Malte Blattmann
Benedikt Elßmann
Semen Gaidenbrik
Edwin Knese
Dustin Kröger
René Lindenberg
Markus Reinisch
Jonathan Schlue

Betreuer: Robert Schädlich
Abgabedatum: 24.06.2017

Entwurfsbeschreibung

Inhaltsverzeichnis

1 Allgemeines

1.1 Allgemeines

Das Semantic Web und seine dazugehörenden Komponenten werden von Tag zu Tag bedeutsamer.
Eine große Aufgabe in der Kooperation zwischen Informatik und Linguistik ist deshalb die Aufbereitung vorhandener Daten, um sie maschinenlesbarer zu machen.
Aus diesem Grund besteht momentan großes Interesse seitens der Linguistic Linked Open Data (LLOD) Community, die bereits existierenden Datensets durch die Datenbank unseres Projekts zu erweitern.
Dazu wird unsere relationale Datenbank in einen RDF-Triple-Store umgewandelt und in die LLOD-Cloud eingehängt.
Die Information soll danach über verschiedene, im folgenden erörterte Wege abgefragt werden können.

2 Produktübersicht

Das Produkt besteht aus einem Frontend und einem Backend.
Das Frontend besteht aus einer überarbeiteten Version der alten Lido-Webseite, durch die die vorhandenen Informationen für Laien zugänglich gemacht werden sollen. Im Rahmen dieses Online-Nachschlagewerkes können linguistische Termini, Konzepte und bibliografische Angaben betrachtet werden.
Zur Arbeit des Backends gehört die Erstellung einer Ontologie, welche das RDF-Schema beschreiben soll und als Wissensbasis dient.
Danach erfolgt die Überführung der relationalen Datenbank in das RDF-Schema (Mapping) mittels Sparqlify.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

3.1 Generelle Idee

Unser Projekt lässt sich in verschiedene Teilprojekte zerlegen, welche durch unterschiedliche Teammitglieder ausgeführt werden.
Das erste Teilprojekt ist das Frontend, welches in der Produktübersicht schon genauer beschrieben wurde.
Das zweite Teilprojekt befasst sich mit dem Erstellen einer Ontologie.
Ein weiteres großes Gebiet ist die Erstellung des passenden Mappings zwischen Datenbank und zuvor erstellter Ontologie, um einen RDF-Triple-Store zu generieren.
Der Triple-Store soll daraufhin durch einen geeigneten Mechanismus in regelmäßigen Abständen aktualisiert werden.
Abschließend sollen die Daten in die bereits bestehende LLOD-Cloud eingebunden werden.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Erstellung eines RDF Triple-Stores

4.1.1 Erstellung einer Ontologie

Um die Erstellung der Ontologie zu erleichtern verwenden wir den Ontologie-Editor Protegé .
Hiermit können OWL-Dateien eingesehen und durch eine grafische Oberfläche visualisiert werden.
Dadurch wird die Sprache RDF abstrahiert und somit die Komplexität verringert.
Dies ist essentiell für die Arbeit mit der OnLit-Ontologie welche von unserer Kundin begonnen wurde und von unserem Team fertig gestellt werden soll.
Die allgemeine Unterteilung einer Ontologie besteht aus verschiedenen Entities.
Hierbei entspricht die Unterteilung in Protegé in die Kategorien Classes, Object properties und Data properties der RDF-spezifischen Unterteilung in Subjekt, Prädikat und Objekt.
Die Classes sind hierbei die verschiedenen Objekte aus der LiDo-SQL-Datenbank, welche nun in der Ontologie beschrieben werden und ihren Primärschlüssel aus der Datenbank als Identifier zugewiesen bekommen.
Weiterhin geben die Object properties die Beziehungen zwischen zwei Classes an. Diese Relation bekommt einen eigenen Namen, sowie eine Beschreibung, und gibt eine Domain an, von welcher Class die Relation ausgehen kann und die Range, also zu welcher Class diese Relation führen kann.
Zudem gibt es noch die Data Properties, welche Beziehungen zwischen Classes und einfachen Literalen angeben. Hier wird als Domain die jeweilige Class angegeben und als Range wird das Literal erfasst.
Die Objekte aus den Classes können mit Hilfe der Individuals instanziiert werden.
Die Ontologie wurde auf Grundlage des Schemas des gegebenen SQL-Dumps entworfen, welches im Folgenden abgebildet ist.

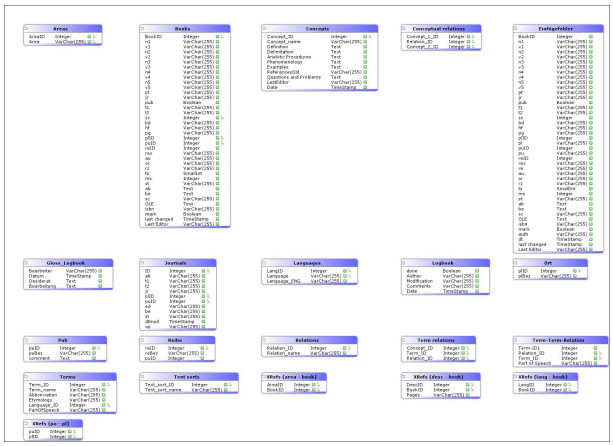

Abbildung 1: Schema d. SQL-Dumps

4.1.2 Mapping mit Sparqlify

Nach der Fertigstellung der Ontologie sollen die bestehenden SQL-Daten mit Hilfe des Tools Sparqlify in das RDF-Format umgewandelt werden. Die Daten liegen uns in einem PostgreSQL-Dump vor. Das Tool selbst unterstützt die Sparqlification Mapping Language, welche für das Mapping des SQL-Dumps auf die RDF-Daten genutzt wird.
Mit Hilfe von Docker erstellen wir mehrere Container, die auf ihre Aufgaben speziell abgestimmt sind:

Der erste Container beinhaltet eine vollständige PostgreSQL-Installation inklusive aller Treiber und dem importierten Dump.
Der zweite Container beinhaltet Sparqlify inklusive aller notwenigen Pakete zum Kompilieren des Tools. Des Weiteren ist der JBDC-Treiber wichtig, welcher die ordnungsgemäße Kommunikation zwischen PostgreSQL und Sparqlify sicherstellt.
Der dritte Docker beinhaltet eine lauffähige Instanz unserer Frontend-Webapplikation, die bereits auf den vom oben genannten Docker bereitgestellten SparQL-Endpunkt zugreift und beispielhaft von dort Daten ausließt und ausspielt.
Ein weiteres Feature von Sparqlify ist die automatische Generierung eines SparQL-Endpunkts, der Teil unserer Aufgabe ist. Dieser wird vom Docker internen Netzwerk nach außen zugänglich gemacht und verbindet sich mit unserer Website.

4.1.3 Microservice-Architektur

Die Administration und Wartung unserer Application steuern wir über die Virtualisierungstechnologien Docker und Docker-Compose. Dabei sind die Teilapplikationen Frontend, Sparqlify-Backend und SQL-Datenbankserver wie oben beschrieben in einzelne, autarke Container separiert. Langfristig sichert diese Modulariserungsmaßnahme ein hohes Maß an Portabilität und Skalierbarkeit zu.

4.2 Frontend

Die Webseite unterteilt sich thematisch in Bibliographical Search und Glossary. Das Navigieren zwischen diesen Ebenen wird durch Tabs umgesetzt.
Auf der Webseite ist es möglich nach linguistischen Konzepten in verschiedenen Sprachen zu suchen. Die Auswahl der Konzepte, sowie Terme erfolgt durch ein Autovervollständigungs-DropDown-Menü wodurch Werte eingegeben bzw. gesucht werden können. Als Ergebnis der Suche folgen Definitionen der Konzepte und Referenzen zu verwandten Themen. Durch Bibliographical Search werden Informationen zu Autoren und Büchern angegeben und es kann nach Büchern zu gewissen Kriterien gesucht werden. Im Allgemeinen werden wir uns bei der Umsetzung am Konzept der bereits bestehenden Webseite orientieren, damit Nutzer sich nicht zu sehr umstellen müssen, allerdings versuchen wir durch verschiedene Änderungen die Benutzerfreundlichkeit zu erhöhen.

Die Datenanbindung geschieht über einen Adapter, der eine Verbindung zu unserem SPARQL-Endpunkt aufbaut und entsprechende Anfragen stellt. So kann auf eine zusätzliche Datenbank verzichtet werden. Die Website wird mit dem Ruby on Rails Framework umgesetzt, wodurch auch maschinenlesbare URIs einfach generiert werden können. Konzepte werden durch eine URI im Format: /c/ID referenziert. Weiter können Terme via: /c/ID/t/ID aufgerufen werden.
Im Folgenden ist ein Bild unseres erneuterten Frontends zu sehen.

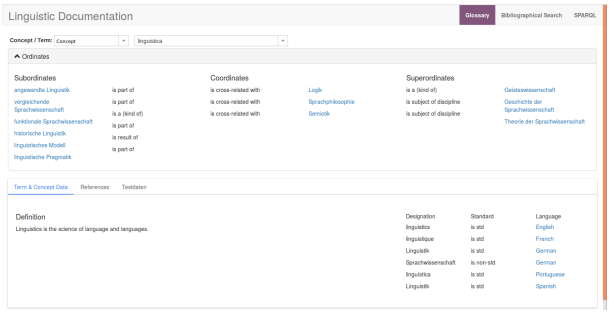

Abbildung 2: Frontend

5 Datenmodell

Das RDF-Modell ist ein Datenmodell mit einer wohldefinierten formalen Semantik, das auf gerichteten Graphen basiert.
In diesem Modell werden Daten in Tripeln, sogenannten Elementaraussagen, strukturiert.
Diese lassen sich weiter in Ressource, Eigenschaft und Aussage unterteilen.

5.1 Ressource

Eine Ressource wird mit RDF-Ausdrücken beschrieben. Folglich muss diese durch eine URI (Universal Resource Identifier) referenzierbar sein und eindeutig von dieser identifiziert werden können.

5.2 Eigenschaft

Eigenschaften sind die Attribute von Ressourcen. Im Datenmodell legen sie die erlaubten Werte, den Typ und die Relation der Ressourcen zu anderen Ressourcen fest.

5.3 Aussage

Eine Ressource, kombiniert mit ihrer Eigenschaft und dem Wert, den die Eigenschaft dieser Ressource hat, nennt man Aussage. Das heißt eine Aussage besteht aus Subjekt-Prädikat-Objekt.

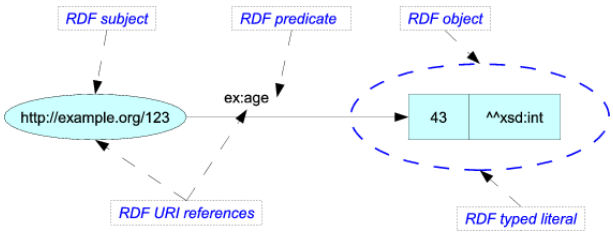

Abbildung 3: Allgemeines RDF-Schema

6 Glossar

Protégé

Bei Protégé handelt es sich um einen Open Source Editor für Ontologien der Stanford University in Kalifornien.

RDF

RDF steht für Resource Description Framework und bildet die Basis zur Verarbeitung von Daten innerhalb des semantischen Webs.

Ruby

Ruby ist eine sehr einfach zu erlernende, dynamische Programmiersprache mit weiten Einsetzungsmöglichkeiten.

SPARQL

SparQL ist eine graphenbasierte Abfragesprache für RDF.

Sparqlify

Ist ein Tool mit dessen Hilfe RDF-Schema auf relationale Datenbanken gemappt wird und danach SPARQL-Abfragen auf dieser ausgeführt werden können.

LLOD Cloud

LLOD Cloud steht kurz für Linguistic Linked Open Data Cloud. (erreichbar unter http://linguistic-lod.org/llod-cloud).

LiDo

Die Linguistic Documentation ist eine Website bzw. Datenbank der Universität Regens- burg.

Ontologie

Ontologien in der Informatik sind meist sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begrifflichkeiten und der zwischen ihnen bestehenden Be- ziehungen in einem bestimmten Gegenstandsbereich.

Semantic Web

Unter dem Konzept des Semantic Webs versteht man die Anreicherung von Webdoku- menten mit maschinenlesbaren Code. Dadurch wird der Austausch und die Verwertbar- keit dieser vereinfacht und die Daten in einen Kontext (für Maschinen) gesetzt.

URI

Ein Uniform Resource Identifier ist eine Zechenfolge, die der eindeutigen Identifikation einer Ressource dient.
Die URI gibt auch Aufschluss über die Art der Ressource.