

LinkedSpending

SWP14-LS

- Projektangebot -

Inhaltsverzeichnis

1. Zielbestimmung	3
2. Voraussetzungen	3
3. Designübersicht und Funktionalität	3
3.1 Design	3
3.2 Funktionalität	4
4. Arbeitspakete und Meilensteine	4
4.1 Arbeitspakete	4
4.2 Meilensteine	5
5. Qualitätssicherung	5
6. Glossar	6

1. Zielbestimmung

OpenSpending ist ein Projekt mit dem Ziel Finanzdaten über Regierungs- und Unternehmensausgaben zu sammeln und ansprechend zu visualisieren. LinkedSpending verknüpft diese Datenmenge mit dem SemanticWeb indem es diese in das RDF-Format überführt wodurch die Zusammenhänge besser verarbeitet und somit komplexere Visualisierungen automatisch erstellt werden können.

Ein Prototyp ist für diese Aufgabe schon vorhanden, dieser weist allerdings noch einige Schwachstellen auf. Die Aufgabe der Projektgruppe besteht darin auf dieser Grundlage aufzubauen:

- die Daten des OpenSpending mittels LIMES verknüpfen
- die Qualität dieser Links auf Dauer gewährleisten
- die Daten des OpenSpending regelmäßig automatisch synchronisieren
- Logs über den Konvertierungsvorgang erstellen
- Nutzerstatistiken des SPARQL Endproduktes auszuwerten und anzubieten

2. Voraussetzungen

Das LinkedSpending Projekt baut auf schon vorhandenen Datensätzen des OpenSpending auf. Hierbei kann weder die ständige Verfügbarkeit noch die vollständige Konsistenz aller Daten gewährleistet werden. Entsprechend fehlertolerant muss LinkedSpending konzipiert sein.

Die verarbeiteten Daten werden in einer Datenbank gespeichert, die entsprechende Datenbankanbindung wird also benötigt. Die Datenausgabe wird über einen SPARQL-Endpunkt realisiert.

3. Designübersicht und Funktionalität

3.1 Design

Rollen:

- Benutzer
- Server-Administrator
- Daten-Administrator

Anwendungsszenarien:

- Benutzer (OntoWiki-Betreiber)
 - ...ruft Daten ab:
Der Benutzer stellt über das SPARQL-Endpunkt eine Anfrage an das System und erhält eine aufbereitete Form der passenden Datensätze oder Relationen zurück.
 - ...visualisiert Daten:
Der Benutzer stellt über den SPARQL-Endpunkt eine Anfrage an das System indem er zueinanderpassende Daten auswählt und erhält eine Graphik welche die Relation der ausgewählten Daten informativ aufarbeitet.

- Server-Administrator
 - ...aktualisiert automatischen Datenabgleich:
Der Server-Administrator verändert die Konditionen, unter denen die OpenSpending-Daten automatisch heruntergeladen und konvertiert werden.
 - ...gleicht Datenset manuell ab:
Der Server-Administrator führt ein Aktualisierungsprogramm auf dem Projekt-Server aus.
- Daten-Administrator
 - ...führt Interlinking aus:
Der Daten-Administrator kreiert neue Verlinkungen zwischen vorhandenen Datensets oder mit anderen RDF-Wissensbasen.
 - ...prüft Daten auf Konsistenz:
Der Daten-Administrator prüft neue Datensets auf ihre Qualität und formale Übereinstimmung mit vorhandenen Daten und nimmt gegebenenfalls Anpassungen vor.

3.2 Funktionalität

Automatische Aktualisierung:

Das Programm wird automatisch in regelmäßigen Abständen aktualisiert.. Es werden neue Datasets aus dem OpenSpending-Projekt heruntergeladen und automatisiert in RDF umgewandelt.

Interface:

Eine REST-Schnittstelle ermöglicht es ein Webinterface anzubinden. Dieses ermöglicht es Benutzern Anfragen zu stellen woraufhin die aufgearbeiteten Daten in verständlicher Form ausgegeben werden. Dies schließt Visualisierungen über CubeViz mit ein.

Optimierter Code:

Der Code von LinkedSpending ist auf Fehlertoleranz und Zeitaufwand hin optimiert.

Interlinking:

Die Daten sind untereinander sinnvoll verlinkt sodass Anfragen schnell und umfassend bearbeitet werden können. Außerdem besteht die Möglichkeit die Daten mit externen Wissensbasen zu verknüpfen.

4. Arbeitspakete und Meilensteine

4.1 Arbeitspakete

Einarbeitung in die vorhandene Codebasis (15%):

Da es schon einen Prototypen gibt besteht der erste Schritt das Vorhandene zu überschauen, aufzuarbeiten und anzupassen. Diesen Prototypen in einen lauffähigen und ausbaufähigen Zustand zu überführen ist teilweise Thema des Vorprojektes.

Interlinking (25%):

Ziel ist die vorhandenen Daten zu analysieren und Muster zu entwerfen mit denen zueinander passende Datensets automatisch oder manuell ausfindig gemacht werden können. Diese sollen dann mittels LIMES miteinander verknüpft und in LinkedSpending integriert werden.

Visualisierung (25%):

Es wird eine REST-Schnittstelle bereitgestellt mittels welcher die vorhandenen Daten abgefragt werden können. Diese können nun entweder direkt mittels CubeViz dargestellt oder aber über Vorfilter, welche geeignete Dimensionen zur Visualisierung erkennen, vorher aufbereitet werden.

automatische Aktualisierung (15%):

Es muss ermittelt werden wie oft und wie viele der Daten automatisch abgeglichen werden sollten. Die automatische Aktualisierung sollte hierbei automatisch auf etwaige Verbindungsprobleme zum OpenSpending reagieren und außerdem manuell ausführbar sein.

Optimierung (10%):

Die Datenausgabe muss auf Geschwindigkeit optimiert sein, die automatische Aktualisierung darf nicht zu starken Leistungseinbußen führen, nicht konsistente Daten von OpenSpending dürfen nicht den gesamten Datenbestand beschädigen.

Interface (optional) (15%):

Eine ansprechende graphische Benutzeroberfläche für Endnutzer und Administratoren kann zusätzlich entwickelt werden.

Einbindung in das OntoWiki (optional) (10%):

Es kann eine Schnittstelle für das OntoWiki bereitgestellt werden.

Dokumentation (10%):

Die Dokumentation beschreibt das erwünschte Eingabeformat, die REST-Schnittstelle, gegebenenfalls das Interface und sonstige relevante Informationen sowohl für Endnutzer als auch für Administratoren.

4.2 Meilensteine

1. Meilenstein:

Die Rahmenbedingungen und Aufgaben für das Projekt sind bekannt, Rollenverteilung, Risikoanalyse, Recherchebericht, Qualitätssicherungskonzept und Projektangebot sind erstellt. Das Team kennt die Anforderungen und befasst sich nun mit der konkreten Modellierung und Implementierung des Projektes.

2. Meilenstein:

Die relevanten Module des Projektes sind voll funktionsfähig. Offene Aufgaben umfassen hauptsächlich Optimierung, Qualitätssicherung und Dokumentation, gegebenenfalls auch optionale Module.

5. Qualitätssicherung

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität		x		
Zuverlässigkeit	x			
Benutzbarkeit			x	
Effizienz		x		
Änderbarkeit	x			
Übertragbarkeit			x	

6. Glossar

CubeViz:

CubeViz ist ein Browser um statistische Daten über das RDF Data Cube Vokabular, kompatibel mit SDMX, interaktiv darzustellen.

Dataset:

Ein Dataset bezeichnet eine größere, zusammenhängende Datenmenge und besteht aus Metadaten welche den Inhalt beschreiben und der dazugehörigen Sammlung von Einträgen.

LIMES (Link Discovery Framework for Metric Spaces):

LIMES ist ein Framework zum Auffinden von Links im Datennetz.

OntoWiki:

Das Ontowiki ist ein semantisches Programm für Wissensmanagement im Semantic Web Kontext.

RDF (Resource Description Framework):

Das Resource Description Framework bezeichnet eine technische Herangehensweise im Internet zur Formulierung logischer Aussagen über. Im RDF-Modell besteht jede Aussage aus den drei Einheiten Subjekt, Prädikat und Objekt.

REST (Representational State Transfer):

REST bezeichnet ein Programmierparadigma für Webanwendungen welche besagt dass eine URL genau einen Seiteninhalt als Ergebnis einer serverseitigen darstellt.

OpenSpending:

OpenSpending ist eine offene Plattform zum Datenaustausch welche versucht alle Transaktionen zwischen Regierungen und Unternehmen aufzuzeichnen und visuell zu präentieren.

SemanticWeb:

Das Semantic Web ist eine Instanz von semantischen Netzen und außerdem eine Erweiterung des WWW. Ziel ist es die Bedeutung von Informationen für Computer verwertbar zu machen und damit automatisch für die interessierten Nutzer im Zuge einer Abfrage zu ordnen.

SPARQL (SPARQL Protocol and RDF Query Language):

SPARQL ist eine graph-basierte Abfragesprache für RDF. Mit SPARQL ist es möglich komplexere Anfragen zu stellen als es reine Textsuchen ermöglichen würden.