



Lastenheft

Projektnummer: swp10-7

Projekttitle: jQuery Plugin - Erweiterte Autovervollständigung auf der Basis von SPARQL Endpunkten

Abgabe: 26.04.2010

Verantwortlicher: Clemens Hoffmann & Frederik Baumgardt



Inhaltsverzeichnis

Zielbestimmung	1
Produkteinsatz	1
Produktübersicht	2
Produktfunktionen	4
Muss-Funktionen	4
Suchanfragen	4
SPARQL-Anfragen	4
Ergebnismenge verarbeiten	4
Widget einrichten	5
Kann-Funktionen	6
SPARQL-Anfragen	6
Ergebnismenge verarbeiten	6
Produktdaten	7
Produktleistungen	7
Qualitätsanforderungen	8
Ergänzungen	8



Zielbestimmung

Die zu entwickelnde Anwendung soll den Nutzern während der Manipulation von RDF-basierten Ontologien einen ergonomischen und intuitiven Zugriff auf die zugrundeliegenden RDF-Graphen bieten.

Konkret wird das in OntoWiki vorhandene Eingabefeld mit Autovervollständigung für die Bearbeitung von Ressourcen durch ein Eingabefeld inklusive jQuery-Plugin ersetzt. Das Plugin ermöglicht eine erweiterte Definition der Suchanfrage im Rahmen der Autovervollständigung und übersichtlichere Präsentationsformen der Ergebnismenge.

Aus Sicht der Entwickler ist eine wichtige Anforderung an das Plugin ein modularer Aufbau, der schnelle und umfangreiche Anpassungen über Callback Methoden ermöglicht.

Produkteinsatz

Zunächst soll das Widget in OntoWiki eingesetzt werden. Dort unterstützt es die Benutzer - die mehr fachlich und weniger technisch versiert sind - bei der Erstellung und Bearbeitung von Ressourcen.

Durch die Implementierung als jQuery-Plugin und den modularen Aufbau sind weitere Einsatzfelder denkbar.



Produktübersicht

Das Widget besteht aus einer Editierfeld (Suchmaske) zur Eingabe der gesuchten Zeichenkette und einem auf dieses Feld angewandten jQuery-Plugin.

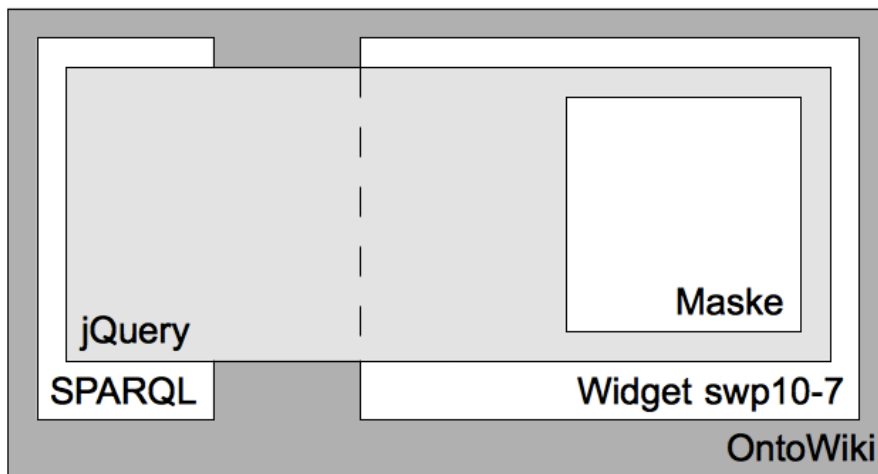


Abb. 1: Die Übersicht der Software-Architektur

Die jQuery-Methode erweitert das Feld um ein Drop-Down Menü zur Darstellung der Ergebnismenge einer zur SPARQL-Anfrage übersetzten Suchanfrage des Benutzers und ein Drop-Up Menü zur Bedienung von teilweise dynamisch bestimmten Filter- und Darstellungsoptionen. Beide Menüs lassen sich von Entwicklern mittels jQuery Callback-Methoden modifizieren, um die Ergebnismenge visuell aufzubereiten oder anwendungsbezogene Optionen anzubieten.

Die Verarbeitung einer Suchanfrage des Benutzers funktioniert, indem bei Eingabe von mehr als 3 Zeichen aus Suchanfrage und Facetten eine SPARQL-Anfrage erzeugt wird, diese an die ausgewählten SPARQL-Endpunkte gesendet wird, und der Rückgabewert nach Maßgabe der gewählten Darstellungsoptionen in einer DOM-Struktur formatiert wird.

Der erste Suchlauf findet ohne Filterung durch Facetten statt. Die optionalen Facetten werden aus der ungefilterten Ergebnismenge abgeleitet.

Hat der Benutzer die gesuchte Ressource gefunden, kann er sie mit Maus oder Tastatur selektieren und in das Editierfeld übernehmen. Durch Anklicken des Rückgabe-Buttons oder drücken der Return-Taste wird das Ergebnis an einen weiteren Prozess im OntoWiki übergeben.

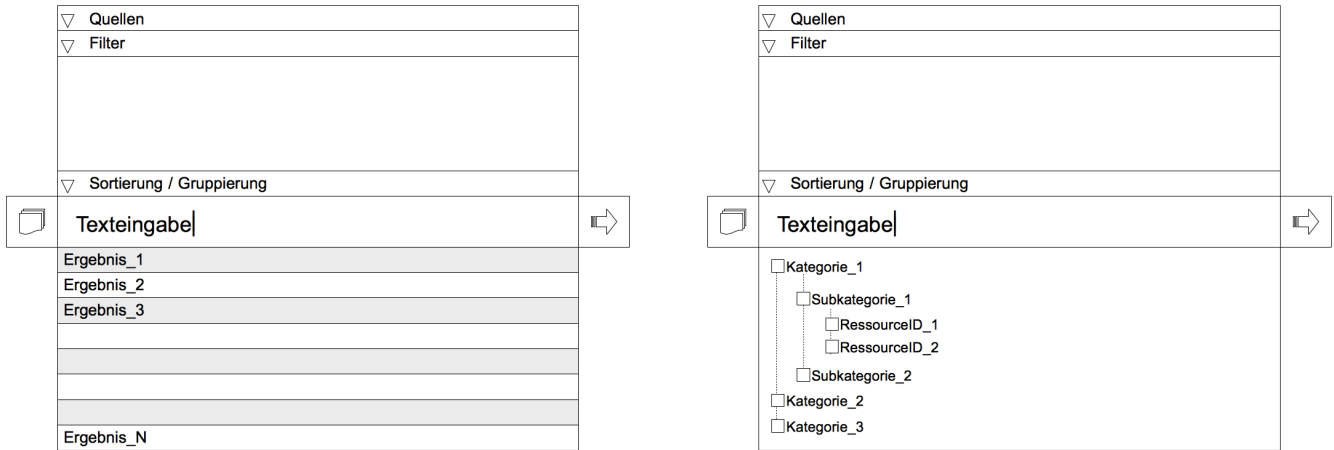


Abb. 2: Die Entwürfe der Benutzerschnittstelle mit a) Liste und b) Hierarchie

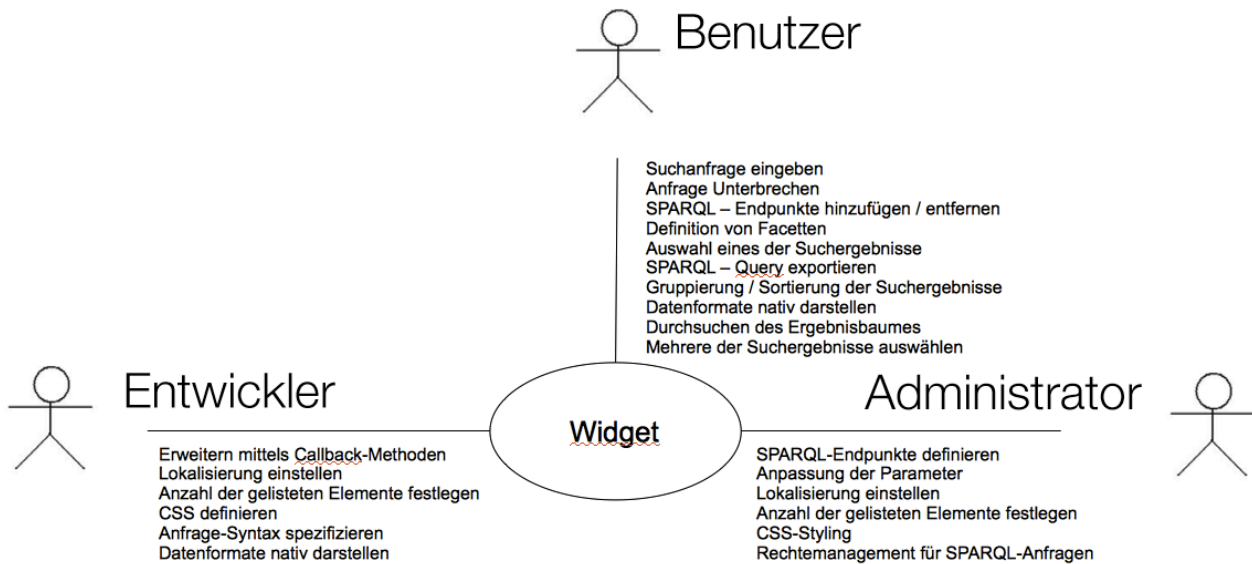


Abb. 3: Das Kontextdiagramm



Produktfunktionen

Muss-Funktionen

Suchanfrage

- /LF10/ Geschäftsprozess: Suchanfrage eingeben
Akteur: Benutzer
Beschreibung: Der Nutzer kann seine Suchanfrage in ein Textfeld eingeben
- /LF20/ Geschäftsprozess: Anfrage unterbrechen
Akteur: Benutzer
Beschreibung: Der Nutzer kann durch defokussieren der Suchleiste eine bereits gestartete Anfrage unterbrechen

SPARQL - Anfrage

- /LF30/ Geschäftsprozess: einen o. mehrere SPARQL Endpunkte definieren
Akteur: Administrator
Beschreibung: Dem Widget können mehrere SPARQL Endpunkte als Parameter übergeben werden, an die die Suchanfrage parallel gestellt wird und deren Rückgabewerte asynchron in der Ergebnisliste dargestellt werden
- /LF31/ Geschäftsprozess: SPARQL Endpunkt hinzufügen
Akteur: Benutzer
Beschreibung: Der Nutzer kann aus der Liste der o.g. zur Verfügung gestellten SPARQL-Endpunkte die für ihn inhaltlich relevanten zu seiner Anfrage hinzufügen
- /LF32/ Geschäftsprozess: SPARQL Endpunkt entfernen
Akteur: Benutzer
Beschreibung: Der Nutzer kann ausserdem bereits gewählte SPARQL Endpunkt wieder deselektieren

Ergebnismenge verarbeiten

- /LF40/ Geschäftsprozess: Definition von Facetten
Akteur: Benutzer



Beschreibung: Der Nutzer kann die Darstellung der Ergebnismenge durch eine GUI-basierte Definition von Facetten filtern und reduzieren

/LF50/ Geschäftsprozess: Auswahl eines der Suchergebnisse
 Akteur: Benutzer
 Beschreibung: Durch anklicken eines Elements der Ergebnismenge wird dieses in die Suchleiste eingetragen

/LF60/ Geschäftsprozess: Erweiterbarkeit mittels Callback Methoden
 Akteur: Entwickler
 Beschreibung: Die Filterung, Sortierung und Darstellung der Ergebnismenge lassen sich mittels Callback Methoden anpassen oder erweitern

Widget einrichten

/LF70/ Geschäftsprozess: Anpassung der Parameter
 Akteur: Administrator
 Beschreibung: Mit der Verwendung des Widgets erhält der Nutzer die Möglichkeit durch Überschreiben von vordefinierten DEFAULT Werten zum Beispiel Autofill zu aktivieren, das erste Element in der DDB zu markieren oder die Weite der DDB zu definieren

/LF80/ Geschäftsprozess: Lokalisierung einstellen
 Akteur: Entwickler, Administrator
 Beschreibung: Die Sprache lässt sich aus einer Liste zur Verfügung gestellter Lokalisierungen auswählen, oder mittels Parameter eine eigene Lokalisierung einsetzen

/LF90/ Geschäftsprozess: Anzahl der gelisteten Elemente aus der Ergebnismenge festlegen
 Akteur: Entwickler, Administrator
 Beschreibung: Es lässt sich eine Obergrenze für die Größe der Ergebnismenge festlegen (damit ist nicht die Höhe des Drop-Down Menüs gemeint)

/LF100/ Geschäftsprozess: CSS-Styling des Widgets



Akteur: Entwickler, Administrator
 Beschreibung: Die DOM-Elemente des Widgets sind als CSS-Klassen realisiert und können mit eigenen Style-Definitionen versehen werden

Kann-Funktionen

SPARQL - Anfrage

/LF110/ Geschäftsprozess: Rechtemanagement für SPARQL - Anfragen
 Akteur: Administrator
 Beschreibung: Der Administrator kann die Zugriffsrechte der Benutzer kontrollieren, indem er über einen Parameter festgelegte Bedingungen zu den SPARQL Anfragen hinzu fügt

/LF120/ Geschäftsprozess: Anfrage - Syntax spezifizieren
 Akteur: Entwickler
 Beschreibung: Mittels einer Callback Methode lässt sich eine Syntax für eine erweiterte Funktionalität der Suche einsetzen

/LF130/ Geschäftsprozess: SPARQL Query exportieren
 Akteur: Benutzer
 Beschreibung: Die SPARQL Query einer der Ergebnismenge zu grundlegenden SPARQL Anfrage lässt sich als Plaintext exportieren

Ergebnismenge verarbeiten

/LF140/ Geschäftsprozess: Gruppierung der Suchergebnisse
 Akteur: Benutzer
 Beschreibung: Die Ergebnismenge lässt sich anhand von Attributen gruppieren, bspw. nach Prädikaten oder Anfangsbuchstaben d. Präfixe

/LF141/ Geschäftsprozess: Sortierung der Suchergebnisse
 Akteur: Benutzer
 Beschreibung: Die Ergebnismenge lässt sich anhand von Attributen sortieren, bspw. nach Prädikaten oder Anfangsbuchstaben d. Literale



/LF150/	Geschäftsprozess:	Datenformate nativ darstellen
	Akteur:	Entwickler, Benutzer
	Beschreibung:	Entwickler können mittels Callback Methoden Ressourcen mit Multimedia-Inhalten nativ darstellen, Benutzer können diese Funktion zur Laufzeit ein- und ausstellen
/LF160/	Geschäftsprozess:	Durchsuchen des Ergebnisbaumes
	Akteur:	Benutzer
	Beschreibung:	Benutzer können eine Baumansicht der Ergebnismenge aufrufen und sich darin durch expandieren und kollabieren der Zweige bewegen
/LF170/	Geschäftsprozess:	mehrere Ergebnisse auswählen
	Akteur:	Benutzer
	Beschreibung:	Durch Anklicken eines oder mehrerer Elemente der Ergebnismenge werden diese markiert und in die Suchleiste eingetragen

Anmerkung: *Die Begriffe SPARQL-Anfrage und SPARQL-Query werden synonym verwendet.*

Produktdaten

Es werden keine Nutzdaten gespeichert.

Produktleistungen

/LL10/ Ergebnisse sollen ohne spürbare Verzögerung angezeigt werden

/LL20/ Ist keiner der SPARQL-Endpunkte erreichbar soll dem Benutzer innerhalb von 10 Sekunden Rückmeldung gegeben werden.



Qualitätsanforderungen

Produktqualität	Sehr Gut	Gut	Normal	Irrelevant
Funktionalität			X	
Zuverlässigkeit			X	
Benutzbarkeit	X			
Effizienz			X	
Änderbarkeit		X		
Übertragbarkeit		X		

Ergänzungen

/LE10/ Die Anwendung wird dual-lizenziert veröffentlicht, um sowohl den akademischen wie den kommerziellen Einsatzfeldern gerecht zu werden.

/LE20/ Die jQuery-Bibliothek zur Entwicklung des Widgets zu verwenden, sowie Erweiterbarkeit durch Callback-Methoden zu gewährleisten, sind zwingende Vorgaben.

/LE30/ Bei Verwendung mehrerer SPARQL-Endpunkte kann es zu Redundanzen in den Ergebnismengen der verschiedenen Endpunkte kommen. Diese müssen beim Aufbau der DOM-Struktur abgefangen werden.