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

Betreuer: Robert Schädlich
Abgabedatum: 02.05.2017

Arbeitsbericht

1 Projektvision

Die Universität Regensburg stellt ihren Studenten und Mitarbeitern eine sprachwissenschaftliche Datenbank, namens LiDo zur Verfügung, die mit vielen Definitionen und Literaturangaben bei der Recherche und dem wissenschaftlichen Arbeiten weiterhelfen kann. Diese Daten sind jedoch kaum wiederverwendbar, da das bisherige System durch fehlende dereferenzierende Links, keine Zitierbarkeit gewährleistet.
Weiterhin ist jegliche maschinelle Wiederverwendung und interoperable Nutzung der Daten derzeit unmöglich, obwohl ein großes Interesse in der Linguistic Linked Open Data (LLOD) Community darin besteht, bereits existierende Datensets mit den Informationen aus dieser Datenbank semantisch zu verknüpfen.
Ziel ist es daher, im Rahmen des Softwaretechnik Praktikums 2017 die vorhandenen Daten in ein RDF-Datenset umzuwandeln und in der LLOD Cloud bereit zu stellen.
Da die Datenbank stetig erweitert wird, wäre außerdem eine automatische Generierung einer neuen Version des RDF-Datensets in regelmäßigen Zeitabständen sinnvoll. Des Weiteren sollte das RDF-Datenset ebenfalls durch ein Web-Frontend navigierbar sein, welches jedoch die Dereferenzierbarkeit einzelner Dateneinträge berücksichtigt.

2 Voraussetzungen

Damit unsere Software der Projektvision entsprechen kann, werden folgende Vorraussetzungen definiert:

2.1 Teamintern

Jedes Teammitglied muss sowohl in Slack, als auch im Gitlab registriert und mit der Bedienung dieser Tools vertraut sein. Die erforderliche Einrichtung wurde durch den Projektleiter bereits realisiert. Des Weiteren finden wöchentlich zuätzliches Treffen neben den Abgabeterminen statt, um sicherzustellen, dass das Team die gleichen Ziele verfolgt und und alle Aufgaben fristgemäß und vollständig erfüllt werden.
Zur Entwicklung des Frontends benötigen die Teammitglieder grundlegende Kenntnisse in Ruby, sowie dem Framework Ruby on Rails, sowie in Webdesign mit HTML und CSS. Es werden folgenden zuverwendende Versionen festgelegt: Ruby Version: 2.4.0; Rails Version: 5.0.1; zusätzlich wird ein Gem-File zur Verfügung gestellt. Zur Entwicklung des Backends werden unter anderem der Ontologieeditor Protégé und das Mappingtool Sparqlify benutzt.

Alle Teammitglieder erhalten über den Recherechebericht die nötigen Grundkenntnisse, um am Projekt mitwirken zu können. Seitens des Teams wird gefordert, dass jedes Mitglied die nötige Zeit und Motivation aufbringt.

2.2 Nutzer und Entwickler

Nutzer, die bisher mit LiDo gearbeitet haben brauchen keine zusäzlichen Voraussetzungen, um mit der Website in gewohnter Art und Weise zu arbeiten. Unser Entwurf sieht die gleiche Funktionalität mit einem frischeren Design vor. Zusätzlich erhält die LLOD Cloud-Community einen Zugang auf die Daten.
Entwickler, die ähnliche relationale Datenbanksysteme in ein RDF-Schemata überführen möchen können unsere Ontologie und unser Mapping gern als Musterlösung verwenden. Wir weisen jedoch darauf hin, dass jede relationale Datenbank ein eigenes Ontologie-Schemata benötigt um effektiv arbeiten zu können. Wir können als Mappingtool Sparqlify empfehlen. Sparqlify, eine Weiterentwicklung der Universität Leipzig von Triqlify, ist ein Tool, um Mithilfe von SML, einer Mapping-Language relationale Daten in RDF-Stores zu speichern. Wichtig für eine lauffähige Installation ist Oracle-Java. Quelloffene Java-Projekte, wie OpenJDK werden nicht unterstützt.

3 Designübersicht und Funktionalität

3.1 Design

Der Nutzer soll in Anlehnung an die Orginal-Version von LiDo `http://linguistik.uni-regensburg.de:8080/lido/Lido` die Möglichkeit haben:

- **Sprache per Dropdown-Liste**
  die Sprache des gesuchten Begriffs über ein alphabetisch sortiertes Auswahlmenü festzulegen.
- **Stichworte per Dropdown-Liste**
  die auf eine zuvor festgelegte Sprache in der Liste erscheineden Stichworte auszuwählen bzw. deren Unique Identifier in (lateinischer) Herkunfstsprache.
- **Bibliographische Suche**
  Literatur durch Angabe eines Suchbegriffs, Orts, einer Sprache, des Namens des Autors, des Titels, der Art oder des Zeitpunkts der Veröffentlichung und anderen bibliographischen Angaben finden
- **verlinkte Begriffe**
  Begriffe, die die mit anderen in Beziehung stehen, direkt über einen Hyperlink aufrufen können

Die Anwendung soll automatisch bei einem aktuelleren Datensatz den RDF-Dump und die Website aktualisieren. Zudem soll die Möglichkeit bestehen, in diesen Prozess über eine Administrator-Oberfläche händisch einzugreifen.

3.2 Funktionalität

Zur Veranschaulichung der Funktionsweise unseres Projekts hilft die folgende Abbildung 1. Die unten abgebildete Numerierung entspricht zudem den weiter unten definierten Arbeitspaketen.

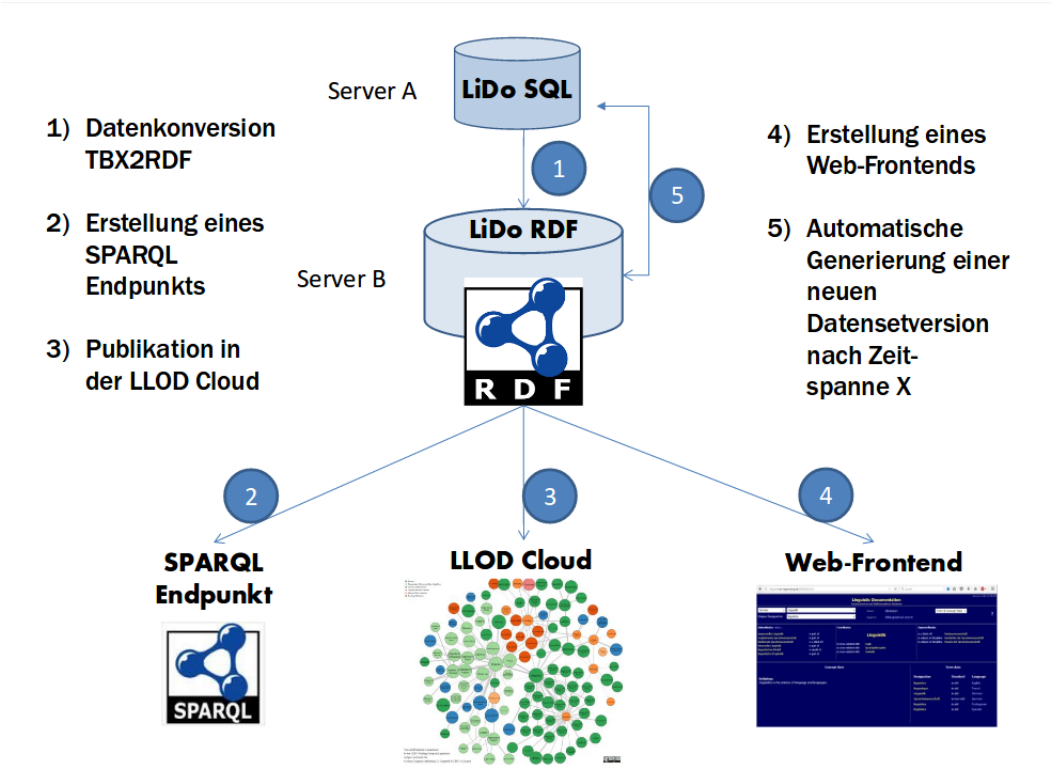

Abbildung 1: Aufbau unseres Projektes

4 Arbeitspakete

4.1 SQL-2-RDF (40%)

Die Konvertierung einer relationalen Datenbank in das RDF-Format setzt sich aus folgenden beiden Aufgaben zusammen:

4.1.1 Ontologie (15%)

Basierend auf der bestehenden Datenbank wird eine Ontologie entwickelt, um alle Daten verlustfrei zu übernehmen. Die unsere Ontologie erweitert das Onlit-Modell, entwickelt von BettinaKlimek bei einem Versuch LiDo in ein RDF-Format umzuwandeln.

4.1.2 Mapping und Umwandlung (25%)

Nach der Fertigstellung der Ontologie sollen die bestehenden SQL-Daten lokal mit Hilfe des Tools Sparqlify und der erstellten Ontologie gemapped und in das RDF-Format umgewandelt werden. Spaqlify unterstützt die Sparqlification Mapping Language - die Herausforderung besteht in dem Mapping des PostgreSQL-Dumps auf die RDF-Daten.

4.2 SPARQL-Endpunkt (10%)

Als öffentliche Schnittstelle soll ein SPARQL-Endpunkt bereitgestellt werden. Dieser kann von Sparqlify selbst, über das Mapping-File, bereitgestellt werden und benötigt den erfolgreichen Abschluss des Kapitels 4.1.

4.2.1 Server Administration

Vorbereitung des Projekt-Servers für die Installation/Kompilieren des Sparqlify Source-Codes mit Maven. Wie bereits Erwähnt ist zwingend Oracle-Java zu verwenden, da es sonst zu unvorhergesehenen Problemen kommt.

4.2.2 Setup Sparqlify

Einrichtung des Sparqlify-Servers als SPARQL-Endpunkt für die RDF-Daten.

4.3 LLOD Cloud (5%)

4.3.1 RDF-Export

Die mit Sparqlify generierten RDF-Daten werden sondiert und für die Veröffentlichung hinterlegt.

4.3.2 Bereitstellung

Nachdem die RDF-Daten den Richtlinien der LLOD Cloud entsprechen, werden sie im Datahub mit den entsprechenden Tags hinzugefügt. Ggfs. Kontaktaufnahme mit John McCrae für die Validierung.

4.4 Web-Frontend (40%)

4.4.1 Angleichung an LiDo

Auf Grundlage des bestehenden Designs der Lido-Website wird ein statisches Template erstellt, das als Mockup die Oberfläche des neuen Web-Frontends veranschaulichen soll. Dabei wird ggf. mit Rücksprache auf eine Vereinfachung der Ansichten und Erhöhung der Benutzerfreundlichkeit geachtet. Folgend wird die Oberfläche mittels Programmlogik in Ruby an die in Arbeitspaket 1 generierte Datenbank angebunden um so auch die Funktionalität wiederherzustellen.

4.4.2 Referenzierbarkeit

Im Gegensatz zum aktuellen Stand der LiDo-Website soll der Inhalt des neuen Web-Frontends maschinenlesbar und referenzierbar sein, sodass Anzeigeergebnisse zu einem späteren Zeitpunkt mittels einer URI wiederhergestellt werden können.

4.5 Re-Import (5%)

4.5.1 Cron-Job

Ein Cron-Job auf dem Projektserver prüft nach einer vordefinierten Zeit auf eine neue Version der SQL-Daten. Diese werden analog zum Arbeitspaket 1, jedoch auf dem Projektserver mittels des Sparqlify CLI konvertiert. Somit wird die Grundlage des SPARQL-Endpunktes, sowie des Web-Frontendes aktualisiert. Abschließend wird eine e-Mail zur Benachrichtigung versandt.

4.5.2 Monitoring

Um Fehler beim Empfang der SQL-Daten, der Konvertierung und Bereitstellung abzufangen und auswertbar zu machen, soll der gesamte Ablauf dokumentiert werden mit einer entsprechenden Fehlerausgabe.

4.5.3 Archivierung

Es soll eine Archivierung eingeführt werden, um auch im Nachhinein alte Stände der RDF-Datenbasis zugänglich zu machen.

4.6 Kann-Ziele

4.6.1 Admin-Oberfläche (10%)

Oberfläche des Web-Frontends, um eine RDB-2-RDF Konvertierung händisch anzustoßen, verschiedene Export-Möglichkeiten, Auswahl einer RDF-Version (Revision des Datensatzes) und weitere Möglichkeiten.

5 Vorprojekt

Im Vorprojekt steht die Arbeitseinteilung der Teammitglieder und Arbeitspaket 1 wurde abgearbeitet. Ferner steht zum 4. Arbeitspaket bereits ein erstes Mockup mit eingeschränkter Funktionalität ohne Programmlogik. Die Recherchephase wurde weitestgehend abgeschlossen und es wurde sich in Tools eingearbeitet, sowie die Grundlagen der Programmiersprache Ruby erlernt. Das Vorprojekt bildet die Basis des ganzen Projektes für eine nahtlose Erweiterung und Vervollständigung.

6 Glossar

CSS Cascading Style Sheets (kurz: CSS) ist eine formale Sprache zum Festlegen des Erscheinungsbild von Dokumenten (hauptsächlich HTML- und XML-Dokumente) in der Informationstechnik.[17][18]. 1

Datahub Datahub ist, ähnlich wie Github für Programmcode, eine Plattform für offene Daten, die unter bestimmten Lizenzen veröffentlicht werden.
Für unser Projekt gilt es die vom LiDo umgewandelte RDF-Daten im Datahub zur Verfügung zustellen.. 4

Framework Ein Programmiergerüst bzw. Framework beschreibt eine oft in der objektorientierten bzw. komponentenbasierten Entwicklung verwendete Programmierunterstützung. Ein Framework dient als Entwurfsmuster und enthält verschiedene Basisbausteine, die in Kombination eine fertige Software ergeben.. 1

HTML Die Hyptext Markup Language ist eine Auszeichnungssprache, die auf Text basiert, um digitale Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten zu strukturieren.. 1

LiDo Die Linguistic Documentation ist eine Website bzw. Datenbank der Universität Regensburg.. 2, 3

LLOD Cloud LLOD Cloud steht kurz für Linguistic Linked Open Data Cloud. (erreichbar unter http://linguistic-lod.org/llod-cloud). 1, 2, 4

Maven Maven ist ein Java-Build-Tool um aus Quellcode eine ausführbare Software zu generieren.. 4

Ontologie Ontologien in der Informatik sind meist sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begrifflichkeiten und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Gegenstandsbereich.. 3

Protégé Bei Protégé handelt es sich um einen Open Source Editor für Ontologien der Stanford University in Kalifornien.. 1

RDF RDF steht für Resource Description Framework und bildet die Basis zur Verarbeitung von Daten innerhalb des semantischen Webs.[27]. 1–5

Ruby Ruby ist eine sehr einfach zu erlernende, dynamische Programmiersprache mit weiten Einsetzungsmöglichkeiten.. 1, 4, 5

SPARQL SparQL ist eine graphenbasierte Abfragesprache für RDF.. 4, 5

Sparqlify Ist ein Tool mit dessen Hilfe RDF-Schema auf relationale Datenbanken gemappt wird und danach SPARQL-Abfragen auf dieser ausgeführt werden können. [24] . 1–5

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.. 4

Literatur

[1] https://de.wikipedia.org/wiki/GitLab

[2] https://en.wikipedia.org/wiki/GitLab

[3] https://about.gitlab.com

[4] https://www.w3.org/standards/

[5] https://de.wikipedia.org/wiki/World_Wide_Web_Consortium

[6] http://www.w3.org/standards/techs/skos

[7] https://de.wikipedia.org/wiki/Simple_Knowledge_Organisation_System

[8] https://de.wikipedia.org/wiki/Syntax

[9] https://de.wikipedia.org/wiki/Semantik

[10] https://www.java.com/de/download/faq/whatis_java.xml

[11] https://de.wikipedia.org/wiki/Java_(Programmiersprache)

[12] SWT Vorlesung WS 2016 - Professor Dr.-Ing. Klaus-Peter Fähnrich, Dr. Michael Martin, Roy Meissner

[13] https://daringfireball.net/projects/markdown/

[14] https://www.ruby-lang.org/de/about/

[15] https://jekyllrb.com/docs/home/

[16] https://de.onpage.org/wiki/Framework

[17] https://de.wikipedia.org/wiki/Cascading_Style_Sheets

[18] https://de.wikipedia.org/wiki/Stylesheet-Sprache

[19] https://de.wikipedia.org/wiki/Ajax_(Programmierung)

[20] https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[21] https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure

[22] https://de.wikipedia.org/wiki/Turtle_(Syntax)

[23] https://de.wikipedia.org/wiki/Semantic_Web

[24] http://aksw.org/Projects/Sparqlify.html

[25] https://en.wikipedia.org/wiki/Triplestore, Zugriff 23.04.2017

[26] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure, Semantic Web Grundlagen, Erste Auflage 2008, Seite 40

[27] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure, Semantic Web Grundlagen, Erste Auflage 2008, Seite 36

[28] https://dbs.uni-leipzig.de/file/dbs1-ws1617-kap5-half.pdf

[29] http://www.informatik.uni-leipzig.de/alg/lehre/ws12_13/DISK/folien.pdf Seite 40

[30] http://www.duden.de/rechtschreibung/Linguistik

[31] https://de.wikipedia.org/wiki/Abfragesprache

[32] http://www.gruenderszene.de/lexikon/begriffe/html

[33] https://de.wikipedia.org/wiki/Ontologie_(Informatik)

[34] http://www.itwissen.info/Metadaten-meta-data.html

[35] https://www.w3.org/TR/rdf-sparql-query/

[36] http://scrum-master.de/Scrum-Einfuehrung

[37] https://www.w3.org/DesignIssues/LinkedData.html

[38] https://de.wikipedia.org/wiki/Web_Ontology_Language

[39] http://stackoverflow.com/questions/4913343/what-is-the-difference-between-uri-url-and-urn

[40] https://www.w3.org/OWL/