notwendigerweise verstehen, RDF und andere Technologien des semantischen Webs. Solche Benutzer haben Computerkenntnisse und sind in der Lage, Informationen mithilfe einer Suchmaschine zu finden.
- Die **technischen Benutzer** sind Menschen mit Erfahrung im semantischen Web und seiner Technologie, z.B. sind sie in der Lage, eine Ontologie zu verstehen und zu interpretieren.

Außerdem definieren sie eine dritte Gruppen von Benutzer, die entweder zu den Laien (normaler Nutzer) oder den technischen Benutzern gehören. Sie sind die **Domäne-Experte**. Abhängig von dieser Unterteilung sieht man beispielsweise, ob das Ergebnis einer Anfrage für einen technischen Benutzer RDF oder JSON geeignet ist, während für einen normalen Nutzer eine Tabelle oder einer Visualisierung angenehmer und verständlicher wäre.

3.5.2 Entwurfsrichtlinien

Die erste Gruppe von allgemeinen Anforderungen ist die Unterstützung des Grundprinzips der Informationssuche [48] von Shneiderman (siehe Abschnitt 5.1.2).

Die zweite Gruppe besteht aus Aufgaben in Bezug auf die LD :

- Verarbeitung multidimensionale Daten
- Identifizieren und Hervorheben von Beziehungen zwischen den Daten
- Daten-Extraktion

Die dritte Gruppe besteht aus Aufgaben, die technische Benutzer typischerweise durchführen, wie die visuelle Darstellung und die Analyse der Informationen:

- Navigation durch die Linked Data - Strukturen
- Exploration der Daten:
  - zum Erhalt eines Verständnisses von Struktur und Inhalt
  - zur Identifizierung der Verbindungen innerhalb und zwischen den Datensätzen
  - zur Identifizierung von Fehlern in Inhalt und Syntax
- Abfragen, auch mit formaler Abfrage-Syntax
- Veröffentlichung

- Daten-Extraktion für die Wiederverwendung bei anderen Anwendungen (ohne Änderungen des Formats)

Die letzte Gruppe besteht aus Aufgaben, die normale Benutzer ausführen, ohne die Vorteile (Verbindungen, Struktur, Bedeutung, etc.) des semantischen Webs zu verlieren:

- Intuitive Navigation durch die großen, komplexen und multidimensionalen Daten
- Unterstützung von Abfragen für normale Benutzer
- Darstellung der Ergebnisse der Analyse für unterschiedliche Zielgruppen

Die dritte und vierte Gruppe von Anforderungen sind sehr ähnlich. Der Unterschied besteht in der Perspektive des Benutzers.

3.6 Werkzeuge zur Visualisierung von LOD im Browser

Die Visualisierung umfasst viele Bereiche. Deshalb ist es normal, dass viele Lösungen in den unterschiedlichen Bereichen entwickelt werden. In diesem Abschnitt werden einige bestehende, technische Lösungen kurz beschrieben. Die Lösungen wurden nach Funktionalität sortiert.

3.6.1 LOD Strukturen und Abfragen

LOD Browsers

Tabulator [69] und Explorator [70] ermöglichen es, durch die Daten zu navigieren. Dazu nutzen sie Tabellen mit Schlüssel/Wert-Paaren. Beide Entwicklungen wurden vor mehr als vier Jahren durchgeführt und haben einige Kompatibilitäts-Probleme mit den aktuellen Browsers. Tabulator könnte auch zur Kategorie- Visualisierung von SPARQL -Ergebnissen passen, da Tabulator anbieten sollte, die Funktionalität der SPARQL -Ergebnisse in einer Karte oder einem Kalender sichtbar zu machen.

Rhizomer [71] ermöglicht, durch die Daten zu navigieren und gleichzeitig Abfragen durchzuführen. Die Navigation ist einfach und intuitiv und man merkt, dass sich die Anwendung an dem Laien-Benutzer orientiert und dass auf das Grundprinzip der Informationssuche (siehe 5.1.2) in der Implementierung geachtet wurde.

LOD Graphen-Struktur

IsaViz [72], RDF-Gravity [73], Fenfire [74] schlagen eine Navigation durch Graphen vor. Die Anwendungen stellen die RDF-Datensätze als gerichtete Graphen dar. Ressourcen und Literale werden als Knoten gezeichnet und die Eigenschaften als Kanten.

LOD Web-Mashups

DERI Pipes [75] wurde von Yahoo Pipes und Microsoft Popfly inspiriert. Ziel der Anwendung ist eine effektive und einfache Nutzung von RDF-Daten, die im Internet verfügbar sind.

3.6.2 SPARQL Ergebnisse

Eingebettet in HTML

SPARK[25] und Sgvizler[26] [76] sind zwei JavaScript-Entwicklungen, die versuchen, SPARQL-Ergebnisse im HTML-Code zu integrieren. Die Syntax von beiden ist sehr ähnlich und sieht wie das Beispiel unten aus. Beide definieren einen Endpoint, einen Grafik-Typ und die SPARQL-Abfrage

Listing 14: Syntax von SPARK und Sgvizler

```
<div class="spark"
endpoint="http://dbpedia.org/sparql"
format="http://.../piechart"
query="select distinct ?entity ?label ?value
    where {
    ?entity dbpprop:callingCode ?cc .
    ?entity rdfs:label ?label .
    ?entity dbpprop:populationEstimate ?value . }">
```

Karten im Browser

Map4rdf [77], LinkedGeoData [78] und DBpedia-Mobil [79] sind Web-Anwendungen zur Visualisierung von RDF -Datensätzen mit Geoinformationen. Sie zeigen in einer Karte diverse Geoinformationen. DBpedia-Mobil erlaubt außerdem die Information aus DBpedia zu sehen. Dazu wird Marbles verwendet.

LOD Visualisierung

Semasearch[27] wurde entwickelt, um die Funktionen von SemaVis zu demonstrieren. Das SemaVis ist ein Framework zur Visualisierung semantischer Daten unter der Leitung des Fraunhofer Instituts für Graphische Datenverarbeitung (IGD) mit dem Ziel, den Zugang zu Informationen zu vereinfachen. Semasearch stellt die RDF-Datensätze in verschwindenden Formen wie Graphen oder in einer Karte dar.

EWR Daviz [28] ist ein Web-Tool, das von der Europäischen Umweltagentur entwickelt wurde, um interaktiven Visualisierungen von Daten einfach über den Web-Browser zu erstellen. Die Quelle der Daten können CSV/TSV, JSON oder das Ergebnis einer SPARQL-Abfrage sein.

[25] http://code.google.com/p/rdf-spark/ (Stand 14.03.2013)
[26] http://code.google.com/p/sgvizler/ (Stand 19.03.2013)
[27] http://www.semasearch.net/search/ (Stand 05.05.2013)
[28] http://plone.org/products/eea.daviz (Stand 07.05.2013)

4 Ontologie zur Visualisierung von Linked Open Data mit Zeit- und Ortsbezug

Es gibt viele Darstellungen von Daten, und jede hat Ihre Eigenschaften und Ziele. Deswegen ist es wichtig, die Darstellungen nach einigen Kriterien zu klassifizieren, damit der Benutzer, der LOD visualisieren möchte, ein Ergebnis erhält, das seinen Bedürfnissen entspricht.

Für diese Masterarbeit ist es relevant festzulegen, welche Darstellungen nach bestimmten Kriterien zur Verfügung stehen, um Möglichkeiten zur Darstellung von Daten mit Orts- und Zeitbezug herauszufinden. In diesem Kapitel wird eine Ontologie zur Klassifizierung der Visualisierung beschrieben.

Diese Ontologie ist in zwei Teile unterteilt. Einerseits die möglichen Darstellungen (allgemeine Ontologien) und anderseits die Daten mit Zeit- und Ortsbezug (Domänen-Ontologien). Die Ontologie wurde nach der im Abschnitt 2.2.4 beschriebene Methodologie erstellt. Zur Formalisierung wurde RDF-S/OWL als Ontologie-Sprache verwendet.

4.1 Grafik-Ontologie

4.1.1 Feststellung der Domäne, Bereich und Anwendung der Ontologie

Die Grafik-Ontologie soll bei der Identifizierung von Grafiken helfen, die zu einer bestimmten Art von Daten und Zweck passen. Dazu beschreibt die Ontologie Eigenschaften von Daten, die in einer Grafik dargestellt werden können, und die normale Verwendung einer Grafik. Die Ontologie kann im Bereich Informations-Visualisierung verwendet werden.

Die Methodologie schlägt in diesem Schritt vor, Fragen zu stellen, die die Ontologie beantworten soll, damit am Ende des Erstellungsprozesses überprüft werden kann, ob die Ontologie das erwartete Ergebnis bietet.

Es wurden folgende Fragen gestellt:

- Wie können die Daten darstellen werden?
- Wozu kann eine spezifische Darstellung verwendet werden?
- Welchen Unterschied gibt es zwischen einem Punktdiagramm und einem Balkendiagramm?
- Wie viel Merkmale können dargestellt werden?
- Sind die Farben brauchbar?
- Kann ich diskrete/stetige Daten darstellen?
- Kann ich das Verhalten der Daten sehen?

- Wie viel Dimensionen kann eine Grafik darstellen?
- Müssen alle Dimensionen die gleiche Skala haben?
- Passen quantitativen Daten in ein Wasserfalldiagram?
- Wie viele Items können in einem Balkendiagramm dargestellt werden? (Maximal, Minimal)

Die Frage „Welchen Unterschied gibt es zwischen einem Punktdiagramm und einem Balkendiagramm?“ wird hier nicht berücksichtigt, weil sie sich auf die Beschreibung der Grafiken bezieht, und nicht zu der Domäne der Ontologie gehört.

4.1.2 Wiederverwendung der vorhandenen Ontologie

In Abschnitt 3.3.2 wird die Ontologie VISO beschrieben, die die Komponenten des Visualisierungsprozesses beschreibt. Diese Ontologie verfügt zwar über Begriffe, die mit der Visualisierung der Information zu tun haben, aber sie werden nicht verwendet, weil die Anwendung von VISO mit dem Zweck dieser Ontologie nicht übereinstimmt.

Folgende sind die konzeptionellen Unterschiede, die diese Ontologie mit VISO hat:

- VISO unterscheidet Nominal-, Ordinal-, quantitative und unstrukturierte Skalen. Diese Ontologie wird in Nominal-, Ordinal-, Intervall- und Ratioskala unterschieden.
- VISO definiert die Dimensionen wie Rollen der Variablen. Für diese Grafik-Ontologie ist eine Dimension des Raums oder der Form, mit denen die entsprechenden Werte dargestellt werden können.
- VISO definiert keinen Zweck für die Grafiken.
- VISO versucht den Visualisierungsprozess zu definieren. Das Ziel dieser Ontologie ist nur die Identifizierung von Grafiken, die zu einer bestimmten Art von Daten und Zweck passen.

4.1.3 Identifikation relevanter Begriffen

- **Kontinuität der Daten:** Die Werte der Daten können entweder diskret oder ständig sein. Beide Begriffe wurden im Abschnitt 2.1.1 definiert.
- **Daten Typen:** Die Daten können Qualitativ oder Quantitativ sein (siehe Abschnitt 2.1.1).
- **Skala:** Es gibt im Prinzip vier Skalen; Nominalskala, Ordinalskala, Intervallskala und Ratioskala, wie im Abschnitt 2.1.1 beschrieben wurde.

- **Zweck:** In Abschnitt 2.4.3 wurde erläutert, wieso die Auswahl des Ziels einer Visualisierung wichtig ist. Eine gute Möglichkeit, um die Auswahl zu vereinfachen, ist die Frage nach dem Zweck der Visulisierung, die dargestellt werden möchte. Dieser Begriff bezieht sich auf den Zweck der Visualisierung.

  Eine Abstraktion der Aufgabe, die die Beziehung zwischen den Daten zeigt, ist notwendig, um zuordnen zu können. Es wurden folgende Zwecke als Basis festgelegt:

  - Vergleich
  - Verteilung
  - Tendenz
  - Zusammensetzung (Struktur)
  - Beziehung
  - Muster (Verhaltensmuster)
  - Ortsbestimmung

- **Dimension:** Eine Dimension ist der Raum oder die Form, mit denen die entsprechende Werte dargestellt werden. Folgende Beispiele sollen den Begriff erläutern:

  - Balkendiagramm: Ein Balkendiagramm stellt zwei Dimensionen dar. Die Dimensionen sind die x-Achse und y-Achse.
  - Blasendiagramm: Ein Blasendiagramm stellt drei Dimensionen dar. Die Dimensionen sind die x-Achse, die y-Achse und entweder Farbe oder Größe.

  Die Anzahl der Dimensionen, die dargestellt werden können, ist vom Diagramm abhängig. Mögliche Werte für eine Dimension sind x-Achse, y-Achse, z-Achse, Farbe, Größe und geografische Position.

- **Anzahl der Items, die idealerweise dargestellt werden sollen:** Es gibt eine bestimmte Anzahl von Items, die dargestellt werden können, um das Diagramm für jeden Benutzer verständlich zu machen. Diese Anzahl ist abhängig vom Darstellungstyp und begrenzt. Sehr häufig gilt die Regel: je mehr Items es gibt, desto weniger verständlich ist die Darstellung für die Nutzer.

- **Zeitraum:** Als Zeitraum oder Periode wird eine mathematische Beziehung zwischen der Anzahl der Items, die dargestellt werden sollen, und den Zeiträumen, die bewertet werden können.

Ein Beispiel

Um die oben genannten Begriffe zu erläutern, wird das Balkendiagramm verwendet.

Für ein Balkendiagramm ist der Zweck „Der Vergleich von Daten“ und hat zwei Dimensionen. Eine wird in der x-Achse dargestellt und die andere in der y-Achse.

In der x-Achse sind die Daten quantitativ und entweder diskret mit Ordinalskala oder kontinuierlich mit Intervallskala. In der y-Achse sind die Daten diskret, qualitativ und ihre Skala ist die Nominalskala. Die Mindestanzahl der Items ist zwei, weil der Zweck Vergleich ist.

Die Höchstzahl von Items ist zwanzig, damit das Diagramm für jeden Nutzer verständlich wird. Der Zeitraum ist nur eins. Wenn es ein gruppiertes Balkendiagramm wäre, könnte es mehr als einen Zeitraum geben. Die Anzahl der Periode wird mit der folgenden Formel berechnet:

(Höchstzahl von Items) / (Anzahl von Items per Periode)

Beispiel:

Höchstzahl der Items = 20 / Anzahl der Items per Periode = 5

folglich: Periode = 4.

4.1.4 Erstellung der Konzepthierarchie

Klassen

Einer der schwierigsten Teile in der Konstruktion einer Ontologie ist zu wissen, wie weit sie modelliert werden soll. Ein Beispiel ist die Klasse Zweck.

Einerseits kann man denken, dass alle möglichen Werte eine Unterklasse bilden können, weil der Vergleich von Autos anders ist als der Vergleich von Computern. Wenn das angenommen wird, müssen alle oben genannten Zwecke als Unterklassen modelliert werden. Andererseits sind beide Vergleiche im Wesentlichen gleich, da beide den gleichen Zweck erfüllen. In diesem Fall modelliert man nur Zweck als Klasse und die Werte als Ausprägungen:

- Für die Domäne der Ontologie sind die Unterschiede zwischen den verschiedenen Arten von Vergleichen, die existieren können, unwichtig. Deswegen wird nur eine Klasse „Zweck“ definiert, die Ausprägungen wie Vergleich, Verteilung, Tendenz, Zusammensetzung, Beziehung, Muster und Ortsbestimmung haben kann.
- Skala ist eine Klasse, die vier Unterklassen hat. Diese sind Nominalskala, Ordinalskala, Intervallskala und Ratioskala.
- Daten-Typen ist eine Klasse, die nur zwei Ausprägungen (Qualitativ, Quantitativ) haben darf.
- Daten und Dimension sind Klassen. X-Achse und Farbe sind Unterklassen von Dimension, wenn sie eine Dimension darstellen.

- Die Klasse Grafik ist eine Klasse, die folgende Attribute hat: Zweck, Dimension, Anzahl der Items und Periode.
- Kontinuität der Daten ist die andere Klasse der Ontologie, die auch nur zwei Ausprägungen (Diskret, Kontinuierlich) haben darf.

4.1.5 Definition der Beziehungen

Die folgenden Beziehungen wurden identifiziert:

- Zweck, Skala, Daten -Typen, Kontinuität der Daten, Daten, Dimension und Grafik sind Dinge.
- Die Skala hat eine Kontinuität. Diskret und Kontinuierlich sind eine Kontinuität der Daten.
- Nominalskala, Ordinalskala, Intervallskala und Ratioskala sind eine Skala.
- Qualitativ ist ein Daten -Typen, Quantitativ ist ein Daten -Typen.
- Daten haben Daten -Typen, Daten haben eine Kontinuität der Daten, Daten haben eine Skala.
- Dimension hat Daten, x-Achse ist eine Dimension, y-Achse ist eine Dimension, Farbe ist eine Dimension, Größe ist eine Dimension, z-Achse ist eine Dimension.
- Eine Grafik hat ein oder mehrere Zweck(e), eine Grafik hat ein oder mehrere Dimension(en), eine Grafik hat eine Anzahl von Items, eine Grafik hat eine Periode.

4.1.6 Formalisierung von Klassen

Einer der Gründe für die Erstellung von Ontologien ist die Fähigkeit, Komponenten wiederzuverwenden. Deswegen werden die Komponenten der Ontologie im Prozess der Formalisierung auf Englisch beschrieben. In diesem Kapitel werden die Komponenten grafisch gezeigt und nur einige Komponenten als Beispiel in RDF-S/OWL beschrieben, da die Syntax viel Platz benötigt.

Die Kontinuität der Daten und ihre Ausprägungen werden in Turtle Syntax wie folgt geschrieben.

```
1
2 //Turtle
3 DataContinuity
4       rdf:type owl:Class ;
5       rdfs:label "Kontinuität der Daten"@de ,
6                  "DataContinuity"@en ;
7       rdfs:subClassOf owl:Thing ;
8       owl:oneOf (Continuous Discrete) .
```

```
9
10 Continuous
11       rdf:type DataContinuity ;
12       rdfs:label "Ständig"@de , "Continuous"@en ;
13       owl:differentFrom Discrete .
14
15 Discrete
16       rdf:type DataContinuity ;
17       rdfs:label "Diskret"@de , "Discrete"@en .
```

Der Code zeigt, dass Kontinuität der Daten (DataContinuity) eine Klasse, die Unterklasse von Ding (owl: Thing) ist. Die Klasse besteht aus zwei möglichen Ausprägungen (Kontinuierlich und Diskret). Kontinuierlich (Continuous) ist eine Ausprägung, die anders als Diskret (Discrete) ist, und Diskret ist eine Ausprägung, die anders als Kontinuierlich ist.

Abbildung 26 zeigt die Klasse DataContinuity und ihre Ausprägungen.

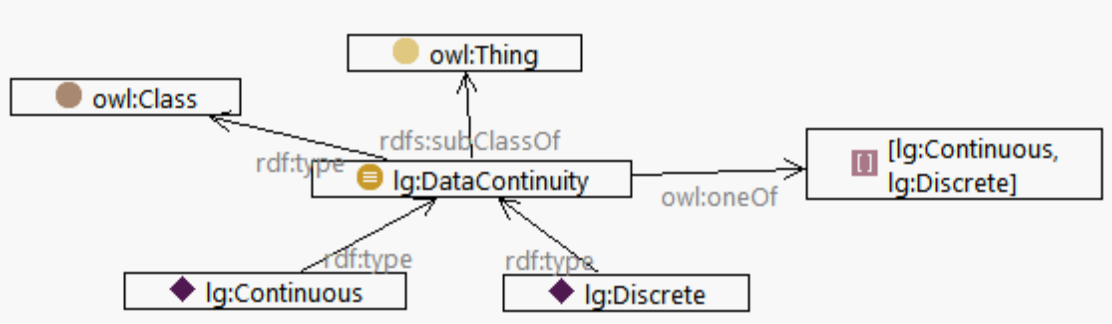


Abbildung 26: OWL Klasse Kontinuität der Daten

Abbildung 27 zeigt die Klasse Zweck (Purpose) und ihre Ausprägungen Vergleich, Verteilung, Tendenz, Zusammensetzung, Beziehung, Muster und Ortsbestimmung.

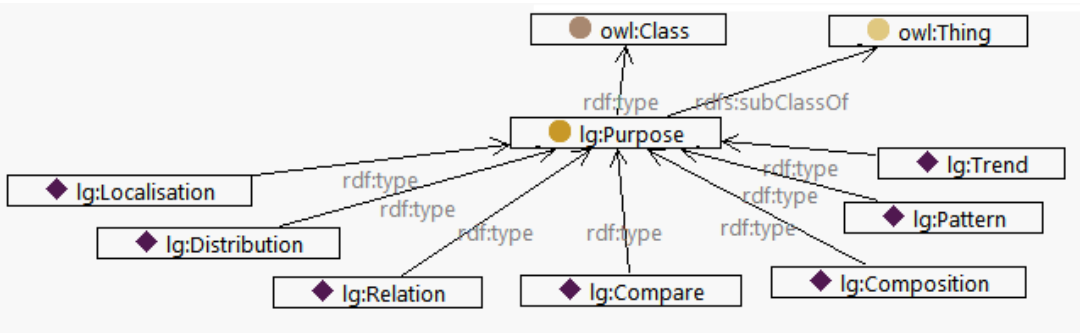


Abbildung 27: OWL Klasse Zweck (Purpose)

Abbildung 28 zeigt die Klasse Dimension und ihre Unterklassen.

Die Klasse Grafik (Graphic) wird in Turtle Syntax wie folgt geschrieben.

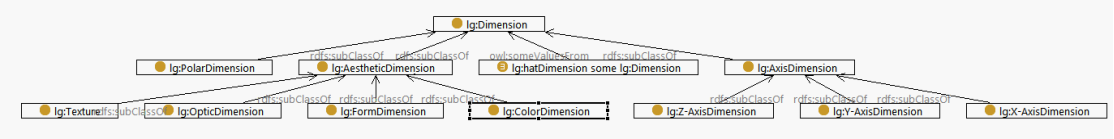


Abbildung 28: OWL Klasse Dimension und Unterklassen

```
Graphic
      rdf:type owl:Class ;
      rdfs:label "Grafik"@de , "Graphic"@en ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:onProperty hatPurpose ;
                owl:someValuesFrom Purpose
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:minCardinality
                   "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hatDimention
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:onProperty hatDimention ;
                owl:someValuesFrom Dimension
              ] ;
      rdfs:subClassOf
              [ rdf:type owl:Restriction ;
                owl:minCardinality
                   "1"^^xsd:nonNegativeInteger ;
                owl:onProperty hatPurpose
              ] .
maxRecomendedAmountItems
      rdf:type owl:DatatypeProperty ;
      rdfs:domain Graphic ;
      rdfs:label "Höchstzahl der Items von Items"@de ,
                "Max recomended amount items"@en ;
      rdfs:range xsd:integer .
minAmountItems
      rdf:type owl:DatatypeProperty ;
      rdfs:domain Graphic ;
      rdfs:label "Mindestanzahl"@de ,
                  "Min amount items"@en ;
      rdfs:range xsd:integer .
```

Der Code zeigt, dass eine Grafik (Graphic) eine Klasse ist, die mindestens einen Zweck (Purpose) hat, die mindestens eine Dimension hat. Eine Grafik hat eine Höchstzahl von Items und eine Mindestanzahl von Items.

Abbildung 29 zeigt die Klassenhierarchie von Grafik.

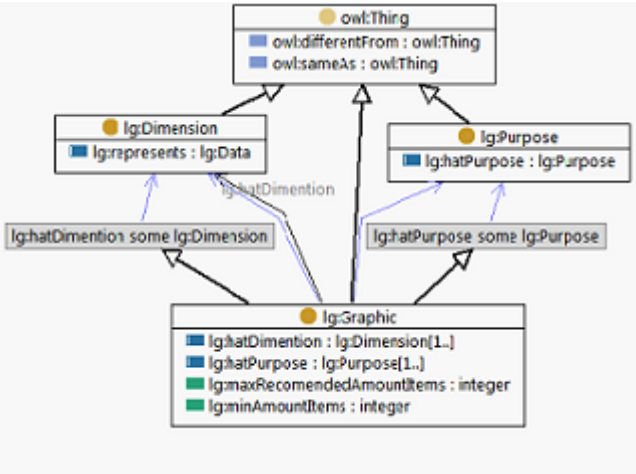


Abbildung 29: OWL Klassenhierarchie von Grafik (Graphic)

Zuletzt zeigt die Abbildung 31 die Klassenhierarchie der gesamten Ontologie.

4.1.7 Beispiel

Im Abschnitt 4.1.3 wird ein Balkendiagramm als Beispiel verwendet, um die Begriffe zu erläutern. Nun wird mit diesem Diagramm eine Ausprägung der Klasse Grafik gezeigt. Die Ausprägung wird in Turtle Syntax wie folgt geschrieben.

```
1
2 BarChart
3       a        Graphic ;
4       rdfs:label "Bar Chart"@en , "Balkendiagramm"@de ;
5       hatDimension X-AxisDimension-barChart ,
           Y-AxisDimension-QualitativeNominalData ;
6       hatPurpose Compare ;
7       maxRecomendedAmountItems 25 ;
8       minAmountItems 2 ;
9       =  <http://de.dbpedia.org/page/Balkendiagramm> .
```

Die Abbildung 30 stellt die Ausprägung Balkendiagramm dar und ihre Hierarchie in der Ontologie.

4.2 GeoZeit-Ontologie

Die zweite Ontologie dient LOD mit Zeit- und Ortsbezug und Ihrer entsprechenden Grafik zur Klassifizierung. Sie wird mit der gleichen Methode der ersten Ontologie (siehe Abschnitt 2.2.4) erstellt.

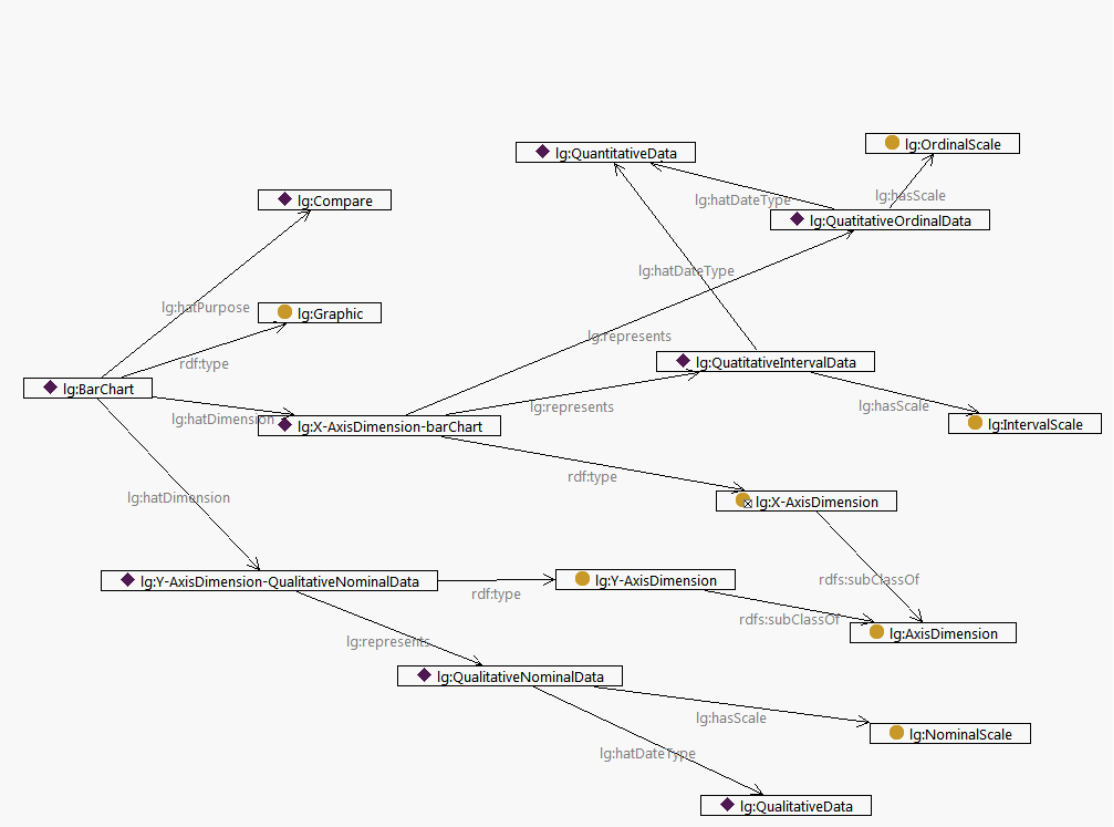


Abbildung 30: OWL Ausprägung - Balkendiagramm

4.2.1 Feststellung der Domäne, des Bereichs und die Anwendung der Ontologie

Die Ontologie beschreibt Eigenschaften der LOD mit Zeit- und Ortsbezug und ihre Darstellung und ermöglicht die Identifizierung von Grafiken, die zu Daten mit Zeit- und Ortsbezug passen.

- Fragen, die die Ontologie beantworten soll, sind folgende:
  - Wie könnten die Daten dargestellt werden?
  - Können zeitabhängige Daten dargestellt werden?
  - Ist die Zeit immer kontinuierlich?
  - Wie viel Merkmale können dargestellt werden?
  - Wie viele Items (Elemente) können in einer Hitzekarte dargestellt werden? (Maximal, Minimal)
  - Kann ich ein Flächendiagramm mit einer Zeitleiste kombinieren?
  - Wie stellt man Änderungen in der Position fest?
  - Welche Arten von räumlichen Zustandsänderungen unterscheidet man?

– Wie kann man tausend Punkte in einer Karte darstellen?

4.2.2 Wiederverwendung der vorhandenen Ontologie

Es werden drei Ontologien als Basis wiederverwendet. Die erste ist die Grafik-Ontologie[29], die in Abschnitt 4.1 beschrieben wurde. Die zweite wird als WGS84_pos[30] verwendet, die in Abschnitt 2.5.1 beschrieben wurde. Die dritte ist „OWL Ontology of Time“ (OWL-Time)footnotehttp://www.w3.org/2006/time, die zeitliche Konzepte beschreibt.

4.2.3 Identifikation relevanter Begriffen

- **Form der Zeit:** Müller und Schumann [80] unterscheiden zwischen zwei Formen der Zeit: lineare und zyklische Zeit. Lineare Zeit setzt einen Startpunkt fest und definiert eine lineare Zeitachse, in der sich die Elemente befinden. Allerdings sind viele natürliche Prozesse, wie die Jahreszeiten oder Wochentage, zyklisch.
- **Dauer:** Es gibt im Prinzip zwei Arten von Dauer: diskrete Zeitpunkte und Zeitintervalle. Diskrete Zeitpunkte sind eine Abstraktion der Zeit, vergleichbar mit diskreten Punkten im euklidischen Raum [80]. Diese Punkte haben keine Dauer. Dagegen haben Zeitintervalle per Definition eine Dauer größer als Null.
- **Ordinale vs. kontinuierliche Zeit:** Ordinale Zeiten sind Zeiten, in denen nur der Vergleich „Früher“ und „Später“ Sinn hat. Kontinuierliche Zeit erlaubt die Quantifizierung der Zeitdifferenz zwischen dem Auftreten von Datenelementen.
- **Räumliches Ding:** Jedes Ding, das in einem Ortungssystem angeordnet werden kann.
- **Ort:** Raum, in dem sich ein Objekt befindet.
- **Veränderung des Raums:** Es gibt zwei Arten von Geodaten, abhängig von der räumlichen Zustandsänderung. Diese sind diskrete und kontinuierliche Daten. Diskrete Daten beschreiben unterschiedliche Aktivitäten, die im Verlauf der Aktivität keine Veränderung der Position haben. Beispiele wären die Aktivitäten einer Person am einem Tag: Geld abheben, ins Kino gehen, einkaufen, etc. Kontinuierliche Daten beschreiben die Bewegung eines Objekts wie z.B. eine Autofahrt von der Universität nach Hause.
- **Politische Ordnungen:** Ein Ort ist abhängig von politischen Ordnungen in einer bestimmten Gruppe. Die Gruppe kann beispielsweise eine Region, ein Staat oder ein Kontinent sein.

[29] http://www.linkedgraphics.com/ontology/graphic
[30] http://www.w3.org/2003/01/geo/wgs84_pos

- **Dynamische vs. statische Darstellungen:** Müller und Schumann [80] unterscheiden zwischen zwei Arten von Darstellungen für die Zeitdaten, dynamische und statische Darstellungen. Statische Darstellungen werden nur durch die Interaktion mit dem Benutzer geändert. Dynamische Darstellungen ändern sich automatisch im Laufe der Zeit. Wilkinson [9] unterteilt auch die dynamischen Darstellungen in dynamisch und real-time, aber für die Ontologie wird nur die Unterteilung dynamisch und statisch betrachtet.
- **Zeitdaten:** Müller und Schumann [80] unterscheiden vier Arten von Darstellungen für die Zeitdaten, dynamische und statische Darstellungen. Statische Darstellungen werden nur durch die Interaktion mit dem Benutzer geändert. Dynamische Darstellungen ändern sich automatisch im Laufe der Zeit.
- **Speicherung der Daten:** Die Daten mit Zeitbezug werden normalerweise mit „Date time“ gespeichert, und der Ort mit zwei oder drei Koordinaten des Vokabulars WGS84_pos.

4.2.4 Erstellung der Konzepthierarchie

Klassen

- Form der Zeit ist eine Klasse, die nur zwei Ausprägungen (lineare und zyklische Zeit) haben darf.
- Dauer ist eine Klasse, die auch nur zwei Ausprägungen (diskreter Zeitpunkt und Zeitintervall) haben darf.
- Es wird eine Klasse mit dem Name Zeitskala definiert, die zwei Unterklassen hat. Diese sind Ordinale und Kontinuierliche Zeitskala. Zeitskala ist eine Unterklasse von Skala.
- Es wird eine Klasse mit dem Name Zeitdaten definiert, die vier Unterklassen hat. Diese sind kontinuierliche Zeitdaten, kumulative Zeitdaten, zyklische Zeitdaten und augenblickliche Zeitdaten. Zeitdaten ist eine Unterklasse von Daten.
- Es wird eine Klasse mit dem Name Geodaten definiert, die zwei Unterklassen hat. Diese sind Kontinuierliche Geodaten und Diskrete Geodaten. Geodaten ist eine Unterklasse von Daten.
- Dynamische und statische Darstellungen werden als Unterklassen von Grafik definiert.
- Ort ist eine Klasse.

4.2.5 Definition der Beziehungen

Folgenden Beziehungen wurden identifiziert:

- Zeit, Dauer, Zeitskala, Geodaten, Ort, Zeitdaten, Ordinalzeit und Kontinuierliche Zeit sind Dinge(owl:Thing).
- Ein Ort liegt in einer Region, einem Staat und einem Kontinent.
- Geodaten enthalten Punkte, die räumliche Dinge sind.
- Zeitdaten hat eine Zeitform.
- Kontinuierliche Geodaten haben stetige Kontinuität.

4.2.6 Formalisierung von Klassen

Die Formalisierung von Klassen wurde mit OWL gemacht (Siehe Appendix).

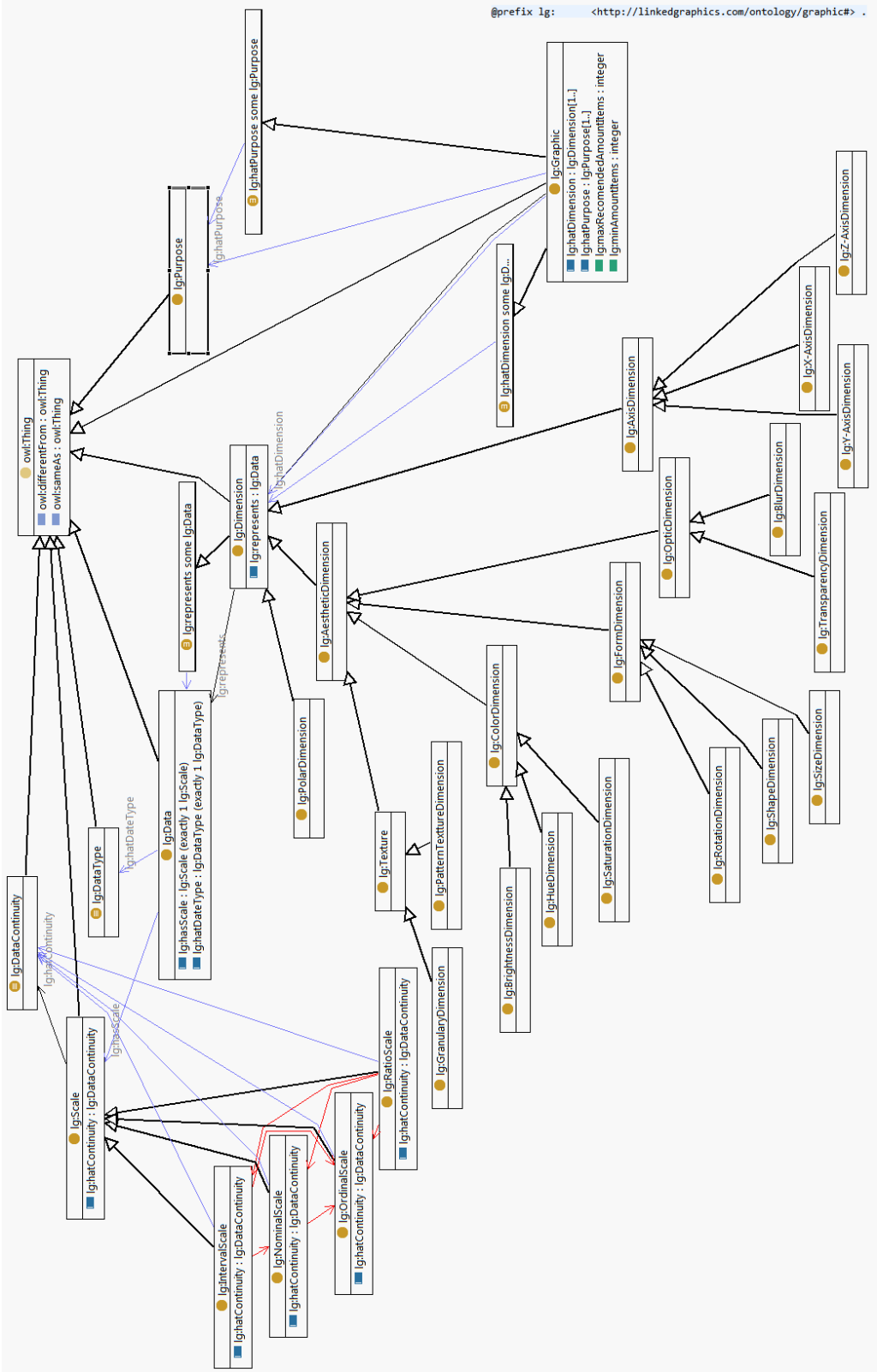


Abbildung 31: OWL Klassenhierarchie von Grafik (Graphic)

5 Entwicklung

Nach der Erstellung der Ontologie wird in diesem Kapitel ein Modell im Abschnitt 5.1.1 zur Visualisierung von Daten beschrieben, die sechs Komponenten beinhaltet. Das Modell hilft die Komponenten der Visualisierung von LOD zu bestimmen. Mit Hilfe dieser Definition werden Anwendungsfälle im Abschnitt 5.3 beschrieben und implementiert, die dienen, die grafische Darstellung im Kapitel 6 auszuwerten. Es werden außerdem die dazu notwendigen Implementierungstechniken kurs beschrieben und deren Einsatz in der Entwicklung erläutert.

5.1 Entwurf

5.1.1 Ein Modell zur Visualisierung von LOD

Basierend auf den Visualisierungsmodellen, die in Abschnitt 3.3 beschrieben wurden, wird ein Modell vorgeschlagen, das zur Visualisierung von Daten verwendet werden kann.

Das Modell schlägt den Austausch der Komponente „Visualisierungsabstraktion“ mit einer neuen Komponente „Visualisierungszweckabstraktion“ vor, die nur diejenigen Visualisierungen zeigt, die zum ausgewählten Zweck passen, damit der Laien-Benutzer[31] die Information, die er mit der Visualisierung kommunizieren möchten, einfacher übertragen kann.

Das Modell besteht aus sechs Komponenten und sieben Transformationen. Die sechs **Komponenten** des Modells sind folgende:

1. **LOD:** die Rohdaten, die im Netz zur Verfügung stehen (Taxonomien, Vokabulare, Ontologien, etc.) und durch SPARQL abfragbar sind
2. **SPARQL-Ergebnis:** das Ergebnis einer SPARQL-Abfrage[32] (SELECT und ASK).
3. **Datensätze:** verarbeitende Daten, die andere Komponenten des Modells bearbeiten können
4. **Visualisierungszweckabstraktion:** Darstellung möglicher Visualisierungen, die zum ausgewählten Zweck passen (siehe Abschnitt 5.1.3)
5. **Visuelle Strukturen:** Kodierung der Informationen durch die Beschreibung grafischer Eigenschaften
6. **View:** das Ergebnis der Visualisierung.

Die Interaktion zwischen den Komponenten des Modells wird mit folgenden sieben **Transformationen** hergestellt:

[31] Laien-Benutzer wurden im Abschnitt 3.5
[32] Siehe http://www.w3.org/TR/2013/REC-sparql11-results-json-20130321/

1. **SPARQL-Abfrage:** Abfrage an einen SPARQL-Endpoint
2. **Daten-Transformation:** Verarbeitung von Daten, damit die anderen Komponenten des Modells diese Daten bearbeiten können
3. **Daten-Auswahl:** Auswahl der relevanten Daten sowie Reduktion, Operationen und Gruppierung von Daten
4. **Auswahl des Zwecks:** Auswahl des Zwecks der Visualisierung
5. **Visuelles Mapping:** Zuordnung der Visualisierung zu der Datenstruktur
6. **View-Transformation:** Erzeugung der Grafik
7. **View-Operationen:** View-Interaktionstechniken

Abbildung 32 zeigt die Interaktion der verschiedenen Komponenten.

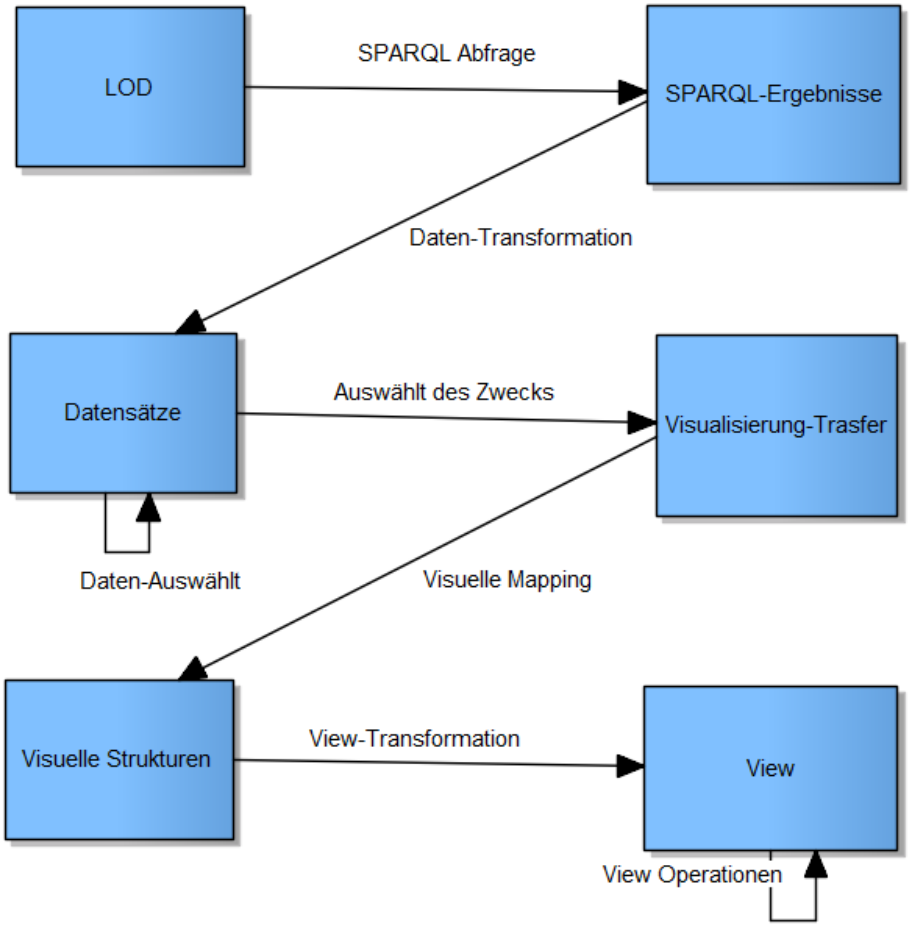


Abbildung 32: Interaktionsdiagramm der Komponenten

5.1.2 Interaktion des Benutzers

Basierend auf dem oben beschriebenen Visualisierungsmodell wird eine mögliche Interaktion mit dem Benutzer vorgeschlagen. Folgende Schritte für die Interaktion des Laien-Benutzers[33] mit der graphischen Nutzeroberfläche sind möglich:

Normaler Ablauf

1. Zuerst schickt der Benutzer eine SPARQL -Abfrage zur Anwendung, die durch die Anwendung weitergeleitet wird. Der Endpunkt gibt das Ergebnis im JSON-Format zurück. Das Ergebnis wird dem Benutzer durch die Anwendung gezeigt.

2. Der Benutzer wählt die Daten aus, die er benötigt. Danach wird durch die Anwendung eine Auswahlliste von Zwecken gezeigt, damit die Anzahl von Visualisierungen kleiner wird.

3. Der Benutzer wählt eine oder mehrere Zwecke aus. Danach werden die entsprechenden Grafiken durch die Anwendung gezeigt.

4. Die Grafik wird von dem Benutzer ausgewählt. Ein Formular wird angezeigt, um die Daten der entsprechenden Grafik-Dimension anzupassen.

5. Die Anwendung führt eine Visual Transformation und Grafikstellung durch.

Optional

4a. Wenn die Anzahl der Items kleiner als die Anzahl der Datensätze ist, die visualisiert werden möchten, wird eine Gruppierung der Datensätze erforderlich. Damit wird ein Formular gezeigt, mit dem der Benutzer definiert, nach welchem Kriterium die Daten gruppiert werden sollen.

Abbildung 33 zeigt den oben betreibenden Interaktionsablauf des Benutzers.

Für diese Masterarbeit werden die Komponenten des Modells in drei Gruppen aufgeteilt:

1. Die Abfrage von LOD (LOD, SPARQL-Ergebnis und Datensätze)
2. Die Visualisierungszweckabstraktion
3. Die grafische Darstellung (Visuelle Strukturen und Views)

Diese Gruppen sind die Basis für die Anwendungsfälle, die in Abschnitt 5.3 implementiert werden.

[33] Laien-Benutzer wurden im Abschnitt 3.5 beschrieben

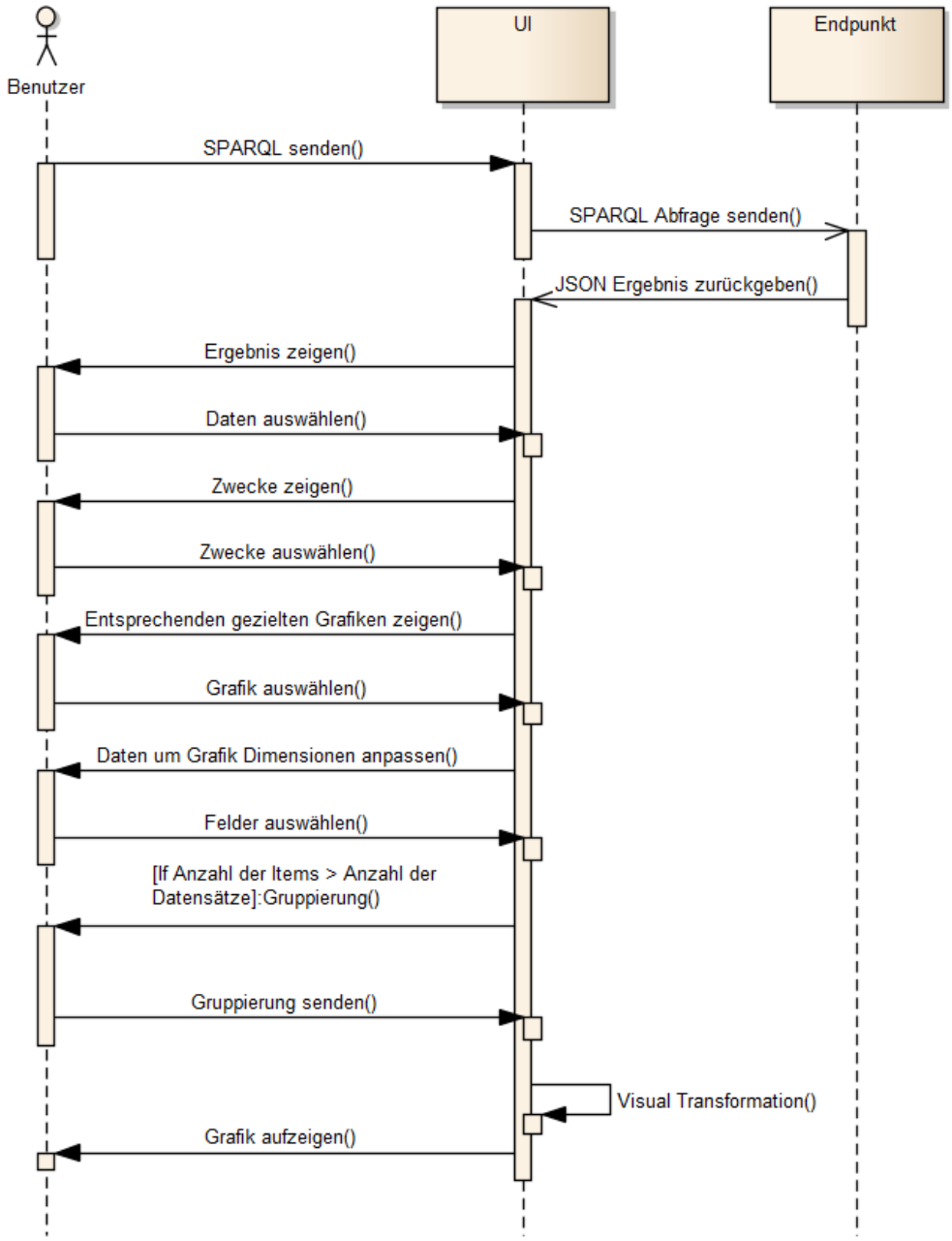


Abbildung 33: Interaktionsdiagramm des Benutzers

5.1.3 Visualisierungszweckabstraktion

Die Komponente Visualisierungszweckabstraktion ist eine Komponente, die nur dank der Klassifizierung von Visualisierungen möglich ist. Deswegen braucht die Implementierung dieser Komponente die Untersuchung von Eigenschaften und Zwecken verschiedener Grafiken.

Es wurden 35 Grafiken untersucht und mithilfe der Grafik-Ontologie die Eigenschaften klassifiziert. Das Ergebnis der Klassifizierung wird in Tabelle 3 gezeigt.

Name	Zweck	Anzahl der Dimensionen	Dimension 01			Dimension 02			Dimension 03			Anzahl der		Periode	
			Datentyp	Skala	Art	Datentyp	Skala	Art	Datentyp	Skala	Art	Min	Max	Min	Max
Arc-Diagramm	Muster, Wiederholung	1	Qualitativ	Ordinal	x-Achse							2	n	n.a.	n.a.
Balkenchart	Preisänderung	2	Qualitativ	Ordinal	x-Achse	Quantitativ	Intervall	y Achse				3	n	n.a.	n.a.
Balkendiagramm	Vergleich	2	Quantitativ	Ordinal	x-Achse	Qualitativ	Nominal	y Achse				2	20	n.a.	n.a.
Balkendiagramm mit negativen Stapel	Vergleich	3	Quantitativ	Ordinal	x-Achse	Qualitativ	Ordinal	y Achse	Qualitativ	Nominal	Farbe	4	25	n.a.	n.a.
Bereich Säulendiagramme	Tendenz	2	Qualitativ	Ordinal	x-Achse	Quantitativ	Intervall	y Achse				2	20	n.a.	n.a.
Blasendiagramm	Vergleich, Beziehung	3	Quantitativ	Intervall	x-Achse	Quantitativ	Intervall	y Achse	Quantitativ	Intervall	Größe, Farbe	2	30	n.a.	n.a.
Boxplot	Verteilung	1	Quantitativ	Intervall	x-Achse							1	10	n.a.	n.a.
Bullet-Diagramm	Vergleich	2	Quantitativ	Ordinal	x-Achse	Qualitativ	Nominal	Farbe				1	20	n.a.	n.a.
Fehlerbalken	Vergleich	2	Quantitativ	Intervall	x-Achse	Qualitativ	Nominal	y Achse				2	20	n.a.	n.a.
Flächenbereich mit Liniendiagramm	Vergleich und Tendenz	2	Qualitativ	Nominal	x-Achse	Quantitativ	Intervall	y Achse				1	20	n.a.	n.a.
Flächendiagramm	Vergleich	2	Quantitativ	Intervall	x-Achse	Quantitativ	Intervall	y Achse				2	20	n.a.	n.a.
Geglättetes Liniendiagramm	Tendenz	2	Qualitativ	Intervall	x-Achse	Quantitativ	Intervall	y Achse				1	6	1	8
Gestapeltes Balkendiagramm	Zusammensetzung,Vergleich	2	Quantitativ	Ordinal	x-Achse	Qualitativ	Intervall	y Achse				2	20	n.a.	n.a.
Gestapeltes Flächendiagramm	Zusammensetzung Vergleich	2	Qualitativ	Ordinal	x-Achse	Quantitativ	Intervall	y Achse				1	6	n.a.	n.a.
Gestapeltes Säulendiagramm	Zusammensetzung Vergleich	2	Qualitativ	Nominal	x-Achse	Quantitativ	Intervall	y Achse				2	12	n.a.	n.a.
Gestapeltes Splineflächendiagramm	Verteilung, Vergleich	2	Qualitativ	Nominal	x-Achse	Quantitativ	Intervall	y Achse				1	8	1	8
Gruppiertes Balkendiagramm	Vergleich	3	Quantitativ	Ordinal	x-Achse	Qualitativ	Ordinal	y Achse	Qualitativ	Nominal	Farbe	2	20	1	*
Gruppiertes Säulendiagramm	Vergleich	3	Qualitativ	Ordinal	x-Achse	Quantitativ	Ordinal	y Achse	Qualitativ	Nominal	Farbe	2	12	1	*
Histogramm	Verteilung	2	Quantitativ	Intervall	x-Achse	Quantitativ	Intervall	y Achse				2	r	1	1
Kreisdiagramm / Tortendiagramm	Zusammensetzung	1	Quantitativ	Ordinal	polar							2	10	n.a.	n.a.
Kreisdiagramm mit zerbrechbaren Scheiben	Zusammensetzung	1	Quantitativ	Ordinal	polar							2	10	n.a.	n.a.
Liniendiagramm	Tendenz	2	Qualitativ	Intervall	x-Achse	Quantitativ	Intervall	y Achse				1	6	1	8
Mengendiagramm	Beziehung	1	Qualitativ	Nominal	Farbe							2	6	n.a.	n.a.
Ringdiagramm	Zusammensetzung	1	Quantitativ	Ordinal	polar							3	n	n.a.	n.a.
Säulendiagramm	Vergleich	2	Qualitativ	Nominal	x-Achse	Quantitativ	Ordinal	y Achse				2	12	n.a.	n.a.
SchlagworZweilke	Informationsvisua-lisierung	1	Quantitativ	Ordinal	Größe							1	n	n.a.	n.a.
Schwimmendes Balkendiagramm	Verteilung	2	Quantitativ	Intervall	x-Achse	Qualitativ	Nominal	y Achse				2	20	n.a.	n.a.
Spark-Säulendiagramm	Tendenz	1	Quantitativ	Ordinal	x-Achse							2	20	n.a.	n.a.
Splinediagramm	Verteilung	2	Qualitativ	Nominal	x-Achse	Quantitativ	Intervall	y Achse				1	8	1	8
Splineflächendiagramm	Verteilung	2	Qualitativ	Nominal	x-Achse	Quantitativ	Intervall	y Achse				1	8	1	8
Streudiagramm	Korrelation	2	Quantitativ	Intervall	x-Achse	Quantitativ	Intervall	Größe,				10	n	n.a.	n.a.
Trichter Diagramme	Zusammensetzung	1	Quantitativ	Ordinal	Größe							2	7	n.a.	n.a.
Wasserfall	Verteilung, Zusammensetzung	2	Qualitativ	Nominal	x-Achse	Quantitativ	Ordinal	y Achse				4	20	n.a.	n.a.
Weg-Schritt-Diagramm	Daten Analyse	2	Qualitativ	Nominal	x-Achse	Quantitativ	Ordinal	y Achse				1	6	1	8
Windrose	Verteilung	2	Quantitativ	Ordinal	polar	Quantitativ	Ordinal	Farbe				16	16	n.a.	n.a.

* = Höchstzahl der Items / (Anzahl der Items x Anzahl der Periode)
n = Größe Anzahl de Items (mehr 10000)
m = Multipel
r = Rang / Anzahl der Klassen

Tabelle 1: Klassifizierung von Grafiken

Die Höchstzahl von Elementen, die eine Grafik darstellen soll, ist eine Schätzung, die auf keiner wissenschaftlichen Methodik basiert. Der Wert basiert auf einer persönlichen Wahrnehmung der grafischen Darstellung in einem Web-Browser mit einer Bildschirmauflösung von 1360x768 Pixeln. Deswegen kann der Wert nur als Beispiel betrachtet werden.

Der Wert der Mindestanzahl basiert auf dem Zweck der Grafik. Folgendes Beispiel erläutert die Schätzung.

Beispiel: Wenn der Zweck „Vergleich“ ist, ist die Mindestanzahl 2, weil immer mindestens zwei Elemente nötig sind, um vergleichen zu können.

5.2 Implementierungstechniken

5.2.1 Das Webframework

Die Entwicklung von Webanwendungen erfolgt heutzutage normalerweise mit Hilfe eines Webframeworks, weil die Frameworks die Entwicklung vereinfachen und die Kompatibilität mit den verschieden Browsern vergrößern.

Es gibt zwei verschiedene Annäherungen in der Entwicklung von Webanwendungen, abhängig von dem Auslösemechanismus. Einerseits die anfragegetriebenen, in denen eine eingehende Anfrage eine bestimmte Funktion durch ein (domänespezifischen) URL und Request-Parameter aufruft. Andererseits die eventgetriebenen, in denen die Anfrage durch ein Ereignis ausgelöst wird. Beide sehen relativ ähnlich aus. Der Unterschied liegt jedoch in der Art, wie die Interaktion erzeugt wird.

Für die Implementierung der Anwendungsfälle wird eine eventgetriebenes Framework benutzt. Das Framework heißt GWTP. GWTP basiert auf GWT (Google Web Toolkit) und auf der MVP-Architektur.

GWT

GWT (Google Web Toolkit) ist unter Apache-Lizenz Version 2.0 lizenziert und vereinfach die Anwendung verschiedener Aspekte von AJAX. Außerdem ist GWT kompatibel mit vielen Browsern und wird deswegen zur Entwicklung von Webanwendungen eingesetzt.

Das Konzept des Google Web Toolkit ist ganz einfach. Zunächst wird es im Java-Code geschrieben und der Java-Compiler übersetzt den Code in HTML und JavaScript. Die Kommunikation mit einem Server ist über Remote-Procedure-Calls möglich. Die Entwicklung ist der Entwicklung von Desktop-Anwendungen sehr ähnlich.

Nicht alle Klassen, die in Java geschrieben werden, sind mit Javascript kompatible, deswegen wird GWT in GWT-Client und GWT-Server unterteilt. Im GWT-Client laufen nur die Klassen, die in JavaScript übersetzbar sind. Im GWT-Server können alle Java -Klassen laufen.

Es wurde die Version 2.5 von GWT verwendet.

MVP Architektur

Der Code in der Benutzeroberfläche einer Anwendung ist sehr schwierig zu testen, weil die Benutzeroberfläche sehr häufig mehr tut, als sie sollte. Ohne eine klare Trennung von Verantwortlichkeiten kann die GUI-Schicht viel Logik erhalten, die eigentlich zu anderen Schichten der Anwendung gehört. Ein Entwurfsmuster, das „Model-View-Presenter“ (MVP), eignet sich besonders gut zur Lösung dieses Problems.

MVP geht aus dem „Model-View-Controller“ (MVC) hervor und ist als Entwurfsmuster zum Aufbau der Benutzeroberfläche geeignet. MVP vereinfacht die auto-

matischen Unit-Tests und verbessert die Trennung der verschiedenen Schichten der Anwendung.

MVP basiert wie MVC auch auf drei Komponenten:

- Modell (model)
- Ansicht (view)
- Präsentator (presenter)

Die Ansicht enthält keine Logik und ist nur für die Darstellung verantwortlich. Die Ansicht hat keinen Zugriff auf die Funktionalität des Präsentators und das Modell. Nur der Präsentator kann die Ansicht und das Modell steuern. Das Modell weißt nicht, wer die Ansicht und der Präsentator sind.

MVP wurde von Google als Architektur für die Entwicklung von großen Anwendungen vorgeschlagen[34].

GWTP

GWTP[35] (gwt-platform) ist eine komplette Implementierung des Entwurfsmusters MVP für GWT. GWTP vereinfacht die Arbeit mit GWT dank der Trennung von Verantwortlichkeiten. Das Framework wurde in der Entwicklung des Prototyps verwendet. GWTP ist unter Apache-Lizenz Version 2.0 lizenziert.

Einige Features, die durch GWTP unterstützt werden, sind folgende:

- Dependency-Injection durch GIN und Guice
- Implementierung von Events durch das Entwurfsmusterkommando.
- Einfache Modularisierung
- Einfache Testbarkeit
- Es gibt ein Eclipse-Plugin für die Erstellung von Modulen, Events und Aktionen.
- Mögliche Integration mit Jukito[36] (A Test Framework für GWT)
- Codegenerierung durch Annotationen

Es wurden folgende Maven-Archetypes verwendet, um die Struktur des Prototyps zu generieren:

```
1 mvn archetype:generate
     -DarchetypeGroupId=com.arcbees.archetypes \
2 -DarchetypeRepository=
3 https://oss.sonatype.org/content/repositories/snapshots/ \
4 -DarchetypeArtifactId=gwtp-basic-archetype \
5 -DarchetypeVersion=1.0-SNAPSHOT \
```

[34] Siehe https://developers.google.com/web-toolkit/articles/mvp-architecture
[35] https://github.com/ArcBees/GWTP
[36] http://jukito.arcbees.com

```
6 -DgroupId=com.projectname.project \
7 -DartifactId=new-project-name \
8 -Dmodule=Project
```

Es wurde die Version 1.0 von GWTP verwendet.

5.2.2 Die SPARQL-Abfrage

Es werden zwei Möglichkeiten berücksichtigt, um die SPARQL-Abfrage an einem LOD Endpoint durchzuführen.

Die erste Möglichkeit ist, dass GWT die Abfrage mittels AJAX macht und das Ergebnis im Browser bearbeitet. Abbildung 34 zeigt die Kommunikation zwischen GWT und Endpoint. (In GWT erfolgt die HTTP-Anfrage mit der Klasse RequestBuilder.)

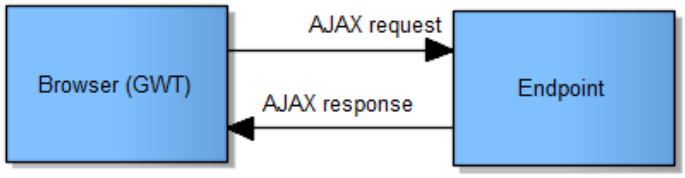


Abbildung 34: Abfrage GWT-EndPoint

Die zweite Möglichkeit ist die Verwendung von Jena. Um die Abfrage durchzuführen ist die Kommunikation des GWT-Clients mit dem GWT-Server erforderlich, weil Jena nicht in Javascript übersetzbar ist und im GWT-Server laufen muss.

Abbildung 35 zeigt die Kommunikation zwischen GWT-Client (Javascript) mit GWT-Server (Java) und Endpoint.

Jena

Jena ist ein Java Framework, um Semantik-Web-Anwendungen aufzubauen. Das Jena Framework enthält:

- eine API zum Lesen, Schreiben und Bearbeiten von RDF Daten in XML, N-Triple und Turtle -Formaten
- eine API für die Handhabung von OWL und RDFS Ontologien
- eine regelbasierte Inferenzmaschine für die Argumentation mit RDF und OWL Datenquellen
- eine große Zahl von RDF Triple, die leistungsfähig gespeichert werden können
- eine Unterstützung für SPARQL-Abfrage

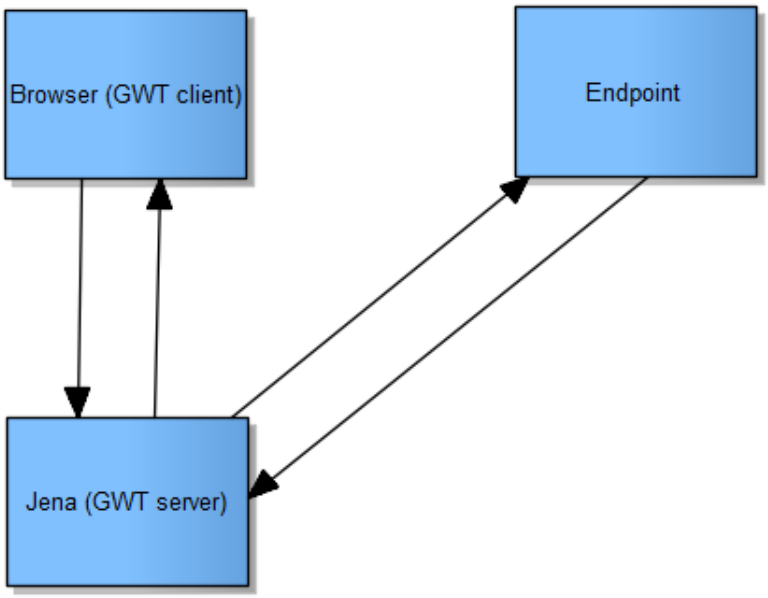


Abbildung 35: Abfrage GWT-Jena-EndPoint

XSL-Transformation

Sparqlxml2googlejson.xsl[37] ist eine XSL-Transformation, um das Ergebnis einer SPARQL-Abfrage im XML fomat umwandeln. Die Transformation gibt die JSON Format, die die Google Visualization API verwendet.

5.2.3 Die Javascript-APIs

Zur Generierung von Grafiken im Browser gibt es verschiedene Möglichkeiten wie z.B. SVG, HTML5 oder JavaScript. In dieser Masterarbeit werden die Grafiken mit JavaScript erzeugt.

Die Tabelle 2 zeigt einen Vergleich zwischen diversen JavaScript Frameworks und Bibliotheken zur Generierung von Grafiken, die für die Implementierung ausgewertet wurden. Folgende Punkte wurden dabei betrachtet:

- letzte veröffentlichte Version (Datum)
- Lizenz
- Unterstützung von GWT
- letzte veröffentlichte Version des GWT-Wrappers (Datum)

Die Tabelle enthält außerdem den Namen, das URL und die Versionsnummer.

[37] Siehe http://data-gov.tw.rpi.edu/wiki/Sparqlxml2googlejson.xsl

ID	Name	URL	Latest Release	Version	License	GWT Support	GWT URL	release	Version
1	ArcadiaCharts	http://www.arcadiacharts.com/	03-2013	1.0.5	ArcadiaCharts license, Commercial, Educational	Yes	http://www.arcadiacharts.com/gwt_charts		
2	AmCharts	http://www.amcharts.com/	03-2013	2.10.0	CC Attribution 3.0, Comercial	No			
3	Cubism.js	http://square.github.com/cubism/	05-2012	1.0.0	Apache 2.0	No			
4	D3	http://d3js.org/	03-2013	3.1.4	BSD	No			
5	Dojo GFX	http://dojotoolkit.org/features/graphics-and-charting	08-2012	1.8	modified BSD license, Academic Free License version 2.1	No	https://code.google.com/p/tatami/	2009	
6	Dygraphs	http://dygraphs.com/	12-2012	unknow	MIT	No			
7	Flot	http://www.flotcharts.org/	01-2011	0.7	MIT/BSD	Yes	https://github.com/nmorel/gflot	01-2013	3.0.2
1	Google Chart Tools	https://google-developers.appspot.com/chart/interactive/docs/index	09-2012	unknow	Terms of Service.	Yes	https://code.google.com/p/gwt-charts/	12-2012	0.9.9
							https://code.google.com/p/gwt-google-apis/wiki/VisualizationGettingStarted		
10	gRaphaël	http://g.raphaeljs.com/	03-2012	2.10.0	MIT	Yes	https://code.google.com/p/raphael4gwt/	03-2013	0.4
11	Highcharts	http://www.highcharts.com/		3.0.0	CC Attribution 3.0, Comercial	Yes	http://www.moxiegroup.com/moxieapps/gwt-highcharts/	12-2012	1.5.0
12	JavaScript InfoVis Toolkit	http://thejit.org/	05-2011	2.0.1	MIT	no			
13	jqPlot	http://www.jqplot.com/	03-2013	1.0.8	MIT/GPL v2	No			
14	jQuery Sparklines	http://omnipotent.net/jquery.sparkline/#s-about	01-2013	2.1.1	New BSD License	No			
15	Protovis	http://mbostock.github.com/protovis/	09-2010	3.3	Apache 2.0	yes	https://code.google.com/p/choosel/wiki/ProtovisGWT	08-2011	0.4.1
16	RGraph	http://www.rgraph.net/	03-2013	3/31/20	CC Attribution 3.0, Comercial	No			
17	Rickshaw	http://code.shutterstock.com/rickshaw	01-2012	unknow	MIT	Yes	https://github.com/jmadureira/rickshaw-gwt	03-2012	
18	Timeplot	http://simile-widgets.org/timeplot/	06-2007		BSD	Yes	https://code.google.com/p/gwtsimiletimeplot/	11-2009	
19	Timeline	http://simile-widgets.org/timeline/	03-2009	2.3.1	BSD	yes	https://code.google.com/p/gwtsimiletimeline/		
20	Timemap	https://code.google.com/p/timemap/	08-2011	2.0.1	MIT	no			
21	Timeglider JS	http://timeglider.com/widget/index.php?p=intro	02-2013	0.2.2	Comercial/non-profit organizations	No			
22	JS Charts	http://www.jscharts.com/	09-2012	3.04	CC Attribution 3.0,Comercial	no			
23	AwesomeChartJS	http://cyberpython.github.com/AwesomeChartJS/	08-2012		Apache 2.0	no			
24	Canvasxpress	http://canvasxpress.org/	12-2012	7.1	LGPLv3	no			

Tabelle 2: Javascript Frameworks zur Generierung von Grafiken

5.2.4 Die Karten

Vor zwölf Jahren unterteilten Kraak und Brown[81] die Arten von Web-Karten in zwei Gruppen: Dynamische und Statische Karten. Statische waren beispielsweise Bilder (BMP, JPG, etc.), Dynamische waren Applets , GIF, AVI und JavaScript.

Diese Unterteilung ist heute nicht mehr aktuell und soll nur als historische Referenz betrachtet werden, da GIF-Animationen und Java-Applets dank der Entwicklung anderer Lösungen, die Karte im Browser darzustellen, in Vergessenheit geraten sind.

Die Visualisierung der geografischen Information im Web erfolgt heutzutage meistens durch die Verwendung von Web-Karten-Services wie Google Maps[38] , Bing Maps[39] oder Nokia Maps[40] und basiert meistenteils auf JavaScript und HTML 5.

Für die Masterarbeit wurde Google Maps als Werkzeug ausgewählt, um Karten darzustellen.

Google Maps

Google Maps ist ein API zur Einbettung von Web-Karten im Browser und wird

[38] https://maps.google.de/
[39] http://www.bing.com/maps/
[40] http://m.here.com/

beispielsweise verwendet:

- um Dienstleistungen für ortsbezogene Anwendungen zu bauen.
- um Hochleistungsanwendungen zu bauen, die auf mehreren mobilen Geräten arbeiten.
- um Geodaten zu visualisieren und 3D-Bilder mit der Earth-API zu erstellen.

Google Maps with GWT

Für die Implementierung der Auswendungsfälle 3,4,5 und 7 wurde GWT-Maps-V3-Api[41]. Diese Bibliothek ist ein Java-Wrapper für die Verwendung der Google Maps JavaScript-API in GWT. Die Installation ist einfach mit Maven.

```
  <!-- GWT Maps API V3 -->
<dependency>
    <groupId>com.github.branflake2267</groupId>
    <artifactId>gwt-maps-api</artifactId>
    <version>3.10.0-alpha-6</version>
</dependency>
```

Zeitleiste with GWT

Für die Implementierung der Auswendungsfälle 6 und 7 wurde „CHAP Links Library“ aus der Liste von Javascript-APIs (siehe Abschnitt 5.2.3) ausgewählt, weil die Bibliothek GWT unterstützt und die aktuelle Version im April dieses Jahres veröffentlicht wurde. SIMILE-Timeline und Timemap wurden nicht Ausgewählt, weil beide keine neuere Version haben und Timemap außerdem von JQuery[42] abhängig ist.

Im Appendix B befindet sich weitere implementierungsdetails.

5.3 Anwendungsfälle

In folgendem Abschnitt werden die Anwendungsfälle, die implementiert werden, erläutert. Sie wurden in drei Gruppen gemäß der Aufteilung in Abschnitt 5.1.2 gegliedert.

5.3.1 LOD-Abfrage

Anwendungsfall 1 - Abfrage von LOD

Folgender Anwendungsfall beschreibt die zwei in Abschnitt 5.2.2 genannten Möglichkeiten zur Abfrage von Daten.

Normaler Ablauf

[41] https://github.com/branflake2267/GWT-Maps-V3-Api
[42] jQuery ist eine JavaScript-Bibliothek

1. Zuerst wird eine SPARQL-Abfrage durch RequestBuilder zum LOD-Endpoint geschickt.
2. Der Endpunkt gibt das Ergebnis im JSON-Format zurück.
3. Das Ergebnis wird im Browser bearbeitet, um ein vereinfachtes Format von Datensätze zu erhalten. (Datentransformation)
4. Das Ergebnis wird in tabellarischer Form angezeigt.

Optional

1a. Die Abfrage wird durch den GWT-Client zum GWT-Server und danach mit Jena zum LOD-Endpoint geschickt.

2a. Das Ergebnis wird im anderen Format zurückgegeben.

3a. Wenn die Abfrage durch Jena durchgeführt wurde, muss das Ergebnis im Server bearbeitet werden.

5.3.2 Visualisierungszweckabstraktion

Dank der Klassifizierung von Visualisierungen (siehe Abschnitt 5.1.3) ist es möglich, die Abfrage nach Eigenschaften der Grafiken durchzuführen.

Anwendungsfall 2 - Auswahl des Zwecks

Dieser Anwendungsfall fragt nach dem Zweck einer Grafik. Dazu muss die folgende SPARQL-Abfrage mindestens ein Datensatz zurückgeben:

```
1
2 @prefix lg:
     <http://linkedgraphics.com/ontology/graphic#> .
3
4 SELECT *
5 WHERE {
6     ?subject lg:hatPurpose ?purpose .
7     ?purpose rdfs:label "Comparation" .
8 }
```

Normaler Ablauf

1. Zuerst wird die SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird in tabellarischer Form angezeigt.

5.3.3 Grafische Darstellung von Ergebnissen

Die folgenden Anwendungsfälle beschreiben die grafische Darstellung von Ergebnissen. Die Daten sind auf Ort, Zeit oder beides bezogen. Die Anwendungsfälle bestehen

grundsätzlich aus zwei Teilen. Der erste Teil ist die Abfrage von LOD durch SPARQL. Der zweite Teil ist die grafische Darstellung mit Google-Maps und ggf einer Javascript-API.

Anwendungsfall 3 - Karte mit Markierungen

Auf einer Karte mit Markierungen soll die Ergebnisse einer SPARQL-Abfrage dargestellt werden. Das erste Ergebnis muss 50 Datensätze haben und das zweite Ergebnis mehr als 10.000 Datensätze haben. Die Datensätze müssen eine geographische Breite und Länge enthalten.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird auf einer Karte mit Markierungen dargestellt.

Anwendungsfall 4 - Hitzekarte

Auf einer Hitzekarte[43] soll das Ergebnis einer SPARQL-Abfrage dargestellt werden . Das erste Ergebnis muss 50 Datensätze haben und das zweite Ergebnis mehr als 10.000 Datensätze haben. Die Datensätze müssen eine geographische Breite und Länge enthalten.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird auf einer Hitzekarte dargestellt.

Anwendungsfall 5 - Karte mit gruppierten Markierungen

Auf einer Karte mit gruppierten Markierungen soll das Ergebnis einer SPARQL-Abfrage dargestellt werden. Das erste Ergebnis muss 50 Datensätze haben und das zweite Ergebnis mehr als 10.000 Datensätze haben. Die Datensätze müssen eine geographische Breite und Länge enthalten.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird auf einer Karte mit gruppierten Markierungen dargestellt.

Anwendungsfall 6 - Zeitleiste

Auf einer Zeitleiste soll das Ergebnis einer SPARQL-Abfrage dargestellt werden. Die Datensätze des Ergebnisses müssen ein Datum im Format xsd:dateTime enthalten.

[43] Hitzekarte wurde im Abschnitt 3.1.4 definiert

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird auf einer Zeitleiste dargestellt.

Auswendungsfall 7 - Interaktion einer Zeitleiste mit einer Karte

Auf einer Zeitleiste soll das Ergebnis einer SPARQL-Abfrage dargestellt werden. Jeder Datensatz enthält eine geographische Breite, eine geographische Länge, sowie ein Datum im Format xsd:dateTime, und stellt ein Ereignis dar.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird auf einer Zeitleiste dargestellt.
3. Wenn der Benutzer auf ein Ereignis klickt, das einen Datensatz repräsentiert, wird die Position des Ereignisses auf einer Karte dargestellt.

Anwendungsfall 8 - Choroplethenkarte

Auf einer Choroplethenkarte soll das Ergebnis einer SPARQL-Abfrage dargestellt werden. Jeder Datensatz muss eine geographische Breite und eine geographische Länge enthalten. Die geographischen Koordinaten sollen durch eine umgekehrte Geocodierung [44] umgewandelt werden.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.
2. Das Ergebnis wird durch die Verwendung von Geonames-API nach Ländern gruppiert.
3. Das Ergebnis wird auf einer Choroplethenkarte dargestellt.

Anwendungsfall 9 - Gruppierung von Daten

Auf einem Balkendiagramm soll das Ergebnis einer SPARQL-Abfrage dargestellt werden. Jeder Datensatz muss eine geographische Breite und eine geographische Länge enthalten. Die Anzahl von Elementen der Datensätze muss größer als die Anzahl der darzustellenden Balken sein. Für diesen Anwendungsfall wird die Höchstzahl von Items eines Balkendiagramms mit 25 angenommen.

Normaler Ablauf

1. Zuerst wird eine SPARQL -Abfrage zum LOD-Endpoint geschickt und das Ergebnis wird bearbeitet.

[44] Der Begriff Umgekehrte Geocodierung bezeichnet die Umwandlung von geographischen Koordinaten in eine von Menschen lesbare Form. Geocodierung wurde im Abschnitt 2.5 beschieben.

2. Das Ergebnis wird durch die Verwendung von Geonames-API nach Ländern gruppiert.
3. Das Ergebnis wird in einem Balkendiagramm dargestellt.

6 Ergebnisse

6.1 Auswertung der implementierten Visualisierungen

Jeder Datensatz, der dargestellt wurde, enthält entweder geografische Koordinaten (geo:lat,geo:long), ein Datum (xls:datetime) oder beide.

6.1.1 Vorgehensweise

1. Bekannte Eigenschaften der Visualisierung werden in tabellarische Form zusammengefasst (siehe Tabelle 3). Diese wurden auf Basis der Ontologie (Kap 4) definiert.
2. Unbekannte Eigenschaften der Visualisierung werden in der Tabelle 2 mit einem ? gekennzeichnet (siehe Tabelle 3).
3. Auswertung der Visualisierung wird durchgeführt
4. Fehlende Tabellen Eigenschaften werden mit Informationen ausgefüllt, die aus der Analyse resultieren (siehe Tabelle 4).

Grafik-Ontologie/ GeoTime-Ontologie																	GeoTime-Ontologie					
Name	Zweck	Anzahl der Dimensionen	Dimension 01				Dimension 02				Dimension 03				Anzahl der Items		Form der Zeit	Dauer	Punkte oder Flächen	Dynamische oder statische Darstellung
			Datentyp	Skala	Art	RDF/XML	Datentyp	Skala	Art	RDF/XML	Datentyp	Skala	Art	RDF/XML	Min	Max				
Karte mit Markierungen	?	2	Quantitativ	SpatialSkala	geographische Länge (x-Achse)	geo:lat	Quantitativ	SpatialSkala	geographische Breite (y-Achse)	geo:long	n.a.	n.a.	n.a.	n.a.	?	?	n.a.	diskrete Zeitpunkte	Punkte	statische Darstellung
Hitzekarte	?	3	Quantitativ	SpatialSkala	geographische Länge (x-Achse)	geo:lat	Quantitativ	SpatialSkala	geographische Breite (y-Achse)	geo:long	Quantitativ		Farbe	geo:long	?	?	n.a.	diskrete Zeitpunkte	Punkte	statische Darstellung
Karte mit gruppierten Markierungen	?	2	Quantitativ	SpatialSkala	geographische Länge (x-Achse)	geo:lat	Quantitativ	SpatialSkala	geographische Breite (y-Achse)	geo:long	n.a.	n.a.	n.a.	n.a.	?	?	n.a.	diskrete Zeitpunkte	Punkte	statische Darstellung
Choroplethenkarte	?	2	Quantitativ	Ordinal	Farbe		Qualitativ	Nominal	Flächen	Umwandlung von (geo:lat, geo:long)	n.a.	n.a.	n.a.	n.a.	?	250	n.a.	diskrete Zeitpunkte	Flächen	statische Darstellung
Zeitleiste und ihre Interaktion mit einer Karte	?	3	Quantitativ	SpatialSkala	geographische Länge (x-Achse)	geo:lat	Quantitativ	SpatialSkala	geographische Breite (y-Achse)	geo:long	Quantitativ	Intervall	z-Achse	datetime	n.a.	n.a.	linear e Zeit	diskrete Zeitpunkte und Zeitinterva l	Punkte	statische Darstellung

Tabelle 3: Bekannter Eigenschaften der Visualisirungen

Folgende Abschnitte beschreiben die Auswertung der Visualisierungen.

6.1.2 Karte mit Markierungen (Auswendungsfall 3)

In der Implementierung wurden 50 (Abbildung 36a) und 10000 (Abbildung 36b) Markierungen dargestellt, die jeweils ein Punkt in der Karte abbilden.

Von der Beobachtung der Grafiken werden vier Schlussforderungen gezogen:

1. Der Zweck dieser Art der Visualisierung ist die Ortsbestimmung.

(a) 50 Punkte

(b) 10000 Punkte

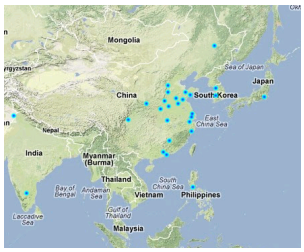
Abbildung 36: Karten mit Markierungen

2. Die Anzahl von Elementen in dieser Visualisierung ist begrenzt (Abbildung 36b), da die Markierungen ab einer bestimmten Punktanzahl nicht mehr erkennbar sind.
3. Diese Art von Visualisierung dienst diskrete Daten zu visualisieren.
4. Mindestanzahl der Items ist eins.

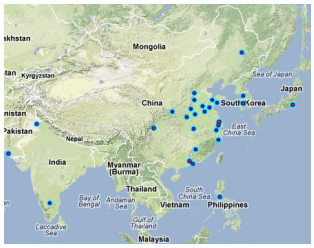
6.1.3 Hitzekarte (Auswendungsfall 4)

Die Hitzekarte von Google erlaubt die Gewichtung der Punkte, die in der Hitzekarte dargestellt werden. Um die Mindestanzahl- und Höchstzahl von Items abschätzen zu können, wurden 50 Punkte zuerst mit Gewichtung gleich 1 und danach mit Gewichtung von 0 bis 5 in Abhängigkeit von der Anzahl der Einwohner einer Stadt dargestellt. Beide Hitzekarten werden in Abbildungen 37a und 37b gezeigt.

Danach wurden mehr als 10000 Punkte mit Gewichtung gleich 1 (Abbildung 38) und mit Gewichtung von 0 bis 5 in Abhängigkeit von der Anzahl der Einwohner einer Stadt (Abbildung 39) dargestellt.



(a) Gewichtung = 1



(b) Gewichtung von 0 bis 5

(c) Gewichtung =1 (Zoom=2)

Abbildung 37: Hitzekarten mit 50 Punkten

Von der Beobachtung der Grafiken werden folgende Schlussforderungen gezogen:

- 1. Wenn die Anzahl der dargestellte Elemente klein ist (50 Punkte):
  - Die Gewichtung macht keinen Unterschied.

  - Es besteht auch keinen Unterschied zu der Funktionalität einer Karte mit Markierungen, da nur kleine Punkte gezeigt werden.
  - Es können unerwünschte visuelle Effekte gezeigt werden (siehe Abbildung 37c) : zum Beispiel, auf den ersten Blick scheint es, dass es in China viele Punkte gibt aber es ist falsch, da die Anzahl von Punkten weniger als 15 ist.

- Eine Abschätzung der Mindestanzahl von Elementen ist bei der Hitzekarte nicht einfach, weil das Zoom eine große Rolle spielt. Um sicher zu sein, müsste die Zoom-Funktion deaktiviert oder nur begrenzt sein.
- Wenn die Anzahl der dargestellte Elemente groß ist (10000 Punkte): Die Verwendung der Gewichtung ist ein wesentlicher Faktor, um den Zweck dieser Darstellung zu bestimmen (Abbildungen 38 und 39)
- Obwohl die gleiche Anzahl von Daten dargestellt werden. Abhängig von der Gewichtung können zwei Zwecke unterschieden werden:
  1. Vergleich: Wenn gewichtete Punkte dargestellt werden, können die Punkte durch die Größe verglichen werden.
  2. Informationsdichte: Wenn Punkte ohne Gewichtung dargestellt werden, kann die Dichte von räumlichen Dingen (Menschen, Orte, Ereignisse) durch die Visualisierung gezeigt werden.

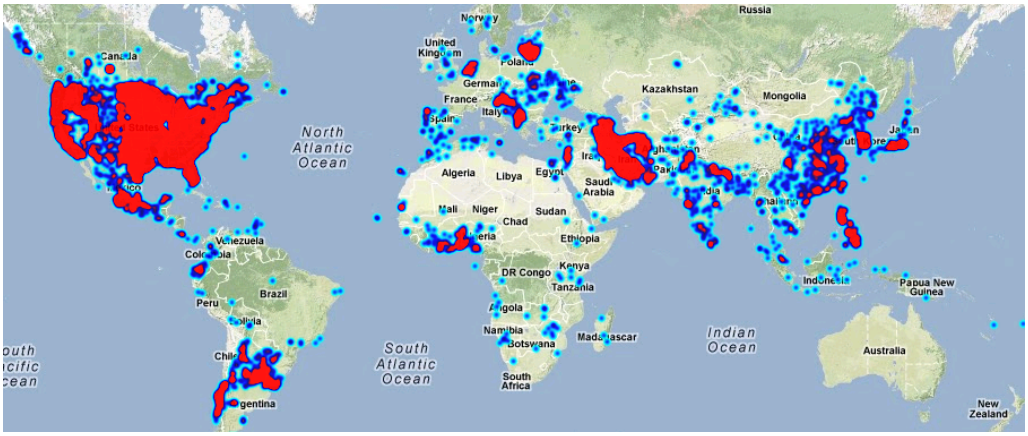


Abbildung 38: Hitzekarte 10000 Punkte ohne Gewichtung

6.1.4 Karte mit gruppierten Markierungen(Anwendungsfall 5)

In der Implementierung wurden 50 (Abbildung 40a) und 10000 (Abbildung 40b) Punkte abgebildet. Die Punkte werden gruppiert, wenn sie nah liegen und in der Karte als eine "gruppierte Markierung"dargestellt. Die Visualisierung verwendet verschiedene Farben, um die Anzahl von Elementen, die jeder Gruppe enthält, zu unterscheiden.

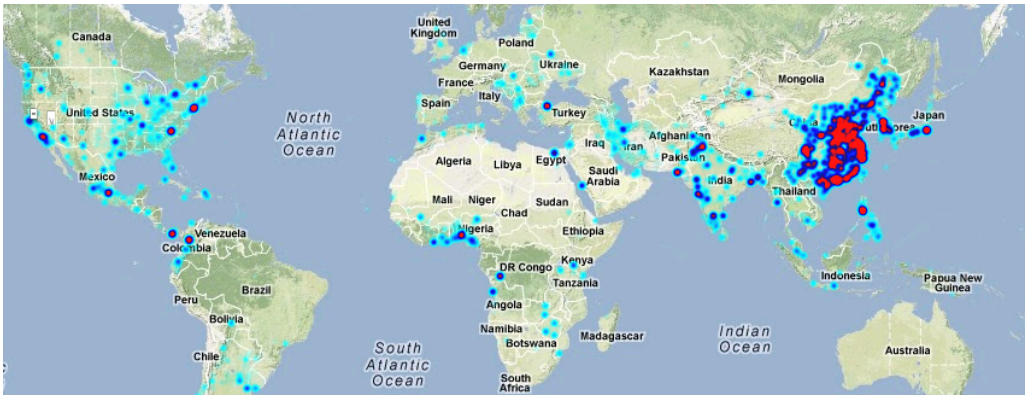
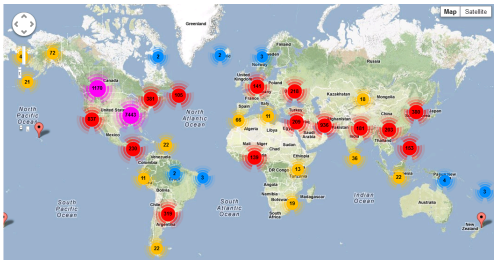


Abbildung 39: Hitzekarte 10000 Punkte mit Gewichtung

(a) 50 Punkte



(b) 10000 Punkte

Abbildung 40: Karte mit gruppierten Markierungen

Von der Beobachtung der Grafiken werden folgende Schlussforderungen gezogen:

1. Karten mit gruppierten Markierungen können als eine Spezialisierung der Visualisierung Karte mit Markierungen betrachtet werden, deswegen haben den gleichen Zweck (Ortsbestimmung).
2. Diese Visualisierungstechnik löst das Problem der Darstellung großer Mengen von Punkten durch die Gruppierung von Punkten, die in der Nähe sind. Die Visualisierung verwendet verschiedene Farben, um die Anzahl von Elementen, die jeder Gruppe enthält, zu unterscheiden.

6.1.5 Choroplethenkarte (Anwendungsfall 8)

Die Implementierung dieses Anwendungsfalls ist anders als die vorherigen, weil eine Umwandlung des Ergebnisses der SPARQL-Abfrage durch einen Geocodierung-Dienst nötig ist.

Von der Beobachtung der Grafiken werden folgende Schlussforderungen gezogen:

1. Die Transformation der geografischen Koordinaten ermöglicht die Gruppierung durch Länder.

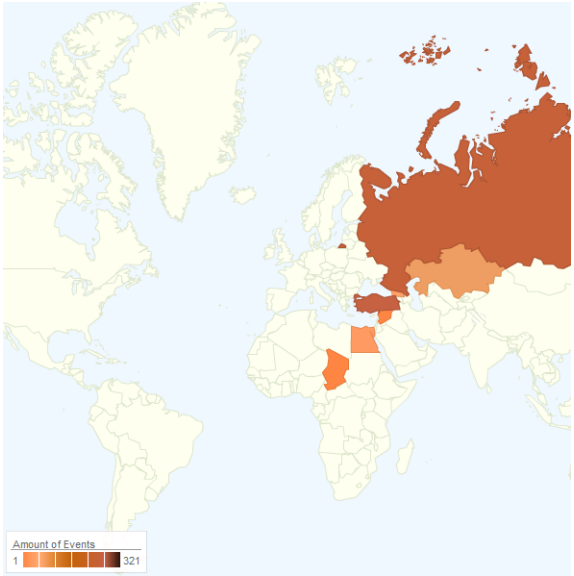


Abbildung 41: Choroplethenkarte

2. Die Mindestanzahl von Elementen ist zwei.
3. Der Zweck der Visualisierung ist Datenvergleich innerhalb von Flächen (diese Visualisierung macht Sinn nur für die Darstellung von quantitativen Daten, die sich auf einer Fläche (PLZ, Region, Länder, etc.) beziehen). Die Visualisierung wird in Abbildung Abbildung 41 gezeigt.

Zeitleiste und ihre Interaktion mit einer Karte

Die Zeitleiste ist eine grafische Darstellung, die sehr häufig mit anderen grafischen Darstellungen verwendet wird. In der Implementierung (siehe Anwendungsfall 7) wird die Position des Ereignisses auf einer Karte gezeigt, wenn der Benutzer ein Ereignis klickt. Die Visualisierung wird Abbildung 42 gezeigt.

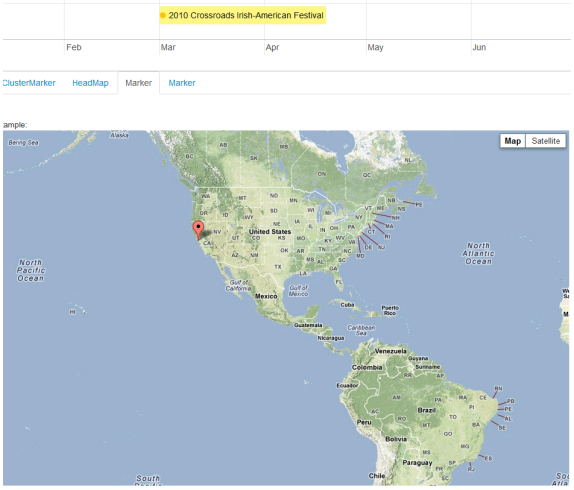


Abbildung 42: Choroplethenkarte

Von der Beobachtung der Grafiken werden folgende Schlussforderungen gezogen:

1. Die Transformation der geografischen Koordinaten ermöglicht die Gruppierung durch Länder.
2. Die Mindestanzahl von Elementen ist eins.

Balkendiagramms

Zweck der Implementierung des Anwendungsfalls 8 war, eine mögliche Gruppierung von Daten zu zeigen, da die Eigenschaften eines Balkendiagramms schon im Abschnitt 4.1.7 beschrieben wurden.

Die Implementierung des Balkendiagramms zeigte wie Geodaten gruppieren lassen (eine Gruppierung der Zeit ist natürlich auch möglich). Das ist wichtig, weil die Gruppierung ermöglicht, verschiedene Perspektiven der Daten anzubieten, unabhängig davon, wie groß die Anzahl von Datensätzen ist. Abbildung 43 zeigt, wie die Gruppierung von räumlich-zeitlichen Daten aussehen kann.

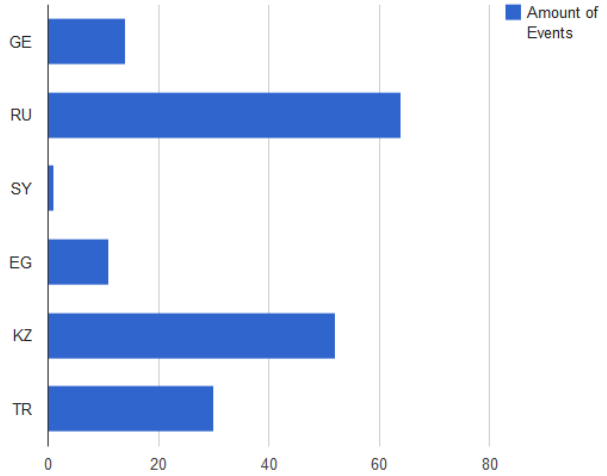


Abbildung 43: Mögliche Gruppierung räumlich-zeitlicher Daten

Zusammenfassung der Ergebnisse

Nach der Untersuchung wird das Ergebnis der Auswertung in der Tabellen 4 zusammengefasst.

			Karte mit Markierungen	Hitzekarte	Karte mit gruppierten Markierungen	Choroplethenkarte	Zeitleiste und ihre Interaktion mit einer Karte
	Name		**Karte mit Markierungen**	**Hitzekarte**	**Karte mit gruppierten Markierungen**	**Choroplethenkarte**	**Zeitleiste und ihre Interaktion mit einer Karte**
GeoTime-Ontologie	**Dynamische oder statiche Darstellung**		statische Darstellung	statische Darstellung	statische Darstellung	statische Darstellung	statische Darstellung
GeoTime-Ontologie	**Punkte oder Flächen**		Punkte	Punkte	Punkte	Flächen	Punkte
GeoTime-Ontologie	**Dauer**		diskrete Zeitpunkte	diskrete Zeitpunkte	diskrete Zeitpunkte	diskrete Zeitpunkte	diskrete Zeitpunkte und Zeitinterval
GeoTime-Ontologie	**Form der Zeit**		n.a.	n.a.	n.a.	n.a.	lineare Zeit
GeoTime-Ontologie	**Anzahl der Items**	**Max**	begrenzt	unbegrenzt	unbegrenzt	gesamte Flächenelemente	n.a.
GeoTime-Ontologie	**Anzahl der Items**	**Min**	1	50	1	2	1
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 03**	**RDF/XML**	n.a.	geo:long	n.a.	n.a.	datetime
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 03**	**Art**	n.a.	Farbe	n.a.	n.a.	z-Achse
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 03**	**Skala**	n.a.		n.a.	n.a.	Intervall
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 03**	**Datentyp**	n.a.	Quantitativ	n.a.	n.a.	Quantitativ
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 02**	**RDF/XML**	geo:long	geo:long	geo:long	Umwandlung von (geo:lat, geo:long)	geo:long
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 02**	**Art**	geographische Breite (y-Achse)	geographische Breite (y-Achse)	geographische Breite (y-Achse)	Flächen	geographische Breite (y-Achse)
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 02**	**Skala**	SpatialSkala	SpatialSkala	SpatialSkala	Nominal	SpatialSkala
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 02**	**Datentyp**	Quantitativ	Quantitativ	Quantitativ	Qualitativ	Quantitativ
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 01**	**RDF/XML**	geo:lat	geo:lat	geo:lat		geo:lat
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 01**	**Art**	geographische Länge (x-Achse)	geographische Länge (x-Achse)	geographische Länge (x-Achse)	Farbe	geographische Länge (x-Achse)
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 01**	**Skala**	SpatialSkala	SpatialSkala	SpatialSkala	Ordinal	SpatialSkala
Grafik-Ontologie/ GeoTime-Ontologie	**Dimension 01**	**Datentyp**	Quantitativ	Quantitativ	Quantitativ	Quantitativ	Quantitativ
Grafik-Ontologie/ GeoTime-Ontologie	**Anzahl der Dimensionen**		2	3	2	2	3
Grafik-Ontologie/ GeoTime-Ontologie	**Zweck**		Ortsbstimmung	Vergleich/ Informationsdichte	Ortsbestimmung	Vergleich	Ortsbestimmung

Tabelle 4: Ergebnis der Auswertung

6.2 Abwägungen

Die Abfrage durch RequestBuilder hat den Vorteil, dass der Browser die Daten ohne Server erhalten kann, aber es gibt zwei Probleme mit dieser Annäherung: Das erste ist die Same-Origin-Policy (SOP) des Browsers, die nicht erlaubt, alle Endpoints zu erreichen. Das zweite Problem setzt ein, wenn das Abfrageergebnis größer als 10 MB ist, denn die Browser haben Schwierigkeiten, die Daten zu verarbeiten.

Die Verwendung von Jena ist eine gute Option für die Bearbeitung der Abfrage im Server. Sie ist jedoch vielleicht nicht nötig, wenn das XML-Ergebnis einer SPARQL-Abfrage durch die Verwendung einer XSL-Transformation umgewandelt wird. Die Transformation vermeidet eine unnötige Bearbeitung des SPARQL-Ergebnisses im JSON-Format und erlaubt die automatische Erzeugung von Google-DataTable, die Methoden bietet, um DataViews zu erzeugen, die ohne Änderung der Daten die Bearbeitung von Zeilen oder Spalten ermöglichen . Das ist ein großer Vorteil, um Operationen zu den Ergebnissen durchzuführen.

Eine Lösung für die automatische Generierung von Grafiken mit Geodaten wurde nicht gefunden, weil das Format eines SPARQL-Ergebnisses XML-Datentypen sind. Das bewirkt, dass die geographischen Koordinaten als Datentyp Float zurückgegeben werden und die Interaktion des Benutzers erforderlich wird.

Dank der Verwendung der Grafik-Ontologie war es möglich die Grafiken nach Dimension/en (Anzahl, Skala und Art), Zweck, Anzahl von Items[45] (Mindestanzahl, Höchstzahl) zu klassifizieren. Es wurde jedoch kein wissenschaftliches Verfahren gefunden, um die Höchstzahl von Elementen abzuschätzen, da viele Faktoren wie Subjektivität, Selektivität, Art der Veröffentlichung und Interaktionstechniken eine Rolle spielen. Eine wissenschaftliche Schätzung braucht eine wissenschaftliche Studie, in der beispielsweise visuelle Wahrnehmung und Wissenstransfer analysiert werden.

Es wurde die GeoZeit-Ontologie Die Ontologie beschreibt Eigenschaften der Daten mit Zeit- und Ortsbezug und ihre Darstellung und ermöglicht die Identifizierung von Grafiken, die zu Daten mit Zeit- und Ortsbezug passen.

Mit Hilfe der Implementierung wurden Unterschiede zwischen die Visualisierung von LOD mit Zeit- und Ortsbezug gefunden und anhand der GeoZeit-Ontologie klassifiziert. Ein Beispiel dafür ist die Hitzekarte, die zwei verschiedene Zwecke Abhängig von der Verwendung der Gewichtung gezeigt hat. Es wurde Karten mit gruppierten Markierungen als gute Alternative gezeigt, wenn der Zweck der Visualisierung von Geodaten Ortsbestimmung und die Anzahl von Elementen groß ist.

Es wurde gezeigt, dass die Interaktion einer Zeitleiste und einer Karte eine gute Möglichkeit ist, räumlich-zeitliche Daten zu visualisieren. Besonders, wenn sie Ereignisse sind. Es könnte jedoch nicht die Funktionalität, die das Projekt

[45] Anzahl von Elementen, die in einer Grafik darstellt werden sollen

Timemap[46] bietet, implementiert, um Daten durch die Zeitleiste zu filtern.

Die Möglichkeit, wie die Daten gruppiert werden können, zeigt, dass die Anzahl von Datensätzen aus dem Ergebnis einer SPARQL-Abfrage kein entscheidender Faktor bei der Auswahl einer bestimmten Visualisierung ist. Ein entscheidender Faktor ist die Kombination von Zweck und Anzahl von Elementen, die in der Visualisierung dargestellt werden können. Durch die Verwendung dieser Kombination kann die Auswahl einer geeigneten Visualisierung für den Benutzer erleichtert werden.

Durch die Implementierung der Anwendungsfälle wurde die Machbarkeit der Komponenten des beschriebenen Modells im Abschnitt 5.1.1 bewiesen.

6.3 Weiterführende Arbeit

Für eine vollständige Implementierung werden folgende Verbesserungen vorgeschlagen: die Anwendung einer Caching-Technik für die SPARQL Abfragen, damit eine effizientere Geokodierung, die Möglichkeit das Ergebnis der Visualisierung zu speichern und die Möglichkeit, dass der Benutzer andere Visualisierungen erzeugen kann, ohne die SPARQL-Abfrage zu wiederholen.

Es ist denkbar die Verwendung der Grafik-Ontologie, um die Grafiken von Webseiten in RDFa oder ein ähnliches Format zu beschreiben. Mit dieser Beschreibung wäre es möglich eine semantische Visualisierung, die viele weitere Möglichkeiten anbieten könnte.

Eine Erweiterung der Grafik-Ontologie kann auch die JavaScript Frameworks und Bibliotheken zur Generierung von Grafiken modelliert werden. Durch diese Beziehung „Anbieter bietet Grafiken an“ kann man wissen, welcher Anbieter für eine spezifische Darstellung es gibt. Man kann ein Stufe weiter gehen und die Lizenz als Attribute der Klasse Anbieter hinzufügen. Damit könnte man nicht nur die Darstellung finden, die zu den Daten passen, sondern auch auswählen, den passende Anbieter.

[46] http://code.google.com/p/timemap/ (Stand 11.05.2013)

Appendices

A Ontologien

Darstellung der Grafik-Ontologie

Classes

AestheticDimention

AestheticDimention $\sqsubseteq$ Dimension

AxisDimention

AxisDimention $\sqsubseteq$ Dimension

ColorDimention

ColorDimention $\sqsubseteq$ AestheticDimention

Data

Data $\sqsubseteq$ = hatDateType DataType

Data $\sqsubseteq$ = hasScale Scale

DataContinuity

DataContinuity $\equiv$ {Discrete} $\sqcup$ {Continuous}

DataContinuity $\sqsubseteq$ Thing

DataType

Dimension

Dimension $\sqsubseteq$ Thing

Dimension $\sqsubseteq$ $\exists$ represents Data

Form

Form $\sqsubseteq$ AestheticDimention

GeoPosition

Graphic

Graphic $\sqsubseteq \geq 1$ hatDimention Thing

Graphic $\sqsubseteq \exists$ hatDimention Dimension

Graphic $\sqsubseteq \geq 1$ hatPurpose Thing

Graphic $\sqsubseteq \exists$ hatPurpose Purpose

IntervalScale

IntervalScale $\sqsubseteq \exists$ hatContinuity {Continuous}

IntervalScale $\sqsubseteq$ Scale

IntervalScale $\sqsubseteq \neg$ RatioScale

IntervalScale $\sqsubseteq \neg$ NominalScale

IntervalScale $\sqsubseteq \neg$ OrdinalScale

NominalScale

NominalScale $\sqsubseteq \exists$ hatContinuity {Discrete}

NominalScale $\sqsubseteq$ Scale

NominalScale $\sqsubseteq \neg$ OrdinalScale

NominalScale $\sqsubseteq \neg$ RatioScale

NominalScale $\sqsubseteq \neg$ IntervalScale

Optics

Optics $\sqsubseteq$ AestheticDimention

OrdinalScale

OrdinalScale $\sqsubseteq$ Scale

OrdinalScale $\sqsubseteq \exists$ hatContinuity {Discrete}

OrdinalScale $\sqsubseteq \neg$ NominalScale

OrdinalScale $\sqsubseteq \neg$ IntervalScale

OrdinalScale $\sqsubseteq \neg$ RatioScale

Point

Point $\sqsubseteq$ SpatialThing

Point

Point $\sqsubseteq$ GeoPosition

Purpose

Purpose $\sqsubseteq$ Thing

RatioScale

RatioScale $\sqsubseteq$ $\exists$ hatContinuity {Continuous}

RatioScale $\sqsubseteq$ Scale

RatioScale $\sqsubseteq$ $\neg$ NominalScale

RatioScale $\sqsubseteq$ $\neg$ IntervalScale

RatioScale $\sqsubseteq$ $\neg$ OrdinalScale

Region

Region $\sqsubseteq$ GeoPosition

Scale

Scale $\sqsubseteq$ Thing

SpatialThing

Texture

Texture $\sqsubseteq$ AestheticDimention

Thing

Object properties

hasScale

$\exists$ hasScale Thing $\sqsubseteq$ Data

$\top \sqsubseteq \forall$ hasScale Scale

hatContinuity

$\exists$ hatContinuity Thing $\sqsubseteq$ Scale

$\top \sqsubseteq \forall$ hatContinuity DataContinuity

hatDateType

$\exists$ hatDateType Thing $\sqsubseteq$ DataType

$\top \sqsubseteq \forall$ hatDateType DataType

hatDimention

$\exists$ hatDimention Thing $\sqsubseteq$ Dimension

$\top \sqsubseteq \forall$ hatDimention Dimension

hatPurpose

$\exists$ hatPurpose Thing $\sqsubseteq$ Purpose

$\top \sqsubseteq \forall$ hatPurpose Purpose

represents

$\exists$ represents Thing $\sqsubseteq$ Dimension

$\top \sqsubseteq \forall$ represents Data

Data properties

maxRecomendedAmountItems

minAmountItems

Individuals

Blur

Blur : Optics

Brightness

Brightness : ColorDimention

Compare

Compare : Purpose

Composition

Composition : Purpose

Continuous

Continuous : DataContinuity

$\{\text{Continuous}\} \not\equiv \{\text{Discrete}\}$

Discrete

Discrete : DataContinuity

$\{\text{Continuous}\} \not\equiv \{\text{Discrete}\}$

Distribution

Distribution : Purpose

Granularity

Granularity : Texture

Hue

Hue : ColorDimention

IntervalScaleInstance

IntervalScaleInstance : IntervalScale

Localisation

Localisation : Purpose

NominalScaleInstance

NominalScaleInstance : NominalScale

OrdinalScaleInstance

OrdinalScaleInstance : OrdinalScale

Orientation

Orientation : Texture

Pattern

Pattern : Purpose

PatternTexture

PatternTexture : Texture

QualitativeData

QualitativeData : DataType

QuantitativeData

QuantitativeData : DataType

RatioScaleInstance

RatioScaleInstance : RatioScale

Relation

Relation : Purpose

Rotation

Rotation : Form

Saturation

Saturation : ColorDimention

Shape

Shape : Form

Size

Size : Form

Trend

Trend : Purpose

Trnasparency

Trnasparency : Optics

X-AxisDimention

X-AxisDimention : AxisDimention

Y-AxisDimention

Y-AxisDimention : AxisDimention

Z-AxisDimention

Z-AxisDimention : AxisDimention

Darstellung der GeoTimeOntologie

Es werden folgende OWL Prefix verwendet.

```
@prefix geoTime:
   <http://linkedgraphics.com/ontology/geoTime#> .
@prefix geopolitical:
   <http://aims.fao.org/aos/geopolitical.owl#> .
@prefix lg:
   <http://linkedgraphics.com/ontology/graphic#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix rdf:
   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time:    <http://www.w3.org/2006/time#> .
```

Abbildung 44 zeigt die Klasse Form der Zeit (TimeForm) und ihre Ausprägungen Linearezeit (LinearTime) und Zyklischezeit (CyclicTime). Form der Zeit ist eine Klasse, die Unterklasse von Ding (owl:Thing) ist. Die Klasse besteht aus zwei mögliche Ausprägungen Lineare- und Zyklischezeit.

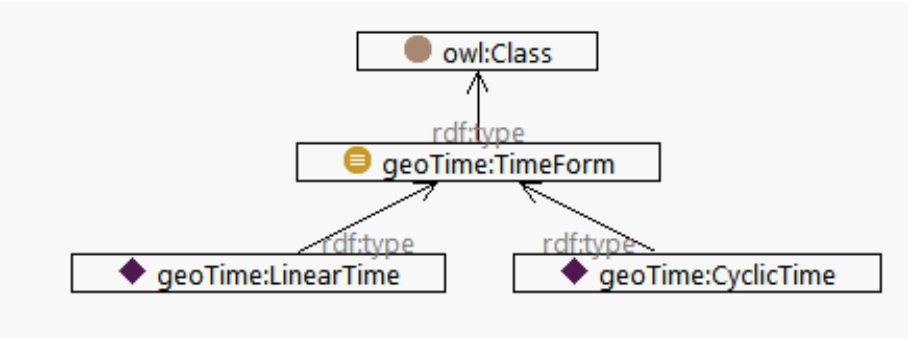


Abbildung 44: OWL Klasse Form der Zeit (TimeForm)

Abbildung 45 zeigt die Klassen räumlichen Skala (SpatialScale) und Zeitskala (TemporalScale), die zwei Unterklassen OrdinalZeitskala (TimeOrdinalScale) und kontinuierliche Zeitskala (TimeContinuosScale) hat. Form der Zeit ist eine Unterklasse von lg:Scala.

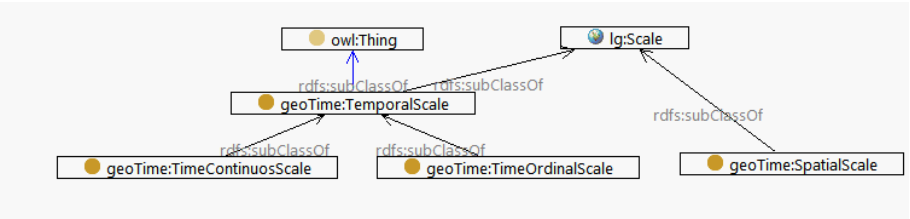


Abbildung 45: OWL Klasse Form der Zeit (TimeScale)

Abbildung 46 zeigt die Klasse Dauer (Duration) und ihre Ausprägungen Diskrete Zeitpunkt (DiscreteTimePoint) und Zeitintervall (IntervalTime). Dauer ist eine Klasse, die Unterklasse von Ding (owl:Thing) ist. Die Klasse besteht aus zwei mögliche Ausprägungen Diskrette Zeitpunkt und Zeitintervall.

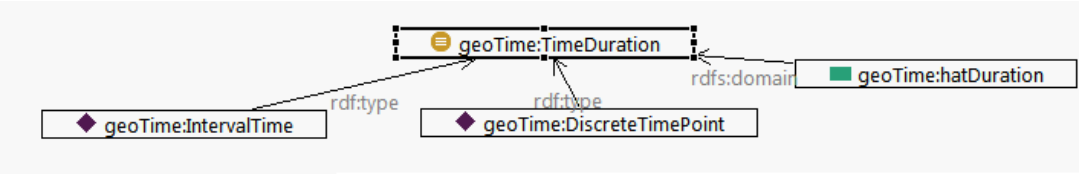


Abbildung 46: OWL Klasse Dauer (Duration)

Abbildung 47 zeigt die Klasse Zeitdaten (Timedata) die vier Unterklassen hat. Kontinuierliche Zeitdaten (ContinuousTimeData) ist eine Klasse, die eine kontinuierliche Zeitskala hat. Kumulative Zeitdaten ist eine Klasse, die eine OrdinalZeitskala hat. Augenblickliche Zeitdaten ist eine Klasse, die eine OrdinalZeitskala hat.

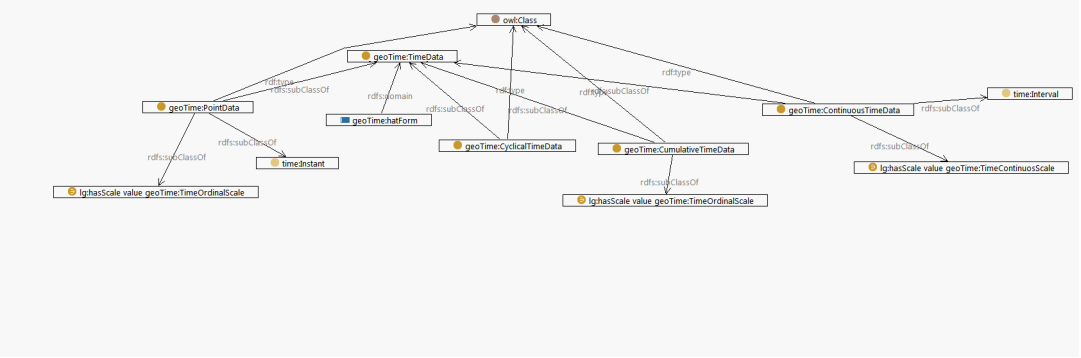


Abbildung 47: OWL Klasse Zeitdaten(TimeData)

Abbildung 48 zeigt die Klasse Geodaten (Geodata) die zwei Unterklassen hat. Kontinuierliche Geodaten (ContinuousGeoData) ist eine Klasse, die eine kontinuierliche Zeitskala hat. Diskrete Geodaten (DiscreteGeoData) ist eine Klasse, die eine OrdinalZeitskala hat. Augenblickliche Zeitdaten ist eine Klasse, die eine OrdinalZeitskala hat.

Abbildung 49 zeigt die Klassenhierarchie der Klasse Ort (Place). Die Hierarchie zeigt, wie ein Ort sich in einer Region oder Staat befindet.

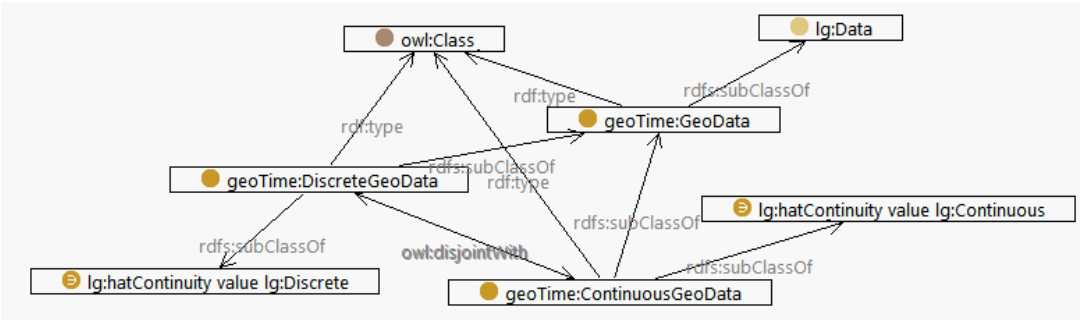


Abbildung 48: OWL Klasse Geodaten (Geodata) und Unterklassen

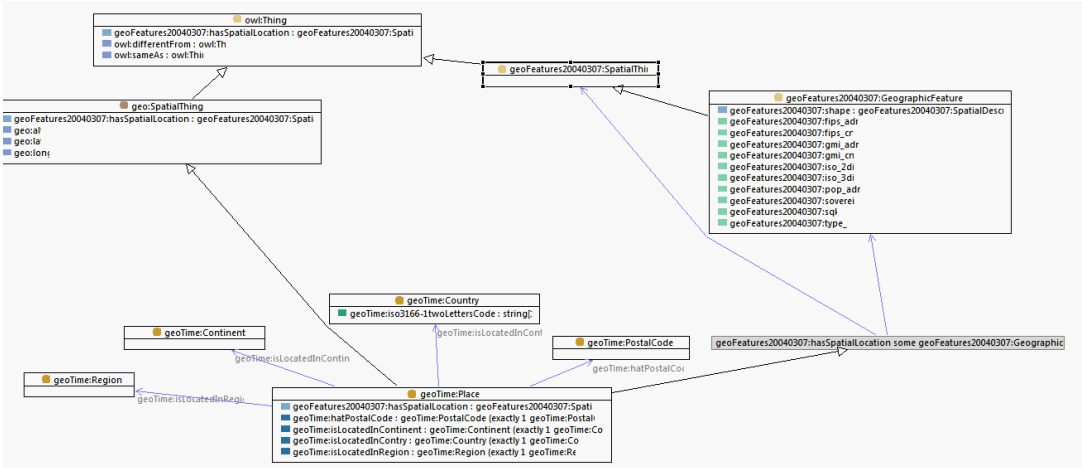


Abbildung 49: OWL Klasse Ort(place)

Classes

City

City $\sqsubseteq$ PoliticalEntities

Continent

Continent $\sqsubseteq$ PoliticalEntities

ContinuousGeoData

ContinuousGeoData $\sqsubseteq \exists$ hatContinuity {Continuous}

ContinuousGeoData $\sqsubseteq$ GeoData

ContinuousGeoData $\sqsubseteq \neg$ DiscreteGeoData

ContinuousTimeData

ContinuousTimeData $\sqsubseteq$ Interval

ContinuousTimeData $\sqsubseteq \exists$ hasScale {TimeContinuosScale}

ContinuousTimeData $\sqsubseteq$ TimeData

Country

Country $\sqsubseteq$ = iso3166-1twoLettersCode

Country $\sqsubseteq$ PoliticalEntities

CumulativeTimeData

CumulativeTimeData $\sqsubseteq \exists$ hasScale {TimeOrdinalScale}

CumulativeTimeData $\sqsubseteq$ TimeData

CyclicalTimeData

CyclicalTimeData $\sqsubseteq$ TimeData

Data

DinamicGraphic

DinamicGraphic $\sqsubseteq$ Graphic

DiscreteGeoData

DiscreteGeoData $\sqsubseteq$ GeoData

DiscreteGeoData $\sqsubseteq \exists$ hatContinuity {Discrete}

DiscreteGeoData $\sqsubseteq \neg$ ContinuousGeoData

GeoData

GeoData $\sqsubseteq$ Data

GeographicFeature

Graphic

Instant

Interval

OrdinalScale

Place

Place $\sqsubseteq$ = hatPostalCode PostalCode

Place $\sqsubseteq$ = isLocatedInContry Country

Place $\sqsubseteq$ = isLocatedInContinent Continent

Place $\sqsubseteq$ = isLocatedInRegion Region

Place $\sqsubseteq \exists$ hasSpatialLocation GeographicFeature

Place $\sqsubseteq$ SpatialThing

Point

Point $\sqsubseteq \exists$ isLocatedIn Place

Point $\sqsubseteq$ SpatialThing

PointData

PointData $\sqsubseteq$ Instant

PointData $\sqsubseteq$ TimeData

PointData $\sqsubseteq \exists$ hasScale {TimeOrdinalScale}

PoliticalEntities

PostalCode

PostalCode $\sqsubseteq$ PoliticalEntities

Region

Region $\sqsubseteq$ PoliticalEntities

Scale

SpatialScale

SpatialScale $\sqsubseteq$ Scale

SpatialThing

StaticGraphic

StaticGraphic $\sqsubseteq$ Graphic

TimeContinuosScale

TimeContinuosScale $\sqsubseteq \exists$ hatContinuity {Continuous}

TimeContinuosScale $\sqsubseteq$ TimeScale

TimeData

TimeData $\sqsubseteq \exists$ hatForm TimeForm

TimeData $\sqsubseteq$ Data

TimeDuration

TimeDuration $\equiv$ {DiscreteTimePoint} $\sqcup$ {IntervalTime}

TimeForm

TimeForm $\equiv$ {LinealTime} $\sqcup$ {CyclicTime}

TimeOrdinalScale

TimeOrdinalScale $\sqsubseteq$ OrdinalScale

TimeOrdinalScale $\sqsubseteq$ TimeScale

TimeScale

TimeScale $\sqsubseteq$ Scale

Object properties

hasScale

hasSpatialLocation

$\exists$ hasSpatialLocation Thing $\sqsubseteq$ SpatialThing

hatContinuity

hatForm

$\exists$ hatForm Thing $\sqsubseteq$ TimeData

$\top \sqsubseteq \forall$ hatForm TimeForm

hatPostalCode

$\sqsubseteq$ isLocatedIn

$\exists$ hatPostalCode Thing $\sqsubseteq$ Place

$\top \sqsubseteq \forall$ hatPostalCode PostalCode

isLocatedIn

$\exists$ isLocatedIn Thing $\sqsubseteq$ Point

$\top \sqsubseteq \forall$ isLocatedIn Place

isLocatedInContinent

$\sqsubseteq$ isLocatedIn

$\exists$ isLocatedInContinent Thing $\sqsubseteq$ Place

$\top \sqsubseteq \forall$ isLocatedInContinent Continent

isLocatedInContry

$\sqsubseteq$ isLocatedIn

$\exists$ isLocatedInContry Thing $\sqsubseteq$ Place

$\top \sqsubseteq \forall$ isLocatedInContry Country

isLocatedInRegion

$\sqsubseteq$ isLocatedIn

$\exists$ isLocatedInRegion Thing $\sqsubseteq$ Place

$\top \sqsubseteq \forall$ isLocatedInRegion Region

isStoredAs

Data properties

hatDuration

iso3166-1twoLettersCode

{iso31661Code} $\equiv$ {iso3166-1twoLettersCode}

iso31661Code

{iso31661Code} $\equiv$ {iso3166-1twoLettersCode}

postalCode

{postalCode} $\equiv$ {PostalCode}

territory

{territory} $\equiv$ {Country}

B Implementierungsdetails

Geocoding

Listing 15: Code Geocoding API

```
// Model
public class GeoConvertion {
  public interface VubReader extends
     JsonReader<GeoConvertion> {}
  public static final VubReader JSON =
     GWT.create(VubReader.class);

  String languages;
  String distance;
  String countryName;
  String countryCode;
}
//Controller

RequestBuilder xmlRequestBuilder =
  new RequestBuilder(RequestBuilder.GET,
  "http://api.geonames.org/countryCode?lat=" + lat +
     "&lng="
        + lon + "&username=XXXXXXX&type=JSON");

    try {
      xmlRequestBuilder.sendRequest(null, new
         RequestCallback() {

        @Override
        public void onResponseReceived(Request request,
           Response response) {
          GeoConvertion vub =
             GeoConvertion.JSON.read((response.getText()));
        }

      });
```

Abfragen SPARQL

Listing 16: Auswendungsfall 1

```

PREFIX dbo: <http://dbpedia.org/ontology/>
 PREFIX dbp: <http://dbpedia.org/property/>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
 SELECT ?subject ((?subject) as ?city) ((?label) as ?label)
((?latitude) as ?latitude)
```

```
8  ((?longitude) as ?longitude)
9  ?populationTotal as ?maxPopulation
10 WHERE { ?subject a dbo:City. ?subject rdfs:label ?label.
11 OPTIONAL {?subject rdfs:comment ?abstract.} ?subject
      dbp:populationTotal ?populationTotal.
12 OPTIONAL {?subject geo:lat ?latitude.}
13 OPTIONAL {?subject geo:long ?longitude.}
14 FILTER (lang(?label) = "en" && lang(?abstract) = "en" &&
      ?populationTotal >= "1"^^xsd:integer  )}
15  GROUP BY ?subject ?latitude ?longitude ?label ORDER BY
       DESC(?maxPopulation)
16 LIMIT 200
```

Listing 17: Auswendungsfälle 3,4 und 5

```
1  PREFIX dbo: <http://dbpedia.org/ontology/>
2   PREFIX dbp: <http://dbpedia.org/property/>
3   PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4   PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
5   SELECT ?subject
6   (sql:SAMPLE(?subject) as ?city)
7  (sql:SAMPLE(?label) as ?label)
8   (sql:SAMPLE(?latitude) as ?latitude)
9   (sql:SAMPLE(?longitude) as ?longitude)
10  max(?populationTotal) as ?maxPopulation
11  max(?rainyDays) as ?rainyDays WHERE { ?subject a
       dbo:City. ?subject rdfs:label ?label.
12 OPTIONAL {?subject rdfs:comment ?abstract.} ?subject
      dbp:populationTotal ?populationTotal.
13 OPTIONAL {?subject geo:lat ?latitude.}
14  OPTIONAL {?subject geo:long ?longitude.}
15  OPTIONAL {?subject dbp:yearPrecipitationDays ?rainyDays.}
16 FILTER (  ?populationTotal >= "1"^^xsd:integer  )} GROUP
      BY ?subject
17 ORDER BY DESC(?maxPopulation)
```

Listing 18: Auswendungsfälle 6 und 7

```
1
2  SELECT DISTINCT ?event ?title ?date ?lat ?lon WHERE {
3  ?event <http://linkedevents.org/ontology/inSpace> ?space .
4  ?space <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat.
5  ?space <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?lon.
6   ?event <http://purl.org/dc/elements/1.1/title> ?title .
7  ?event <http://linkedevents.org/ontology/atTime> ?odate .
8   ?odate <http://www.w3.org/2006/time#inXSDDateTime> ?date .
9  FILTER (regex(str(?date), "2010", "i") )}
10 LIMIT 5
```

Listing 19: Auswendungsfälle 8 und 9

```
1
2  SELECT DISTINCT ?event ?title ?date ?lat ?lon
3  WHERE {
4  ?event <http://linkedevents.org/ontology/inSpace> ?space .
5  ?space <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat.
6  ?space <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?lon.
7  ?event <http://purl.org/dc/elements/1.1/title> ?title .
8  ?event <http://linkedevents.org/ontology/atTime> ?odate .
9   ?odate <http://www.w3.org/2006/time#inXSDDateTime> ?date
       .  FILTER (regex(str(?date), "2010", "i") )}
10 LIMIT 1000
```

Literatur

[1] N. Bartelme, "Einführung," in *Geoinformatik*. Springer Berlin Heidelberg, 2005, pp. 1–42. [Online]. Available: http://dx.doi.org/10.1007/3-540-27201-1_1

[2] M.-j. Kraak, "Timelines, temporal resolution, temporal zoom and time geography," in *Proceedings 22nd International Cartographic Conference, A Coruna Spain*, 2005.

[3] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "Visualisierung zeitlicher Verläufe auf geografischen Karte," in *Visualisierung und Erschliessung von Geodaten: Seminar GEOVIS 2003 26. bis 27. Februar 2003, Universität Hannover*, vol. 7. Kirschbaum Verlag, 2003, p. 47.

[4] X. Li and M.-J. Kraak, "The Time Wave. A New Method of Visual Exploration of Geo-data in Timespace," *Cartographic Journal, The*, vol. 45, no. 3, pp. 193–200, Aug. 2008. [Online]. Available: http://openurl.ingenta.com/content/xref?genre=article&issn=0008-7041&volume=45&issue=3&spage=193

[5] D. J. Duke, K. W. Brodlie, D. A. Duce, and I. Herman, "Do you see what I mean?" *IEEE Computer Graphics and Applications*, vol. 25, no. 3, pp. 6–9, 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1438250

[6] R. B. Haber and D. A. McNabb, "Visualization idioms: A conceptual model for scientific visualization systems," *Visualization in scientific computing*, vol. 74, pp. 74–93, 1990.

[7] E. H.-h. Chi and J. T. Riedl, "An Operator Interaction Framework for Visualization Systems," *IEEE Symposium on Information Visualization (InfoVis),*, 1998.

[8] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann Pub, 1999.

[9] L. Wilkinson and G. Wills, *The grammar of graphics*. Springer Science+Business Media, 2005.

[10] M. Polowinski, Jan and Voigt, "Visualisation Ontology (VISO)." [Online]. Available: http://141.76.68.42/docu/main/index.html

[11] M. Voigt, S. Pietschmann, and K. Meiß ner, "Towards a semantics-based, end-user-centered information visualization process," in *Proc. of the 3rd Intern. Workshop on Semantic Models for Adaptive Interacive Systems (SEMAIS'12)*, 2012.

[12] E. H. Chi, "A Taxonomy of Visualization Techniques using the Data State Reference Model," *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, no. Table 2, pp. 69 – 75, 2000.

[13] J. Brunetti, S. Auer, and R. García, "The Linked Data Visualization Model," *svn.aksw.org*, 2012. [Online]. Available: http://svn.aksw.org/papers/2012/ISWC_Visualization/public.pdf

[14] J. H. Tim Berners-Lee and O. Lassila, "The Semantic Web," *Scientific american*, vol. 284, no. 5, pp. 96–101, 2001. [Online]. Available: http://www-sop.inria.fr/acacia/cours/essi2006/ScientificAmerican_FeatureArticle_TheSemanticWeb_May2001.pdf

[15] P. Mertens, F. Bodendorf, W. König, A. Picot, M. Schumann, and T. Hess, "Daten, Information und Wissen," in *Grundzüge der Wirtschaftsinformatik*, ser. Springer-Lehrbuch. Springer Berlin Heidelberg, 2012, pp. 37–63. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30515-3_3

[16] M. Oestreich and O. Romberg, "Keine Taten ohne Daten!" *Keine Panik vor Statistik!*, pp. 10–28, 2010.

[17] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International journal of human computer studies*, pp. 907–928, 1993. [Online]. Available: http://www.civ.utoronto.ca/sect/coneng/i2c/Civ1283/Civ1283-Ref-Final/Civ1283-AdvancedRef/Ontology/Gruberpaper.pdf

[18] W. Borst, "Construction of engineering ontologies. 1997," *Centre of Telematica and Information Technology, . . .*, 2006.

[19] R. Neches, R. Fikes, T. Finin, and T. Gruber, "Enabling technology for knowledge sharing," *AI magazine*, 1991. [Online]. Available: http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/902

[20] H. Stuckenschmidt, "Beispiele," in *Ontologien Konzepte, Technologien und Anwendungen*, ser. Informatik im Fokus. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 53–90. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-05404-4

[21] ——, "Erstellen von Ontologien," in *Ontologien Konzepte, Technologien und Anwendungen*, ser. Informatik im Fokus. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. 5, pp. 155–205. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-05404-4

[22] T. Berners-Lee and R. Cailliau, "No Title," 1990. [Online]. Available: http://www.w3.org/Proposal

[23] A. Dengel, "Das Resource Description Framework," *Semantische Technologien*, pp. 109–129, 2012.

[24] Wikipedia, "Dokumenttypdefinition Wikipedia." [Online]. Available: http://de.wikipedia.org/wiki/Dokumenttypdefinition

[25] Alfred Moos, "XML-Daten definieren XML Schema," in *XQuery und SQL/XML in DB2-Datenbanken*. Vieweg+Teubner, 2008, pp. 291–313. [Online]. Available: http://dx.doi.org/10.1007/978-3-8348-9556-1_10

[26] R. Burkhard, "Knowledge Visualization: Die nächste Herausforderung für Semantic Web Forschende?" in *Semantic Web*, ser. X.media.press, T. Pellegrini and A. Blumauer, Eds. Springer Berlin Heidelberg, 2006, pp. 201–212. [Online]. Available: http://dx.doi.org/10.1007/3-540-29325-6_13

[27] T. Berners-lee, "Uniform Resource Locators Current practice," www.w3.com, Tech. Rep. July, 1994.

[28] T. Berners-Lee, R. Fielding, and L. Masinter, "RFC 3986," *Uniform Resource Identifier (URI): Generic Syntax*, 2005.

[29] F. Manola and E. Miller, "Resource description framework (RDF) primer," *W3C Recommendation*, vol. 10, 2004.

[30] T. Berners-Lee, "Axioms of Web architecture," *World Wide Web Consortium*, 1998. [Online]. Available: http://www.w3.org/DesignIssues/Metadata.html

[31] Y. Hitzler, P., Krötzsch, M., Rudolph, S., Sure, "Einfache Ontologien in RDF und RDF Schema," in *Semantic Web*, ser. eXamen.press. Springer Berlin Heidelberg, 2008, ch. 3, pp. 33–88. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33994-6_4

[32] D. Brickley, "RDF vocabulary description language 1.0: RDF Schema," *W3C Recommendation*, vol. 10, 2004.

[33] E. PrudHommeaux, A. Seaborne, and Others, "SPARQL query language for RDF," *W3C recommendation*, vol. 15, 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[34] Dr. Prof. h. c. Andreas Dengel, "Anfragesprachen und Reasoning," in *Semantische Technologien Grundlagen Konzepte Anwendungen*, A. Dengel, Ed. Heidelberg: Spektrum Akademischer Verlag, 2012, pp. 161–180. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-8274-2664-2

[35] P. D. P. and Hitzler, , M. M. Krötzsch, , D. S. Rudolph, and D. Y. Sure, "Ontologien in OWL," in *Semantic Web*, ser. eXamen.press. Springer Berlin Heidelberg, 2008, pp. 123–159. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33994-6_6

[36] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 web ontology language primer," *W3C recommendation*, vol. 27, pp. 1–123, 2009.

[37] ——, "OWL 2 web ontology language primer," *W3C recommendation*, 2012.

[38] Thomas A. Runkler, "Visualisierung," in *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*, 2010, pp. 35–55. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2F978-3-8348-9353-6_4

[39] T. Munzner, *Visualization Design and Analysis : Abstractions , Principles , and Methods*, 2012.

[40] B. Preim and R. Dachselt, *Interaktive Systeme*, ser. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-05402-0

[41] C. Moser, "Informationsdesign," in *User Experience Design*, ser. X.media.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 105–120. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-13363-3

[42] R. E. Horn, "Information design: Emergence of a new profession," *Information design*, pp. 15–33, 1999. [Online]. Available: http://www.stanford.edu/~rhorn/a/topic/vl%26id/artclInfoDesignChapter.html

[43] C. und Bosch, S. Schiel, and T. Winder, "Der Prozess der visuellen Wahrnehmung," in *Emotionen im Marketing*, 2006, pp. 361–429.

[44] W. Kroeber-Riel and Weinberg, *Konsumentenverhalten*, 2003.

[45] P. D. G. Raab, D. A. Unger, and P. D. F. Unger, "12.1.1 Begriff und Relevanz," in *Marktpsychologie*, 2010, pp. 169–183.

[46] R. A. Burkhard, "Informationsarchitektur," in *Kompendium Informationsdesign*. Springer Berlin Heidelberg, 2008, pp. 303–320. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69818-0_12

[47] a. V. Moere and H. Purchase, "On the role of design in information visualization," *Information Visualization*, vol. 10, no. 4, pp. 356–371, Sep. 2011. [Online]. Available: http://ivi.sagepub.com/lookup/doi/10.1177/1473871611415996

[48] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=545307http://www.cs.umd.edu/~ben/papers/Shneiderman1996eyes.pdf

[49] D. A. Keim and I. C. Society, "Information Visualization and Visual Data Mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[50] P. Baumann, "Konzeptuelle Modellierung von Geodiensten," *Informatik-Spektrum*, vol. 31, no. 5, pp. 435–450, 2008.

[51] d. L. Professor Dr. Norbert, "Räumliche Objekte und Bezugssysteme," in *Geoinformatik, in Theorie und Praxis*. Springer Berlin Heidelberg, 2006, pp. 159–196. [Online]. Available: http://dx.doi.org/10.1007/3-540-28292-0_5

[52] Dan Brickley, "RDF geo," 2003. [Online]. Available: http://www.w3.org/2003/01/geo/

[53] O. K. Foundation, "Open Data Handbook Documentation," Tech. Rep., 2012. [Online]. Available: http://opendatahandbook.org/pdf/OpenDataHandbook.pdf

[54] ——, "Open Definition." [Online]. Available: http://opendefinition.org/okd/deutsch/

[55] Prof. Dr. Jörn von Lucke Christian P. Geiger, "Open Government Data Frei verfügbare Daten des öffentlichen Sektors Gutachten für die," pp. 6–9„ 2010. [Online]. Available: http://www.zu.de/deutsch/lehrstuehle/ticc/TICC-101203-OpenGovernmentData-V1.pdf

[56] N. Shadbolt, T. Berners-Lee, and W. Hall, "The Semantic Web Revisited," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, May 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1637364

[57] T. Berners-Lee, "Linked Data," 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html

[58] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International Journal on Semantic ...*, vol. 5, no. 3, pp. 1–22, 2009. [Online]. Available: http://www.igi-global.com/article/linked-data-story-far/37496

[59] D. J. Peuquet, "A CONCEPTUAL FRAMEWORK AND COMPARISON OF SPATIAL," vol. 21, no. 4, 1984.

[60] ——, "It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems," *Annals of the Association of american Geographers*, vol. 84, no. April 2013, pp. 441—-461, 1994.

[61] J. L. Mennis and D. J. Peuquet, "A framework for incorporating cognitive principles into geographical database representation," *International Journal of Geographical Information Science*, vol. 14:6, pp. 501–520, 2000.

[62] R. Edsall and D. Peuquet, "A graphical user interface for the integration of time into GIS," in *Proceedings of the 1997 American Congress of Surveying and Mapping Annual Convention and Exhibition, Seattle, WA*, 1997, pp. 182–189.

[63] C. Tominski, P. Schulze-wollgast, and H. Schumann, "Visualisierung zeitlicher Verläufe auf geographischen Karten," pp. 47–57, 2003.

[64] L. Wilkinson, *The Grammar of Graphics*, ser. Statistics and Computing, J. M. Chambers, T. Hothorn, D. T. Lang, and H. Wickham, Eds. Springer, 1999, vol. 49, no. 2. [Online]. Available: http://www.amazon.com/dp/0387245448

[65] M. Voigt and J. Polowinski, "Towards a Unifying Visualization Ontology," {TU} Dresden, Institut f{\textbackslash}textbackslashür Software und Multimediatechnik, Dresden, Germany, Technical Report TUD-FI11-01, 2011.

[66] M. Voigt and S. Pietschmann, "Context-aware recommendation of visualization components," *eKNOW 2012, The Fourth ...*, pp. 101–109, 2012. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=eknow_2012_5_10_60055

[67] O. Gilson, N. Silva, P. Grant, and M. Chen, "From Web Data to Visualization via Ontology Mapping," *Computer Graphics Forum*, vol. 27, no. 3, pp. 959–966, May 2008. [Online]. Available: http://doi.wiley.com/10.1111/j.1467-8659.2008.01230.x

[68] A.-s. Dadzie and M. Rowe, "Approaches to visualising Linked Data : A survey," *Semantic Web*, vol. 2, pp. 89–124, 2011.

[69] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets, "Tabulator: Exploring and analyzing linked data on the semantic web," in *Proceedings of the 3rd International Semantic Web User Interaction Workshop*, vol. 2006, 2006.

[70] S. F. C. De Araújo and D. Schwabe, "Explorator: a tool for exploring RDF data through direct manipulation," in *Linked data on the web WWW2009 workshop (LDOW2009)*, 2009.

[71] J. M. Brunetti, R. Gil, and R. Garc\ia, "Rhizomer: Facets and pivoting for flexible and usable linked data exploration," *Interacting with Linked Data (ILD 2012)*, 2012.

[72] E. Pietriga, "Isaviz: A visual authoring tool for rdf," *World Wide Web Consortium.[Online]. Available: http://www. w3. org/2001/11/IsaViz*, 2003.

[73] S. Goyal and R. Westenthaler, "RDF Gravity (RDF Graph Visualization Tool)," *Salzburg Research, Austria*, 2004.

[74] T. Hastrup, R. Cyganiak, and U. Bojars, "Browsing linked data with Fenfire," 2008.

[75] D. Le Phuoc, A. Polleres, G. Tummarello, and C. Morbidoni, "DERI Pipes: visual tool for wiring Web data sources," Technical report, IDA Business Park, Lower Dangan, Galway, Ireland, Tech. Rep., 2008.

[76] M. G. Skjaeveland, "Sgvizler: A JavaScript Wrapper for Easy Visualization of SPARQL Result Sets," *2012.eswc-conferences.org*, pp. 2–6, 2012. [Online]. Available: http://2012.eswc-conferences.org/sites/default/files/eswc2012_submission_303.pdf

[77] A. de Leon, F. Wisniewki, B. Villazón-Terrazas, and O. Corcho, "Map4rdf-Faceted Browser for Geospatial Datasets," 2012.

[78] S. Auer, J. Lehmann, and S. Hellmann, "Linkedgeodata: Adding a spatial dimension to the web of data," in *The Semantic Web-ISWC 2009*. Springer, 2009, pp. 731–746.

[79] C. Becker and C. Bizer, "Exploring the Geospatial Semantic Web with DBpedia Mobile," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 4, pp. 278–286, Dec. 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1570826809000468

[80] W. Muller and H. Schumann, "Visualization methods for time-dependent data-an overview," in *Simulation Conference, 2003. Proceedings of the 2003 Winter*, vol. 1. IEEE, 2003, pp. 737–745.

[81] M.-J. Kraak and A. Brown, *Web cartography: developments and prospects.* CRC Press, 2001.