



Pflichtenheft

Projektnummer: swp10-7

Projekttitlel: jQuery Plugin – Erweiterte Autovervollständigung auf der Basis von SPARQL Endpunkten

Abgabe: 17.5.2010

Verantwortlicher: Thomas Schöne & Marcus Nitzschke



Inhaltsverzeichnis

1. Zielbestimmung.....	Seite 3
1. Musskriterien.....	Seite 3
2. Kannkriterien.....	Seite 3
2. Produkteinsatz	Seite 3
1. Anwendungsbereiche	Seite 3
2. Zielgruppen	Seite 3
3. Produktübersicht.....	Seite 4
4. Produktfunktionen.....	Seite 5
4.1 Muss-Funktionen.....	Seite 5
4.2 Kann-Funktionen.....	Seite 7
5. Produktdaten.....	Seite 8
6. Produktleistungen.....	Seite 8
7. Qualitätsanforderungen	Seite 9
8. Benutzungsoberfläche.....	Seite 9
9. Nichtfunktionale Anforderungen.....	Seite 9
10. Technische Produktumgebung	Seite 9
1. Software	Seite 9
11. Spezielle Anforderungen an die Entwicklungsumgebung	Seite 9
12. Gliederung in Teilprodukte	Seite 9
13. Ergänzungen	Seite 9



1. Zielbestimmung

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

Konkret wird das in OntoWiki vorhandene Eingabefeld mit Autovervollständigung für die Auswahl 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.

1.1 Musskriterien

- Suchanfragen eingeben und abbrechen
- SPARQL Endpunkte definieren, hinzufügen oder entfernen
- Definition von Facetten zur Eingrenzung der Suchergebnisse
- Auswahl eines Suchergebnisses
- Erweiterbarkeit mittels Callback-Methoden
- Anpassung der Parameter des Widgets
- Anzahl gelisteter Elemente der Ergebnismenge festlegen
- CSS-Styling des Widgets

1.2 Kannkriterien

- Exportieren der SPARL Query
- Gruppierung der Suchergebnisse
- Sortieren der Suchergebnisse
- Datenformate nativ darstellen
- Einstellen der Sprache

2. Produkteinsatz

2.1. Anwendungsbereiche

Zunächst soll das Widget im OntoWiki eingesetzt werden. Dort soll es den Benutzern ermöglichen, mittels eines Eingabefeldes Ressourcen schneller zu finden. Weiterhin ermöglicht es eine erweiterte Definition der Suchanfrage im Rahmen der Autovervollständigung und übersichtlichere Präsentationsformen der Ergebnismenge.

2.2 Zielgruppen

Die Zielgruppe sind die mehr fachlich und weniger technisch versierten Benutzer des OntoWikis, denen aber durch das Plugin der volle Informationsgehalt eröffnet werden soll.



3. Produktübersicht

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

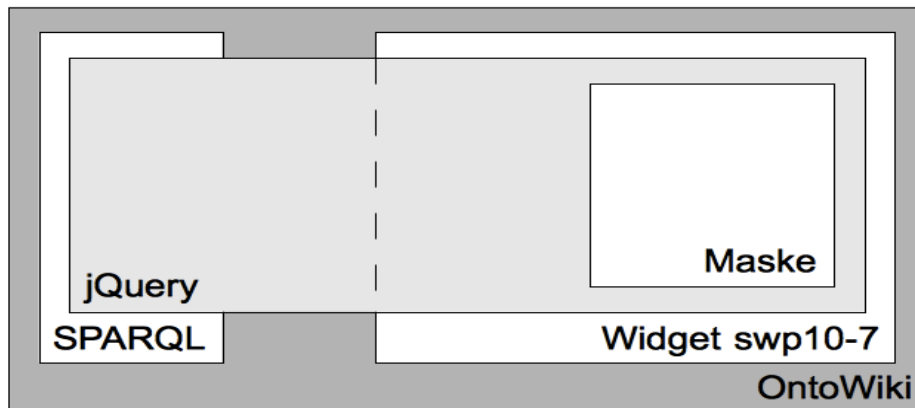


Abb. 1: Die Übersicht der Software-Architektur

Das jQuery-Plugin 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 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.

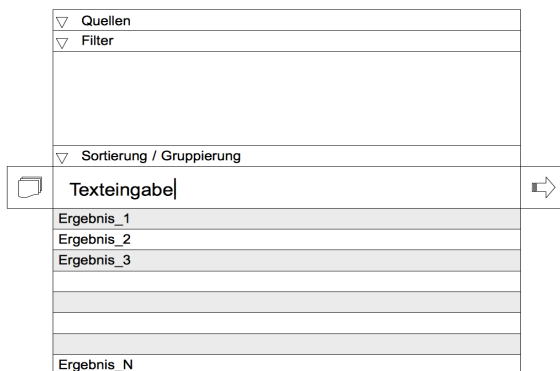




Abb. 2: Entwurf der Benutzerschnittstelle

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.

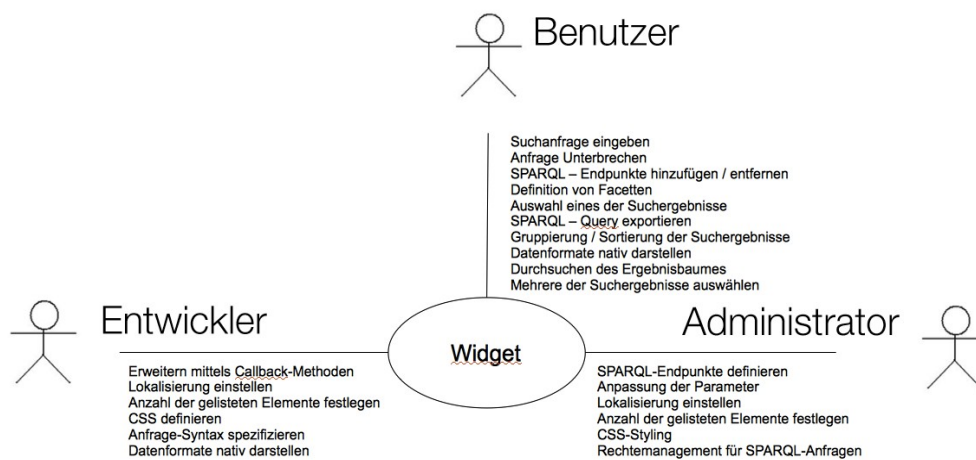


Abb. 3: Das Kontextdiagramm

4. Produktfunktionen

4.1 Muss-Funktionen

Suchanfrage

/F10/	Geschäftsprozess: Akteur: Beschreibung:	Suchanfrage eingeben Benutzer Der Nutzer kann seine Suchanfrage in ein Textfeld eingeben; nach 3 Zeichen wird der erste Query abgeschickt und das Plugin öffnet seine GUI
/F20/	Geschäftsprozess: Akteur: Beschreibung:	Anfrage unterbrechen Benutzer Der Nutzer kann durch Defokussieren der Suchleiste eine bereits gestartete

Verantwortlicher : Thomas Schöne & Marcus Nitzschke Projektleiter: Marcus Nitzschke



Anfrage unterbre
Plugin beendet

SPARQL – Anfrage

/F30/	Geschäftsprozess: Akteur: Beschreibung:	einen oder mehrere SPARQL Endpunkte definieren Administrator 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
/F31/	Geschäftsprozess: Akteur: Beschreibung:	SPARQL Endpunkt aktivieren Benutzer Der Nutzer kann aus der Liste der in F30 zur Verfügung gestellten SPARQL-Endpunkte die für ihn inhaltlich relevanten zu seiner Anfrage hinzufügen, dabei werden die Endpunkte vorher auf Erreichbarkeit geprüft und bei Bedarf ausgegraut
/F32/	Geschäftsprozess: Akteur: Beschreibung:	SPARQL Endpunkt deaktivieren Benutzer Der Nutzer kann ausserdem bereits gewählte SPARQL Endpunkt wieder deselektieren
Ergebnismenge verarbeiten		
/F40/	Geschäftsprozess: Akteur: Beschreibung:	Definition von Facetten Benutzer Der Nutzer kann die Darstellung der Ergebnismenge durch eine GUI-basierte Definition von Facetten filtern und reduzieren; dabei können schrittweise auf die Ergebnismenge angepasste Facetten gewählt werden
/F50/	Geschäftsprozess: Akteur: Beschreibung:	Auswahl eines der Suchergebnisse Benutzer Durch Anklicken eines Elements der Ergebnismenge wird dieses in Form eines human-readable Formates in die Suchleiste eingetragen



/F60/	Geschäftsprozess: Akteur: Beschreibung:	Erweiterbarkeit mittels Callback Methoden Entwickler Die Filterung, Sortierung und Darstellung der Ergebnismenge lassen sich mittels Callback Methoden anpassen oder erweitern
Widget einrichten		
/F70/	Geschäftsprozess: Akteur: Beschreibung:	Anpassung der Parameter Administrator Mit der Verwendung des Widgets erhält der Administrator die Möglichkeit durch Überschreiben von vordefinierten Default- Werten das Plugin genauer anzupassen
/F90/	Geschäftsprozess: Akteur: Beschreibung:	Anzahl der gelisteten Elemente aus der Ergebnismenge festlegen Entwickler, Administrator Es lässt sich eine Obergrenze für die Zahl der angezeigten Ergebnisse festlegen
/F100/	Geschäftsprozess: Akteur: Beschreibung:	CSS-Styling des Widgets Entwickler, Administrator Die DOM-Elemente des Widgets sind als CSS-Klassen realisiert und können mit eigenen Style-Definitionen versehen werden

4.2 Kann-Funktionen

/F80/	Geschäftsprozess: Akteur: Beschreibung:	Lokalisierung einstellen Entwickler, Administrator Die Sprache lässt sich aus einer Liste zur Verfügung gestellter Lokalisierungen auswählen, oder mittels Parameter eine eigene Lokalisierung einsetzen
/F130/	Geschäftsprozess: Akteur: Beschreibung:	SPARQL Query exportieren Benutzer Die SPARQL Query einer der Ergebnismenge zugrundeliegenden SPARQL Anfrage lässt sich als Plaintext



exportieren

Ergebnismenge verarbeiten

/F140/	Geschäftsprozess: Akteur: Beschreibung:	Gruppierung der Suchergebnisse Benutzer Die Ergebnismenge lässt sich anhand von Attributen gruppieren, bspw. nach Prädikaten oder Anfangsbuchstaben d. Präfixe
/F141/	Geschäftsprozess: Akteur: Beschreibung:	Sortierung der Suchergebnisse Benutzer Die Ergebnismenge lässt sich anhand von Attributen sortieren, bspw. nach Prädikaten oder Anfangsbuchstaben d. Literale
/F150/	Geschäftsprozess: Akteur: Beschreibung:	Datenformate nativ darstellen Entwickler, Benutzer Entwickler können mittels Callback Methoden Ressourcen mit Multimedia-Inhalten nativ darstellen, Benutzer können diese Funktion zur Laufzeit ein- und ausstellen

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

5. Produktdaten

Es werden keine Nutzdaten gespeichert.

6. Produktleistungen

/L10/ Ergebnisse sollen ohne spürbare Verzögerung angezeigt werden.

/L20/ Ist keiner der SPARQL-Endpunkte erreichbar soll dem Benutzer innerhalb von 10 Sekunden

Rückmeldung gegeben werden.



7. Qualitätsanforderungen

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

8. Benutzeroberfläche

/B10/ Die Benutzeroberfläche wird durch eine GUI dargestellt

/B20/ Der Stil der Benutzeroberfläche wird dem OntoWiki angepasst

/B30/ Die Benutzeroberfläche wird hauptsächlich per Maus bedient, soll aber auch durch Tastatur benutzt werden können.

9. Nichtfunktionale Anforderungen

10. Technische Produktumgebung

10.1 Software

Voraussetzung für das Plugin ist eine aktuelle Version des jQuery Frameworks.

11. Spezielle Anforderungen an die Entwicklungs-Umgebung

12. Gliederung in Teilprodukte

Eine Gliederung in Teilprodukte ist nicht vorgesehen.

13. Ergänzungen

/E10/ Die Anwendung wird dual-lizenziert veröffentlicht, um sowohl den akademischen wie den kommerziellen Einsatzfeldern gerecht zu werden.

/E20/ Die jQuery-Bibliothek zur Entwicklung des Widgets zu verwenden, sowie Erweiterbarkeit durch Callback-Methoden zu gewährleisten, sind zwingende Vorgaben.

/E30/ 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.