



Projektangebot

SWP14-SF

- Sebastian Kunadt -

- Marc Vester -

Inhalt

1. Zielbestimmung	2
2. Voraussetzungen	2
3. Designübersicht und Funktionalität	2
3.1 Design	2
3.2 Funktionalität	3
4. Arbeitspakete und Meilensteine	3
4.1 Arbeitspakete	3
4.1.1 Vorprojekt	3
4.1.2 Dynamische Formulargenerierung	3
4.1.3 Eingabe-/Ausgabemasken	4
4.1.4 Validierung und Speicherung	4
4.1.5 Dokumentation	4
4.1.6 Übergabe	4
4.2 Meilensteine	4
4.2.1 erster Meilenstein	4
4.2.2 zweiter Meilenstein	4
5. Qualitätssicherung	4
6. Glossar	5

1. Zielbestimmung

Das Ziel bei der Erstellung unserer Webanwendung zur Erfassung und Bearbeitung von Daten liegt darin, dynamisch ein auf vorliegende Daten angepasstes Formular zu generieren, welches Daten im RDF-Format speichern bzw. bearbeiten kann. Eine solche Webanwendung existiert bislang so nicht. Da sie jedoch eine essentielle Rolle im Semantic Web spielt, ist ihre Erstellung darum umso wichtiger.

Um die Möglichkeiten von Semantic Forms einem weit gefassten Benutzerkreis vom Laien bis zum erfahrenen Anwender zugänglich zu machen, legen wir dabei den Schwerpunkt neben einer hohen Anpassbarkeit insbesondere auf die Benutzerfreundlichkeit der Anwendung.

2. Voraussetzungen

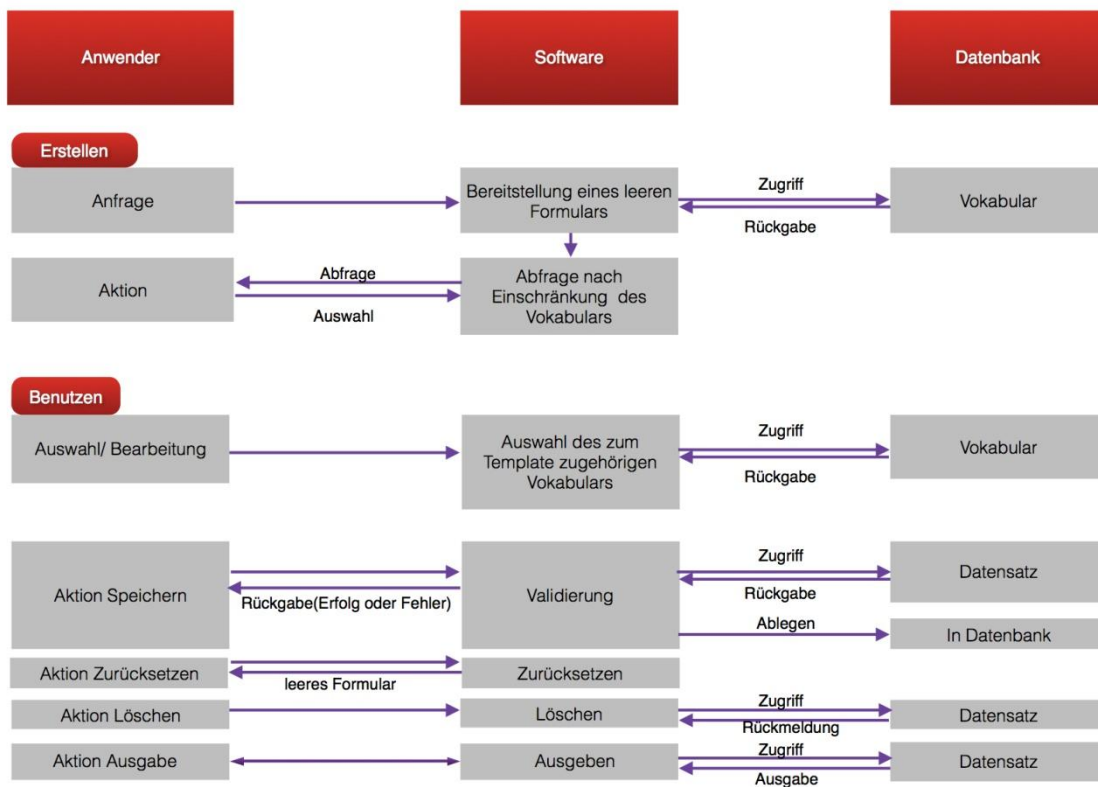
Um dem Kriterium der einfachen Bedienung gerecht zu werden, gilt es, die Anwendungsoberfläche überschaubar, intuitiv und effizient nutzbar zu halten. Funktionen müssen selbsterklärend benannt sein und ihre Anordnung sollte im Gesamtbild den Regeln einer sinnvollen Gestaltung folgen.

Voraussetzung für die dynamische Generierung von Formularen ist die Datenbankanbindung und somit der Einsatz der standardisierten Abfragesprache SPARQL (siehe Recherchebericht).

3. Designübersicht und Funktionalität

3.1 Design

Das Hauptaugenmerk in diesem Projekt liegt auf der Erstellung von Formularen durch Nutzen vorhandener RDF-Strukturen. Die Umsetzung wird wie folgt gestaltet:



Zuerst wird ein leeres Formular bereitgestellt. Dieses kann durch den Admin bei Bedarf eingeschränkt werden. Hierfür wird das bestehende Vokabular abgefragt und zur Einschränkung angeboten. Nach dem fertigstellen entsteht das fertige Template, aus dem dann das Formular für den Benutzer generiert wird.

Nun kann der Benutzer auswählen, bearbeiten und speichern, sowie die eingegebenen Daten zurücksetzen oder aus der DB Daten löschen.

Beim Speichern wird eine Validierung erforderlich, wofür ein Zugriff auf die DB erfolgt und bei Erfolg die Daten abgelegt werden.

3.2 Funktionalität

Die Funktionalität der Software gliedert sich in zwei Aspekte: Die Erstellung eines Formularmusters (Template) und das Ausfüllen, Anzeigen, Validieren und Speichern eines Formulars anhand eines Musters.

Für die Erstellung eines Templates kann der Benutzer bestehende Vokabulare (aus externen Quellen und bereits angelegten Templates) laden, und diese bei Bedarf einschränken. Bereits angelegte Templates können per Dropdown-Menü ausgewählt werden, für externe Vokabulare kann eine URI angegeben werden. Für die Einschränkung können Klassen und Properties abgewählt werden. Des Weiteren ist es möglich für die einzelnen Felder des Formularmusters festzulegen ob die Daten Muss- oder Kann- Daten sind, und bei Bedarf

Validierungskriterien festgelegt werden. Dieses Muster kann daraufhin in die Datenbank als ein Vokabular in einem RDF-Format gespeichert werden.

Für das Anlegen eines Datensatzes als Instanz eines Formulares kann ein Formularmuster aus der Datenbank geladen werden. In Textfelder kann der Benutzer nun Literale oder URIs von Ressourcen eintragen. Es ist möglich das Formular zurückzusetzen oder zu speichern. Im Falle des Speicherns wird das Formular validiert, und falls Felder den Validierungskriterien nicht genügen, werden diese markiert und dem Benutzer zur Bearbeitung zurückgemeldet. Ist die Validierung erfolgreich wird das Formular in einem RDF-Format gespeichert.

Des weiteren ist es möglich ein bestehendes Formular anzuzeigen, und bei Bedarf zu bearbeiten oder löschen. Im Falle der Bearbeitung findet vor dem Speichern eine Validierung statt. Bei Nicht-Erfolg wird dem Benutzer signalisiert welche Felder aus welchem Grund zurückgewiesen wurden, und korrigiert werden müssen.

4. Arbeitspakete und Meilensteine

4.1 Arbeitspakete

4.1.1 Vorprojekt 10%

Im Vorprojekt sollen grundlegende Funktionalitäten implementiert werden. Ziel ist dabei nicht eine detaillierte Umsetzung aller Anforderungen, sondern ein weitgehend fertiges Frontend, welches die Kernfunktionalitäten des fertigen Projekts demonstriert. Der Auftraggeber soll sich ein Bild von der Endanwendung machen können. Details zum Vorprojekt und dessen Umfang sind unter dem Hauptpunkt "Vorprojekt" aufgeführt. Darueberhinaus dient das Resultat des Vorprojekts allen Projektteilnehmern als eine Art Referenz.

4.1.2 Dynamische Formulargenerierung 40%

Einer der Hauptpunkte ist die dynamische Generierung der Formulare zur Datenerfassung und deren Bearbeitung. Die Generierung erfolgt durch eine vorliegende Strukturbeschreibung. Eine solche wird mittels Extraktion des Vokabulars über einen Formulareditor erstellt und in RDF abgespeichert, wobei auch mehrere Beschreibungen zusammengeführt werden können. Bei der Templategenerierung muss es möglich sein, das Vokabular einzuschränken, da je nach Anwendungsfall sehr viele Ressourcen vorhanden sein können, die im Einzelfall beim Hinzufügen/Bearbeiten eines Datensatzes nicht alle benötigt werden.

4.1.3 Eingabe-/Ausgabemasken 20%

Zur Eingabe und Bearbeitung der Daten sind spezielle Masken nötig, die dem Benutzer (vorausgefüllte) Eingabemöglichkeiten für alle Ressourcen in Abhängigkeit seiner Einschränkungen bieten. Desweiteren muss es möglich sein, die Daten anzuzeigen. Dieses Arbeitspaket ist eng verbunden mit der dynamischen Formulargenerierung.

4.1.4 Validierung und Speicherung 20%

Nach erfolgreicher Eingabe neuer/bearbeiteter Daten durch den Benutzer müssen die Daten entsprechend ihrer Strukturbeschreibung validiert und als RDF-Daten gespeichert werden. Dieser Vorgang ist besonders kritisch, da Fehler zu erheblichen Inkonsistenzen führen können, die in großen Datensammlungen nicht mehr rückgängig zu machen sind. Optional ist auch die Unterstützung weiterer Formate (etwa XML) möglich.

4.1.5 Dokumentation 10%

Zum Umgang mit der Anwendung wird eine kurze Dokumentation erstellt.

4.1.6 Übergabe 10%

Das fertige Projekt wird an den Auftraggeber übergeben und evtl. eine kurze Schulung zur Benutzung durchgeführt, sofern die Benutzung nicht selbsterklärend ist.

4.2 Meilensteine

4.2.1 erster Meilenstein

Alle grundlegenden Voraussetzungen für das Eintreten des Teams in die nächste Phase (SoftwareStudie und Entwurfsbeschreibung, Modellierung usw.) sind abgeschlossen. Die Rollenverteilung ist klar, das Arbeitsumfeld einsatzbereit, Risikoanalyse, Recherchebericht und Qualitätssicherungskonzept sind fertig. Alle Teammitglieder haben gleiche Vorstellungen vom Projekt, dessen Anforderungen und der geplanten Funktionalität. Ab diesem Zeitpunkt befasst sich das Team bereits mit der konkreten Umsetzung des Projekts.

4.2.2 zweiter Meilenstein

Das Team ist bereits weit fortgeschritten in der Umsetzungsphase und nähert sich dem Abschluss des Projekts. Ab hier treten gegenüber der Endphase der Programmierung die Tests zur Qualitätssicherung und die Dokumentation in den Vordergrund um die endgültige

Abnahme bestmöglich vorzubereiten. Es soll bereits eine funktionsfähige Testversion zur Verfügung stehen, die alle Kernfunktionalitäten bereitstellt.

5. Qualitätssicherung

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		x		
Richtigkeit		x		
Interoperabilität			x	
Ordnungsmäßigkeit		x		
Sicherheit			x	
Zuverlässigkeit				
Reife			x	
Fehlertoleranz		x		
Wiederherstellbarkeit		x		
Benutzbarkeit				
Verständlichkeit	x			
Erlernbarkeit	x			
Bedienbarkeit	x			
Effizienz				
Zeitverhalten		x		
Verbrauchsverhalten		x		
Änderbarkeit				
Analysierbarkeit		x		
Modifizierbarkeit		x		
Stabilität	x			
Prüfbarkeit		x		
Übertragbarkeit				
Anpassbarkeit	x			
Installierbarkeit			x	
Konformität		x		
Austauschbarkeit		x		

6. Glossar

Dropdown-Menü Auswahlfenster, das sich unterhalb des Mauszeigers oder einer Menüleiste öffnet

Formular standardisiertes Mittel zur Erfassung, Ansicht und Aufbereitung von Daten

Maske	wie ein Formular aufgebaute Bildschirmoberfläche eines Computers, die zum Eintragen von strukturierten Daten auf den Bildschirm abgerufen werden kann
RDF	Ressource Description Framework, System zur Beschreibung von Ressourcen
Ressource	Dateien oder Dienste, die in einem Netzwerk bereitgestellt werden
Validierung	Wert beziehungsweise Prüfung der Eignung einer Software bezogen auf ihren Einsatzzweck
Webanwendung	Anwendungsprogramm, das über Internet ausgeführt wird
XML	extensible Mark-up Language, Sprache zur Definition von Auszeichnungssprachen