Projektangebot

swp-helios

7. Februar 2014

Inhaltsverzeichnis

1 Zielbestimmung

Das Ziel dieses Projektes ist es eine Software zu entwickeln, die Pläne zur Ausführung einer Linkspezifikation im Browser darstellen kann, sowie einen möglichst effizienten Plan vorauswählen kann. Besonderer Augenmerk soll darauf liegen einen möglichst effizienten Plan auszuwählen. Der Benutzer soll dabei das Programm über eine Weboberfläche bedienen und den Plan auch manuell ändern können.

2 Voraussetzungen

Eine der wichtigsten Aufgaben des Internets ist die Vermittlung von Informationen. Das Konzept des Semantischen Webs soll in Anbetracht der heute unüberschaubaren Informationsvielfalt die Kategorisierung und Zuordnung von Informationen und Daten im Internet verbessern, indem es Daten mit Bedeutungen versieht und diese so besser sortierbar und auch für Maschinen verwertbar macht. Dadurch ist es möglich, Themen in Breite und Tiefe zu erweitern und neue, interessante Zusammenhänge zwischen ehemals getrennten Daten zu finden. Mit der Entstehung des Semantic Webs haben sich allerdings viele kleine und große Wissensbasen mit verknüpften Daten zu unterschiedlichen Sachgebieten und Themenbereichen gebildet, die jedoch auch untereinander verknüpft werden müssen, damit eine möglichst große Informationsvielfalt erreicht werden kann. Da naive Ansätze zur Verknüpfung von Informationen zwischen Wissenbasen durch die große Menge der schon vorhandenen RDF-Tripel zu zeitintensiv sind, wird deswegen nach alternativen Algorithmen gesucht, die schnellere Verlinkung ermöglichen. Unser Programm soll die Verlinkung von Wissensbasen beziehungsweise Teilen dieser unterstützen, indem es den schnellsten zur Verfügung stehenden Ausführungsplan zur spezifizierten Verlinkung findet und anzeigt.

Voraussetzungen für das Projekt ist die Bereitstellung von LIMES und dessen Nutzung im Rahmen des Projekts, dazu Platz auf einem vernetzen Server, um das Programm online abrufen zu können.

3 Designübersicht und Funktionalität

Hauptbestandteil ist ein Java-Programm zur Berechnung von effektiven Ausführungsplänen für Linkspezifikationen mithilfe eines von uns gefundenen Algorithmus, hinzu kommt eine mit JavaScript geschrieben im Browser aufrufbare Benutzeroberfläche zur einfachen Bedienung des Programms. Das Programm wird dabei auf dem Server ausgeführt und visualisiert die Ergebnisdaten als Graphen, die im Browser des Benutzers angezeigt werden, nachdem der Benutzer eine Anfrage gesendet hat.

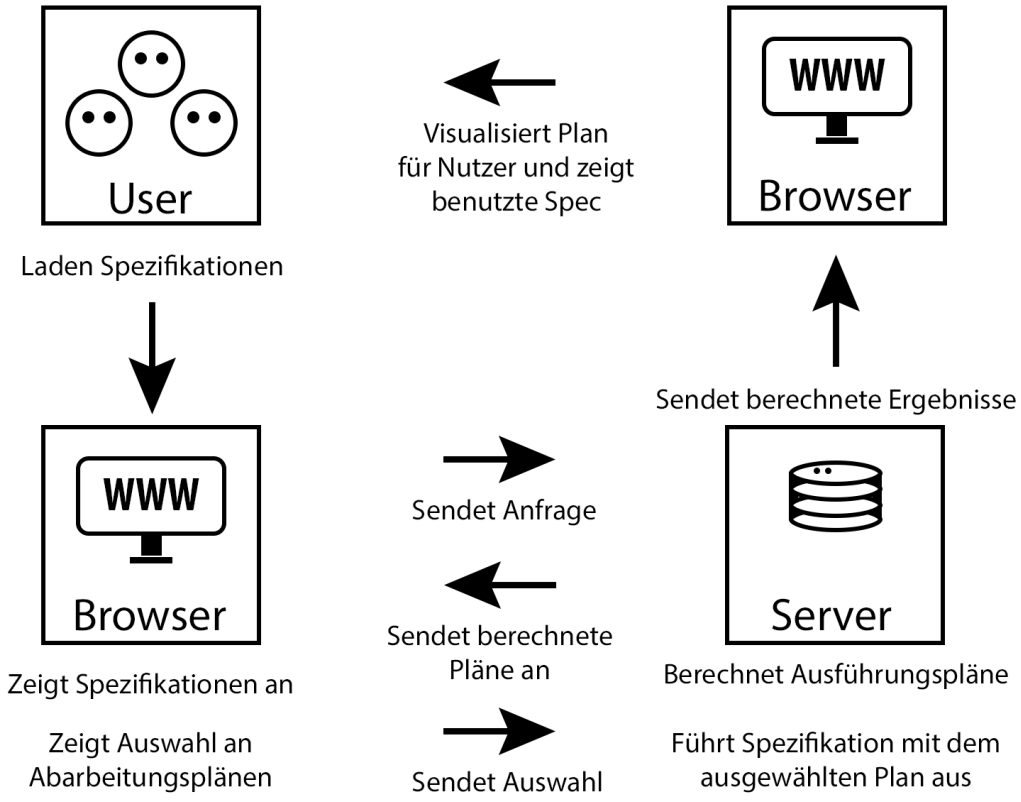

Abbildung 1: Prinzipielles Ablaufdiagramm

3.1 Funktionalitäten

/LF11/Geschäftsprozess: Zeit messen
Beschreibung: Der Benchmark soll nach der Auswahl eines Planungsalgorithmus' die benötigte Zeit zum Finden eines geeigneten Plans ausgeben und diese auch abspeichern. Somit können mit den gesammelten Daten später Vergleiche mit unserem Algorithmus ausgeführt werden, wodurch wir so über dessen Effizienz urteilen können.

/LF13/Geschäftsprozess: Ausführungsplan finden
Beschreibung: Beschreibung: Unser Algorithmus soll nach Eingabe einer LinkSpec mit geringem Zeitaufwand einen möglichst schnellen Ausführungsplan finden.

/LF17/Geschäftsprozess: Algorithmus ausführen
Beschreibung: Wir werden ein Java-Programm schreiben, in welchem der Algorithmus ausgeführt wird. Dieses benötigt hier keine eigene Oberfläche und wird nur über die Shell bedient. Während der Phase der Algorithmusfindung, ist dieses Programm vor allem für Tests gedacht und soll nur den Algorithmus ausführen können. Später wird das Programm angepasst, um den Algorithmus auf dem Server auszuführen.

/LF19/Geschäftsprozess: Datensätze einlesen
Beschreibung: Der Benutzer referenziert Datensätze, die unter Verwendung des LIMES-Frameworks eingelesen werden. Dies wird von der Benutzeroberfläche bereitgestellt.

/LF23/Geschäftsprozess: Manuell Pläne wählen
Beschreibung: Dem Benutzer soll es möglich sein, einen Ausführungsplan direkt auszuwählen. Dabei soll er die Reihenfolge manuell ändern können, durch Verschieben der Planteile in der Benutzeroberfläche. Fehlerhafte Eingaben werden dabei abgefangen.

/LF29/Geschäftsprozess: LinkSpecs eingeben
Beschreibung: Der Benutzer muss LinkSpecs eingeben können, damit die Datensätze, die ebenfalls vom Benutzer angegeben werden, sinnvoll verglichen werden können. Dies wird von der Benutzeroberfläche bereitgestellt.

/LF31/Geschäftsprozess: Pläne ausführen
Beschreibung: Beschreibung: Die Pläne werden unter Verwendung des LIMES-Frameworks ausgeführt.

4 Arbeitspakete und Meilensteine

4.1 Benchmark erstellen

Ein wichtiger Punkt der Beurteilung des Algorithmus ist es gute Benchmarks zu finden um ihn mit anderen Ansätzen vergleichen zu können. Bei diesem Arbeitspaket geht es darum Daten und Linkspezifikationen zu finden um den Algorithmus zu testen, sowie einen Weg zu finden diese Tests zu automatisieren. Dieses Arbeitspaket wird im Vorprojekt bearbeitet.
Muss: Statistiken aufstellen zur einfachen Vergleichbarkeit mit mindestens 10 LinkSpecs
Kann: Automatisierung des Benchmarking-Prozesses
zeitlicher Aufwand: 10%

4.2 Algorithmus konzipieren

In diesem Arbeitspaket geht es darum einen Algorithmus zu finden, der einen möglichst effizienten Plan zum Ausführen einer gegebenen Linkspezifikation findet. Wir planen hier zwei verschiedene Lösungsansätze zu verfolgen und sie im Laufe des Projekts wiederholt miteinander zu vergleichen. Der Fokus liegt dabei darauf, eine stabile, gut funktionierende Lösung zu finden, an zweiter Stelle steht der Anspruch an eine gute Geschwindigkeit.
Muss: Heuristik für die Größe der Ergebnismengen und Kosten aufstellen
zeitlicher Aufwand: 40%

4.3 Algorithmus implementieren

Hier geht es darum, den zuvor erdachten Algorithmus korrekt und wohldokumentiert zu implementieren. Das Ergebnis ist ein in der Konsole lauffähiges und per Tastatur bedienbares Java-Programm.
zeitlicher Aufwand: 30%

4.4 Oberfläche erstellen

Zuletzt folgt die Einbettung des fertiggestellten Programms in ein Webinterface auf JavaScript-Basis, um eine benutzerfreundliche Bedienung zu gewährleisten und der Programmumgang für den Endbenutzer möglichst komfortabel und verständlich wird. Benutzerfreundlichkeit und Bedienumfang sind hier die Hauptgesichtspunkte.
Muss: Pläne eindeutig visualisieren
zeitlicher Aufwand: 20%

4.5 Algorithmus/Implementierung optimieren

Sofern alle anderen Arbeitspakete zufriedenstellend abgeschlossen sind, kann zusätzlich an weiterer Optimierung des Algorithmus, der Implementie-

rung oder zusätzlichen, nicht unbedingt notwendigen Funktionen der Weboberfläche gearbeitet werden.
zeitlicher Aufwand: 30%

5 Qualitätssicherung

Funktionalität des Algorithmus und Bedienbarkeit der Benutzeroberfläche für einen reibungslosen Programmablauf stehen mit an erster Stelle, zudem die Effizienz des Planungsalgorithmus, ein zu langsames Ergebnis wäre unbenutzbar. An Zuverlässigkeit und Änderbarkeit werden keine weiteren Anforderungen gestellt.

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

6 Glossar

6.1 Benchmarking

Bewertungs- und Messverfahren zur Beurteilung der Performance. Benchmarks dienen zum Vergleich der Leistungsfähigkeit verschiedener Systeme.

6.2 GUI

Graphical User Interface, dt. grafische Benutzeroberfläche, die die Bedienung eines Programms für unerfahrene Anwender vereinfacht und die ausführbaren Programmoptionen und Parameter übersichtlich und geordnet, dazu möglichst schön auf dem Bildschirm anzeigt.

6.3 JavaScript

Skriptsprache zur Auswertung von Benutzerinteraktion im Webbrowser.

6.4 Java

Weitverbreitete objektorientierte Programmiersprache, die plattformunabhängig genutzt werden kann und auch mit Hilfen von JavaScript in Webanwendungen eingebettet werden kann.

6.5 LIMES

Limes ist ein Framework, welches Links im Datennetz auffindet. Es stellt Methoden für eine performante und umfangreiche Linkermittlung innerhalb eines metrischen Raumesbereit (der gleichnamige Algorithmus führt die Berechnungen im metrischen Raum aus). Mittels Web-Interface ist es leicht konfigurierbar. Auch kann es als eigenständiges Tool aus dem Internet herunterladen werden, um Links lokal zu ermitteln. Die Ergebnisse werden als RDF oder TSV Daten ausgegeben.

6.6 Linkspezifikation/Link Spec

Beschreibung der Mengen und deren Ähnlichkeitskriterien von zwei verschiedenen RDF-Ressourcen, die verknüpft werden sollen.

6.7 RDF

Resource Description Framework, ein System zur Beschreibung von Resourcen im Internet. Es wird benutzt, um Daten Bedeutungen zuzuweisen, indem aus einem Subjekt (Ausgangspunkt), einem Prädikat (Relationstyp) und einem Objekt (Endpunkt) ein RDF-Tripel gebildet wird, dass zwei Datenpunkte und deren Beziehung zueinander darstellt. Durch Verknüpfung

eines Subjekts zu verschiedenen Objekten und mit verschiedenen Beziehungen entsteht so ein semantisches Netz.