



LinkedGeoData.org

— Entwurfsbeschreibung —

Von: Martin Grohmann TA

Abgabe: 10. April 2012

Gruppe: SWP12-11



Inhaltsverzeichnis

1	Allgemeines	3
2	Produktübersicht	3
2.1	OpenStreetMap	3
2.2	OpenLayers	3
3	Grundsätzliche Struktur- und Entwurfsprinzipien des Gesamtsystems	4
4	Grundsätzliche Struktur- und Entwurfsprinzipien einzelner Pakete	4
4.1	dataMapPoint.js	5
4.2	dataTree.js	5
4.3	dropdown.js	5
4.4	global.js	5
4.5	gui.js	5
4.6	linking.js	5
4.7	map.js	5
4.8	mapPoint.js	5
4.9	sparqlLib.js	6
4.10	tree.js	6
4.11	treeEvent.js	6
4.12	username.js	6



Die Entwurfsbeschreibung enthält alle wichtigen Informationen zu den Struktur- und Entwurfsprinzipien der LinkedGeoData-Erweiterung zur Verlinkung von zwei SPARQL-Endpunkten.

1 Allgemeines

Das von uns zu erstellende Softwareprodukt soll eine Benutzerschnittstelle zur Verlinkung von Daten aus zwei SPARQL-Endpunkten bilden. Da die eigentliche Verknüpfung durch einen Algorithmus, welcher sich noch in Bearbeitung befindet, geschehen soll, konzentriert sich dieses Projekt hauptsächlich auf die Benutzeroberfläche (GUI) und die benutzerseitige Kontrolle der Verknüpfungsergebnisse. Hierfür bestehen in erster Linie besondere funktionale Anforderungen an die GUI.

2 Produktübersicht

In diesem Abschnitt wird auf die äußerlichen Funktionsmerkmale des Softwareproduktes eingegangen.

Das Softwareprodukt wird als Webseite in einem Internetbrowser angezeigt, welche mit JavaScript dynamisch verändert wird. Die Webseite gliedert sich in folgende Bereiche:

2.1 OpenStreetMap

Die Grundlage des Bildschirminhaltes bildet eine OpenStreetMap ¹

2.2 OpenLayers

Am linken und rechten Bildschirmrand sind Bäume aus zwei Wissensbasen über die Map gelegt. In ihnen wählt der Benutzer später die entsprechenden Kategorien aus. Die Wissensbasen können über ein Dropdown Menü am oberen Bildschirmrand ausgewählt werden.

Weitere Elemente des oberen Bildschirmrandes sind ein User Login, ein Button 'Get Links' um zu Verlinkende Daten aus den Wissensbasen präsentiert zu bekommen, und ein Schalter um sich schon gespeicherte Links anzeigen zu lassen.

Das linke der beiden unteren Felder enthält die Daten aus den SPARQL Abfragen aus den Wissensbasen. Dabei wird eine Liste von Link-Paaren angegeben. Die Paare beinhalten jeweils einen Punkt aus den zwei Wissensbasen. Als „Punkte“ sind hier Einträge in den Wissensbasen gemeint; zum Beispiel auf Basis der Koordinaten.

Des Weiteren werden die Paare auf der Karte angezeigt. Daten aus der einen Wissensbasis bekommen einen roten Punkt auf der Karte, Daten aus der anderen Wissensbasis einen grünen Punkt. Die Verlinkung wird mittels einer Geraden zwischen den beiden Punkten dargestellt.

¹<http://openstreetmap.org>



Wenn der Nutzer auf das Link-Paar in der Liste klickt wird der Kartenausschnitt automatisch auf die beiden zugehörigen Punkte auf der Karte zentriert.

Für die Nutzer Interaktion in Bezug auf die Rückmeldung der Verlinkungen sind die Buttons auf der unteren rechten Seite gedacht. In diesem befinden sich die Schaltflächen für „Accept Link“, „Ignore Link“ und „Reject Link“. Die Informationen aus diesen drei Buttons werden dann gespeichert. Ein weiterer Button „resolve URL“ ist dafür da um sich alle Informationen aus den Wissensbasen zu den markierten Punkten anzeigen zu lassen. Dazu öffnen sich zwei neue Tabs mit der entsprechenden URL.

Um die erforderliche Speicherung durchzuführen wurde eine Datenbank aufgesetzt. In dieser werden folgende Informationen pro bewerteten Linkpaar gespeichert:

- Username welcher vom Benutzer als „Login“ eingetragen wurde. (wenn keine Username gesetzt wurde wird die IP-Adresse abgespeichert)
- „Subject“, „Prädikat“ und „Objekt“ von beiden Links
- Entscheidung des Nutzers („accept“, „reject“, „ignore“)
- Timestamp

3 Grundsätzliche Struktur- und Entwurfsprinzipien des Gesamtsystems

Da es sich um eine browserbasierte Anwendung handelt bildet die `index.html` die Ausgangsposition. Diese ist in HTML geschrieben.

Um eine ansprechende Nutzerinteraktion und dynamische Inhalte gestalten zu können wird JavaScript als Programmiersprache eingesetzt. Für komfortable Nutzung und einfache Erweiterung der Document Object Model (kurz 'DOM') wird die JavaScript-Klassenbibliothek jQuery² genutzt.

JavaScript-Dateien werden im Ordner *scripts* abgelegt. Dabei wird eine flache Ablagestruktur verfolgt.

Um ein einheitliches Design durchsetzen zu können, wird eine `style.css` im Ordner *styles* für das Hauptdesign genutzt. Weitere Designs, z.B. für den Datatree oder das Dropdown-Menü, werden in eigenen `*.css` Dateien abgelegt.

4 Grundsätzliche Struktur- und Entwurfsprinzipien einzelner Pakete

In folgendem Abschnitt soll auf einzelne JavaScript-Pakete eingegangen werden. Eine ausführlichere und aktuelle Hilfe ist stets unter <http://pcai042.informatik.uni-leipzig.de/~swp12-11/jsdoc/> zu finden

²<http://jquery.com>



4.1 dataMapPoint.js

Diese Klasse erstellt die entsprechenden Punkte auf der Karte.

4.2 dataTree.js

Mithilfe dieser Klasse werden die von SPARQL erzeugten JSON-Objekte verwaltet und an die Wissensbasen-Bäume geleitet.

4.3 dropdown.js

Die Dropdown-Klasse ist für die Auswahl der Wissensbasen verantwortlich.

4.4 global.js

Dieses Paket instantiiert die wichtigsten Klassen, welche benötigt werden um die Grundfunktionalität herzustellen.

Darunter zählen unter anderem die Klassen GUI, Map und DataTree.

4.5 gui.js

Diese Klasse ist für die Anzeige der Tooltips und Message-Boxen zuständig und managt das Layout der GUI.

4.6 linking.js

Die Linking-Klasse bekommt zwei Klassen URLs, aus unterschiedlichen Wissensbasen, von `treeEvent.js` übergeben und bekommt Links als JSON-Objekt von einem SPARQL-Endpunkt. Dieses Objekt übergibt `linking.js` an `linkListjs`.

4.7 map.js

Die OpenStreetMap Karte wird geladen und die wichtigsten Parameter werden eingestellt. Dabei wird auf die Funktionalität der OpenLayers zurückgegriffen. Das Laden der Karte erfolgt z.b. über

```
this.map = new OpenLayers.Map();
```

Zu weiteren Parametern gehören Longitude und Latitude sowie die Möglichkeit des Freihandtools und der Zoom-Funktion. Die Koordinaten werden bei der Initiierung innerhalb der `global.js` übergeben.

4.8 mapPoint.js

Diese Klasse repräsentiert einen Punkt auf der Karte.



4.9 sparqlLib.js

Die SparqlLib-Klasse verwaltet die Anfragen an die SPARQL-Endpunkte über die AJAX-Funktionen des Browsers.

4.10 tree.js

Mithilfe dieser Klasse werden die Bäume generiert und an die GUI übergeben. Sie ist für die korrekte hierarchische Darstellung zuständig.

4.11 treeEvent.js

Die Tree-Event-Klasse managt Klapp- und Markierfunktionen der Bäume.

4.12 username.js

Username.js ist für die Benutzeranmeldung und -verwaltung verantwortlich.