

SectionTagging – Ein Tagging-System für das Austria-Forum

Masterarbeit
an der
Technischen Universität Graz

vorgelegt von
Wolfgang Pürstner, Bakk.rer.soc.oec.

Betreuer:
Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic

Institut für Wissensmanagement (IWM)



August 2010

© Copyright 2010 Wolfgang Pürstner

Kurzfassung

Tagging-Systeme erlauben es Benutzern, unterschiedliche Inhalte mit Schlagworten (engl. Tags) zu versehen um diese zu beschreiben. Die Tags können dann unter anderem auch zur Suche und zur Navigation benutzt werden. Im Allgemeinen erfreuen sich solche Systeme großer Beliebtheit und es existieren zahlreiche verschiedene Variationen. Für Wiki-Systeme, speziell dem JSPWiki, gibt es jedoch keine zufrieden stellenden Lösungen, die das Taggen einzelner Abschnitte erlauben. Weiters können die mit Tags versehenen Abschnitte nicht ohne Umwege für die Erstellung neuer Wiki-Seiten verwendet werden. In dieser Masterarbeit wird zunächst ein Überblick über die verschiedenen Tagging-Systeme und Wiki-Systeme gegeben. Im Anschluss wird ein Prototyp vorgestellt, der eigens für das Austria-Forum, welches auf JSPWiki basiert, entwickelt wurde. Dieser integriert ein benutzerfreundliches Tagging-System, welches die Tagvergabe auf einzelne Abschnitte erlaubt. Darüber hinaus bietet dieses System Autoren die Möglichkeit, zuvor getaggte Abschnitte für die Erstellung neuer Wiki-Seiten zu nutzen.

Master's Thesis

**Section Tagging –
A tagging system for
the Austria-Forum**

Wolfgang Pürstner, Bakk.rer.soc.oec.

Thesis supervisor:
Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic

Knowledge Management Institute (KMI)
Graz University of Technology



August 2010

© Copyright 2010 Wolfgang Pürstner

Abstract

Tagging systems enable users to label various content with descriptive tags. These tags can subsequently be used for many tasks, such as searching for corresponding content or navigation. Numerous variations of these systems exist that generally enjoy great popularity. However, there are still no satisfying solutions for wiki systems, in particular for JSPWiki, which allow single sections to be tagged. Moreover the tagged sections cannot be used to create new wiki pages. This thesis will initially provide an overview on various tagging and wiki systems. Subsequently a prototype, developed to be used with the JSPWiki-based Austria-Forum, is introduced. The prototype integrates a user-friendly tagging system and also allows the tagging of single sections. In addition this system lets authors use previously tagged sections for the creation of new wiki pages.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am
.....
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date

.....
(signature)

I. Danksagung

Als erstes möchte ich mich an dieser Stelle bei meinem Betreuer Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic bedanken, der mir diese Masterarbeit ermöglicht hat und mit dessen Unterstützung ich jederzeit rechnen konnte.

Weiters gilt ein großer Dank an Dipl.-Ing. Christoph Trattner für seine Hilfe und Anregungen besonders während des praktischen Teils der Arbeit.

Auch ein Dankeschön möchte ich an meine Eltern richten, die mir das Studium ermöglicht und mich in jeder Hinsicht unterstützt haben.

Zu guter Letzt danke ich meiner lieben Freundin Marie. Sie hat mit ihren motivierenden Worten wesentlich zur zügigen Fertigstellung dieser Arbeit beigetragen.

II. Inhaltsverzeichnis

I.	DANKSAGUNG	6
II.	INHALTSVERZEICHNIS	7
1	EINLEITUNG	10
2	TAGGING-SYSTEME	12
2.1	ALLGEMEINES.....	12
2.1.1	<i>Das Tag</i>	13
2.1.2	<i>Folksonomie</i>	13
2.1.3	<i>Tagklassen</i>	14
2.1.4	<i>Einflussfaktoren beim Tagging</i>	16
2.1.5	<i>Rechte bei der Tagvergabe</i>	19
2.1.6	<i>Arten der Tagvergabe</i>	19
2.1.7	<i>Navigation</i>	21
2.1.8	<i>Zusammenfassung</i>	25
2.2	BESCHREIBUNG VORHANDENER SYSTEME.....	26
2.2.1	<i>Scuttle</i>	26
2.2.2	<i>Flickr</i>	27
2.2.3	<i>SparTag.us</i>	30
2.2.4	<i>ESP Game</i>	34
2.2.5	<i>Automatische Tagextraktion</i>	36
2.2.6	<i>Zusammenfassung</i>	39
3	WIKI-SYSTEME	40
3.1	WIKI-SYSTEME	40
3.1.1	<i>Historische Entwicklung</i>	41
3.1.2	<i>Einsatzgebiete</i>	42
3.1.3	<i>Kernfunktionen</i>	42
3.1.4	<i>Benutzergruppen</i>	44
3.1.5	<i>Auszeichnungssprache</i>	45
3.1.6	<i>Probleme</i>	47
3.1.7	<i>Wiki-Klone</i>	49
3.2	TECHNIK.....	51
3.2.1	<i>Webtechnologien</i>	52
3.2.2	<i>Webserver</i>	54
3.2.3	<i>Datenspeicher</i>	55

3.3	JSPWIKI	58
3.3.1	<i>Java Servlets und JSP</i>	58
3.3.2	<i>Funktionsumfang</i>	60
3.3.3	<i>Auszeichnungssprache</i>	61
3.3.4	<i>Datenspeicherung</i>	62
3.3.5	<i>Installation</i>	63
3.4	DAS AUSTRIA-FORUM	65
3.4.1	<i>Allgemeines</i>	65
3.4.2	<i>Entstehung</i>	66
3.4.3	<i>Funktionen</i>	68
4	SYSTEMARCHITEKTUR.....	70
4.1	GRUNDIDEE	70
4.2	ANFORDERUNGEN.....	71
4.2.1	<i>SectionTagging und SectionRanking</i>	71
4.2.2	<i>JSPWiki-Kompatibilität</i>	72
4.2.3	<i>Browserunabhängigkeit</i>	72
4.2.4	<i>Datenbank</i>	73
4.2.5	<i>Zusätzliche Anforderungen</i>	73
4.3	DATENBANK-DIAGRAMM	74
4.3.1	<i>Tabelleninhalt</i>	75
4.3.2	<i>Spezifische Verwendung</i>	76
4.4	RANKING	77
4.4.1	<i>Verwendeter Algorithmus</i>	77
4.4.2	<i>Hypothesen</i>	77
4.5	ARCHITEKTUR VON JSPWIKI.....	79
4.5.1	<i>Plugin</i>	79
4.5.2	<i>Filter</i>	80
4.5.3	<i>Ablauf einer Seitengenerierung</i>	81
4.5.4	<i>Aufbau einer URL</i>	81
4.6	SECTIONTAGGING.....	84
4.7	SECTIONRANKING	84
5	IMPLEMENTIERUNG	85
5.1	SECTIONTAGGING.....	85
5.1.1	<i>Integrierung der SectionTagging-Komponente</i>	85
5.1.2	<i>Taganzeigefenster</i>	87
5.1.3	<i>Tagbearbeitungsfenster</i>	89

5.2	SECTIONRANKING	91
5.2.1	<i>Tagauswahl</i>	91
5.2.2	<i>SectionRanking-Fenster</i>	91
5.2.3	<i>Optionsmenü</i>	94
5.2.4	<i>Neue Wiki-Seite aus Sektionen erstellen</i>	95
5.3	VERWENDETE TECHNIKEN.....	96
5.3.1	<i>JavaScript</i>	96
5.3.2	<i>Cascading Style Sheets</i>	96
5.3.3	<i>Extensible Markup Language</i>	97
5.3.4	<i>Asynchronous JavaScript and XML</i>	98
5.3.5	<i>Yahoo! User Interface Library</i>	99
5.3.6	<i>Prepared Statements</i>	99
5.4	ABLAUF	101
5.4.1	<i>SectionTagging</i>	101
5.4.2	<i>SectionRanking</i>	103
5.5	INSTALLATION	104
5.5.1	<i>Systemanforderungen</i>	104
5.5.2	<i>Dateistruktur</i>	105
5.5.3	<i>JSPWiki Anpassungen</i>	107
5.5.4	<i>Filter aktivieren</i>	109
5.5.5	<i>Kodierung</i>	109
5.5.6	<i>Konfigurationsdatei</i>	110
6	KONKLUSION	112
7	AUSBLICK	113
III.	ABKÜRZUNGEN	114
IV.	ABBILDUNGSVERZEICHNIS	116
V.	FORMELVERZEICHNIS	118
VI.	QUELLCODEVERZEICHNIS	118
VII.	TABELLENVERZEICHNIS	119
VIII.	REFERENZEN	120
IX.	INDEX	129

1 Einleitung

Seit mittlerweile bereits mehreren Jahren erfreuen sich so genannte Tagging-Systeme zunehmender Beliebtheit. Sie erlauben es Benutzern, verschiedenen Ressourcen frei wählbare Schlagworte (sog. „Tags“) zuzuordnen um diese zu beschreiben. Diese Ressourcen können Bilder, Videos, Bookmarks sowie andere Inhalte aus dem Internet sein. [Marlow et al. 2006]

Der sich daraus ergebende Vorteil dieser Systeme liegt in einer neuartigen und persönlichen Organisation von bestimmten Inhalten sowie der damit einhergehenden, vereinfachten Navigation durch diese Inhalte. Im Vergleich zu herkömmlichen Taxonomien ist Tagging weder exklusiv noch hierarchisch [Golder und Huberman 2005]. Zudem ermöglichen sie – im Vergleich zu herkömmlichen Systemen – eine wesentlich einfachere Suche nach mit Tags markierten Ressourcen.

Ein Hauptgrund für die Beliebtheit von Tagging-Systemen ist die Erweiterung um die soziale Komponente. Das bedeutet, dass eigene Tags mit den Tags anderer Benutzer vernetzt werden. Die deshalb auch als „Social-Tagging-Systeme“ bezeichneten Applikationen bieten einige Vorteile. Es stehen nun auch die Tags der anderen Benutzer zur Verfügung, welche erkundet und durchsucht werden können um neue Ressourcen zu finden. [Furnas et al. 2006]

Eine der ersten Webseiten, die ein Tagging-System einsetzten, war Del.icio.us¹. Dort ist es möglich, Bookmarks mit Tags zu versehen. Bald darauf folgte Flickr², wo Bilder markiert werden konnten. Eine weitere bekannte Homepage, die ein solches System beinhaltet, ist Youtube³. Hier kann man Tags für Videos vergeben. So hat sich im Laufe der Jahre eine Vielzahl von Seiten, die Tagging-Systeme für unterschiedliche Inhalte und Zwecke einsetzen, entwickelt. Die verschiedenen Tagging-Systeme unterscheiden sich jedoch meist in der Art und Weise wie Tags vergeben werden können. Zudem wird die Vergabe von Tags oft auf eine bestimmte Ressource eingeschränkt.

Wie sieht jedoch die Vergabe von Tags für Texte in einem Wiki aus? Laut [Ebersbach et al. 2008] wird ein Wiki folgendermaßen definiert: *„Ein Wiki ist eine webbasierte Software, die es allen Betrachtern einer Seite erlaubt, den Inhalt zu ändern, indem sie diese Seite online im Browser editieren. Damit ist das Wiki eine einfach und leicht zu bedienende Plattform für kooperatives Arbeiten an Texten und Hypertexten.“* Es stellt sich auch die Frage, was genau mit Tags markiert werden soll: Soll es der gesamte Beitrag sein, jeder Abschnitt oder sogar einzelne Sätze? Grundsätzlich besitzen Wikis keine Tagging-Systeme, die direkt in die Webseite integriert sind und Zugriff auf die Ressource haben. Wird von der Webseite kein integriertes Tagging-System angeboten, können externe Tagging-Systeme eingesetzt werden, die zum Beispiel versuchen, anhand eines Links oder anderer Positionsangaben eine Verknüpfung zwischen Tag und der Ressource auf der Webseite herzustellen. Aber wie aus der Definition von Wiki zu erkennen ist, handelt es sich bei den Texten um dynamische Einträge, die sich im Laufe der Zeit ändern können. Somit kann es vorkommen, dass durch Änderungen im Wiki Verknüpfungen zwischen Tags und Ressourcen ungültig werden.

¹ <http://del.icio.us> (04.01.2010)

² <http://www.flickr.com> (04.01.2010)

³ <http://www.youtube.com> (04.01.2010)

Aus den oben ausgeführten Beschreibungen geht hervor, dass ein Tagging-System benötigt wird, welches an die speziellen Anforderungen eines Wiki-Systems angepasst ist. In Kapitel 2 wird auf das Thema Tagging-Systeme genau eingegangen. Dabei werden die einzelnen Bestandteile, die dem Tagging zu Grunde liegen, genau erläutert. Weiters wird in Kapitel 2.2 ein Überblick über die verfügbaren Tagging-Systeme gegeben. Es werden die Unterschiede der gelisteten Systeme aufgezeigt. Nach den Grundlagen von Tagging-Systemen und deren Variationen werden in Kapitel 3 Wiki-Systeme ausführlich beschrieben. Dabei wird besonders auf die JSPWiki Version, die für die Webseite des Austria-Forums⁴ Verwendung findet, näher eingegangen. In Kapitel 4 wird ein Prototyp eines Tagging-Systems definiert, welcher im Austria-Forum eingesetzt werden kann. Die Implementierung dieses Prototyps und die verwendeten Technologien werden in Kapitel 5 vorgestellt. Kapitel 6 und 7 enthalten das Resümee und geben einen Ausblick auf künftige, wünschenswerte Erweiterungen.

⁴ <http://www.austria-lexikon.at> (04.01.2010)

2 Tagging-Systeme

Kapitel 2.1 widmet sich der Theorie und den Grundlagen eines Tagging-Systems. Im Anschluss werden in Kapitel 2.2 verschiedene Tagging-Systeme im Einsatz vorgestellt.

2.1 Allgemeines

Wie bereits erwähnt ist es mit Hilfe von Tagging-Systemen möglich, bestimmten Ressourcen Tags zuzuweisen. Um die genaue Funktion dieser Systeme besser verstehen zu können, müssen ihre einzelnen Teile genauer betrachtet werden. Durch verschiedene Einstellungen und Eigenschaften entstehen unterschiedliche Ausprägungen. Diese beeinflussen den Anwender in seiner Tagwahl und beim Hinzufügen von Tags. Die Vergabe von Tags soll intuitiv und einfach gestaltet sein, um den Benutzer zur Eingabe verschiedener, für einen Inhalt treffender Tags, zu animieren. Jedes hinzugefügte Tag erweitert die Metadaten (= Daten über Daten) aller Benutzer und stellt somit eine Bereicherung für die Gemeinschaft dar [Marlow et al. 2006, Taylor 2003]. Ein Vorteil, der bei einer größeren Anzahl von Metadaten entsteht, ist, dass Inhalte jetzt eben unter vielen verschiedenen ihnen beschreibenden Tags gefunden werden können. Die Wahrscheinlichkeit ist also höher, dass jemand unter einem passenden Begriff sucht. Die Tags können auch für andere Zwecke genutzt werden, sei es für die Kategorisierung oder für Empfehlungen.

In dieser Arbeit wird zwischen externen und internen Tagging-Systemen unterschieden. Als externe Tagging-Systeme werden solche bezeichnet, die keinen direkten Einfluss auf die in der Webseite enthaltenen Ressourcen nehmen können, da sie nur Lesezugriff haben. Ihnen steht nur ein Link oder eine andere Pfadangabe zu einer Ressource zur Verfügung. Dies bedeutet, dass in externen Tagging-Systemen der originale Inhalt nicht geändert werden kann. Es kann nur durch nachträgliche Veränderungen im Browser auf die für den Benutzer dargestellte Webseite Einfluss genommen werden oder eine eigene Webseite generiert werden, in die die Ressource eingebunden oder verlinkt wird. Interne Tagging-Systeme haben hingegen direkten Zugriff auf die Ressource, weil sie in die Webseite integriert sind. Zudem können sie unmittelbar bei der Generierung der Webseite Einfluss auf deren Aufbau nehmen. Auch Änderungen in der Webseite wirken sich nicht zwingend nachteilig auf die Konsistenz und Integrität der Tag-Ressource-Verknüpfungen aus. Im Gegensatz dazu kann bei einem externen Tagging-System ein Link oder eine Positionsangabe auf eine Ressource innerhalb einer dynamischen Webseite in Kürze ungültig werden. Ein weiterer Vorteil eines internen Systems liegt darin, dass man Benutzerstatistiken der Webseite einsehen kann und diese zum Beispiel für Tagempfehlungen nutzen könnte. Beispiele für externe Systeme sind Del.icio.us⁵, SparTag.us⁶ und CiteULike⁷. Interne Tagging-Systeme werden bei Flickr⁸, YouTube⁹ und bei der Software von WordPress¹⁰ eingesetzt.

⁵ <http://www.delicious.com> (04.01.2010)

⁶ <http://www.spartag.us> (04.01.2010)

⁷ <http://www.citeulike.org> (04.01.2010)

⁸ <http://www.flickr.com> (04.01.2010)

⁹ <http://www.youtube.com> (04.01.2010)

¹⁰ <http://www.wordpress.org> (04.01.2010)

2.1.1 Das Tag

Wie schon in Kapitel 1 erwähnt, helfen Tags Ressourcen näher zu beschreiben. Jeder Benutzer darf frei aus seinem Wortschatz wählen, da Tags keinem geordneten Vokabular unterliegen. Ein Tag besteht zumeist aus einem einzelnen Wort. Es kann aber auch aus mehreren Wörtern bestehen. Es ist vom Tagging-System abhängig, ob auch Tags mit mehreren Wörtern unterstützt werden. Wird als Trennzeichen zwischen Tags ein Leerzeichen benutzt, so werden mit der Eingabe der Stadt „Rio De Janeiro“ drei Tags, nämlich „Rio“, „De“ und „Janeiro“, erzeugt. Eine einfache Möglichkeit, um dies zu verhindern, wäre, nach der Eingabe eines Tags diesen zu bestätigen bevor ein nächster eingegeben wird. Jedoch erhöht sich dadurch der Aufwand bei der Tagvergabe und deswegen ist diese Variante nicht gutzuheißen. Empfehlenswerter ist die Möglichkeit, die einzelnen Tags bei der Eingabe durch spezielle Satzzeichen voneinander zu trennen um den Eingabefluss nicht zu behindern.

2.1.2 Folksonomie

Diese Sammlung an Tags, die durch die freie Vergabe entsteht, wird Folksonomie genannt [Vander Wal 2005]. Dieser Begriff wurde von Thomas Vander Wal eingeführt und ist ein Neologismus zwischen den englischen Begriffen „Folk“ und „Taxonomy“ und bedeutet so viel wie „die pluralistische Stimme des Volkes“ [Müller-Prove 2008]. Damit ist gemeint, dass jeder Benutzer Tags zur Kategorisierung von Inhalten vergeben kann. Es sind keine Experten nötig, die ein so genanntes „Top-Down“-Vokabular aufbauen, sondern die Benutzer selbst erzeugen durch ihre Tags ein „Bottom-Up“-Vokabular, welches deren Wissen und deren Sicht der Dinge besser widerspiegelt als ein kontrolliertes Vokabular. [Macgregor und McCulloch 2006, Shirky 2005]

Ein Nachteil eines „Bottom-Up“-Vokabulars sind Störungen in Form von Synonymen, Homonymen, Homographen und anderen lexikalischen Anomalien [Macgergor und McCulloch 2006]. Synonyme bezeichnen verschiedene Wörter mit der gleichen oder fast gleichen Bedeutung, Homonyme hingegen Wörter mit mehreren Bedeutungen und Homographen sind Wörter mit gleicher Schreibweise aber mit unterschiedlicher Aussprache und Bedeutung. So kann ein Bild von einem Streichholz auch mit dem synonymen Tag „Zündholz“ versehen werden. Bei einer Suche mit dem Tag „Zündholz“ werden Ressourcen, die nur mit „Streichholz“ markiert worden sind, nicht gefunden. Auf der anderen Seite liefert eine Suche nach Homonymen und Homographen zu viele Ergebnisse. Wird also nach der Frucht „Kiwi“ gesucht, so werden auch Ressourcen zum Vogel „Kiwi“ angezeigt. Dasselbe passiert bei dem Homograph „August“. Dort werden bei einer Suche Ergebnisse des Monats „August“ und des Männervornamen „August“ geliefert. Mit einer Methode, die Synonyme anhand der Inhaltsähnlichkeit und der negativen Benutzerähnlichkeit erkennt, lässt sich das Problem mit Synonymen etwas eingrenzen [Clements et al. 2008]. Bei Homonymen lassen sich die Ergebnisse leicht mit einem zusätzlichen Suchwort, welches mit dem ursprünglichen Suchwort in Bezug steht, filtern [Golder und Huberman 2005].

Durch die freie Tagwahl kommt es vor, dass die Nutzer nicht nur die Grundform eines Wortes verwenden. Dadurch entstehende lexikalische Anomalien wie Mehrzahlbildungen, Flexionen und andere grammatikalische Abwandlungen. Diese Anomalien versucht man durch eine so genannte Normalisierung in den Griff zu bekommen. Eine Möglichkeit hierfür ist es, Wörter

auf ihre Grundform zurückzuführen (Lemmatisierung). Ein bekannter Algorithmus, der versucht ein Wort auf seinen Wortstamm zurückzuführen, ist der „Porter Stemming Algorithmus“ [Porter 1997]. Auch andere Formen der Normalisierung, wie Entfernung von Satz- und Sonderzeichen, Konvertierung in Kleinbuchstaben und Korrektur von Rechtschreibfehlern können herangezogen werden [Lohmann und Ziegler 2007]. Obwohl Tags frei vergeben werden können, sollte man sich überlegen, wie brauchbar diese für die Gemeinschaft sind. So können einfache Tagging-Regeln (z.B. Tags nur in Pluralform hinzufügen) bei der Eingabe von Tags helfen, lexikalische Anomalien größtenteils zu vermeiden und zu einem konvergenten Vokabular beizutragen [Mejías 2005].

Das zum Taggen verwendete Vokabular kann auf zwei Arten gespeichert werden. Als erstes kann ein so genanntes „Bag-model“ verwendet werden, welches die Tagzuweisung als Tripel <Benutzer, Tag, Ressource> abspeichert. Bei der zweiten Variante, dem „Set-model“, wird nur das Tupel <Tag, Ressource> verwendet. Bei ersterem können verschiedene Benutzer den gleichen Tag an die Ressource vergeben, was bei der anderen Variante nicht möglich ist, da das Tag nur einmal vergeben werden kann. Das Vokabular, welches die Tagzuweisung als Tripel speichert, wird auch „Broad Folksonomy“ genannt und erlaubt es, zum Beispiel Statistiken über Häufigkeiten von Tags durchzuführen. Bei der „Narrow Folksonomy“ hingegen, welche Tupels verwendet, lassen sich nur schwer Annahmen über das Tagverhalten der Benutzer machen. Dafür bietet es Vorteile beim Taggen von schwer auffindbaren Ressourcen, da Tags spezifischer vergeben werden. [Vander Wal 2005a, Marlow et al. 2006]

2.1.3 Tagklassen

Tags werden aus verschiedenen Motivationsgründen vergeben und diese lassen sich verschiedenen Gruppen zuweisen. Golder und Huberman [Golder und Huberman 2005] haben sieben verschiedene Klassen identifiziert, die bei Bookmarks verwendet werden.

1. Bestimmen, um was (oder wen) es sich handelt:
Dies entspricht der größten Gruppe von Tags und bezeichnet das Thema oder den Inhalt.
2. Bestimmen, um welche Ressource es sich handelt:
Es wird definiert, um welche Art von Ressource es sich handelt. Bei Bookmarks zum Beispiel ob es sich um einen Artikel, Blog oder ein Buch handelt.
3. Bestimmen, wem es gehört:
Der Ersteller der Ressource – bzw. wem diese gehört – wird festgelegt.
4. Verfeinerung der Kategorien:
Tags werden benutzt um Kategorien herzustellen oder zu verfeinern. Dafür werden oft Zahlen benutzt.
5. Qualitäten oder Charakteristiken bestimmen:
Dieser Punkt umfasst Eigenschaften, die den Inhalt bezeichnen – abhängig von der Meinung des Benutzers.

6. Selbstreferenz:

Tags die mit ihrer Bezeichnung einen Bezug auf den Ersteller des Tags herstellen, zum Beispiel mit dem Präfix „mein“ („meinUrlaub“, „meineWebseite“).

7. Aufgabenorganisation:

Um für eine Aufgabe alle benötigten Ressourcen wieder zu finden, werden Tags mit dem Namen der Aufgabe vergeben (z.B. „ausdrucken“).

Generell sind diese sieben Klassen nicht nur auf Bookmarks beschränkt. Auch auf andere Ressourcen lassen sich diese Klassen anwenden, wie folgendes Beispiel anhand eines Bildes in Abbildung 1 zeigt.

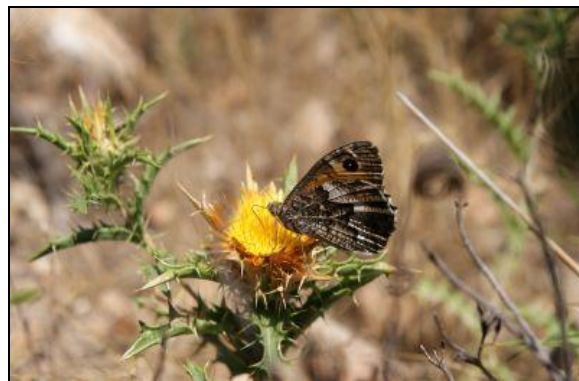


Abbildung 1: Bild für Tag-Klassifizierung

Klasse	Tags
1. Inhalt, Thema	Schmetterling, Blume
2. Art der Ressource	Bild, Makroaufnahme
3. Ersteller, Photograph	Wolfgang Pürstner
4. Kategorie	Tiere, Pflanzen
5. Eigenschaften	dornig, braun
6. Selbstreferenz	meinFoto
7. Aufgabenorganisation	Fotoalbum (z.B. <i>alle Bilder mit diesen Tags werden für ein Album verwendet</i>)

Tabelle 1: Tag-Klassifizierung

Golder und Huberman zeigen auch auf, dass die Tags, die in den ersten drei Klassen vergeben werden, nicht unbedingt mit der Person des Taggers in Verbindung stehen. Sie sind sachlicher Natur und beschreiben Fakten. Daher sind auch Überschneidungen der Tags bei einer hohen Anzahl von Taggern zu erwarten. Auch bei der vierten Klasse ist dies zu erwarten. Hingegen stehen die letzten drei Klassen in Bezug zum Tagger oder sind nur für ihn relevant.

Daher fassen Sen et al. [Sen et al. 2006] diese sieben Klassen zu drei allgemeineren Klassen in ihrer Studie mit Filmempfehlungen zusammen. Die erste Klasse „Sachliche Tags“ entspricht den ersten vier Klassen von Golder und Huberman. Die zweite Klasse „Subjektive Tags“ ist deckungsgleich mit der fünften Klasse und in der dritten Klasse von Sen et al. „Persönliche Tags“ werden die letzten zwei Klassen von Golder und Huberman zusammengefasst. Dadurch kann eine leichtere Zuordnung der Tags erfolgen. Jedoch treten Probleme bei der Zuordnung auf, wenn das Tag selbst nicht verständlich ist.

2.1.4 Einflussfaktoren beim Tagging

Schon einige Zeit vor der Geburt kann der Mensch mit seinen Sinnesorganen erste Eindrücke sammeln. Zusammen mit den ersten Wochen nach der Geburt entsteht in dieser Zeit das Grundmuster in unserem Gehirn. Grundlegende Fähigkeiten wie Verstehen und Denken kristallisieren sich heraus. In der Entwicklung werden die Lebens- und Lernerfahrungen durch Kultur, Religion, Familie und weitere Faktoren beeinflusst. Die Fähigkeit, lebenslang zu lernen, eröffnet weiters neue Möglichkeiten mit Problemen umzugehen. [Muchitsch 2007] Muchitsch schreibt weiters: *„Jeder Mensch ist dem Wesen nach ein Produkt seiner Umwelt und Genetik. Das gilt für sein Innenleben in gleicher Weise wie für alle Projektionen nach außen – also für die Summe seiner Reaktionen und Verhaltensweisen.“*

Diese unterschiedliche Prägung eines jeden Menschen und die damit einhergehenden Vorlieben, Abneigungen, Werte und Einstellungen führen auch zu Unterschieden im Tagverhalten. So kann zum Beispiel eine Person, welche Rosinen mag, ein Bild von einem Kaiserschmarrn mit Rosinen als „lecker“ taggen, hingegen wird eine Person, die Rosinen nicht mag, das Bild eher mit dem Tag „ungenießbar“ versehen. Nicht nur die Vorlieben, Erfahrungen und Eindrücke, die eine Person ihr Leben lang gesammelt hat, beeinflussen sie bei der Tagvergabe. Auch andere Faktoren spielen bei der Tagvergabe eine Rolle.

Grundsätzlich unterscheidet man bei der Eingabe von Tags zwischen drei verschiedenen Varianten. Jede davon beeinflusst das Tagverhalten von Benutzern zu einem gewissen Grad. Zum einen gibt es das „Blinde Tagging“. Hierbei werden während der Eingabe der Tags zu einer bestimmten Ressource die Tags der anderen Benutzer nicht angezeigt. Dadurch wird ein Einfluss der anderen Tags vermieden. Bei der zweiten Variante, „Sichtbares Tagging“ genannt, werden die Tags der anderen Benutzer angezeigt und bei der dritten Variante, „Empfohlenes Tagging“, werden nicht nur die Tags der anderen Benutzer angezeigt, sondern auch Tags, die das System empfiehlt. Diese empfohlenen Tags können unterschiedlichen Ursprungs sein. Entweder wurde eine ähnliche Ressource mit dem gleichen Tag versehen, der Tag wurde aus der Ressource selbst extrahiert oder über einen anderen Algorithmus beschafft. [Marlow et al. 2006]

Es ist nicht immer möglich alle Tags bei den Varianten „Sichtbares Tagging“ und „Empfohlenes Tagging“ anzuzeigen, da es vorkommen kann, dass eine Ressource zu viele Tags besitzt, um sie alle darstellen zu können. Hier werden dann Selektionsalgorithmen eingesetzt, die verschiedene Faktoren berücksichtigen. Die Anzahl der Benutzer, die ein bestimmtes Tag gesetzt haben, kann solch ein Faktor sein. [Sen et al. 2006]

Weiters zeigen Sen et al. in ihrer Studie den Einfluss des „Sichtbaren Taggings“ und des „Empfohlenen Taggings“ auf das Tagverhalten des Benutzers auf.

Selektionsalgorithmen

Bei den Selektionsalgorithmen unterscheidet man zwei große Gruppen. In der ersten Gruppe „Implizite Funktionen“ wird versucht die Reihenfolge der Tags ohne Benutzeraufwand zu ermitteln, um dadurch die Qualität der Tagvergabe zu verbessern. In der zweiten Gruppe der „Expliziten Funktionen“ fließen hingegen auch Benutzerbewertungen in den Algorithmus mit ein. Der bekannteste implizite Algorithmus ist der, bei dem die Anzahl der gleich vergebenen Tags zu einem bestimmten Inhalt gezählt wird und sich dadurch die Reihenfolge der Tags ergibt. Jedoch existieren zahlreiche Varianten um die Reihenfolge von Tags festzulegen. [Sen et al. 2009]

Feature	Hypothese	Beschreibung
<i>random</i>	-	Zufällige Anordnung der Tags
<i>num-item-apps</i>	Tags, die öfters einem bestimmten Inhalt zugeordnet wurden, sind relevanter	Reihung der Tags nach ihrer Anzahl, der sie einem Inhalt zugeordnet sind
<i>num-apps</i>	Tags, die öfters auf allen Inhalten vergeben wurden, sind relevanter	Reihung der Tags nach ihrer Anzahl, der sie allen Inhalten zugeordnet sind
<i>num-users</i>	Tags, die öfters von verschiedenen Benutzer über alle Inhalte vergeben wurden, sind relevanter	Reihung der Tags nach der Anzahl der Benutzer, die diese über alle Inhalte vergeben haben
<i>num-searches</i>	Tags, nach denen öfters gesucht wurde, sind relevanter	Reihung der Tags nach der Anzahl der Suchen nach ihnen
<i>num-search-users</i>	Tags, nach denen mehrere Benutzer gesucht haben, sind relevanter	Reihung der Tags nach der Anzahl der Benutzer, die nach ihnen gesucht haben
<i>avg-fraction-items-tagged</i>	Tags, die von dem User häufig vergeben wurden, sind wahrscheinlich Markierung-Tags und weniger relevant für die Gemeinschaft	Reihung der Tags anhand des Durchschnitts vom Anteil aller Inhalte, welche vom Benutzer markiert wurden und diesen Tag enthalten
<i>apps-per-item</i>	Tags, welche häufiger zu einem bestimmten Inhalt zugewiesen worden sind als zu anderen, sind relevanter	Reihung der Tags anhand der Durchschnittsanzahl der Zuweisungen zu einem Inhalt
<i>num-tag-words</i>	Tags mit mehreren Wörtern sind wenig wünschenswert	Reihung der Tags nach der Anzahl der Wörter in ihnen
<i>tag-length</i>	Tags mit sehr wenigen Buchstaben sind wenig wünschenswert	Reihung der Tags nach der Anzahl der Buchstaben in ihnen

Tabelle 2: Implizite Funktionen

In der Tabelle 2 und Tabelle 3 sind einige der häufigsten impliziten und expliziten Funktionen aufgelistet. Sie zeigen einen Teil aus einer Auflistung, welche unter [Sen et al. 2009] zu finden ist. Die Bayessche-Schätzfunktion wird bei den expliziten Funktionen verwendet, um Inhalte und Tags mit einer geringen Anzahl von Benutzerbewertungen anzupassen und so ein realistischeres Ergebnis zu erzielen.

Feature	Hypothese	Beschreibung
<i>global-avg</i>	Tags, die öfters bewertet wurden, sind relevanter	Reihung der Tags anhand der Bayessche-Schätzfunktion über den Anteil der positiven Bewertungen
<i>global-app-avg</i>	Tagzuweisungen, die zu einem höher bewerteten Inhalt gehören, sind relevanter	Reihung der Tagzuweisungen anhand der Bayessche-Schätzfunktion über den Anteil der positiven Bewertungen
<i>user-average</i>	Tags, die von den Benutzern höher bewertet wurden, sind relevanter	Reihung der Tags anhand der Bayessche-Schätzfunktion über den Anteil der positiven Bewertungen der Benutzer
<i>user-rating</i>	Tags, bei denen die Tagzuweisungen höher bewertet wurden, sind relevanter	Reihung der Tags anhand der Benutzerwertung der Tagzuweisungen
<i>normalized-global-avg</i>	Tags, nach denen mehrere Benutzer gesucht haben, sind relevanter	Reihung der Tags nach der Anzahl der Benutzer, die nach ihnen gesucht haben
<i>reputation</i>	Tags, die von Benutzern vergeben wurden die hoch bewertete Tags verwenden, sind relevanter	Reihung der Tags nach dem Durchschnittswert der vom Benutzer vergebenen Tags.

Tabelle 3: Explizite Funktionen

Bei der Studie von [Sen et al. 2009] liefern bei den impliziten Funktionen der bekannte Algorithmus „num-item-apps“ und der „apps-per-item“-Algorithmus gute Ergebnisse. Die Messung wurde mit Hilfe der Item-Top-3 Metrik durchgeführt, welche deren Einträge aus Umfrageergebnissen von Benutzern erhält. Am besten schneidet jedoch ein Algorithmus ab, der eine Mischung aus fünf verschiedenen impliziten Funktionen verwendet. Bei den expliziten Funktionen zeigt sich das gleiche Bild, bei denen sich auch durch eine Kombination verschiedener Algorithmen eine gute Leistung ergibt. Werden die impliziten und expliziten Funktionen zusammen verwendet, lässt sich die Qualität der Reihenfolge der Tags noch weiter steigern.

2.1.5 Rechte bei der Tagvergabe

Ein Kriterium bei einem Tagging-System sind die Rechte bei der Tagvergabe. Das Tagging-System kann zum Beispiel den Benutzer einschränken, sodass er nur für seine eigenen Ressourcen Tags vergeben kann oder es erlaubt jedem Benutzer jede Ressource zu taggen. Es sind auch weitere Lösungen möglich, wo man unterschiedlichen Gruppen von Benutzern verschiedene Rechte zum Taggen auf ihre eigenen Ressourcen einräumt. Nicht nur beim Hinzufügen von Tags können Rechte eine wichtige Rolle spielen. Auch beim Entfernen von Tags selbst gibt es Unterschiede. So kann entweder jedem, dem ursprünglichen Tagger, oder dem Besitzer der Ressource das Recht zum Entfernen eines Tags gegeben werden. [Marlow et al. 2006]

Verschiedene Systeme zu diesen Varianten werden in Kapitel 2.2 vorgestellt.

2.1.6 Arten der Tagvergabe

Ein Hauptaugenmerk eines Tagging-Systems liegt darin, dem Benutzer die Eingabe von Tags auf einfache Weise zu ermöglichen. Am weitesten ist die Type2Tag-Technik verbreitet, bei der der Benutzer seine Tags zu einem bestimmten Inhalt per Tastatur manuell in eine Textbox eingibt. Bei der zweiten Methode, der Click2tag-Technik, wird der zu vergebene Tag aus dem Inhalt oder einer angezeigten Liste ausgewählt. So wird der Aufwand, um einen Tag hinzuzufügen, möglichst gering gehalten. Leider können dadurch keine Tags gewählt werden, die sich nicht im Inhalt oder in der Liste mit den vorgeschlagenen Tags befinden. Eine Auswertung von Tags der Webseite del.icio.us¹¹ hat jedoch gezeigt, dass die Hälfte der vergebenen Tags auch selbst im Inhalt der Seite vorkommt [Hong et al. 2008]. Um auch Tags eingeben zu können, die nicht im Inhalt enthalten sind, können beide Techniken kombiniert werden, so wie beim Annotierungstool SparTag.us¹² (siehe Kapitel 2.2.3). Dort wird eine Textbox zur Tageingabe dargestellt und dahinter werden vorgeschlagene Tags angezeigt. Sollte dort oder im Text der Seite ein Term dabei sein den man eingeben möchte, so kann dieser durch einen Klick auf ihn als Tag hinzugefügt werden.

Durch die Click2Tag-Technik lassen sich aller Voraussicht nach auch einige Folksonomie-Probleme eindämmen, welche in Kapitel 2.1.2 erläutert wurden. Tippfehler, unterschiedliche Schreibweisen und verschiedene Numeri lassen sich dadurch größtenteils vermeiden. Bei der Type2Tag-Technik kann auch eine Autovervollständigung eingesetzt werden, um den gleichen gewünschten Effekt zu erzielen. Dies fördert nicht nur die leichtere Handhabung, sondern führt auch zu einem konvergenten Vokabular [Xu et al. 2006]. Werden Tags zum ersten Mal eingegeben, gibt es natürlich keine Autovervollständigung und dann kann zum Beispiel auch ein falsch geschriebenes Tag in die Liste aufgenommen werden. Mit der Zeit jedoch und bei einer Vielzahl an Tags und Benutzern kommt die „Selbstkorrektur“ der Folksonomie zum Tragen unter der Annahme, dass der Großteil der vergebenen Tags korrekt geschrieben und für die Gemeinschaft brauchbar ist [Oracle 2007]. Dabei werden die falsch geschriebenen Tags durch die korrekt geschriebenen, die demnach häufiger vorkommen, in der Autovervollständigung verdrängt.

¹¹ <http://del.icio.us> (13.02.2010)

¹² <http://spartag.us> (13.02.2010)

Um die beiden Verfahren Click2Tag und Type2Tag zu vergleichen führten Hong et al. eine empirische Studie durch. Dabei wurden Passagen mit im Durchschnitt 267 Wörtern Probanden zum Lesen gegeben. Einige Passagen sollten nur gelesen werden und andere wiederum sollten mit Hilfe von Click2Tag oder Type2Tag mit Tags versehen werden. Als Ergebnis war ein signifikanter Unterschied in der Lesezeit (vgl. Tabelle 4) zu erkennen. Erwartungsgemäß wurden die Passagen, wo keine Tags vergeben werden sollten, im Durchschnitt am schnellsten abgearbeitet. Auch Click2Tag schnitt signifikant besser als die Type2Tag-Technik ab, da die Zeit für die Eingabe der Tags per Tastatur entfällt. Es wurden auch Unterschiede in der Anzahl der vergebenen Tags festgestellt. Bei Click2Tag wurden im Durchschnitt mehr Tags vergeben, was auf die Einfachheit der Click2Tag-Technik gegenüber der Type2Tag-Technik zurückzuführen ist. [Hong et al. 2008]

	Lesezeit (Sekunden)	Anzahl der Tags
Keine Tags	81,38	0
Click2Tag	96,01	6,95
Type2Tag	102,92	3,98

Tabelle 4: Lesezeit und Anzahl der Tags [Hong et al. 2008]

Mit Hilfe einer Blickbewegungsregistrierung ist in Abbildung 2 das Blickverhalten eines Probanden zu beobachten. Rot markierte Stellen bedeuten eine längere Verweildauer auf dieser Position. Wie vorher angenommen fixierten bei Click2Tag die Probanden bestimmte Wörter im Text. Da die Tags direkt aus dem Inhalt übernommen werden, entsteht auf diese Weise eine „bottom-up“-Struktur. Im Gegensatz dazu wird bei Type2Tag zuerst der Text gelesen und anschließend werden Tags unter Einfluss bestehender Erfahrungen und vorhandenen Wissens in das Textfeld eingetragen, wodurch eine „top-down“-Struktur entsteht. Eine frühere Studie [Pirolli und Anderson 1985] führt zudem zu der Annahme, dass die erhöhte Aufmerksamkeit, die dem Text bei Click2Tag zu Gute kommt, auch dem verbesserten Erinnern der im Text enthaltenen Fakten dient. Dieses Phänomen bei Click2Tag wird auch durch die Ergebnisse der Studie von Budiu et al. bestätigt. [Hong et al. 2008, Budiu et al. 2009]

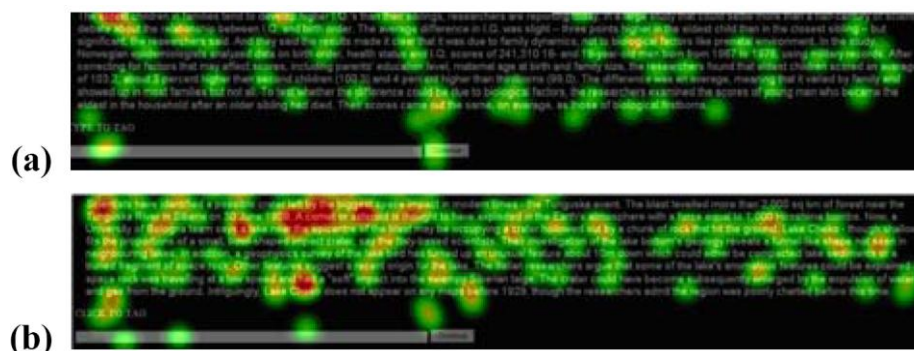


Abbildung 2: Blickbewegungen bei (a) Type2Tag und (b) Click2Tag [Hong et al. 2008]

Werden Tags mittels Click2Tag-Technik irrtümlich hinzugefügt, so erscheinen diese im Eingabefeld und müssen in der Folge wieder entfernt werden. Meistens wird dazu folgende

Vorgehensweise benutzt: Der Tastaturcursor wird mit einem Klick in das Eingabefeld gesetzt und man entfernt das falsche Tag mit der „Backspace“- oder „Entfernen“-Taste. Jedoch muss dafür die Tastatur benutzt werden, was einen zusätzlichen Aufwand erzeugt. Gänzlich auf die Tastatur verzichtet folgende Variante: Mit der Maus wird das falsche Tag markiert und danach wird das Browser-Menü mit einem Rechtsklick geöffnet und das Löschen-Kommando ausgewählt. Dazu sind drei Arbeitsschritte notwendig, nämlich Markieren, Rechtsklick und Auswählen des Löschkommandos. Um den Benutzer hier bei der Entfernung falsch eingegebener Tags mit Hilfe der Click2Tag-Technik zu unterstützen ist es sinnvoll, nach jedem Tag, der im Eingabefeld erscheint, ein Entfernen-Symbol anzuhängen. Dadurch kann ein falsches Tag mit nur einem Klick wieder entfernt werden und erspart dem Anwender zusätzlichen Aufwand.

Ein weiterer Vorteil der Click2Tag-Technik kommt bei mobilen Geräten, wie Mobiltelefonen, zum Vorschein. Dort ist meistens keine vollwertige Tastatur enthalten, was wiederum die Eingabe von Tags erschwert. Deswegen erfolgen Eingaben auf aktuellen Smartphones und Tablet-PCs größtenteils über einen integrierten Touchscreen. Dort können Tags zwar mit Hilfe der Bildschirmtastatur eingegeben werden, aber diese verringern natürlich auch die Anzeigefläche für den eigentlichen Inhalt. Wird jedoch Click2Tag verwendet, wird die Anzeigefläche nicht verkleinert und als Ersatz für die Maus dient entweder ein Stift oder der Finger. Durch das oftmalige Fehlen einer vollwertigen Tastatur bei mobilen Geräten sind auch andere Eingabemöglichkeiten in den Vordergrund getreten. Einige Geräte lassen sich mit Hilfe von Sprachbefehlen steuern. Bei der „Google Mobile App“¹³ für Smartphones beispielsweise kann der gewünschte Sucheintrag mittels Spracheingabe erfolgen. So könnte diese Spracherkennung auch dazu genutzt werden um Tags zu vergeben.

2.1.7 Navigation

Wie in den vorhergehenden Kapiteln bereits beschrieben dienen Tags unter anderem dazu, eine einfache Navigation zu ermöglichen und andere Inhalte zu erkunden. Es gibt verschiedene Ansätze, wie Tags angeordnet sein können, damit sie zur Navigation geeignet sind. Im folgenden Abschnitt werden diese kurz erläutert.

Listen

Der häufigste Ansatz ist es, Tags in einer Liste darzustellen. Dabei kann die Liste vertikal oder horizontal angeordnet werden. Auch die Einträge selbst können nach dem Alphabet sortiert sein oder nicht. Eine Liste, die alphabetisch sortiert ist, wird auch als Index bezeichnet. In Abbildung 3 ist ein Index, der die eigenen Tags auf Flickr darstellt, zu sehen. Dort werden nur die 150 häufigsten selbst vergebenen Tags angezeigt, da der Index sonst zu unübersichtlich wäre.

¹³ <http://www.google.de/mobile/google-mobile-app/> (15.03.2010)

	Tag	Entered as	On		
A	alghero	alghero	2 items	Edit	Delete
	beach	beach	2 items	Edit	Delete
B	bear	bear	1 item	Edit	Delete
	bed	bed	1 item	Edit	Delete
	boat	boat	1 item	Edit	Delete
	bug	bug	1 item	Edit	Delete
	butterfly	butterfly	1 item	Edit	Delete
	church	church	2 items	Edit	Delete
C	city	city	1 item	Edit	Delete
	cork	cork	1 item	Edit	Delete
	flower	flower	3 items	Edit	Delete
F	forest	forest	2 items	Edit	Delete
	girl	girl	1 item	Edit	Delete
G	grassland	grassland	1 item	Edit	Delete
	gull	gull	1 item	Edit	Delete
	isolaspargi	isola spargi	1 item	Edit	Delete
I					
M	mountain	mountain	1 item	Edit	Delete
	oak	oak	1 item	Edit	Delete
O					
P	path	path	1 item	Edit	Delete
	ristorante	ristorante	1 item	Edit	Delete
R	room	room	1 item	Edit	Delete
	sea	sea	3 items	Edit	Delete
S	stairs	stairs	1 item	Edit	Delete
	street	street	1 item	Edit	Delete
	view	view	1 item	Edit	Delete
V					

Abbildung 3: Taglist Flickr

Wird ein Tag angeklickt, gelangt man zu den Inhalten, die diesen Tag erhalten haben. So besteht auf einfache Weise die Möglichkeit, verschiedene Inhalte zu einem Thema zu durchforsten.

Tag-Clouds

Oft wird nicht nach einem bestimmten Tag gesucht, sondern es soll einfach der Inhalt anderer Benutzer durchstöbert werden. Dazu eignet sich eine andere Navigationsmöglichkeit, nämlich die Tag-Cloud (Schlagwortwolke). Sie stellt eine Menge von Wörtern (in diesem Fall Tags) dar und passt die Schriftgröße oder Schriftfarbe einer Eigenschaft (z.B. Häufigkeit) des Tags an [Rivadeneira et al. 2007]. Je größer zum Beispiel das Tag in der Tag-Cloud abgebildet ist, desto häufiger wurde dieses vergeben (siehe Abbildung 4).

Abbildung 4: Tagcloud Flickr¹⁴

So lassen sich auf den ersten Blick die beliebtesten Tags erkennen. Durch Klicken auf das jeweilige Tag gelangt man zu dessen Inhalten und zu anderen in Bezug stehenden Tags. Seit dem ersten Einsatz einer Tag-Cloud im Jahr 2002 [Flanagan 2002] erfolgten viele Neuerungen, die den Umgang mit Tag-Clouds erleichtern und deren Visualisierungen verbessern. Aus den statischen und nach dem Alphabet sortierten Schlagwortwolken haben sich interaktive und zum Teil auch animierte Tag-Clouds entwickelt. Mit Hilfe von Clustering können ähnliche oder bezugsnahe Tags gefunden werden [Begelman et al. 2006]. Diese können dann in der Tag-Cloud nebeneinander angeordnet werden, um die Verbindung zwischen den Tags auch in der Visualisierung zu verdeutlichen. Diesem Ansatz folgen auch Montero und Solana [Montero und Solana 2006]. Sie wählen Tags nach ihrer Nützlichkeit aus und ordnen diese in Bezug zueinander in der Wolke an. Ein Überblick über die Grundlagen und verschiedenen Variationen von Tag-Clouds ist unter [Pürstner 2007] zu finden. In der Abbildung 5 ist eine animierte Tag-Cloud zu erkennen, welche nach einem Klick auf ein Tag dieses hervorhebt und dessen Verbindungen zu anderen Tags anzeigt.

¹⁴ <http://www.flickr.com> (04.01.2010)

Abbildung 5: Elastic Tag Maps¹⁵

In einer Studie von Halvey und Kean [Halvey und Kean 2007] wurden horizontale Listen, vertikale Listen und Tag-Clouds – jeweils nach dem Alphabet sortiert bzw. unsortiert – miteinander verglichen, um zu zeigen, welche Darstellung die Suche nach einem Tag erleichtert. Ziel der Probanden war es, ein bestimmtes Tag, das sie angezeigt bekamen, in der Liste oder in der Tag-Cloud möglichst schnell zu markieren. Das Ergebnis der Studie war, dass alphabetische Sortierungen und größere Schriftgrößen einzelner Tags einen positiven Einfluss auf die Geschwindigkeit bei der Tagfindung haben. Weiters scheinen Benutzer dazu zu neigen, Listen und Tag-Clouds eher zu scannen als zu lesen. Die Probanden brauchten auch länger bei Tag-Clouds um die gesuchten Tags zu finden, was dafür spricht, dass Tag-Clouds zur gezielten Suche von bestimmten Tags nicht geeignet sind.

Eine Studie von Helic et al. zeigt, dass die Annahme, dass Tag-Clouds eine effiziente Navigation unterstützen, sich in einigen Fällen nicht bewahrheitet, obwohl sich durch verschiedene Anpassungen im Benutzerinterface die Verwendbarkeit der Tag-Clouds zur Navigation verändern lässt. [Helic et al. 2010]

¹⁵ http://well-formed-data.net/experiments/tag_maps_v5/ (05.01.2010)

2.1.8 Zusammenfassung

In den vorigen Kapiteln wurden ausführlich die verschiedenen Aspekte rund um das Thema Tagging beleuchtet. Dabei wurde klar, dass bei der Erstellung eines Tagging-Systems schon im Vorhinein klar gestellt sein muss wofür dieses genau eingesetzt wird. Das Hauptaugenmerk liegt hierbei auf der Art und Weise wie ein Tag eingegeben wird. Dort gibt es die meisten Gestaltungsmöglichkeiten, die das daraus resultierende Vokabular beeinflussen.

Um Unterschiede in der Folksonomie zwischen den Methoden Type2Tag und Click2Tag aufzuzeigen bedarf es noch weiteren Studien, die diesen Sachverhalt näher untersuchen. Anzunehmen ist aber, dass das Vokabular bei Click2Tag schneller konvergiert, da gewisse Störungen in der Folksonomie durch die beschränkte Wortwahl leichter vermieden werden können. Je nachdem, welche Systemkonfiguration zur Verfügung steht, ergeben sich weitere Unterschiede in der Eingabe. So kann ein Tag per Tastatur, per Mausklick oder per Sprachbefehl eingegeben werden. Wird das Tag mit Hilfe der Tastatur eingegeben, so sind Tagging-Regeln hilfreich um ein konvergenteres Vokabular zu erreichen. Jedoch sollen diese Regeln den Benutzer nicht hindern, seine Tags einzugeben, sondern eher einen Leitfaden darstellen. Die Eingabe der Tags sollte intuitiv aufgebaut sein, um den Benutzer bei der Vergabe zu unterstützen und nicht zu behindern.

Oft unterscheiden sich Tagging-Systeme in der Eingabe von Tags, die aus mehreren Wörtern bestehen. Einige Systeme verlangen einen Beistrich als Trennzeichen zwischen den Tags. Bei anderen wiederum muss der Tag, der aus mehreren Wörtern besteht, in Hochkommas eingefasst werden, wenn die Tags voneinander durch Leerzeichen getrennt sind. Deswegen ist es sinnvoll, dem Anwender durch einen Hinweis die verwendete Methode mitzuteilen.

2.2 Beschreibung vorhandener Systeme

Im folgenden Unterkapitel werden einige Tagging-Systeme vorgestellt, die unterschiedliche Inhalte mit Tags versehen können. Dabei wird nicht auf die volle Funktionalität der Systeme eingegangen, sondern es wird der Vorgang des Taggens näher beschrieben, da sich die Systeme erheblich in der Art und Weise unterscheiden, in der Tags vergeben werden können. Als erstes werden Systeme gezeigt, die manuell vom Benutzer bedient werden. Danach sind halbautomatische Systeme an der Reihe, die den Benutzer nur zur Taggenerierung benötigen und zuletzt werden Systeme gezeigt, die vollkommen automatisiert Tags aus Inhalten generieren können. Anschließend wird in Kapitel 2.2.6 ein Überblick über die vorgestellten Systeme gegeben.

2.2.1 Scuttle

Scuttle¹⁶ ist ein webbasiertes frei verfügbares Bookmarking-System. Die Software wurde in PHP geschrieben und steht unter der GNU General Public License (GPL). Dadurch ist es möglich, sein eigenes Bookmarking-System zu erstellen. Scuttle erlaubt Benutzern, Links zu speichern und mit anderen Benutzern zu teilen. Das Programm ist darauf ausgerichtet, den Benutzer bei der Vergabe von Tags zu unterstützen.

Tags hinzufügen

Um Tags erstellen zu können gibt es verschiedene Möglichkeiten. Zuerst muss jedoch die URL eingegeben werden, welche die Tags erhalten soll. Entweder gibt man die URL, die getaggt werden soll, manuell im Menü von Scuttle ein (Abbildung 6) oder durch einen Klick auf ein Scuttle-Lesezeichen, welches vorher in einer Symbolleiste hinzugefügt wurde. Es besteht auch die Möglichkeit über ein Firefox-Plugin (Scuttle 0.3.1) eine URL hinzuzufügen. Durch einen Klick auf das Lesezeichen öffnet sich das Menü „Favorit hinzufügen“ von Scuttle und die URL und der Titel sind schon in dem jeweiligen Eingabefeld eingetragen. Dadurch entfällt eine manuelle Eingabe der URL und die Tags können – mit einem Beistrich voneinander getrennt – eingegeben werden. Durch Klick auf einzelne Tags in der Tag-Cloud „Populäre Tags“ werden diese dem Tageingabefeld automatisch hinzugefügt.

¹⁶ <http://sourceforge.net/projects/scuttle/> (19.01.2010)

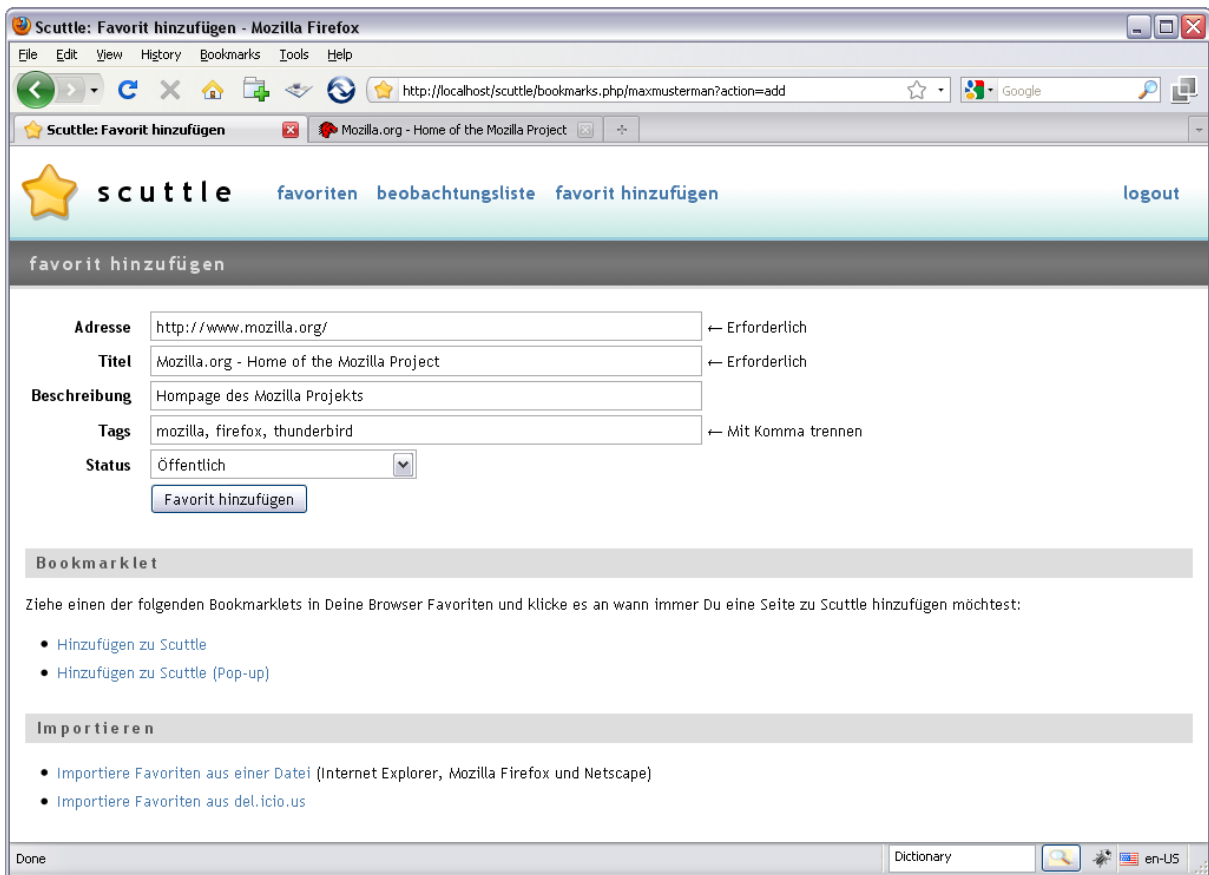


Abbildung 6: Scuttle - Favorit hinzufügen

2.2.2 Flickr

Bei der Webseite Flickr¹⁷ handelt es sich um ein Serviceportal zum Hochladen von Bildern und Videos, um diese mit Familie, Freunden und mit anderen Mitgliedern zu teilen. Die Inhalte können unter anderem direkt im Portal mit Tags und Kommentaren versehen werden. Weiters besteht die Möglichkeit, sie auf einer Landkarte zu positionieren und Personen, die darauf vorkommen, zuzuordnen. Hauptsächlich werden Bilder mit Tags belegt, da diese den größten Teil der Inhalte ausmachen. Auch hier unterstützt das interne Tagging-System den Benutzer bei der Vergabe von Tags.

¹⁷ <http://www.flickr.com> (19.01.2010)

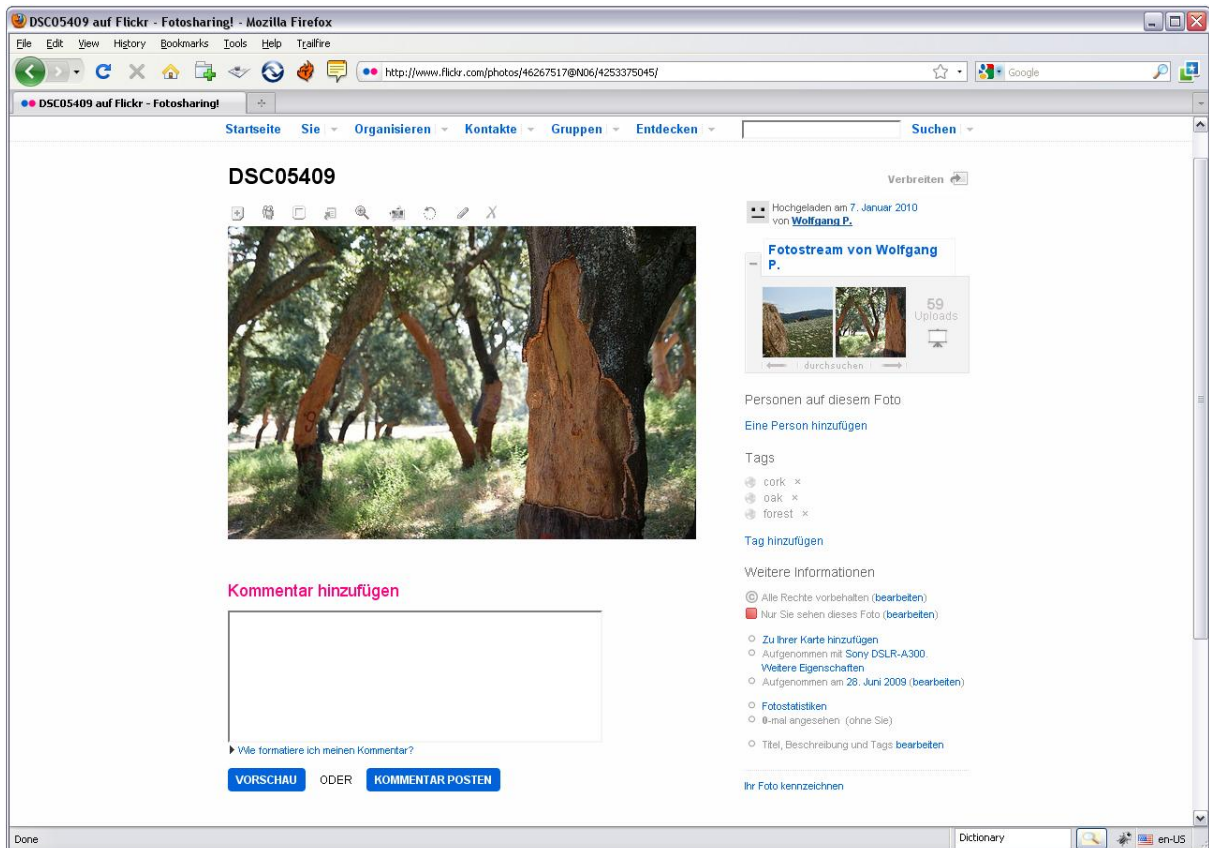


Abbildung 7: Flickr - Bildübersicht

Tags können auf eigene Bilder vergeben werden und auf Bilder von Familie und Freunden, sofern diese dafür freigegeben sind. Sichtbar sind sie dennoch für alle, die Zugriff auf das Bild haben. Die Tagliste, rechts neben dem Bild in Abbildung 7 erkennbar, enthält drei Einträge pro Zeile. Der erste Eintrag, ein Symbol der Erde, zeigt auf Klick alle Inhalte von allen Benutzern an, die den gleichen Tag besitzen. Wird der Tag direkt angeklickt, erscheinen nur die eigenen Inhalte, die mit diesem Tag markiert worden sind. Das letzte Symbol dient zum Löschen des Tags.

Tags hinzufügen

Durch einen Klick auf „Tag hinzufügen“ (Abbildung 7) erscheint ein Textfeld (Abbildung 8), in dem die Tags eingegeben werden können. Eine Anleitung, wie Tags, die aus mehreren Wörtern bestehen, eingegeben werden können, wird auch angezeigt.

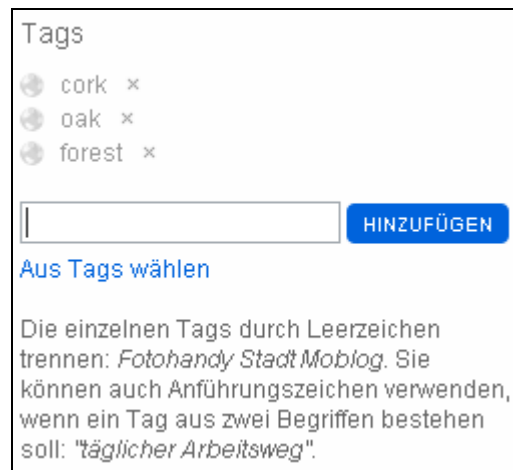


Abbildung 8: Flickr - Tag hinzufügen

Click2Tag wird auch unterstützt indem die eigenen Tags durch Klick auf „Aus Tags wählen“ angezeigt werden. Bereits verwendete Tags zu diesem Inhalt werden in einer anderen Farbe als die übrigen Tags dargestellt (Abbildung 9).

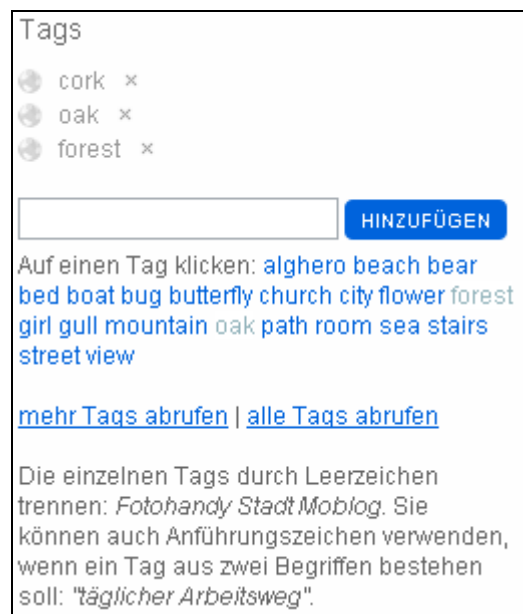


Abbildung 9: Flickr - Aus Tags auswählen

Das Empfehlen von Tags wird nicht unterstützt. Dennoch gibt es einige Studien ([Garg und Weber 2008a, Garg und Weber 2008b, Sigurbjörnsson und Van Zwol 2008]), die sich mit diesem Thema befassen und Möglichkeiten für dessen Einsatz auf Flickr aufzeigen. Ferner gibt es eine mobile Software namens ZoneTag¹⁸, die Tagempfehlungen für Flickr ermöglicht. ZoneTag ist eine Applikation für Mobiltelefone, welche es erlaubt, die mit der Kamera aufgenommenen Fotos direkt auf Flickr hochzuladen. Beim Upload werden Tags empfohlen, die für das Bild verwendet werden können. Diese werden aufgrund des Standorts, der Zeit und der persönlichen Tagging-Gewohnheit ermittelt [Ames und Naaman 2007].

¹⁸ <http://zonetag.research.yahoo.com> (19.01.2010)

2.2.3 SparTag.us

SparTag.us¹⁹ ist ein soziales Annotierungstool, um Paragraphen auf verschiedenen Webseiten mit Tags zu versehen und um Webinhalte zu markieren. Es besteht aus einer Client-Server-Architektur. Auf der Client-Seite wird ein Firefox-Plugin verwendet und auf der Server-Seite läuft ein Apache Tomcat Server mit Java Servlets und Anschluss an eine MySQL-Datenbank. [PARC 2009a]

Das Firefox-Plugin, zur Zeit dieser Arbeit in Version 1.0.3 verfügbar, welches den Client darstellt, zeigt nach der Installation eine Symbolleiste (Abbildung 10) im Browser an, die zur Bedienung der einzelnen Elemente dient.

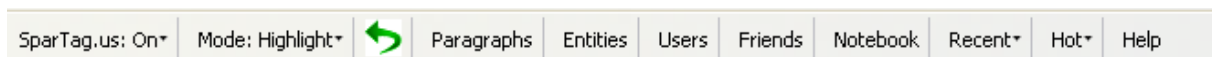


Abbildung 10: SparTag.us - Symbolleiste

Tags hinzufügen

Ist in der Symbolleiste SparTag.us eingeschaltet, so wird jede Webseite, die in den Browser geladen wird, so vom Plugin verändert, dass jeder Paragraph und jedes Wort in diesem anklickbar gemacht wird. Wird der Mauszeiger über einen tagbaren Paragraphen bewegt, so verändert der Paragraph seine Hintergrundfarbe, sodass der genaue Abschnitt des Paragraphen erkennbar ist (vgl. Abbildung 11 und Abbildung 12).

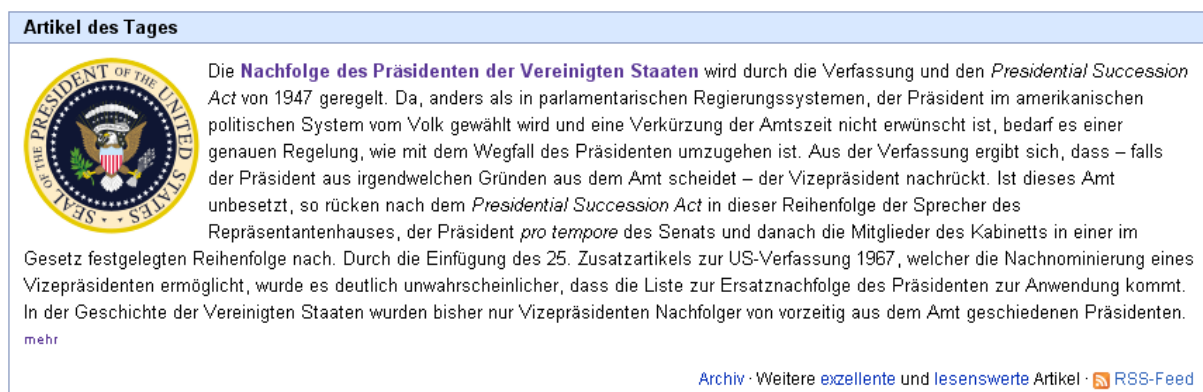



Abbildung 11: Artikel des Tages²⁰ – Original

¹⁹ <http://spartag.us> (20.01.2010)

²⁰ <http://de.wikipedia.org/wiki/Wikipedia:Hauptseite> (20.01.2010)

Artikel des Tages



Die **Nachfolge des Präsidenten der Vereinigten Staaten** wird durch die Verfassung und den *Presidential Succession Act* von 1947 geregelt. Da, anders als in parlamentarischen Regierungssystemen, der Präsident im amerikanischen politischen System vom Volk gewählt wird und eine Verkürzung der Amtszeit nicht erwünscht ist, bedarf es einer genauen Regelung, wie mit dem Wegfall des Präsidenten umzugehen ist. Aus der Verfassung ergibt sich, dass – falls der Präsident aus irgendwelchen Gründen aus dem Amt scheidet – der Vizepräsident nachrückt. Ist dieses Amt unbesetzt, so rücken nach dem *Presidential Succession Act* in dieser Reihenfolge der Sprecher des Repräsentantenhauses, der Präsident *pro tempore* des Senats und danach die Mitglieder des Kabinetts in einer im Gesetz festgelegten Reihenfolge nach. Durch die Einfügung des 25. Zusatzartikels zur US-Verfassung 1967, welcher die Nachnominierung eines Vizepräsidenten ermöglicht, wurde es deutlich unwahrscheinlicher, dass die Liste zur Ersatznachfolge des Präsidenten zur Anwendung kommt. In der Geschichte der Vereinigten Staaten wurden bisher nur Vizepräsidenten Nachfolger von vorzeitig aus dem Amt geschiedenen Präsidenten.


[mehr](#)

Archiv · Weitere [exzellente](#) und [lesenswerte](#) Artikel · [RSS-Feed](#)

Abbildung 12: Artikel des Tages – Hervorgehoben

Um nun ein Tag hinzufügen zu können, genügt ein Klick in den Paragraph und am Ende dieses wird eine Textbox (siehe Abbildung 13) zur Eingabe der Tags angezeigt. Es besteht auch die Möglichkeit direkt das Wort im Text anzuklicken um es als Tag hinzuzufügen.

Artikel des Tages



Die **Nachfolge des Präsidenten der Vereinigten Staaten** wird durch die Verfassung und den *Presidential Succession Act* von 1947 geregelt. Da, anders als in parlamentarischen Regierungssystemen, der Präsident im amerikanischen politischen System vom Volk gewählt wird und eine Verkürzung der Amtszeit nicht erwünscht ist, bedarf es einer genauen Regelung, wie mit dem Wegfall des Präsidenten umzugehen ist. Aus der Verfassung ergibt sich, dass – falls der Präsident aus irgendwelchen Gründen aus dem Amt scheidet – der Vizepräsident nachrückt. Ist dieses Amt unbesetzt, so rücken nach dem *Presidential Succession Act* in dieser Reihenfolge der Sprecher des Repräsentantenhauses, der Präsident *pro tempore* des Senats und danach die Mitglieder des Kabinetts in einer im Gesetz festgelegten Reihenfolge nach. Durch die Einfügung des 25. Zusatzartikels zur US-Verfassung 1967, welcher die Nachnominierung eines Vizepräsidenten ermöglicht, wurde es deutlich unwahrscheinlicher, dass die Liste zur Ersatznachfolge des Präsidenten zur Anwendung kommt. In der Geschichte der Vereinigten Staaten wurden bisher nur Vizepräsidenten Nachfolger von vorzeitig aus dem Amt geschiedenen Präsidenten.

[mehr](#)


wolfgangpue: Verfassung
save
delete
email

Archiv · Weitere [exzellente](#) und [lesenswerte](#) Artikel · [RSS-Feed](#)

Abbildung 13: Artikel des Tages – Tagging

Nach der Eingabe der Tags per Tastatur oder per Click2Tag können die Tags mit der Schaltfläche „save“ gespeichert werden. Danach erscheinen im Anschluss an den Paragraphen die vergebenen Tags und deren Autor. Beim nächsten Besuch der Seite werden erneut die vergebenen Tags angezeigt (siehe Abbildung 14). Beim Speichern werden nicht nur die Tags, der Autor und der Link auf die Webseite an den Server gesendet, sondern die ganze Webseite wird mitgeschickt, um Veränderungen in der Seite beim nächsten Aufruf erkennen zu können [PARC 2009a].

Artikel des Tages



Die **Nachfolge des Präsidenten der Vereinigten Staaten** wird durch die Verfassung und den *Presidential Succession Act* von 1947 geregelt. Da, anders als in parlamentarischen Regierungssystemen, der Präsident im amerikanischen politischen System vom Volk gewählt wird und eine Verkürzung der Amtszeit nicht erwünscht ist, bedarf es einer genauen Regelung, wie mit dem Wegfall des Präsidenten umzugehen ist. Aus der Verfassung ergibt sich, dass – falls der Präsident aus irgendwelchen Gründen aus dem Amt scheidet – der Vizepräsident nachrückt. Ist dieses Amt unbesetzt, so rücken nach dem *Presidential Succession Act* in dieser Reihenfolge der Sprecher des Repräsentantenhauses, der Präsident *pro tempore* des Senats und danach die Mitglieder des Kabinetts in einer im Gesetz festgelegten Reihenfolge nach. Durch die Einfügung des 25. Zusatzartikels zur US-Verfassung 1967, welcher die Nachnominierung eines Vizepräsidenten ermöglicht, wurde es deutlich unwahrscheinlicher, dass die Liste zur Ersatznachfolge des Präsidenten zur Anwendung kommt. In der Geschichte der Vereinigten Staaten wurden bisher nur Vizepräsidenten Nachfolger von vorzeitig aus dem Amt geschiedenen Präsidenten.

[mehr](#)

wolfgangpue: Verfassung

[Archiv](#) · Weitere [exzellente](#) und [lesenswerte](#) Artikel · [RSS-Feed](#)

Abbildung 14: Artikel des Tages - Taganzeige

In einer Studie stellte sich heraus, dass ungefähr die Hälfte der Tags, die auf Del.icio.us vergeben wurden, auch im Inhalt der Webseite selbst vorkommt. Dieser Umstand wird auch bei Spartag.us berücksichtigt und Click2Tag eine entscheidende Rolle bei der Vergabe von Tags zugeordnet, da jedes Wort im Paragraph durch einen Klick in die Tagliste hinzugefügt werden kann. Dadurch wird der Aufwand für den Benutzer weiter verringert. Auch das Eingabefeld erscheint nicht in einem speziellen Fenster, sondern wird direkt nach dem Paragraph in die Webseite integriert, was die Aufmerksamkeit des Benutzers beim gelesenen Paragraphen belässt. [Hong et al. 2008]

Eine weitere Funktion unterstützt den Benutzer bei der Vergabe von Tags. Mit der Schaltfläche „Paragraphs“ lassen sich jene Paragraphen sichtbar machen, die mit Tags markiert werden können. In Abbildung 15 ist eine Webseite einmal ohne aktivierte Paragraphenumrandung und in Abbildung 16 einmal mit aktivierter Paragraphenumrandung zu sehen. Dadurch lassen sich die Bereiche sofort erkennen, welche zum Taggen geeignet sind.



Abbildung 15: Wikipedia Startseite²¹ - Ohne Paragraphenhervorhebung

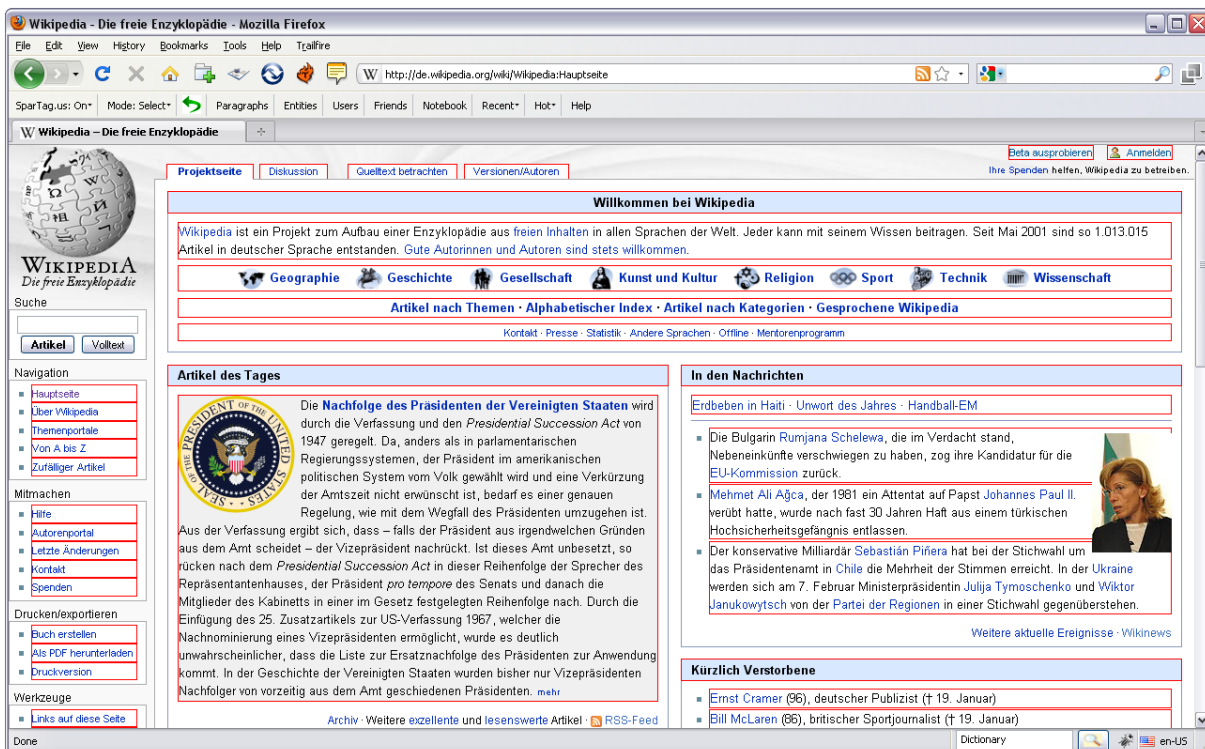


Abbildung 16: Wikipedia Startseite - Mit Paragraphenhervorhebung

²¹ <http://de.wikipedia.org/wiki/Wikipedia:Hauptseite> (20.01.2010)

2.2.4 ESP Game

Jedes Jahr werden Milliarden an Stunden mit Computerspielen verbracht. Wie wäre es, wenn die ganze Zeit und Energie, die dafür aufgewendet wird, zur Lösung von komplexen Problemen verwendet werden würde? Trotz der stetigen Weiterentwicklungen in Bereichen der künstlichen Intelligenz oder visuellen Wahrnehmung können Maschinen es in bestimmten Bereichen nicht mit der Leistungsfähigkeit des menschlichen Gehirns aufnehmen. Um einen Menschen dazu zu bringen, seine Fähigkeiten konstruktiv zur Lösung von Aufgaben einzusetzen, braucht dieser einen Anreiz. Onlinespiele sind eine gute Methode um diesen Anreiz zu liefern. Dazu wird die Durchführung einer Aufgabe, um ein Problem zu lösen, in ein Spiel verpackt. [von Ahn 2006]

Das ESP-Game²² ist ein solches Spiel und wurde von Luis von Ahn entwickelt. Grundgedanke dahinter ist, dass im Internet Millionen von Bildern existieren und es keine textuellen Beschreibungen dazu gibt. Bilderkennungsalgorithmen können den Inhalt noch nicht in einem zufrieden stellenden Maß identifizieren. Einige andere Algorithmen versuchen anhand der Webseite, in der das Bild eingebettet ist, den Inhalt des Bildes zu bestimmen. Diese Methode ist aber ineffizient, da oft der Text um das Bild herum spärlich ist und dieser auch zu falschen Beschreibungen führen kann [Carson und Ogle 1996]. Beschreibungen (i.e.S. Tags) von Bildern sind aber für Bildsuchmaschinen von enormer Bedeutung. Der Mensch kann noch am besten den Inhalt eines Bildes bestimmen und in Form eines Spieles wird dieser dazu gebracht von sich aus Bilder mit Tags zu versehen, da es für ihn unterhaltsam ist. [von Ahn und Dabbish 2004]

Spielbeschreibung

Das ESP-Game (Abbildung 17) wird zu zweit gespielt, wobei der Partner zufällig aus allen Spielern, die momentan online sind, ausgewählt wird. Beide können nicht miteinander kommunizieren und bekommen nach dem Start des Spiels das gleiche Bild angezeigt. Nun versuchen beide Tags einzugeben, die das Bild beschreiben und von denen sie glauben, dass der andere Mitspieler sie auch eingeben könnte. Ziel ist es, eine Übereinstimmung eines Tags mit dem Partner zu erzielen um zum nächsten Bild zu gelangen. Dabei kommt es nicht auf die Reihenfolge der Eingaben an, sondern nur, ob der Partner auch dasselbe Tag verwendet hat. Umso mehr Bilder, also Übereinstimmungen, innerhalb von 2,5 Minuten geschafft werden, umso mehr Spielpunkte werden beiden gutgeschrieben. Aus Sicht des Spielers geht es also nur darum, die Begriffe des Partners zu einem Bild zu erraten. Es besteht auch die Möglichkeit ein schwieriges Bild zu passen, sollte keine Übereinstimmung zustande kommen. Ein Bild kann aber nur dann gepasst werden, wenn beide Spieler damit einverstanden sind. Ein wichtiges Element dieses Spiels sind die Tabu-Wörter. Diese sind Tags, die zu dem Bild passen und nicht zum Beschreiben eingegeben werden dürfen. Die Tabu-Wörter stammen von vorhergehenden Spielrunden und sind Übereinstimmungen der Spieler gewesen. Wird das Bild also zum ersten Mal verwendet, so existieren noch keine Tabu-Wörter. Dadurch, dass man gewisse Begriffe nicht mehr verwenden darf, muss man sich spezifischere Tags überlegen, die der Partner auch nehmen könnte und dies führt zu einer umfangreicheren Beschreibung des Bildes. [von Ahn und Dabbish 2004]

²² <http://www.espgame.org> (03.02.2010)

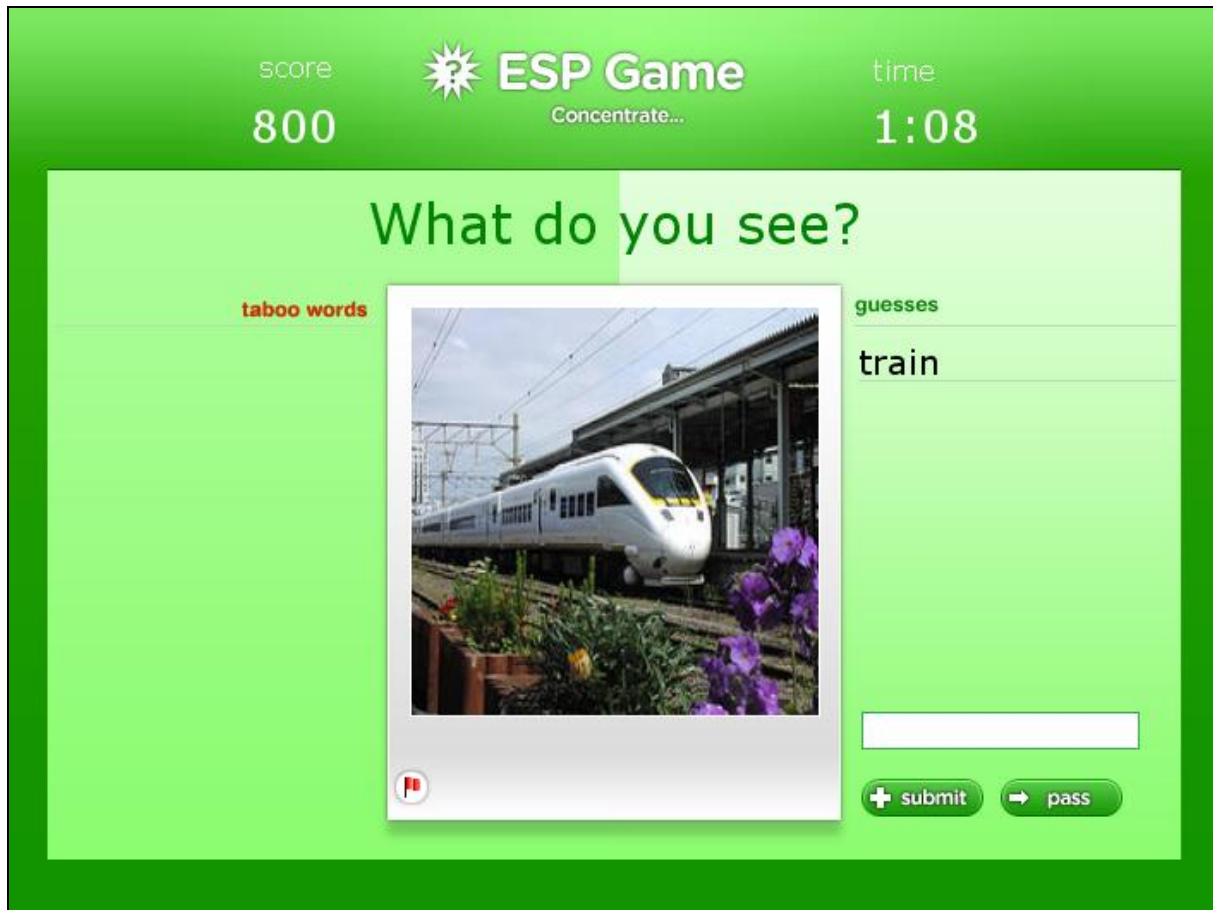


Abbildung 17: ESP-Game

Weitere Eigenschaften

Besitzt ein Bild genug Tabu-Wörter die es beschreiben und konnte in den letzten Spielrunden keine neue Übereinstimmung mehr erzielt werden, so kann das Bild aus dem Spiel genommen werden, da es schon genug Tags hat, die den Inhalt genau beschreiben. Da das Spiel immer zu zweit gespielt wird, kann es vorkommen, dass ein Spieler bei der Partnerzuweisung übrig bleibt. Dieser bekommt dann als Partner den Computer zugewiesen. Dieser beschreibt die Bilder einfach mit Begriffen von anderen Spielern aus früheren Spielrunden. Der menschliche Partner bemerkt dadurch keinen Unterschied ob sein Partner ein Mensch ist oder durch den Computer gestellt wird. Auch hier können Übereinstimmungen erzielt werden um dadurch die Tabu-Wörter für die nächsten Runden zu erweitern. [von Ahn und Dabbish 2004]

Google Image Labeler

Google lizenzierte das ESP-Game und entwarf eine eigene Version namens „Google Image Labeler“²³ (Abbildung 18). Die Tabu-Wörter werden hier „off-limits“ genannt und zusätzlich wird die Anzahl der eingegebenen Begriffe des Partners angezeigt. Am Ende einer Spielrunde wird eine Zusammenfassung angezeigt, die alle Bilder nochmals enthält. Zu jedem Bild werden der Original-Link und die Tags des Partners, die dieser vergeben hat, dargestellt.

²³ <http://images.google.com/imagelabeler/> (03.02.2010)



Abbildung 18: Google Image Labeler

Ein anderes Spiel mit dem Namen „PopVideo“²⁴ ersucht den Spieler Tags auf Videos zu vergeben. Dabei kann man während des Videos alles beschreiben was man hört oder sieht. Dieses Spiel wird nicht zu zweit gespielt sondern zu fünft. Dadurch lassen sich leichter Übereinstimmungen mit einem anderen Spieler finden. Außerdem entsteht während der Spielrunde ein Wettkampf um die meisten Übereinstimmungen, da die Punkte direkt im Video eingeblendet werden.

2.2.5 Automatische Tagextraktion

Um aus Inhalten automatisch Tags zu generieren gibt es verschiedene Möglichkeiten. Die häufigste Methode, um aus einem Text Tags zu gewinnen, basiert auf der Häufigkeit der einzelnen Worte. Dahinter liegt die Annahme, je öfter ein bestimmter Term im Text vorkommt, umso wichtiger ist dieses Wort im Zusammenhang mit dem Text. Da zum Beispiel jedoch Artikel, Fürwörter und Verhältniswörter sehr häufig in einem Text vorkommen und diese als Tag nicht geeignet sind werden so genannte Stopplisten verwendet. Auf diesen Listen sind jene Worte vermerkt, die als Tags nicht verwendet werden sollen. Jedoch müssen für jede Sprache eigene Stopplisten angelegt werden, da die Wörter eins zu eins verglichen werden.

Diese Methode wird unter anderem von der Webseite Tagcrowd²⁵ verwendet. Dort wird aus dem Text von verschiedenen Eingabequellen (Webseite, Datei) eine Tag-Cloud generiert. Stopplisten werden anhand der gesetzten Sprache gewählt und es können auch eigene Stopplisten verwendet werden. Die in Abbildung 19 gezeigte Grafik ist eine Tag-Cloud, die mit Hilfe von Tagcrowd von der Webseite „Wired News“²⁶ erstellt wurde. Hier kann man

²⁴ <http://www.gwap.com> (03.02.2010)

²⁵ <http://www.tagcrowd.com> (04.03.2010)

²⁶ <http://www.wired.com> (04.03.2010)

auch eine Schwachstelle dieser Methode erkennen. Auf der Webseite „Wired News“ steht neben jedem Artikel wann dieser erstellt wurde (z.B. „10 hours ago“). Die verwendete englische Stoppliste enthält nicht das Wort „ago“ welches dann aber durch dessen Häufigkeit in der Tag-Cloud angezeigt wird, obwohl es als Tag nicht geeignet ist. Damit ist ein manueller Eingriff nötig um die Stoppliste zu erweitern und dadurch eignet sich diese Methode nur bedingt für die automatische Extraktion von Tags.

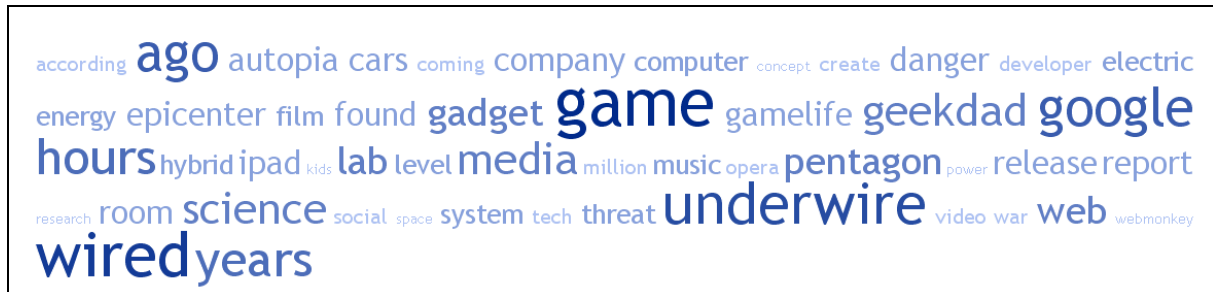


Abbildung 19: Tagcloud von Tagcrowd

Eine bessere und komplexere Möglichkeit Tags automatisiert zu gewinnen, ist die klassische TFIDF (term frequency – inverse document frequency) – Methode, welche auf Dokumente angewendet wird. Dabei handelt es sich um eine Gewichtungsmethode, die als Produkt der beiden Faktoren TF (term frequency) und IDF (inverse document frequency) hervorgeht. Deren erster Faktor TF bestimmt die Häufigkeit eines Wortes in einem Dokument und der zweite Faktor IDF gibt die Termspezifität an und basiert unter anderem auf die Anzahl der Dokumente, in der der Term vorkommt. Die Termspezifität gibt somit die Trennungsschärfe von Worten an, mit der sich Dokumente gut unterscheiden lassen. Diese Methode wurde schon 1972 von Karen Spärck Jones vorgestellt und ist bis heute im Information Retrieval im Einsatz. [Robertson 2004, Salton und Yang 1973]

$$tfidf = tf * idf$$

Formel 1: TFIDF

In der Formel 1 gibt der TF-Wert die Häufigkeit eines Wortes in einem Dokument an. IDF gibt hingegen einen Richtwert über die Nutzung dieses Wortes im Allgemeinen über alle betrachteten Dokumente an. [Brooks und Montanez 2006]

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Formel 2: Term frequency

Bei dem Ausdruck TF (Formel 2) stellt der Zähler die Anzahl der Vorkommnisse des Terms i in einem bestimmten Dokument j dar. Im Nenner wiederum steht die Summe der Häufigkeit aller Terme im selben Dokument j .

$$idf_i = \log_{10} \frac{|D|}{|(d_i \supset t_i)|}$$

Formel 3: Inverse document frequency

IDF (Formel 3) ist der Logarithmus aus der Anzahl aller betrachteten Dokumente gebrochen durch die Anzahl der Dokumente in denen der Term vorkommt. Wird ein Term gewählt, der in keinem der Dokumente aufscheint, so ergibt sich eine Division durch Null. Dies kann dadurch vermieden werden, dass entweder nur Terme verwendet werden, die in diesem Dokumente existieren oder der Nenner wird zum Beispiel um „+1“ erweitert.

$$tfidf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} * \log_{10} \frac{|D|}{|(d_i \supset t_i)|}$$

Formel 4: TFIDF Detailansicht

Wie beschrieben gibt TF die Häufigkeit eines Terms in einem Dokument an. So bekommen Artikel, Bindewörter, Fürwörter und einige weitere Wortarten einen hohen Wert für TF. Aber diese Wörter sind nicht geeignet um ein Dokument zu beschreiben. IDF verhindert, dass solche Wörter eine hohe Gewichtung bekommen, dadurch, dass, sollten diese Wörter auch in anderen Dokumenten vorkommen, der TF-Wert durch die Multiplikation mit der „inverse document frequency“ abgeschwächt wird. Umgekehrt funktioniert dies auch mit Wörtern, die in einem Dokument nur spärlich erscheinen und in anderen Dokumenten überhaupt nicht auftreten. Bei diesen Wörtern wird die „term frequency“ durch die Multiplikation mit der IDF aufgewertet. Dies führt dazu, dass gebräuchliche Wörter einen sehr kleinen TFIDF-Wert bekommen und selten benutzte Wörter einen hohen TFIDF-Wert [Brooks und Montanez 2006]. Dadurch eignet sich diese Methode auch dazu um maschinell passende Tags aus mehreren Dokumenten zu gewinnen.

2.2.6 Zusammenfassung

Die ersten vorgestellten Tagging-Systeme (Scuttle, Flickr und SparTag.us) unterstützen den Benutzer bei der Vergabe von Tags zu bestimmten Inhalten. Hingegen wird bei den ESP-Games der Spieler wiederum benutzt um Tags zu generieren. Gänzlich auf den Menschen verzichten die automatischen Tag-Extraktionsalgorithmen.

Die folgende Tabelle 5 gibt einen Überblick über die beschriebenen Tagging-Systeme, die den Benutzer unterstützen.

	Scuttle	Flickr	SparTag.us
Externes/Internes System:	extern	intern	extern
Type2Tag:	Ja	Ja	Ja
Click2Tag:	Populäre Tags	Alle selbst vergebenen Tags	Jedes Wort im Paragraphen; Vorhandene Tags zu dem Paragraphen
Tagbare Inhalte:	URLs	Bilder, Videos	Textparagraphen
Wer darf Inhalte taggen:	Jeder	Ersteller, Familie und Freunde bei Freigabe	Jeder
Art – Speicherung:	URL	intern	URL und die ganze Webseite selbst
Tagempfehlung:	Populäre Tags (über alle)	Nein	Vorhandene Tags zu dem Paragraphen
Autovervollständigung:	nein	nein	nein
Anmeldung erforderlich:	Ja	Ja	Ja
Tagsharing:	Ja	Ja	Ja
Bag/Set-Model:	Bag	Bag	Bag
Max. Anzahl Tags pro Inhalt:	-	75	-

Tabelle 5: Überblick über die unterstützten Tagging-Systeme

3 Wiki-Systeme

Im folgenden Kapitel werden Wiki-Systeme beschrieben. Zuerst wird ein Einblick in die geschichtliche Entstehung solcher Systeme gegeben bis hin zu heutigen Anwendungsgebieten. Es werden verschiedene Varianten vorgestellt, die auf unterschiedlichen Technologien basieren. Im Speziellen wird die Wiki-Version, die mit JavaServer Pages arbeitet, nämlich JSP-Wiki, näher aufgegriffen. Abschließend wird das Austria-Forum vorgestellt, welches auf dieser Version beruht. Auf die Frage „Warum Wikis funktionieren?“ wird nicht näher eingegangen, da das Hauptaugenmerk auf den technischen Aspekten liegt. Die Antwort auf diese Frage wird aber in [C2 2010c] und [Riehle 2006] gegeben.

3.1 Wiki-Systeme

In der Einleitung wurde bereits ein Wiki definiert: *„Ein Wiki ist eine webbasierte Software, die es allen Betrachtern einer Seite erlaubt, den Inhalt zu ändern, indem sie diese Seite online im Browser editieren. Damit ist das Wiki eine einfach und leicht zu bedienende Plattform für kooperatives Arbeiten an Texten und Hypertexten.“* [Ebersbach et al. 2008]

Grundsätzlich hat daher jeder die Möglichkeit, sein Wissen in einem Wiki mit anderen zu teilen. Es können Artikel neu verfasst, erweitert oder korrigiert werden. Dafür stellt das Wiki einfache Funktionen zur Verfügung um dies zu bewerkstelligen. Es ist auch keine zusätzliche Software nötig, da ein Wiki aus einer Client-Server Architektur besteht. Der Client wird im Browserfenster dargestellt und der Server führt die entsprechenden Aktionen aus. Weiters werden die Texte in einer Markup-Sprache verfasst, welche einfachen Regeln unterliegt. So übernimmt das Wiki größtenteils die visuelle Gestaltung des Artikels und der Autor kann sich auf den eigentlichen Inhalt konzentrieren. Dadurch wird auch eine einheitliche Visualisierung des Wikis erreicht. Mit Hilfe der einfachen Markup-Sprache werden in den Texten auch Verweise auf andere Artikel hergestellt und somit entstehen daraus Hypertexte.

Mit Wiki ist in dieser Arbeit im weiteren Sinne das Konzept dahinter gemeint – als eine spezielle Software. Eine Implementierung davon wird dann konkret als Wiki-System oder Wiki-Engine bezeichnet.

3.1.1 Historische Entwicklung

Das erste Wiki im eigentlichen Sinne wurde 1994 von Ward Cunningham unter dem Namen WikiWikiWeb entwickelt. Der Name stammte von einer Hawaiireise, die Cunningham unternahm. Dort hieß der Bus am Flughafen Wiki Wiki Bus, wobei Wiki „schnell“ bedeutet und die zweifache Verwendung eine Steigerung darstellt [Cunningham 2005]. Dies entspricht auch der Zielsetzung, die hinter dem Konzept von WikiWikiWeb steht. Es geht darum, Inhalte schnell und einfach zu publizieren und zu verändern. Cunningham wollte damals ein Dokumentationssystem schaffen, welches stärker auf die Bedürfnisse von Programmieren ausgelegt ist. Als Grundlage für seine Entwicklung diente ihm Apple's HyperCard Software [Leuf und Cunningham 2001].

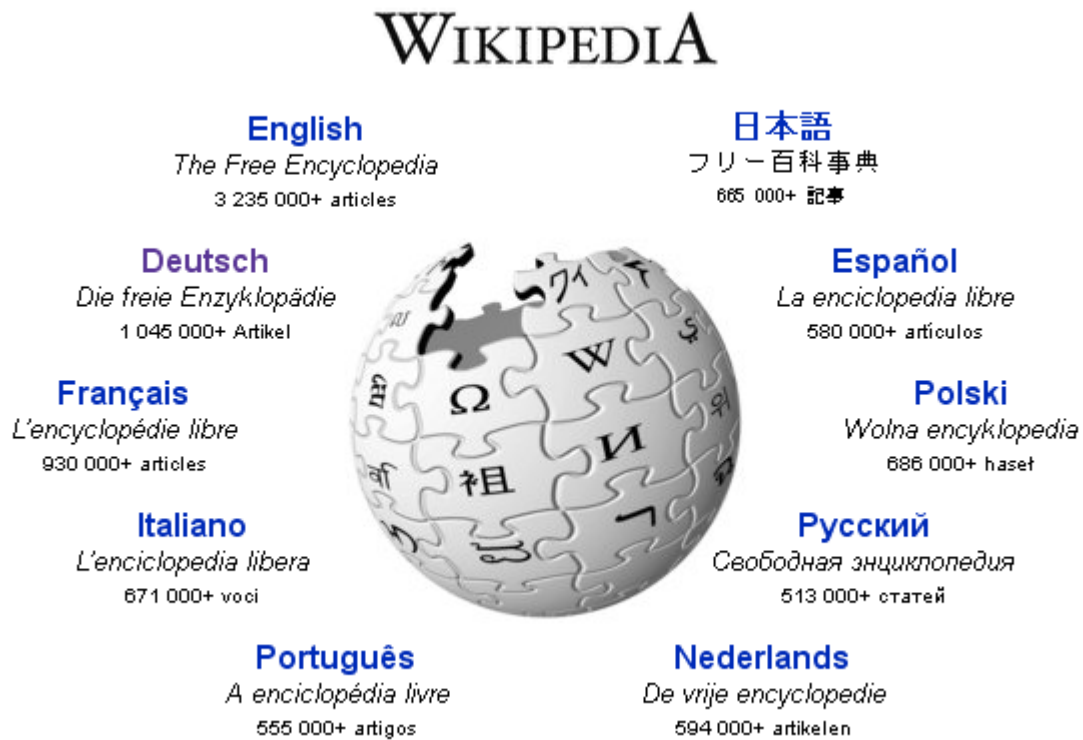
Seinem ersten Programm, ein „cgi“-Skript, welches er nach dem WikiWikiWeb-Konzept entwickelt hat, gab er den abgekürzten Namen „wiki“. Diese Abkürzung wurde auch von den Anwendern aufgegriffen und hat sich seitdem als Konzeptbezeichnung durchgesetzt. [Cunningham 2005]

Seit damals stieg die Anzahl der Zugriffe kontinuierlich und erreichte im Jahr 1997 schon 70.000 Zugriffe im Monat [C2 2010a]. Durch den Erfolg entstanden auch andere Wikis. Diese nutzten auch andere serverseitige Technologien als das Ursprungs-Wiki von Cunningham.

Wikipedia

Das Wiki-Konzept wurde erst im Jahr 2001 mit dem Start der Online-Enzyklopädie Wikipedia so richtig populär. Ursprünglich lief dieses Projekt unter dem Namen Nupedia und sollte laut dem Gründer Jimmy Wales zur größten frei verfügbaren Enzyklopädie der Welt werden. Die Artikel, welche von Freiwilligen geschrieben werden, sollen nach der Einreichung von qualifizierten Experten aus dem jeweiligen Fachgebiet geprüft werden. Dazu gehören unter anderem eine Faktenprüfung und die Kontrolle durch ein Lektorat. Trotz der Bereitschaft der Autoren Artikel zu verfassen scheiterte dieses Unternehmen an den Hürden der darauf folgenden Kontrollen und den damit verbundenen Schwierigkeiten. So entstanden in den ersten drei Jahren seit dem Bestehen von Nupedia nur 30 Artikel. Als Konsequenz daraus versuchte man am 15. Jänner 2001 das strenge System der Expertenprüfung mit Hilfe eines Wikis in eine offene von jedermann editierbare Enzyklopädie umzuwandeln und nannte diese ab sofort „Wikipedia“. Da nun die Kontrolle durch Experten wegfiel, war es einfach, einen neuen Artikel zu veröffentlichen oder einen bestehenden zu bearbeiten. Die Bereitschaft der Freiwilligen Artikel zu verfassen war immer noch vorhanden und einen Monat später erreichte Wikipedia schon eine Sammlung von 1.000 Artikeln und bis September desselben Jahres wurden schon 10.000 Artikeln verfasst. [Möller 2003]

Heute gilt Wikipedia als eines der bekanntesten Wikis. Es umfasst mittlerweile mehrere Millionen von Artikeln, welche in den verschiedensten Sprachen verfasst wurden. Ein Auszug über die am häufigsten verwendeten Sprachen und die Anzahl der Artikeln, die mit der jeweiligen Sprache geschrieben wurden, ist in Abbildung 20 zu sehen.

Abbildung 20: Wikipedia²⁷ - Artikelübersicht

3.1.2 Einsatzgebiete

Wikis werden in verschiedenen Bereichen eingesetzt. Zum Beispiel gibt es eine freie Online-Enzyklopädie namens „Wikipedia“²⁸, einen weltweiten Reiseführer „Wikitravel“²⁹ und ein freies Wörterbuch „Wiktionary“³⁰. Heutzutage gibt es nahezu zu jedem Thema ein entsprechendes Wiki. Es existiert sogar ein Wiki (Wikiindex³¹) über die verschiedenen verfügbaren Wikis. Dabei können die einzelnen Wikis auf unterschiedlichen Technologien basieren, welche in Kapitel 3.2 vorgestellt werden.

3.1.3 Kernfunktionen

Grundsätzlich verfügen alle Wiki-Systeme über bestimmte Kernfunktionen. Diese erleichtern den Umgang mit dem Wiki und dienen dem Benutzer dazu, Korrekturen und Erweiterungen einfach durchführen zu können. Ebersbach et al. [Ebersbach et al. 2008] zählen sechs wichtige Bereiche auf, die im Folgenden beschrieben werden:

²⁷ <http://wikipedia.org> (30.03.2010)

²⁸ <http://www.wikipedia.org> (30.03.2010)

²⁹ <http://www.wikitravel.org> (30.03.2010)

³⁰ <http://www.wiktionary.org> (30.03.2010)

³¹ <http://www.wikiindex.org> (30.03.2010)

- **Editing:** Jeder Nutzer hat die Möglichkeit eine Wiki-Seite zu editieren. Bei bestimmten Wiki-Systemen kann eine Anmeldung erforderlich sein und bei anderen wiederum kann eine Seite auch anonym bearbeitet werden. Oft lassen sich auch Bilddateien hochladen, welche in den Artikel eingebunden werden können.
- **Links:** In der Seite selbst kann auf andere Artikel verwiesen werden. Dabei wird der Titel des verwiesenen Artikels in Klammern oder als Binnenversalie³² geschrieben, um ihn als Link zu markieren. Je nach Wiki kommen auch hier unterschiedliche Methoden zum Einsatz. Wird auf einen Artikel verwiesen, der noch nicht existiert, so erscheint nach Öffnen des Links ein Bearbeitungsfenster um diesen Artikel zu erstellen.
- **History:** Dadurch werden Änderungen des Artikels von der ersten Version bis zur aktuellen Version nachvollziehbar. Jeder Artikel kann auf eine frühere Version zurückgestellt werden. Durch diesen so genannten „Rollback“ lassen sich mutwillige Löschungen von Artikeln wieder rückgängig machen. Moderne Wikis bieten zusätzlich noch eine Vergleichsfunktion: Diese Funktion zeigt die Unterschiede zwischen den verschiedenen Versionen eines Artikels an. Dafür werden beide Artikel nebeneinander positioniert, die Änderungen werden markiert und es lässt sich leicht erkennen welche Passagen hinzugefügt oder entfernt wurden. Durch diese Funktion entfällt der zeilenweise Vergleich zwischen zwei Versionen per Hand.
- **RecentChanges:** Diese Seite enthält einen Überblick über neu verfasste Artikel und über Seiten, die kürzlich geändert worden sind. Sie ist jedoch nicht editierbar, da sie automatisch generiert wird.
- **Sandbox:** Damit neue Anwender sich mit der Markup-Sprache des Wikis vertraut machen können, gibt es eine Sandbox („Sandkasten“). Dies ist eine leere Wiki-Seite, wo man erste Schritte mit der Markup-Sprache unternehmen kann ohne dabei auf eine schon existierende Seite zuzugreifen und deren Inhalt ungewollt zu ändern.
- **Suchfunktion:** Eine Volltext- und Titelsuche kann in den meisten Wikis durchgeführt werden. Damit lassen sich Artikel mit einem aussagekräftigen Titel leicht finden.

Weiters wird häufig eine Diskussionsmöglichkeit zu jedem Artikel angeboten um Änderungen vorzuschlagen und um sich mit anderen Autoren über den Inhalt zu unterhalten oder Meinungsverschiedenheiten auszutragen.

³² Wörter mit einem Großbuchstaben im Wortinneren

3.1.4 Benutzergruppen

Bei einem Wiki unterscheidet man die Benutzer je nachdem welche Rechte sie besitzen. Grundsätzlich wird ein Wiki als offenes System gestaltet. Das bedeutet, dass jeder Zugriff auf die Artikel hat und diese auch bearbeiten kann. Dies führt natürlich auch zu Problemen, die in Kapitel 3.1.6 näher erklärt werden. Aus diesem Grund können unterschiedliche Abstufungen bei den Benutzerrechten eingeführt werden.

Grob zusammen gefasst unterscheidet man drei Gruppen. Als erstes gibt es die Leser und Autoren, welche das Wiki benutzen und Artikel erstellen. Als zweite Gruppe sind die Wiki-Administratoren zu nennen, die erweiterte Rechte haben um Benutzer zu sperren und Seiten zu löschen. Ihre Hauptaufgabe liegt darin, das Wiki zu pflegen. Und als dritte Gruppe gibt es die Systemadministratoren, die dafür sorgen, dass das Wiki ordnungsgemäß funktioniert. Sie haben also direkten Zugriff auf die Server auf denen das Wiki läuft und sind im Gegensatz zu den Lesern, Autoren und Wiki-Administratoren nicht auf den Wiki-Client angewiesen. [Ebersbach et al. 2008]

Oft ist es nötig noch weitere Unterteilungen vorzunehmen, um Rechte einzelnen Personengruppen besser zuordnen zu können. Um dies zu verdeutlichen sind als Beispiel einige Benutzergruppen von über 20 Gruppen aus dem MediaWiki aufgeführt. [WikiMedia 2010]

- **Anonymer Benutzer:** Kann Seiten anlegen und bearbeiten
- **Angemeldeter Benutzer:** Kann zusätzlich Seiten verschieben und Dateien hochladen
- **Administrator:** Kann Seiten schützen und geschützte Seiten bearbeiten. Außerdem kann er Seiten löschen und wiederherstellen und hat die Möglichkeit Benutzer oder IP-Adressen zu sperren.
- **Bürokrat:** Kann anderen Benutzern Administrator-, Bot- und Bürokratenrechte vergeben und auch wieder entziehen
- **Bot:** Dient als Benutzer für automatische Skripte (z.B. Tippfehlerkorrekturen). Bei diesem Typ können die Einträge in den „Letzten Änderungen“ vermieden werden damit diese Seite nicht durch ein Skript geflutet wird.

Mit Hilfe von Erweiterungen können auch weitere Typen ins MediaWiki eingeführt werden um eine noch feinere Unterteilung zu erreichen.

3.1.5 Auszeichnungssprache

Um bestimmte Elemente in einem Dokument zu markieren kann eine Auszeichnungssprache (englisch: Markup Language) eingesetzt werden. Dabei wird der zu markierende Inhalt direkt im Dokument durch Tags eingeschlossen. Der Begriff Tag wird in diesem Sinne als Auszeichnungselement verwendet. Mithilfe dieser Tags lassen sich die markierten Stellen leicht finden und dadurch auch einfach maschinell verarbeiten. Eine der bekanntesten Auszeichnungssprachen ist HTML, das für die Erstellung von Webseiten verwendet wird. Dort werden Strukturbeschreibungen und Visualisierungstags gemeinsam in einem Dokument verwendet. Heutzutage wird jedoch eine Trennung zwischen dem Inhalt und der Darstellung bevorzugt um zum Beispiel die Visualisierung einfacher ändern zu können. [Bodendorf 2008]

Auch bei der Erstellung einer Wiki-Seite kommt eine so genannte Markup-Sprache zum Einsatz. Es handelt sich hierbei um eine Wiki-Syntax, die von Wiki zu Wiki unterschiedlich definiert wird. Sie unterliegt einfachen Regeln und soll dem Autor die Erstellung eines Artikels erleichtern. Dazu gehören einfache Textanpassungen sowie Definitionen über die korrekte Verwendung von Verweisen. In der Tabelle 6 ist ein Auszug von drei verschiedenen Wiki-Markup-Sprachen zu sehen mit dem daraus generierten HTML und der entsprechenden Darstellung im Browser. Die Visualisierung im Browser ist nur beispielhaft angeführt und kann je nach Wiki oder verwendetem Layout variieren. Von oben beginnend werden einfacher Text, Überschrift, fett geschriebener Text und unterstrichener Text dargestellt.

MediaWiki ³³	JSPWiki ³⁴	TWiki ³⁵	HTML	Darstellung im MediaWiki
Text	Text	Text	<p>Text</p>	Text
==Level==	! Level	----+ Level	<h1>Level</h1> >	<u>Level</u>
'''bold'''	__bold__	*bold*	bold	bold
<u>line</u>	%%(text-decoration:underline) line %%	<u>line</u>	<u>line</u>	<u>line</u>

Tabelle 6: Markup-Vergleich

Da der Großteil der Webseiten mit HTML aufgebaut ist und Wiki-Seiten im Endeffekt auch als HTML im Browser dargestellt werden stellt sich die Frage, warum nicht HTML direkt für den Aufbau einer Wiki-Seite verwendet wird. Dagegen sprechen einige wichtige Gründe, die in [C2 2010b] gelistet sind.

- Der Schwerpunkt liegt im Inhalt und nicht in der Visualisierung. Die Konzentration soll auf den eigentlichen Text gerichtet sein und nicht durch Gestaltung der Elemente gestört werden.
- Die Textformatierungsregeln sollen so einfach wie möglich sein. In HTML zum Beispiel kann das gleiche Ergebnis auf verschiedene Wege erreicht werden.

³³ <http://www.mediawiki.org> (14.04.2010)

³⁴ <http://www.jspwiki.org> (14.04.2010)

³⁵ <http://www.twiki.org> (14.04.2010)

- Das Einfügen von HTML-Tags bricht den Gedankenfluss ab, da diese beim Einfügen sehr zeitaufwendig sind.
- Der Wiki-Text soll auch in der Markup-Sprache leicht lesbar sein um ihn einfach bearbeiten zu können.
- Mit HTML kann jeder eine Wiki-Seite so aussehen lassen wie er es möchte. Mit unterschiedlichen Schriftgrößen und Farben geht dann das einheitliche Design der Seiten verloren.
- Durch HTML entstehen auch Sicherheitsrisiken, da in HTML unter anderem JavaScript eingebunden werden kann.
- Wiki-Markup ist wesentlich einfacher zu lernen als HTML weil es nur wenige Formatierungsmöglichkeiten gibt.

Natürlich gibt es auch einige Argumente, die für den Einsatz von HTML sprechen, jedoch überwiegen die Nachteile gegenüber einer eigenen Markup-Sprache für Wikis.

Ein Problem, das dadurch entsteht, ist, dass fast jede Wiki-Implementierung eine eigene Syntax entwickelt hat und diese untereinander nicht kompatibel sind. Die Webseite WikiMarkupStandard³⁶ listet die Syntaxen einiger Wiki-Systeme auf und beschreibt die Vor- und Nachteile der einzelnen Varianten. Zur Lösung dieses Problems würde eine einheitliche standardisierte Wiki-Syntax beitragen. Bestehende Wikis könnten diesen Standard dann auch für ihre Wiki-Engines nutzen. Dies würde auch die Entwicklung von Editoren und zusätzlichen Hilfswerkzeugen vorantreiben. [Sauer et al. 2006]

Seit Mitte 2007 gibt es den Creole 1.0³⁷ Standard, der eine standardisierte Syntax enthält. Mittlerweile ist dieser auch von einigen Wikis implementiert und die Nachfolgerversion Creole 2.0 ist schon in Entwicklung. Es werden die verschiedenen Wiki-Syntaxen nicht ersetzt sondern nur ergänzt. Und dadurch lassen sich Inhalte von verschiedenen Wikis auf andere nahtlos übertragen [Sauer et al. 2007].

Um den Autoren die Arbeit noch weiter zu erleichtern, besitzen viele Wikis auch eine Toolbar im Editor. Diese ermöglicht das einfache Einfügen von Markup in den Text. In Abbildung 21 ist die Toolbar von MediaWiki zu sehen. Mit Hilfe dieser können Bilder, externe Links und mathematische Formeln durch Knopfdruck hinzugefügt werden.



Abbildung 21: Editortoolbar von MediaWiki (Version 1.15.3)

³⁶ <http://meatballwiki.org/wiki/WikiMarkupStandard> (14.04.2010)

³⁷ <http://www.wikicreole.org> (14.04.2010)

3.1.6 Probleme

Bei der Verwebung einer neuen Technologie, in diesem Fall das Wiki-Konzept, können auch Umstände auftreten die zu Problemen führen. In diesem Unterkapitel wird gezielt auf die wichtigsten Problemstellungen eingegangen und Lösungen aufgezeigt.

Durch das offene Prinzip kann jeder Benutzer Einträge bearbeiten oder löschen. Der Missbrauch davon wird als Vandalismus bezeichnet. So können ganze Beiträge ohne Grund mutwillig gelöscht werden. Geschieht das, dann kommen die Schutzmechanismen eines Wikis zu tragen. Durch die verwendete Versionierung ist eine Rückführung der Seite auf ältere Versionen auf einfache Weise möglich. Je stärker ein Wiki frequentiert wird umso schneller fallen solche mutwilligen Zerstörungen auf und lassen sich durch andere Benutzer rückgängig machen.

Da Veränderungen von Seiten weit weniger auffallen als die Löschung derer, ist es hier schwieriger entsprechend schnell darauf zu reagieren. Im Jahr 2006 ist publik geworden, dass Wikipedia-Seiten, die Biographien von amerikanischen Politikern beinhalten, von Kongress-Rechnern aus manipuliert worden sind. Dabei wurden kindische und böartige Kommentare bei politischen Gegnern eingefügt und die Seiten der eigenen Politiker geschönt, indem unpassende Fakten entfernt worden sind. [Stöcker 2006]

Die Sperrung von registrierten Benutzern und IP-Adressen, von denen aus die Manipulationen stattfinden, führt nur kurzfristig zum Erfolg. Gesperrte Benutzer können ein neues Benutzerkonto anlegen und IP-Sperren können umgangen werden.

Deshalb hat man sich bei Wikipedia vom offenen Prinzip bei einigen Seiten abgewendet und führte einen Schreibschutz ein. So sind gewisse Seiten, die wichtige Informationen liefern, für unangemeldete und neue Benutzer schreibgeschützt. Darunter fallen zum Beispiel die Hauptseite und häufig benutzte Vorlagen. Eine Vollsperrung erlaubt es nur mehr Administratoren die Seite zu ändern. Auch wenn die Sperrung von Benutzern und IP-Adressen nicht zum gewünschten Erfolg führt und Seiten weiterhin Vandalismus ausgesetzt sind, können diese vor Veränderungen anhand des Schreibschutzes bewahrt werden. Ein weiterer Fall um Seiten mit einem Schreibschutz zu versehen tritt dann auf, wenn sich Autoren einen so genannten „Edit-War“ liefern. Dabei überschreiben zwei oder mehrere Benutzer abwechselnd die Änderungen des anderen Benutzers oder machen diese rückgängig. In Abbildung 22 ist ein Auszug aus solch einem Edit-War zu sehen. Meinungsverschiedenheiten sollen aber nicht im Beitrag ausgefochten werden sondern auf der dazugehörigen Diskussionsseite besprochen werden. Aus diesem Grund können Seiten auch kurzfristig gesperrt werden bis die Angelegenheit geklärt ist. [Wikipedia 2010]

- (cur | prev) ○ 19:38, 2 April 2009 189.75.33.106 (talk) (1,329 bytes) (*Undid revision 280792502 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:46, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:24, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280789461 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:23, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:15, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280788257 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:15, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:15, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280788001 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:13, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:13, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280787862 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:12, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:11, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280787637 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:11, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 03:09, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280786606 by Troy 07 (talk)*) (undo)
- (cur | prev) ○ 03:04, 31 March 2009 Troy 07 (talk | contribs) m (1,344 bytes) (*Reverted edits by 201.34.148.22 (talk) to last version by Troy 07*) (undo)
- (cur | prev) ○ 02:49, 31 March 2009 201.34.148.22 (talk) (1,329 bytes) (*Undid revision 280776444 by Troy 07 (talk)*) (undo)

Abbildung 22: Edit-War bei Wikipedia³⁸

Da Wikipedia weiterhin ein offenes System bleiben soll, wird die Möglichkeit einen Schreibschutz über eine Seite zu verhängen nur in wichtigen Fällen ausgeübt. Laut des Wikimedia Toolserver³⁹ sind nur 0,0198 Prozent aller Artikel der deutschen Wikipedia mit einer Vollsperrung belegt und 0,2397 Prozent aller Artikel können nur von länger registrierten Benutzern verändert werden.

Ein weiteres Problem kann bei verschiedenen Markup-Sprachen auftreten. Das erste Wiki, nämlich WikiWikiWeb, verwendete für interne Verweise auf andere Seiten Binnenversalien, auch CamelCase genannt. Jedes Wort also, das im Wortinneren einen Großbuchstaben besitzt, wurde von der Wiki-Engine als interner Link im Markup erkannt und entsprechend auf der Wiki-Seite dargestellt. Bei Seiten, deren Titel aus mehreren Wörtern besteht, werden deshalb die Zwischenräume einfach weggelassen. So wird aus dem Markup-Text CamelCase der Link zur Wikiseite CamelCase generiert.

Schwierig wurde es, wenn der Verweis nur aus einem Wort besteht. So konnte das Markup zu der Wiki-Seite „Computer“ entweder als „ComPuter“ oder auch „CompUter“ geschrieben werden. Bei den unterschiedlichen Schreibweisen würden notwendige Links nicht gesetzt und der Text wird auch nicht lesbarer. Deshalb schlug Clifford Adams, einer der Hauptautoren der UseMod-Software, die für Wikipedia benutzt wurde, eine neue Syntax vor um Links zu setzen. Bei der neuen Syntax werden die internen Verweise nicht mehr als CamelCase dargestellt sondern in doppelte eckige Klammern eingefasst. [Möller 2003]

Dieses „Free Links“ genannte Verfahren setzte sich auch bei vielen anderen Wikis durch und dadurch konnten die oben beschriebenen Probleme verhindert werden und ungewollte Verlinkungen, die bei Namen wie „McCullar“ entstehen, konnten gar nicht erst auftreten.

³⁸ <http://en.wikipedia.org/w/index.php?title=Portal:Pope/Intro&action=history> (06.05.2010)

³⁹ <http://toolserver.org/~avatar/sperrungen.php?wiki=de> (15.04.2010)

3.1.7 Wiki-Klone

Nach dem Erfolg des ersten Wikis, nämlich WikiWikiWeb, entstanden bis heute zahlreiche Wiki-Engines, die sich in verschiedenen Punkten von dem ursprünglichen Wiki unterscheiden. Auf der Webseite WikiMatrix⁴⁰ sind über 120 verschiedene Wiki-Engines gelistet, mit der Möglichkeit diese auf bestimmte Kriterien hin zu vergleichen um so das passende Wiki für seine Anforderungen zu finden. Auf der Webseite WikiIndex⁴¹ ist auch ein grober Entwicklungsbaum der Wiki-Engines zu sehen. Im Folgenden werden einige beliebte Wiki-Engines vorgestellt, die heute im Einsatz sind. Alle diese Wikis unterliegen einer Lizenz für freie Software und sind daher kostenlos verfügbar.

bitweaver

Diese Software bietet neben dem eigentlichen Wiki auch zahlreiche Features, die optional mitinstalliert werden können. Dazu gehören Blogs, eine Kalenderfunktion, Bildergalerien und Newsletters. Es gibt sogar ein Feature das Google Maps⁴² integriert. Auch durch weitere Erweiterungsmöglichkeiten ist bitweaver daher ein vollständiges Content Management System (CMS) und für umfangreiche Webseiten geeignet, die einen hohen Datenverkehr haben. Für die Datenspeicherung kann zwischen mehreren Datenbanken gewählt werden.

Homepage:	http://www.bitweaver.org	Version 2.7
------------------	---	-------------

DokuWiki

Hierbei handelt es sich um einfaches Wiki, das hauptsächlich zur Dokumentation gedacht ist. Zur Zielgruppe gehören Entwicklerteams, Arbeitsgruppen und kleinere Unternehmen. Die Engine ist in PHP geschrieben und verwendet ausschließlich einfache Textdateien zur Datenspeicherung.

Homepage:	http://www.dokuwiki.org	Version 2009-12-25
------------------	---	--------------------

JSPWiki

Diese Engine wurde in Java geschrieben und besitzt zahlreiche Features. In Kapitel 3.3 wird diese Engine ausführlich beschrieben.

Homepage:	http://www.jspwiki.org	Version 2.8.3
------------------	---	---------------

⁴⁰ <http://www.wikimatrix.org> (20.04.2010)

⁴¹ <http://www.wikiindex.org/WikiEngineTree> (20.04.2010)

⁴² <http://maps.google.com> (20.04.2010)

MediaWiki

Ursprünglich wurde diese Software speziell für die Wikipedia-Webseite geschrieben und ist deswegen genau auf deren Bedürfnisse ausgelegt. Die Syntax erinnert stark an die von UseMod⁴³, eine Wiki-Engine die zuerst für die Wikipedia-Webseite eingesetzt wurde. Mittlerweile wird MediaWiki auch von anderen Projekten der Wikimedia Foundation sowie für andere Webseiten genutzt. Die in PHP geschriebene Software unterstützt zahlreiche Sprachen und besitzt eine ausgeprägte Rechteverwaltung.

Homepage:	http://www.mediawiki.org	Version 1.15.3
------------------	---	----------------

MoinMoin

Bei MoinMoin handelt es sich um ein simples Wiki das ausschließlich einfache Textdateien zur Datenspeicherung nutzt. Es wurde in Python geschrieben und unterstützt keine Namensräume, welche die Verwendung von Wiki-Seiten mit gleichen Namen ermöglichen.

Homepage:	http://moinmo.in	Version 1.9.2
------------------	---	---------------

TWiki

Das TWiki unterliegt zwar der GPL, wird aber auch kommerziell vertrieben, da für den Support oder für das Hosting Kosten entstehen. Es wird vor allem von Unternehmen in deren Intranet eingesetzt, da es zu den ausgereiftesten Wiki-Klonen zählt. Namhafte Firmen wie Motorola, Nokia oder Disney zählen zu den Kunden. Zahlreiche Erweiterungen lassen dieses in Perl geschriebene Wiki zu einem umfangreichen Wissens- und Dokumentenmanagement-System wachsen.

Homepage:	http://twiki.org	Version 4.3.2
------------------	---	---------------

⁴³ <http://www.usemod.com> (20.04.2010)

3.2 Technik

Für die Benutzung eines Wikis ist für den Benutzer keine zusätzliche Software nötig. Um Artikel zu verfassen oder zu bearbeiten genügt ein gängiger Browser ohne zusätzliche Plugins. Wie in Abbildung 23 zu sehen ist, ist ein Wiki als Client-Server-Anwendung realisiert. Der Server bedient mehrere Clients, die über das Internet oder Intranet mit ihm verbunden sind.

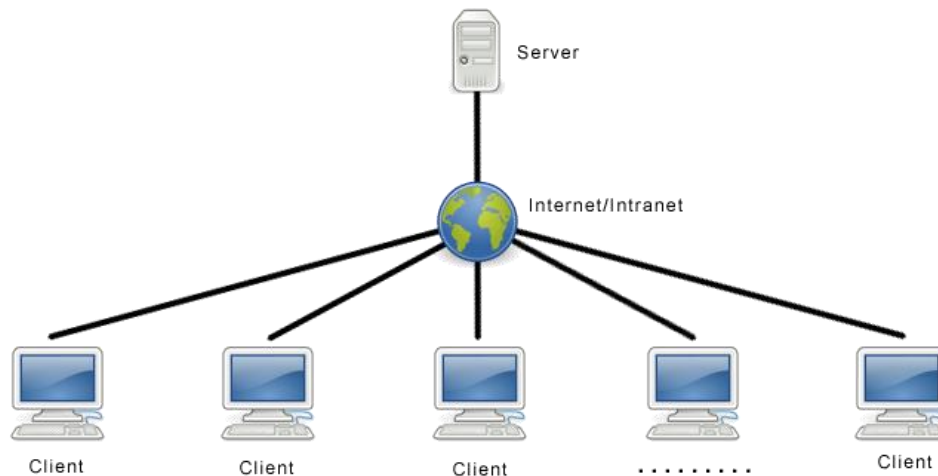


Abbildung 23: Client-Server-Architektur

Wird eine Wiki-Seite vom Client angefordert, so wird der entsprechende Markup-Code meistens aus einer Datenbank geladen, und ein Skript generiert am Server aus diesem eine Hypertext Markup Language (HTML)-Seite, die an den Client zurückgeschickt wird. Auf der Client-Seite braucht dann der HTML-Code nur mehr in einem Browser dargestellt werden (siehe Abbildung 24).

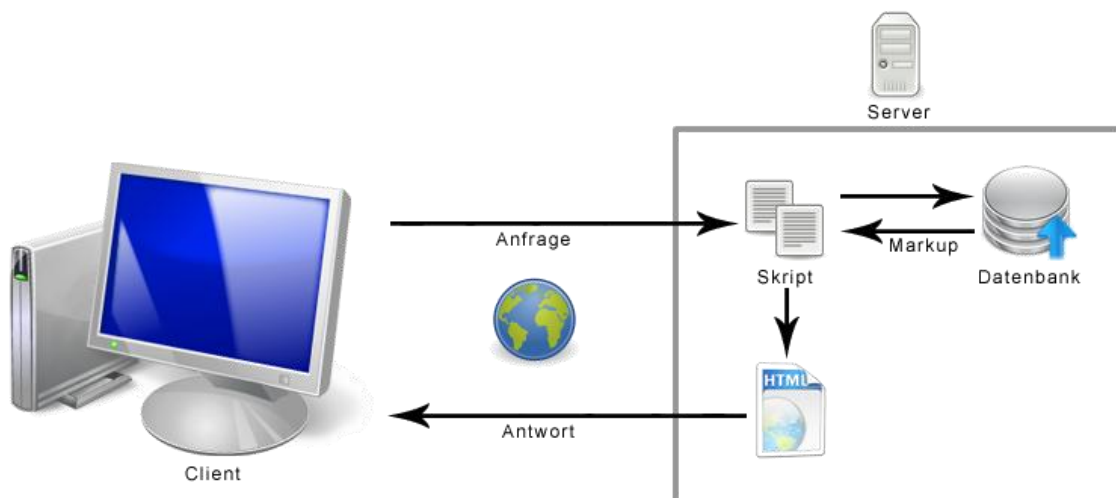


Abbildung 24: Wiki-Anfrage und Antwort

Soll die Seite bearbeitet werden, so wird der Wiki-Markup-Code an den Client geschickt, welcher dort im Browser verändert werden kann. Der veränderte Code wird anschließend zurück an den Server geschickt, von einem Skript verarbeitet und in einer Datenbank abgelegt (siehe Abbildung 25).

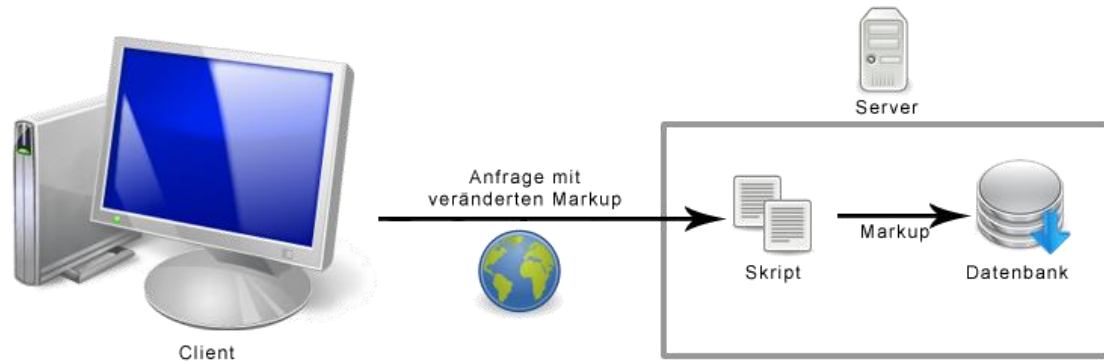


Abbildung 25: Wiki-Anfrage mit verändertem Markup

3.2.1 Webtechnologien

Anfragen werden vom Client an den Server geschickt, der diese bearbeitet und eine entsprechende Antwort zurückliefert. Die Kommunikation findet dabei über das Hyper Transfer Protocol (HTTP) [Fielding et al. 1999] mit den dazugehörigen Methoden statt. Zwei wichtige Methoden, die hier zum Einsatz kommen, sind:

- **GET-Methode:** Sie dient dazu, einen Datensatz vom Server abzurufen. Zusätzliche Daten, die für die Anfrage nötig sind, können als Parameter in der Uniform Resource Locator (URL) mit übergeben werden.
- **POST-Methode:** Diese Methode eignet sich, um größere Datenmengen – wie Text und Bilder – an den Server zu schicken, da die Daten nicht wie bei der GET-Methode innerhalb der URL mit übergeben werden, sondern gesondert im Körper der Nachricht.

Auf der Serverseite können aufbauend auf diesen Basismethoden verschiedene Webtechnologien zum Einsatz kommen, die den HTML-Code für die Clients generieren. Das erste Wiki von Cunningham war ein Unix-Programm und verwendete das Common Gateway Interface (CGI) [Cunningham 2005]. Das CGI definiert die Schnittstelle zwischen einem Webserver und einem externen Programm oder Skript und ermöglicht dadurch, dynamische Webseiten auf einfache Weise zu erstellen. Das externe Programm, beziehungsweise Skript, kann in verschiedenen Programmiersprachen, wie C/C++ oder PHP, geschrieben sein. Eine der beliebtesten Sprachen für CGI ist jedoch Perl. So kann nun ein Webserver mit der Verwendung von CGI neben statischen Inhalten auch dynamische Inhalte an den Client zurückliefern. Wird ein CGI-Skript angefragt, so führt der Webserver das CGI-Skript als externe Anwendung aus und liefert die Ausgaben dieses Skripts an den Client zurück als ob es eine statische Webseite wäre. Durch Umgebungsvariablen, die das CGI zur Verfügung stellt, können Informationen über den Client, sowie Parameter, die bei der GET-Methode in der URL mit übergeben wurden, ausgelesen werden. Ein Nachteil von CGI ist, dass für jede

Anfrage ein separater Prozess für das Skript gestartet wird, und dies führt zu Performanceproblemen. Es entwickelten sich auch andere Technologien, die dieses Manko umgehen, aber auch das gleiche Ziel haben, nämlich Client-Anfragen mit dynamischen Webseiten beantworten zu können. [Guelich et al. 2000, Fraley 2004]

Im folgenden Abschnitt werden die wichtigsten Webtechnologien angeführt, die heute für die Bereitstellung von dynamischen Webseiten von Wikis verwendet werden.

FastCGI

FastCGI ist im Grunde genommen das normale CGI mit einigen Erweiterungen. Wie CGI ist es programmiersprachenunabhängig, führt externe Skripte in eigenen Prozessen aus und schützt dadurch den Serverkern vor fehlerhafter Software. Das Performanceproblem von CGI, das durch startende Prozesse bei jeder Anfrage entsteht, wird auch umgangen. Nun ist ein Prozess nicht mehr für eine einzelne Anfrage zuständig und wird dann beendet, sondern er bleibt bestehen und verarbeitet weitere Anfragen. Ein weiterer großer Unterschied ist die Verwendung einer Vollduplexkommunikation für das FastCGI-Protokoll. Dadurch können FastCGI-Programme auf externen Maschinen laufen und mit einer Transmission Control Protocol (TCP)-Verbindung mit dem Webserver verbunden werden. [Open Market 1996]

Hypertext Preprocessor (PHP)

Hierbei handelt es sich um eine Skriptsprache, die in erster Linie für die Erstellung dynamischer Webseiten verwendet wird. Des Weiteren unterstützt PHP das Generieren von Extensible Markup Language (XML)-Dokumenten, Bildern in verschiedenen Formaten und Portable Document Format (PDF)-Dateien. Auch eine große Anzahl von Datenbanken wird unterstützt. Darunter befinden sich MySQL, PostgreSQL und Oracle. [Lerdorf et al. 2006]

Java Servlets und JavaServer Pages (JSP)

Ein Servlet ist eine Java-Klasse, welche dynamisch geladen werden kann um den Server mit Funktionen zu erweitern. Es läuft innerhalb der Java Virtual Machine (JVM) auf dem Server und ist daher auch sicher und portabel. Anders als bei CGI und FastCGI werden die Anfragen nicht von mehreren Prozessen verarbeitet, sondern es wird ein Prozess mit mehreren Threads verwendet. Mehrere Prozesse kommen zum Einsatz, wenn die Anfragen auf mehrere Server aufgeteilt werden. Dadurch lässt sich diese Konfiguration auch sehr gut nach oben skalieren um höheren Anforderungen gerecht zu werden. [Hunter 2001]

JSP sind sehr eng mit Java Servlets verbunden, da JSP-Seiten in Servlets übersetzt werden. Eine sehr vereinfachte Sicht um den Unterschied zwischen Java Servlets und JSP darzustellen, ist die, dass Servlets Java-Programme mit eingebetteten HTML sind, und JSP sind HTML-Seiten mit eingebettetem Java-Code. [Hall und Brown 2003]

Dadurch wird eine höhere Trennung zwischen der Darstellung und den Daten geschaffen [Bergsten 2003]. Im Detail wird diese Technologie in Kapitel 3.3 beschrieben.

Active Server Pages .NET(ASP.NET)

ASP.NET ist ein Bestandteil des Microsoft .NET Frameworks und dadurch wird der Zugriff auf zahlreiche schon verfügbare Funktionen und Klassen ermöglicht. Der Code kann in verschiedenen unterstützten Sprachen geschrieben werden, da dieser in eine Universalsprache namens „Common Intermediate Language (CIL)“ übersetzt wird. Zur Webseitenerstellung werden nun nicht mehr Skripte interpretiert, sondern es wird kompilierter Code verwendet, der durch die Common Language Runtime (CLR) ausgeführt wird. [Mitchell et al. 2001]

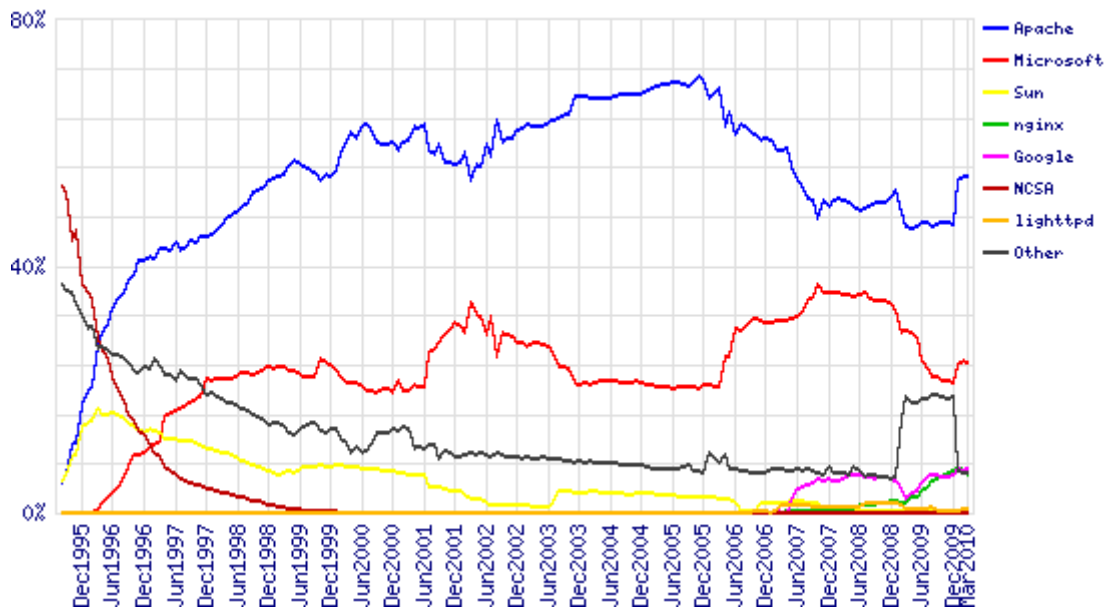
3.2.2 Webserver

Webserver erstellen und übertragen die dynamischen Seiten mit den vorher genannten Technologien an den Client. So wird CGI von allen gängigen Webservern unterstützt. Da CGI nur eine Schnittstelle zwischen Webserver und externen Programmen (bzw. Skripten) darstellt, ist es möglich, PHP über CGI an einen Webserver anzubinden. Diese Variante ist zwar nicht so performant wie ein Webserver, der direkt PHP unterstützt, dafür kann ein fehlerhafter PHP-Code aufgrund der externen Ausführung nicht den Serverkern beeinflussen. Dasselbe gilt für Java-Programme, die über CGI dynamische Webseiten erstellen. Weiters können Anwendungen, die nur über CGI mit dem Webserver kommunizieren, einige Fähigkeiten des Servers nicht in Anspruch nehmen sobald sie gestartet sind, da sie in einem eigenen Prozess ablaufen. Dazu gehört unter anderem die Möglichkeit Logdateien des Servers zu beschreiben. [Hunter 2001]

Wie in Abbildung 26 zu sehen ist, sind die zurzeit am häufigsten verwendeten Webserver der Apache HTTP Server⁴⁴ (Version 2.2.5) und die Dienstplattform Internet Information Services⁴⁵ (IIS) von Microsoft (Version 7.5), die auch als Webserver fungiert. Beide haben eine CGI und FastCGI-Schnittstelle. Unterstützungen für PHP und ASP.NET sind entweder schon integriert oder können durch Module hinzugefügt werden. Servlets und JSP lassen sich mit einem Webcontainer realisieren, da eine Java-Engine benötigt wird.

⁴⁴ <http://httpd.apache.org> (12.04.2010)

⁴⁵ <http://www.iis.net> (12.04.2010)

Abbildung 26: March 2010 Web Server Survey⁴⁶

3.2.3 Datenspeicher

Ein Wiki kann unterschiedliche Arten von Datenspeicher einsetzen, um Seiten zu speichern. Dazu gibt es verschiedene Möglichkeiten:

Einfache Textdatei:

Eine einfache Lösung, um eine Wiki-Seite abzuspeichern, ist, sie als Dokument in einen Ordner zu geben. Dies ist ein nahe liegender Weg, weil die Wiki-Seiten selbst nur aus Text bestehen. Jede Wiki-Seite stellt dann ein Dokument im Ordner dar und lässt sich dadurch auch leicht bearbeiten, da nur ein Text-Editor nötig ist. Um Datensicherungen durchzuführen müssen nur die gewünschten Dokumente kopiert werden. Im Allgemeinen sind einfache Textdateien nicht so performant und leicht skalierbar wie dafür ausgelegte Datenbanken. Außerdem muss der Webserver dem Skript auch Schreiboperationen auf Dateien erlauben, damit Textdateien zur Datenspeicherung eingesetzt werden können. Weiters ist die Sicherheit, Integrität und Konsistenz der Daten nicht gewährleistet.

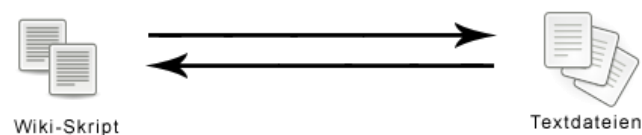


Abbildung 27: Datenspeicherung mit Textdateien

⁴⁶ http://news.netcraft.com/archives/web_server_survey.html (06.04.2010)

Datenbank:

Wenn von einer Datenbank gesprochen wird, ist grundsätzlich zumeist das Datenbanksystem gemeint. Es besteht aus dem Datenbankmanagementsystem (DBMS), welches den Zugriff auf die Datenbank steuert und Kontroll- und Überwachungsaufgaben übernimmt, und der Datenbank selbst, die die eigentlichen Daten beinhaltet [Kofler 2005]. Der Einfachheit halber ist daher außerhalb dieses Kapitels der Begriff Datenbank mit dem Begriff Datenbanksystem gleichbedeutend.

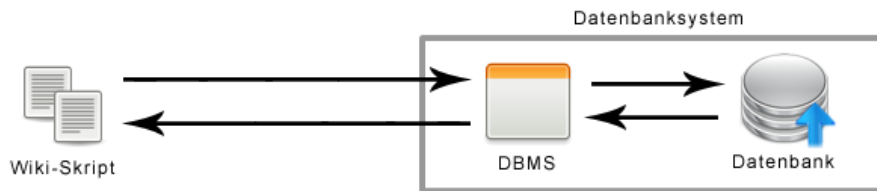


Abbildung 28: Datenspeicherung mit Datenbanken

Dadurch, dass der Zugriff nur über das DBMS auf die Datenbank möglich ist, übernimmt dieses die Aufgabe, die Integrität und Konsistenz der Daten sicherzustellen. Das DBMS kann automatisch Backups durchführen und sorgt anhand von Zugriffsrechten auch für die Sicherheit der Daten. Hauptsächlich kommen für Wikis relationale Datenbanken zum Einsatz. Hierbei handelt es sich um ein Datenmodell, in dem die Daten zeilenweise in Tabellen gespeichert werden und untereinander verknüpft sein können. Zu den wichtigsten eingesetzten Datenbanksystemen für Wikis zählen MySQL⁴⁷, PostgreSQL⁴⁸, Oracle⁴⁹ und SQLite⁵⁰. Um mit den genannten Systemen zu kommunizieren kann SQL verwendet werden.

SQL ist eine Datenbanksprache und wird von ANSI- und ISO-Standards definiert. Dennoch ist sie nicht standardisiert und wird deshalb auch von unterschiedlichen Plattformen unterschiedlich implementiert. Sie ermöglicht verschiedene Datenmanipulationen und dient zur Transaktionssteuerung. [Gennick 2004]

⁴⁷ <http://www.mysql.com> (13.04.2010)

⁴⁸ <http://www.postgresql.org> (13.04.2010)

⁴⁹ <http://www.oracle.com> (13.04.2010)

⁵⁰ <http://www.sqlite.org> (13.04.2010)

Versionsverwaltungssystem:

Eine weitere Möglichkeit Wiki-Seiten zu speichern besteht in der Verwendung eines Versionsverwaltungssystems. Dies ist eine Software, die Änderungen an Dateien erfasst, und dadurch lassen sich Bearbeitungsschritte an Wiki-Seiten zurückverfolgen. Einige Systeme, wie das Revision Control System⁵¹ (RCS), speichern die Änderungen in der Datei selbst ab und andere besitzen ein zentrales Projektarchiv das alle Änderungen beinhaltet. Dazu gehört auch das beliebte Subversion⁵² (SVN). Je nach dem, um welches System es sich handelt, kann für die Speicherung der Änderungen im Hintergrund ein Datenbanksystem oder direkt das Dateisystem verwendet werden.

Mit Hilfe eines Versionsverwaltungssystems ist es einem Wiki möglich, die Versionierung in ein solches System auszulagern. Änderungen an einer Wiki-Seite können dann auch direkt dort durchgeführt werden und müssen nicht über das Wiki-Benutzerinterface erfolgen.

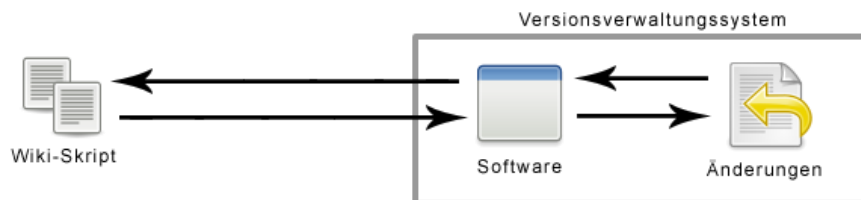


Abbildung 29: Datenspeicherung mit Versionsverwaltungssystemen

⁵¹ <http://www.gnu.org/software/rcs> (13.04.2010)

⁵² <http://subversion.apache.org> (13.04.2010)

3.3 JSPWiki

Diese Wiki-Implementierung wird in diesem Kapitel im Detail beschrieben, da sie für das Austria-Forum (siehe Kapitel 3.4) eingesetzt wird. Wie der Name schon sagt handelt es sich bei diesem System um ein Wiki, welches JSP verwendet. Die gesamte Implementierung wurde mit den Komponenten der Java Platform Enterprise Edition (Java EE) realisiert und ist zum Zeitpunkt dieser Arbeit in Version 2.8.3 verfügbar.

3.3.1 Java Servlets und JSP

Durch den Einsatz von JSP wird eine Trennung zwischen der Darstellung der Daten und den eigentlichen Daten erreicht. JSP sind im Grunde genommen HTML-Seiten, die einen dynamischen Java-Code enthalten. Diese werden dann anschließend in Servlets übersetzt, kompiliert und bei Anforderung ausgeführt. Obwohl JSP-Seiten im weiteren Sinne Servlets sind, wird dennoch durch ihre Verwendung die Visualisierung von der Programmlogik getrennt. Trotzdem gibt es Unterschiede wann man JSP oder Servlets einsetzen soll. Eine grobe Faustregel besagt, dass Servlets für Auftragsabwicklungen besser geeignet sind, hingegen liegt der Vorteil von JSP bei der Auftragsdarstellung. Häufig ist es auch nützlich oder sogar nötig beide Technologien gemeinsam zu verwenden. [Hall und Brown 2003]

In den nachfolgend angeführten Abbildungen (Quellcode 1 und Quellcode 2) wird jeweils ein HTML 4.0 Dokument erzeugt, welches „Hello World“ ausgibt. Einmal ist es als Servlet geschrieben und das andere Mal wird JSP verwendet.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
            \"Transitional//EN\">\n" +
            "<HTML>\n" +
            "<HEAD>\n" +
            "<TITLE>Example</TITLE>\n" +
            "</HEAD>\n" +
            "<BODY>\n" +
            "Hello World\n" +
            "</BODY>\n" +
            "</HTML>");
    }
}
```

Quellcode 1: Servlet "Hello World" Beispiel

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Example</TITLE>
  </HEAD>
  <BODY>
    <% out.println("Hello World") %>
  </BODY>
</HTML>

```

Quellcode 2: JSP "Hello World" Beispiel

Die Aufgaben eines Servlets bestehen im Wesentlichen darin, die Daten aus einer Anfrage vom Client zu lesen, diese zu bearbeiten und die Antwort wieder an den Client zu schicken. Die Daten bestehen dabei aus den expliziten Daten, die den Inhalt der Nachricht darstellen, und den impliziten Daten, die aus dem Kopfteil der Nachricht stammen. [Hall 2010]

Es gibt wesentliche Vorteile von Servlets und dadurch auch von JSP gegenüber dem traditionellen CGI, die von Hall und Brown [Hall und Brown 2003] aufgeführt sind.

- Hohe Effizienz:** Eine Anfrage wird im Servlet von einem Thread bearbeitet und nicht in einem eigenen ressourcenhungrigen Prozess. Bei mehreren gleichen Anfragen existiert auch nur eine Kopie der Servlet-Klasse im Speicher des Servers.
- Komfortabel:** Es werden schon etliche Funktionen für das Parsen und Verarbeiten von HTTP-Anfragen bereitgestellt. Diese müssen deshalb nicht selbst neu programmiert werden. Java wird als Programmiersprache verwendet und es muss keine neue Syntax erlernt werden.
- Leistungsstark:** Servlets können direkt mit dem Webserver kommunizieren, auch ohne eine spezielles Application Programming Interface (API). Außerdem können Daten zwischen den Servlets geteilt werden, um gemeinsam auf Ressourcen zugreifen zu können.
- Portabel:** Servlets werden von den gängigsten Webservern direkt oder durch ein Plugin unterstützt. So kann ein Servlet, das für einen Apache Tomcat⁵³ geschrieben wurde, auch auf einem IBM WebSphere Server⁵⁴ ausgeführt werden.
- Preiswert:** Es gibt eine Menge an kostenfreier und kostengünstiger Serversoftware. Projekte können mit einer freien Software realisiert werden. Sollte das Projekt ein Erfolg sein und der Bedarf einer leistungsfähigeren und leichter administrierbaren Umgebung aufkommen, dann kann auf eine performantere kostenpflichtige Systemsoftware gewechselt werden, ohne die Servlets neu schreiben zu müssen.

⁵³ <http://tomcat.apache.org> (21.04.2010)

⁵⁴ <http://www.ibm.com/software/de/websphere> (21.04.2010)

- Sicher:** Bei CGI werden die Skripte und Programme durch die Betriebssystem-Shell gestartet und dies stellt eine große Sicherheitslücke dar. Auch Pufferüberläufe können auftreten, da CGI Programmiersprachen unterstützt, die nicht automatisch die Grenzen von Speicherbereichen überwachen.
- Mainstream:** Servlets und JSP werden von zahlreichen Servern unterstützt und sind die häufigste eingesetzte Technologie bei mittleren bis großen Webapplikationen. Google listet über 500 Millionen Webseiten, die JSP verwenden. [Hall 2010]

3.3.2 Funktionsumfang

Bei JSPWiki (Version 2.8.3) handelt es sich um eine ausgereifte Software, die seit ihrer ersten Version zahlreiche neue Funktionen erhalten hat. Erweiterungen können einfach über Plugins eingebunden werden, und von denen existiert eine umfangreiche Sammlung. Diese können unter anderem wesentliche Bestandteile der eigentlichen Software sein und dadurch grundlegende Aufgaben übernehmen.

Ein Auszug der Features, die aus [JSPWiki 2008] und [WikiMatrix 2009] entnommen wurden, ist hier gelistet:

- Umfangreiches Wiki-Markup (siehe Kapitel 3.3.3)
- Hochgeladene Dateien können an die Wiki-Seite angehängt werden
- Einsatz von unterschiedlichen Datenspeichern (siehe Kapitel 3.3.4)
- Templates werden unterstützt. Dadurch kann das Design einfach geändert werden.
- UTF-8-Unterstützung (Unicode)
- Seitensperrung zum Schutz vor Vandalismus
- Einfaches Plugin- und Filtersystem für Erweiterungen
- Einfache Installation (siehe Kapitel 3.3.5)
- Blogs und Foren können erstellt werden
- Mehrere Sprachen werden unterstützt. Es können auch welche verwendet werden, die von rechts nach links geschrieben werden.

Über das Plugin-System können unter anderem auch Video- und Flash-Dateien in die Wiki-Seite integriert werden. Weiters können Bildergalerien, mathematische Formeln und eine Kalenderfunktion hinzugefügt werden.

Plugins werden im Markup aufgerufen und ersetzen ihren Eintrag im Markup mit deren generiertem Ausgabe-Code. Filter hingegen dienen dazu, den ganzen Markup oder HTML-Code vor dem Versand an den Client durchzugehen und Manipulationen auszuführen. Auch vor dem Abspeichern im Datenspeicher kann das Markup manipuliert werden. Zusätzlich können Filter auch nach dem Abspeichern aufgerufen werden, wobei das Markup nicht mehr geändert werden kann. Dieser Aufruf kann zum Beispiel für E-Mail-Benachrichtigungen dienen. [JSPWiki 2010a]

3.3.3 Auszeichnungssprache

Das JSPWiki verwendet eine Markup-Sprache, die sich in einigen Punkten wesentlich von anderen Sprachen wie die von MediaWiki unterscheidet. Mit Hilfe von Plugins und Filtern hat man Zugriff auf den Quellcode der Wiki-Seiten und kann Manipulation am Markup ausführen bevor daraus eine HTML-Seite generiert wird. Dadurch kann durch Verwendung des Creole Filters⁵⁵ der Creole 1.0 Standard unterstützt werden. In der Tabelle 7 sind die grundlegenden Textformatierungsmöglichkeiten von der JSPWiki-Markup-Sprache aufgelistet.

Wiki-Markup	Aktion
----	Erstellt eine horizontale Linie. Zusätzliche '-' werden ignoriert
\\	Erzwingt einen Zeilenumbruch
[link]	Erstellt einen Hyperlink auf eine interne Wiki-Seite mit dem Namen 'Link'
[this is also a link]	Erstellt einen Hyperlink auf eine interne Wiki-Seite mit dem Namen 'ThisIsAlsoALink', wobei der Link mit Leerzeichen angezeigt wird
[a sample link]	Erstellt einen Hyperlink auf eine interne Wiki-Seite mit dem Namen 'Link', wobei der Link mit 'a sample' dargestellt wird, anstatt 'Link'
~NoLink	Verhindert, dass ein Link aus dem CamelCase-Wort entsteht
[1]	Erstellt einen Verweis auf die Fußnote Nummer 1
[#1]	Markiert die Fußnote Nummer 1
[[link]	Erstellt den Text '[link]'
!heading	Kleine Überschrift mit dem Text 'heading'
!!heading	Mittlere Überschrift mit dem Text 'heading'
!!!heading	Große Überschrift mit dem Text 'heading'
''text''	Stellt den Text 'text' kursiv dar
__text__	Stellt den Text 'text' in Fettschrift dar
{{text}}	Stellt den Text 'text' in einer Festbreitenschrift dar
[text]	Stellt den Text 'text' unterstrichen dar
* text	Erstellt einen Aufzählungspunkt mit dem Text 'text'
# text	Erstellt einen nummerierten Aufzählungspunkt mit dem Text 'text'
=;term:ex	Erstellt eine Definition für 'term' mit der Erklärung 'ex'

Tabelle 7: Markup - Textformatierungen⁵⁶

Weiters existieren noch Markup-Regeln für das Einfügen von Bildern, Tabellen und Dateianhängen, ohne zusätzliche Filter verwenden zu müssen. Wie ein Plugin ins Markup einzufügen ist wird im Quellcode 3 veranschaulicht. Parameter sind optional und daher kann auch das 'WHERE' entfallen. Seit Version 1.9.3 ist auch das Tag 'INSERT' nicht zwingend notwendig und daher erfolgt der kürzeste Pluginaufruf wie im Quellcode 4 beschrieben. [JSPWiki 2010a]

⁵⁵ <http://www.jspwiki.org/wiki/WikiCreole> (28.04.2010)

⁵⁶ <http://www.jspwiki.org/wiki/TextFormattingRules> (28.04.2010)

```
[{INSERT <plugin class> WHERE <param1=value1>,<param2=value2>,...}]
```

Quellcode 3: Markup für Plugin

```
[{PluginName}]
```

Quellcode 4: Markup für Pluginaufruf

Eine Besonderheit stellt die Möglichkeit dar, Cascading Style Sheets (CSS) für Textformatierungen verwenden zu können. Der zu verändernde Text wird in doppelte Prozentzeichen eingefasst und zu Beginn wird das zu verwendende CSS in Klammern definiert. Es können auch mehrere Styles angewendet werden, wenn sie durch einen Strichpunkt getrennt werden. So können direkt im Markup CSS-Style-Regeln auf den Text angewendet werden um die Darstellung zu ändern, wie die Beispiele es in Tabelle 8 verdeutlichen. Daher ist es möglich, einige Textformatierungen, die als Markup in Tabelle 7 gelistet wurden, auch mit CSS zu realisieren.

Wiki-Markup	Darstellung
Normaler Text	Normaler Text
%%(color: red) Roter Text %%	Roter Text
%%(text-decoration: underline) Text %%	<u>Text</u>
%%(font-style: italic) Text %%	<i>Text</i>

Tabelle 8: CSS im Wiki-Markup

3.3.4 Datenspeicherung

Bei der Datenspeicherung im JSPWiki kommt der so genannte PageProvider zum Einsatz. Dieser stellt ein Interface zur Verfügung, mit dem Speicher- und Ladeoperationen ausgeführt werden können. Durch dieses Interface ist es möglich, jede beliebige Speicherform für Wiki-Seiten dahinter zu implementieren. In der Grundversion enthält JSPWiki drei verschiedene PageProviders.

- **FileSystemProvider:** Speichert die Wiki-Seiten in simplen Textdateien ohne Versionsarchivierung
- **VersioningFileProvider:** Wiki-Seiten werden in simplen Textdateien geschrieben und ältere Versionen werden in eigenen Ordnern abgelegt und sind somit wieder aufrufbar
- **RCSFileProvider:** Ein RCS (siehe Versionsverwaltungssystem Kapitel 3.2.3) wird für die Speicherung eingesetzt. Jedoch sollte stattdessen der VersioningFileProvider eingesetzt werden, da das RCS unter Last nicht stabil läuft.

Es stehen auch einige zusätzliche PageProviders auf der Entwicklerhomepage⁵⁷ zum Download zur Verfügung, die von Anwendern beigesteuert wurden. Darunter befinden sich auch folgende PageProviders:

- **AntCvsProvider:** Erlaubt den Zugriff auf ein CVS-Archiv mit Hilfe von ANT⁵⁸-Befehlen.
- **FedoraProvider:** Benutzt die Software Fedora⁵⁹ um Wiki-Seiten und deren Anhänge in einem digitalen Archiv zu speichern.
- **JDBCProvider:** Verwendet den Java Database Connectivity (JDBC) Standard um verschiedene relationale Datenbanken zur Speicherung zu nutzen
- **SubversionProvider:** Synchronisiert das Seiten-Verzeichnis mit einem Subversion Archiv.

Somit werden alle drei Speicherformen, die in Kapitel 3.2.3 aufgezählt wurden, entweder direkt oder durch Hinzufügen eines neuen Providers von JSPWiki unterstützt.

3.3.5 Installation

Da JSPWiki keine hohen Anforderungen an die Serverhardware (siehe Tabelle 9) stellt und Betriebssystem und Webserver auch kostenlos verfügbar sind, kann eine sehr preisgünstige Lösung für den Wiki-Betrieb realisiert werden. Es sind jedoch Kenntnisse für die Konfiguration des Webserver und den Umgang mit WAR-Dateien (Webarchive) notwendig.

System	Anforderungen
Serverhardware:	Ab Pentium II 266 MHz und 192 MBytes Arbeitsspeicher
Betriebssystem:	Jede Plattform, die das Java Development Kit (JDK) 1.5+ unterstützt
Webserver:	Jeder Server, der Servlets 2.4+ unterstützt

Tabelle 9: Systemanforderungen von JSPWiki [JSPWiki 2010b]

⁵⁷ <http://www.jspwiki.org/wiki/ContributedProviders> (29.04.2010)

⁵⁸ <http://ant.apache.org> (29.04.2010)

⁵⁹ <http://www.fedora-commons.org> (29.04.2010)

Für JSPWiki existieren drei Versionen, die herunter geladen werden können.

Stable: Diese Version ist getestet und für den produktiven Betrieb geeignet

Beta: An dieser Version wird gerade gearbeitet

Latest: Diese Version ist das letzte Release mit den aktuellsten Änderungen. Sie ist aber noch nicht ausführlich getestet und kann Serverprobleme verursachen.

Für den produktiven Einsatz sollte die Stable-Version gewählt werden, außer man entwickelt selbst Erweiterungen und möchte dafür die aktuellste Version haben. Nach dem Herunterladen einer Version wird die „JSPWiki.war“-Datei, das Webarchiv, in den Servlet-Container des Webservers eingefügt. Erst danach kann man auf <http://myhost/JSPWiki/Install.jsp> zugreifen und den Installationsanweisungen folgen. Dort werden einige wichtige Grundeinstellungen, wie die Angabe wo Wiki-Seiten und Log-Dateien abgelegt werden sollen, gesetzt. Für alle anderen Einstellungen werden Default-Werte verwendet, die später noch geändert werden können.

Es existiert auch eine Installationsvariante für Windows für unerfahrene Benutzer. Dafür sind keine Java-Server-Kenntnisse nötig. Hierbei handelt es sich um eine vereinfachte Version von JSPWiki, die für jene Benutzer ausgerichtet ist, die nur ein Wiki auf ihrem Rechner laufen lassen wollen und keine speziellen Plugins und Features benötigen. Dafür sind folgende Schritte nötig: [JSPWiki 2009]

- Schritt 1:** Installieren von Java 5⁶⁰ oder höher
- Schritt 2:** Entpacken von simpleJSPWiki⁶¹ in ein Grundverzeichnis eines Laufwerks
- Schritt 3:** Optional die Zugriffs-URL eintragen, um anderen Anwendern den Zugriff auf das Wiki zu ermöglichen
- Schritt 4:** Ausführen der „startwiki.bat“-Datei
- Schritt 5:** Im Browser kann nun auf das Wiki zugegriffen werden

⁶⁰ <http://www.java.com> (29.04.2010)

⁶¹ <http://www.informatrix.ch/jspwiki/simpleJSPWiki.zip> (29.04.2010)

3.4 Das Austria-Forum

In diesem Kapitel wird zuerst ein kurzer Überblick über das Austria-Forum gegeben. Danach wird die Entstehungsgeschichte beschrieben und anschließend wird ein Auszug über die Funktionen des Forums wiedergegeben.

3.4.1 Allgemeines

Laut [AF 2010] ist das Austria-Forum wie folgt definiert: „*Das Austria-Forum ist eine von unabhängigen Wissenschaftlern und Publizisten gestaltete und der Allgemeinheit zur Verfügung gestellte Wissens- und Diskussionsplattform mit Schwerpunkt Österreich*“.

Das Forum ist nach dem Wiki-Konzept realisiert und bietet daher registrierten Benutzern die Möglichkeit, eigene Beiträge zu verfassen und zu kommentieren. Im Gegensatz zu Wikipedia handelt es sich hierbei nicht um ein komplett offenes System, sondern Artikel können in gewissen Bereichen gesperrt werden und lassen sich daher vor Veränderungen schützen. So kann ein registrierter Benutzer seine Beiträge im Community-Bereich laufend verändern. Hingegen werden Beiträge, die für den Bereich der redaktionellen Wissenssammlung erstellt wurden, gesperrt. Diese Beiträge werden vom Austria-Forum-Team erarbeitet und sind größtenteils qualitativ geprüft. Durch die Sperrung der Seite und die eindeutige Zuordnung der Autoren zum Artikel werden diese im vollen Umfang zitierbar. Außerdem erfolgt die Prüfung durch Experten des jeweiligen Gebiets, das der Artikel umfasst. Weiters kann ein Artikel aus dem Community-Bereich – sofern er formal und inhaltlich den Richtlinien entspricht – nach erfolgreicher Prüfung in den redaktionellen, daher zitierbaren Bereich aufgenommen werden. [AF 2010, Trattner 2009]

Neben dem Community-Bereich und dem Bereich der redaktionellen Wissenssammlung gibt es noch das AEIOU-Lexikon (siehe Kapitel 3.4.2), welches umfangreiche Informationen über Österreich beinhaltet und den größten Teil des Austria-Forums darstellt. Darüber hinaus gibt es noch den Bereich Austria-Forum, in welchem Nutzungsrichtlinien, Grundsätze, Hilfestellungen und zusätzliche informative Angaben über die Verwendung des Austria-Forums beschrieben sind. [AF 2010]

Das Austria-Forum deckt eine breite Palette über Wissen im Zusammenhang mit Österreich ab. Neben Geschichtlichem befinden sich auch Informationen über Denkmäler, Briefmarken, Fauna und Flora im Forum. Weiters ist eine umfangreiche Biographiensammlung berühmter Österreicher vorhanden. Die Artikel sind in Wissenssammlungen organisiert, die zum Zeitpunkt der vorliegenden Arbeit die Anzahl von 20 überschreiten.

Offiziell wurde das Austria-Forum am 19. Oktober 2009 der Öffentlichkeit mit einem Datenbestand von 90.000 Inhalten (Dokumente, Bilder, etc.) vorgestellt. Bis Februar 2010 ist der Datenbestand auf über 120.000 Inhalte gestiegen. [Trattner et al. 2010]

3.4.2 Entstehung

Die Idee einer österreichischen Wissenssammlung existierte schon früher. 1996 entstand durch die Initiative des Bundesministeriums für Wissenschaft und Forschung das österreichische Kulturinformationssystem AEIOU⁶² (Annotierbares Elektronisches Interaktives Oesterreichisches Universal - Informationssystem). Zur Realisierung wurde das Informationssystem Hyperwave⁶³ verwendet, welches auf der TU Graz entwickelt wurde. Insgesamt wurden ungefähr zehn Gigabyte an Daten vom System bereitgestellt. Unter diesen Daten befanden sich unter anderem Tonbeispiele des Instituts für Musikwissenschaft⁶⁴ der Universität Graz und Bildmaterial des Instituts für Realienkunde des Mittelalters und der frühen Neuzeit⁶⁵ der Österreichischen Akademie der Wissenschaften. Laufend wurden Gespräche mit Dateneinbringern geführt und dadurch versucht das Angebot von Informationen zu erweitern. [AEIOU 2010]

Unter dem Namen „Alexander“ wurde 2006 die Weiterentwicklung von AEIOU vorangetrieben. Es wurde versucht die Benutzer und Leser stärker in das System einzubinden und damit deren Mitarbeit an den Beiträgen zu erhöhen. Aus finanziellen Gründen jedoch wurde diese Weiterentwicklung kurze Zeit später eingestellt. [Trattner 2009]

Aus einer Neuausrichtung entstand schlussendlich das Austria-Forum, welches eine zitierfähige Online-Enzyklopädie darstellt. Ursprünglich wurde das Forum noch nicht als Wiki realisiert, sondern es wurde erst in einer späteren Version auf dieses Konzept umgestellt, weil das damals verwendete System nicht mehr zeitgemäß war. Da eine Vielzahl von Wikis zur Verfügung stand, wurde eine Analyse durchgeführt, die die Vor- und Nachteile der einzelnen Systeme gegenüberstellt. Der Vergleich der einzelnen Systeme in Bezug auf die Verwendung für das Austria-Forum ist in der Arbeit von Trattner [Trattner 2009] zu finden. Als Ergebnis dieser Studie wurde als Wiki-Engine JSPWiki eingesetzt, da es die Bedürfnisse des Austria-Forums am besten abdeckt. Auch in Bezug auf das Design und die Bedienbarkeit wurde eine Studie durchgeführt und das alte System mit dem neu eingesetzten JSPWiki verglichen. Details zu den Vorteilen und auch zu auftretenden Schwierigkeiten mit dem neuen System sind in den Arbeiten von [Fressel 2009, Trattner 2009] zu finden.

In der Abbildung 30 ist das ursprüngliche Design des Austria-Forums zu sehen. Beim offiziellen Start des Forums wurde schon ein neues Design, wie in Abbildung 31 zu sehen ist, verwendet.

⁶² <http://www.aeiou.at> (31.05.2010)

⁶³ <http://www.hyperwave.com> (31.05.2010)

⁶⁴ <http://www.uni-graz.at/muwi> (31.05.2010)

⁶⁵ <http://www.imareal.oeaw.ac.at> (31.05.2010)

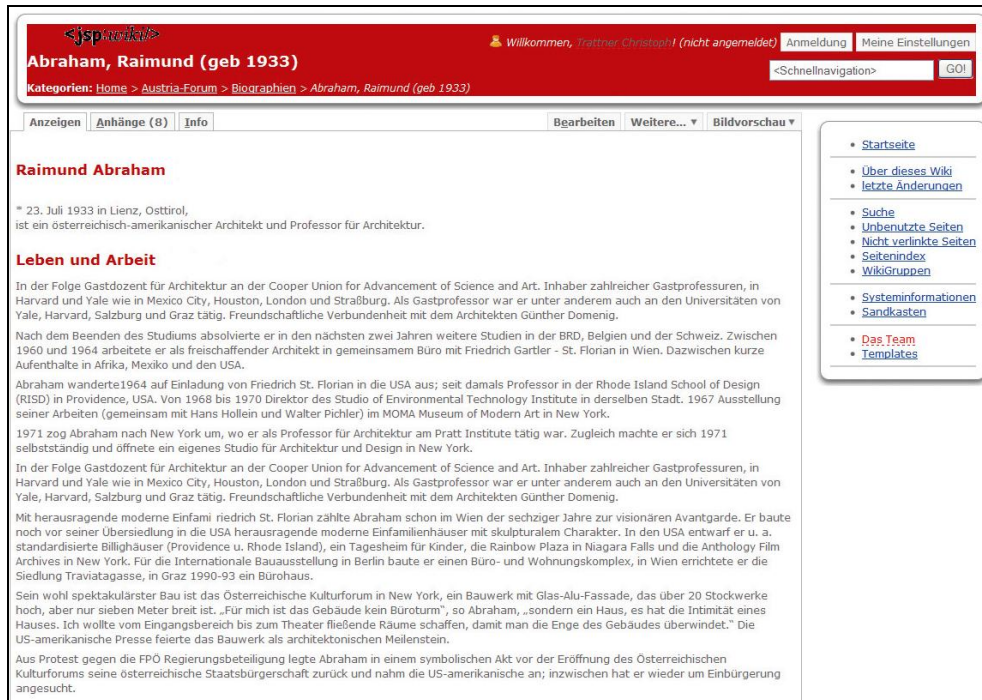


Abbildung 30: Austria-Forum (Altes Design)

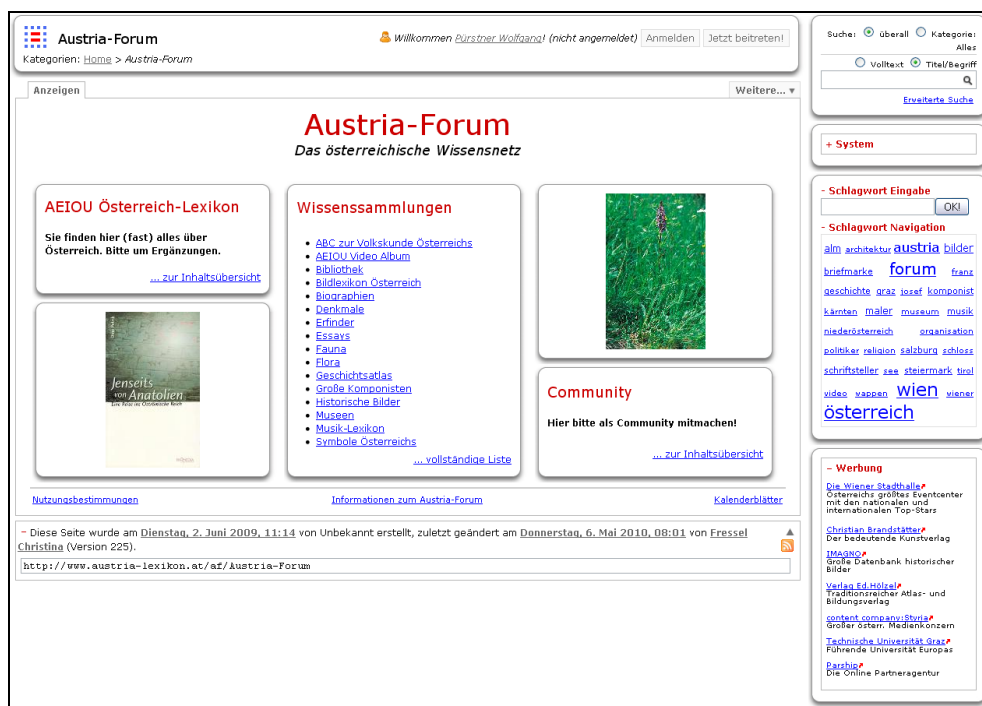


Abbildung 31: Austria-Forum⁶⁶ (Aktuelles Design)

⁶⁶ <http://www.austria-lexikon.at> (28.06.2010)

3.4.3 Funktionen

Wie schon in Kapitel 3.4.1 erläutert ist das Austria-Forum als Wiki realisiert worden und bietet daher auch die für das Wiki-Konzept bekannten Kernfunktionen. Dennoch existieren Einschränkungen, um Probleme, die mit webbasierten Bildungssystemen auftreten, zu verhindern. In der Arbeit von [Trattner et al. 2010] werden drei Hauptprobleme gelistet, die im folgenden Abschnitt kurz zusammengefasst sind.

- Oft werden Texte einfach kopiert und als eigener Text ausgegeben und dies führt zu Plagiaten. Es existieren zwar Programme, mit denen solche Texte untersucht werden und die einen Plagiatsverdacht aufzeigen, aber damit werden nur die Konsequenzen bekämpft und nicht die Ursachen, die ein Plagiat in der Entstehung verhindern.
- Die Informationen, die aus dem Web verwendet werden, müssen nicht korrekt sein oder sie stellen eine verzerrte Sicht dar. Bei Inhalten, auf die mehrere Leute Zugriff haben und diese auch bearbeiten können, existiert die These von der „Weisheit der Menge“ [Surowieki 2005]. Diese besagt, dass durch eine höhere Anzahl an Personen, die den Inhalt bearbeiten, dieser selbst eine höhere Qualität erhält und weniger Fehler beinhaltet. Dennoch kann der Artikel falsche Tatsachen enthalten, da dieser nicht von Experten geprüft ist.
- Ein weiteres Problem besteht darin, dass ein Artikel nicht den gesamten Wissensdurst zu einem bestimmten Bereich abdeckt und sich auf einer Seite häufig nur ein Teil des Gesuchten befindet. Deshalb werden verschiedene Seiten besucht und dort werden auch nur die relevanten Inhalte gelesen – der Rest an Inhalten wird überflogen. Ergebnis eines solchen Suchverhaltens ist die Aneignung eines so genannten „Fragmentierten Wissens“, welches zumeist oberflächlich ist. Weiters ist der Zusammenhang zwischen den verschiedenen Inhalten und Seiten nicht klar erkennbar, da man oft über Suchmaschinen zu den Artikeln gelangt ist.

Zu den genannten Problemen liefern [Trattner et al. 2010] auch Lösungsansätze, die für das Austria-Forum angewendet werden. Das Austria-Forum ist nicht darauf ausgelegt, Allgemeinwissen über die Welt zu publizieren, sondern Detailwissen über Themen im Zusammenhang mit Österreich zu liefern. Ein wichtiger Punkt ist es, zitierfähige Artikel im Austria-Forum zu haben. Dies bedeutet, dass der Autor mit dem realen Namen erkennbar sein muss und der Artikel sich nicht mehr ändern darf. Jedoch beruht das Wiki-Konzept auf der Offenheit des Systems, wobei sich die Frage stellt: Wie gewährleistet man die Vertrauenswürdigkeit und Seriosität des Inhalts, wenn jeder darauf Zugriff hat und diesen verändern kann? Beim Austria-Forum wird dies mit einer Veröffentlichungsprozedur realisiert, die wie folgt konzipiert ist: Im Bereich AEIOU-Lexikon befinden sich Artikel, die von Editoren bearbeitet, auf Qualität und Korrektheit kontrolliert und abschließend gesperrt werden um zitierfähig zu sein. Die redaktionellen Wissenssammlungen werden von Verlagen gespeist, in denen normale Benutzer und Editoren nur Kommentarfunktion haben. In dem letzten Bereich, dem Community-Bereich, kann jeder Benutzer Artikel verfassen oder bestehende verändern. Jedoch werden diese erst gesperrt und damit voll zitierfähig, wenn sie gewisse Kriterien erfüllen und eine Prüfung durch Experten bestehen. Um das Problem mit dem „Fragmentierten Wissen“ einzuschränken, wird versucht mit Hilfe von Kategorien, Empfehlungssystemen und Taggingssystemen Struktur in ein sonst flaches Wiki zu bringen.

Ziel ist es, dem Benutzer ein besseres Verständnis für den Zusammenhang der einzelnen Wiki-Seiten zu geben und somit in ihm ein Gefühl für die Vernetzung der einzelnen Artikel zu erzeugen. [Trattner et al. 2010]

4 Systemarchitektur

In Kapitel 2 wurden ausführlich die verschiedenen Arten und Funktionsweisen von Tagging-Systemen vorgestellt. Kapitel 3 hingegen beschäftigte sich mit dem Wiki-Konzept und verschiedenen Implementierungen davon. Besonders auf das JSPWiki wurde eingegangen sowie auf das Austria-Forum, welches auf dieser Wiki-Engine basiert. In diesem Kapitel wird eine Architektur für ein Plugin für das JSPWiki vorgestellt, das ein Tagging auf Wiki-Seiten erlaubt.

4.1 Grundidee

Wie schon in der Einleitung angeführt besteht ein wesentliches Problem darin, vorhandene Tagging-Systeme auf Wiki-Seiten anzuwenden, weil Wiki-Seiten im Grunde genommen nicht statisch sind sondern sich durch das offene Prinzip auch verändern lassen. Dadurch können direkte Links auf einzelne Passagen, die getaggt wurden, inkonsistent werden.

Weiters besteht mit verfügbaren Tagging-Systemen nicht die Möglichkeit einzelne Sektionen einer Wiki-Seite zu taggen. Entweder kann nur die ganze Seite, Paragraphen (vgl. Spatag.us Kapitel 2.2.3) oder ein anderer Teil einer Wiki-Seite getaggt werden. Jedoch existiert momentan keine Lösung, die sich auf Sektionen beschränkt und sich voll ins Wiki integrieren lässt. Oft wollen Benutzer nicht den ganzen Inhalt markieren sondern nur einen bestimmten Teil mit einem Tag versehen, der für sie wichtig erscheint. Besonders augenscheinlich wird dies bei langen Artikeln. Wird hier die ganze Ressource markiert, obwohl man das Tag eigentlich nur auf einen Absatz anwenden möchte, so wird der ganze Inhalt mit dem Tag versehen, auch wenn dieser nicht im Zusammenhang mit dem Tag steht. Durch die Gliederung der Sektionen werden Tags genauer den betreffenden Inhalten zugeordnet, und diese Stellen lassen sich dann leicht wieder finden, ohne den gesamten Artikel zu durchforsten.

Informationen, die für einen wichtig sind, können auf unterschiedlichen Wiki-Seiten eingebettet sein. Es ist dann durchaus hilfreich, wenn diese Informationen nur auf einer Seite angezeigt werden. Durchs Tagging der Sektionen ist es möglich, diese Bereiche zu markieren. Nun wäre es auch wünschenswert, wenn die markierten Sektionen auf nur einer Seite angezeigt werden würden, welche dann auch gespeichert werden kann. Dabei soll die Option bestehen, Sektionen umzuordnen oder auch zu verwerfen.

Zusammengefasst soll eine Erweiterung konzipiert werden, die zum einen das Taggen von Sektionen erlaubt (SectionTagging) und zum anderen diese markierten Sektionen bei Bedarf listen kann um sie weiter zu verwenden (SectionRanking). Am IICM der TU Graz wurde in der Theorie schon solch ein Konzept erarbeitet und in einer Arbeit von [Helic et al. 2009] abgehandelt. Dieses Konzept dient als Grundlage und wird auf das JSPWiki und in diesem Zusammenhang auch auf das Austria-Forum angewendet.

4.2 Anforderungen

Es ist eine Erweiterung zu implementieren, die folgende Punkte erfüllt. Dabei handelt es sich um eine grobe Zusammenstellung, die später ausführlich im Detail beschrieben wird.

- Wiki-Sektionen sollen einen einfachen Taggingmechanismus erhalten.
- Wiki-Sektionen sollen zu einem bestimmten Tag angezeigt werden, um diese zu sortieren und dann als neue Wiki-Seite speichern zu können.
- Die Erweiterung soll als Plugin/Filter für JSPWiki 2.8.3 verfügbar sein. Der Kern von JSPWiki sollte daher wenn möglich nicht verändert werden, um auch zukünftig Updates auf höhere Versionen ohne großen Aufwand durchführen zu können.
- Sie soll kompatibel zur Scuttle-Datenbank und dem TagCloudPlugin [Zöhrer 2009] sein.

Die zu entwickelnde Software soll dem KISS-Prinzip folgen. KISS ist ein Akronym und hat mehrere bedeutungsähnliche Übersetzungen. „Keep it small and simple“ ist eine Übersetzung davon und bedeutet, dass man ein System so klein und einfach wie möglich gestalten soll um die gesetzten Anforderungen zu erfüllen.

Das Benutzerinterface soll einfach zu bedienen sein und im Design an das Austria-Forum angepasst werden. Animationen, in Form von Einblendungen, sollen die Erweiterung dynamischer wirken lassen. Dabei ist zu beachten, dass solche Animationen nie länger als eine Sekunde dauern, um den Benutzer nicht in seinem Prozess zu behindern.

4.2.1 SectionTagging und SectionRanking

Die Architektur besteht aus zwei Teilen. Zuerst ist das SectionTagging zu nennen, welches den Hauptteil darstellt und deswegen auch den Namen dieser Arbeit geprägt hat. Aufgabe dieser Erweiterung ist es, wie der Name schon sagt, Sektionen, in diesem Fall Abschnitte mit eigenen Überschriften, mit Tags zu versehen.

Da Artikel sehr lang werden können, wäre eine Tagvergabe auf Seitenbasis nicht sinngemäß. Tags hingegen für einzelne Worte zu vergeben, würde das Ziel natürlich auch verfehlen. Deswegen wird eine Unterteilung für das Tagging getroffen, die auf den verschiedenen Abschnitten eines Artikels beruht. Ein Tag wird dabei nicht für den Paragraphen vergeben sondern für die verschiedenen Überschriften, da diese einen neuen Abschnitt einleiten. Besitzt ein Artikel nur eine Hauptüberschrift, so soll auch nur diese mit einem Tag versehen werden können.

Die Tagvergabe soll einfach gestaltet sein und direkt neben der Überschrift aufrufbar sein. Dabei können zu jeder Überschrift die vorhandenen Tags angezeigt werden oder neue vergeben werden.

Das SectionRanking setzt dann auf die vorhandenen Tags auf. Wird ein Tag ausgewählt, so werden alle Sektionen, für die dieser Tag vergeben wurde, in einer Liste angezeigt. Diese Liste kann neu geordnet werden und dann als neue Wiki-Seite abgespeichert werden.

4.2.2 JSPWiki-Kompatibilität

Das Austria-Forum basiert auf JSPWiki und daher soll sich die Erweiterung nahtlos in die bestehende Umgebung einfügen. JSPWiki beinhaltet ein modulares Konzept, in dem Erweiterungen in Form eines Plugins oder Filters hinzugefügt werden können. Der Unterschied zwischen Plugin und Filter ist in Kapitel 3.3.2 beschrieben.

Zusätzlich soll die Erweiterung auch Rücksicht auf bestehende Plugins nehmen, die eigens für das Austria-Forum entwickelt wurden. Die Tags sollen zum Beispiel in einer Datenbank gespeichert werden, die auch von dem TagCloudPlugin [Zöhler 2009] benutzt wird. Dabei handelt es sich um ein Plugin, das eine Tag-Cloud, wie in Kapitel 2.1.7 beschrieben, erzeugt. Diese wird dann zur Navigation und zum Browsen im bestehenden Austria-Forum verwendet.

4.2.3 Browserunabhängigkeit

Da das Austria-Forum für jeden frei zugänglich ist, soll auch jeder mit seinem Browser darauf zugreifen können. Dies gilt natürlich auch für die Erweiterungen des Austria-Forums. Das Plugin muss daher in jedem Browser gleich dargestellt werden. Hauptgewicht liegt auf den gängigen Browsern wie Mozilla Firefox⁶⁷, Microsoft Internet Explorer⁶⁸ (IE), Apple Safari⁶⁹, Google Chrome⁷⁰ und Opera⁷¹. In Tabelle 10 ist eine Browserstatistik von den ersten Monaten im Jahr 2010 zu sehen.

2010	Firefox	IE	Chrome	Safari	Opera
Mai	46,9%	32,2%	14,5%	3,5%	2,2%
April	46,4%	33,4%	13,6%	3,7%	2,2%
März	46,2%	36,2%	12,3%	3,7%	2,2%
Februar	46,5%	35,3%	11,6%	3,8%	2,1%
Jänner	46,3%	36,2%	10,8%	3,75	2,2%

Tabelle 10: Browserstatistik⁷² für das Jahr 2010 von Jänner bis Mai

⁶⁷ <http://www.mozilla.com> (04.06.2010)

⁶⁸ http://www.microsoft.com/internet_explorer (04.06.2010)

⁶⁹ <http://www.apple.com/at/safari> (04.06.2010)

⁷⁰ <http://www.google.com/chrome> (04.06.2010)

⁷¹ <http://www.opera.com/browser> (04.06.2010)

⁷² http://www.w3schools.com/browsers/browsers_stats.asp (04.06.2010)

Die Spalte für den Internet Explorer setzt sich größtenteils aus den drei letzten Hauptversionen zusammen. Die Erweiterung richtet sich aber an der Version 8 aus, da diese standardkonformes HTML und CSS besser unterstützt und dadurch auch korrekt darstellen kann.

4.2.4 Datenbank

Als Datenspeicher soll eine schon existierende Datenbank verwendet werden. Dabei handelt es sich um das Datenbankschema von Scuttle. Scuttle ist ein Social Bookmarking System, welches in Kapitel 2.2.1 beschrieben wurde. Da diese Datenbank auch vom TagCloudPlugin benutzt wird, soll sie ohne Änderungen im gleichen Umfang verwendet werden, sodass die vom SectionTaggingPlugin gespeicherten Tags auch zur TagCloud-Erzeugung dienen. Auf die Datenbank kann direkt zugegriffen werden, ohne das Interface zu benutzen, das Scuttle zur Verfügung stellt. Dadurch, dass auch keine neuen Tabellen angelegt werden sollen, ist die Benutzung der Scuttle-Oberfläche durchaus möglich.

4.2.5 Zusätzliche Anforderungen

Da JSPWiki vollständig UTF-8 für die Kodierung der Seiten verwendet, soll auch die Erweiterung diese Kodierungsmöglichkeit nutzen. Daher ist es zum Beispiel auch möglich, Tags mit kyrillischen Zeichen zu vergeben.

Die Benutzerdaten werden automatisch aus dem JSPWiki gelesen und für die Benutzergenerierung und die Tagspeicherung in der Scuttle-Datenbank verwendet.

Wiki-Seiten, die nicht gesperrt sind, unterliegen einer ständigen Änderung, was zur Folge hat, dass gesetzte Tags auf Überschriften ungültig werden können. Um einem späteren Tool oder Plugin die Suche nach solch ungültigen Tags zu erleichtern und entsprechende Maßnahmen zu ergreifen, soll in der Scuttle-Datenbank zu jedem Bookmark auch die Versionsnummer der Wiki-Seite abgespeichert werden.

Die Einstellungen und ausgegebenen Texte beziehungsweise Fehlermeldungen werden in Konfigurationsdateien verwaltet, damit eine leichte Änderung von diesen stattfinden kann.

Weiters sollen nur registrierte und eingeloggte Benutzer die Möglichkeit haben zu taggen. Anonyme Benutzer haben zwar Lesezugriff auf die vorhandenen Tags, können diese aber nicht um eigene erweitern.

4.3 Datenbank-Diagramm

Wie schon erwähnt, wird eine vorhandene Datenbank von dem frei verfügbaren Bookmarking-System Scuttle übernommen. Diese relationale Datenbank verfügt schon über alle notwendigen Einträge, um ein System mit den Anforderungen von SectionTagging implementieren zu können. Verwendet wird das Datenbankschema der Version 0.7.4, um die Kompatibilität zu anderen Plugins (siehe TagCloudPlugin) zu gewährleisten. Es sind vier Tabellen vorhanden, die wie folgt aufgebaut sind:

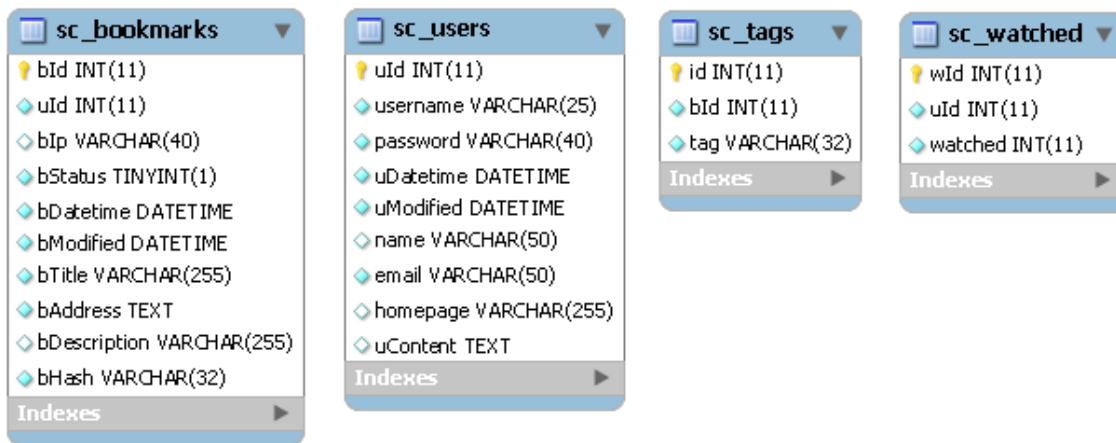


Abbildung 32: Datenbankschema von Scuttle

Die Beziehung der Tabellen untereinander soll folgendes Entity-Relationship-Model (ERM)-Diagramm in Abbildung 33 zeigen:

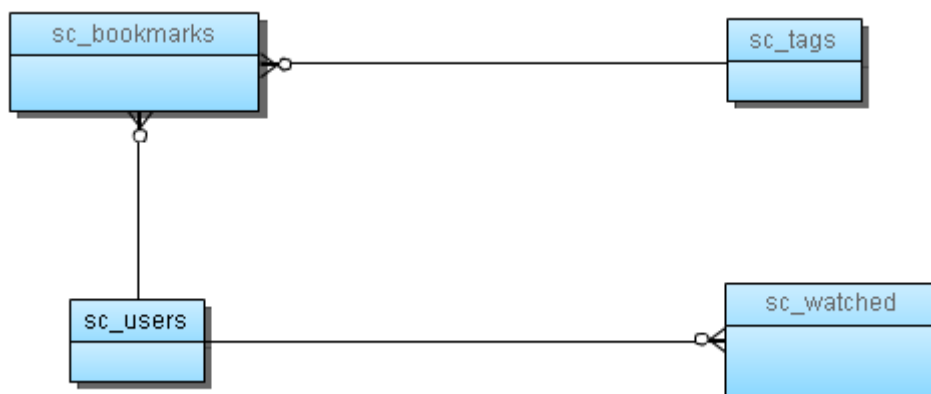


Abbildung 33: ERM-Diagramm von Scuttle

4.3.1 Tabelleninhalt

Die Tabelle „sc_bookmarks“ enthält die Bookmarks im eigentlichen Sinne und besteht aus folgenden Attributen:

Attribute	Funktion
bId	Eindeutige Identifikationsnummer des Bookmarks
uId	Eindeutige Identifikationsnummer des Users, der diesen Bookmark erstellt hat
bIp	IP-Adresse des Users, der diesen Bookmark erstellt hat
bStatus	Enthält den Status des Bookmarks (Private, Public oder Shared)
bDatetime	Datum und Zeit der Bookmarkerstellung
bModified	Datum und Zeit der letzten Änderung an einem Attribut der Tabelle
bTitle	Titel des Bookmarks
bAddress	URL, die auf die zu verlinkende Seite zeigt
bDescription	Eine Beschreibung des Bookmarks
bHash	Speichert den Hash des Bookmarks

Tabelle 11: Datenbanktabelle "sc_bookmarks"

Die Tabelle „sc_user“ beinhaltet alle Angaben des Benutzers, der Bookmarks erstellt, und setzt sich wie folgt zusammen:

Attribute	Funktion
uId	Eindeutige Identifikationsnummer des Benutzers
username	Benutzername
password	Enthält einen Secure Hash Algorithm (SHA)-Wert des Passworts
uDatetime	Datum und Zeit der Benutzererstellung
uModified	Datum und Zeit der letzten Änderung an einem Attribut der Tabelle
name	Vor- und Nachname des Benutzers
email	Email-Adresse
homepage	Homepage
uContent	Beschreibung

Tabelle 12: Datenbanktabelle "sc_user"

Die Tabelle „sc_tags“ beinhaltet die Tags, die zu einem Bookmark gehören:

Attribute	Funktion
id	Eindeutige Identifikationsnummer des Tags
bId	Eindeutige Identifikationsnummer des Bookmarks, zu dem das Tag gehört
tag	Das Tag selbst

Tabelle 13: Datenbanktabelle "sc_tags"

Zusätzlich gibt es noch eine Tabelle „sc_watched“, in der Beobachtungslisten gespeichert werden.

Attribute	Funktion
wId	Eindeutige Identifikationsnummer der Beobachtung
uId	Eindeutige Identifikationsnummer des Benutzers, der einen anderen Nutzer beobachtet (Beobachter)
watched	Eindeutige Identifikationsnummer des Beobachteten

Tabelle 14: Datenbanktabelle "sc_watched"

4.3.2 Spezifische Verwendung

Vom TagCloudPlugin werden nur zwei Tabellen verwendet, nämlich „sc_bookmarks“ und „sc_tags“, da für die Tagcloud-Generierung die verschiedenen Benutzer und Beobachtungslisten nicht von Bedeutung sind.

Das SectionTaggingPlugin wird alle Tabellen außer „sc_watched“ verwenden. Beobachtungslisten spielen für das SectionTagging und SectionRanking keine Rolle. Dennoch steht diese Funktion der Beobachtungslisten weiterhin über die Scuttle-Oberfläche zur Verfügung. Da auch die Versionsnummer der Wiki-Seite abgespeichert werden soll, aber kein Attribut in der entsprechenden Tabelle existiert und ferner keine Änderungen an dieser durchgeführt werden sollen, wird die Versionsnummer einfach im Attribut „bDescription“ abgespeichert. Das Attribut „bDescription“ besitzt für das SectionTagging und SectionRanking keine relevante Bedeutung und wird daher für die Versionierung benutzt.

4.4 Ranking

Ein bestimmtes Tag kann verschiedenen Sektionen vergeben werden. Auch eine Sektion kann mehrmals dasselbe Tag bekommen, sofern die Vergabe von unterschiedlichen Benutzern durchgeführt wurde. So kann ein Tag einer Vielzahl von Sektionen zugeordnet sein und dadurch entsteht auch das Bedürfnis einer sinnvollen Sortierung, wenn alle Sektionen angezeigt werden sollen. In den folgenden Unterkapiteln wird ein Algorithmus vorgestellt, der auf zwei wesentlichen Hypothesen beruht und anhand eines Anwendungsbeispiels demonstriert wird.

4.4.1 Verwendeter Algorithmus

Wenn die verfügbaren Tags zu einer Überschrift angezeigt werden, unterliegen sie einer Rangfolge, nach der sie sortiert werden. In Kapitel 2.1.4 wurden schon die bekanntesten Selektionsalgorithmen aufgezählt. Die Reihung der Tags bei der Taganzeige erfolgt mittels des Selektionsalgorithmus „num-item-apps“, der die einzelnen Tags nach deren Häufigkeit auf einen bestimmten Inhalt reiht. Dieser Algorithmus gehört zu den impliziten Algorithmen und stellt eine einfache Möglichkeit dar, dem Benutzer die Beliebtheit der Tags zu einer Ressource zu verdeutlichen.

4.4.2 Hypothesen

Für das SectionRanking ist eine sinnvolle Reihung der einzelnen Sektionen von entscheidender Bedeutung, sodass wichtige Informationen, die dieses Tag betreffen, an vorderster Stelle gereiht werden. Für diese Rangfolge wird ein eigener Selektionsalgorithmus angewendet, der auf folgenden zwei Hypothesen beruht:

Hypothese 1: Vergibt ein Benutzer weniger Tags zu einem bestimmten Inhalt, so kommt diesen eine höhere Bedeutung zu, da sie gezielt ausgewählt wurden. Vergibt ein Benutzer hingegen viele Tags, so erhalten diese eine geringere Bedeutung bei der Reihung im Vergleich zu gleichen Tags von anderen Inhalten.

Hypothese 2: Die Position, an der ein Tag in einer Reihe von Zuordnungen vergeben wurde, ist von hoher Bedeutung, da das erste vergebene Tag dem Benutzer zuerst in den Sinn gekommen ist. Zur Verdeutlichung dieser Hypothese soll folgendes Beispiel dienen:



Abbildung 34: Schildkröte

Soll der Inhalt der Abbildung 34 mit Tags versehen werden, so wird dieser höchstwahrscheinlich wie folgt getaggt:

Schildkröte, Wasser, Stein, Gras, Algen, Welle

Dementsprechend kann man davon ausgehen, dass die zuerst eingegebenen Tags dem Betrachter zuerst in den Gedanken aufgeblitzt sind und den Inhalt womöglich am besten beschreiben. Die darauf folgenden Tags sind dann von geringerer Bedeutung und verlieren je nach Position an Wichtigkeit.

Auf diesen zwei Hypothesen beruht der Selektionsalgorithmus für das SectionRanking. Der Ablauf sieht dann folgendermaßen aus:

Ein Tag wird vorgegeben, auf dem die Reihenfolge beruhen soll. Zu diesem werden alle Sektionen (Bookmarks) in der Datenbank herausgesucht. Jede einzelne Sektion bekommt Punkte vergeben, deren Höhe auf die beiden Hypothesen beruht. Je höher die Punktezahl, umso wichtiger ist die Sektion. Da ein Tag auch mehrmals von verschiedenen Benutzern an eine Sektion vergeben werden kann, werden die Punkte der Sektionen, die sich auf denselben Inhalt beziehen, zusammengezählt. Abschließend werden die Sektionen nach ihrem Ergebnis gereiht, wobei die Sektion mit den meisten Punkten an erster Stelle steht.

4.5 Architektur von JSPWiki

Um zu verstehen wie das SectionTagging- und das SectionRanking-Modul in das JSPWiki eingefügt werden, muss die Architektur von JSPWiki genauer betrachtet werden. Wichtig für die Erweiterung sind das Plugin- und das Filtersystem, welche in den folgenden zwei Unterkapiteln erklärt werden.

4.5.1 Plugin

Der Aufruf von einem Plugin im Markup-Text erfolgt – wie in Kapitel 3.3.3 beschrieben – mit dem Namen des Plugins, eingefasst in Klammern. Parameter können optional übergeben werden. Dieser Markup-Code vom Aufruf wird dann bei der Seitengenerierung durch den HTML-Code, welcher vom Plugin erzeugt wird, ersetzt.

Um ein Plugin zu erstellen, wird eine eigene Pluginklasse von einem Interface (WikiPlugin) implementiert und im selben Verzeichnis gespeichert. Der Aufruf des Plugins erfolgt durch den Namen der Klasse. Dabei wird die Funktion „execute()“ aufgerufen, die den Rückgabewert (HTML-Code) in die generierte Wiki-Seite einfügt. Im Quellcode 5 ist ein einfaches Plugin zu sehen, das ein HTML Span-Element mit dem Inhalt „output“ zurückliefert. Als Parameter stehen dem Benutzer der Kontext der Seite sowie die im Markup mitübergebenen Parameter als Datentyp „Map“ zur Verfügung.

```
package com.ecyrd.jspwiki.plugin;

import com.ecyrd.jspwiki.*;

public class MyPlugin implements WikiPlugin
{
    public String execute( WikiContext Context, Map ParameterMap )
    {
        return "<span>output</span>";
    }
}
```

Quellcode 5: JSPWiki - Plugin

4.5.2 Filter

Bei dem Filter-System handelt es sich um ein Interface, welches die Manipulation von Wiki-Seiten erlaubt. Der Zugriff auf Wiki-Seiten kann auf unterschiedlichen Stufen der Seitengenerierung oder Speicherung erfolgen. So kann der Markup-Code der Seite verändert werden oder auch der HTML-Code der Seite, der dann im Browser angezeigt wird. Im Quellcode 6 ist das Interface mit den verschiedenen Eingriffsmöglichkeiten zu sehen.

```
public interface PageFilter
{
    public void initialize( WikiEngine engine, Properties properties )
        throws FilterException;

    public String preTranslate( WikiContext wikiContext, String content )
        throws FilterException;

    public String postTranslate( WikiContext wikiContext,
                                String htmlContent )
        throws FilterException;

    public String preSave( WikiContext wikiContext, String content )
        throws FilterException;

    public void postSave( WikiContext wikiContext, String content )
        throws FilterException;

    public void destroy( WikiEngine engine );
}
```

Quellcode 6: JSPWiki Filterinterface

Beim Start von JSPWiki wird die Funktion „initialize()“ jedes aktivierten Filters aufgerufen, in der verschiedene Initialisierungen durchgeführt werden können um den Filter betriebsbereit zu machen. Wird eine Wiki-Seite angefordert und das Markup aus dem Datenspeicher geladen, so erhält man mit der Funktion „preTranslate()“ Zugriff auf dieses Markup, um es zu verändern. Danach wird eine HTML-Seite daraus generiert und ein Zugriff, um diesen HTML-Code zu manipulieren, wird mit der Funktion „postTranslate()“ zur Verfügung gestellt. Erst danach erscheint die Wiki-Seite dem Benutzer im Browser. Mit der Funktion „preSave()“ kann wiederum das Markup vor dem Abspeichern im Datenspeicher verändert werden, wenn zum Beispiel ein Benutzer eine Seite geändert hat. Die letzte Zugriffsmöglichkeit, die nach dem Abspeichern der Wiki-Seite aufgerufen wird, erhält man mit der Funktion „postSave()“. In dieser Funktion kann zwar der Markup- oder HTML-Code einer Wiki-Seite nicht mehr geändert werden, aber es können verschiedene Routinen, wie Email-Benachrichtigungen, durchgeführt werden. So wie es eine Initialisierungsfunktion gibt, existiert natürlich auch ein Gegenstück dazu. Mit der Funktion „destroy()“ können abschließende Operationen ausgeführt werden.

4.5.3 Ablauf einer Seitengenerierung

Die Funktionsweise des Plugin- und Filtersystems soll anhand eines Ablaufs eines Seitenaufrufs verdeutlicht werden. Es wird angenommen, dass JSPWiki erfolgreich gestartet ist und die Initialisierungsfunktionen der Filter aufgerufen wurden.

1. Wiki-Seite wird von Benutzer angefordert
2. Wiki-Seite wird aus dem Datenspeicher geladen
3. „preTranslate()“ wird aufgerufen
4. Aus dem Markup wird ein HTML-Code generiert, wobei unter anderem auch Plugins ausgeführt werden. Bei den Plugins wird die Funktion „execute()“ aufgerufen und der Rückgabewert (HTML-Code) an der Stelle in der Wiki-Seite eingefügt, an der der Pluginaufruf stand.
5. „postTranslate()“ wird aufgerufen
6. Wiki-Seite wird an den Browser des Benutzers gesendet

4.5.4 Aufbau einer URL

Der Aufbau einer URL im JSPWiki ist für die Erweiterung von enormer Wichtigkeit, da die Tagzugehörigkeit zu einer Wiki-Seite dadurch eindeutig bestimmt werden soll. Da in der Datenbank absolute Links gespeichert werden um die Funktion der Scuttle-Weboberfläche zu gewährleisten, wirkt sich eine Änderung des Domain-Namens oder des Wiki-Namens fatal aus, nämlich derart, dass alle Links in der Datenbank ungültig werden. Eine URL, die auf einen Artikel im JSPWiki verweist, besitzt folgenden Aufbau:

http:// <Domain-Name> / <Wiki-Name> / <Kategorie(n)> / <Wiki-Seite>

Als Beispiel dient die Wiki-Seite vom Austria-Forum von der Tanne⁷³:

http://www.austria-lexikon.at/af/Wissenssammlungen/Flora/Tanne

URL-Bestandteil	Erklärung
www.austria-lexikon.at	Domain-Name
af	Wiki-Name
Wissenssammlungen/Flora	Kategorie
Tanne	Wiki-Seite

Tabelle 15: URL-Bestandteile

⁷³ <http://www.austria-lexikon.at/af/Wissenssammlungen/Flora/Tanne> (10.06.2010)

Wie zu erkennen ist werden gleich zwei Kategorien angesprochen. Zuerst die Hauptkategorie „Wissenssammlungen“ und dann die Unterkategorie „Flora“. Die Kategorie kann natürlich auch entfallen, sollte sich die Wiki-Seite im Grundverzeichnis befinden.

Da jedoch nicht die Seite selbst getaggt werden soll, sondern eine Überschrift, ist eine feinere Unterteilung nötig. JSPWiki beinhaltet bereits eine solche Gliederung mit Hilfe von Anker (siehe [Raggett et al. 1999]). Dabei wird jede Überschrift mit einer Identifikation versehen und lässt sich daher gezielt mit einer erweiterten URL ansprechen. Anhand der Wiki-Seite „Graz“⁷⁴ des Austria-Forums wird diese Vorgehensweise veranschaulicht.

Der Link zu der Seite sieht wie folgt aus:

<http://www.austria-lexikon.at/af/AEIOU/Graz>

Mit diesem Link gelangt man direkt auf die Wiki-Seite, welche dann von Beginn an dargestellt wird. Diese Seite enthält auch kleinere Abschnitte, die durch Überschriften eingeleitet werden und durch einen Anker markiert sind. Um den Abschnitt mit der Überschrift „Klima“ zu adressieren ist folgende URL nötig:

http://www.austria-lexikon.at/af/AEIOU/Graz#section-AEIOU_2FGraz-Geschichte

Der erste Teil der URL entspricht denselben Kriterien wie der Aufruf zur Seite „Graz“. Jedoch wird mit dem Rautezeichen „#“ die Sektion definiert und nach Aufruf dieses Links wird die Seite automatisch zum Kapitel „Geschichte“ gescrollt. Der Sektionsteil ist wie folgt definiert:

[#section-AEIOU_2FGraz-Geschichte](#)

Da „section-AEIOU_2FGraz-Geschichte“ genau der Identifikation der Überschrift entspricht, wird das Browserfenster zu diesem Abschnitt geleitet. Zur besseren Erklärung, wie JSPWiki genau die Identifikation der Sektionen vergibt, soll die nachstehende Aufschlüsselung dienen:

Identifikationsbestandteil	Erklärung
#	Leitet die Sektionsdefinition ein
section-	Konstante von JSPWiki um eine Sektion zu markieren
AEIOU	Kategorie(n) in der sich die Seite befindet
_2F	Steht für einen Schrägstrich „/“
Graz	Definiert die Wiki-Seite
-	Leitet die Kennung für die Sektion ein
Geschichte	Steht für den Namen der Überschrift

Tabelle 16: Identifikationsbestandteile der Sektion

⁷⁴ <http://www.austria-lexikon.at/af/AEIOU/Graz> (10.06.2010)

Weiters erschwert die Verwendung von Steuer- und Sonderzeichen in der Kategorie, im Namen oder in der Sektionsüberschrift die Lesbarkeit des Links, weil JSPWiki diese alle umwandelt. Diese Umwandlung gilt fast ausschließlich für den Sektionsdefinitionsteil. Dabei werden die Steuer- und Sonderzeichen in ASCII-Zahlenwerte transferiert und das führende Prozentzeichen „%“ durch einen Unterstrich „_“ ersetzt.

Als Beispiel dient die Biographie eines Professors vom Austria-Forum mit dem Namen Christian Köberl⁷⁵. Um dort die Hauptüberschrift anzusteuern, die „Köberl, Christian“ lautet, ist folgender Link notwendig:

http://www.austria-lexikon.at/af/Wissenssammlungen/Biographien/Köberl%2CChristian#section-Wissenssammlungen_2FBiographien_2FK_C3_B6berl_2C+Christian-K_C3_B6berlChristian

Durch die Umwandlung der Umlaute, Beistriche und anderen Sonderzeichen wird der Link im Sektionsteil schwer lesbar. Um die Sektion im Wiki-Markup unter zu Hilfenahme dieses Links wieder zu finden, ist ein entsprechender Algorithmus anzuwenden, der die verschiedenen Umwandlungen wieder rückgängig macht um den ursprünglichen Überschriftsnamen „Köberl, Christian“ zu erlangen.

Ein weiteres Problem entsteht dann, wenn zwei Überschriften dieselbe Bezeichnung besitzen. Da sie anhand der Identifikation unterscheidbar sein müssen, wird die der zweiten Überschrift mit dem Zusatz „-2“ verlängert. Sollten noch weitere Überschriften mit selben Namen folgen, erhöht sich der Zusatz der Identifikation jeweils um eins.

Um nun ein Tag eindeutig einer Sektion einer Wiki-Seite zuweisen zu können, wird neben dem Link auf die Sektion auch noch die Versionsnummer benötigt. Wie schon erwähnt wird diese nicht im Link mit abgespeichert, sondern sie wird in der Beschreibung des Bookmarks in der Datenbank aufgenommen.

⁷⁵ <http://www.austria-lexikon.at/af/Wissenssammlungen/Biographien/K%C3%B6berl%2C%20Christian> (20.06.2010)

4.6 SectionTagging

Den Ablauf zur Seitengenerierung, welcher in Kapitel 4.5.3 beschrieben wurde, wird sich die SectionTagging-Erweiterung zunutze machen, um die Tagging-Funktion in die Wiki-Seite zu integrieren. Dabei sind folgende Punkte zu berücksichtigen:

- Die Überschriften müssen identifiziert werden. Dies geschieht in der Funktion „preTranslate()“, weil dort im Markup die Überschriften mit Rufzeichen gekennzeichnet sind. Rufzeichen, die nicht am Zeilenbeginn stehen, stellen auch keine Überschriften dar und werden aussortiert. Danach wird der SectionTagging-Pluginaufruf in der nächsten Zeile nach jeder Überschrift eingefügt, um die entsprechenden Stellen zu markieren. Die Identifizierung muss in der Funktion „preTranslate()“ durchgeführt werden, da in der Funktion „postTranslate()“ auch Textbausteine die gleiche Formatierung erhalten können wie Überschriften und daher von diesen nicht mehr unterscheidbar wären.
- Nach dem Aufruf der Funktion „preTranslate()“ sind alle Überschriften mit dem SectionTaggingPlugin versehen. Bei der Übersetzung des Markups in den HTML-Code wird dann anstelle des Pluginaufrufs eine weitere Markierung gesetzt, um für die nachfolgende Methode die ursprüngliche Aufrufstelle erkennbar zu machen.
- Erst in der Funktion „postTranslate()“ werden die eigentlichen SectionTagging-Komponenten in den HTML-Code eingefügt. Diese erlauben es dem Benutzer dann, Überschriften mit Tags zu versehen und schon vergebene Tags anzeigen zu lassen.

Danach steht dem Benutzer auf der Wiki-Seite zu jeder Überschrift ein Button zur Verfügung. Wird der Mauszeiger über den Tagging-Button bewegt, so werden alle vorhandenen Tags zu dieser Überschrift angezeigt. Klickt der Benutzer dann einen dieser Tags an, wird dieses Tag sogleich an die SectionTagging-Komponente weitergereicht, welche dann eine Reihung aller Sektionen, die dieses Tag enthalten, erstellt.

Möchte ein Benutzer selbst Tags vergeben, so klickt er den Tagging-Button direkt an und es erscheint ein Fenster, wo er seine Tags eingeben kann. Hat er jedoch schon Tags vergeben, werden diese im Fenster auch angezeigt, wodurch diese sich auch verändern lassen.

4.7 SectionRanking

Das SectionRanking bekommt den gleichen Platz im Browserfenster wie ein Artikel einer Wiki-Seite. Die Kopfleiste und das Menü links und rechts bleiben bestehen und in der Mitte werden dann zwei Spalten angezeigt. Die rechte Spalte enthält alle Sektionen, die mit einem bestimmten Tag versehen wurden. Hingegen dient die linke Spalte zur Neusortierung der Sektionen, die dann im Anschluss auch als neue Wiki-Seite abgespeichert werden kann.

5 Implementierung

Dieses Kapitel beschreibt den Prototypen und dessen spezifische Implementierung. Die im Kapitel 4 dargestellten Anforderungen wurden durch Einsatz verschiedener Techniken umgesetzt. Darum wird in den folgenden Abschnitten auch auf die verwendeten Technologien und ihren Einsatz näher eingegangen. Zuerst wird die Hauptkomponente, das SectionTagging, im Detail beschrieben und anschließend das SectionRanking.

5.1 SectionTagging

Für SectionTagging wurden unterschiedliche Technologien eingesetzt. Für die Integration in die Wiki-Seite wurde auf das Erweiterungskonzept von JSPWiki zurückgegriffen. Für das Einfügen der SectionTagging-Komponente wurden ein spezieller Filter (SectionTaggingFilter) und ein spezielles Plugin (SectionTaggingPlugin) entwickelt. Der Filter fügt JavaScript-Code in die Wiki-Seite ein, um die Tageingabe und Taganzeige zu realisieren. Das JavaScript greift mittels Asynchronous JavaScript and XML (AJAX) dann auf eigens dafür entwickelte JSP-Dateien zu, um Daten zu der benötigten Sektion zu bekommen oder zu speichern. In den JSP-Dateien werden Java-Klassen aufgerufen, die Zugriff auf die Datenbank haben.

5.1.1 Integrierung der SectionTagging-Komponente

Da Filter und Plugins auf allen Bereichen der Wiki-Seite (Hauptseite, linkes/rechtes Menü, Kopfleiste) angewendet werden, wird eine Unterscheidung getroffen, sodass die Tagging-Funktion nur auf der Hauptseite möglich ist.

Wird eine Wiki-Seite aus dem Datenspeicher geladen, so durchsucht die erste Filterfunktion „preTranslate()“ das Markup nach Rufzeichen am Zeilenbeginn, weil diese Überschriften darstellen. Nach jeder Überschrift wird dann der Pluginaufruf „[{SectionTaggingPlugin}]“ eingefügt. Dies geschieht aber in der darauf folgenden Zeile, da sonst der Pluginaufruf beim Übersetzen in HTML auch als Teil der Überschrift erkannt wird und dementsprechend behandelt und formatiert wird. Ein Pluginaufruf findet dabei nicht statt. Dadurch, dass der Pluginaufruf in der nächsten Zeile nach der Überschrift eingefügt wurde, wird während des Übersetzens in HTML-Code der Pluginaufruf durch die Markierung, wie sie in Quellcode 7 ersichtlich ist, ersetzt.

```
<span class="sectionTagging"></span>
```

Quellcode 7: Pluginausgabe

Es handelt sich hierbei um ein SPAN-Element mit dem class-Attribut „sectionTagging“. Dadurch lassen sich die markierten Stellen von der darauf folgenden Filterprozedur „postTranslate()“ leicht finden. Diese fügt dann innerhalb der davor liegenden Überschrift einen Button (siehe Abbildung 35) für das SectionTagging in den HTML-Code ein.



Abbildung 35: SectionTagging-Button

Der aufwendige Vorgang des Einfügens ist deswegen nötig, weil der Button noch nicht in der „preTranslate()“-Funktion eingefügt werden kann, da es sich hier um ein Wiki-Markup handelt. Die „preTranslate()“-Funktion ist aber für die Auffindung der Überschriften unverzichtbar. Weiters kann das Plugin den Button nicht einfügen, obwohl es sich um ein HTML-Markup handelt. Das liegt daran, dass das Plugin eine Zeile unterhalb der Überschrift eingefügt wurde, der Button jedoch neben der Überschrift erscheinen soll. Deswegen ist noch ein Aufruf der „postTranslate()“-Funktion nötig, die den Button in die davor liegende Überschrift integriert.

In der Abbildung 36 ist eine Seite⁷⁶ vom Austria-Forum ohne SectionTagging zu sehen und in Abbildung 37 mit aktiviertem SectionTagging. Es befinden sich auf der Hauptseite insgesamt drei Überschriften, die mit aktiviertem SectionTagging einen Button zum Taggen erhalten.

The screenshot shows the Austria-Forum website. At the top, there is a navigation bar with the forum logo and the text 'Austria-Forum'. Below it, there are search options: 'Suche: überall' (selected) and 'Fauna' (selected). The main content area features a large image of three hedgehogs (one adult and two juveniles) with the title 'Igel'. To the right of the image, there is taxonomic information: 'Klasse: Säugetiere/Mammalia', 'Ordnung: Insektenfresser/Insectivora', and 'Familie: Igel/Erinaceidae'. Below this, two species are listed: 'Braunbrustigel (Erinaceus europeus)' and 'Weißbrustigel (Erinaceus europaeus roumanicus)'. A detailed description follows. The sidebar on the right contains several widgets: '+ System', '- Schlagwort Eingabe' with an 'OK!' button, '- Schlagwort Navigation' with the word 'maus', and '- Werbung' with several advertisement links.

Abbildung 36: Wiki-Seite ohne SectionTagging

⁷⁶ <http://www.austria-lexikon.at/af/Wissenssammlungen/Fauna/Igel> (11.06.2010)

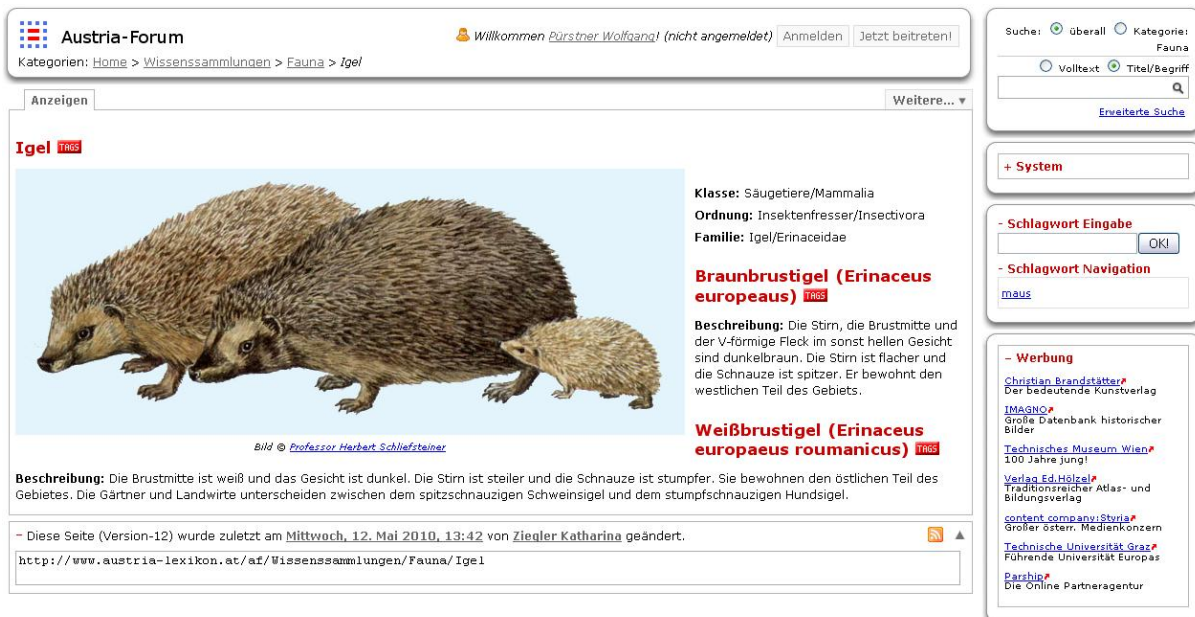


Abbildung 37: Wiki-Seite mit SectionTagging

Nachdem der Tagging-Button in die Seite integriert ist, kann dieser auch genutzt werden. Es existieren zwei Funktionen. Zum einen können schon vergebene Tags angezeigt werden, indem man den Mauszeiger über den Button bewegt. Daraufhin öffnet sich das Taganzeigefenster (siehe Kapitel 5.1.2). Zum anderen dient der Button auch zur Tagvergabe. Das Tagbearbeitungsfenster (siehe Kapitel 5.1.3) wird durch einen Mausklick auf den Button geöffnet.

5.1.2 Taganzeigefenster

Das Taganzeigefenster dient zur Anzeige der schon vergebenen Tags. Es wird rechts unterhalb des Mauszeigers versetzt dargestellt. Weiters wird durch einen Klick auf ein vorhandenes Tag die SectionRanking-Komponente gestartet. Um den Benutzer nicht zu irritieren, wird das Taganzeigefenster nicht sofort beim Überfahren des Tagging-Buttons angezeigt, sondern mit einer kurzen Zeitverzögerung, sofern er mit dem Mauszeiger über dem Button verweilt. Dadurch wird ein Öffnen des Taganzeigefensters vermieden, sollte der Benutzer nur zufällig den Tagging-Button mit dem Mauszeiger streifen. Ferner verringert sich auch die Anzahl der Anfragen an den Server, da die Tags nur dann geladen werden, wenn sich das Anzeigefenster öffnet. Sind keine Tags vorhanden, wird ein Fenster – wie in Abbildung 38 ersichtlich – dargestellt.



Abbildung 38: Taganzeigefenster ohne Tags

Auch etwaige Fehlermeldungen, die während des Ladevorgangs auftreten können, werden in diesem Format angezeigt. Sollten Tags zu einer Sektion vergeben worden sein, so werden diese – wie in Abbildung 39 ersichtlich – im Anzeigefenster positioniert.



Abbildung 39: Taganzeigefenster mit Tags

Die Tags werden von links nach rechts und von oben nach unten gereiht, wobei sie durch einen Beistrich getrennt sind. Sortiert werden sie zuerst nach Häufigkeit, da auch andere Benutzer gleiche Tags vergeben können. Sollte ein Tag öfters als einmal auftreten, so wird die Anzahl der Vorkommnisse hinter dem Tag in Klammern angefügt. Eine weitere Sortierung erfolgt nach der Identifikationsnummer in der Datenbank. Tags, die zuerst eingetragen wurden, erhalten automatisch eine niedrigere Identifikationsnummer und werden somit zuerst gereiht.

Wird der Mauszeiger nun vom Tagging-Button wieder entfernt, verschwindet auch das Anzeigefenster nach einigen Sekunden, sofern der Mauszeiger nicht über dem Anzeigefenster positioniert wird.

Das Anzeigefenster wird mit JavaScript erzeugt, welches in die Wiki-Seite eingebunden wurde. Dadurch lassen sich auch Effekte wie Einblendungen realisieren. Da dies alles auf der Client-Seite basiert, wird der Server nur mehr für Anfragen von Tags oder zur Speicherung von diesen benötigt. Die Kommunikation erfolgt mittels AJAX, wobei XML-Dokumente als Antwort vom Server übertragen werden. Die Wiki-Seite muss während der Benutzung der SectionTagging- und SectionRanking-Erweiterung nicht neu geladen werden.

Eine Anfrage wird über einen HTTP-Request abgesetzt und auf der Serverseite werden XML-Dokumente mit Hilfe von JSP 2.0 als Antwort generiert. Die Document Type Definition (DTD) für ein Dokument, das Tags an den Client liefert, entspricht der im Quellcode 8 dargestellten Form.

```
<!ELEMENT response (user, tags, error)>
<!ELEMENT user (#PCDATA)>
<!ELEMENT tags (tag)*>
<!ELEMENT tag (#PCDATA)>
<!ELEMENT error (#PCDATA)>
```

Quellcode 8: Document Type Definition XML-Antwort

Die Antwort enthält den Benutzernamen, die verfügbaren Tags und eine etwaige Fehlermeldung, falls die Antwortgenerierung nicht korrekt abgelaufen ist. Der Benutzername entfällt in der Antwort, wenn sich der Anwender nicht eingeloggt hat und somit anonym das Austria-Forum nutzt, so wie im Beispiel, welches im Quellcode 9 abgebildet ist.


```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE response PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://localhost:8080/JSPWiki/sectionTagging/sectiontagging.dtd">
<response>
  <user></user>
  <tags>
    <tag><![CDATA[Doppler]]></tag>
    <tag><![CDATA[Christian]]></tag>
    <tag><![CDATA[Doppler-Effekt]]></tag>
    <tag><![CDATA[Salzburg]]></tag>
    <tag><![CDATA[Prag]]></tag>
  </tags>
  <error></error>
</response>

```

Quellcode 9: XML-Antwort von STGetTags.jsp

Zusätzlich werden die Tags noch in `<![CDATA[]]>` eingefasst, um dem Parser mitzuteilen, dass sich innerhalb der Klammern Daten befinden und nicht Markup. Sonst könnte ein Tag, das die Zeichenkette „`</p>`“ enthält, zu einem Parserfehler führen. Somit ist die einzige Zeichenkette, die nicht Bestandteil eines Tags sein darf, das Datenschluss-Tag „`]]>`“. Auf diese Zeichenkette hin wird bei der Tageingabe gesucht und bei deren Auftreten eine entsprechende Fehlermeldung auf der Client-Seite ausgegeben.

5.1.3 Tagbearbeitungsfenster

Bei dem Tagbearbeitungsfenster handelt es sich um ein Fenster, das eine Eingabezeile besitzt um neue Tags zu vergeben und bestehende zu verändern. Es können aber nur jene Tags geändert werden, die von einem selbst vergeben wurden. Um dieses Fenster aufzurufen, ist ein Klick auf den Tagging-Button notwendig. Daraufhin erscheint rechts unter dem Mauszeiger das Bearbeitungsfenster (siehe Abbildung 40).



Abbildung 40: Tagbearbeitungsfenster - leer

Wurden noch keine eigenen Tags vergeben, so bleibt das Textfeld leer. Auf der rechten Seite befinden sich zwei Buttons. Der obere von beiden dient zum Schließen des Bearbeitungsfensters und damit zum Abbruch der Tageingabe. Hingegen wird der untere Button dazu benutzt die eingegebenen Tags an den Server zu schicken, wo sie gespeichert werden. Die Tags werden durch Beistriche getrennt voneinander eingegeben. Ein Leerzeichen bewirkt keine Trennung der Tags. Dadurch sind Tags wie „New York“ ohne Probleme möglich. Leerzeichen werden nur dann entfernt, wenn sie entweder direkt vor oder nach einem Beistrich vorkommen. Durch die konsequente Verwendung von UTF-8 auf ganzer Linie stellen andere Zeichensätze aus dem kyrillischen, chinesischen oder hebräischen Sprachraum kein Problem dar, wie in Abbildung 41 zu erkennen ist.

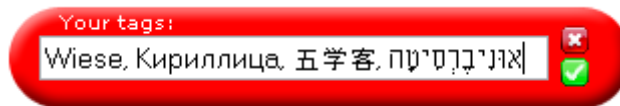


Abbildung 41: Tagbearbeitungsfenster - befüllt

Während der Speicherung werden das Eingabefeld und die Buttons deaktiviert (siehe Abbildung 42). Traten bei der Speicherung keine Fehler auf, so werden die Tags nochmals in dem Eingabefeld angezeigt. Ist jedoch ein Fehler aufgetreten, so wird die Fehlermeldung im Eingabefeld ausgegeben. Nach der Speicherung schließt sich das Bearbeitungsfenster automatisch nach ein paar Sekunden. Zur Speicherung muss nicht unbedingt der entsprechende Button betätigt werden, sondern es reicht auch wenn die Tags mit der Eingabetaste bestätigt werden.



Abbildung 42: Tagbearbeitungsfenster - Speichervorgang

Bei der Speicherung der Tags erhalten diese eine Identifikationsnummer, die mit der Position im Eingabefeld steigt. Sobald jemand seine Tags wieder verändert, werden diese aus der Datenbank gelöscht und bekommen dann bei erneuter Speicherung wieder eine von ihrer Position abhängige Identifikationsnummer zugeordnet. Somit ist gewährleistet, dass Tags, die weiter vorne im Eingabefeld stehen, auch eine niedrigere Nummer besitzen als weiter hinten stehende. Dieses Kriterium ist für die SectionRanking-Erweiterung bei der Sortierung von hoher Bedeutung.

5.2 SectionRanking

Das SectionRanking basiert auf den Tags, die mit der SectionTagging-Komponente vergeben wurden. Der Aufruf erfolgt auch aus dieser Komponente. Das SectionRanking dient zur Darstellung der Sektionen, die mit einem bestimmten Tag versehen wurden, und auch zur Neuerstellung von Seiten, die auf einer Zusammenstellung jener Sektionen beruhen. In den folgenden Unterkapiteln werden die einzelnen Bereiche dieser Komponente ausführlich erklärt.

5.2.1 Tagauswahl

Wird der Mauszeiger über ein Tag des Anzeigefensters bewegt, so verändert dieses seine Farbe, um sich optisch hervorzuheben und die Abgrenzung gegenüber den anderen Tags dem Benutzer zu verdeutlichen (siehe Abbildung 43). Durch einen Mausklick auf das gewünschte Tag wird dann das SektionRanking aktiviert.



Abbildung 43: Auswahl aus dem Taganzeigefenster

5.2.2 SectionRanking-Fenster

Wird die SectionRanking-Komponente aktiviert, so wird das Hauptfenster im JSPWiki durch diese ersetzt (siehe Abbildung 44). Je nach Größe des JSPWikis im Browserfenster passt sich auch das SectionRanking an diese Gegebenheiten an und teilt sich weiters in drei wesentliche Bestandteile auf. Ganz oben befindet sich das Optionsmenü, welches in Kapitel 5.2.3 genau beschrieben wird. Darunter folgen zwei gleich große Spalten. In der linken davon werden alle Sektionen, die das vorher ausgewählte Tag beinhalten, gelistet.

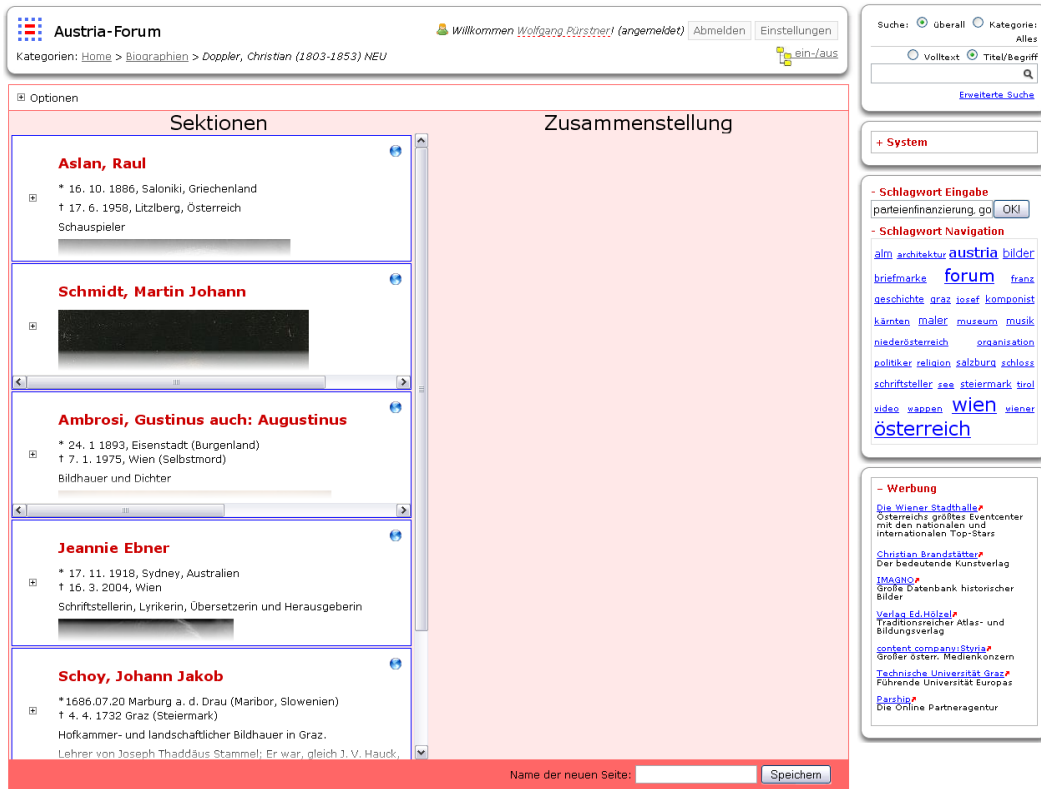


Abbildung 44: SectionRanking - Ausgangsseite

Um eine neue Wiki-Seite aus den Sektionen zu erstellen, können diese von der rechten Spalte mittels Drag & Drop (siehe Abbildung 45) in die linke Spalte gezogen werden. Die Sortierung innerhalb einer Spalte erfolgt auch mittels Drag & Drop.

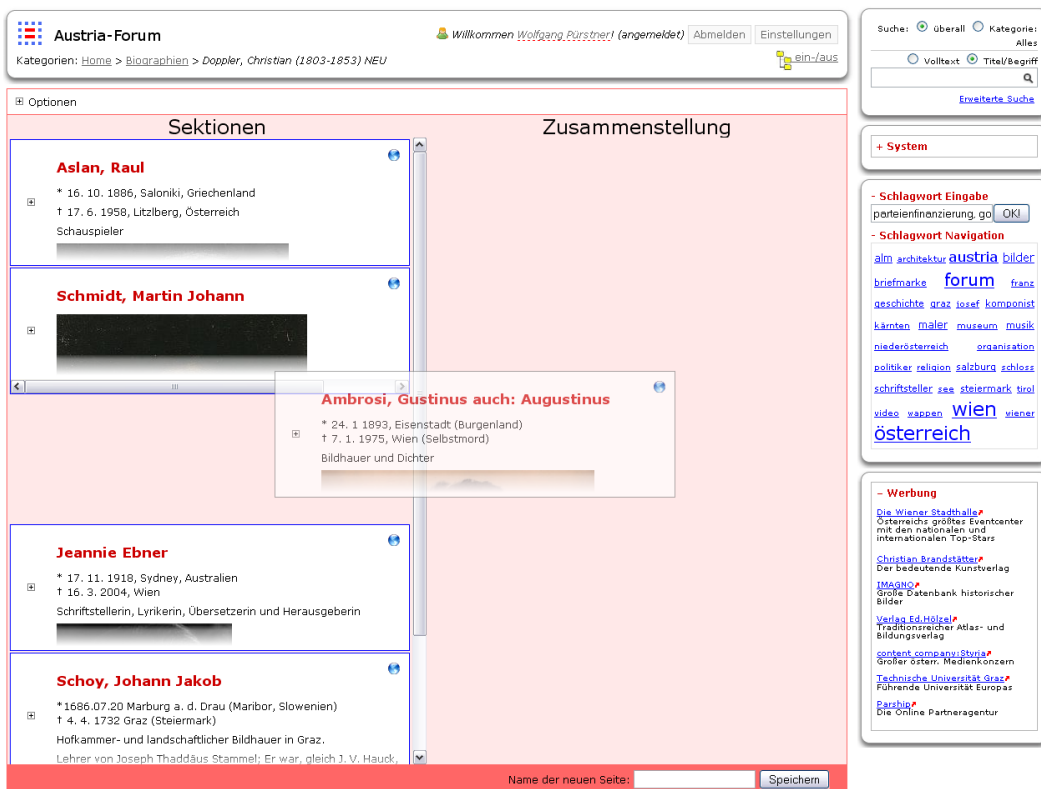


Abbildung 45: SectionRanking - Drag & Drop

Befinden sich die ausgewählten Sektionen nun in der Reihenfolge, in der man sie haben möchte, wie zum Beispiel in Abbildung 46, so kann die Seite nun gespeichert werden (siehe Kapitel 5.2.4).

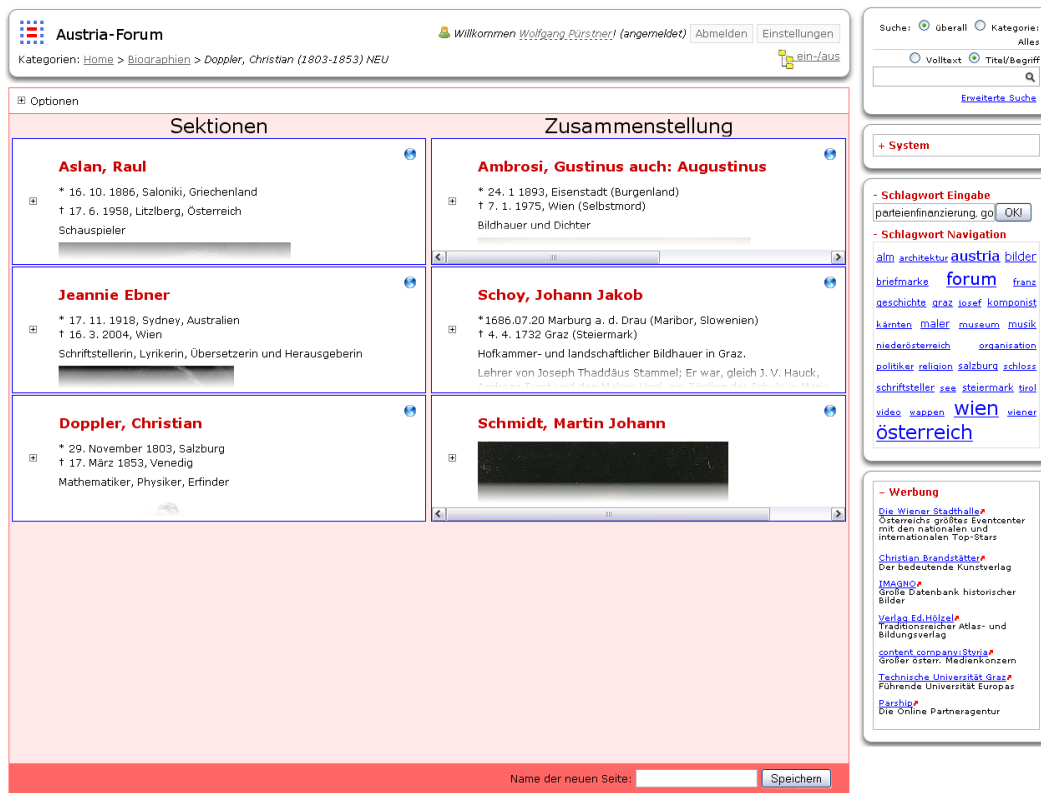


Abbildung 46: SectionRanking - Neuordnung

Die einzelnen Sektionen werden in einem kompakten Format dargestellt, um eine leichte Neuordnung zu gewährleisten. Der per Drag & Drop verschiebbare Rahmen besteht aus einem Aufklappsymbol in der Mitte des linken Randes, einem Direktlink zur Sektion in der linken oberen Ecke und der Sektion selbst, welche nach unten hin zum Rand bei Überlänge ausgeblendet wird (siehe Abbildung 47). Um die Sektion vollständig einsehen zu können, kann der Sektionsrahmen aufgeklappt werden (siehe Abbildung 48).



Abbildung 47: Rahmen für Drag & Drop (eingeklappt)

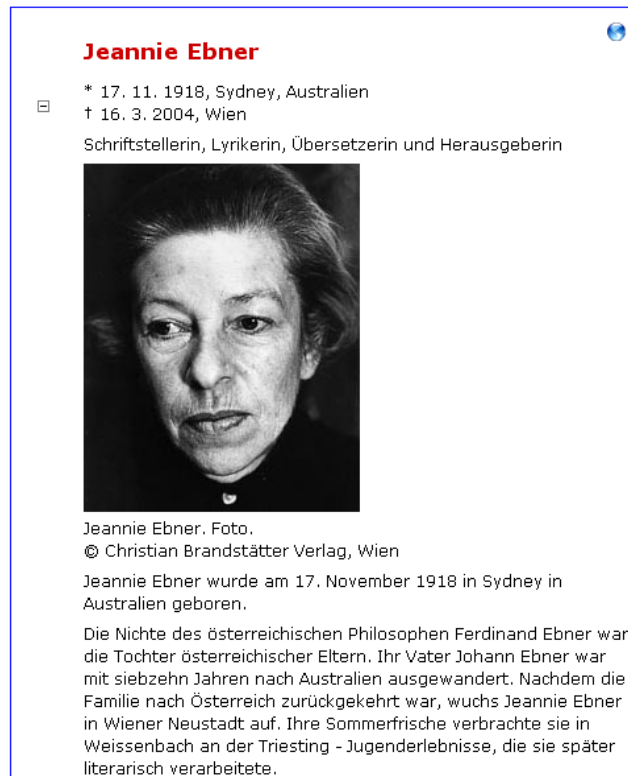


Abbildung 48: Rahmen für Drag & Drop (aufgeklappt)

5.2.3 Optionsmenü

Im Optionsmenü werden Einstellungen vorgenommen, die das SectionRanking betreffen. Um diese zu verändern, muss zuerst das Plus-Symbol betätigt werden, um das Optionsmenü aufzuklappen. Um es wieder zu schließen, genügt ein Klick auf das Minus-Symbol im geöffneten Optionsmenü. Vorerst kann nur die Anzahl der dargestellten Ergebnisse eingegrenzt werden. Mit dem Button „Suchen“ werden die Sektionen in der ersten Spalte aktualisiert. In der Abbildung 49 ist das geschlossene Optionsmenü dargestellt und in Abbildung 50 das geöffnete Menü.



Abbildung 49: Optionsmenü (geschlossen)

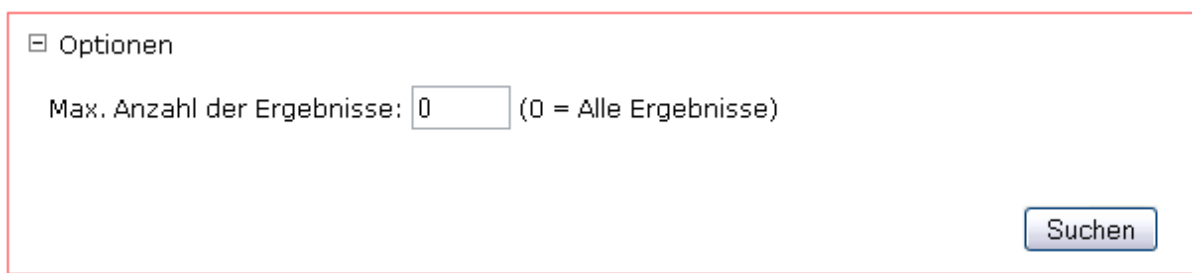


Abbildung 50: Optionsmenü (geöffnet)

5.2.4 Neue Wiki-Seite aus Sektionen erstellen

Dazu wird ein noch nicht vergebenen Wiki-Seitenname unter den Spalten in der in Abbildung 51 dargestellten Textbox eingetragen und der Button „Speichern“ betätigt. Daraufhin wird die neue Wiki-Seite im Edit-Modus geöffnet, um weitere Änderungen durchführen zu können. Sollte die Wiki-Seite schon existieren, so wird eine entsprechende Fehlermeldung ausgegeben.

The image shows a red-bordered box containing a text input field with the placeholder text 'Name der neuen Seite:' and a blue button labeled 'Speichern' to its right.

Abbildung 51: Wiki-Seite Speicherung

Nach dem Speichern wird die Seite im Bearbeitungsmodus (siehe Abbildung 52) geöffnet, um weitere Änderungen durchführen zu können. Die Seite wird standardmäßig in der Hauptkategorie abgelegt, kann aber jedoch in eine andere Kategorie verschoben werden. Zusätzlich werden noch alle Dateianhänge der Sektionen, wie zum Beispiel Bilder, auf die neue Wiki-Seite kopiert, damit diese dort auch verfügbar sind und verändert werden können.

The image shows the Wiki edit interface. At the top, there are buttons for 'Bearbeiten', 'Hilfe', and 'Weitere...'. Below these are 'Speichern', 'Vorschau', and 'Abbrechen'. A toolbar contains various editing tools like bold, italic, link, and undo/redo. Below the toolbar is a text area with the following content:

Lehrer von Joseph Thaddäus Stammel; Er war, gleich J. V. Hauck, Andreas Trost und den Malern Vogl, ein Zögling der Schule in Maria Rast. Über seinen künstlerischen Bildungsgang und seine sonstigen Lebensverhältnisse ist nichts bekannt. Dass er ein Künstler von grosser Begabung war, zeigen seine Werke. Er war der "führende Grazer und steirische Bildhauer in der Ersten Hälfte des 18. Jahrh." (Vrizer). "Mit Marx Schokotnigg hatte die Provinz Untersteiermark, mit Johann Jakob Schoy ihre Hauptstadt Marburg künstlerisch von der steirischen Hauptstadt Graz Besitz ergriffen. Ging ein Großteil des Ruhmes Schokotniggs auf effektvolle Pose, wenn nicht gar Theatralik zurück, so erklärt sich die tiefe nachhaltige Wirkung der Werke Schoys aus seiner beseelten Innigkeit, seinem künstlerischen Ernst, seiner echten Religiosität" (Kohlbach, Bildh., 179). Er wäre bei seinen Talenten vielleicht ein Plastiker ersten Ranges geworden, wenn er in einer Stadt mit grösseren künstlerischen Anregungen gelebt hätte, als Graz sie bieten konnte. Seine hinterlassene Witwe, Anna Katharina, heiratete am 18. September 1733 dessen Schüler Philipp Jakob Straub.

 Below the text area are 'Speichern', 'Vorschau', and 'Abbrechen' buttons. A checkbox for 'Schnellvorschau' is visible. At the bottom, there is a status bar indicating the page was last edited on 'Montag, 28. Juni 2010, 09:27' by 'Wolfgang Pürstner' and a URL: 'http://localhost:8080/af/af/K%C3%BCnstler-Biographien'.

Abbildung 52: Bearbeitungsfenster einer Wiki-Seite

5.3 Verwendete Techniken

Für die Implementierung wurden verschiedene Techniken eingesetzt. Der SectionTagging-Button wurde mittels Plugin- und Filtersystem des JSPWikis in die Wiki-Seite integriert, das heißt es wurde somit die Programmiersprache Java verwendet. Das Taganzeigefenster und das Tagbearbeitungsfenster sowie das SectionRanking wurde vollkommen in JavaScript geschrieben, wobei die Darstellung in Cascading Style Sheets definiert wurde. Anfragen an den Server werden mit AJAX gelöst und die Antworten werden im XML-Format geliefert. Am Server werden diese Antworten von JavaServer Pages generiert, die wiederum Java-Komponenten aufrufen um die benötigten Informationen zu bekommen. Einige dieser Technologien wurden schon in vorangegangenen Kapiteln erläutert oder kurz erwähnt (siehe JSP – Kapitel 3.3.1). In diesem Kapitel wird auf einige weitere Techniken eingegangen, die für die Erweiterung maßgeblich sind.

5.3.1 JavaScript

Hierbei handelt es sich um eine Skriptsprache, die hauptsächlich clientseitig ausgeführt wird. Mit Skriptbefehlen lassen sich Webseiten dynamisch verändern und somit auch das Document Object Model (DOM), welches die Webseite als Baumstruktur darstellt. Datenübertragungen lassen sich mit Hilfe von AJAX (siehe Kapitel 5.3.4) realisieren.

So wurde clientseitig die SectionTagging- und SectionRanking-Erweiterung komplett mit JavaScript programmiert, um den Server nur für die speziellen Abfragen zu nutzen und ihn nicht zusätzlich belasten zu müssen. Plausibilitätsprüfungen finden daher schon im Browser des Anwenders statt.

5.3.2 Cascading Style Sheets

In den frühen Anfängen der Webseitengestaltung wurden alle Informationen einer Seite, wie Inhalt, Struktur und Darstellung, in ein Dokument gepackt. Dies machte es natürlich schwierig bestimmte visuelle Änderungen einfach durchzuführen. Cascading Style Sheets erlauben es hingegen, die Darstellung von bestimmten Elementen in HTML oder XML-Dokumenten zu definieren. Somit wird der Inhalt von der Darstellung getrennt. Neben Farben, Umrandungen, Schriftgrößen, Positionen und Hintergrundbildern gibt es noch zahlreiche Einstellungen, die das Aussehen von Elementen definieren können. Darüber hinaus kann die Darstellung des Inhalts auf verschiedenen Ausgabemedien festgelegt werden. Für mobile Geräte kann auf diese Weise zum Beispiel die Seitenbreite verringert werden oder für den Ausdruck einer Webseite kann der Entfall des Hintergrundbildes nützlich sein.

Für das Taganzeigefenster, Tagbearbeitungsfenster und SectionRanking-Modul werden CSS-Einstellungen zentral in einer Datei „sectiontagging.css“ zusammengefasst. Damit lassen sich Designänderungen schnell und effizient durchführen, ohne den Quellcode der Erweiterung zu verändern. Der Quellcode 10 zeigt die CSS-Definition des Rahmens, in der eine Sektion eingebettet ist, welcher dann beim SectionRanking per Drag & Drop verschoben werden kann.


```
.st_dragelement {
  height:          150px;
  border:          thin blue solid;
  margin-top:      2px;
  margin-bottom:   2px;
  margin-left:     3px;
  margin-right:    3px;
  overflow:        hidden;
  position:        relative;
  background-color: white;
  overflow-x:      auto;
}
```

Quellcode 10: CSS-Definition einer Dragbox (Sektion)

5.3.3 Extensible Markup Language

Extensible Markup Language (XML)-Dateien werden nur vom Server als Antwort für den Client generiert. Der Client selbst schickt seine Daten als Parameter im GET-Request an den Server. Zur Generierung der XML-Dateien werden JavaServer Pages in der Version 2 benutzt, da mit Hilfe dieser Version sich wohlgeformte XML-Dokumente mit eindeutigen Namensräumen erzeugen lassen. Im Grunde genommen sind alle Serverantworten der Erweiterung, also XML-Dokumente, wie im Quellcode 11 aufgebaut. Zuerst wird eine XML-Deklaration verwendet, die die XML-Version, die Kodierung und eine optionale Dokumententypdefinition enthält. Als Wurzelement fungiert der Knoten „response“, weil es sich bei den XML-Dokumenten ausschließlich um Antworten handelt. Als Kindsknoten sind dann die jeweiligen Daten enthalten, die der Client angefragt hat. Zusätzlich existiert in jeder Antwort ein Kindsknoten mit dem Namen „error“, in dem aufgetretene Fehlermeldungen an den Client übertragen werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  ...DATEN...
  <error></error>
</response>
```

Quellcode 11: Grundgerüst einer XML-Antwort

5.3.4 Asynchronous JavaScript and XML

Asynchronous JavaScript and XML (AJAX) kombiniert verschiedene Technologien, wie zum Beispiel JavaScript und XML, die schon in der eigentlichen Bezeichnung enthalten sind. AJAX wird dazu benutzt um Datenübertragungen zwischen einem Client, beziehungsweise Browser, und einem Server durchzuführen ohne die ganze Webseite neu laden zu müssen. Dazu werden HTTPRequests (GET und POST-Methode, siehe Kapitel 3.2.1) synchron oder asynchron abgesetzt und dann bearbeitet. In Abbildung 53 ist ein Vergleich zwischen einer traditionellen Webanwendung und einer Anwendung mit AJAX zu sehen.

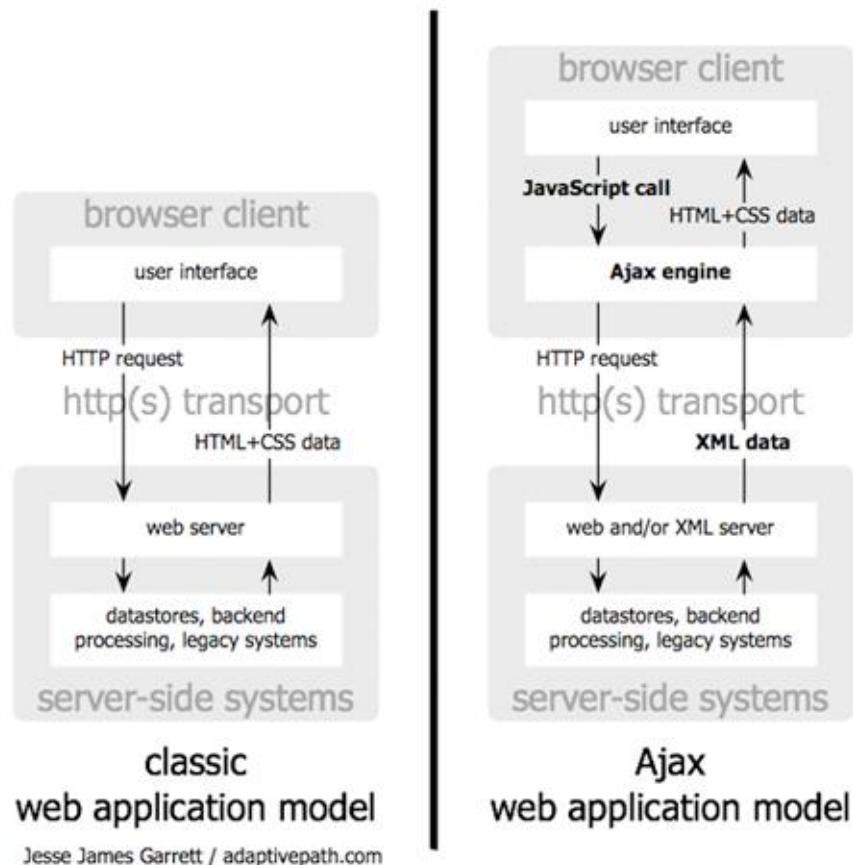


Abbildung 53: Vergleich traditionelle Webanwendung (links) zu einer mit AJAX (rechts) [Garret 2005]

Alle Anfragen und Antworten für das SectionTagging-Modul und für die SectionRanking-Komponente basieren auf AJAX. Sollen zum Beispiel die Tags zu einer Sektion angezeigt werden und es wird der Mauszeiger über dem Tagging-Button platziert, so wird ein AJAX-Request abgesetzt, der die benötigten Tags vom Server anfordert. Diese werden dann im Taganzeigefenster dargestellt ohne die ganze Seite neu laden zu müssen.

5.3.5 Yahoo! User Interface Library

Die Yahoo! User Interface Library (YUI Library) ist eine quelloffene JavaScript-Bibliothek, die umfangreiche Funktionen für Webapplikationen bereitstellt. Damit lassen sich auf einfache Weise interaktive Applikationen mit AJAX entwickeln. Über zusätzliche Konstruktionen ist es auch möglich, Objektorientierte Programmierung (OOP) anzuwenden. Einfache Komponenten wie Kalender, Schaltflächen, Animationen oder Menüs sind ebenso enthalten wie ein umfangreicher Rich-Text Editor. Besonders sticht die Modularität von YUI hervor. So muss nicht das ganze Framework in die Seite eingebunden werden, sondern es werden nur die Bestandteile verwendet, die auch wirklich benötigt werden. Dies verringert den Ladeaufwand erheblich, da von der YUI-Bibliothek nur die essentiellen Teile nachgeladen werden. [Steyer 2008]

Für die Drag & Drop Funktionalität des SectionRankings wird die YUI-Bibliothek genutzt. Damit ist es möglich, DIV-Container, in diesem Fall Sektionen, zwischen den Spalten hin und her zu schieben und diese auch untereinander zu ordnen. Dabei werden diverse Animationen und Effekte benutzt. Beim Anheben („Drag“) der Sektion wird diese leicht transparent dargestellt und beim Loslassen („Drop“) derselben gleitet diese animiert an ihren Bestimmungsort, wobei etwaige platzversperrende Sektionen verschoben werden.

5.3.6 Prepared Statements

Für den Zugriff auf die Datenbank werden ausschließlich Prepared Statements verwendet. Dabei handelt es sich um vordefinierte Anweisungen für ein Datenbanksystem, um SQL-Einschleusungen durch einen Angreifer zu verhindern. Variablen werden dann als Parameter der vorkompilierten Anweisung übergeben und auf ihre Gültigkeit hin überprüft. Im Quellcode 12 ist eine SQL-Anweisung definiert, die alle Tags zu einer bestimmten URL aufsteigend nach der Identifikationsnummer liefert. Da nur die URL variabel ist, wird an ihrer Stelle ein Fragezeichen als Platzhalter verwendet.

```
private static final String GET_TAGS_BY_URL =
    "SELECT tag " +
    "FROM sc_tags INNER JOIN sc_bookmarks " +
    "ON sc_tags.bID=sc_bookmarks.bId " +
    "WHERE bAddress=? " +
    "ORDER BY sc_tags.id ASC";
```

Quellcode 12: Prepared Statement – String (Java)

Die URL selbst wird mit der Funktion „setString()“ eingefügt, wobei der erste Parameter die Nummer des Platzhalters angibt. Sind mehrere Platzhalter in der SQL-Anweisung vorhanden, so bestimmt sich die Nummer anhand der Reihenfolge des Fragezeichens in der Anweisung. Im Quellcode 13 ist solch ein exemplarischer Aufruf dargestellt.

```
PreparedStatement pstmt = connection.prepareStatement( GET_TAGS_BY_URL );
pstmt.setString( 1, url );
ResultSet result = pstmt.executeQuery();
```

Quellcode 13: Prepared Statement – Ausführung (Java)

5.4 Ablauf

In diesem Unterkapitel wird der Ablauf zwischen Client und Server anhand einiger Beispiele grafisch dargestellt. Dabei werden zwei Abläufe des SectionTaggings und ein Ablauf des SectionRankings näher betrachtet.

5.4.1 SectionTagging

In Abbildung 54 ist die Anforderung einer Wiki-Seite an den Server dargestellt, in der auch die SectionTagging-Komponente in die Seite integriert wird.

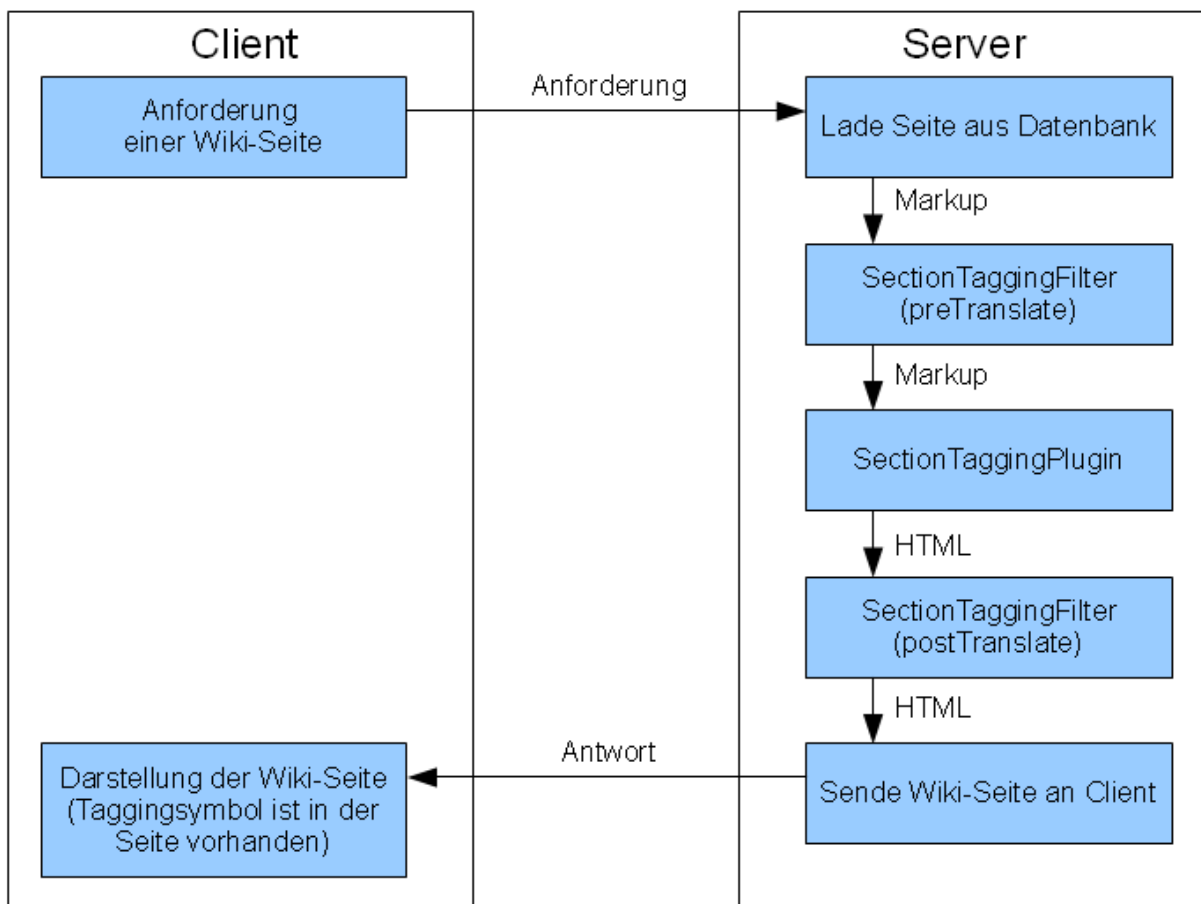


Abbildung 54: Ablauf - Wiki-Seitengenerierung

In Abbildung 55 wird veranschaulicht, wie Tags aus der Datenbank geladen werden und anschließend in der Tagbox ausgegeben werden. Zu beachten ist, dass die Tags erst bei Anforderung des Benutzers geladen werden und nicht schon in die Wiki-Seite integriert sind.

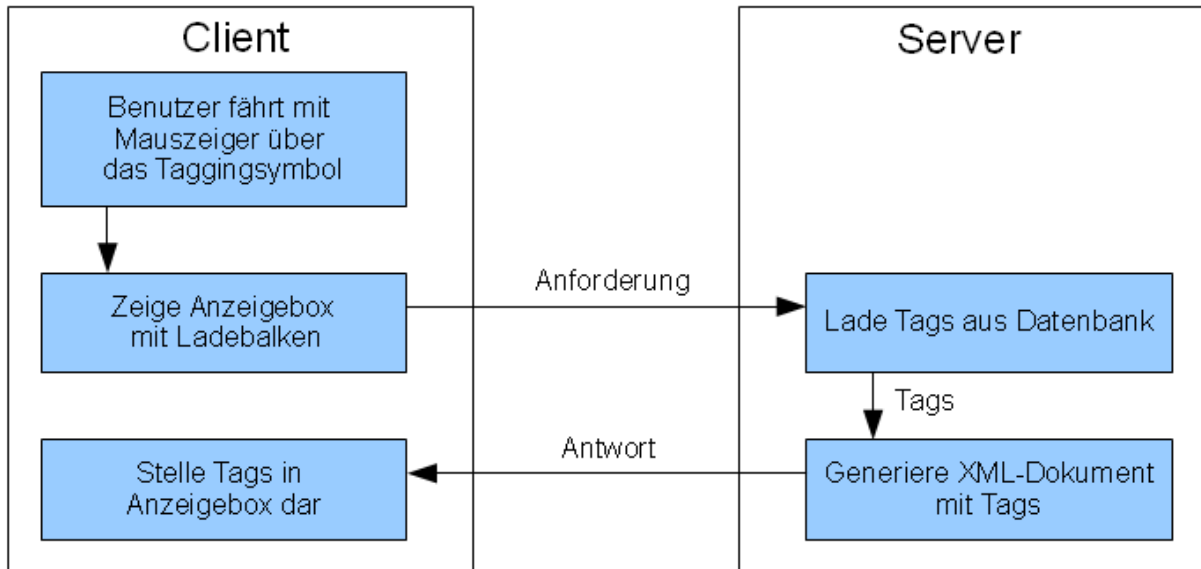


Abbildung 55: Ablauf - Tagsanforderung

5.4.2 SectionRanking

Beim Start der SectionRanking-Komponente durch einen Klick auf ein Tag in dem Taganzeigefenster wird bei der Anforderung an den Server das Tag mitgeschickt. Dort werden zuerst nur die Links zu den Sektionen zurück übertragen und der Client fordert dann jede Sektion einzeln an um diese darzustellen (siehe Abbildung 56).

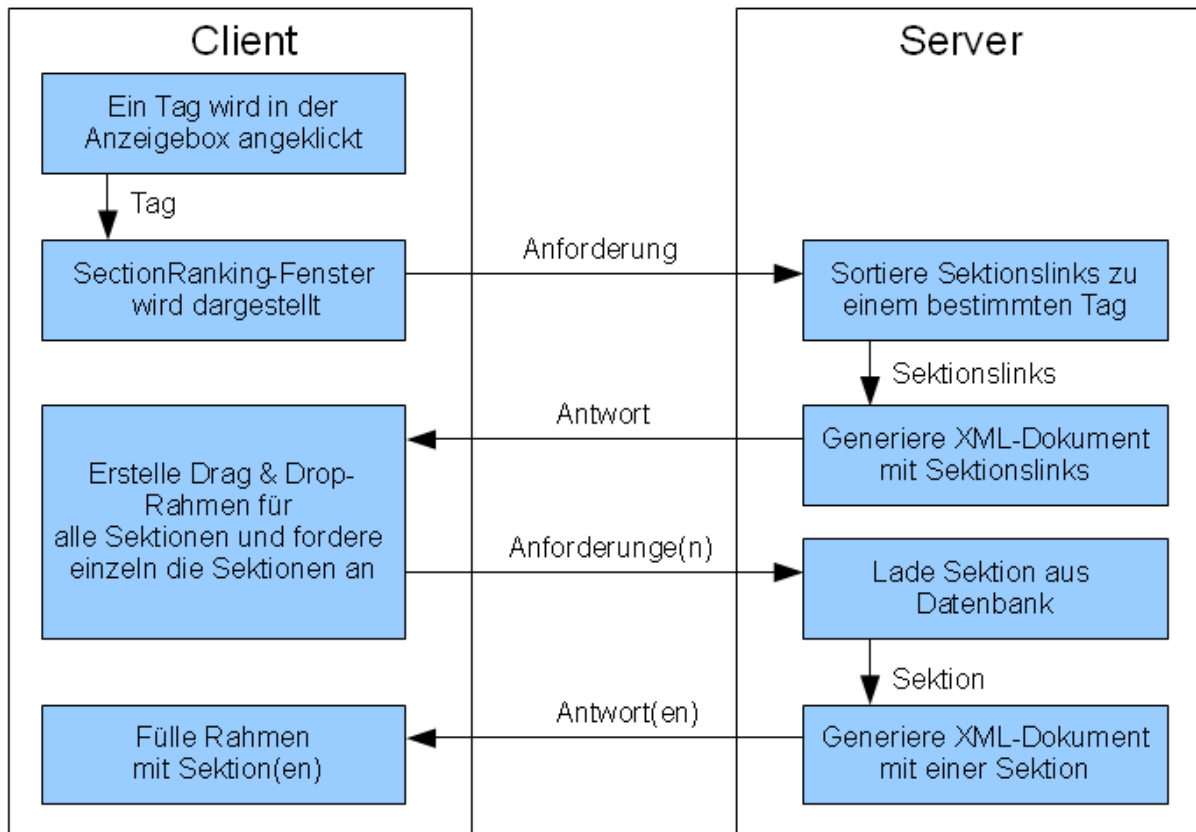


Abbildung 56: Ablauf - SectionRanking

5.5 Installation

Die Anleitung dient zur Installation ins bestehende Austria-Forum, da die Erweiterung speziell für dieses entwickelt wurde. Diese Anleitung kann jedoch auch analog für andere JSPWikis verwendet werden. Die Installationsanweisungen für Scuttle und JSPWiki für das Austria-Forum selbst sind in [Zöhrer 2009] und in Kapitel 3.3.5 beschrieben. Darum wird hier für die nächsten Anweisungen schon von einem lauffähigen Austria-Forum ausgegangen:

5.5.1 Systemanforderungen

Die Systemanforderungen werden wie folgt auf der Serverseite festgelegt:

- JSPWiki 2.8.3 mit Austria-Forum-Anpassung
- Apache Tomcat 6.0 oder gleichwertig
- TagCloudPlugin 1.0
- Scuttle 0.7.4
- Xerxes 1.4.4
- MySQL-JDBC Connecto 5.1.7

Für die Clientseite wird ein aktueller Browser benötigt. Es werden folgende Varianten empfohlen.

- Firefox 3 oder höher
- Chrome 4 oder höher
- Safari 4 oder höher
- Internet Explorer 8 oder höher

5.5.2 Dateistruktur

Da die SectionTagging-Komponente mit der SectionRanking-Komponente eng verbunden ist teilen sie sich auch die Dateien. Diese lassen sich daher nicht einem bestimmten Teil zuordnen. In der Tabelle 17 sind die Dateien der Erweiterung gelistet mit der dazugehörigen Funktionsbeschreibung.

Typ	Dateiname	Funktion
Java	ConfigReader.java	Dient zum Auslesen der Konfigurationsdatei
Java	Database.java	Führt die gesamten Anfragen in der Datenbank durch
XML	sectiontagging.conf.xml	Konfigurationsdatei
CSS	sectiontagging.css	Enthält die CSS-Informationen der Erweiterung
DTD	sectiontagging.dtd	Beinhaltet die Document Type Definition
Java	SectionTagging.java	Programmlogik – Server
JS	sectiontagging.js	Programmlogik – Client
Java	SectionTaggingFilter.java	Filter für das JSPWiki
Java	SectionTaggingPlugin.java	Plugin für das JSPWiki
Bild	st_cancel.png	Abbruchbutton
Bild	st_editbox.png	Editbox
Bild	st_fading.png	Überblendung
Bild	st_loading.gif	Ladestatus
Bild	st_minus.png	Zuklappsymbol
Bild	st_ok.png	OK-Button
Bild	st_plus.png	Aufklappsymbol
Bild	st_tag.png	Tagsymbol
Bild	st_tagbox_bottom.png	Tagbox Umrandung
Bild	st_tagbox_top.png	Tagbox Umrandung
Bild	st_urlbutton.png	Direktlink
JSP	STGetPageStatus.jsp	Gibt den Verfügbarkeitsstatus einer Seite zurück
JSP	STGetRanking.jsp	Liefert die Sektionslinks gereiht zu einem Tag
JSP	STGetSection.jsp	Liefert eine bestimmte Sektion
JSP	STGetTags.jsp	Liefert die Tags zu einer Sektion
JSP	STSetTags.jsp	Speichert die Tags zu einer Sektion
TXT	Anleitung_Edit.text	Enthält den Code, der in die Edit.jsp eingefügt wird

Tabelle 17: Dateistruktur

Diese Dateien müssen dann in den jeweiligen Ordner kopiert werden, um die Kompilierung einer neuen Version zu ermöglichen. JSPWiki bezeichnet in der Tabelle 18 den Basisordner, der den Quellcode enthält.

Dateiname	Ordner
ConfigReader.java	JSPWiki\src\SectionTagging\
Database.java	JSPWiki\src\SectionTagging\
sectiontagging.conf.xml	JSPWiki\etc\
sectiontagging.css	JSPWiki\src\webdocs\templates\default\
sectiontagging.dtd	JSPWiki\src\SectionTagging\
SectionTagging.java	JSPWiki\src\SectionTagging\
sectiontagging.js	JSPWiki\src\webdocs\scripts\
SectionTaggingFilter.java	JSPWiki\src\com\ecryd\jspwiki\filters\
SectionTaggingPlugin.java	JSPWiki\src\com\ecryd\jspwiki\plugin\
st_cancel.png	JSPWiki\src\webdocs\templates\default\images\
st_editbox.png	JSPWiki\src\webdocs\templates\default\images\
st_fading.png	JSPWiki\src\webdocs\templates\default\images\
st_loading.gif	JSPWiki\src\webdocs\templates\default\images\
st_minus.png	JSPWiki\src\webdocs\templates\default\images\
st_ok.png	JSPWiki\src\webdocs\templates\default\images\
st_plus.png	JSPWiki\src\webdocs\templates\default\images\
st_tag.png	JSPWiki\src\webdocs\templates\default\images\
st_tagbox_bottom.png	JSPWiki\src\webdocs\templates\default\images\
st_tagbox_top.png	JSPWiki\src\webdocs\templates\default\images\
st_urlbutton.png	JSPWiki\src\webdocs\templates\default\images\
STGetPageStatus.jsp	JSPWiki\src\webdocs\
STGetRanking.jsp	JSPWiki\src\webdocs\
STGetSection.jsp	JSPWiki\src\webdocs\
STGetTags.jsp	JSPWiki\src\webdocs\
STSetTags.jsp	JSPWiki\src\webdocs\

Tabelle 18: Installationsverzeichnisse

Zusätzlich werden noch folgende Libraries benötigt, um die Erweiterung lauffähig zu machen. Wird das vorher erwähnte TagCloudPlugin bereits eingesetzt, so sind diese Bibliotheken schon verfügbar und müssen nicht erst installiert werden. Bei einem Fehlen der Bibliotheken müssen die einzelnen jar-Dateien in den Ordner „JSPWiki\lib“ kopiert werden. Sollte schon eine Xerxes-Parser-Library vorhanden sein, kann es zu Inkompatibilitäten kommen, die sich durch Parser-Fehler beim SectionTagging bemerkbar machen.

Library	Ordner
mysql-connector-java-5.1.12-bin.jar	JDBC – MYSQL Connector ⁷⁷ (Version 5.1.12)
resolver.jar	Xerxes – Java ⁷⁸ (Version 1.4.4)
serializer.jar	Serializer ⁷⁹ (Version 2.7.1)
xml-apis.jar	Xalan – Java ⁸⁰ (Version 2.7.1)

Tabelle 19: Bibliotheken

⁷⁷ <http://www.mysql.com/downloads/connector/j/> (28.06.2010)

⁷⁸ <http://xerces.apache.org/xerces-j/> (28.06.2010)

⁷⁹ <http://www.java2s.com/Code/Jar/JKL/Downloadserializerjar.htm> (28.06.2010)

⁸⁰ <http://xml.apache.org/xalan-j/downloads.html> (28.06.2010)

Die Yahoo! User Interface Library – bestehend aus dem Ordner „yui“ – wird in den Skript-Ordner „JSPWiki\src\webdocs\scripts\“ kopiert. Die zu verwendende Version 2 steht auf der Entwicklerseite⁸¹ von Yahoo! zum Download bereit.

5.5.3 JSPWiki Anpassungen

Um die YUI-Library, eigene JavaScript- und CSS-Dateien für die SectionTagging- und SectionRanking-Komponente zugänglich zu machen, werden diese ins JSPWiki-Template integriert. Dazu werden in der Datei „commonheader.js“, welche sich im Ordner „JSPWiki\src\webdocs\templates\“ befindet, folgende Änderungen vorgenommen.

Unter der Zeile, die in Quellcode 14 ersichtlich ist, wird Quellcode 15 eingefügt.

```
<wiki:IncludeResources type="stylesheet"/>
```

Quellcode 14: CSS-Markierung

```
<link rel="stylesheet" type="text/css"
  href="<wiki:Link format='url' templatefile='sectiontagging.css' />" />
```

Quellcode 15: Verweis auf CSS-Datei

Und unter der Zeile, die in Quellcode 16 steht, werden die Zeilen vom Quellcode 17 eingefügt. Dadurch ist der Zugriff auf diese Dateien von jeder Wiki-Seite aus möglich.

```
<wiki:IncludeResources type="script"/>
```

Quellcode 16: JS-Markierung

```
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/yui/build/yahoo/yahoo-min.js' />"></script>
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/yui/build/dom/dom-min.js' />"></script>
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/yui/build/event/event-min.js' />"></script>
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/yui/build/animation/animation-min.js' />"></script>
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/yui/build/dragdrop/dragdrop-min.js' />"></script>
<script type="text/javascript" src="<wiki:Link format='url'
jsp='scripts/sectiontagging.js' />"></script>
```

Quellcode 17: Verweis auf JS-Dateien

⁸¹ <http://developer.yahoo.com/yui/2/> (28.06.2010)

Für die SectionRanking-Komponente muss noch eine Änderung der Edit.jsp durchgeführt werden. Zusätzlich muss, um das Bearbeitungsfenster nach der Speicherung einer neuen Zusammenstellung mit den gewählten Sektionen zu füllen, unter der Zeile vom Quellcode 18 der Abschnitt aus Quellcode 19 eingefügt werden. Dieser Quellcode befindet sich auch in der Datei „Anleitung_Edit.txt“ der Installation. Die Zeilen im Quellcode 20 werden am Anfang bei den Importierungen angefügt.

```
if( ok != null || captcha != null )
```

Quellcode 18: Markierung in der Edit.jsp

```
String sectionEditing = request.getParameter("sectiontagging");
if (sectionEditing != null)
{
    try {
        if(wiki.pageExists(pagereq) == false)
        {
            SectionTagging tagging = new SectionTagging();
            String[] list = sectionEditing.split(",");
            String markup = "";
            ArrayList<Attachment> wholeAttachmentList = new
                ArrayList<Attachment>();
            for (int i = 0; i < list.length; i++)
            {
                markup += tagging.getSection(list[i], wiki, "WIKIMARKUP");
                ArrayList attachmentList =
                    tagging.getAttachmentListOfLastSection();
                while (attachmentList.size() > 0)
                    wholeAttachmentList.add((Attachment)attachmentList.remove(0));
            }
            wiki.saveText(wikiContext, markup);
            // Copy attachments of the old section to the new page
            for (int i = 0; i < wholeAttachmentList.size(); i++)
            {
                Attachment att = wholeAttachmentList.get( i );
                Attachment newAttachment = new Attachment( wiki, pagereq,
                    att.getFileName() );
                wiki.getAttachmentManager().storeAttachment( newAttachment,
                    wiki.getAttachmentManager().getAttachmentStream( att ) );
            }
            ok = null; // don't enter one of the branches below
            captcha = null;
            preview = null;
            cancel = null;
        } else {
            log.debug("Can't overwrite existing page = " + pagereq +
                "(SectionTagging)");
        }
    } catch (Exception e) {
        log.debug("Error in Edit.jsp (Sectiontagging) " + e.getMessage());
        System.out.println("Error in Edit.jsp (Sectiontagging) " +
            e.getMessage());
    }
}
```

Quellcode 19: Zusatzcode der Edit.jsp

```
<%@ page import="com.ecyrd.jspwiki.attachment.Attachment" %>
<%@ page import="sectionTagging.SectionTagging" %>
```

Quellcode 20: Zusätzliche Datei-Importierungen der Edit.jsp

In der SectionTagging.java ist die Konstante URLPREFIX zu setzen, die den ShortUrlConstructor beinhaltet. Normalerweise ist diese „wiki/“. Für das Austria-Forum jedoch lautet diese „af/“ und ist standardmäßig in den Installationsdateien auch so vergeben.

5.5.4 Filter aktivieren

Um die SectionTagging-Komponente zu aktivieren, muss der entsprechende Eintrag in der Konfigurationsdatei „filters.xml“ vom JSPWiki wie im Quellcode 21 gesetzt werden.

```
<?xml version="1.0"?>
<pagefilters>
  ....
  <filter>
    <class>com.ecyrd.jspwiki.filters.SectionTaggingFilter</class>
  </filter>
  ....
</pagefilters>
```

Quellcode 21: Aktivierung des Filters

5.5.5 Kodierung

Da auf ganzer Linie UTF-8 verwendet wird, muss gewährleistet sein, dass dieses auch von jedem Bereich unterstützt wird. So muss der Webserver UTF-8 verarbeiten können. Der Apache Tomcat wird zum Beispiel mit der Zeile im Quellcode 22 auf UTF-8 eingestellt, welche in der Datei „server.xml“ im Konfigurationsordner des Webserver gesetzt wird.

```
<Connector port="8080" URIEncoding="UTF-8"/>
```

Quellcode 22: Tomcat UTF-8 Einstellung

Es ist auch darauf zu achten, dass die einzelnen Tabellen der Scuttle-Datenbank auf „utf8_general_ci“ gestellt sind, um auch eine Speicherung in UTF-8 zu garantieren.

5.5.6 Konfigurationsdatei

Die Konfigurationsdatei befindet sich im WEB-INF-Ordner am Webserver. In dieser Datei können alle Einstellungen vorgenommen werden, die die Erweiterung betreffen. Es werden die Zugriffsdaten für die Scuttle-Datenbank gesetzt, Werte für das SectionRanking vergeben, und Fehlermeldungen in der jeweiligen Sprache definiert. Dabei werden die Fehlermeldungen unterschieden in welche, die in den Log-Dateien des Webserver ausgegeben werden und andere, die der Anwender in der Benutzeroberfläche angezeigt bekommt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties [<!ELEMENT properties (entry*) ><!ELEMENT entry (#PCDATA) ><!ATTLIST entry key CDATA #REQUIRED]>
<properties>

  <!-- TEST ENTRY TO CHECK IF CONFIG IS READ CORRECTLY-->
  <entry key="test">test</entry>

  <!-- ALLOW ANONYMOUS USERS TO TAG HEADINGS -->
  <entry key="config.anonymousallowed">>false</entry>

  <!-- DATABASE CONNECTION -->
  <entry key="database.server">localhost</entry>
  <entry key="database.name">scuttle</entry>
  <entry key="database.user">root</entry>
  <entry key="database.password"></entry>

  <!-- RANKING -->
  <entry key="ranking.maxpositionpoints">100</entry> <!-- early position in tag list -> more points -->
  <entry key="ranking.maxnumberpoints">100</entry> <!-- less tags -> more points -->

  <!-- ERROR MESSAGES SYSTEM -->
  <entry key="error.database.initialize">[SECTIONTAGGING] Error: Initialising database</entry>
  <entry key="error.database.connect">[SECTIONTAGGING] Error: connecting database</entry>
  <entry key="error.filter.noheadingend">[SECTIONTAGGING] Error: There is no heading-end before the plugin</entry>
  <entry key="error.filter.nolink">[SECTIONTAGGING] Error: There is no link before the plugin</entry>

  <!-- ERROR MESSAGES GUI -->
  <entry key="error.cantconnectdb">Can't connect to database</entry>
  <entry key="error.cantquerydb">Can't query the database</entry>
  <entry key="error.cantsplittag">Can't split tag-string</entry>
  <entry key="error.tagistolong">Tag is to long (MAX=32)</entry>
  <entry key="error.sametagtwice">Same tag occured twice</entry>
  <entry key="error.youhavetobloggedin">You have to be logged in</entry>
  <entry key="error.notaginparaurl">No tag in parameter url</entry>
  <entry key="error.nodelimiterfound">No section delimiter found</entry>
  <entry key="error.nowikinamefound">No wiki name found</entry>
  <entry key="error.nobeginsectionfound">Begin of section in url not found</entry>
  <entry key="error.sectionisnowikipage">Section-page is not wiki-page</entry>
  <entry key="error.pagedoesntexist">Page doesn't exist: </entry>
  <entry key="error.sectionnotfound1">Section </entry>
  <entry key="error.sectionnotfound2"> not found</entry>
  <entry key="error.noformatselected">Section format not selected (HTML oder WIKIMARKUP)</entry>

</properties>
```

Abbildung 57: Konfigurationsdatei

Wie in Abbildung 57 ersichtlich existiert auch ein Eintrag mit dem Namen „config.anonymousallowed“. Standardmäßig ist dieser auf „false“ gesetzt. Wird dieser Eintrag jedoch mit „true“ aktiviert, so können auch anonyme Benutzer des Austria-Forums Tags vergeben. Für die Tagzuweisung zu einem Benutzer wird ein anonymer Account in der Datenbank angelegt, auf den alle Zugriff haben. Dadurch kann jeder nicht eingeloggte Benutzer Tags hinzufügen oder die von anderen anonymen Benutzern ändern, da diese unter dem gleichen Namen gespeichert sind.

Um die Gewichtung für den SectionRanking-Algorithmus zu ändern, dienen die zwei Werte im Abschnitt Ranking. Durch Erhöhung eines der beiden Werte bekommt die dazu gehörende Bewertungsmethode einen höheren Anteil an der Gesamtwertung einer Sektion. Umgekehrt verringert sich die Bedeutung bei einem niedrigeren Wert.

6 Konklusion

Die Verwendung eines bestehenden Tagging-Systems auf einem Wiki ist mit Problemen verbunden, die zu Anfang näher erläutert wurden. Deswegen war die Entwicklung eines eigenen Tagging-Systems speziell für das JSPWiki unumgänglich. Bei der Entwicklung konnten die Vorteile der verschiedenen Tagging-Systeme aufgegriffen werden und in die eigene Implementierung einfließen um für das JSPWiki, im speziellen für das Austria-Forum, ein benutzerfreundliches Tagvergabesystem zu schaffen.

Nun ist es durch die Verwendung der Erweiterung möglich, den Benutzern des Austria-Forums zusätzliche Informationen zu bestimmten Artikeln zu liefern und Querverbindungen zu anderen Artikeln, die dasselbe Tag besitzen, aufzuzeigen. Die ursprüngliche Methode, Tags anhand der Scuttle-Oberfläche zu vergeben, wird nun durch eine einfach zu bedienende und voll integrierte Oberfläche abgelöst. Dadurch lässt sich auch annehmen, dass die Bereitschaft zur Tagvergabe steigt. In Verbindung mit dem TagCloudPlugin [Zöhrer 2009], welches zur Suche und Navigation dient, kann mit dieser Erweiterung dem Benutzer ein intuitives Tool zum Umgang mit Tags zur Verfügung gestellt werden. Außerdem erhalten Autoren durch die SectionRanking-Komponente eine einfache Möglichkeit, markierte Sektionen für neue Wiki-Seiten zu verwenden.

7 Ausblick

Da die Implementierung auf das Erweiterungskonzept des JSPWikis aufbaut, lässt sich diese auch mit geringem Aufwand erweitern oder durch ein neues Plugin beziehungsweise einen neuen Filter ergänzen. Durch die SectionTagging-Komponente wurde die Tagvergabe realisiert, und mit der SectionRanking-Komponente wurde eine einfache Möglichkeit geschaffen, Sektionen zu sortieren und diese dann als neue Seite zu speichern.

Für das Austria-Forum wurden bis zum offiziellen Start einige Erweiterungen programmiert und das Design wurde mehrmals überarbeitet. Es ist anzunehmen, dass sich das Austria-Forum in Zukunft auch weiter entwickeln und vergrößern wird und damit einhergehend werden auch neue Anforderungen an das integrierte Tagging-System gestellt werden.

Das SectionRanking-Modul bietet jetzt schon ein Optionsmenü an, welches auf zukünftige Erweiterungen ausgelegt ist. Dadurch können neue Gewichtungparameter für die Sortierung der Sektionen eingebunden werden oder neue Filter hinzugefügt werden.

Auch die SectionTagging-Komponente bietet auf ihrer Seite Funktionen, um neuen Features spezielle Daten zur Verfügung zu stellen. So wird die Versionsnummer der Wiki-Seite schon beim Taggen gespeichert und kann künftig zur genaueren Zuteilung oder zur Überprüfung der Gültigkeit der Datenbankeinträge in Bezug auf das aktuelle Wiki verwendet werden. Das Benutzerinterface kann auch erweitert werden, indem die Click2Tag-Technik integriert wird, um dem Benutzer einen zusätzlichen Tag-Vergabemodus zu bieten.

Somit ist die Weiterentwicklung des Tagging-Systems eng mit dem Erfolg des Austria-Forums verbunden.

III. Abkürzungen

AEIOU	Annotierbares Elektronisches Interaktives Oesterreichisches Universal – Informationssystem
AJAX	Asynchronous JavaScript and XML
ANSI	American National Standards Institute
API	Application Programming Interface
ASP	Active Server Pages
CGI	Common Gateway Interface
CIL	Common Intermediate Language
CLR	Common Language Runtime
CMS	Content Management System
CSS	Cascading Style Sheets
DBMS	Datenbankmanagementsystem
DOM	Document Object Model
DTD	Document Type Definition
ERM	Entity-Relationship-Model
GPL	GNU General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDF	Inverse Document Frequency
IE	Internet Explorer
IICM	Institut für Informationssysteme und Computer Medien
IIS	Internet Information Services
IP	Internet Protocol
ISBN	International Standard Book Number
ISO	International Organization for Standardization
Java EE	Java Platform Enterprise Edition
JDBC	Java Database Connectivity
JDK	Java Development Kit
JS	JavaScript
JSP	JavaServer Pages
JVM	Java Virtual Machine
OOP	Objektorientierte Programmierung

PDF	Portable Document Format
PHP	Hypertext Preprocessor
RCS	Revision Control System
SHA	Secure Hash Algorithm
SVN	Subversion
UCS	Universal Character Set
URL	Uniform Resource Locator
UTF-8	8-bit UCS Transformation Format
TCP	Transmission Control Protocol
TF	Term Frequency
TFIDF	Term Frequency – Inverse Document Frequency
WAR	Web Archive
XML	Extensible Markup Language
YUI	Yahoo! User Interface

IV. Abbildungsverzeichnis

Abbildung 1: Bild für Tag-Klassifizierung	15
Abbildung 2: Blickbewegungen bei (a) Type2Tag und (b) Click2Tag [Hong et al. 2008]	20
Abbildung 3: Taglist Flickr	22
Abbildung 4: Tagcloud Flickr	23
Abbildung 5: Elastic Tag Maps.....	24
Abbildung 6: Scuttle - Favorit hinzufügen.....	27
Abbildung 7: Flickr - Bildübersicht	28
Abbildung 8: Flickr - Tag hinzufügen.....	29
Abbildung 9: Flickr - Aus Tags auswählen.....	29
Abbildung 10: SparTag.us - Symbolleiste	30
Abbildung 11: Artikel des Tages - Original.....	30
Abbildung 12: Artikel des Tages - Hervorgehoben	31
Abbildung 13: Artikel des Tages - Tagging	31
Abbildung 14: Artikel des Tages - Taganzeige.....	32
Abbildung 15: Wikipedia Startseite - Ohne Paragraphenhervorhebung	33
Abbildung 16: Wikipedia Startseite - Mit Paragraphenhervorhebung.....	33
Abbildung 17: ESP-Game.....	35
Abbildung 18: Google Image Labeler.....	36
Abbildung 19: Tagcloud von Tagcrowd	37
Abbildung 20: Wikipedia - Artikelübersicht.....	42
Abbildung 21: Editortoolbar von MediaWiki (Version 1.15.3).....	46
Abbildung 22: Edit-War bei Wikipedia	48
Abbildung 23: Client-Server Architektur.....	51
Abbildung 24: Wiki-Anfrage und Antwort.....	51
Abbildung 25: Wiki-Anfrage mit verändertem Markup	52
Abbildung 26: March 2010 Web Server Survey	55
Abbildung 27: Datenspeicherung mit Textdateien.....	55
Abbildung 28: Datenspeicherung mit Datenbanken	56
Abbildung 29: Datenspeicherung mit Versionsverwaltungssystemen.....	57
Abbildung 30: Austria-Forum (Altes Design)	67
Abbildung 31: Austria-Forum (Aktuelles Design).....	67
Abbildung 32: Datenbankschema von Scuttle	74
Abbildung 33: ERM-Diagramm von Scuttle	74

Abbildung 34: Schildkröte	78
Abbildung 35: SectionTagging-Button	86
Abbildung 36: Wiki-Seite ohne SectionTagging	86
Abbildung 37: Wiki-Seite mit SectionTagging.....	87
Abbildung 38: Taganzeigefenster ohne Tags	87
Abbildung 39: Taganzeigefenster mit Tags	88
Abbildung 40: Tagbearbeitungsfenster - leer	89
Abbildung 41: Tagbearbeitungsfenster - befüllt	90
Abbildung 42: Tagbearbeitungsfenster - Speichervorgang.....	90
Abbildung 43: Auswahl aus dem Taganzeigefenster	91
Abbildung 44: SectionRanking - Ausgangsseite.....	92
Abbildung 45: SectionRanking - Drag & Drop	92
Abbildung 46: SectionRanking - Neuordnung	93
Abbildung 47: Rahmen für Drag & Drop (eingeklappt)	93
Abbildung 48: Rahmen für Drag & Drop (aufgeklappt).....	94
Abbildung 49: Optionsmenü (geschlossen)	94
Abbildung 50: Optionsmenü (geöffnet)	94
Abbildung 51: Wiki-Seite Speicherung	95
Abbildung 52: Bearbeitungsfenster einer Wiki-Seite	95
Abbildung 53: Vergleich traditionelle Webanwendung (links) zu einer mit AJAX (rechts)...	98
Abbildung 54: Ablauf - Wiki-Seitengenerierung.....	101
Abbildung 55: Ablauf - Tagsanforderung	102
Abbildung 56: Ablauf - SectionRanking.....	103
Abbildung 57: Konfigurationsdatei.....	110

V. Formelverzeichnis

Formel 1: TFIDF	37
Formel 2: Term frequency	37
Formel 3: Inverse document frequency	38
Formel 4: TFIDF Detailansicht	38

VI. Quellcodeverzeichnis

Quellcode 1: Servlet "Hello World" Beispiel	58
Quellcode 2: JSP "Hello World" Beispiel	59
Quellcode 3: Markup für Plugin	62
Quellcode 4: Markup für Pluginaufruf	62
Quellcode 5: JSPWiki - Plugin	79
Quellcode 6: JSPWiki Filterinterface	80
Quellcode 7: Pluginausgabe	85
Quellcode 8: Document Type Definition XML-Antwort	88
Quellcode 9: XML-Antwort von STGetTags.jsp	89
Quellcode 10: CSS-Definition einer Dragbox (Sektion)	97
Quellcode 11: Grundgerüst einer XML-Antwort	97
Quellcode 12: Prepared Statement – String (Java)	99
Quellcode 13: Prepared Statement – Ausführung (Java)	100
Quellcode 14: CSS-Markierung	107
Quellcode 15: Verweis auf CSS-Datei	107
Quellcode 16: JS-Markierung	107
Quellcode 17: Verweis auf JS-Dateien	107
Quellcode 18: Zusätzliche Datei-Importierungen der Edit.jsp	109
Quellcode 19: Markierung in der Edit.jsp	108
Quellcode 20: Zusatzcode der Edit.jsp	108
Quellcode 21: Aktivierung des Filters	109
Quellcode 22: Tomcat UTF-8 Einstellung	109

VII. Tabellenverzeichnis

Tabelle 1: Tag-Klassifizierung.....	15
Tabelle 2: Implizite Funktionen	17
Tabelle 3: Explizite Funktionen	18
Tabelle 4: Lesezeit und Anzahl der Tags [Hong et al. 2008].....	20
Tabelle 5: Überblick über die unterstützenden Tagging-Systeme	39
Tabelle 6: Markup-Vergleich	45
Tabelle 7: Markup - Textformatierungen.....	61
Tabelle 8: CSS im Wiki-Markup	62
Tabelle 9: Systemanforderungen von JSPWiki [JSPWiki 2010b].....	63
Tabelle 10: Browserstatistik für das Jahr 2010 von Jänner bis Mai.....	72
Tabelle 11: Datenbanktabelle "sc_bookmarks"	75
Tabelle 12: Datenbanktabelle "sc_user"	75
Tabelle 13: Datenbanktabelle "sc_tags"	76
Tabelle 14: Datenbanktabelle "sc_watched"	76
Tabelle 15: URL-Bestandteile.....	81
Tabelle 16: Identifikationsbestandteile der Sektion	82
Tabelle 17: Dateistruktur.....	105
Tabelle 18: Installationsverzeichnisse.....	106
Tabelle 19: Bibliotheken	106

VIII. Referenzen

[AEIOU 2010]

AEIOU: „Über das Projekt“,
URL: <http://www.aeiou.at/aeiou/project.htm>
Letzter Abruf: 31.05.2010

[Ames und Naaman 2007]

Ames, M., Naaman, M.: „Why we tag: motivations for annotation in mobile and online media“, In Proceedings of the conference on Human Factors in Computing Systems, San Jose, Kalifornien, USA, pp. 971-980, ACM Press New York, NY, USA (2007), ISBN-13: 978-1-59593-593-9.

[AF 2010]

Austria Forum: „Grundsätze und Struktur des Austria-Forums“,
URL: http://austria-lexikon.at/af/Infos_zum_AF/Grunds%C3%A4tze
Erstelldatum: 12.03.2010, Letzter Abruf: 31.05.2010

[Begelman et al. 2006]

Begelman, G., Keller, P., Smadja, F.: „Automated Tag Clustering: Improving search and exploration in the tag space“, In Proceedings of the 15th International World Wide Web Conference (WWW2006) – Workshop on Collaborative Web Tagging, Edinburgh, Schottland, 2006.

[Bergsten 2003]

Bergsten, H.: „JavaServer Pages“, Third Edition, O’Reilly Verlag, USA, 2003, ISBN-10: 0-596-00563-6.

[Bodendorf 2008]

Bodendorf, F.: „Daten- und Wissensmanagement“, Zweite Auflage, Springer Verlag, Berlin, 2008, ISBN-13: 978-3540287438.

[Brooks und Montanez 2006]

Brooks, C. H., Montanez, N.: „Improved Annotation of the Blogosphere via Autotagging and Hierarchical Clustering“, In Proceedings of the 15th international conference on World Wide Web, Edinburgh, Schottland, pp. 625-632, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-323-9.

[Budiu et al. 2009]

Budiu, R., Pirolli, P., Hong, L.: „Remembrance of Things Tagged: How Tagging Affects Human Information Processing“, In Proceedings of the 27th conference on Human Factors in Computing Systems, pp. 615-624, ACM Press New York, NY, USA, 2009, ISBN-13: 978-1-60558-246-7.

[C2 2010a]

Cunningham & Cunningham, Inc.: „Wiki History“,
URL: <http://c2.com/cgi/wiki?WikiHistory>
Erstelldatum: 11.03.2010, Letzter Abruf: 31.03.2010

[C2 2010b]

Cunningham & Cunningham, Inc.: „Why doesn't Wiki do HTML“,
URL: <http://c2.com/cgi/wiki?WhyDoesntWikiDoHtml>
Erstelldatum: 28.02.2010, Letzter Abruf: 14.04.2010

[C2 2010c]

Cunningham & Cunningham, Inc.: „Why Wiki Works“,
URL: <http://c2.com/cgi/wiki?WhyWikiWorks>
Erstelldatum: 04.05.2010, Letzter Abruf: 05.05.2010

[Carson und Ogle 1996]

Carson, C., Ogle, V. E.: „Storage and Retrieval of Feature Data for a Very Large Online Image Collection“, IEEE Computer Society Bulletin of the Technical Committee on Data Engineering, 1996, Vol. 19 No. 4.

[Clements et al. 2008]

Clements, M., De Vries, A. P., Reinders, M. J. T.: „Detecting Synonyms in Social Tagging Systems to Improve Content Retrieval“, In Proceedings of the 31st conference on Research and Development in Information Retrieval, pp. 739-740, ACM Press New York, NY, USA, 2008, ISBN-13: 978-1-60558-164-4.

[Cunningham 2005]

Cunningham, W.: „Correspondence on the Etymology of Wiki“,
URL: <http://c2.com/doc/etymology.html>
Erstelldatum: 2005, Letzter Abruf: 17.03.2010

[Ebersbach et al. 2008]

Ebersbach, A., Glaser, M., Heigl, R., Warta, A.: „WikiTools: Kooperation im Web“, Zweite Auflage, Springer Verlag Berlin Heidelberg, 2008,
ISBN-13: 978-3-540-35110-8.

[Fielding et al. 1999]

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: „Hyper Transfer Protocol – HTTP/1.1 (RFC 2616): Method Definitions“,
URL: <http://ftp.ics.uci.edu/pub/ietf/http/rfc2616.pdf>
Erstelldatum: 1999, Letzter Abruf: 01.04.2010

[Flanagan 2002]

Flanagan, J.: „Search Referral Zeitgeist“, Ein Auszug des Artikels ist unter: „History of the Tag Cloud“ zu finden,
URL: <http://tagging.wordpress.com/tag/history/>
Erstelldatum: 10.04.2007, Letzter Abruf: 05.01.2010

[Fraley 2004]

Fraley, C. R.: „How to conduct behavioral research over the internet“, The Guilford Press, New York, USA, 2004,
ISBN-10: 1-57230-997-0.

[Fressel 2009]

Fressel, C.: „Standardkonformes Web Design anhand des Beispiels 'Austria-Forum““, Masterarbeit am Institut für Informationssysteme und Computer Medien (IICM) der TU Graz, URL: www.iicm.tu-graz.ac.at/thesis/cfressel.pdf
Erstelldatum: September 2009, Letzter Abruf: 31.05.2010

[Furnas et al. 2006]

Furnas, G. W., Fake, C., Von Ahn, L., Schachter, J., Golder, S., Fox, K., Davis, M., Marlow, C., Naaman, M.: „Why Do Tagging Systems Work?“, In Proceedings of the 1st conference on Human Factors in Computing Systems, pp. 36-39, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-298-4.

[Garg und Weber 2008a]

Garg, N., Weber, I.: „Personlized Tag Suggestion for Flickr“, In Proceedings of the 17th international conference on World Wide Web, Beijing, China, pp. 1063-1064, ACM Press New York, NY, USA, 2008, ISBN-13: 978-1-60558-085-2.

[Garg und Weber 2008b]

Garg, N., Weber, I.: „Personlized, Interactive Tag Recommendation for Flickr“, In Proceedings of the 2008 ACM conference on Recommender systems, Lausanne, Schweiz, pp. 67-74, ACM Press New York, NY, USA, 2008, ISBN-13: 978-1-60558-093-7.

[Garret 2005]

Garret, J. J.: „Ajax: A new Approach to Web Application“, URL: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>
Erstelldatum: 19.02.2005, Letzter Abruf: 20.06.2010

[Golder und Huberman 2005]

Golder, S. A., Huberman, B. A.: „The Structure of Collaborative Tagging Systems“, URL: <http://arxiv.org/ftp/cs/papers/0508/0508082.pdf>
Erstelldatum: 18.08.2005, Letzter Abruf: 12.10.2009

[Gennick 2004]

Gennick, J.: „SQL kurz & gut“, Deutsche Ausgabe, O'Reilly Verlag, Deutschland, 2004, ISBN-10: 3-89721-268-4.

[Guelich et al. 2000]

Guelich, S., Gundavaram, S., Birznieks, G.: „CGI-Programming with PERL“, Second Edition, O'Reilly Verlag, USA, 2000
ISBN-10: 1-56592-419-3.

[Hall 2010]

Hall, M.: „An Overview of Servlet & JSP Technology“, URL: <http://courses.coreservlets.com/Course-Materials/pdf/csajsp2/01-Overview-and-Setup.pdf>
Erstelldatum: 2010, Letzter Abruf: 21.04.2010

[Hall und Brown 2003]

Hall, M., Brown, L.: „Core Servlets and JavaServer Pages“, Second Edition, Prentice Hall Verlag, New Jersey, USA, 2003, ISBN-13: 978-0130092298.

[Halvey und Kean 2007]

Halvey, M., Kean, M. T.: „An Assessment of Tag Presentation Techniques“, In Proceedings of the 16th conference of World Wide Web, pp. 1313-1314, ACM Press New York, NY, USA, 2007, ISBN-13: 978-1-59593-654-7.

[Helic et al. 2009]

Helic, D., Us Saeed, A., Trattner, C.: „Creating Dynamic Wiki Pages with Section-Tagging“, In Proceedings of the 1st Workshop on New Forms of Xanalogical Storage and Function, held as part of the ACM Hypertext 2009, Turin, Italy, pp. 8-22, URL: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-508/paper4.pdf>
Erstelldatum: 2009, Letzter Abruf: 10.06.2010

[Helic et al. 2010]

Helic, D., Trattner, C., Strohmaier, M., Andrews, K.: „On the Navigability of Social Tagging Systems“, The Second IEEE International Conference on Social Computing (SocialCom2010), Minneapolis, Minnesota, USA, 2010.

[Hong et al. 2008]

Hong, L., Chi, E. H., Budiu, R., Pirolli, P., Nelson, L.: „SparTag.us: A Low Cost Tagging System for Foraging a Web Content“, In Proceedings of the working conference on Advanced Visual Interfaces, Neapel, Italien, pp. 65-72, ACM Press New York, NY, USA, 2008, ISBN-13: 1-978-60558-141-5.

[Hunter 2001]

Hunter, J., with Crawford, W.: „Java Servlet Programming“, Second Edition, O'Reilly Verlag, USA, 2001, ISBN-10: 0-596-00040-5.

[JSPWiki 2008]

JSPWiki: „JSP Wiki Features“, URL: <http://www.jspwiki.org/wiki/JSPWikiFeatures>
Erstelldatum: 21.06.2008, Letzter Abruf: 22.04.2010

[JSPWiki 2009]

JSPWiki: „Simple JSPWiki“, URL: <http://doc.jspwiki.org/2.4/wiki/Simple%20JSPWiki>
Erstelldatum: 27.08.2009, Letzter Abruf: 29.04.2010

[JSPWiki 2010a]

JSPWiki: „DocMain“, URL: <http://doc.jspwiki.org>
Erstelldatum: 29.04.2010, Letzter Abruf: 29.04.2010

[JSPWiki 2010b]

JSPWiki: „InstallingJSPWiki“,
URL: <http://doc.jspwiki.org/2.4/wiki/InstallingJSPWiki>
Erstelldatum: 20.12.2009, Letzter Abruf: 05.05.2010

[Kahlbach 2007]

Kahlbach, J.: „Designing Web Navigation“, O’Reilly Verlag, Sepastopol, 2007,
ISBN-13: 978-0-596-52810-2.

[Kofler 2005]

Kofler, M.: „MySQL 5: Einführung, Programmierung, Referenz“, Dritte Auflage, Addison-Wesley Verlag, Deutschland, 2005,
ISBN-10: 3-8273-2253-7.

[Lerdorf et al. 2006]

Lerdorf, R., Tatro, K., MacIntyre, P.: „Programming PHP“, Second Edition, O’Reilly Verlag, USA, 2006, ISBN-10: 0-596-00681-0.

[Leuf und Cunningham 2001]

Leuf, B., Cunningham, W.: „The Wiki way: quick collaboration on the Web“, Addison-Wesley, 2001, ISBN-13: 978-0201714999.

[Lohmann und Ziegler 2007]

Lohmann, S., Ziegler, J.: „Bringing Semantics into Folksonomies - Semantische Analyse nutzergenerierter Indexierungen“, In: Informatik 2007 – Informatik trifft Logistik! – Band 1, Beiträge der 37. Jahrestagung der Gesellschaft für Informatik, LNI P-109, pp. 175-179, Köllen, 2007.

[Macgregor und McCulloch 2006]

Macgregor, G., McCulloch, E.: „Collaborative Tagging as a Knowledge Organisation and Resource Discovery Tool“, Library Review, Vol. 55, No. 5, pp. 291-300, 2006.

[Marlow et al. 2006]

Marlow, C., Naaman, M., Boyd, D., Davis, M.: „Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead“, In Proceedings of the 17th conference on Hypertext and Hypermedia, pp. 31-40, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-417-0.

[Mejías 2005]

Mejías, U. A.: „Tag Literacy“,
URL: <http://blog.ulisesmejias.com/2005/04/26/tag-literacy/>
Erstelldatum: 26.04.2005, Letzter Abruf: 29.10.2009

[Mitchell et al. 2001]

Mitchell, S., Walther, S., Seven, D., Mack, D., Payne, C., Anders, B., Nathan, A., Wahlin, D.: „ASP.NET: Tips, Tutorials, and Code“, Sams Verlag, USA, 2001,
ISBN-10: 0-672-32143-2.

[Montero und Solana 2006]

Hassan-Montero, Y., Herrero-Solana, V.: „Improving Tag-Clouds as Visual Information Retrieval Interfaces“, In Proceedings of the 1st International Conference on Multidisciplinary Information Sciences and Technologies (InSciT2006) – Workshop on Information Visualization, Mérida, Spanien, 2006.

[Möller 2003]

Möller, E.: „Das Wiki-Prinzip - Tanz der Gehirne Teil 1“,
URL: <http://www.heise.de/tp/r4/artikel/14/14736/1.html>
Erstelldatum: 09.05.2003, Letzter Abruf: 30.03.2010

[Muchtisch 2007]

Muchtisch, M.: „Tagging – Das Wie und Warum“, Institut für Informationssysteme und Computer Medien (IICM), Technische Universität Graz, 2007.

[Müller-Prove 2008]

Müller-Prove, M.: „Modell und Anwendungsperspektive des Social Tagging“ in Medien in der Wissenschaft, Band 47, Good Tags – Bad Tags: Social Tagging in der Wissensorganisation, Herausgeber: Gaiser, B., Hampel, T., Panke, S., Waxmann Verlag, pp. 15-22, Münster, 2008, ISBN-13: 978-3-8309-2039-7.

[Open Market 1996]

Open Market Inc.: „FastCGI: A High-Performance Web Server Interface“, Technical White Paper,
URL: <http://www.fastcgi.com/devkit/doc/fastcgi-whitepaper/fastcgi.htm>
Erstelldatum: April 1996, Letzter Abruf: 12.04.2010

[Oracle 2007]

Oracle Business & Technology White Paper: „Folksonomy, Keywords, & Tags: Social & Democratic User Interaction in Enterprise Content Management“, Juli 2007
URL: <http://www.oracle.com/technology/products/content-management/pdf/OracleSocialTaggingWhitePaper.pdf>
Erstelldatum: 20.07.2007, Letzter Abruf: 04.01.2010

[PARC 2009a]

Palo Alto Research Center (PARC): „Technology“,
URL: <https://salami.parc.com/spartag/Technology.html>
Erstelldatum: 2009, Letzter Abruf: 20.01.2010

[PARC 2009b]

Palo Alto Research Center (PARC): „Technology“,
URL: https://salami.parc.com/spartag/How_To.html
Erstelldatum: 2009, Letzter Abruf: 20.01.2010

[Pirolli und Anderson 1985]

Pirolli, P. L., Anderson, J. R.: „The Role of Practice in Fact Retrieval“, Journal of Experimental Psychology: Learning, Memory, and Cognition, Volume 11, Nummer 1, pp. 136-153, 1985.

[Porter 1997]

Porter, M. F.: „An algorithm for suffix stripping“, Readings in information retrieval, pp. 313-316, Morgan Kaufmann Publishers Inc., San Francisco, Kalifornien, USA, 1997, ISBN-10: 1-55860-454-5.

[Pürstner 2007]

Pürstner, W.: „Grundlagen und Variationen von Tag-Clouds“, Institut für Informationssysteme und Computer Medien (IICM), Technische Universität Graz, 2007.

[Raggett et al. 1999]

Raggett, D., Le Hors, A., Jacobs, I.: „HTML 4.01 Specification - WC3 Recommendation 24 December 1999“, Kapitel 12 – Links, URL: <http://www.w3.org/TR/html401/>
Erstelldatum: 24.12.1999, Letzter Abruf: 10.06.2010

[Riehle 2006]

Riehle, D.: „How and Why Wikipedia Works: An Interview with Angela Beesley, Elisabeth Bauer, and Kizu Naoko“, In Proceedings of the 2006 International Symposium on Wikis (WikiSym '06), Odense, Dänemark, pp. 3-8, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-413-8.

[Rivadeneira et al. 2007]

Rivadeneira, A.W., Gruen, D.M., Muller, M.J., Millen, D.R.: „Getting Our Head in the Clouds: Toward Evaluation Studies of Tagclouds“, In Proceedings of the 25th International Computer/Human Interaction Conference (CHI2007), San Jose, pp. 995-998, ACM Press New York, NY, USA, 2007, ISBN-13: 978-1-59593-593-9.

[Robertson 2004]

Robertson, S.: „Understanding Inverse Document Frequency: On the theoretical arguments for IDF“, Journal of Documentation, Volume 60, Nummer 5, pp. 503–520, 2004.

[Salton und Yang 1973]

Salton, G., Yang, C. S.: „On the Specification of Term Values in Automatic Indexing“, Journal of Documentation, Volume 29, Nummer 4, pp. 351-372, MCB UP Ltd, 1973.

[Sauer et al. 2006]

Sauer, C., Smith, C., Benz, T.: „Wiki Markup Standard Workshop“, In Proceedings of the 2006 international Symposium on Wikis (WikiSym '06), Odense, Dänemark, pp. 129-130, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-413-8.

[Sauer et al. 2007]

Sauer, C., Smith, C., Benz, T.: „WikiCreole: a common wiki markup“, In Proceedings of the 2007 international Symposium on Wikis (WikiSym '07), Montreal, Quebec, Canada, pp. 131-142, ACM Press New York, NY, USA, 2007, ISBN-13: 978-1-59593-861-9.

[Sen et al. 2006]

Sen, S. W., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., Riedl, J.: „tagging, communities, vocabulary, evolution“, In Proceedings of the 20th anniversary conference on Computer supported cooperative work, pp. 181-190, ACM Press New York, NY, USA, 2006, ISBN-10: 1-59593-249-6.

[Sen et al. 2009]

Sen, S. W., Vig, J., Riedl, J.: „Leraning to Recognize Valuable Tags”, In Proceedings of the 13th international conference on Intelligent user interfaces, pp. 87-96, ACM Press New York, NY, USA, 2009, ISBN-13: 978-1-60558-168-2.

[Shirky 2005]

Shirky, C.: „Ontology is Overrated: Categories, Links and Tags”,
URL: http://shirky.com/writings/ontology_overrated.html
Erstelldatum: 2005, Letzter Abruf: 28.10.2009

[Sigurbjörnsson und Van Zwol 2008]

Sigurbjörnsson, B., Van Zwol, R.: „Flickr Tag Recommendation based on Collective Knowledge“, In Proceedings of the 17th international conference on World Wide Web, Peking, China, pp. 327-336, ACM Press New York, NY, USA, 2008, ISBN-13: 978-1-60558-085-2.

[Steyer 2008]

Steyer, R.: „AJAX Frameworks – RIAs mit Dojo & Co”, Addison-Wesley Verlag, 2008, ISBN-13: 978-3-8273-2728-4.

[Stöcker 2006]

Stöcker, C.: „Politiker schönen ihre Lebensläufe“, Manager Magazin,
URL: <http://www.manager-magazin.de/it/artikel/0,2828,398453,00.html>
Erstelldatum: 01.02.2006, Letzter Abruf: 15.04.2010

[Surowieki 2005]

Surowieki, J.: „The Wisdom of Crowds”, Anchor, 2005, ISBN-13: 978-0385721707.

[Taylor 2003]

Taylor, A. G.: „The Organization of Information”, Second Edition, Libraries Unlimited, 2003, ISBN-13: 978-1563089695.

[Trattner 2009]

Trattner, C.: „Vom Austria-Forum zum Wiki-Konzept”, Masterarbeit am Institut für Informationssysteme und Computer Medien (IICM) an der TU Graz,
URL: <http://www.iicm.tu-graz.ac.at/thesis/DA-Trattner.pdf>
Erstelldatum: Jänner 2009, Letzter Abruf: 30.05.2010

[Trattner et al. 2010]

Trattner, C., Hasani-Mavriqi, I., Helic, D., Leitner, H.: „The Austrian way of Wiki(pedia)! - Development of a Structured Wiki-based Encyclopedia within a Local Austrian Context“, In Proceedings of the 6th International Symposium on Wikis and Open Collaboration (WikiSym '10), Gdansk, Poland, ACM Press New York, NY, USA, 2010, ISBN-13:978-1-4503-0056-8.

[Vander Wal 2005]

Vander Wal, T.: „Folksonomy Definition and Wikipedia“,
URL: <http://www.vanderwal.net/random/entrysel.php?blog=1750>
Erstelldatum: 02.11.2005, Letzter Abruf: 08.10.2009

[Vander Wal 2005a]

Vander Wal, T.: „Explaining and Showing Broad and Narrow Folksonomies“

URL: <http://www.vanderwal.net/random/entrysel.php?blog=1635>

Erstelldatum: 21.02.2005, Letzter Abruf: 29.10.2009

[von Ahn 2006]

von Ahn, L.: „Games with a Purpose“, IEEE Computer Magazine, pp. 96-98, 2006,

URL: <http://www.cs.cmu.edu/~biglou/ieee-gwap.pdf>

Erstelldatum: Juni 2006, Letzter Abruf: 30.01.2010

[von Ahn und Dabbish 2004]

von Ahn, L., Dabbish, L.: „Labeling Images with a Computer Game“, In Proceedings of the SIGCHI conference on Human factors in computing system, Wien, Österreich, pp. 319-326, ACM Press New York, NY, USA, 2004, ISBN-10: 1-58113-702-8.

[WikiMatrix 2009]

WikiMatrix: „JSP-Wiki“,

URL: <http://www.wikimatrix.org/show/JSPWiki>

Erstelldatum: 19.11.2009, Letzter Abruf: 28.04.2010

[WikiMedia 2010]

WikiMedia: „User groups“,

URL: http://meta.wikimedia.org/wiki/User_groups

Erstelldatum: 25.03.2010, Letzter Abruf: 01.04.2010

[Wikipedia 2010]

Wikipedia: „Geschützte Seiten“,

URL: http://de.wikipedia.org/wiki/Wikipedia:Gesch%C3%BCtzte_Seiten

Erstelldatum: 09.04.2010, Letzter Abruf: 15.04.2010

[Xu et al. 2006]

Xu, Z., Fu, Y., Mao, J., Su, D.: „Towards the Sematic Web: Collaborative Tag Suggestions“,

In Proceedings of the Collaborative Web Tagging Workshop at the World Wide Web 2006

Edinburgh, Schottland, 2006.

[Zöhner 2009]

Zöhner, M.: „Navigation in Social Tagging Systemen“,

Masterarbeit am Institut für Informationssysteme und Computer Medien (IICM) der TU Graz,

2009.

IX. Index

A

AEIOU 66
 AJAX 98
 Anonymer Benutzer 73, 110
 Apache 54
 ASP.NET 54
 Austria-Forum 65, 66, 72

B

Benutzerrechte 44
 Bilderkennungsalgorithmen 34
 Binnenversalie 43, 48
 bitweaver 49
 Blickverhalten 20
 Blinde Tagging 16
 Bürokrat 44

C

Cascading Style Sheets 62, 96
 Click2tag-Technik 19
 Click2Tag-Technik 29, 31
 Clustering 23
 Common Gateway Interface 52, 59
 Computerspiele 34
 Creole 46, 61

D

Datenbankmanagementsystem 56
 DokuWiki 49
 Drag & Drop 92, 93, 99

E

Edit-War 47
 Empfohlenes Tagging 16
 Enzyklopädie 41
 ESP-Game 34
 Explizite Funktionen 17

F

FastCGI 53
 Filter 80
 Firefox-Plugin 26, 30
 Flickr 21, 27
 Folksonomie 13

G

Google Image Labeler 35

H

Homographen 13
 Homonymen 13
 HTML 45
 Hypertext 40
 Hypothesen 77

I

Implizite Funktionen 17
 Index 21

J

Java Servlets 30, 58
 JavaScript 85, 96
 JavaServer Pages 53, 58
 JSPWiki 49, 58, 72

K

Kategorien 82
 Kategorisierung 13

L

Lemmatisierung 14

M

Markup-Sprache 45, 61
 MediaWiki 44, 50
 MoinMoin 50

N

Navigation 21
 Normalisierung 14
 Nupedia 41

O

Onlinespiele 34
 Optionsmenü 94

P

PHP 53
 Plugin 61, 79

S

Sandbox 43
 Scuttle 26, 73, 81
 SectionRanking 70, 72, 84, 91
 SectionTagging 70, 71, 85
 Sektionen 70
 Selbstkorrektur 19
 Selektionsalgorithmen 17, 77
 Server 51
 Sichtbares Tagging 16
 Spartag.us 30
 Spracherkennung 21
 Stopplisten 36
 Synonymen 13
 Syntax 46

T

Tabu-Wörter 34, 35

Tag 12, 13
Tag-Cloud 22, 26, 36
TagCloudPlugin 72, 76, 106
Tagcrowd 36
Tagging-Systeme 12, 26
Tagvergabe 16
Technologien 85
Textdatei 55
TFIDF 37
Trennzeichen 13
TWiki 50
Type2Tag-Technik 19

V

Vandalismus 47
Vergleichsfunktion 43
Versionierung 47, 57, 76

Versionsnummer 73, 83
Versionsverwaltungssystem 56
Visualisierung 23, 45
Vokabular 13, 19

W

Webserver 54, 110
Wikipedia 41, 47
Wiki-Systeme 40
WikiWikiWeb 40, 48
Wissenssammlung 65

Y

YUI Library 99, 107