

Projektvertrag

swp-helios

22. April 2014

Inhaltsverzeichnis

1	Einleitung	1
2	Zielbestimmung	2
3	Funktionalität und Arbeitspakete	2
3.1	Funktionalitäten	2
3.2	Arbeitspakete	3
3.2.1	Benchmark erstellen	3
3.2.2	Algorithmus konzipieren	4
3.2.3	Algorithmus implementieren	4
3.2.4	Oberfläche erstellen	4
3.2.5	Algorithmus/Implementierung optimieren	4
4	Qualitätssicherung	4
5	Designübersicht	5
6	Glossar	5

1 Einleitung

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.

Voraussetzung für das Projekt ist die Bereitstellung von LINES und dessen Nutzung im Rahmen des Projekts, dazu Platz auf einem vernetzten Server, um das Programm online abrufen zu können. Desweiteren müssen die Projektmitglieder den Umgang mit LINES beherrschen und die Programmiersprache Java und die Scriptsprache JavaScript benutzen können.

2 Zielbestimmung

Das Ziel dieses Projekts ist es, eine Software zu entwickeln, die Pläne zur Ausführung einer Linkspezifikation im Browser darstellen, sowie einen zeiteffizienten Plan vorauswählen kann. Besonderes 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.

3 Funktionalität und Arbeitspakete

Hauptbestandteil ist ein Java-Programm zur Berechnung von schnellen Ausführungsplänen für Linkspezifikationen mithilfe eines von uns programmierten genetischen Algorithmus. Hinzu kommt eine im Browser aufrufbare Benutzeroberfläche zur einfachen Bedienung des Programms.

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: 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: Die Pläne werden unter Verwendung des LIMES-Frameworks ausgeführt.

3.2 Arbeitspakete

3.2.1 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%

3.2.2 Algorithmus konzipieren

In diesem Arbeitspaket geht es darum, einen Algorithmus zu finden, der einen möglichst effizienten Plan zum Ausführen einer gegebenen Linkspezifikation findet. Es wird ein genetischer Algorithmus verwendet, der eine lokale Suche auf dem Raum aller möglichen und korrekten Pläne für die gegebene Metrik ausführt.

Muss: Heuristik für die Größe der Ergebnismengen und Kosten aufstellen; Hohe Geschwindigkeit und Stabilität

zeitlicher Aufwand: 40%

3.2.3 Algorithmus implementieren

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

zeitlicher Aufwand: 30%

3.2.4 Oberfläche erstellen

Der Benutzer wird das Programm über ein Webinterface auf Apache-Wicket-Basis bedienen, welches möglichst benutzerfreundlich und verständlich aufgebaut sein wird.

Muss: Pläne als Graphen visualisieren

zeitlicher Aufwand: 20%

3.2.5 Algorithmus/Implementierung optimieren

Sofern alle anderen Arbeitspakete zufriedenstellend abgeschlossen sind, kann zusätzlich an weiterer Optimierung des Algorithmus, der Implementierung oder zusätzlichen, nicht unbedingt notwendigen Funktionen der Weboberfläche gearbeitet werden.

zeitlicher Aufwand: 30%

4 Qualitätssicherung

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

5 Designübersicht

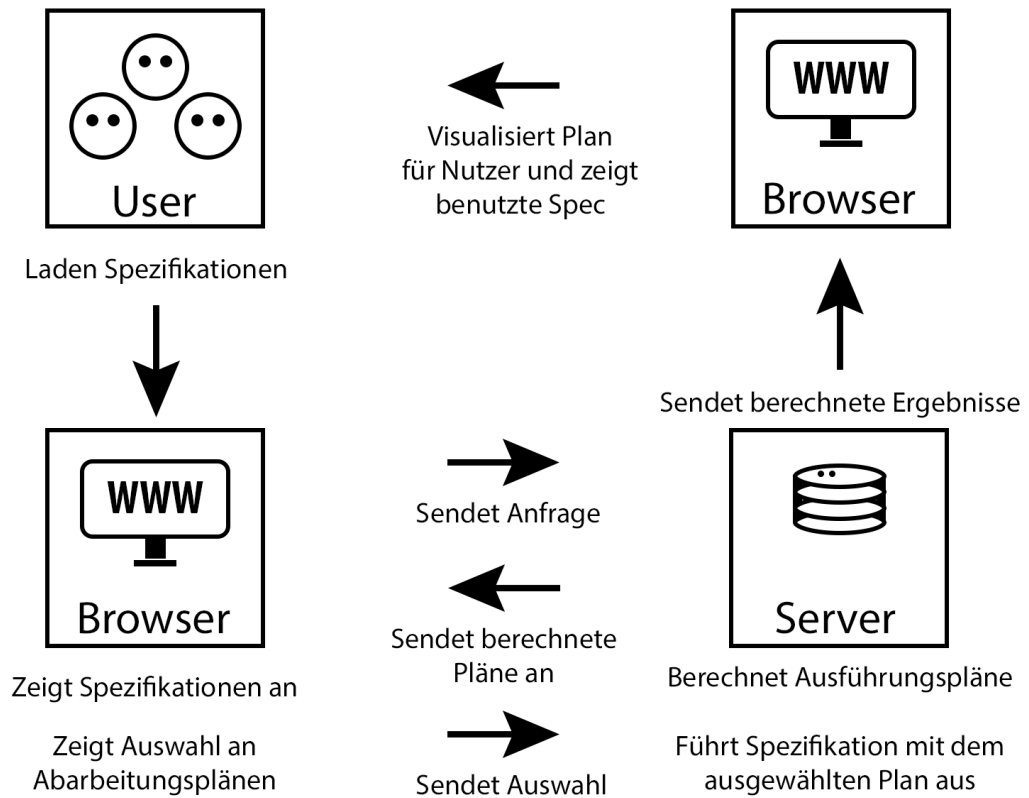


Abbildung 1: Prinzipielles Ablaufdiagramm

6 Glossar

Apache Wicket

Apache Wicket ist ein komponentenbasiertes objektorientiertes Web-Framework für Java. Hauptgedanke des Frameworks ist die Trennung von in Java geschriebener Logik und der Darstellung in HTML bzw. CSS.

Benchmark

Ein Benchmark ist eine Vergleichende Analyse. In unserem Fall sollen Algorithmen verglichen werden. Dabei ist besonders die Performance entscheidender Bewertungsfaktor.

Genetischer Algorithmus

Heuristik zur Lösung von Entscheidungsproblemen. Genetische Algorithmen dienen der Lösung von Optimierungsaufgaben.

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.

LIMES

Limes ist ein Framework, welches Links im Datennetz auffindet. Es stellt Methoden für eine performante und umfangreiche Linkermittlung innerhalb eines metrischen Raumes bereit (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 heruntergeladen werden, um Links lokal zu ermitteln. Die Ergebnisse werden als RDF oder TSV Daten ausgegeben.

Link Discovery

Das Auffinden von Links zwischen Wissensbasen im Semantic Web bezeichnet man als Link Discovery.

Linkspezifikation

Beschreibung der Ähnlichkeitskriterien zweier verschiedener RDF-Ressourcen zum Zweck der Verknüpfung.

Planungsalgorithmus

Ein Planungsalgorithmus ist ein Algorithmus, der aus einem Abhängigkeitsgraph von Arbeitsschritten eine die Abhängigkeiten nicht verletzende lineare Abfolge der Arbeitsschritte erstellt.