



- Entwurfsphase: Entwurfsbeschreibung Gesamtsystem -

Version: 1.0

Projektbezeichnung	MSP-13 - Integration eines Semantischen Tagging Systems in Microsoft Sharepoint	
Projektleiter	Martin John	
Verantwortlich	Martin John	
Erstellt am	06.04.2013	
Zuletzt geändert	21.05.2013 09:46	
Bearbeitungszustand	<input type="checkbox"/>	in Bearbeitung
	<input checked="" type="checkbox"/>	vorgelegt
	<input type="checkbox"/>	fertig gestellt
Dokumentablage	git- Dokumente\Entwurfsbeschreibung\	
V-Modell-XT Version	1.4	

INHALTSVERZEICHNIS

1. Einleitung	4
2. Produktübersicht	4
3. Grundsätzliche Struktur- und Entwurfsprinzipien	5
4. Struktur- und Entwurfsprinzipien einzelner Pakete	6
<i>Sharepoint Paket (Skos-Tag-Feature)</i>	
<i>Button-Integration-Feature</i>	
<i>Webfeature</i>	
<i>TabNavApp</i>	
<i>Search/Add Tags:</i>	
<i>Edit-Tags</i>	
<i>Explorer für die aktuelle Taxonomie</i>	
<i>Virtuoso-Skos-Silverlight</i>	
<i>DocGraphUpdate</i>	
5. Modellierung	8
<i>Kontext</i>	
<i>ER-Modell</i>	
<i>Datenmodell</i>	

Änderungsverzeichnis

Änderungen			Geänderte Kapitel	Beschreibung der Änderungen	Autor	Zustand
Nr.	Datum	Version				
1	06.04.	0.1	alle	initiale Erstellung	JM	fertig
2	08.04.	0.2	6	Testkonzept hinzugefügt	JM	fertig
3	22.04.	0.3	2,3,4,5	umfangreiche Ergänzungen	JM	fertig
4	20.05	0.9	2,3,4,5	Anpassung an Ergebnis der Implementierung	JM	fertig
5	22.05	1.0	3,4,5	Aktualisierung der Modelle	JM	in Bearbeitung

Prüfverzeichnis

Die folgende Tabelle zeigt einen Überblick über alle Prüfungen – sowohl Eigenprüfungen wie auch Prüfungen durch eigenständige Qualitätssicherung – des vorliegenden Dokumentes.

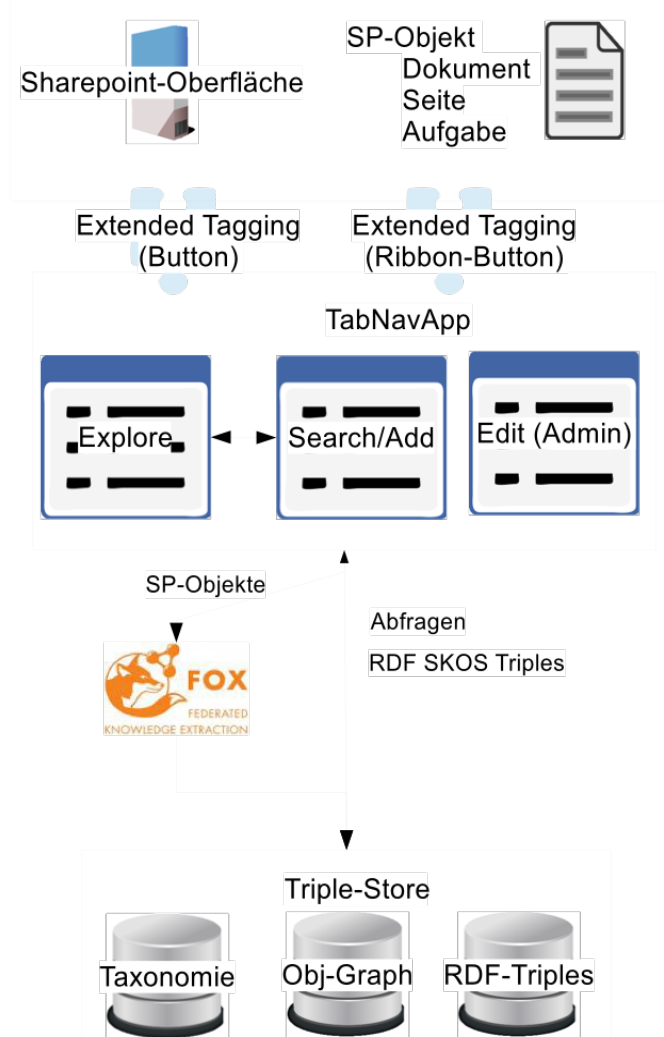
Datum	Geprüfte Version	Anmerkungen	Prüfer	Neuer Produktzustand
08.04.2013	0.1	Änderungen aufgezeichnet und Kommentare eingefügt.	RA	In Bearbeitung
21.05.2013	1.0	Abnahme	RA	Vorgelegt

1. Einleitung

Die folgende Entwurfsbeschreibung gibt einen Überblick über die erfolgte Umsetzung des Projekts. Dadurch sollen sich Programmierer, die das Projekt nicht kennen schnell in die Grundlagen des Projekts einarbeiten können. Nähere Informationen könne dem Administrator-Handbuch entnommen werden.

2. Produktübersicht

Die grundsätzliche Funktion wird durch folgende Übersicht dargestellt:



Für die Implementierung wurde das Projekt in drei Hauptkomponenten gegliedert:

Virtuoso-SKOS-Silverlight zur Speicherung und Verwaltung der RDF-Triples .

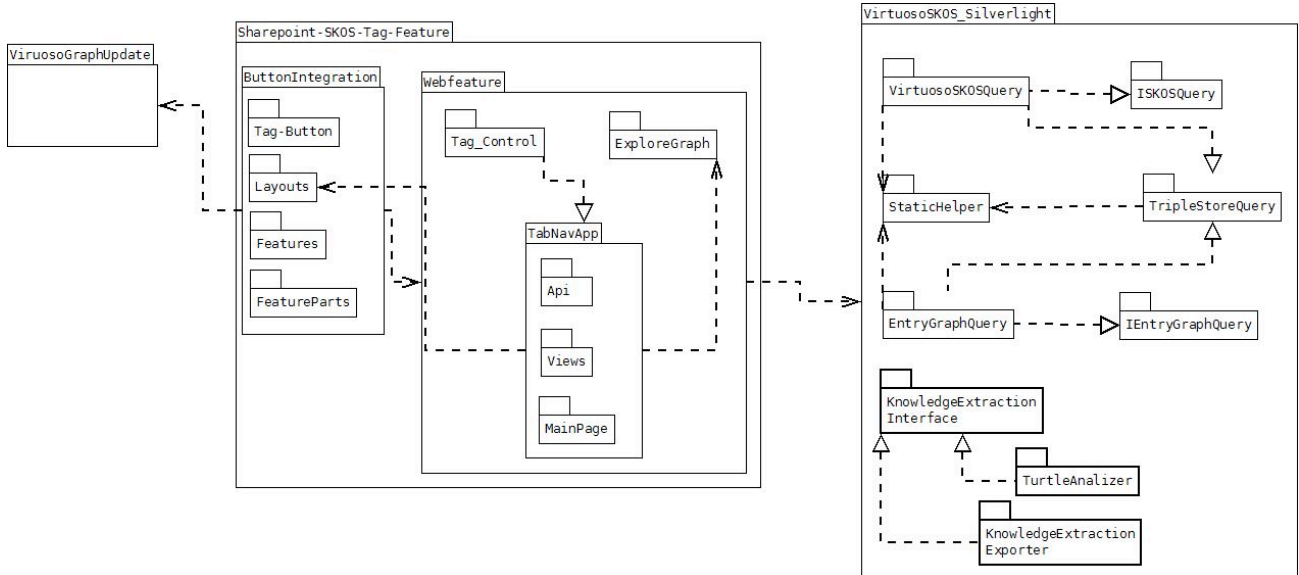
Farmfeature zur Deaktivierung des Standard-Taggings und Starten des erweiterten Taggings.

Webfeature als Interface zur Darstellung und Bearbeitung der Tags.

3. Grundsätzliche Struktur- und Entwurfsprinzipien

Für die Darstellung dynamischer Inhalte wurde aufgrund der gegebenen Kompatibilität auf Microsoft-Silverlight zurückgegriffen.

Die Hauptkomponenten sind in weitere Pakete unterteilt. Die folgende Darstellung zeigt eine Paketübersicht.



4. Struktur- und Entwurfsprinzipien einzelner Pakete

4.1. Sharepoint Paket (Skos-Tag-Feature)

Die Skos-Tag Solution ist in verschiedene Pakete und Module gegliedert, die sich grundsätzlich auf zwei Sharepointfeatures aufteilen und zusammen die View- und ViewModel-Schicht einer MVVM-Architektur darstellen.

4.1.1. Button-Integration-Feature

Dieses Feature hat den Scope 'Farm', d.h. Nach dessen Aktivierung ist es auf der gesamten Serverfarm vorhanden. Es umfasst den Button in der oberen rechten Ecke jeder Sharepointseite und den Tag-Button in den Ribbons. Die standardmäßig vorhandenen Buttons für die Sharepoint Social Features ('I Like', 'Tags & Comments') sollen dadurch ersetzt werden. Daher muss dieses Feature auch zuvor deaktiviert werden.

Die Buttons rufen den Inhalt des zweiten Features in einem Popupfenster auf. Markierte Objekte aus Sharepoint, wie Seiten, Aufgaben oder Dokumente werden dabei mit an das Webfeature übergeben.

4.1.2. Webfeature

Das Webfeature wird automatisch über 'Event Receiver' aktiviert.

Dieses Feature basiert auf einer einzelnen aspx-Seite deren einziger Inhalt eine Silverlightanwendung ist, welche alle anderen Bestandteile dieses Features beheimatet.

Silverlight (Version 5.1) kommt an dieser Stelle aus o.g. Gründen zum Einsatz.

4.1.2.1. TabNavApp

Diese Anwendung ist zentraler Bestandteil des Web basierten Features und damit auch Kern der gesamten Solution. In ihr vereinigen sich alle Benutzeroberflächeninhalte, sowie die gesamte Logik der Triple-Store Abfragen und die Verarbeitung der von dort erhaltenen Antworten.

Verschiedene Ansichten wie: Dokumentzusammenfassung mit Tag-Vorschlägen, Tag Suchfunktion mit Graphvisualisierung für den Frontendeinsatz, sowie zusätzliche Backendfunktionen für Administratoren soll über ein einfaches Tab-Control realisiert werden.

Der Haupttab ist der Search-Add-Tab, der beim öffnen des Tagging-Features angezeigt wird. Jedes Sharepoint Projekt wird mit den beiden Dateien 'App.xaml' and 'MainPage.xaml' initialisiert.

Die 'App.xaml' dient zur Ressourcendefinition, während die 'MainPage.xaml' die Definition der Oberfläche der automatisch generierten Startseite der Anwendung beinhaltet.

Wird die Anwendung mit Tab-Control initialisiert kommen die Ordner Assets (hier können verschieden Anzeige Templates hinterlegt werden) und Views (beinhaltet alle Tab Seiten) hinzu.

Allgemein: Hinter jeder .xaml Datei befindet sich eine so genannten 'Code-behind' Datei, z.B. In C# oder JavaScript, welche die gesamte Logik für die in der .xaml definierten Oberfläche zur Verfügung stellt und das Bindeglied zur Datenzugriffsschicht darstellt (Model). Silverlight setzt damit auf die von WPF oder HTML5 bekannte Architektur: Model View ViewModel (MVVM) und beinhaltet View und ViewModel.

4.1.2.2. Search/Add Tags:

Der Tab Search/Add Tags ist wie folgt gegliedert:

a) Tag Suchfunktion mit Ergebnisliste

Setzt die searchTags Methode der VirtuosoSkosQuery.cs um und Zeigt die top-k Suchergebnisse an.

Mit einen Klick auf 'View' wird ein Graphvisualisierung initialisiert.

b) Dokumentensuchfunktion mit Ergebnisliste

c) Schalter für das Hinzufügen/Löschen von Tags

d) Schalter für das Auswählen von Dokumenten und Tags

4.1.2.3. Edit-Tags

Ein Tab zur Verwaltung der Taxonomie für den Taxonomieadministrator.

4.1.2.4. Explorer für die aktuelle Taxonomie

Die Graphvisualisierung ist interaktiv. Mit einem Klick auf ein beliebiges Concept wird dieses zum zentralen Konzept und dessen Umgebungskonzepte angezeigt. Mit Strg+Click wurde ein Instant-Tagging realisiert.

Diese Visualisierung beruht auf dem bagotricks-Projekt¹. Dieses wurde aber an die Bedürfnisse dieses Projekts angepasst und entsprechend verändert.

Die Graph Mechanik findet sich in der SilverlightCompLib.dll, einige Funktionen für das Frontend wurden in die Klasse SearchGrapg.xaml.cs übernommen.

4.2. Virtuoso-Skos-Silverlight

Dieses Paket umfasst einen Teil der zweigeteilten Datenzugriffsschicht dieses Projektes und stellt den Zugriff auf einen Virtuoso-Triple-Store sicher. Die zweite Datenzugriffsschicht wird von Sharepoint über das Data-Object-Model, das auf einer SQL-Datenbank beruht, zu Verfügung gestellt.

In diesem Paket greifen wir auf die umfangreiche API Dot-Net-RDF² zurück, welche alle Funktionen die zur Kommunikation mit einem RDF-Triple-Store von Nöten sind bietet. Ein Virtuoso-Server mit aktiven SPARQL-Endpunkt bildet die Datenschicht für unser Projekt³ und ist mit dem Skos-Graphen des UNESO-Theasaurus⁴ versehen, welches die Grundlage für alle Überlegungen dieses Paketes bildete. Der von Virtuoso eingesetzte SPARQL-Dialekt setzt bereits auf einige Konzepte der Sparql-Version 1.1⁵ aber hat noch nicht alle beschriebenen Konzepte von W3C für SPARQL 1.1 implementiert⁶. Im Folgenden beschreiben 'Virtuoso' und 'Triple-Store' das selbe.

Die Bibliotheken haben den folgenden groben Aufbau:

ISkosQuery: ist das definierende Interface welches den aktuellen Funktionsumfang festlegt.

VirtuosoSkosQuery: dessen Implementierung für einen Virtuoso-Sparql-Endpunkt auf Grundlage von Dot-Net-RDF und einem Skos-Graphen der broader, sowie narrower als Prädikate bereitstellt (am Beispiel UNESCO-Theasaurus). Abweichungen von diesen Eckpunkten können durch eigene Implementierungen von IskosQuery begegnet werden.

IEntryGraphQuery: Entsprechend der IskosQuery ein Interface für den EntryGraph. Entries sind die Objekte, die auf einer SP-Seite getaggt werden können.

EntryGraphQuery: Der Entrygraph umfasst eine eigene Datenbank, in der getaggte Objekte registriert werden und mit den Tags in Verbindung gebracht werden. Somit werden Funktionen ermöglicht, die zu Tags gehörige Objekte und auch zu Objekten gehörige Tags

1 <https://github.com/lvaleriu/bagotricks>

2 <https://bitbucket.org/dotnetrdf/dotnetrdf/wiki/User%20Guide>

3 <http://www.openlinksw.com/wiki/main/>

4 <http://skos.um.es/unescothes/>

5 <http://www.w3.org/TR/rdf-sparql-query/>

6

<http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtTipsAndTricksSPARQL11FeaturesExamplesCollection>

nutzen. Weiterhin beinhaltet der Entrygraph Funktionen, um Objekte an ein Knowledge-Extraction-Framework zu senden und um Informationen, die Knowledge-Extraction-Frameworks generieren zu verarbeiten.

StaticHelper: Bietet statische Methoden, welche beliebig ergänzt werden können und zur Umwandlung von SPARQL-Resultsets in andere Metadatenformate dienen.

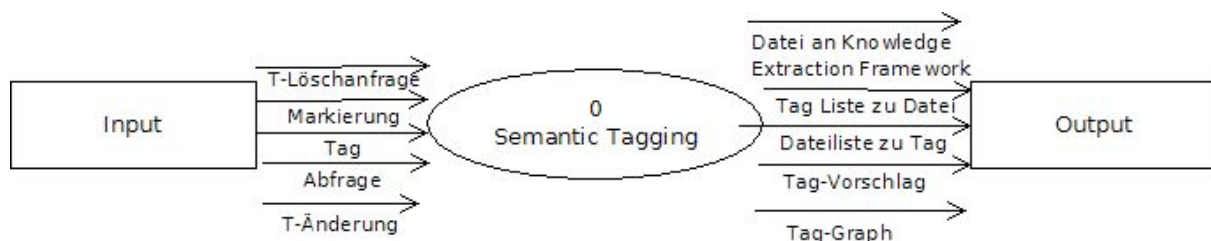
4.3. DocGraphUpdate

Das Paket DocGraphUpdate übernimmt die Verwaltung des Doc-Graphen im Virtuoso, wenn auf der Sharepoint-Oberfläche die Silverlight-App nicht aktiv ist. Wird ein Dokument von einer Sharepoint-Seite gelöscht, so werden auch die zugehörigen Einträge im Doc-Graphen gelöscht. Wird ein Dokument hinzugefügt, wird entsprechend ein Eintrag im Doc-Graphen angelegt.

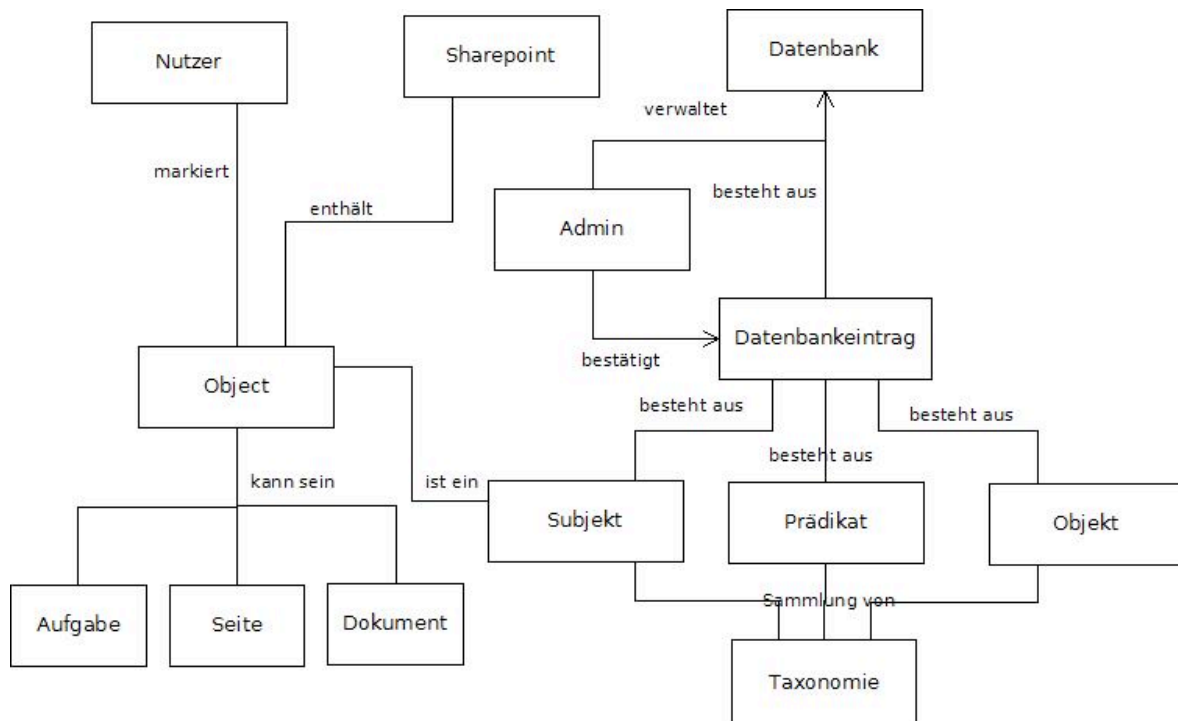
Die Klasse wird aus dem Feature-Paket unter Nutzung der Sharepoint-Event-Receiver aufgerufen.

5. Modellierung

5.1. Kontext



5.2. ER-Modell



5.3. Datenmodell

Das Datenmodell befindet sich in der Anlage 1.