Professur für Grafische Datenverarbeitung
Fachbereich Informatik und Mathematik

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Diplomarbeit

Extraktion semantischer Informationen aus WIKI-Systemen

vorgelegt von
Sarah Voß

Prüfer: Prof. Dr.-Ing. Detlef Krömker
Betreuer: Silvan Reinhold

25. September 2006

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe und nur unter Verwendung der zulässigen Mittel sowie der angegebenen Literatur angefertigt habe.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Frankfurt am Main, den 25.09.2006

_____

Sarah Voß

Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde ein Konzept zur Extraktion von semantischen Informationen aus Wiki-Systemen entwickelt. Ausgangspunkt ist die Tatsache, dass in einem Wiki-System eine Reihe von Informationen in strukturierten, semi-strukturierten oder unstrukturierten Texten vorliegen, deren Semantik nicht immer auf den ersten Blick ersichtlich ist. Daher umfasste die Analyse zum einen, welche Informationen explizit und welche implizit vorhanden sind und zum anderen, welche Beziehungen sich aus den gefundenen Informationen ableiten lassen. Dabei handelt es sich beispielsweise um Beziehungen zwischen verschiedenen Seiten oder um Beziehungen zwischen Wörtern. Hierfür wurde eine Schablone definiert, die jede Information, die extrahiert werden kann, im Detail beschreibt. Dies beinhaltet sowohl die Semantik und die Datenquelle, aus der die Informationen extrahiert werden können, als auch eine Anleitung zur Extraktion und die abschließende Darstellung als XML-Element. Da aber nicht jede Information und deren Semantik sicher ist, wird zwischen sicheren und unsicheren Informationen unterschieden. Die Analyse hat allerdings ergeben, dass es eine Reihe an Informationen gibt, denen nicht automatisch eine Semantik zugewiesen werden kann. Außerdem wurden die Gemeinsamkeiten und Unterschiede der verschiedenen Wiki-Systeme analysiert, die für die Entwicklung des Konzeptes notwendig waren.

Im Konzept ist die Gesamtarchitektur zur Extraktion von semantischen Informationen enthalten. Zwei Hauptsystemkomponenten waren hierfür notwendig: Wrapper und Mediator. Aufgrund der Unterschiede der Wiki-Systeme, wie beispielsweise die verwendete Programmiersprache, Datenbank oder Datei und Wiki-Syntax, wurde eine Wrapper eingesetzt. Der Mediator dient hingegen als Vermittler zwischen der jeweiligen Anwendung und dem Wiki-System.

Durch die prototypische Implementation des Konzeptes ist die Durchführbarkeit bewiesen, bestimmte semantische Informationen zu extrahieren und diese in eine für die Weiterverarbeitung geeignete Form zu bringen. Das heißt, bestimmte Informationen können automatisch oder halb-automatisch in eine semantische Beziehung zueinander gesetzt werden.

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich während des gesamten Studiums und in der Zeit meiner Diplomarbeit begleitet haben.

Ein besonderer Dank geht an meine Eltern, Hilke Voß-Davies und Joachim Krause, für ihre tatkräftige Unterstützung. Simone Fiedler, danke ich für ihre Geduld, Verständnis und für das fleißige Korrektur lesen. Bei Sebastian Schäfer und Christiane Hoos möchte ich mich für ihren fachlichen Rat und das Korrektur lesen bedanken. Abschließend danke ich meinem Betreuer, Silvan Reinhold, für seine zahlreichen Tipps und Anregungen zur Anfertigung dieser Diplomarbeit.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
CERN	Conseil Européen pour la Recherche Nucléaire
CMS	Content Management System
CSS	Cascading Style Sheets
CSV	Character (auch Comma) Separated Values
DARPA	Defense Advanced Research Projects Agency
DM	Data-Mining
DOM	Document Object Model
DTD	Document Type Definition
FDL	Free Documentation License
GPL	General Public License
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IE	Information Extraction
IIS	Internet Information Services
IR	Information Retrieval
KDD	Knowledge Discovery in Database
MUC	Message Understanding Conference
PHP	Hypertext Preprocessor (urspr.: Personal Home Page)
PMH	Protocol for Metadata Harvesting
POETIC	Portable Extendable Traffic Information Collator
SAX	Simple API for XML
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
SVG	Scalable Vector Graphics
UDDI	Universal Description Discovery and Integration
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
W3C	World Wide Web Consortium
WG	Wrapper Generation
WML	Website Meta Language
WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

Kapitel 1

Einleitung

1.1 Motivation

In einem Wiki lässt sich eine Reihe von semantischen Informationen in strukturierten, semi-strukturierten und unstrukturierten Texten finden. Einige der Informationen sind explizit, andere wiederum nur implizit ersichtlich. Eine automatische Zuweisung der Semantik ist nicht immer möglich, wie das folgende Beispiel zeigt. Links sind unter anderem ein Konzept der Wikis, um Inhalte miteinander zu verknüpfen. Sie ermöglichen dadurch nicht nur die Navigation, sondern stellen auch inhaltliche Zusammenhänge zwischen Seiten her. Das Problem hierbei ist, falls ein Link zwei Seiten in eine semantische Beziehung setzt, kann man diese allerdings nicht automatisch benennen.

Ein weiteres Konzept der Wikis sind Kategorien. Sie dienen dazu, inhaltlich ähnliche Seiten zusammenzufassen. Welche Semantik diese Seiten allerdings miteinander verbindet, bleibt offen. Innerhalb eines Wikis lassen sich nicht nur Beziehungen zwischen Seiten, sondern auch zwischen Begriffen wie der hierarchischen Relation (z.B. die Hyponymie 2.4.2) oder Äquivalenzrelation (z.B. Synonyme 2.4.2), finden.

Ward Cunningham, der Begründer des ersten Wikis, spricht in seinem Buch „The Wiki Way. Quick Collaboration on the Web“ (vgl. [CL01]) über die Existenz einer Wiki-Struktur. Diese Grundstruktur ergibt sich aus der Verlinkung der Seiten, der Textabschnitte und der daraus resultierenden Beziehungen. In einem Wiki kann jeder Autor willkürlich Seiten anlegen, diese in Kategorien einordnen und Links setzen. Dadurch bringt jeder Autor seine eigene Struktur in das Wiki ein. Ward Cunningham vergleicht diese Struktur mit einer Sanddüne (vgl. [CL01]). Die Grundstruktur bleibt erhalten, es findet jedoch immer eine Bewegung statt. Wenn es, wie Cunningham beschreibt, eine Struktur innerhalb der Wikis gibt, so muss genau daraus eine Semantik ableitbar sein.

1.2 Aufgabenstellung und Zielsetzung

Im Rahmen dieser Diplomarbeit wird eine Analyse darüber durchgeführt, welche Informationen in einem Wiki-System identifizierbar sind und wie diese in eine semantische Beziehung gesetzt werden können. Der Frage, wie man diese Informationen extrahieren, bewerten und darstellen kann, wird nachgegangen. In diesem Zusammenhang wird erarbeitet, welche Informationen auf algorithmischem Weg oder mittels Heuristiken gefunden werden können und wie diese im Detail aussehen. Nach der Erstellung eines allgemeinen Konzeptes und der Integration in eine Gesamtarchitektur, stellt eine prototypische Implementierung die Funktionsweise unter Beweis. Als Basisplattform wird das MediaWiki unter Verwendung der Technologien PHP[1], HTML[2], XML[3] und JavaScript gewählt.

Ziel ist es, implizit vorhandene Informationen explizit zu machen. Zum Beispiel steht eine Seite A mit einer anderen Seite B im Zusammenhang. Es gibt keinen Link von Seite A zu Seite B oder umgekehrt. Wie man diese Beziehung trotzdem herstellen kann, soll analysiert werden. Ein weiteres Ziel ist, aus heterogenen Daten eine uniforme Repräsentation zu erlangen. Die Ausgabe soll so gewählt werden, dass die extrahierten Informationen unabhängig von der Anwendung und dem Anwendungsszenario weiterverarbeitet oder visualisiert werden können. Das entwickelte Konzept soll allgemein auf alle Wiki-Systeme anwendbar sein, unabhängig von deren Programmiersprache (PHP, Python, Perl usw.) und dem Anwendungsgebiet. Hierfür müssen die Schnittstellen der einzelnen Systemkomponenten (Anwendung, Wiki-System, Wörterbuch, usw.) definiert werden.

1.3 Aufbau der Arbeit

Der Inhalt dieser Diplomarbeit ist in sechs Hauptkapitel gegliedert. Das erste Kapitel, die **Einleitung**, führt mit der Motivation (1.1), der Aufgabenstellung und Zielsetzung (1.2) in das Thema ein. Mit dem zweiten Kapitel werden die **Grundlagen** für die folgenden Kapitel vermittelt. Das beinhaltet zunächst eine Einführung in das Thema Wiki-Systeme (2.1). Im Anschluss daran, folgt die Vorstellung des Forschungsgebietes Informationsextraktion (2.2) und der Techniken des Web-Minings (2.3). Abschließend wird auf die Definition von semantischen Informationen (2.4) eingegangen. Im dritten Kapitel, der **Analyse**, findet eine Untersuchung der Ausgangssituation statt. Am Ende dieses Kapitels sind die Anforderungen (3.7) an das zu entwickelnde System formuliert. Zu Beginn des vierten Kapitels, des **Konzeptes**, werden die einzelnen Systemkomponenten (Kommunikation und Ausgabeformat 4.1.1, Wrapper 4.1.2, Mediator 4.1.3, Anwendung

[1] Abk. für Hypertext Preprocessor (urspünglich: Personal Home Page).

[2] Abk. für Hypertext Markup Language. HTML ist eine Auszeichnungssprache für Hypertexte im WWW. Die Spezifikation befindet sich unter http://www.w3.org/MarkUp/.

[3] Abk. für Extensible Markup Language.

4.1.4) vorgestellt, die sich aus den Anforderungen ergeben. Das Folgekapitel 4.3 beschäftigt sich mit der Frage, welche Informationen gefunden werden und welche Semantik man davon ableiten kann. Ob diese Semantik eine sichere oder unsichere Aussage ist, beschreibt das darauf folgende Kapitel 4.3.3. Abschließend werden die Algorithmen und Heuristiken (4.4) definiert. Das vorletzte Hauptkapitel, die **Implementation**, behandelt die Umsetzung des Konzeptes. Dabei wird unter anderem auf die verwendeten Technologien, die realisierten Funktionen und die Probleme (5.2) eingegangen, die sich bei der Implementation ergaben. Abschluss des fünften Kapitels bildet die Vorstellung des Prototyps anhand von Screenshots. Das letzte Kapitel enthält die **Zusammenfassung** der Arbeit und schließt mit dem **Ausblick** (6) auf weitere Entwicklungen ab. Es werden Ideen und Lösungsansätze für aufbauende Arbeiten gegeben.

Die Beispiele in den Kapiteln 3 (Analyse) und 4 (Konzept) stammen aus der freien Online-Enzyklopädie Wikipedia. Der Grund liegt zum einen darin, dass MediaWiki ursprünglich für Wikipedia entwickelt wurde und zum anderen, dass es zu den größten und bekanntesten Wiki-Systemen gehört. Alle Beispiele hätten auch aus einem anderen Wiki-System ausgewählt werden können.

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Zunächst erfolgt unter 2.1 eine kurze Definition von Wiki-Systemen. Anschließend wird beschrieben, wie Wiki-Systeme entstanden sind (2.1.1) und eine Auswahl von Wiki-Klonen (2.1.4) vorgestellt. Da Informationsextraktion (2.2) noch ein sehr junges Forschungsgebiet ist, wird in 2.2.1 beschrieben, was man genau darunter versteht und in welchen Bereichen sie ihre Anwendung (2.2.3) findet. Unter 2.2.4 findet eine Abgrenzung beziehungsweise ein Vergleich zwischen Informationsextraktion und Information Retrieval statt. Um Informationen extrahieren und auswerten zu können, werden Techniken aus dem Data-Mining verwendet. In 2.3 werden diesbezüglich die verschiedenen Verfahren aus dem Bereich Web-Content-Mining (2.3.1) und dem Web-Structure-Mining (2.3.2) dargestellt. Im letzten Unterkapitel 2.4 des Grundlagenkapitels wird eine Definition von semantischen Informationen (2.4.1) vorgenommen. Außerdem werden die verschiedenen semantischen Relationen (2.4.2) vorgestellt und geklärt, was unter semantischen Informationen im Bereich der Wiki-Systeme (2.4.3) zu verstehen ist.

2.1 Wiki

Ein **Wiki**, auch WikiWiki oder WikiWeb genannt, ist eine Sammlung dynamischer Webseiten, die von den Benutzern nicht nur gelesen, sondern auch online geändert, gelöscht und neu angelegt werden können. Die Wiki-Software wird aber auch in Intranets oder auf privaten Rechnern eingesetzt. Wikis ähneln Content Management Systemen (CMS). Die Hauptaufgabe eines CMS liegt in der Verwaltung des Inhalts. Viele große Firmen oder Universitäten benutzen CMS für ihre Webauftritte. Ein Wiki kann als CMS eingesetzt werden oder auch als Ergänzung zu einem CMS. Es gibt eine gemeinsame Schnittmenge beider Systeme, aber auch einige Unterschiede (vgl. [Lan05]). Der Name WikiWiki stammt von „wikiwiki“, dem hawaiianischen Wort für „schnell“, das in diesem Zusammenhang für das schnelle und unkomplizierte Hinzufügen von Inhalten zu einem Thema steht. Wie bei Hypertexten üblich, sind die einzelnen Seiten eines Wikis durch Querverweise (Links) miteinander verbunden. Jeder Nutzer kann direkt im Brow-

ser Artikel editieren und neu anlegen. Dadurch wird eine ursprüngliche und zuvor nicht verwirklichte Idee des WWW von Tim Berners-Lee realisiert (vgl. [BL06]). Berners-Lee war schon zu Beginn der Ansicht, dass ein Web-Browser eine Kombination aus Viewer und Editor sein sollte. Anfang der neunziger Jahre schlug er seinem damaligen Arbeitgeber CERN[1] ein Projekt vor, das auf dem Prinzip des Hypertextes beruhte und den weltweiten Austausch sowie die Aktualisierung von Informationen zwischen Wissenschaftlern vereinfachen sollte. Damit wurde der Grundstein für das WWW, wie es heute existiert, gelegt.

2.1.1 Geschichte

Im Jahr 1995 entstand das erste Wiki nach einer Idee des amerikanischen Software-Entwicklers Ward Cunningham (vgl. [cun06]). Dieses sogenannte Ur-Wiki „Portland Pattern Repository“ ist in Perl geschrieben und wurde anfangs als Wissenssammlung und als Forum für Design Patterns (Entwurfsmuster)[2] genutzt. Heute findet es immer noch seine Anwendung hauptsächlich in Pattern und Extreme Programming[3]. Zusätzlich enthält es noch Informationen zu Personen und Projekten in diesem Bereich (vgl. [Lan05]). Mittlerweile existieren zahlreiche Wiki-Engines in unterschiedlichen Programmiersprachen. Zu den bekanntesten zählen MediaWiki, MoinMoin und UseModWiki (siehe 2.1.4). Mit dem Durchbruch von Wikipedia[4] im Jahr 2001, einer freien Online-Enzyklopädie, bekamen die Wikis einen immer größeren Stellenwert in der Öffentlichkeit. Zur Zeit registriert Wikipedia 25,6 Millionen Besucher im Monat - das ist Platz 18 der meist besuchten Internet-Angebote (vgl. [der06]). Wie man an der Popularität von Wikipedia sieht, ist das Interesse an einer Enzyklopädie, an der jeder mitwirken kann, sehr groß.

Die Seiten eines Wikis können von jedem Nutzer, der einen Browser besitzt, erzeugt, bearbeitet und gelöscht werden. Dafür ist keine Installation einer Software notwendig. Durch die einfache Syntax ist das Editieren bzw. Formatieren auch für Benutzer ohne HTML-Kenntnisse schnell und einfach möglich. Es können nicht nur Seiten neu angelegt, sondern auch existierende bearbeitet werden. Dadurch

[1] Abk. für Conseil Européen pour la Recherche Nucléaire. Europäische Organisation für Kernforschung.

[2] „Ein Entwurfsmuster beschreibt ein spezielles Entwurfsproblem, das in bestimmten Entwurfskontexten immer wieder auftaucht, und liefert ein bewährtes generisches Schema für dessen Lösung.“ [Dro05].

[3] „Extreme Programming (XP) ist eine leichtgewichtige Softwareentwicklungsmethodik, insbesondere für kleine bis mittlere Projektteams in Projekten mit sich ändernden bzw. entwickelnden Anforderungen, benutzt bekannte und bewährte Entwicklungsstrategien und Praktiken, die sowohl dem Programmierer als auch dem Projektmanagement entgegenkommen und führt sie konsequent („extrem“) durch.“ [Dro05].

[4] http://www.wikipedia.org. Wikipedia ist ein Projekt zum Aufbau einer freien Enzyklopädie in mehr als 200 Sprachen. Jeder kann mit seinem Wissen beitragen. Seit 2001 entstanden so 1.357.323 Artikel in englischer und 458.158 in deutscher Sprache. Stand: 01.09.2006.

können gefundene Fehler von jedem Benutzer geändert werden, so dass eine hohe Qualität der Informationen gewährleistet ist.

Lizenzen

In einem Wiki kann jeder Nutzer Inhalte hinzufügen. Wichtig ist nur, dass die Urheberrechte eingehalten werden. Materialien, wie Texte oder Bilder, aus anderen Quellen dürfen nicht ohne Angabe des Autors und der Quelle übernommen werden. Viele Wiki-Systeme, wie auch das MediaWiki, verwenden die GNU General Public License (GPL) FDL[5]. Die GPL gewährt jedermann, das Programm für jeden Zweck ohne Einschränkung zu nutzen. Kommerzielle Nutzung ist hierbei ausdrücklich eingeschlossen. Kopien des Programms können frei weitergegeben und sogar verkauft werden. Jedoch muss der Quelltext mitgeliefert oder auf Anfrage zur Verfügung gestellt werden. Jeder Nutzer darf das Programm an seine eigenen Bedürfnisse anpassen. Alle abgeleiteten Programme eines unter der GPL stehenden Werkes, dürfen nur dann verbreitet werden, wenn sie ebenfalls unter der GPL lizenziert werden (Copyleft-Prinzip). (vgl. [Lan05])

2.1.2 Allgemeine Funktionsweise

Charakteristisch für Wikis sind der **Bearbeitungsmodus**, die **interne Verlinkung**, die **Speicherung der Versionen** und die **Suchfunktion**. Im Bearbeitungsmodus kann ein Autor einen Artikel neu anlegen oder verändern. Der eingegebene Text ist in den meisten Fällen kein HTML, sondern eine Wiki-Syntax (siehe 2.1.3). Dies soll die Formatierung und Strukturierung der Seite für die Autoren vereinfachen. Die interne Verlinkung wird ausführlich in dem Kapitel Wikilinks (3.3.1) behandelt. Die Speicherung der Versionen dokumentiert alle vorausgegangenen Versionen einer einzelnen Seite. Sie erlaubt es auch, eine alte Version wieder herzustellen und ist damit ein wirksames Mittel gegen Vandalismus und Spamming. Unter Vandalismus wird die gezielte Zerstörung von Seiten verstanden. Darunter fällt, die komplette Löschung des Inhalts einer Seite, das Entfernen längerer Textabschnitte ohne Begründung, das Einfügen von Unsinn und das absichtliche Verfälschen von Informationen. Beim Spamming werden Beiträge automatisiert übermittelt, die durch ihre Masse zunehmend die redaktionellen Kapazitäten der Wiki-Benutzer überfordern. Viele Wikis bieten zudem eine Differenzfunktion, die die Änderungen zwischen zwei Versionen einer Seite anzeigt. Durch optische Hervorhebungen sieht der Benutzer auf einem Blick, welche Teile einer Seite korrigiert, gelöscht oder ergänzt wurden (vgl. [Bir06]). Über die Suchfunktion läßt sich ein Beitrag direkt über den Titel oder mittels der Volltextsuche finden.

[5] Abk. für Free Documentation License.

2.1.3 Syntax

Die Formatierungssyntax[6] der einzelnen Wiki-Systeme ist unterschiedlich, was eine Datenmigration schwierig macht. Es gibt Bestrebungen, hier einen Standard zu entwickeln (vgl. [wik06d]). Dabei geht es nicht darum, eine neue und einheitliche Syntax zu entwickeln, sondern vielmehr aus den vielen verschiedenen Syntaxen einige Regeln festzulegen. Das ermöglicht Autoren, die in verschiedenen Wikis aktiv sind, auch eine einfachere Handhabung. Im Anhang auf Seite 113 befindet sich eine Tabelle, die die Syntax ausgewählter Wikis im Vergleich darstellt.

2.1.4 Wiki-Klone

Es gibt heute ungefähr 200 verschiedene Wiki-Klone (vgl. [wik06c]). Sie unterscheiden sich unter anderem in ihrem Anwendungsgebiet, der Funktionalität, dem Installationsschwierigkeitsgrad und der Syntax. Ein Wiki kann als Wissensdatenbank, Diskussionsforum, als Web Content Management System oder auch als Groupware genutzt werden. Auch wenn sich die Wiki-Klone, wie oben beschrieben, sehr stark unterscheiden können, existieren Funktionen, die in allen Wiki-Systemen gefunden werden. In jedem Wiki-System gibt es einen Bearbeitungsmodus, interne Verlinkungen, die Möglichkeit Versionen zu speichern und eine Suchfunktion. Eine nähere Beschreibung der Funktionen kann im Kapitel 2.1.2 nachgelesen werden. Die bekanntesten Wiki-Klone sind MediaWiki, TWiki, TikiWiki, MoinMoin und UseMod, die jeweils in diesem Kapitel kurz vorgestellt werden.

Um die einzelnen Wikis miteinander zu vergleichen, findet man im Internet einen Wizard, der aus vorher ausgewählten Wikis eine Vergleichsmatrix aufstellt (siehe [wik06g]). In der Auflistung erhält man Informationen über die Lizenz, die verwendete Programmiersprache, die Art der Datenspeicherung, den Entwicklungsstatus, die Systemanforderungen, die Sicherheitsvorkehrungen, die Anti-Spam Mechanismen, die Syntax sowie über die verschiedenen Ausgabeformate.

MediaWiki [7]: MediaWiki ist eine Software, die ursprünglich für Wikipedia, eine freie Online-Enzyklopädie, entwickelt wurde. Die Software unterliegt der GNU General Public License (GPL) (vgl. [gnu06]). Mittlerweile findet MediaWiki nicht nur in Wikipedia, sondern auch in vielen anderen Projekten,

[6] Auch Wiki-Syntax, Wikitext oder Wiki-Markup genannt.
[7] http://www.mediawiki.org. Offizielle Seite von MediaWiki.

wie dem Wictionary[8], Wikibooks[9] und Wikinews[10] ihre Verwendung. MediaWiki ist jetzt in Version 1.7.0 erschienen. Die Entwickler haben die Unterstützung von PHP 4, MySQL 3.23.x und den experimentellen Support von Oracle-Datenbanken über verworfen. Dafür gibt es jetzt eine experimentelle Unterstützung von PostgreSQL. Neu ist auch die Option nicht nur gelöschte Seiten, sondern auch Dateien und Bilder in einem nicht aus dem Web erreichbaren Verzeichnis zwischenzuspeichern, um sie so bei Bedarf mittels Undelete wiederherstellen zu können. Als Standardeinstellung ist das Einbinden von externen Bildern deaktiviert. Diese Maßnahme soll dazu beitragen, die Sicherheit von Wiki-Projekten zu verbessern.[11]

TWiki [12] ist in Perl geschrieben und gehört mit seinen vielen Plugins und Features zu den umfangreichsten Wiki-Klonen. Es wurde zur Nutzung in Firmenintranets konzipiert. Aufgrund seines hohen Entwicklungsstandes setzt man es zunehmend für kommerzielle Zwecke ein. Es zeichnet sich außerdem durch seine Groupware-Fähigkeiten aus. Die aktuelle Version ist 4.0.4. (vgl. [Lan05]).

TikiWiki [13] ist ein leistungsfähiges Open Source Content Management System auf der Basis von PHP, ADOdb[14] und Smarty. Es wird dazu verwendet, alle Arten von Web-Anwendungen zu realisieren: Sites, Portale, Intranets und Extranets.

MoinMoin [15] ist in Python geschrieben. Die erste Version entstand im Jahr 2000. Der Name „MoinMoin“ ist eine Anspielung auf den norddeutschen Gruß „Moin“ sowie auf die Doppelung und „CamelCase“-Schreibweise[16] von „WikiWiki“, einem Synonym von „Wiki“. Durch die Nutzung von Python

[8] Wictionary ist ein frei verfügbares, mehrsprachiges Wörterbuch für den Wortschatz aller Sprachen. Es existiert seit dem 1. Mai 2004 und umfasst derzeit 20.882 Einträge. http://de.wiktionary.org/wiki/Wiktionary:Hauptseite. Stand 04.06.2006.

[9] Wikibooks ist eine mehrsprachige Bibliothek mit Lehrbüchern und anderen Lern- und Lehrmaterialien. Die deutschsprachige Ausgabe gibt es seit dem 21. Juli 2004. Seither hat eine Vielzahl von Freiwilligen 5.166 Buchkapitel in 461 Büchern angefertigt. http://de.wikibooks.org/wiki/Hauptseite. Stand 04.06.2006.

[10] Wikinews ist ein Wiki für Nachrichten. http://de.wikinews.org/wiki/Hauptseite. Stand 04.06.2006.

[11] http://www.heise.de/newsticker/meldung/75219. MediaWiki 1.7.0 erschienen. Stand: 08.07.2006.

[12] http://twiki.org/.

[13] http://tikiwiki.org/tiki-index.php.

[14] Abk. für ActiveX Data Objects Database.

[15] http://moinmoin.wikiwikiweb.de/.

[16] In den meisten Wikis werden interne Links im CamelCase erzeugt, d.h. Wörter werden mit großen Anfangsbuchstaben versehen und ohne Zwischenraum aneinander gesetzt. Der Begriff CamelCase ist auf die verwendeten Großbuchstaben zurückzuführen. In den so formatierten Wörtern erinnern sie vom Aussehen an ein Kamel und dessen Höcker.

kann MoinMoin bei einigen Webhostern nicht installiert werden, da sie nur PHP als Programmiersprache installiert haben.

UseMod [17] (Usenet Moderation Project) ist eine der ältesten und die meist eingesetzte Wiki-Software. Es wurde von Clifford Adams in Perl geschrieben. Bei diesem Wiki existieren eine Menge Patches, um diverse Zusatzfunktionen zu ermöglichen. Das Ziel von UseMod ist es, Usenet-News-Beiträge zu bewerten, Meinungen, Zusammenfassungen und nachträgliche Änderungen miteinander auszutauschen (vgl. [Lan05]). UseMod-Wikis setzen üblicherweise CamelCase-Links ein, das ist aber nicht zwingend. Vor allem die Syntax des UseMod sowie seiner Patches und Klone hatte großen Einfluss auf die Entwicklung anderer Wikis. So ist zum Beispiel MediaWiki syntaktisch stark an das UseMod angelehnt. Das bekannteste mit UseMod betriebene Wiki ist das MeatballWiki[18], das die Entwicklung der UseMod-Software stark beeinflusst hat.

2.2 Informationsextraktion

2.2.1 Definitionen

Informationsextraktion (IE) beschäftigt sich mit der Aufgabe aus unstrukturierten oder semi-strukturierten Dokumenten (wie zum Beispiel HTML-Dokumenten) Informationen zu extrahieren, die im gegebenen Anwendungskontext benötigt werden. Dabei kommen Werkzeuge wie z.B. Wrapper zum Einsatz, die zumeist strukturierte XML-Dokumente als Output erzeugen.

[Eik99] definiert einen Wrapper folgendermaßen.

> „A Wrapper can be seen as a procedure that is designed for extracting content of a particular information source and delivering the content of interest in a self-describing representation."

Im Problembereich der automatisierten Informationsextraktion aus HTML-Seiten des WWW versteht man unter Wrappern spezialisierte Softwareroutinen, die im Wesentlichen drei Aufgaben erfüllen: Erstens müssen sie HTML-Seiten von einer Website herunterladen, zweitens die gewünschten Daten in den Seiten lokalisieren sowie extrahieren und drittens, die so gewonnenen Daten in einem geeignet strukturierten Ausgabeformat für die weitere Manipulation zur Verfügung stellen. Die

[17] http://usemod.com/cgi-bin/wiki.pl.

[18] Das MeatballWiki (http://www.usemod.com/cgi-bin/mb.pl), wurde 2000 von Sunir Shah als Forum für kollaborative Hypermedien, insbesondere Wikis, gegründet. Zu fast jeder Wiki-Engine und jedem größeren Projekt gibt es bei Meatball eine Diskussion. Mit „meatballs" (Fleischklopse) sind hier metaphorisch Inhalte gemeint, die durch Hyperlinks (vergleiche Spaghetti) miteinander verbunden sind (vgl. [Lan05]).

Daten können dann von anderen Anwendungen eingelesen und weiterverarbeitet werden (vgl. [KS06]).

Im Datenbankbereich wird unter einem Wrapper eine Softwarekomponente verstanden, die Daten und Abfragen von einem Modell in ein anderes umwandelt. Im WWW ist die Aufgabe eines Wrappers implizite Informationen in HTML-Dokumenten in explizite Daten in einer Datenstruktur zur Verfügung zu stellen, die für weitere Vorgänge wieder verwendet werden können. Die Extraktion des Inhalts geschieht mittels Regeln. Um aus verschiedenen Quellen Informationen zu extrahieren, wird eine Bibliothek von Wrappern benötigt. Das Bedürfnis nach solchen Werkzeugen, die Informationen aus verschiedenen Web-Quellen extrahieren, führte zur Wrapper Generation (WG).

Ein Wrapper kann **manuell**, **halb-automatisch** oder **automatisch** konstruiert werden. Der Nachteil eines manuell erstellten Wrapper ist, dass die Programmierung sehr zeitaufwendig und fehleranfällig ist. Da sich das WWW extrem schnell verändert, muss der Wrapper bei jeder Änderung neu angepasst oder komplett neu geschrieben werden. Das kann zu erhöhten Wartungskosten führen.
Damit auf Änderungen eingegangen werden kann, bieten sich Mechanismen an, die sich automatisch auf eventuell neue Formate anpassen. Ein semi-automatischer Wrapper bietet eine GUI[19], in der man die zu extrahierenden Informationen auswählen kann. Diese Methode benötigt keine direkte Kodierung mehr, sondern nur das Expertenwissen des jeweiligen Gebietes. Der Vorteil liegt darin, dass das System weniger fehleranfällig als das direkte Programmieren ist. Der Nachteil besteht darin, dass für jede neue Seite ein Durchlauf gestartet werden muss, um dem Wrapper die extrahierenden Informationen zu zeigen.
Um schnell auf Änderungen innerhalb einer Seite zu reagieren, bieten sich die automatischen Wrapper an. Dafür werden einfache bis sehr komplexe Lernalgorithmen verwendet. Diese Systeme erfordern nur ein Minimum an Experteninteraktion. Zunächst müssen sie eine Trainingsphase durchlaufen, in der sie unter Aufsicht mit Trainingsbeispielen gefüttert werden. Nach der Trainingsphase können automatische Wrapper ohne Interaktion Informationen extrahieren.(vgl. [KT02, Eik99])

Informationsextraktion ist in Kombination mit anderen Technologien, wie zum Beispiel dem Data-Mining (siehe 2.3), eine sehr mächtige Technik, um Informationen aus Texten zu extrahieren und auszuwerten (vgl. [GY98]).

[19] Abk. für Graphical User Interface.

2.2.2 Geschichte

Die Entwicklung auf dem noch recht jungen Forschungsgebiet der Informationsextraktion wurde durch die MUCs[20] vorangetrieben. Die sieben MUC wurden von 1987 bis 1997 von der DARPA[21], der zentralen Forschungs- und Entwicklungseinrichtung des US-amerikanischen Verteidigungsministeriums, veranstaltet (vgl. [App99]).

Die Art der Dokumente, auf die die IE-Systeme angewendet werden, hat sich seit 1987 sehr verändert. Ursprünglich wurden IE-Systeme zur Extraktion von Informationen aus Nachrichten angewendet. Heutzutage findet sich ihre Anwendung in verschiedenen Bereichen (siehe 2.2.3) wie zum Beispiel auf Webseiten im Internet. Im Vergleich zu den traditionellen Dokumenten (Nachrichten) unterscheiden sich Webseiten sehr stark. Im Web gibt es eine Unmenge an Dokumenten. Täglich kommen neue Dokumente hinzu, Inhalte werden ständig geändert und die meisten Dokumente enthalten sowohl strukturierte als auch semi-strukturierte Texte. Die Existenz von Hyperlinks ist ebenfalls neu. Die dynamische Veränderung der Hypertexte stellte eine neue Herausforderung an die IE-Systeme. (vgl. [Eik99])

2.2.3 Anwendungsgebiete

IE-Systeme werden in vielen verschiedenen Anwendungsgebieten eingesetzt. Wie die Geschichte der Entstehung des Forschungsgebietes der Informationsextraktion zeigt, ist eines der häufigsten Einsatzgebiete das Militär. Aber auch in Finanzanwendungen, in der Medizin, bei der Polizei sowie in der Wissenschaft werden diese Systeme eingesetzt. In der Medizin ermöglichen IE-Systeme beispielsweise die Klassifikation von Patientenakten und Entlassungsbriefen und unterstützen somit die öffentliche Gesundheitsforschung sowie die ärztliche Revision.

Das AVENTINUS Projekt[22] entwickelt Werkzeuge, welche die Polizei bei der Bekämpfung des Drogenhandels unterstützen sollen. Ein weiteres Projekt ist das POETIC[23], ein IE-System zur Extraktion von Straßenverkehrsinformationen aus Störungsprotokollen. Dabei verfolgt man das Ziel, zeitnah Verkehrsinformationen für Verkehrsteilnehmer aus Live-Daten zu liefern. (vgl. [GY98])

2.2.4 Informationsextraktion vs. Information Retrieval

Informationsextraktion (IE) und **Information Retrieval** (IR) werden oft gleichgesetzt. Sie unterscheiden sich jedoch nicht nur in ihren Zielen, sondern

[20] Abk. für Message Understanding Conference.
[21] Abk. für Defense Advanced Research Projects Agency.
[22] http://www.svenska.gu.se/aventinus/. Stand: 08.07.2006.
[23] Abk. für POrtable Extendable Traffic Information Collator.
http://www.informatics.susx.ac.uk/research/nlp/poetic/poetic.html.

auch in den verwendeten Techniken. IR[24] sucht gezielt nach relevanten Dokumenten zu einer Suchanfrage und IE nach relevanten Informationen innerhalb eines Dokuments.

> „[...] The contrast between the aims of IE and IR sytems can be summed up: IR retrieves relevant documents from collections, IE extracts relevant information from documents. The two techniques are therefore complementary [...]“ [GY98]

Die Standardmetrik im IR ist das Recall-und Precision-Maß. Der Recall gibt den Anteil der relevanten Dokumente an, die gefunden wurden und Precision den Anteil der relevanten Dokumente unter den gefundenen [Dro05]. Abgeleitet von dieser Definition ist in der IE der Recall ein Maß für den Anteil der relvanten Informationen, die korrekt extrahiert wurden und Precision ein Maß für den Anteil der extrahierten Informationen, die korrekt sind.

2.3 Web-Mining

Der englische Begriff „Mining“ bedeutet übersetzt „Bergbau“. Wie beim Bergbau der Boden erkundet, erforscht, untersucht und sondiert wird, werden beim Data-Mining Daten untersucht (vgl. [HQW06]).

Unter **Data-Mining** (DM) oder auch **Knowledge Discovery in Database** (KDD) versteht man einen Prozess, bei dem aus großen Mengen gespeicherter Daten - in den meisten Fällen aus einer Datenbank - neuartige und potentiell nützliche Muster oder Regeln erkannt werden. Ziel ist es, aus vorhandenen Daten neue, ungeahnte Erkenntnisse auf möglichst automatisierte Weise zu extrahieren. KDD beschreibt automatisierte Verfahren, mit denen Regelmäßigkeiten in Mengen von Datensätzen gefunden und in eine Form für die Weiterverarbeitung gebracht werden (vgl. [Fer03]). Obwohl KDD und DM oft synonym verwendet werden, grenzt [ST02] die Begriffe ab. Dr. Schmidt-Thieme bezeichnet KDD als den Prozess der Identifikation von interessanten, nicht trivialen, impliziten, vorher unbekannten und potenziell nützlichen Informationen oder Mustern aus Daten in großen Datenbanken. DM wird somit nur als ein Teilschritt des KDD-Prozesses verstanden. Er besteht aus Algorithmen, die in akzeptabler Rechenzeit aus einer vorgegebenen Datenbasis eine Menge von Mustern liefern. Überträgt man die Data-Mining-Techniken auf Daten im Internet, so spricht man vom **Web-Mining**. Web-Mining lässt sich nach den verwendeten Datentypen folgendermaßen klassifizieren:

[24] Information Retrieval bedeutet übersetzt Informations-Wiedergewinnung.

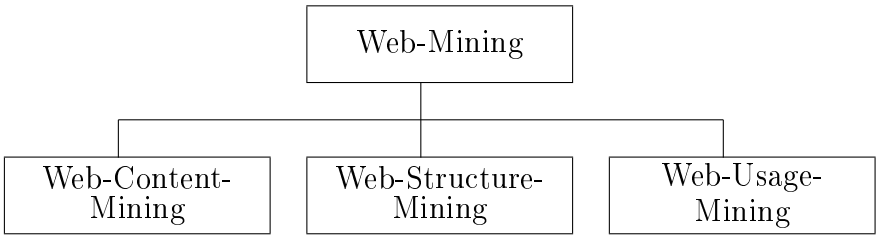


Abbildung 2.1: Taxonomie Web-Mining

Web-Content-Mining beschreibt das Entdecken nützlicher Informationen in Texten, Bildern, Audiodateien, Videodateien, Metadaten und Hyperlinks im Web. Texte können unstrukturiert oder semi-strukturiert sein. Zur Analyse des Textes kommen unter anderem Methoden aus dem Text-Mining zum Einsatz.

Web-Structure-Mining befasst sich mit der Hyperlinkstruktur zwischen Webseiten.

Web-Usage-Mining beschäftigt sich mit dem Benutzerverhalten.

Im Rahmen dieser Diplomarbeit werden Techniken aus dem Bereich Web-Content-Mining und Web-Structure-Mining verwendet. Der Bereich Web-Usage-Mining, das Data-Mining-Techniken beispielsweise auf Log-Files anwendet, ist nicht Bestandteil dieser Arbeit.

2.3.1 Web-Content-Mining

Web-Content-Mining bezieht sich auf jegliche Daten im Web. Mit dem Auffinden von Informationen in unstrukturierten Texten (siehe 3.1.2) beschäftigt sich das **Text-Mining**. Als Ziele verfolgt das Text-Mining, die Identifikation von relevanten und spezifischen Fachausdrücken eines Anwendungskontextes, die Berechnung von semantischen Relationen zwischen Wörtern sowie das Auffinden von Ähnlichkeiten zwischen Begriffen (vgl. [HQW06]). Die hierfür verwendeten Verfahren sind **statistische**, **musterbasierte** und **clusterbasierte Verfahren**.

Statistische Verfahren werden für die Bestimmung der Häufigkeitsverteilung von Wortformen sowie bei statistischen Abhängigkeiten zwischen Wortformen verwendet. Bei den musterbasierten Verfahren werden Texte nach vorher definierten Mustern (regulären Ausdrücken) durchsucht. Reguläre Ausdrücke finden ihre Anwendung in der Theoretischen Informatik und in der Softwareentwicklung. G. Schnitger, Professor der Theoretischen Informatik an der Johann Wolfgang Goethe-Universität in Frankfurt am Main, definiert reguläre Ausdrücke folgendermaßen [Sch02]:

Das Alphabet $\Sigma = \{a_1, \ldots, a_k\}$ sei gegeben.
(a) $\emptyset$, ϵ, $a_1, \ldots, a_k$ sind reguläre Ausdrücke (für die Sprachen $\emptyset$, $\{\epsilon\}$ und $\{a_1\}$, $\ldots$, $\{a_k\}$).
(b) Sei R ein regulärer Ausdruck (für die Sprache L), dann ist R*[25] ein regulärer Ausdruck (für die Sprache L*[26]), und (R) ist ein regulärer Ausdruck (für die Sprache L).
(c) Seien R_1 und R_2 reguläre Ausdrücke (für die Sprachen L_1 und L_2). Dann sind auch $R_1 + R_2$ und $R_1 \cdot R_2$ reguläre Ausdrücke (für die Sprachen $L_1 \cup L_2$[27] und $L_1 \circ L_2$[28]).
Die Menge der regulären Ausdrücke ist die kleinste Menge mit den Eigenschaften (a), (b) und (c).

In der Theoretischen Informatik werden durch reguläre Ausdrücke die regulären Sprachen definiert. In PHP (vgl. [LT03]) oder Perl versteht man unter einem regulären Ausdruck einen String, der ein Muster repräsentiert. Er besteht aus Atomen (einzelnen Zeichen oder Wortformen) und Operatoren, welche die grammatikalischen Regeln ausdrücken. Für fast alle Programmiersprachen existieren Implementierungen für reguläre Ausdrücke.
Um ein sich wiederholendes Muster in einem regulären Ausdruck festzulegen, werden Quantifier benutzt. Ein Quantifier legt fest, wie oft sich das Muster wiederholen darf (siehe Tabelle 2.1). Die Tabelle 2.2 zeigt spezielle Zeichen und ihre Bedeutung innerhalb regulärer Ausdrücke.

Zeichen	**Bedeutung**
?	Der Ausdruck davor ist optional. Er kommt null- oder genau einmal vor.
*	Der Ausdruck davor darf beliebig oft, auch nullmal vorkommen.
+	Der Ausdruck davor muss mindestens einmal vorkommen, darf aber beliebig oft wiederholt werden
{n}	Der Ausdruck davor darf genau n-mal vorkommen.
{n, m}	Der Ausdruck davor darf mindestens n-mal, aber nicht mehr als m-mal vorkommen.
{n, }	Der Ausdruck davor darf mindestens n-mal vorkommen.

Tabelle 2.1: Quantifier. Quelle: vgl. [LT03]

[25] *: Der reguläre Ausdruck darf beliebig oft, auch nullmal vorkommen.

[26] L* heißt der Kleenesche Abschluss von L. L*= $\bigcup_{n=0}^{\infty} L^n$ mit $L^0 = \{\epsilon\}$ (vgl. [Sch02]).

[27] Für Sprachen L_1, L_2 über $\sum$ bezeichnet $L_1 \cup L_2$, die Vereinigung von L_1 und L_2 (vgl. [Sch02]).

[28] Für Sprachen L_1, L_2 über $\sum$ bezeichnet $L_1 \circ L_2 = \{uv | u \in L_1, v \in L_2\}$ die Konkatenation von L_1 und L_2 (vgl. [Sch02]).

Zeichen	Bedeutung
.	genau ein beliebiges Zeichen
\s	Leerzeichen
\S	kein Leerzeichen
\w	Ziffer oder Buchstabe
\W	keine Ziffer und kein Buchstabe
\d	Ziffer
\D	keine Ziffer

Tabelle 2.2: Spezielle Zeichen. Quelle: vgl. [LT03]

Die zweite Definition von regulären Ausdrücken ermöglicht das Identifizieren von Wörtern und Textpassagen, so dass sie anschließend extrahiert und weiterverarbeitet werden können. Genau das ist eine Aufgabe dieser Arbeit.

Bei der Cluster-Analyse wird eine Menge von Elementen (Daten, Objekte) in Cluster (Teilmengen, Gruppen, Klassen, Kategorien) eingeteilt. Die clusterbasierten Verfahren werden genutzt, um semantisch ähnliche Wörter oder Dokumente zu ermitteln. Die zu bildenden Teilmengen werden aus den Eigenschaften der Elemente selbst abgeleitet. Clustering-Verfahren können hierarchisch oder nichthierarchisch arbeiten und die einzelnen Elemente hart oder soft den Teilmengen zuordnen. Hart bedeutet, dass jedes Element nur einem Cluster und bei soft mehreren Clustern zugeordnet wird. Nicht-hierarchische Verfahren arbeiten meist iterativ, d.h. eine initiale Einteilung der Elemente in Cluster wird sukzessiv verbessert. Beispiele für nicht-hierarchische Verfahren sind k-means (vgl. [Mac67]) und Expectation Maximization (vgl. [MS99]). Bei hierarchischen Clustering-Verfahren wird eine Hierarchie von Clustern gebildet. Ein Cluster auf einer höheren Ebene ist jeweils die Vereinigung von zwei Clustern der unmittelbar darunterliegenden Ebene. Die Algorithmen arbeiten entweder agglomerativ[29] oder divisiv[30] (vgl. [HQW06]).

2.3.2 Web-Structure-Mining

Web-Structure-Mining beschäftigt sich mit der Analyse von Hyperlinks. Wenn die Seiten A und B durch einen Hyperlink miteinander verbunden sind, ist die Wahrscheinlichkeit, dass sich beide Seiten mit derselben Thematik beschäftigen,

[29] bottom up: Zu Beginn bildet jedes Element ein eigenes Cluster. In jedem Schritt werden zwei Cluster mit der größten Ähnlichkeit verschmolzen. Der Verschmelzungsprozess endet mit dem Cluster der höchsten Hierarchieebene, das alle Elemente enthält. (vgl. [HQW06]).

[30] top down: Zu Beginn enthält ein Cluster alle Elemente. In jedem Schritt wird das Cluster mit der geringsten Kohärenz geteilt. Der Prozess endet, wenn in der untersten Ebene nur noch einelementige Cluster vorhanden sind. (vgl. [HQW06]).

größer als wenn diese nicht miteinander verbunden wären (vgl. [Dav00]). Wird vom Autor ein Hyperlink erzeugt, so kann er diesem einen Text zuordnen. Dieser Text, auch Ankertext genannt, besteht in der Regel nur aus wenigen Worten und kann als Beschreibung von Hypertext-Dokumenten genutzt werden.

2.4 Semantische Informationen

2.4.1 Definition

Der Begriff **semantische Informationen** ist zusammengesetzt aus dem Wort „Semantik“, der „Lehre von der Bedeutung von Wörtern und sprachlicher Zeichen“ [deu03] und „Information“ (lat. informatio = Bildung, Belehrung, Deutung, Erläuterung) „eine vermittelnde Kenntnis“ [deu03]. Sowohl die Definition des Begriffs Semantik als auch die der Information stellt sich als schwierig heraus, da es sich um sehr komplexe Begriffe handelt, die in verschiedenen Bereichen (Informatik, Informationstheorie, Informationswissenschaft, Nachrichtentechnik, Semiotik und Linguistik) ihre Anwendung finden.

Aus Sicht der Informatik besteht eine Information aus drei Teilen:

1. Einem syntaktischen Teil, der die zulässige Struktur der Bausteine beschreibt, aus denen sich die Information zusammensetzt.
2. Einem semantischen Teil, der die Bedeutung der Information angibt.
3. Einem pragmatischen Teil, aus dem sich der Zweck der Information und die erhofften Handlungen ergeben.

(vgl. [dud93])

2.4.2 Semantische Relationen

Als Grundlage der **semantischen Relationen** dienen Zusammenhänge, die in der Linguistik als **syntagmatische** und **paradigmatische Relationen** beschrieben werden. Zwei Wortformen stehen in syntagmatischer Relation, wenn sie gemeinsam auftreten. Syntagmatische Relationen erfassen typischerweise inhaltliche Zusammenhänge zwischen Wortformen in einem Satz oder zwischen Wortformen, die häufig in unmittelbarer Nachbarschaft gemeinsam auftreten. Beispiele sind die Beziehung zwischen einem Nomen (Sonne) und einem dazupassenden Verb (scheint) oder einer Funktionsbezeichnung und einem Namen einer Person. Wenn zwei Wortformen in ähnlichen Kontexten auftreten, dann stehen sie in paradigmatischer Relation. Die paradigmatischen Relationen kann man in drei Gruppen einteilen: In die Äquivalenz-, Identitäts- und Synonymierelation, in die Polaritäts- und Oppositionsrelation und als dritte Gruppe die hierarchischen Relationen. Die

Tabelle 2.3 zeigt die Relationen innerhalb jeder Gruppe (vgl. [Sch92]). Zwischen zwei Wortformen einer Sprache besteht eine semantische Relation nur dann, wenn sie in einer syntagmatischer oder einer paradigmatischer Relation stehen. Aus diesen Relationen ergeben sich semantische Zusammenhänge zwischen Wörtern (vgl. [HQW06]).

Äquivalenz-, Identitäts- und Synonymierelation	**Polaritäts- und Oppositionsrelation**	**Hierarchische Relation**
Synonyme	Antonyme	Hyponymie
	Komplementarität	Partonymie
	Konversheit	Kollektive

Tabelle 2.3: Paradigmatische Relationen

Synonyme sind Wörter, die eine identische oder ähnliche Bedeutung haben. Sie stehen in einer Äquivalenzrelation[31] zueinander. Zum Beispiel: Wiki, Wiki-Wiki oder Wikiweb. „Die Tatsache, dass es für eine Sache unterschiedliche Bezeichnungen gibt, hängt oft mit Dialekten (z.B. Streichholz/Zündholz, Heidelbeere/Blaubeere), unterschiedlicher Fachsprachlichkeit (z.B. Fernseher/TV-Gerät, Auto/Kfz) oder historischen Umständen (z.B. Astronaut für amerikanische, Kosmonaut für russische Raumfahrer) zusammen“ [HQW06]. Die Synonymierelation ist symmetrisch und transitiv.

Antonyme sind gradierbare Gegensätze, d.h. es gibt Zwischenstufen. Beispiel: Die Wörter heiß und kalt sind graduell antonym, weil es dazwischen auch noch Abstufungen wie z. B. kühl und warm gibt. Diese Relation ist symmetrisch, aber nicht reflexiv und nicht transitiv.

Komplementarität Darunter versteht man nicht gradierbare Gegensätze. Beispiele: lebendig-tot, Mann-Frau oder ledig-verheiratet.

Konversheit Konverse Gegenteile benennen ein Geschehen von unterschiedlichen Bezugspunkten. Beispiel: links-rechts bei Lage eines Gebäudes. Wenn ein Gebäude links liegt, kann es nicht aus derselben Perspektive rechts liegen.

Hyponymie ist eine Relation, die zwischen einem allgemeinen und einem spezifischen Begriff besteht. Beispiel: Naturwissenschaft (Hyperonym=Oberbegriff)

[31] R ist eine Äquivalenzrelation auf M, wenn sie reflexiv, symmetrisch und transitiv ist.
R ist reflexiv, wenn für alle x $\in$ M gilt: xRx.
R ist symmetrisch, wenn für alle x,y $\in$ M gilt: Aus xRy folgt yRx.
R ist transitiv, wenn für alle x,y,z $\in$ M gilt: Aus xRy und yRz folgt xRz. (vgl. [Sie92]).

und Physik (Hyponym=Unterbegriff). Die Oberbegriffs- und Unterbegriffsbeziehung ist nicht symmetrisch. Wenn ein Begriff B zu einem Begriff A ein Oberbegriff ist, kann der Begriff A nicht gleichzeitig zu diesem Begriff B ein Oberbegriff sein. Der Begriff A kann aber zu einem weiteren Begriff C ein Oberbegriff sein. Beide Relationen sind daher transitive Relationen. Der Oberbegriff eines Begriffs A ist stets auch ein Oberbegriff aller Unterbegriffe von A. Hat ein Begriff mehrere Unterbegriffe, werden diese auch als **Kohyponyme** bezeichnet.

Partonymie ist die Teil-Ganzes-Beziehung. Beispiel: Ast-Baum. Ein Ast ist ein Teil vom Baum.

Kollektive Das Wort Kollektivum fasst eine Gruppe gleichartiger Lebewesen oder Dinge zusammen (vgl. [deu03]) - z.B. Familie: Vater, Mutter, Kinder.

Logische Relationen sind semantische Relationen, die logische Folgerungen unterstützen. Hierzu zählen insbesondere die Oberbegriffs- und Unterbegriffsbeziehungen, Synonyme, Gegensätze, Antonyme, Komplementärbegriffe und Konverse.

2.4.3 Semantische Informationen im Bereich der Wiki-Systeme

Semantische Informationen im Bereich der Wiki-Systeme können sehr vielfältig sein. Angenommen es existiert ein Link von einer Seite über Berlin auf eine andere, die das Thema Deutschland thematisiert. Berlin wird demnach in Relation zu Deutschland gesetzt, vorausgesetzt es handelt sich nicht um einen Link, der hauptsächlich zur Navigation dient. Es besteht somit eine semantische Beziehung. In diesem Beispiel ist die Semantik, dass Berlin die Hauptstadt von Deutschland ist. Semantische Relationen setzen Wörter in Beziehung. Welche semantischen Informationen überhaupt aus einem Wiki extrahiert werden können, wird ausführlich im nächsten Kapitel, der Analyse, behandelt.

Kapitel 3

Analyse

Um die Anforderungen an das zu entwickelnde System definieren zu können, findet zunächst eine ausführliche Analyse der Ausgangssituation statt. Das Ziel ist ein allgemeines Konzept. Deshalb müssen die Gemeinsamkeiten und Unterschiede der verschiedenen Wiki-Systeme berücksichtigt werden. Die prototypische Umsetzung erfolgt auf der Basis des MediaWikis, das erklärt, weshalb der Schwerpunkt bei der Analyse auf diesem System liegt. Die Analyse beschäftigt sich zum einen damit, welche Informationen explizit und welche implizit vorhanden sind (3.1.1) und in welcher Form (strukturiert, unstrukturiert oder semi-strukturiert) sie vorliegen (3.1.2). Und zum anderen damit, welche Beziehungen zwischen Informationen bestehen können (3.1.3). Weiterhin wird untersucht wie Wikis, Unterseiten (3.2.1) erzeugen, Namensräume (3.2.2) definieren und welche Zeichenkodierung (3.2.3) benutzt wird. In den darauf folgenden Unterkapiteln werden die verschiedenen Linktypen (3.3) und eine Analyse der Volltextanteile (3.4) durchgeführt. Wie ein Wiki eine angeforderte Seite erstellt, erklärt das Content-Model im Kapitel 3.5. Anschließend wird auf die Architektur von Wiki-Systemen eingegangen und die Datenquellen (3.5.3), aus denen Informationen extrahiert werden können, untersucht. Im vorletzten Kapitel werden die Ausgabeformate, die zur Auswahl stehen, vorgestellt. Auf Grundlage der gesamten Analyse der Ausgangssituation werden in dem Kapitel 3.7 die Anforderungen aus Sicht des Benutzers und des Entwicklers an das System definiert.

3.1 Informationen

Wikis können für verschiedene Zwecke eingesetzt werden, zum Beispiel als Enzyklopädie, Wörterbuch, Forum, Wissensdatenbank, Zitatsammlung, für News-Meldungen oder auch als Dokumentationstool. Man unterscheidet dabei zwischen einem Wiki, das nur innerhalb einer geschlossenen Arbeitsgruppe verwendet wird und einem, das über das WWW jedem Internetnutzer zur Verfügung steht. Gemeinsam ist allen Seiten, dass jedes Dokument einen Namen hat, von einer Person verfasst oder auch von mehreren Personen bearbeitet wurde. Dies geschieht immer zu einem bestimmten Zeitpunkt, d.h. Datum und Uhrzeit lassen sich zuordnen. Anhand des Datums lässt sich die Aktualität der Seite feststellen. Die Quelle, aus

der die Informationen stammen, ist ebenfalls charakteristisch für eine Seite. Abhängig von einem Dokument lassen sich folgende Informationen, die in Abbildung 3.1 dargestellt sind, finden.

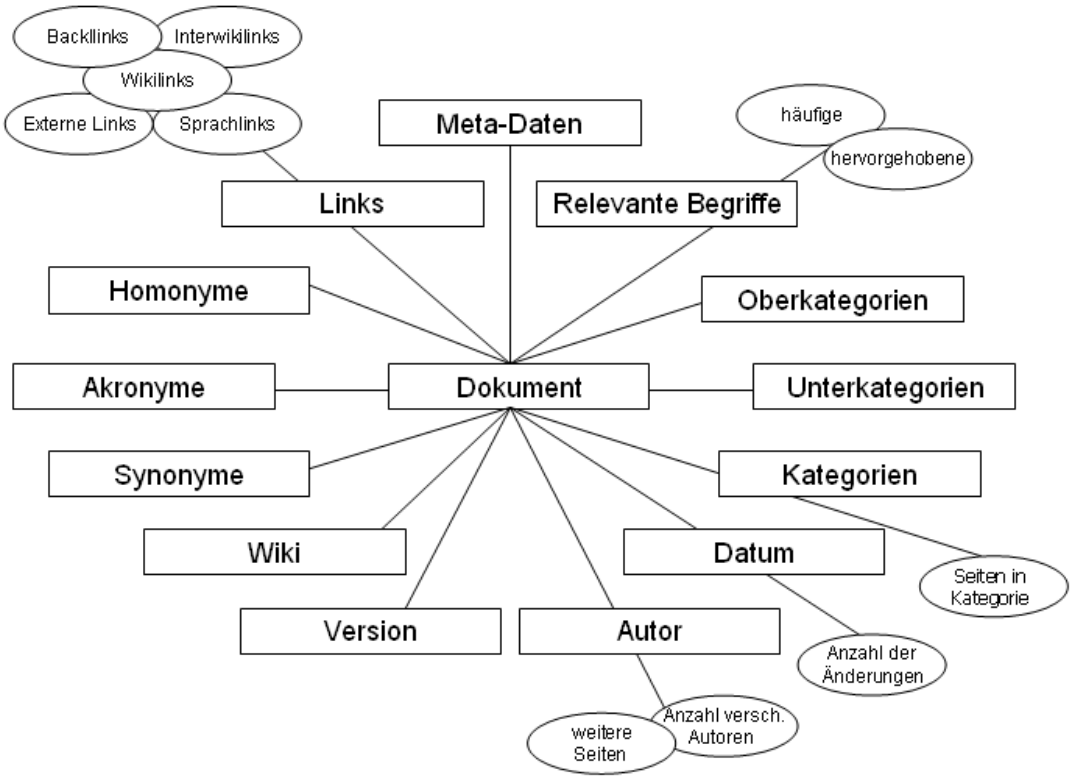


Abbildung 3.1: Ergebnis der Analyse der Informationen

3.1.1 Explizite und implizite Informationen

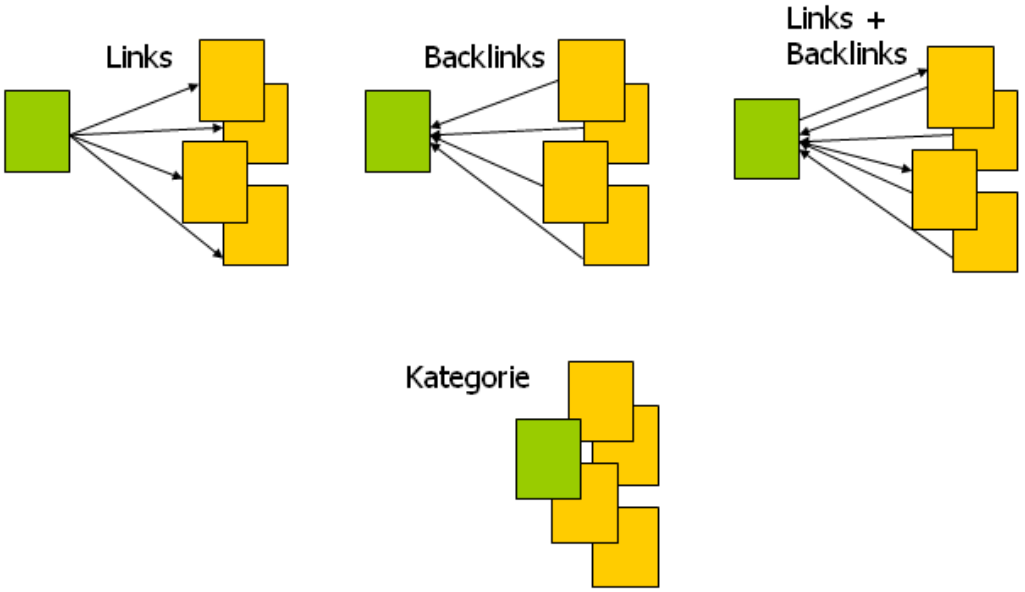


Abbildung 3.2: Explizite und implizite Informationen

Bei der Identifizierung der Informationen muss zwischen **explizit** und **implizit** unterschieden werden. Die Abbildung 3.2 erläutert den Unterschied. Alle ausgehenden Links einer Seite sind explizit ersichtlich. In MediaWiki sind sie beispielsweise blau hervorgehoben. Im Gegensatz dazu sind die Backlinks (eingehende Links) eine implizite Information, da sie nicht sofort erkennbar sind. Erst durch den Aufruf einer Funktion werden die Backlinks einer Seite angezeigt. Die Spezialfunktion „Was zeigt hierhin“ in MediaWiki erzeugt eine neue Seite mit allen Seitennamen als Liste. Um Beziehungen zwischen Seiten ableiten zu können, ist die Information, dass sich zwei Seiten gegenseitig verlinken, von Interesse. Auch diese Information ist nur implizit vorhanden. Erst durch die Bildung der Schnittmenge von Links und Wikilinks werden diese Informationen explizit.

Die Kategorien, in denen eine Seite eingeordnet ist, sind weitere explizite Informationen. In MediaWiki findet man die Kategorien am Ende einer Seite. Um inhaltliche Beziehungen zu anderen Seiten herzustellen, sind die Seiten in den selben Kategorien von Interesse. Diese Informationen erhält man jedoch erst, wenn die Kategorieseite aufgerufen wird. Alle Seiten in den selben Kategorien, werden durch die Vereinigung[1] gebildet. Die Kategorien sind wie oben beschrieben expli-

[1] Alle Seiten in der Kategorie 1 $\cup$ alle Seiten in der Kategorie 2 $\cup \ldots \cup$ alle Seiten in der Kategorie n.

zite und die weiteren Seiten der Kategorien implizite Informationen. Erst durch die Bildung der Vereinigung werden sie zu expliziten Informationen gemacht.

Das letzte Beispiel erläutert den Unterschied zwischen impliziten und expliziten Informationen anhand der Wortbeziehungen. Auf einer Seite befindet sich ein Link von dem Wort Wiki auf die Seite WikiWeb. WikiWeb ist demnach ein Synonym für Wiki. Diese Information wird erst explizit, wenn man die Begriffsrelation angibt.

3.1.2 Strukturierte, semi- und unstrukturierte Informationen

Je nachdem wie der Inhalt einer Seite organisiert ist oder woher die Daten stammen, unterscheidet man zwischen **strukturierten**, **semi-strukturierten** und **unstrukturierten Informationen**.

Daten aus der Datenbank oder Informationen in tabellarischer Form sind strukturiert. Strukturierte Daten lassen sich sehr einfach extrahieren.

Die meisten Informationen im WWW sind semi-strukturiert. Diese Informationen sind nicht so geordnet wie Datenbankeinträge, erhalten aber durch die HTML-Tags eine Formatierung. Überschriften werden durch ihre Größe vom gesamten Text hervorgehoben. Durch das eingeschlossene Tag *<h1>*, zum Beispiel für die Überschrift eins, kann automatisch der Text der Überschrift extrahiert werden.

Im Gegensatz dazu, spricht man von unstrukturierten Informationen bei Texten mit minimaler Formatierung, die nur aus natürlicher Sprache bestehen. In diesem Fall ist es schwierig auf einzelne Teile in einem Dokument zuzugreifen, weil keine Layoutinformationen vorliegen (vgl. [AGM03]).

3.1.3 Seite-Seite, Seite-Wort, Wort-Wort

Eine semantische Beziehung kann zwischen zwei Seiten (**Seite-Seite**), einem charakteristischen Wort und einer Seite (**Seite-Wort**) und zwischen zwei Wörtern innerhalb einer Seite (**Wort-Wort**) bestehen. Folgende Kriterien erlauben eine Schlussfolgerung auf den Zusammenhang zweier Seiten:

Link Ein Link zwischen zwei Seiten kann ein Hinweis auf eine semantische Beziehung sein.

Inhalt Ist der Inhalt beider Seiten ähnlich, kann ein Zusammenhang bestehen.

Autor und Sprachstil Die meisten Autoren haben Vorlieben für bestimmte Wortformen oder Wortgruppen, die sich in ihren verschiedenen Dokumenten wie-

derfinden. Außerdem lässt sich beobachten, dass ein Autor nur zu bestimmten Themen Beiträge abgibt. In den meisten Fällen betreffen diese den Beruf oder auch das Hobby des Autors.

Gemeinsame Wörter Die Ähnlichkeit zweier Seiten ergibt sich aus der Anzahl gemeinsamer Wörter.

In der Seite-Wort Beziehung stehen alle Wörter, die für eine Seite relevant sein können. Das sind wiederholt auftretende Wörter wie beispielsweise Fachtermini oder wichtige Wörter. Ein Dokument wird visuell in verschiedene Abschnitte gegliedert. Wichtige Überschriften werden in einer größeren Schrift und wichtige Wörter werden hervorgehoben (fett, kursiv oder unterstrichen) dargestellt (vgl. [Bur04]). Unter einer Wort-Wort Beziehung versteht man alle Begriffsbeziehungen wie Synonyme, Akronyme oder Homonyme.

3.2 Seiten

Eine Wiki-Seite hat immer einen Namen und sollte idealerweise nur ein Thema behandeln. Es kann allerdings vorkommen, dass dieses Thema unter mehreren Bezeichnungen bekannt ist. Damit es auch unter einem bestimmten Namen gefunden wird, existiert in vielen Wikis das Konzept der Weiterleitung. Weiterleitungsseiten sind Seiten, die nur einen Link auf eine andere Seite besitzen. „Unter einer Weiterleitung, häufig auch mit dem englischen Redirect bezeichnet, versteht man in MediaWiki die Umleitung eines Artikels auf einen anderen. Wird der erste Artikel aufgerufen, zeigt MediaWiki stattdessen den zweiten Artikel an. Damit kann man beispielsweise erreichen, dass beim Aufruf des Artikels Vicco von Bülow stattdessen der Artikel Loriot erscheint. Eine Weiterleitung entspricht also im Wesentlichen dem „siehe: xxxx“ einer klassischen Papierenzyklopädie. Das ganze wird mit dem REDIRECT-Kommando realisiert.“ [wik05c]. In MediaWiki ist die Syntax zum Beispiel: *#REDIRECT [[Zielseite]]*.

3.2.1 Unterseiten

Mit einem Schrägstrich im Titel lassen sich Unterseiten anlegen, zum Beispiel *[[Benutzer:BMW/Getriebe]]*. MediaWiki erzeugt auf der Seite automatisch einen Link zurück zum übergeordneten Artikel, also in diesem Fall *[[Benutzer:BMW]]*. Im präfixlosen Bereich ist diese Funktion ausgeschaltet.

3.2.2 Namensräume

Namensraum ist ein Wiki-Konzept zur Gruppierung von Seiten. MediaWiki unterscheidet verschiedene Namensräume:

- Hauptnamensraum (ohne Bezeichnung),

- Diskussion: eine Diskussionsseite zu einer bereits existierenden Seite,
- Benutzer: für die Benutzer-Homepages,
- Bild: für Bilder und ihre Beschreibungsseiten,
- MediaWiki: enthält die Texte der MediaWiki-Software,
- Vorlage: für Vorlagen und Textbausteine,
- Hilfe: für Hilfeseiten und Anleitungen zum Wiki,
- Spezial: für von der Wiki-Software generierten Spezialseiten,
- Kategorie: für Seitenkategorien,
- Meta: für den Projektnamensraum; Üblicherweise ist dies der Name des Wikis.

(vgl. [wik06h])

Für jeden Namensraum, außer für Spezial, gibt es einen weiteren Namensraum mit Diskussionsseiten. Der Namensraumpräfix steht mit einem Doppelpunkt vor dem eigentlichen Seitentitel. Zum Beispiel ist *[[Benutzer:BMW]]* ein Link auf eine Seite aus dem Namensraum „Benutzer“ - „BMW“ ist der Name des betreffenden Users. Davon unbeschadet kann ein Eintrag [[BMW]] lauten, der zu einer Automarke führt. So werden Konflikte von Benutzer- und Seitennamen verhindert.

3.2.3 Zeichenkodierung

Die ersten Wikis erlaubten nur Zeichen im 7-Bit-ASCII-Zeichensatz. Mittlerweile unterstützen alle Wikis zumindest einen 8-Bit-Zeichensatz wie Latin1 (ISO-8859-1), so dass man beispielsweise deutsche Umlaute direkt eingeben kann. Neuere Wikis kommen, zumindest optional, auch mit Unicode zurecht, meist in der Variante UTF-8[2]. Mit der Umstellung auf MediaWiki 1.5 wird in Projekten, die auf dieser Software beruhen, UTF-8 verwendet. (vgl. [Lan05])

3.3 Links

Unter einem **Link** oder auch **Hyperlink** versteht man den Verweis auf eine Textstelle, ein Dokument oder auch auf eine Datei. Ein Link besteht immer aus einem Verweisziel und einem Text, der das Ziel beschreibt. Viele Links haben nur die Navigation innerhalb eines Dokuments oder zwischen Dokumenten als Zweck. Man kann ihnen keine weitere Bedeutung geben (vgl. [VKV+06]). Aber in den

[2]Abk. für Unicode Transformation Format.

meisten Fällen stellt der Autor durch das Setzen eines Links eine Beziehung zwischen Seiten her, die man auch näher definieren kann.
Je nachdem wohin der Verweis führt, werden innerhalb eines Wikis sechs verschiedene Linktypen unterschieden. Das Ziel kann eine Seite innerhalb eines Projektes (siehe Wikilink 3.3.1), eine Seite in einem anderen Projekt (siehe Interwiki Link 3.3.2), eine externe Webseite (siehe Externer Link 3.3.3), eine Seite desselben Projektes, aber in einer anderen Sprache (siehe Sprachlink 3.3.4), ein Bild (siehe Bildlink 3.3.5) oder ein Link auf eine Kategorieseite (siehe Kategorie 3.3.6) sein. (vgl. [VKV+06])

3.3.1 Wikilink

Ein **Wikilink**, auch **interner Link** genannt, verlinkt von einer Seite auf den Titel einer anderen Seite innerhalb eines Projektes. In den meisten Wikis werden interne Links im CamelCase erzeugt, d.h. Wörter werden mit großen Anfangsbuchstaben versehen und ohne Zwischenraum aneinander gesetzt - Beispiel: *WikiEngine*. Wörter in CamelCase-Schreibweise werden automatisch als Links erkannt. Diese Art einen Link zu erzeugen ist einfach, erschwert aber die Lesbarkeit eines Textes und erzeugt gelegentlich unbeabsichtigte Links. Aus diesem Grund verwenden die meisten Wikis parallel dazu eine andere Schreibweise wie z.B. eckige Klammern bei MediaWiki - *[[Seitenname/Linkname]]*. Der Linkname ist optional. Wenn ein Link innerhalb eines Wikis auf eine nicht vorhandene Seite verweist, wird er optisch anders dargestellt als ein Link zu einer vorhandenen Seite. In MediaWiki werden Links zu nicht existierenden Seiten (Brokenlinks), auch rote Links genannt, weil sie in der Ansicht rot angezeigt werden. Ein Klick auf einen Brokenlink führt in den Bearbeitungsmodus. Dort kann der Inhalt der Seite eingegeben werden. (vgl. [med06a])

Allen Wikis gemeinsam ist der Backlink-Mechanismus, der alle Seiten, die auf die aktuelle Seite verlinken, auflistet [Aum05]. MediaWiki zum Beispiel verwendet einen eigenen Menüpunkt namens „Links auf diese Seite“, der eine Spezialseite aufruft. Diese Funktion listet alle Seiten auf, die mit der aktuell gewählten Seite verknüpft sind. Eine Spezialseite ist eine MediaWiki-Erweiterung, die den Inhalt größtenteils automatisch erstellt und die nicht unmittelbar vom Benutzer geändert werden kann. Man unterscheidet zwischen Spezialseiten[3], die nur bei angemeldeten Benutzern verfügbar sind und Seiten, die immer verwendet werden. (vgl. [wik05a])

3.3.2 Interwiki Link

Als Ward Cunnigham das Wiki erfand, griff er die Idee von Tim Berners-Lee (siehe 2.1), dem Erfinder des WWW, auf. Ein Wiki sollte demnach nicht in sich

[3]Eine Auflistung aller Spezialseiten befindet sich auf Seite 115 im Anhang.

abgeschlossen, sondern mit anderen Wikis vernetzt sein. Diese Idee ließ sich jedoch nicht umsetzen, weil sich die Wiki-Systeme von ihrer Implementation und Nutzung sehr stark unterscheiden. **Interwiki Links** ermöglichen jedoch die Vernetzung der Wikis untereinander (vgl. [Lan05]). Ein Link von einer Seite, innerhalb des WikiBooks-Projektes zu einem Artikel in Wikipedia, wird in MediaWiki folgendermaßen erzeugt - *[[wikipedia:InterWiki]]*. Aus dem Präfix wird bei der Erstellung der HTML-Seite automatisch die URL eingefügt. Die Präfixe und ihre zugehörigen URLs werden in der Datenbanktabelle „interwiki“ gespeichert. (vgl. [med06a])

3.3.3 Externer Link

Bei einem Link auf eine externe Quelle muss die absolute URL angegeben werden. Es wird dabei empfohlen, eine kurze Beschreibung oder Schlüsselwörter mit anzugeben, die das Dokument beschreiben. Syntax in MediaWiki - *[http://www.ziel.de Beschreibung des Ziels]*. (vgl. [med06a])

3.3.4 Sprachlink

Ein **Sprachlink** verlinkt auf ein Dokument desselben Projektes, nur in einer anderen Sprache. Die Syntax in MediaWiki ist die gleiche wie bei der Erzeugung eines Wikilinks. Als Präfix muss jedoch die Sprache angegeben werden - Beispiel: *[[en:Computer science]]*. Über diesen Link gelangt der Benutzer beispielsweise zu dem englischen Artikel über Informatik.

3.3.5 Bilderlink

Ältere Wiki-Systeme können Bilder nicht selbst verwalten, sie aber aus dem Internet in eine Seite einbinden. MediaWiki greift auf eine eigene Bilddatenbank zurück, in die Bilder hochgeladen werden können (vgl. [Lan05]). Der Link auf ein Bild hat dieselbe Syntax wie ein normaler Link, nur als Präfix den Namensraum Bild - Beispiel: *[[Bild: Beispiel.jpg|Bildbeschreibung]]*. Für jedes Bild kann ein alt-Attribut angegeben werden. Das alt-Attribut enthält einen alternativen Text für ein Bild, um den Inhalt des Bildes auch dann zugänglich zu machen, wenn der User-Agent (z.B. der Browser) keine Bilder unterstützt oder der Benutzer das automatische Laden von Bildern deaktiviert hat.[4]

3.3.6 Kategorie

Kategorien ermöglichen Dokumente in verschiedene Themengebiete, abhängig vom Inhalt, einzuordnen. Die Klassifikation erfolgt, indem ein Link von dem Dokument zur Kategorie gesetzt wird. In MediaWiki werden Kategorielinks in einem

[4] Siehe http://www.w3.org/QA/Tips/altAttribute.

speziellen Format am Ende des Gesamttextes platziert (vgl. [VKV+06]). Ein Kategorielink ist ein ganz normaler Link mit Präfix Kategorie - Beispiel in MediaWiki: *[[Kategorie: Naturwissenschaft]]*. Befindet man sich auf der Kategorieseite, kann man sich alle Rückverweise anzeigen lassen und erhält somit alle Dokumente, die zu dieser Kategorie gehören. Kategorieseiten können wiederum auch einer anderen Kategorie zugeordnet werden. Dadurch entsteht eine hierarchische Struktur (vgl. [Lan05]).

3.3.7 Zusammenfassung

Innerhalb einer Seite lassen sich unterschiedliche Linktypen identifizieren. Alle Linkziele erhalten durch den Linktyp eine Semantik. Je nachdem, ob es sich um einen Link: in eine Kategorie, in ein anderes Projekt oder auch auf ein Bild handelt (siehe Abbildung 3.3).

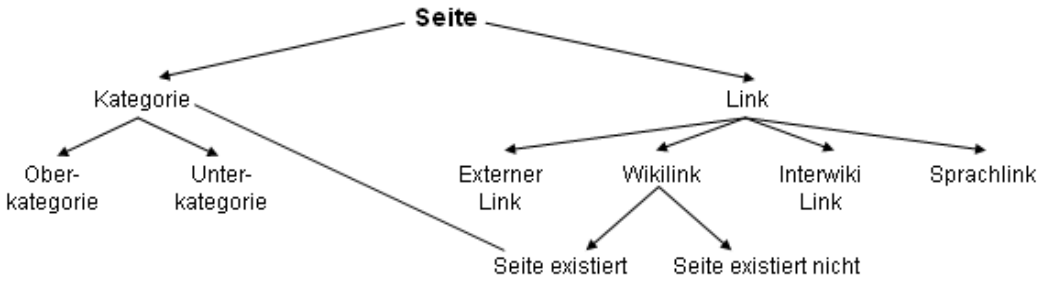


Abbildung 3.3: Links im Überblick

3.4 Volltextanteile

3.4.1 Allgemeine Konventionen

Damit innerhalb eines Wikis die verschiedenen Seiten im Hinblick auf Layout und Textstruktur ähnlich sind, existieren allgemeine Konventionen, die es einzuhalten gilt. Folgende Konventionen verwendet Wikipedia (vgl. [wik06b]):

- Beginne einen Artikel immer mit einer Definition des Schlagworts. Das Schlagwort sollte dabei fett formatiert werden.
- Die Etymologie von Fremdwörtern kommt in Klammern hinter das Schlagwort.
- In Biographien folgen auf den Namen einer Person in Klammern die Lebensdaten. Geburts- und Sterbedatum werden durch ein Semikolon getrennt.

- Zitate müssen sich optisch vom restlichen Text abheben. Dazu werden sie kursiv gesetzt oder eingerückt.

Die meisten Wikis benutzen Textbausteine, um Textabschnitte wiederzuverwenden, die auf vielen Seiten vorkommen. Textabschnitte in MediaWiki sind sogar parametrisierbar. Bei allen Wikis kann ein Textbaustein wiederum auch andere Textbausteine verwenden. Der Nachteil kann jedoch die Auslastung des Servers sein. Denn für jeden Textbaustein ist ein Zugriff auf den Server nötig, sofern nicht ein Teil der Daten im Cache gehalten wird. In MediaWiki sind Textbausteine als Vorlagen (Templates) bekannt. Andere Wikis verstehen unter Templates Formatvorlagen für vollständige Seiten, die dem Benutzer beim Anlegen einer neuen Seite angeboten werden. (vgl. [Lan05])

3.4.2 Worthäufigkeiten

In [Dro05] wird bei der Bestimmung von Termgewichten zwischen lokalen (kontextabhängigen) und globalen (kontextunabhängigen) Einflußfaktoren unterschieden. Ein lokales Kriterium ist zum Beispiel die Anzahl, wie häufig ein Term in einem Dokument auftaucht. Ein globales Kriterium bestimmt die Häufigkeit, mit der ein Term in einer bestimmten Sprache auftaucht. Durch das Zipf'sche Gesetz kann die Verteilung der Wörter in einer Sprache beschrieben werden.

Zipf'sche Gesetz

Für einen repräsentativen Textkorpus D bezeichnet T_D die Menge der Terme, die in D und hfg(t) die Häufigkeit mit welcher der Term t $\in T_D$ in dem Textkorpus D vorkommt. Rang(t) bezeichnet den Rangplatz von t $\in T_D$, wenn die Wörter nach abfallender Häufigkeit sortiert werden. Dann gilt:

$$rang(t) \cdot hfg(t) \sim c = konstant \; \forall \; t \in T_D$$

Daraus lässt sich ableiten, dass die Häufigkeit der Terme in etwa mit $hfg(t) \sim \frac{c}{rang(t)}$ abnimmt. Das heißt, dass eine kleine Anzahl von häufigen Wörtern einen großen Anteil der Texte abdeckt und eine große Anzahl von seltenen Wörtern, die nur einen kleinen Teil des Textes ausmachen. Wörter, die oft vorkommen, sind bekannt als Stopwörter. Stopwortlisten[5] enthalten diese Wörter, die nicht von großer Relevanz für einen Text sind. Im Deutschen sind das zum Beispiel bestimmte Artikel („der“, „die“, „das“), unbestimmte Artikel („einer“, „eine“, „ein“), Konjunktionen („und“, „oder“, „doch“) und häufig gebrauchte Präpositionen („an“, „in“, „von“), sowie die Negation „nicht“. Um relevante Wörter, sogenannte Schlüsselwörter oder Fachterme eines Textes, zu bestimmen, müssen diese Stopwörter

[5] Eine Stopwortliste ist im Anhang auf Seite 116 zu finden.

aus der Gewichtung herausgenommen werden.

Die Gewichtungsfunktionen sind an die eigenen Bedürfnisse anpassbar. Es können beispielsweise die Anzahl der verschiedenen Terme innerhalb eines Dokuments in die Berechnung der Termgewichte eingebaut und/oder Mindestgrenzen für die Vorkommenshäufigkeit eines Terms festgelegt werden. Ein Term muss beispielsweise mindestens fünfmal vorkommen, damit er überhaupt in die Bewertung mit einfließt. Des Weiteren werden Terme hinsichtlich ihres Auftrittsortes im Text bewertet. Damit können Terme in Titeln, in Kapitelüberschriften, in beigefügten Abstracts oder in Bildunterschriften ein höheres Eingangsgewicht bekommen als Terme im übrigen Text. Für diese Festlegungen müssen allerdings einerseits die Eigenschaften verschiedener Textsorten berücksichtigt werden und andererseits entsprechende Textabschnitte identifizierbar sein. Letztere Bedingung setzt voraus, dass der Text zumindest in semi-strukturierter Form (siehe 3.1.2) vorliegt, d.h. in einer der Auszeichnungssprachen HTML, SGML[6] oder XML.

3.4.3 Akronyme

Akronyme sind besondere Abkürzungen, die aus den Anfangsbuchstaben mehrerer (Teil-) Wörter gebildet werden. Im Deutschen werden sie in der Regel ohne Punkte („PC “) geschrieben. Akronyme lassen sich darüber hinaus meist als Wort aussprechen (z.B. „NATO “).[7]

In der Arbeit von Sundaresan [SY00] lag das Ziel darin, Muster für AE-Paare[8] zu finden. Anschließend wurden die Muster, die gute AE-Paare beschreiben, identifiziert. Die Tabelle 3.4 zeigt, wie viele Paare eines Musters in einem Experiment gefunden wurden. Als Apronym bezeichnet man ein Akronym, das ein existierendes Wort ergibt - zum Beispiel: AJAX ist die Abkürzung für Asynchronous JavaScript and XML und der Name eines griechischen Helden, der vor Troja kämpfte.

[6] Abk. für Standard Generalized Markup Language.
[7] http://de.selfhtml.org/html/text/logisch.htm#elemente. Stand: 15.07.2006.
[8] akronym-expansion, d.h. Abkürzung und die zugehörigen Teil-Wörter.

pattern	# of *AE-pairs* extracted by each pattern
expansion (acronym)	896
acronym (expansion)	332
acronym - expansion	98
acronym : expansion	31
expansion [acronym]	9
(acronym) expansion	6
acronym [expansion]	5
acronym, expansion,	5
(expansion) acronym	2
[acronym] expansion	1
	1,385*
	# of new *AE-pairs*
without duality-based mining	1,033**

Abbildung 3.4: AE-Paare. Quelle: [SY00]

(*) Die Summe aller entdeckten AE-Paare ohne Überprüfung, ob ein Paar schon von einem anderen Muster erkannt wurde.
(**) Ohne die AE-Paare, die schon von anderen Mustern entdeckt wurden.

3.4.4 Homonyme

Als **Homonym** bezeichnet man ein Wort, das unterschiedliche Bedeutungen hat - zum Beispiel: Bank (Sitzgelegenheit) und Bank (Geldinstitut). Umgangssprachlich ist ein Homonym auch als „Teekesselchen“ bekannt. Das Gegenteil eines Homonyms ist das Synonym (siehe 2.4.2), bei dem unterschiedliche Bezeichnungen für denselben Begriff verwendet werden.

3.5 Wiki-Systeme

3.5.1 Content-Model

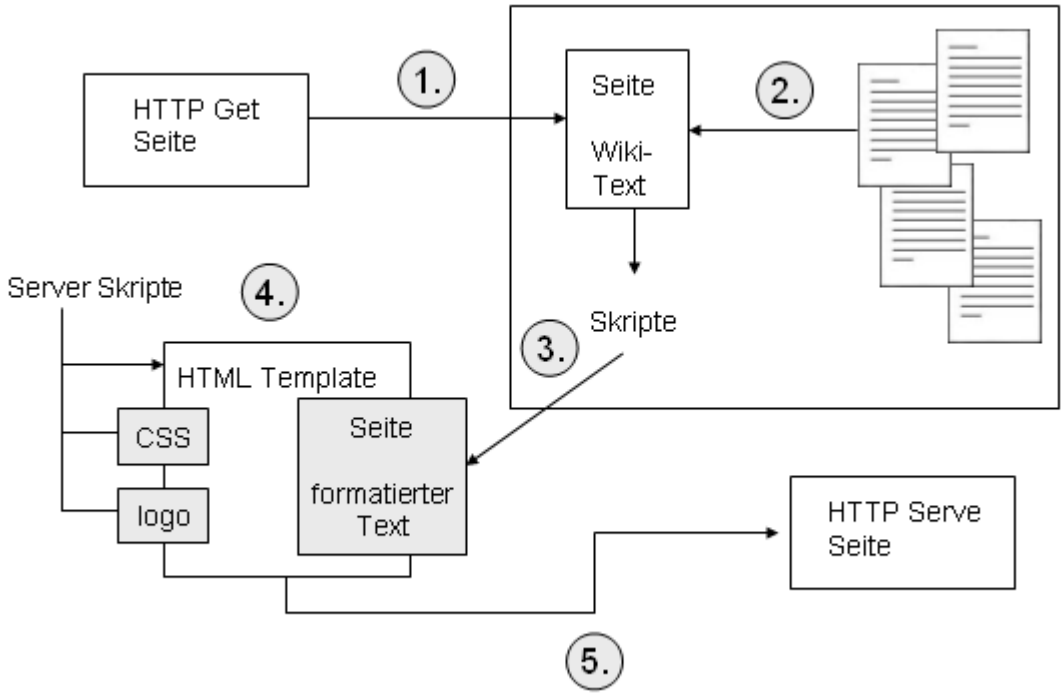


Abbildung 3.5: Content-Model. Quelle: vgl. [CL01]

Die Abbildung 3.5 zeigt, wie eine angeforderte Seite von einem Wiki erstellt wird. (1.) Der Browser erzeugt einen HTTP-Request (Get) und übergibt die angeforderte Seite als Parameter an den Wiki-Server. (2.) Die Skripte des Wiki-Servers erzeugen den Inhalt der Seite aus der Datenbasis. In den meisten Fällen wird der Inhalt der Seiten in Datenbanken auf dem Server gespeichert. Es gibt aber auch Wikis, die den Inhalt in Dateien speichern. MediaWiki und TikiWiki zum Beispiel speichern ihre Daten in Datenbanken. MoinMoin und UseMod benutzen jedoch Dateien als Speicher. Aufgrund des Speicheraufwandes der einzelnen Dateien, hat MoinMoin die Anzahl der Seiten auf eine Größe von 100.000 Seiten beschränkt (vgl. [wik06f]). Der Inhalt der Seiten wird als Wiki-Syntax gespeichert. (3.) Skripte generieren aus den Daten in der Wiki-Syntax eine HTML-Seite. Diese HTML-Seite enthält neben dem Inhalt noch Elemente wie das Navigationsmenü, Inhaltsverzeichnis, Kopf- und Fußnoten usw. (4.) Wie die endgültige Seite aussehen soll, definieren Templates (Seitenvorlagen). Für das Layout, d.h. wie der

Inhalt einer Wiki-Seite dargestellt wird, sind Cascading Style Sheets[9] verantwortlich. Neben der Festlegung der Formatierung des Inhalts, wird das Logo in diesem Schritt hinzugefügt. (5.) Die generierte HTML-Seite wird anschließend an den Browser zurück geschickt (Serve).

Wie man anhand des Content-Models erkennt, entspricht die Wiki-Architektur einer klassischen Client-Server-Architektur. Durch den Aufruf des Browsers (Client) werden die Dokumente vom Wiki-Server (Server) erstellt und an den Browser zurückgesendet.

3.5.2 Architektur von Wiki-Systemen

Der am häufigsten verwendete Web-Server im Internet ist der Apache-Web-Server. Neben dem Apache-Server gibt es noch eine Reihe anderer Webserver wie den Internet Information Services (IIS) von Microsoft, Tomcat oder Cherokee Webserver. Auf dem Webserver müssen die Wiki-Inhalte in geeigneter Form abgelegt werden. Bei den meisten Wikis kommen Datenbanken wie MySQL, Oracle, DB2 oder Microsoft SQL Server zum Einsatz. Sie bieten gegenüber dem Speichern in Dateien eine Reihe von Vorteilen. Unter anderem erlauben sie es mehreren Benutzern gleichzeitig Inhalte zu ändern, ohne dabei Inkonsistenzen zu erzeugen und stellen Funktionen bereit, mit denen automatisch die komplette Datenbasis gesichert und wieder eingespielt werden kann.

Der Zugriff auf die Daten in der Datenbank wird durch Skripte erledigt. Diese sind meist in Skriptsprachen wie PHP, Perl oder Python geschrieben. Alle Programmiersprachen bieten Schnittstellen zu den gängigsten Datenbanken. Werden Inhalte geladen, so erzeugen die Wiki-Skripte aus den Daten HTML oder XHTML-Dokumente und schicken diese an den Client. Ist das Dokument beim Client angekommen, ist keine direkte Kommunikation mit der Datenbasis mehr möglich. Da HTML und XHTML Auszeichnungssprachen ohne Zustand sind, müssen Client-Skriptsprachen wie JavaScript eingesetzt werden, um zusätzliche Funktionen realisieren zu können. Die Kommunikation mit dem Web-Server wird über das HTTP-Protokoll ermöglicht.

User Layer	Web Clients
Network Layer	Apache Web-Server
Logic Layer	MediaWiki's PHP scripts
Data Layer	File-System|MySQL Database|Caching System

Tabelle 3.1: Architektur von MediaWiki. Quelle: [med06b]

[9] Abk. CSS.

Die Programmiersprache PHP ist eine Skriptsprache, die hauptsächlich zur Erstellung dynamischer Webseiten oder ganzer Webanwendungen verwendet wird. Wie die Tabelle 3.1 zeigt, ist die MediaWiki-Software in PHP geschrieben und benutzt die MySQL Datenbank. Beide Technologien werden von einer großen Anzahl von Betriebssystemen unterstützt. Als Web-Server wird von MediaWiki der Apache-Web-Server benutzt.

3.5.3 Wikitext und HTML

Wie im Kapitel über die Syntax (2.1.3) beschrieben, wird der Inhalt einer Seite in die Wiki-Syntax eingegeben. Aus der Eingabe des Autors wird von der jeweiligen Software die HTML-Seite erzeugt. In MediaWiki ist zum Beispiel die Wiki-Syntax eines internen und eines externen Links unterschiedlich.
Interner Link: *[[Ziel]]*
Externer Link: *[http://www.ziel.de Beschreibung des Ziels]*.
In HTML ist die Syntax für beide Links gleich * Ziel *. Wie man an diesem Beispiel gut erkennt, kann in der HTML-Seite nicht mehr zwischen einem internen und einem externen Link unterschieden werden. Im Wiki-Text ist dies aufgrund der unterschiedlichen Syntaxen jedoch möglich. Des Weiteren enthält die HTML-Seite noch zusätzliche Informationen wie das Navigationsmenü, Kopf- und Fußzeile. In MediaWiki wird darüber hinaus aus den Überschriften ein Inhaltsverzeichnis erstellt. Eine Unterscheidung zwischen den Informationen des Autors und denen, die von der Wiki-Software generiert wurden, ist dann nicht mehr möglich.

3.6 Ausgabeformat

Ein weiteres Ziel der Arbeit ist, dass die Visualisierung der extrahierten semantischen Informationen variabel gehalten werden soll. Außerdem müssen die Daten so bereitgestellt werden, dass sie unabhängig aus welchem Wiki-System sie stammen, weiter verarbeitet werden können. Eine Möglichkeit besteht darin, die Daten als **XML** oder **HTML**-Datei zu liefern. XML ist ein Standard, der von der W3C[10] folgendermaßen definiert wird:

> „Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. “ [w3cb]

[10] Abk. für World Wide Web Consortium. Das W3C ist das Gremium zur Standardisierung von Techniken, die das World Wide Web betreffen (z.B. HTML, XML, CSS usw.).

Der Kerngedanke von XML ist eine strukturierte Form der Informationsmodellierung, die es erlaubt, Daten in einem plattformunabhängigen Format darzustellen und weiterzugeben (vgl. [Amm04]). Die Formatierung der Daten erfolgt durch die Stylesheet-Sprache XSL[11]. Wie die Daten eines XML-Dokuments mit dem XSL-Stylesheet umgewandelt werden, beschreibt XSLT[12]. Mit XSLT ist es möglich eine XML-Datei in verschiedene Anzeigeformate (HTML, SVG, CSV[13], WML[14] usw.) zu transformieren. XSLT ist nicht nur, wie häufig angenommen, eine Sprache für die Darstellung von XML, sondern ein sehr mächtiges Werkzeug für die Manipulation und Formatierung von Daten (vgl. [Amm04]). Der Vorteil von XSLT liegt darin, dass:

- es plattformunabhängig ist,
- es relativ weit verbreitet ist,
- es eine Verarbeitung im Web-Browser ermöglicht,
- eine Standard-Transformation (z.B. in HTML) einfach (deklarativ) zu realisieren ist,
- es nicht nur HTML, sondern beliebige andere Sprachen (SVG, WML, CSV usw.) erzeugen kann,
- es extrem flexibel und mächtig ist.

Im Vergleich zu XML ist es in HTML sehr schwierig auf die semantische Struktur zu schließen. Eine Ausgabe in XML kann Meta-Informationen enthalten, welche die Art der Daten beschreiben. [AS98] illustriert dies am Beispiel eines Links auf ein Buch. Ein Link besitzt die URL des Linkziels und einen Linktext, der das Dokument beschreibt. In XML ist es durch die selbst definierten Elemente möglich, auf bestimmte Informationen wie z.B. nur auf den Titel *<Titel></Titel>* des Buches zuzugreifen.

HTML:

```
<a href="Url"> Vorname, Name: Titel </a>
```

XML:

```
<Buch>
    <Url>Url</Url>
    <Autor>Vorname, Name</Autor>
    <Titel>Titel des Buches</Titel>
</Buch>
```

[11] Abk. für Extensible Stylesheet Language.
[12] Abk. für Extensible Stylesheet Language Transformation. Weitere Informationen über XSLT kann auf der Seite des W3C nachgelesen werden (vgl. [w3cc]).
[13] Abk. für Character (auch Comma) Separated Values.
[14] Abk. für Website Meta Language.

Im Gegensatz zu HTML enthält XML keine Informationen darüber, wie es von einer Anwendung visuell dargestellt werden soll. XML hält die Datenstruktur getrennt von der Visualisierung.

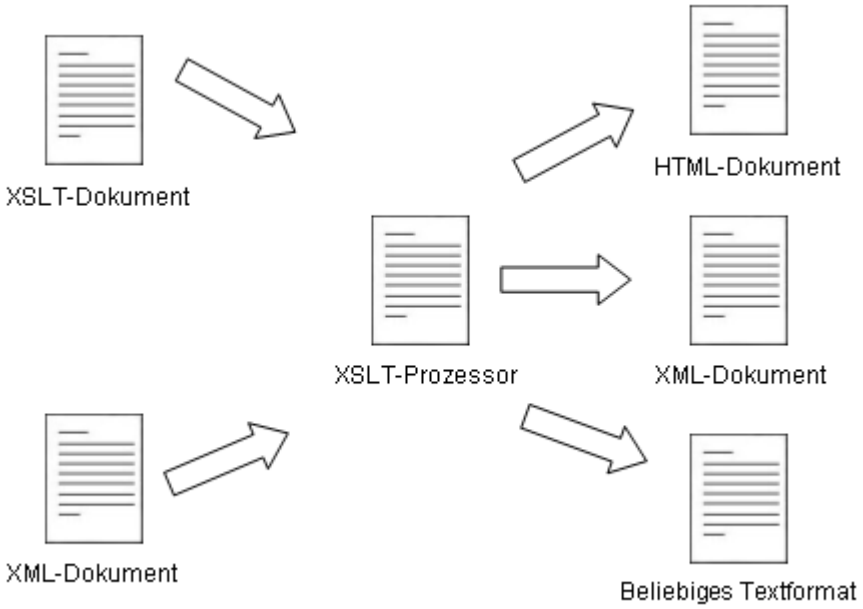


Abbildung 3.6: Formung mit XSLT

Für XML als Standard des W3C[15] gibt es mittlerweile eine ganze Reihe an Bibliotheken, Programmen, Parsern und Prozessoren, die sich an die W3C Empfehlungen halten und somit einen reibungslosen Datenaustausch ermöglichen.

3.7 Anforderungen

Die Analyse der Ausgangssituation hat ergeben, dass in jedem beliebigen Wiki eine Seite über den Namen aufgerufen werden kann. Von dieser Seite ausgehend, lassen sich Informationen extrahieren. Diese Grundvoraussetzung ist bei allen Wiki-Systemen gegeben. Im Unterschied dazu, sind die Anwendung und das Anwendungsszenario nicht bekannt (siehe Zielsetzung 1.2). Aus diesem Grund sind Anpassbarkeit und Modularität die Hauptanforderungen an das Extraktionssystem. Unter Anpassbarkeit ist zu verstehen, dass je nachdem, ob das laufende System über eine Datenbank verfügt, diese als Speicher benutzt werden kann oder auch nicht. In dem Fall, dass keine Datenbank existiert, ist das System trotzdem funktionsfähig. Die Forderung nach Modularität hat den Vorteil, dass

[15]http://www.w3.org. Die offizielle Seite des W3C.

einzelne Systemkomponenten austauschbar sind und das System dadurch besser an eigene Bedürfnisse angepasst werden kann. Im Folgenden werden zuerst die einzelnen Anforderungen aus dem Blickwinkel des Benutzers beziehungsweise der Anwendung und im Anschluss die Anforderungen aus Sicht des Entwicklers definiert.

Das System soll semantische Informationen einer Seite aus einem beliebigen Wiki extrahieren und darüber hinaus von verschiedenen Anwendungen und Rechnern nutzbar sein. Wird eine Seite ausgewählt, die nicht existiert, dann soll das System eine Fehlermeldung ausgeben. Eine Liste aller Seiten innerhalb eines Wikis ist sehr hilfreich. Der Anwendung ist bekannt, welche Wikis als Datenquelle benutzt werden können. Sie ist daher nicht auf die Extraktion aus einem Wiki-System beschränkt. Eine Unterscheidung zwischen der Extraktion der gesamten semantischen oder nur ausgewählter Informationen soll zusätzlich ermöglicht werden. In vielen Fällen ist es schwierig, einer Information automatisch eine Semantik zu geben. Aus diesem Grund muss die Möglichkeit bestehen, dass ein Benutzer die Informationen annotieren kann. Eine Speicherung der semantisch annotierten Informationen soll realisiert werden, damit sie beim nächsten Durchlauf wiederverwendbar sind und nicht neu eingegeben werden müssen. Verifizierte Informationen können daraufhin als sicher gekennzeichnet werden. Der Benutzer erhält die Möglichkeit, extrahierte Informationen zu löschen.

Die Visualisierung der Daten wird nicht festgelegt. Ein Format, das mehrere unterschiedliche Repräsentationen ermöglicht, soll gewählt werden.

Die Einbeziehung eines Wörterbuches ist optional. Das entwickelte System ist somit flexibel an das Anwendungsgebiet anpassbar. Das bedeutet, dass sowohl eine Benutzeraktion oder Speicherung der Daten als auch ein vollständig automatischer Durchlauf notwendig sind.

Aus Entwicklersicht ergeben sich folgende Anforderungen an das System. Das Extraktionssystem kann alle Wiki-Systeme als Datenquelle benutzen. Das erfordert eine Unabhängigkeit von den einzelnen Wiki-Systemen. Für das zu definierende Ausgabeformat gilt, dass es die Visualisierung und Weiterverarbeitung der Daten nicht vorgibt. Weiterhin ist keine Änderung der Ursprungsdaten, d.h. direkt im Quellcode, erwünscht. Der Inhalt und die Struktur bleiben erhalten. Annotationen und Verifikationen des Benutzers sind erwünscht und können in einer Datenbank gespeichert werden. Für statistische Auswertungen wird eine Stopwortliste benötigt. Die Einbindung eines Wörterbuches ist optional möglich. Ein Browser, Netzwerk und evtl. Internet sind Voraussetzung zur Nutzung des Extraktionssystems. Optional kann eine Datenbank als Speicher angebunden werden. Für die Zeichenkodierung ist UTF-8 festgelegt.

Kapitel 4

Konzept

Auf Grundlage der definierten Anforderung in 3.7 wird in diesem Kapitel das daraus resultierende Konzept vorgestellt. Die Unterkapitel behandeln jeweils einen allgemeinen und einen speziellen Teil, der sich auf das MediaWiki bezieht. Unter 4.1 werden die benötigten Systemkomponenten erläutert, die sich aus den Anforderungen ableiten lassen, wie z.B. dass alle Wiki-Systeme als Datenquelle nutzbar sind (Wrapper 4.1.2). Eine Reihe von Aufgaben sind unabhängig vom verwendeten Wiki-System. Aus diesem Grund ist eine Zwischenschicht (Mediator 4.1.3) zwischen Anwendung und Wiki-System nötig. Die Gesamtarchitektur, d.h. wie das System aus seinen Systemkomponenten zusammengesetzt ist und wie diese in Beziehungen stehen, wird in 4.2 behandelt. Die Unterkapitel 4.3.1 und 4.3.2 beschäftigen sich mit den Informationen, die extrahiert werden können, ihrer Semantik und der Definition von Relationen. Mit der Bewertung, ob eine Aussage sicher oder unsicher ist, befasst sich das Kapitel 4.3.3. Eine Zielsetzung der Arbeit (siehe 1.2) ist die Entwicklung von Algorithmen und die Aufstellung von Heuristiken zur Extraktion von Informationen. Wie die definierten Algorithmen und Heuristiken im Detail aussehen, wird in 4.4 beschrieben.

4.1 Umsetzung der Anforderungen

4.1.1 Kommunikation und Ausgabeformat

Eine Hauptanforderung an das System ist die absolute Unabhängigkeit von einem bestimmten System und einer bestimmten Software. **XML** zur Kommunikation zwischen Systemkomponenten zu verwenden, hat den Vorteil, dass es fast jede Programmiersprache auf jedem beliebigen System interpretieren und analysieren kann. Die gängigen Programmiersprachen besitzen eine XML-Schnittstelle, die das Einlesen und Verarbeiten der Daten in ein Programm erlaubt. Weitere Vorteile ergeben sich durch die flexiblen Sprachelemente, dem eindeutigen Informationstransfer, der Lesbarkeit der Dokumente und der Trennung von Layout und Daten. (vgl. [Amm04])

Eine weitere Anforderung an das System ist, dass die Visualisierung der Daten variabel sein soll. Auch hier bietet sich XML hervorragend an. Wie im Kapitel

über das Ausgabeformat (3.6) ausführlich diskutiert, hat die Ausgabe als XML-Dokument den Vorteil, dass das Layout von den Daten getrennt ist und somit eine Reihe von verschiedenen Ausgaben erzeugt werden kann. Durch eine uniforme Repräsentation (XML) ist es außerdem möglich, mittels Data-Mining-Techniken eine automatische Analyse und Interpretation der Muster durchzuführen (vgl. [Eik99]).

Innerhalb des XML-Dokuments lassen sich semantische Annotationen des Benutzers oder auch die Ergebnisse aus der Verifikation durch den Benutzer oder durch ein Wörterbuch darstellen. Dies ist möglich, indem man einem Element ein Attribut zuweist. Aus welcher Quelle die extrahierten Daten stammen, steht zum Beispiel im Element *<sitename>*. Diesem Element wird automatisch eine Semantik durch das Attribut *relation="extrahiert-aus"* zugewiesen. Die Beziehung zwischen einer Seite und dem Wiki, aus dem sie stammt, ist eine sichere Information. Deshalb enthält das Element *<sitename>* zusätzlich das Attribut *result="correct"*. Bei den Wortbeziehungen (Synonyme, Akronyme und Homonyme) gibt die Relation *relation="Synonym, Akronym oder Homonym"* an, in welcher Begriffsrelation zwei Wörter stehen. Das zweite Attribut kann die Werte *possibly-correct*, *correct* oder *incorrect* annehmen, je nachdem, ob es sich um eine Annahme *result="possibly-correct"*, eine sichere *result="correct"* oder unsichere *result="incorrect"* Aussage handelt. Die endgültige XML-Datei hat die Struktur, die Abbildung 4.1 auf Seite 42 zeigt.

Damit die verschiedenen Anwendungen das XML-Dokument interpretieren können, muss die Struktur, d.h. die Elemente und die Attribute einer XML-Datei, definiert sein. Dafür kann eine **DTD**[1] oder ein **XML-Schema** verwendet werden. Worin sich beide Methoden unterscheiden und welche für das Konzept eingesetzt wird, erläutert der nächste Abschnitt.

Eine DTD legt zwar die Struktur eines Dokuments fest, aber nicht den Inhalt. Jedes Dokument, das einer DTD zugeordnet ist, muss genau diesem Aufbau entsprechen. Dabei wird jedes Element, jedes Attribut und jeder mögliche Attributwert einer XML-Datei festgelegt. Eine DTD kann innerhalb einer XML-Datei definiert sein oder auch als externe Datei eingebunden werden. Sie sind wie XML-Dateien plattformunabhängig und können von jedem System gelesen werden. Der Nachteil einer intern definierten DTD ist, dass andere Dokumente nicht darauf zugreifen können.

Das XML-Schema ist eine weitere Möglichkeit ein XML-Dokument zu spezifizieren. Diese neue Idee entstand 1999 in einer Notiz des W3C (vgl. [MM99]). Aufgrund einiger Kritikpunkte an eine DTD wurden neue Ziele festgelegt.

[1] Abk. für Document Type Definition.

Die Kritikpunkte waren:

- eine DTD ist kein XML-Dokument,
- eine DTD hat eine eigene Syntax,
- man kann nur sehr schwammige Aussagen treffen,
- eine genaue Anzahl von Elementen ist nicht definierbar,
- es gibt keine Möglichkeiten Datentypen zu definieren,
- komplexe Datentypen lassen sich nicht erzwingen und kontrollieren,
- der Umgang mit Namensräumen, die über eine DTD so gut wie gar nicht abgebildet werden können.

Für ein neues Schema wurden folgende Ziele festgelegt. Es soll:

- komplett in XML ausgedrückt werden können,
- ausdrucksstärker als eine DTD sein,
- sich selbst dokumentieren,
- netzwerkfähig sein,
- von einem Parser verstanden werden,
- vom Menschen lesbar sein,
- zu anderen Spezifikationen kompatibel sein.

Die aktuelle Entwicklung des XML-Schemas ist auf der Seite des W3C nachzulesen (siehe [w3c06]). Da die Daten in einem XML-Schema genauer definierbar sind als bei einer DTD, wurde dieses zur Beschreibung des erzeugten XML-Dokuments verwendet. Wie in [Amm04] beschrieben, müssen drei Schritte im Wurzelelement des XML-Dokuments vollzogen werden, um ein XML-Dokument mit einem Schema zu verknüpfen:

1. Definition eines Namensraums für das vorliegende Dokument,
2. Definition eines Namensraums für die Verknüpfung,
3. Verknüpfung des Schemas und des XML-Dokuments.

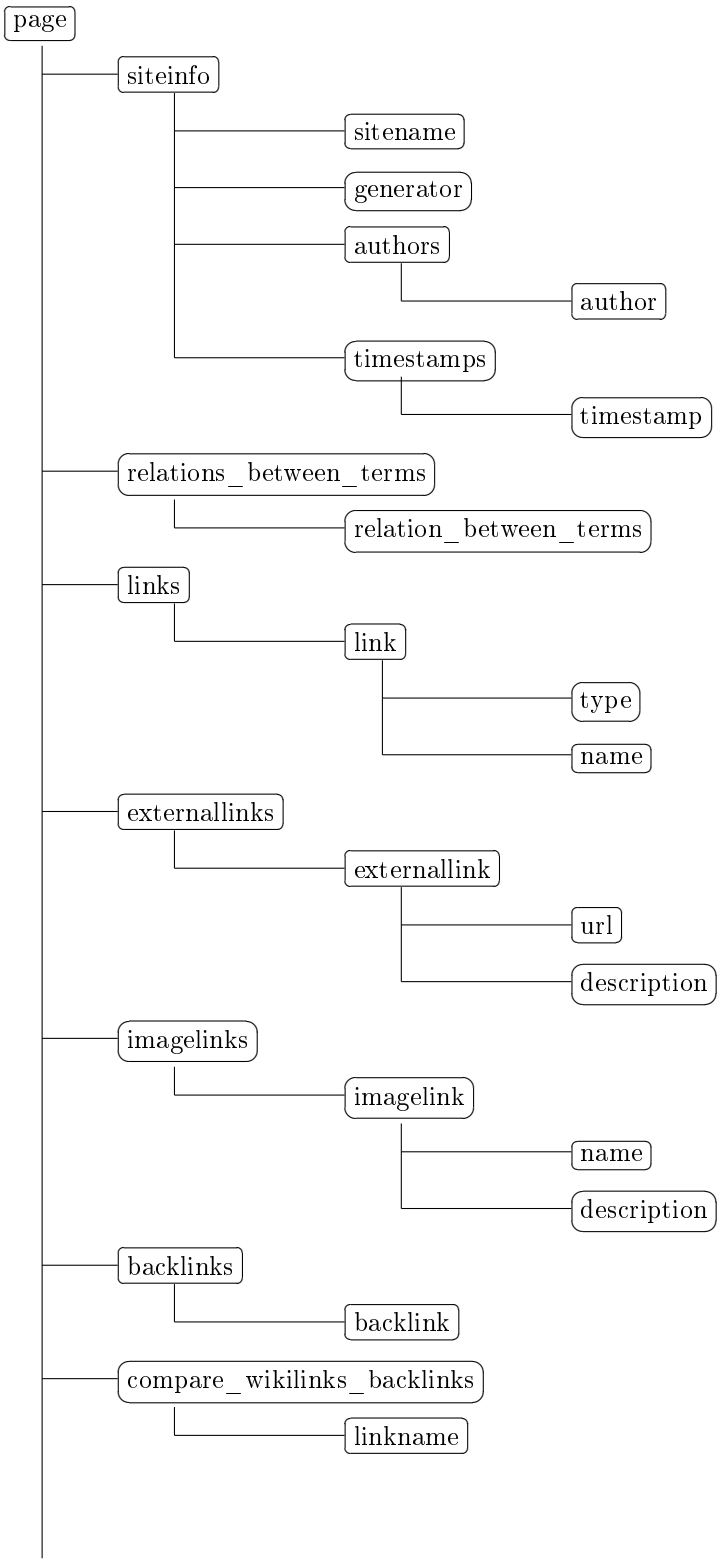
Damit gewährleistet wird, dass die generierten XML-Dokumente (siehe Quellcode 4.1) uniform sind, wurde ein Schema „Output.xsd“ definiert.[2] Da es vorkommen kann, dass bestimmte Informationen in einem Wiki nicht vorhanden sind, wurde eine Reihe der Elemente als optional definiert. Jedes erzeugte XML-Dokument wird gegen dieses Schema validiert. Nur wenn es diesem Schema entspricht, ist es gültig und kann an die Anwendung übergeben werden.

```
<?xml version="1.0" encoding="Utf-8" ?>
<page name="Wiki">
    <siteinfo>
        <sitename relation="extracted-from" result="correct">
            Wikipedia</sitename>
        <generator relation="generated-from" result="correct">
            MediaWiki 1.5.6</generator>
        <authors>
            <author relation="wrote" result="correct">127.0.0.1
                </author>
        </authors>
        <timestamps>
            <timestamp relation="edited-on" result="correct">
                2006-05-08T07:26:49Z
            </timestamp>
        </timestamps>
    </siteinfo>
    <links>
        <link relation="synonym" result="correct">
            <type>Wiki</type>
            <name>WikiWikiWeb</name>
        </link>
        <link relation="belongs-to" result="correct">
            <type>Wikipedia</type>
            <name>WikiNode</name>
        </link>

    </links>
    <externallinks>[...]</externallinks>
    <backlinks>[...]</backlinks>
    <imagelinks>[...]</imagelinks>
    <compare_wikilinks_backlinks>[...]</
        compare_wikilinks_backlinks>
     [...]
</page>
```

Quellcode 4.1: XML-Ausgabedokument: Output.xml

[2] Das Schema für die Ausgabedatei befindet sich im Anhang auf Seite 117.


page
siteinfo
sitename
generator
authors
author
timestamps
timestamp
relations_between_terms
relation_between_terms
links
link
type
name
externallinks
externallink
url
description
imagelinks
imagelink
name
description
backlinks
backlink
compare_wikilinks_backlinks
linkname

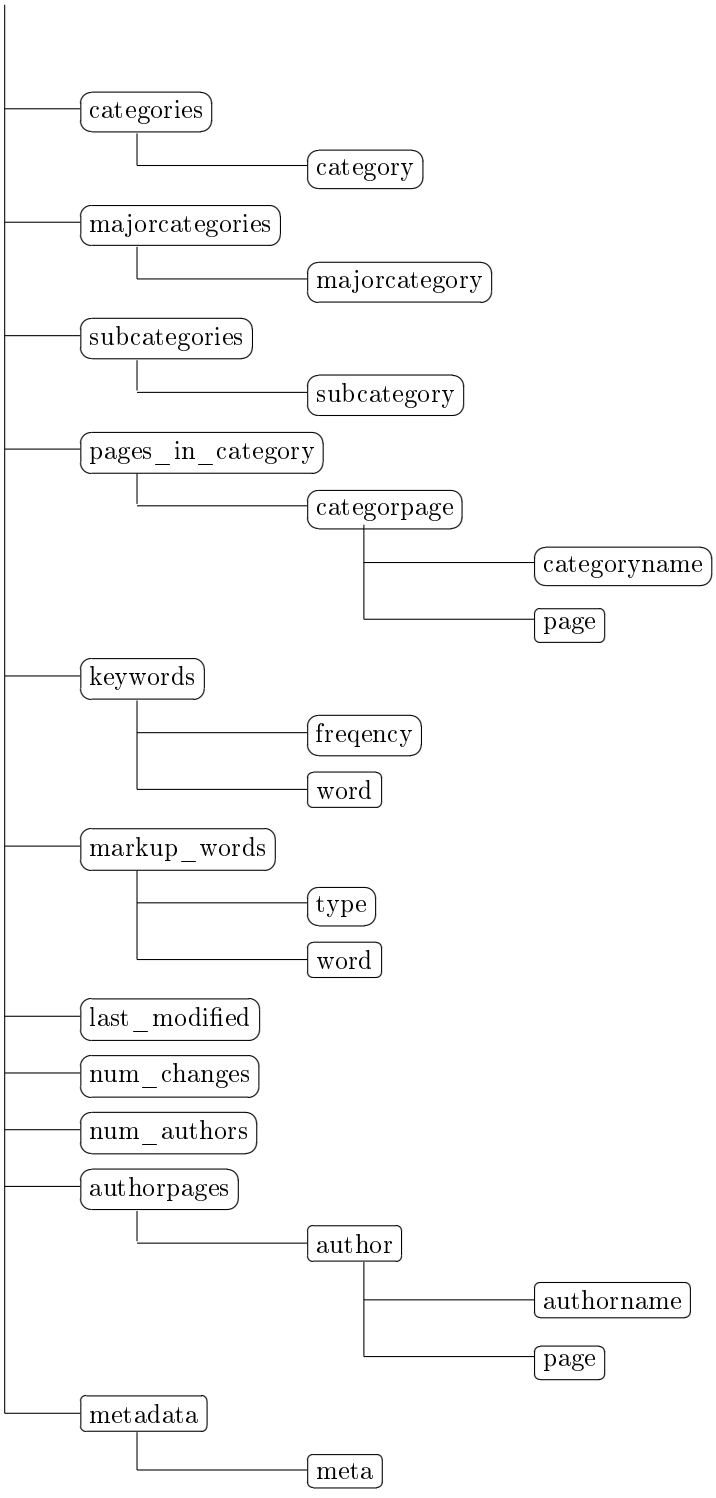


Abbildung 4.1: Struktur der XML-Datei: Output.xml

4.1.2 Wrapper

Eine Anforderung an das zu entwickelnde System ist die Extraktion semantischer Informationen aus einem beliebigen Wiki. Dafür muss eine einheitliche Schnittstelle für die verschiedenen Datenquellen (Wiki-Systeme) geschaffen werden. Wie in 2.2.1 beschrieben, eignen sich **Wrapper** dafür hervorragend. Wrapper werden eingesetzt, wenn aus HTML-Seiten bestimmte Informationen extrahiert und in ein uniformes Format transformiert werden sollen. In den meisten Fällen handelt es sich bei dem uniformen Format um XML-Dokumente. Der Vorteil eines Wrappers ist seine Unabhängigkeit von der Anwendung und des Wiki-Systems. Aufgrund der Unterschiede der Wikis, ist für jedes System ein eigener Wrapper nötig. Die Grundfunktionalität besteht in der Entgegennahme einer Anfrage und der Erzeugung eines XML-Dokuments als Ausgabe. Wie aus der Abbildung 4.2 ersichtlich wird, können für jeden implementierten Wrapper alle Wikis als Datenquelle benutzt werden, die auf dieser Software beruhen. Im Rahmen dieser Arbeit wird ein Wrapper für MediaWiki implementiert. Alle Wikis, die das MediaWiki als Software verwenden, z.B. Wikipedia, WikiBooks oder auch ein eigenes Wiki, können somit als Datenquelle dienen (siehe Abbildung 4.3).

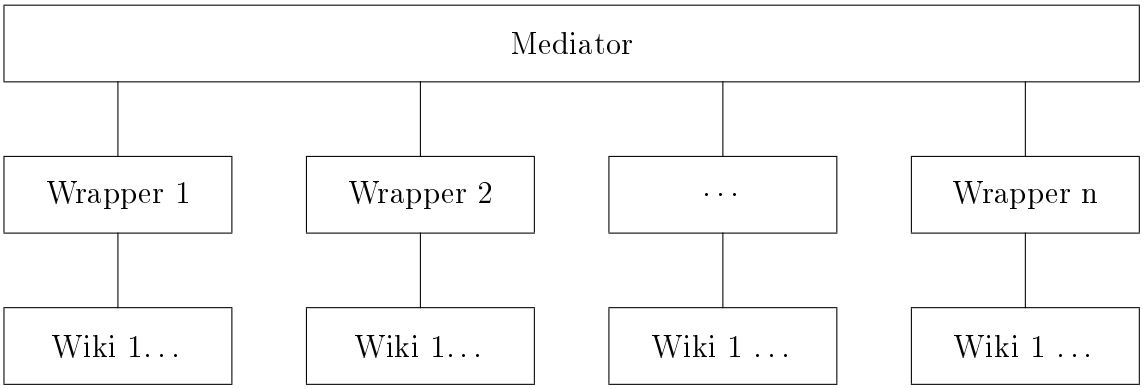


Abbildung 4.2: Mediator-Wrapper-Wiki

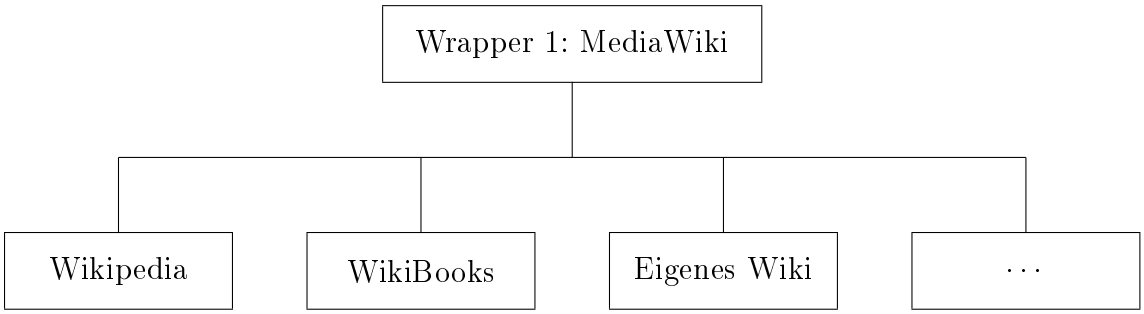


Abbildung 4.3: Wrapper-Wiki

Die ursprüngliche Idee, Informationen direkt aus der Datenbank durch Benutzung existierender Funktionen zu extrahieren, hat sich als nicht vorteilhaft herausgestellt. Einige Funktionen hätten angepasst werden müssen, was die Lauffähigkeit des Wikis unter Umständen gefährdet hätte. Ein Upgrade könnte nach Änderungen im Code zu Konflikten führen. Außerdem können Veränderungen innerhalb einer Funktion Auswirkungen auf andere Funktionen haben. Der Extensions-Mechanismus von MediaWiki ermöglicht es, neue Funktionen in das laufende System zu integrieren, ohne dass der ursprüngliche Code modifiziert wird. Voraussetzung in diesem Fall wäre aber, dass das Wiki-System im selben Netzwerk installiert ist. Damit wäre jedoch nicht die Anforderung der Unabhängigkeit von Anwendung und Wiki gewährleistet.

4.1.2.1 Eingabe

In Kapitel 3.5 wurden die verschiedenen Quellen vorgestellt, aus denen Informationen extrahiert werden können. Die Entscheidung, den Inhalt einer Seite aus einem XML-Dokument zu extrahieren, erfolgte aus mehreren Gründen. Gegen die direkte Extraktion aus der Datenbank spricht, dass ein Wrapper für eine bestimmte Wiki-Software implementiert wird und somit für alle Wikis eines Systems nutzbar ist. Werden Informationen direkt aus der Datenbank extrahiert, muss zunächst eine Verbindung zum Server hergestellt werden. Dies setzt allerdings eine Anmeldeberechtigung, einen Benutzernamen und ein Passwort voraus. Wird das System nur innerhalb eines Netzwerkes verwendet, ist die Beschaffung der Rechte kein Problem. Bei der Verwendung von mehreren Wikis, die im WWW verfügbar sind, ist diese Vorgehensweise nicht geeignet. Es wäre sehr aufwendig bei allen Wikis, die man zur Extraktion von semantischen Informationen benutzen möchte, zuerst die Zugangsdaten und Berechtigungen für jedes Wiki zu erhalten. Damit das System unabhängig vom Wiki bleibt, bietet sich somit die Extraktion aus einer Datenbank nicht an.

Eine Seite wird neu angelegt, indem der Text und die Formatierung in Wiki-Syntax (siehe 2.1.3) eingegeben werden. Daraus erzeugt die Wiki-Software auto-

matisch die HTML-Seite. Der Vorteil, weshalb statt der HTML-Seite der Wiki-Text als Eingabedokument genommen wird, ist in Kapitel 3.5.3 erläutert.

Die Weiterverarbeitung der Daten durch die Anwendung ist variabel. Deshalb ist es wichtig, zu jeder Seite den Namen des Wikis, die Software inklusive Versionsnummer, den Autor und das Datum, an dem die Seite das letzte Mal geändert wurde, ebenfalls zu extrahieren. In MediaWiki gibt es eine Spezialseite, die sich „Seiten exportieren“ nennt. Diese Funktion erzeugt für eine oder auch mehrere Seiten eine XML-Datei mit oder ohne alle vorherigen Versionen. Für jede Version wird das Datum, die Zeit, der Benutzername und eine Bearbeitungszusammenfassung angegeben. Der Quellcode 4.2 zeigt, wie ein XML-Dokument aussieht, das von dieser Funktion erzeugt wurde. Dieses XML-Dokument enthält genau die Informationen, die benötigt werden. Aus diesem Grund wird es als Eingabedokument verwendet.

```
<mediawiki xml:lang="de">
  <siteinfo>
    <sitename>Wikipedia</sitename>
    <base>http://de.wikipedia.org/wiki/Hauptseite</base>
    <generator>MediaWiki 1.8alpha</generator>
    <case>first-letter</case>
      <namespaces>
      [...]
      <namespace key="1">Diskussion</namespace>
      <namespace key="2">Benutzer</namespace>
      [...]
    </namespaces>
  </siteinfo>
  <page>
    <title>Wiki</title>
    <id>5553</id>
    <restrictions>edit=autoconfirmed:move=autoconfirmed</
        restrictions>
    <revision>
      <id>20754338</id>
      <timestamp>2006-08-28T08:35:18Z</timestamp>
      <contributor>
        <username></username>
        <id>221269</id>
      </contributor>
      <minor/>
      <comment></comment>
      <text xml:space="preserve">Ein '''Wiki''', auch
      '''WikiWiki''' und '''WikiWeb''' genannt, ist eine [...]</
          text>
    </revision>
  </page>
</mediawiki>
```

Quellcode 4.2: Spezialseite: Seiten exportieren. Quelle: vgl. [wik06i]

In [wik06i] werden vier verschiedene Methoden zum Export von Seiten vorgestellt.

1. Der Name einer Seite wird in dem Textfeld auf der Spezialseite „Seiten exportieren“ eingegeben oder innerhalb der URL http://Host/Verzeichnis/Special:Export/Seitenname übergeben.

2. Eine Liste aller Seitennamen eines Namensraums kann über die Spezialseite „Alle Seiten“ erhalten werden. Das Backup-Skript „dumpBackup.php“ schreibt alle ausgewählten Wiki-Seiten in eine XML-Datei und arbeitet nur mit MediaWiki 1.5 oder neueren Versionen. Ein direkter Zugang zum Server muss vorhanden sein, um das Skript auszuführen. Dumps der MediaWiki

Projekte, werden mehr oder weniger regelmäßig unter http://download.wikipedia.org zur Verfügung gestellt.

3. Es gibt ein OAI[3]-PMH[4]-Interface, um regelmäßig Seiten zu holen, die zu einem spezifischen Zeitpunkt geändert worden sind. Für Wikimedia-Projekte ist diese Schnittstelle nicht öffentlich verfügbar.

4. Die Verwendung des Wikipedia Robot Framework.

Eine Anforderung an das System ist die Extraktion von semantischen Informationen einer Seite. Aus diesem Grund eignet sich die erste Methode am besten, in der das XML-Dokument über die URL und den Seitennamen erzeugt wird. Die einzelnen Elemente zeigt die Abbildung 4.4 und diese werden im Folgenden kurz erläutert. Welche Elemente wie oft vorkommen müssen und vom welchen Typ sie sind, wird durch das XML-Schema auf Seite 125 im Anhang genau spezifiziert. Das erzeugte XML-Dokument ist in zwei Teile unterteilt. Der erste Teil beschreibt die Seiteninformationen <*siteinfo*> und der zweite die Seite <*page*>. Der Name des Wikis steht in <*sitename*>. In <*base*> findet man die URL der Hauptseite. Im <*generator*>-Tag steht die MediaWiki Version. Die Fälle bei der Verwaltung der Seitennamen unterscheidet <*case*>. Es kann die Werte first-letter, case-sensitiv oder case-insensitiv annehmen. Die Namensräume werden im <*namespace*>-Tag innerhalb von <*namespaces*> aufgelistet.

Als Eingabe für die Extraktion semantischer Informationen muss der Name des Wikis und die MediaWiki Version bekannt sein (siehe Abbildung 4.1). Aus diesem Grund wird der Inhalt von

- <*sitename*> und
- <*generator*>

benötigt. Das Element <*page*> enthält den Titel <*title*> mit Namensraumpräfix, die ID <*id*> der Seite und falls es Seiteneinschränkungen gibt, das <*restrictions*>-Element. Für jede Version einer Seite (als Standardeinstellung wird nur die aktuellste Version exportiert) gibt es das <*revision*>-Element. Für jede Revision können folgende Informationen angegeben werden: die ID <*id*>, das Datum <*timestamp*> gemäß ISO8601 (siehe [wik06i]) und vom Autor <*contributor*>, der Benutzername <*username*>, die ID <*id*> oder die IP-Adresse <*ip*>. Das <*minor*>-Tag ist ein Markierungszeichen, das <*comment*>-Element bezeichnet Kommentare und letztendlich steht innerhalb der <*text*>-Tags der gesamte Text in Wiki-Syntax.

[3] Abk. für Open Archives Initiative.
[4] Abk. für Protocol for Metadata Harvesting.

Als Eingabe für die Extraktion semantischer Informationen müssen das Datum, an dem die Seite erstellt oder geändert wurde, der Autor und das Text-Element existieren (siehe Abbildung 4.1):

- *<timestamp>*
- *<contributor><username><id>*oder *<ip>*
- *<text>*

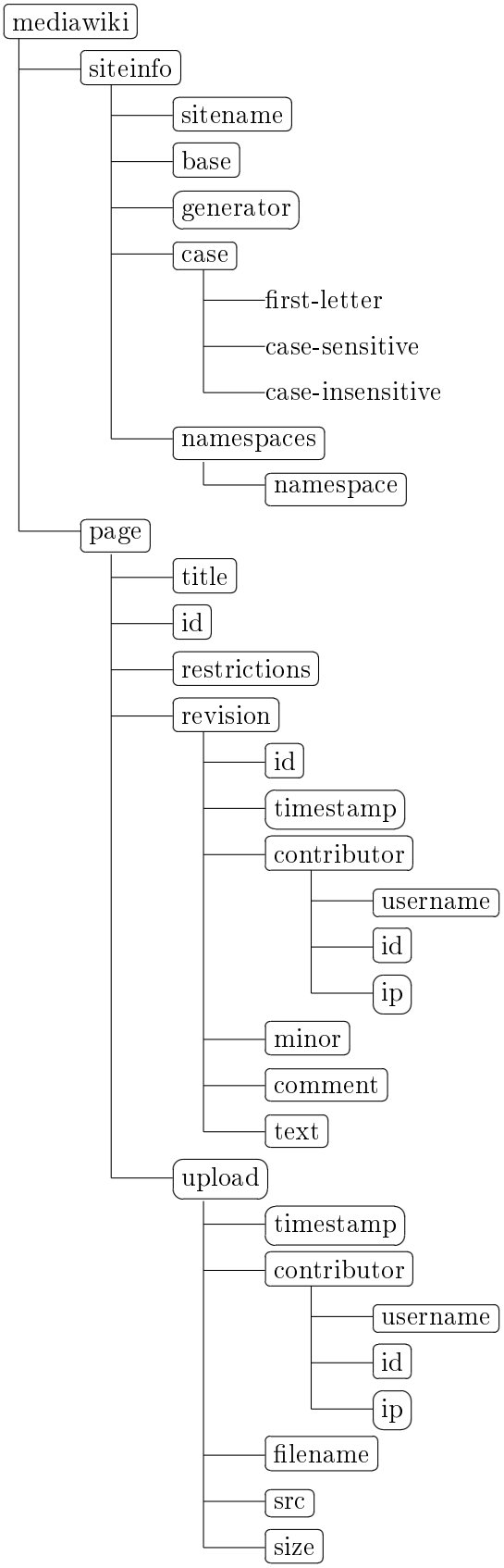


Abbildung 4.4: Struktur der XML-Datei: export-0.3.xml

4.1.2.2 Stopwortliste

Um statistische Auswertungen, wie die Bestimmung wichtiger Wörter einer Seite machen zu können, wird eine Stopwortliste benötigt (siehe 3.4.2). In einer Stopwortliste werden alle Wörter aufgelistet, die nicht in die Auswertung miteinbezogen werden sollen. Sie existiert für fast alle Sprachen. Statt eine bestehende Stopwortliste aus dem WWW zu verwenden, bietet es sich an, eine eigene Liste zu erstellen. Diese ermöglicht durch die Ergänzung und Löschung von Wörtern die anwendungsbezogene Anpassung. Bei weiteren Auswertungen werden diese Wörter nicht mehr mit einbezogen. Eine Stopwortliste befindet sich im Anhang auf Seite 116.

4.1.3 Mediator

Der **Mediator** hat die Aufgabe zwischen der Anwendung und dem Wrapper zu vermitteln. Aus der Anforderung heraus, dass das Gesamtsystem an das Anwendungsszenario anpassbar sein soll, muss der Mediator sehr flexibel sein. Aus diesem Grund wird zwischen festen[5] und optionalen[6] Aufgaben unterschieden. Eine weitere Unterscheidung erfolgt aufgrund der Anforderung, dass eine Benutzeraktion erwünscht sein kann oder auch nicht (automatischer oder halb-automatischer Modus). Anhand der Abbildung 4.5 werden in den nächsten Abschnitten die einzelnen Aufgaben näher erläutert.

Die Anwendung soll vom Wiki-System, das als Datenquelle benutzt wird, unabhängig sein. Sie kennt nur den Namen des Wikis, aus dem die Informationen extrahiert werden sollen. Für die Verwaltung der verschiedenen Wiki-Systeme ist der Mediator verantwortlich. Dafür verwendet er eine Konfigurationsdatei (siehe 4.1.3.1).

[5] In der Abbildung 4.5 die Punkte mit weißem Hintergrund.
[6] In der Abbildung 4.5 die Punkte mit grauem Hintergrund.

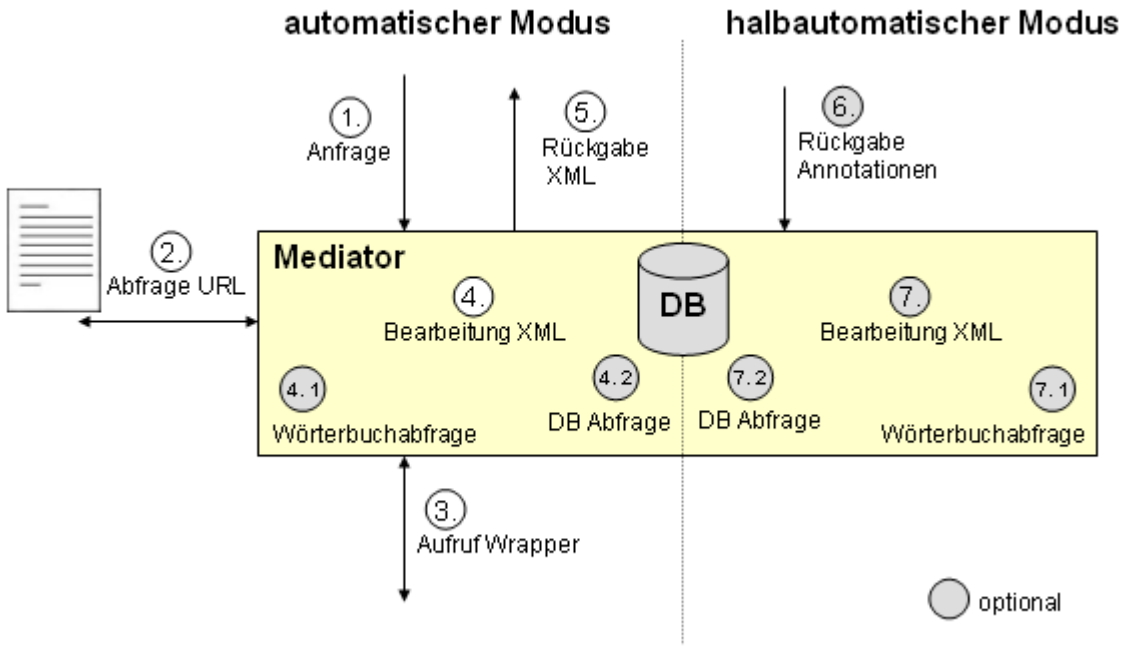


Abbildung 4.5: Aufgaben des Mediators

Die fünf festen Aufgaben des Mediators sind:
1. Zuerst nimmt der Mediator die Anfrage der Anwendung entgegen. 2. Aus der Konfigurationsdatei wird die URL des Wikis abgefragt. 3. Nach Kenntnis der URL wird der zuständige Wrapper aufgerufen. 4. Das vom Wrapper erzeugte XML-Dokument wird anschließend im Mediator bearbeitet. Der Mediator bildet zum Beispiel für die Bestimmung der Seiten, die sich gegenseitig verlinken, die Schnittmenge aus den Links (*<name></name>*) und den Backlinks (*<backlink></backlink>*).

```
<links>
    <link relation="" result="">
        <type>wikilink</type>
        <name>Seite A</name>
    </link>
    <link>
        <type>wikilink</type>
        <name>Seite B</name>
    </link>
</links>
```

```
<backlinks>
  <backlink relation="" result="">Seite A</backlink>
</backlinks>
```

Wenn zwei Seiten sich gegenseitig verlinken, erzeugt der Mediator ein zusätzliches Element *<compare_wikilinks_backlinks>*. Dieses Element wird dem XML-Dokument hinzugefügt.

```
<compare_wikilinks_backlinks>
    <linkname relation="" result="">Seite A</link>
</compare_wikilinks_backlinks>
```

Bei den Wortrelationen überprüft der Mediator, ob einer von den beiden Begriffen nur aus Großbuchstaben besteht. In dem Fall, kann es sich um ein Akronym handeln und das zugehörige Element bekommt die Relation *relation="acronym"* und die Bewertung *result="possibly-correct"* zugewiesen.

```
<relations_between_terms>
    <relation_between_terms relation="acronym"
      result="possibly-correct">
        <term1>DOM</term1>
        <term2>Document Object Model</term2>
    </relation_between_terms>
</relations_between_terms>
```

Diese Aufgaben werden vom Mediator und nicht vom Wrapper durchgeführt, da sie unabhängig vom Wiki-System sind. (5.) Das erzeugte XML-Dokument wird letztendlich an die Anwendung zurückgegeben.
Alle fünf Schritte werden immer ausgeführt, unabhängig davon in welchem Modus (automatisch oder halb-automatisch) das System läuft .

Im automatischen Modus kann der Mediator noch zwei weitere Aufgaben übernehmen: (4.1) die Verwendung eines Wörterbuches zur Verifikation der Synonyme, (4.2) die Abfrage der Daten aus und die Speicherung in eine Datenbank. Wie die Tabelle 4.1 zeigt, müssen vier Fälle[7] unterschieden werden.

[7] Ja: wird verwendet; Nein: wird nicht verwendet.

	Datenbank	Wörterbuch
1. Fall:	Nein	Ja
2. Fall:	Ja	Ja
3. Fall:	Ja	Nein
4. Fall:	Nein	Nein

Tabelle 4.1: Datenbank-Wörterbuch Matrix

Fall 1: Falls die Einbindung eines Wörterbuches erwünscht ist, kommt in Schritt (4.) noch die Aufgabe (4.1), die Verifikation, hinzu. In diesem Schritt wird überprüft, ob die Annahme, dass zwei Wörter synonym sind, korrekt ist. Ist die Annahme richtig, so erhält das XML-Element das Attribut *result="correct"* zugewiesen. Bei falscher Annahme wird das Attribut *result="possibly-correct"* in *result="incorrect"* geändert.

```
<relations_between_terms>
    <relation_between_terms relation="synonym"
      result="correct">
        <term1>Wort 1</term1>
        <term2>Wort 2</term2>
    </relation_between_terms>
</relations_between_terms>
```

Die Überprüfung der Annahme, ob von zwei Wörtern eines das Akronym und das andere aus den Teilwörtern besteht oder ob beide Wörter homonym sind, kann von den meisten Wörterbüchern nicht verifiziert werden. Eine ausführliche Erläuterung zur Verwendung eines Wörterbuches wird in 4.1.3.2 gegeben.

Fall 2: Wird die Verwendung einer Datenbank und die Abfrage eines Wörterbuches gewünscht, so wird die Datenbankabfrage (4.2) vor (4.1) ausgeführt. Für alle extrahierten Informationen wird in der Datenbank abgefragt, ob es einen Eintrag gibt. Für gefundene Einträge wird die Semantik aus der Datenbank als Attribut in das XML-Ausgabedokument übernommen. Beispielsweise für zwei Links die definierte Relation.

```
<links>
    <link relation="belongs-to" result="correct">
        <type>Linktyp</type>
        <name>Seitenname</name>
    </link>
</links>
```

Für alle Elemente, die als Annahme die Relation *relation="synonym"* haben und nicht in der Datenbank gefunden werden, wird zusätzlich eine Wörterbuchabfrage

gestartet. Anschließend werden alle durch das Wörterbuch verifizierten Begriffsrelationen in die Datenbank hinzugefügt.

Fall 3: Im dritten Fall wird nur Schritt (4.2) ausgeführt. Um den extrahierten semantischen Informationen eine Semantik zu geben, wird geprüft, ob es einen Eintrag in der Datenbank gibt. Bei gefundenem Eintrag wird die Relation dem jeweiligem Element als Attribut zugeordnet.

Fall 4: Da keine Datenbankabfrage und Verwendung eines Wörterbuches erwünscht sind, muss der Mediator keine zusätzlichen Aufgaben durchführen.

In vielen Fällen ist es schwierig einer Information automatisch eine Semantik zu geben. Aus diesem Grund ist das Wissen des Benutzers gefordert. Durch den halb-automatischen Modus wird dem Benutzer ermöglicht, die extrahierten Informationen zu annotieren. Nachdem im Schritt (5.) die extrahierten Informationen an die Anwendung zurückgegeben wurden, kann der Benutzer diese annotieren (6.). Tritt Fall 4 ein, so wird nur das XML-Dokument mit den annotierten Informationen ergänzt (7.). Abhängig von den drei Fällen, die in der Tabelle 4.1 dargestellt sind, kommen noch die Aufgaben (7.1), die Verwendung eines Wörterbuches zur Verifikation der Synonyme und (7.2), die Speicherung und Abfrage der Daten in einer Datenbank auf den Mediator hinzu. Diese Aufgaben entsprechen denen im automatischen Modus.

4.1.3.1 Konfigurationsdatei

Als Anforderung an das System wurde definiert, dass die Anwendung alle Wiki-Systeme kennt, aus der sie semantische Informationen extrahieren kann. Die Anwendung soll von der expliziten Adresse des Wiki-Systems unabhängig sein. Um diese Anforderungen zu erfüllen, bietet sich eine Konfigurationsdatei an, in der für jedes Wiki der Name und die Adresse gespeichert werden. Die Anwendung muss somit nur den Namen des Wikis, aus dem die Daten extrahiert werden sollen, an den Mediator übergeben. Der Mediator kann die Adresse (URL) über den Namen des Wikis in der Konfigurationsdatei finden.

Als Konfigurationsdatei bietet sich beispielsweise eine XML-Datei an, weil sie eine einheitliche Syntax und gleichzeitig alle nötigen Werkzeuge besitzen, um sie zu lesen und zu interpretieren (vgl. [Amm04]). SAX[8] ist eine standardisierte Methode, ein XML-Dokument durch einen Parser zu bearbeiten. DOM[9] ist ebenfalls eine Möglichkeit XML-Dokumente auszuwerten und wurde vom W3C standardisiert (vgl. [w3ca]). Für ein einfaches Arbeiten mit XML reicht SAX aus. Für

[8] Abk. für Simple API for XML.
[9] Abk. für Document Object Model.

nachträgliche Manipulationen des XML-Dokuments ist SAX zu eingeschränkt.

Wie eine Konfigurationsdatei aussehen kann, zeigt der Quellcode 4.3. Damit der Wrapper unterschiedliche Seiten (dafür ist nur der Host und das Verzeichnis wichtig) aufrufen kann, wird die URL in drei Teile zerlegt:

- in den Host *<host>*,
- in das Verzeichnis *<url>*,
- in die Funktion, die das XML-Eingabedokument erzeugt *<xml>*.

Der Mediator kann in der Konfigurationsdatei über den Namen des Wikis *<name>* die URL aus dem Inhalt zwischen den Tags *<host>*, *<url>* und *<xml>* erstellen. Wird ein neuer Wrapper für ein Wiki-System implementiert, müssen alle neu hinzugekommenen Wikis in die Liste aufgenommen werden. So können die Anwendungen auch diese zusätzlichen Quellen nutzen.

```
<wikis>
  <wiki>
      <name>Wiki 1</name>
      <host>Host</host>
      <url>Verzeichnis</url>
      <xml>Funktion</xml>
  </wiki>
  <wiki>
      <name>Wiki 2</name>
      <host>Host</host>
      <url>Verzeichnis</url>
      <xml>Funktion</xml>
  </wiki>
    <wiki>
      <name>Wiki 3</name>
      <host>Host</host>
      <url>Verzeichnis</url>
      <xml>Funktion</xml>
  </wiki>
</wikis>
```

Quellcode 4.3: Konfigurationsdatei

4.1.3.2 Wörterbuch

Die Anforderungen an das Wörterbuch sind:

1. Die Verifikation der extrahierten Informationen. Dies ist nur bei den Synonymen, Akronymen, Homonymen und bei Ober- und Unterbegriffen möglich.

2. Die Bildung der Grundform eines Wortes.

Im Internet gibt es eine Reihe von Online-Wörterbüchern. Die Abfrage eines Wortes kann über HTTP oder auch über einen Web-Service erfolgen.

Beim Wortschatzprojekt der Universität Leipzig[10] ist, über Webservices aus einer beliebigen Software heraus, ein direkter Zugriff auf die Daten möglich. Dazu gehören nicht nur Java-Programme, sondern auch Programme beliebiger anderer Sprachen, da die Schnittstelle (SOAP[11]) standardisiert und offengelegt ist.

„Web Services ist der Oberbegriff für eine Sammlung aus den drei Technologien: SOAP, UDDI[12] und WSDL[13]. Web Services erlauben das maschinelle Auffinden und Nutzen von Services, bei denen es sich in der Regel um Softwaremodule handelt. Diese sind in Verzeichnissen (UDDI) beschrieben. Die Kommunikation bei der Suche und der Nutzung wird über das SOAP-Protokoll abgewickelt. Die Services (Funktionen) und deren Parameter werden in der WSDL-Sprache beschrieben.“ [gal06].

Die Tabelle 4.2 zeigt einen Überblick der angebotenen Webservices des Wortschatzprojektes. Die grau hinterlegten Zeilen kennzeichnen genau die Services, die man für die oben definierten Anforderungen an das Wörterbuch benötigt. Jedoch müssen die Nutzungsbedingungen beachtet werden:

„Die vom Projekt Deutscher Wortschatz zur Verfügung gestellten Daten sind urheberrechtlich geschützt. Sie werden zur Nutzung für private und wissenschaftliche Zwecke in angemessenem Rahmen unentgeltlich zur Verfügung gestellt. Jede über die im WWW bereitgestellten Abfragemöglichkeiten hinausgehende Nutzung, automatisierte Abfragen sowie eine kommerzielle Nutzung, Weiterverarbeitung oder Speicherung der Daten sind ohne ausdrückliche schriftliche Zustimmung der Projektleitung untersagt.“ [wor06b]

[10] http://wortschatz.uni-leipzig.de/.

[11] Abk. für Simple Object Access Protocol. Plattformunabhängiges Kommunikationsprotokoll bei Web Services.

[12] Abk. für Universal Description Discovery and Integration. UDDI ist ein Verzeichnisdienst.

[13] Abk. für Web Services Description Language. Definiert eine plattform-, programmiersprachen- und protokollunabhängige XML-Spezifikation zur Beschreibung von Netzwerkdiensten (Web Services) zum Austausch von Nachrichten.

Cooccurrences	Liefert die als statistisch signifikant errechneten Kookkurrenzen (gemeinsames Vorkommen sprachlicher Elemente in derselben Umgebung, z.B. in einem Satz).
Grundform	Liefert zu einem Wort die Grundform sowie die Wortklasse.
SentencesToWord	Liefert zu einem Wort Beispielsätze.
RightOccurrences	Gibt zu einem Eingabewort die als statistisch signifikant errechneten rechten Nachbarn an.
LeftOccurrences	Gibt zu einem Eingabewort die als statistisch signifikant errechneten linken Nachbarn an.
Frequency	Gibt die Frequenz sowie die Häufigkeitsklasse eines Wortes an.
Synonyms	Liefert zu einem Eingabewort die Synonyme.

Tabelle 4.2: Überblick über die Webservices des Wortschatzprojektes. Quelle: vgl. [wor06a]

Ein weiteres Wörterbuch im WWW ist das Wictionary[14], ein auf MediaWiki basierendes Wiki. Es bietet keine direkte Schnittstelle an. Über eine HTTP-Anfrage und der Übergabe des Wortes als Parameter wird das generierte HTML-Dokument zurückgesendet. Dieses wird anschließend geparst. Wie das Ergebnis nach der Suche „Wiki“ aussieht, zeigt die Abbildung 4.6. Im Gegensatz zur Suche im Wortschatzprojekt muss das Wort in der Grundform vorliegen, d.h. für die Bildung der Grundform muss zuerst eine andere Quelle herangezogen werden. Für die Verifikation lassen sich in diesem Wörterbuch zu einem Wort die Synonyme, Ober- und Unterbegriffe finden.

[14] Wictionary ist ein frei verfügbares, mehrsprachiges Wörterbuch für den Wortschatz aller Sprachen. Es existiert seit dem 1. Mai 2004 und umfasst derzeit 20.882 Einträge. http://de.wiktionary.org/wiki/Wiktionary:Hauptseite. Stand 04.06.2006.

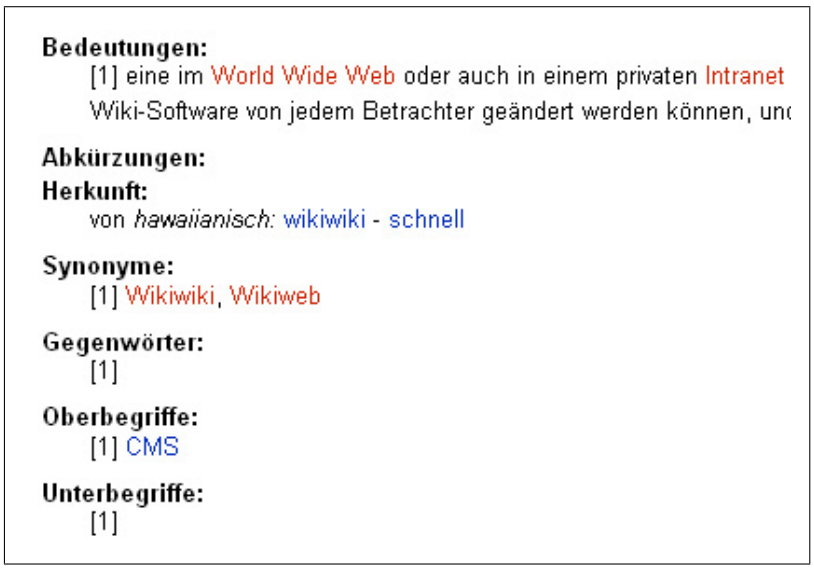


Abbildung 4.6: Screenshot aus dem Wörterbuch: Wictionary

Ob ein Wörterbuch bei der Extraktion der semantischen Informationen mit eingebunden wird, ist von der Anwendung abhängig. Ein Wörterbuch kann sowohl im automatischen als auch im halb-automatischen Modus verwendet werden. Ist eine Verifikation mit einem Wörterbuch erwünscht, wird der Parameter *Dictionary=Yes* dem Mediator übergeben. Als Standardeinstellung (default: *Dictionary=No*) wird keine Wörterbuchabfrage durchgeführt.

Bei zwei Begriffen wird in einem Wörterbuch der erste Begriff gesucht und überprüft, ob der zweite genau der Annahme (Synonym, Akronym usw.) entspricht. Nachdem die Begriffsrelation verifiziert wurde, wird dem entsprechendem Element das Attribut *result="correct"* zugewiesen. Wird ein Eintrag nicht gefunden, bleibt die Annahme *result="possibly-correct"* erhalten.

Vor der Verifikation:

```
<relations_between_terms>
    <relation_between_terms relation="synonym"
      result="possibly-correct">
        <term1>Computer</term1>
        <term2>Rechner</term2>
    </relation_between_terms>
    <relation_between_terms relation="synonym"
      result="possibly-correct">
        <term1>Computer</term1>
        <term2>Informatik</term2>
    </relation_between_terms>
```

```
    <relation_between_terms relation="acronym"
      result="possibly-correct">
        <term1>PC</term1>
        <term2>Personal Computer</term2>
    </relation_between_terms>
</relations_between_terms>
```

Nach der Verifikation:

```
<relations_between_terms>
    <relation_between_terms relation="synonym"
      result="correct">
        <term1>Computer</term1>
        <term2>Rechner</term2>
    </relation_between_terms>
    <relation_between_terms relation="synonym"
      result="incorrect">
        <term1>Computer</term1>
        <term2>Informatik</term2>
    </relation_between_terms>
    <relation_between_terms relation="acronym"
      result="correct">
        <term1>PC</term1>
        <term2>Personal Computer</term2>
    </relation_between_terms>
</relations_between_terms>
```

4.1.3.3 Speicherung der Daten

Damit die extrahierten und annotierten Informationen nicht verloren gehen und für die nächsten Abfragen wieder zur Verfügung stehen, ist die Speicherung der Daten sinnvoll. Folgende Informationen sollen gespeichert werden:

1. Die Relationen, die zwischen zwei Seiten existieren können (Seite 1, Seite 2, Semantik).

Der Benutzer definiert eine Semantik zwischen zwei Seiten. Ohne Speicherung der Daten ist diese Information für die nächsten Durchläufe nicht mehr verfügbar und müsste neu eingegeben werden. Arbeitet eine Anwendung in einem bestimmten Kontext, so sind die Relationen, die auftreten können, immer ähnlich. Aus diesem Grund ist eine zusätzliche Speicherung der Relationen, die dem Benutzer zur Auswahl stehen können, nützlich. Der Vorteil dabei ist, dass er die Relation nicht immer neu eingeben muss, sondern aus einer Liste auswählen kann. Das System ist somit an eigene Bedürfnisse anpassbar.
Als Speicherung dieser Informationen eignet sich eine Datenbank. Es sind die

Tabellen 4.3 und 4.4 zur Speicherung der semantischen Informationen und der Relationen notwendig.

Tabelle:	**information**			
Feld:	term1	term2	type	relation

Tabelle 4.3: Speicherung der semantischen Informationen

Tabelle:	**relation**
Feld:	name

Tabelle 4.4: Speicherung der Relationen

Die Anwendung übergibt dem Mediator als Parameter *DB=Yes/No*, ob eine Speicherung erwünscht ist oder nicht. Für die Speicherung der Informationen (Seite 1, Seite 2, Semantik) sind vier Einträge notwendig (siehe Tabelle 4.3). Zum einen der Name der Seite, von der der Link ausgeht (term1) und zum anderen der Name der Seite, auf die verlinkt wird (term2). Da der Name einer Seite nicht eindeutig ist, wird im Feld „type“ als dritter Eintrag der Namensraum der Seite gespeichert. Der vierte Eintrag „relation“ enthält die Semantik des Links. In der Tabelle 4.4 sind als Standardeinstellung eine Reihe von vordefinierten Relationen eingetragen. Diese stehen dem Benutzer zur Auswahl. Eine Auflistung aller Relationen zeigt die Tabelle 4.8 auf Seite 68.

2. Nach der Verifikation durch ein Wörterbuch oder durch den Benutzer sollen sichere Informationen zwischen Wörtern wie Synonyme, Akronyme und Homonyme gespeichert werden.

Zwischen zwei oder auch mehreren Wörtern können Relationen (siehe 2.4.2) bestehen. Wird das Tripel (Begriff 1, Begriff 2, Relation) gespeichert, so kann auf diese Information auch bei weiteren Durchläufen zugegriffen werden.
Für jede Begriffsrelation gibt es eine Tabelle, in der das Tripel gespeichert wird. Je nachdem, in welcher Tabelle sich zwei Begriffe befinden, kann man feststellen, um welche Relation es sich handelt.

Tabelle:	**synonym**	
Feld:	name1	name2

Tabelle 4.5: Speicherung der Begriffsrelation (Synonym)

Tabelle:	**acronym**	
Feld:	name1	name2

Tabelle 4.6: Speicherung der Begriffsrelation (Akronym)

Tabelle:	**homonym**	
Feld:	name1	name2

Tabelle 4.7: Speicherung der Begriffsrelation (Homonym)

3. Das XML-Dokument mit allen extrahierten Informationen soll ebenfalls gespeichert werden.

Da sich der Inhalt einer Seite innerhalb kürzester Zeit ändern kann, sollte der Anwender entscheiden, ob die semantischen Informationen direkt aus der Datenbank geholt oder erneut aus dem Wiki extrahiert werden sollen. Um die Aktualität des XML-Dokuments zu gewährleisten, könnten sich die Seiten innerhalb der Datenbank regelmäßig aktualisieren. Durch die Speicherung der Semantik zwischen Seiten und Worten in separaten Tabellen, gehen auch die verifizierten und annotierten Informationen nicht verloren. Neu hinzukommende Informationen erhalten nur eine Semantik, wenn diese automatisch zugewiesen werden kann.

Für die Speicherung der Relationen zwischen zwei Seiten und den Begriffsrelationen bietet sich, wie schon erwähnt, eine Datenbank an. In den jeweiligen Tabellen (information, synonym, acronym und homonym) werden die Daten gespeichert und stehen als Informationsquelle bei weiteren Abfragen zur Verfügung. Das XML-Dokument kann als Datei oder als Text in einer Datenbank gespeichert werden. Die Speicherung als Datei im Dateisystem ist für einzelne Dokumente und kleinere Datenmengen ausreichend. Bei großen Datenmengen ist die Speicherung in relationalen Datenbanken geeigneter, weil im Worst-Case ansonsten eine große Anzahl von Dokumenten verwaltet werden müsste.

Bei der Speicherung in einer Datenbank unterscheidet man zwischen elementbasierter und dokumentbasierter Speicherung. Die elementbasierte Speicherung ordnet die einzelnen Elemente und Attribute aus der XML-Datei den Datenbanktabellen zu. Je nach Verschachtelung der Elemente, müssen diese eventuell verschiedenen Tabellen zugeordnet werden. Jedes Element des XML-Dokuments ist somit auch in der Datenbank vorhanden und kann über die Tabellen gesucht werden. Der Nachteil dieser Methode ist, dass das XML-Dokument dafür in seine einzelnen Elemente zerlegt werden muss. Für die Zusammensetzung des XML-Dokuments wiederum sind eine Reihe von Datenbankzugriffen notwendig, was

sehr aufwendig ist.

Bei der dokumentbasierten Speicherung wird auf diese aufwendige Prozedur verzichtet. Stattdessen wird das gesamte Dokument als Text in einem Tabellenfeld gespeichert. In diesem Fall ist es fraglich, ob überhaupt eine Datenbank benötigt wird. In vielen Anwendungen wird eine Mischform aus element- und dokumentbasierten Speicherung verwendet. Bestimmte Teile des XML-Dokuments werden in einer Datenbank zur Verfügung gestellt, auf die seperat zugegriffen werden kann. Das gesamte XML-Dokument wird dann erneut als Text in der Datenbank gespeichert. (vgl. [Loc02])

4.1.4 Anwendung

Eine **Anwendung** kann entweder eine Desktop-Anwendung (Textverarbeitungsprogramm, E-Mail Programm, Datenbankanwendung usw.) oder eine Web-Anwendung, wie zum Beispiel der Prototyp (siehe 5.1), sein. Desktop-Anwendungen sind lokal auf dem Arbeitsplatzrechner installiert und verfügen über eine eigene Benutzeroberfläche. Eine Web-Anwendung ist ein Computer-Programm, das auf einem Webserver ausgeführt wird, wobei eine Interaktion mit dem Benutzer ausschließlich über einen Browser erfolgt. Hierzu sind der Computer des Benutzers (Client) und der Server über ein Netzwerk, wie das Internet oder über ein Intranet, miteinander verbunden, so dass die räumliche Entfernung zwischen Client und Server unerheblich ist.

Die Bearbeitung von Anfrage und Antwort geschieht on-the-fly. Die Anwendung erzeugt eine HTTP-Anfrage. Innerhalb der URL[15] werden folgende Parameter übergeben:

- *Page=Name*,
- *Wiki=Name*,
- *Modus=automatic/semiautomatic* default: *automatic*,
- *Synonym=0/1*,
- *Acronym=0/1*,
- *Homonym=0/1*,
- *Links=0/1*,
- *Wikilinks=0/1*,

[15] Abk. für Uniform Resource Locator. Eine URL bezeichnet die gesamte Adresse einer Internetseite.

- *Externallinks=0/1,*
- *Backlinks=0/1,*
- *Imagelinks=0/1,*
- *CompareWB=0/1,*
- *Categories=0/1,*
- *MajorCategory=0/1, MajorDepth=Tiefe* default: *Tiefe=3,*
- *SubCategory=0/1, SubDepth=Tiefe* default: *Tiefe=3,*
- *PagesInCategory=0/1,*
- *Keywords=0/1, limit=Vorkommenshäufigkeit* default: *limit=5,*
- *Markup=0/1,*
- *LastModified=0/1,*
- *NumChanges=0/1,*
- *NumAuthors=0/1,*
- *AuthorPages=0/1,*
- *MetaData=0/1,*
- *All=0/1,*
- *Dictionary=Yes/No* default: *No,*
- *DB=Yes/No* default: *No.*

Nach Anforderung soll auf die Anfrage nach einer nicht existierenden Seite eine Fehlermeldung ausgegeben werden. Es gibt zwei Möglichkeiten einen Fehler zu verhindern. Zum einen kann der Anwendung bei Benutzung eines Wikis, das auf MediaWiki basiert, über die Spezialfunktion „Alle Artikel“ alle Seiten eines Wikis zur Verfügung gestellt werden. Zum anderen kann bei Verwendung einer Datenbank eine Liste aller Seiten, die schon bis zu diesem Zeitpunkt aufgerufen wurden, erstellt werden. Den Namen der Wikis, die als Datenquelle benutzt werden können, erhält die Anwendung aus der Konfigurationsdatei. Wenn eine übergebene Variable den Wert 0 hat, wird diese Information nicht extrahiert, bei 1 wird sie es.

Die Weiterverarbeitung des erzeugten XML-Dokuments ist vom Anwendungskontext abhängig. Das XML-Dokument kann in verschiedene Formate (Text-, Grafik- oder Multimediaformate) transformiert werden, zum Beispiel mit einer

XSLT-Datei in HTML, SVG, WML, CSV usw. (siehe Abbildung 3.6). Wenn keine direkte Ausgabe erwünscht ist, können die extrahierten Informationen in einer Datenbank gespeichert werden. Damit ist nicht die Datenbank des Extraktionssystems gemeint, sondern die der Anwendung.

4.2 Gesamtarchitektur

Im Unterkapitel 4.1 wurden die einzelnen Systemkomponenten vorgestellt. Die daraus entstandene Gesamtarchitektur zeigt die Abbildung 4.7 auf Seite 65. Die Architektur wird in vier Schichten unterteilt: die Anwendung, der Mediator, die Wrapper und als unterste Schicht die Wikis. Da die Anwendung und die verschiedenen Wikis gegeben sind, besteht das Extraktionssystem in erster Linie aus dem Mediator und den Wrappern. Der Mediator ist Vermittler zwischen der Anwendung und den Wiki-Systemen. Er nimmt die Anfrage der Anwendung entgegen und ruft den zuständigen Wrapper auf. Für Aufgaben, die unabhängig vom Wiki sind, wie die Verifikation durch ein Wörterbuch oder des Benutzers und die Speicherung der Daten, ist ebenfalls der Mediator zuständig. Der Wrapper bildet eine einheitliche Schnittstelle zwischen den unterschiedlichen Wiki-Systemen und dem Mediator. Sie haben die Aufgaben, die semantischen Informationen aus den Wiki-Systemen zu extrahieren und dabei gleichzeitig das XML-Ausgabedokument zu erzeugen. Im nächsten Schritt wird dieses Ausgabedokument an den Mediator übergeben. Die Anwendung hat demnach lediglich die Aufgabe eine Abfrage zu starten und letztendlich das vom Mediator zurückgegebene XML-Dokument weiter zu verarbeiten.

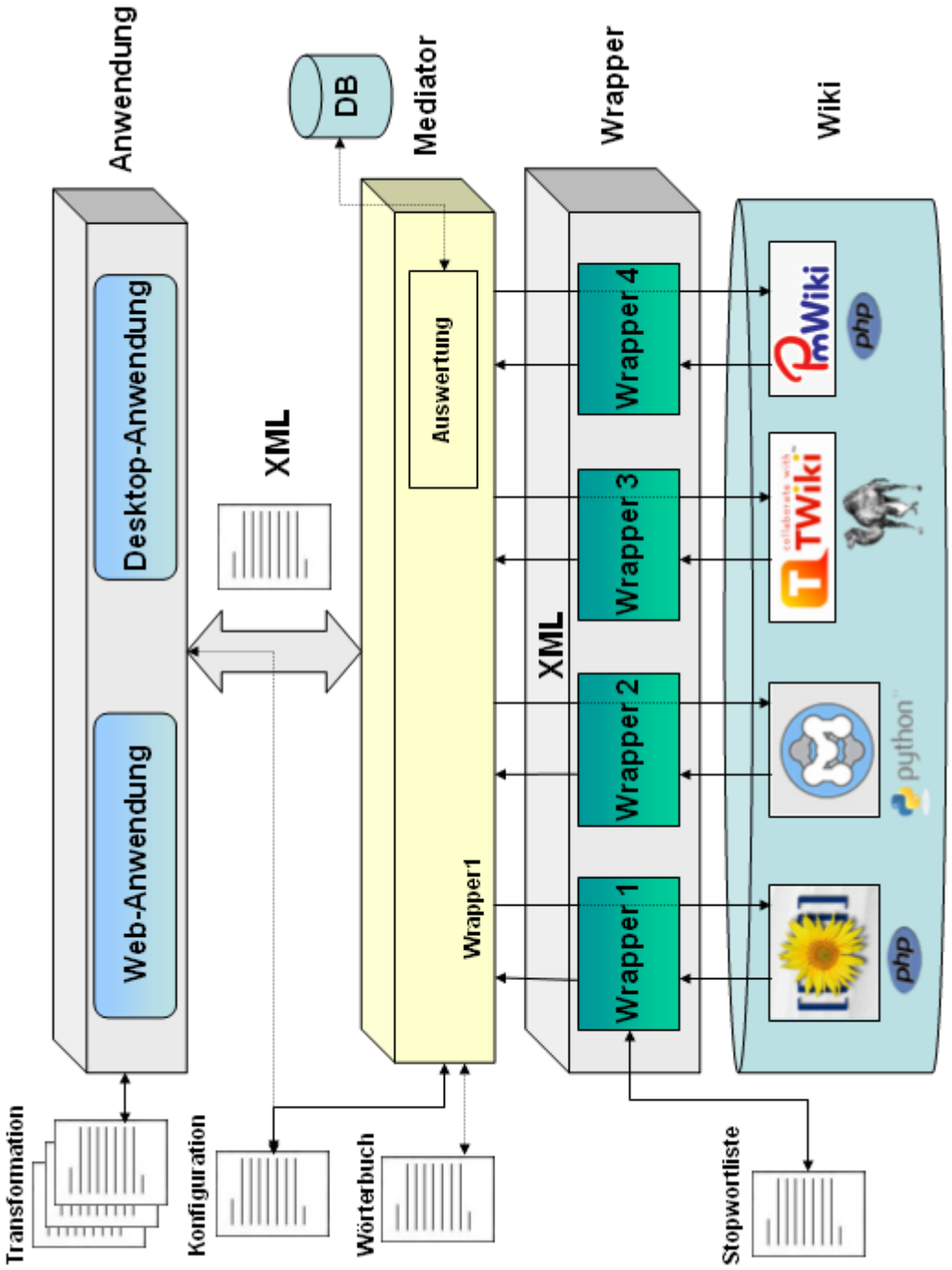


Abbildung 4.7: Gesamtarchitektur

4.3 Seite

4.3.1 Zu extrahierende Informationen

Ziel des Extraktionssystems ist es, die Informationen in ein semantisch annotiertes XML-Dokument zu transformieren.

4.3.1.1 Links

Innerhalb von Wiki-Systemen gibt es eine Reihe von unterschiedlichen Linktypen (siehe 3.3). Durch den Linktyp wird der Seite automatisch eine Semantik gegeben. Eine weitere Semantik kann dem Link über das Attribut „relation“ zugewiesen werden.

Wikilink [[Seite]]

```
<links>
    <link relation="" result="">
        <type>wikilink</type>
        <name>seite</name>
    </link>
</links>
```

[[Präfix:Seite]] Präfix: Portal, Wikipedia, Wikibooks etc.

```
<links>
    <link relation="" result="">
        <type>Präfix</type>
        <name>Seite</name>
    </link>
</links>
```

Externer Link [http://www.ziel.de Beschreibung des Ziels]

```
<externallinks>
    <externallink relation="" result="">
        <url>http://www.ziel.de</url>
        <description>Beschreibung des Ziels</description>
    </externallink>
</externallinks>
```

Bilderlink [[Bild: Beispiel.jpg|Bildbeschreibung]]

```
<imagelinks>
    <imagelink relation="" result="">
        <name>Beispiel.jpg</name>
        <description>Bildbeschreibung</description>
    </imagelink>
</imagelinks>
```

4.3.1.2 Seiten einer Kategorie

Wie in der Analyse (3.3.6) beschrieben, lassen sich Seiten nach ihrem Inhalt in Kategorien einordnen. Die Abbildung 4.8 zeigt, dass man durch die Information zweier Seiten die in der gleichen Kategorie sind, eine Semantik ableiten kann. Die Seite „Informatik“ ist zum Beispiel den zwei Kategorien „Informatik“ und „Wissenschaft“ (grün) zugeordnet. Innerhalb beider Kategorien existieren weitere Seiten wie Aliasing, Mathematik, Wissenschaftsjahr usw. Die Beziehung zwischen der Seite Informatik und Wissenschaftsjahr ist nicht explizit ersichtlich, weil kein Link von Informatik auf das Wissenschaftsjahr verweist. Da das Wissenschaftsjahr 2006 das Informatikjahr ist, besteht somit eine inhaltliche Beziehung zwischen den Seiten. Diese Beziehung lässt sich nur über die Kategorien oder über den Backlink Mechanismus (hellblauer Pfeil) ableiten.

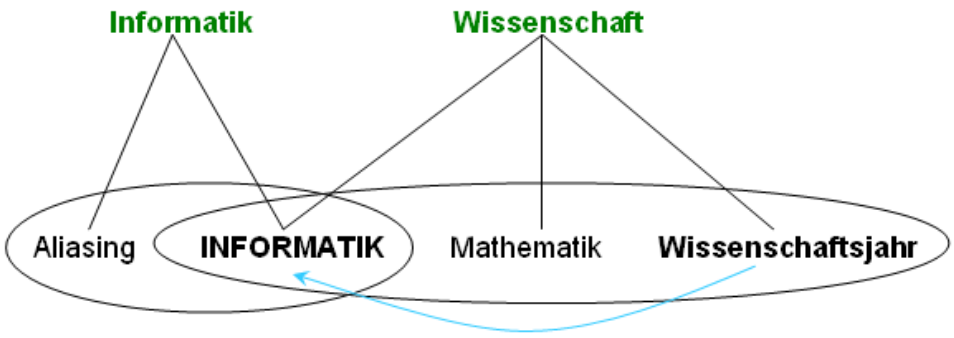


Abbildung 4.8: Seiten einer Kategorie

4.3.2 Definition der Relationen

Um den Informationen eine Semantik zu geben, wird zwischen automatischer Zuweisung oder einer Zuweisung durch den Benutzer unterschieden. Für die automatische Zuweisung sind eine Reihe von Relationen (siehe Tabelle 4.8) vordefiniert. Diese Liste ist erweiterbar.

Information	Quelle	Relation	Übersetzung*
Seite-Wiki	XML	extrahiert-aus	extracted-from
Seite-Wiki	XML	erzeugt-von	generated-from
Seite-Datum	XML	bearbeitet-am	edited-on
Seite-Autor	XML	hat-geschrieben	wrote
Wort-Wort	Wiki-Text	Oberbegriff-von	major-term
Wort-Wort	Wiki-Text	Unterbegriff-von	sub-term
Wort-Wort	Wiki-Text	Synonym-für	synonym
Wort-Wort	Wiki-Text	Akronym-für	acronym
Wort-Wort	Wiki-Text	Homonym-für	homonym
Wort-Wort	Wiki-Text	Teil-von	part-of
Wort-Wort	Wiki-Text	Instrument-für	tool-for
Wort-Wort	Wiki-Text	Ursache-von	reason-for
Wort-Wort	Wiki-Text	gehört-zu	belongs-to

Tabelle 4.8: Definierte Relationen

(*) Die Übersetzung der Relationen ist notwendig, da die Elemente im XML-Dokument auch auf Englisch sind. Als Eingabedokument wird das XML-Dokument verwendet, das von der Funktion „Seiten exportieren“ erzeugt wird. Dort sind alle Elemente auf Englisch definiert. Um konform zu bleiben, werden daher immer alle Elemente und Attribute aus dem Deutschen übersetzt.

Für alle extrahierten Informationen ist es wichtig festzuhalten, aus welchen Quellen (Wiki-Systemen) sie ursprünglich stammen. Bei jedem Extraktionsvorgang kann einer Seite deshalb die Relation *erzeugt-von* und *bearbeitet-am* zugeordnet werden. Die *bearbeitet-am*-Relation bezieht sich nicht auf das Datum, an dem die Daten extrahiert wurden, sondern auf das Datum, an dem die Seite das letzte Mal bearbeitet wurde.
Jedes Dokument wurde von mindestens einem Autor bearbeitet, d.h. zwischen einer Seite und einem Autor besteht eine *hat-Autor*-Beziehung. Daher ist es möglich, alle Dokumente eines Autors, die er geschrieben oder geändert hat, durch den Backlink-Mechanismus zu finden. Dieser Information kann wiederum die *hat-geschrieben*-Beziehung zugeordnet werden.

4.3.3 Bewertung der Informationen

Nach dem Wiki-Prinzip kann jeder Benutzer auf schnelle und einfache Weise Inhalte in ein Wiki-System einfügen. Das Problem hierbei ist, dass es keine Garantie für die Richtigkeit die Inhalte gibt. Dies muss bei der Extraktion von Informationen beachtet werden. Aussagen können nicht immer mit Sicherheit ge-

troffen werden. Die Algorithmen und Heuristiken, die im Detail im Kapitel 4.4 erläutert werden, beschreiben Regeln wie bestimmte Informationen gefunden werden. Informationen, die durch Heuristiken extrahiert werden, sind zunächst nur Annahmen und werden als unsichere Aussage (*result="possibly-correct"*) gekennzeichnet. Erst nachdem diese Informationen von einem Benutzer oder einem Wörterbuch, abhängig von der Information, verifiziert wurden, können sie als sicher (*result="correct"*) oder falsch (*result="incorrect"*) gekennzeichnet werden. Der Benutzer selbst ist für die Richtigkeit seiner annotierten Informationen verantwortlich. Sichere Informationen können in einer Datenbank gespeichert werden. Ein Zugriff auf diese Daten ist in der nächsten Verifikationsphase möglich. Dadurch werden keine zusätzlichen Wörterbuch- oder Benutzerabfragen benötigt. Die Speicherung der verifizierten und annotierten Daten in einer Datenbank hat folgende Vorteile:

1. Wiederverwendbarkeit der Daten,
2. Verringerung der Zugriffszeit.

Der Benutzer kann durch die Annotation die Semantik eines Links explizit angeben. Dazu erhält er bei der Verwendung einer Datenbank eine Liste von möglichen Relationen (siehe Tabelle 4.8), aus der er die passende auswählt. Der Benutzer kann aber auch eine neue Relation definieren, die anschließend in der Datenbank gespeichert und somit in die Liste der vordefinierten Relationen aufgenommen wird. Ist die Verwendung einer Datenbank nicht erwünscht, so kann der Benutzer zwar im halb-automatischen Modus die Informationen annotieren. Er erhält aber keine vordefinierten Relationen zur Auswahl und die von ihm eingegebenen Annotationen werden auch nicht gespeichert.

4.4 Algorithmen und Heuristiken

Eine weitere Aufgabe dieser Diplomarbeit besteht in der Erarbeitung, welche Informationen auf algorithmischen Weg oder mittels Heuristiken gefunden werden können. Ein Algorithmus ist definiert als eine „eindeutige und vollständige Beschreibung eines methodischen Wegs zur Lösung eines Problems“ [uni06]. Bei der Extraktion von semantischen Informationen wird immer, wenn eine Information mit Sicherheit gefunden werden kann und die Semantik eindeutig ist, ein Algorithmus angegeben. In dem Fall, dass eine Information nicht mit Sicherheit zu extrahieren ist und die Semantik nur angenommen werden kann, wird eine Heuristik definiert.

Die in dieser Arbeit aufgestellten Algorithmen und Heuristiken haben folgende Struktur:

Name Jeder Algorithmus und jede Heuristik besitzt einen Namen.

Beschreibung Eine kurze Beschreibung des Algorithmus und der Heuristik.

Semantik Welche Semantik kann aus der Information abgeleitet werden?

Quelle In welcher Datenquelle wird diese Information gefunden?

HTML	http://Host/Verzeichnis/Seitenname
XML	Spezialfunktion „Seiten exportieren“ http://Host/Verzeichnis/Spezial:Export/Seitenname
Wiki-Text	Aus dem XML-Dokument (<text>Text</text>) oder aus der HTML-Seite im Bearbeitungsmodus (http://Host/Verzeichnis/index.php?title=Seitennamen &action=edit <form>Text</form>)
Spezialfunktion	HTML-Seite, die von einer Spezialfunktion erzeugt wird

Tabelle 4.9: Datenquellen

Extraktion Wie kann man diese Information extrahieren?

Darstellung Die Darstellung im XML-Dokument (Output.xml).

Der Name, die Beschreibung, die Semantik und die Darstellung der definierten Algorithmen und Heuristiken, ist auf alle Wiki-Systeme anwendbar. Die angegebene Quelle und die Anleitung zur Extraktion bezieht sich auf Wikis, die auf der MediaWiki-Software beruhen.

4.4.1 Algorithmen

Wiki

Jedes Wiki-System hat einen Namen. Diese Information gibt Auskunft darüber, aus welcher Quelle eine Seite stammt.

Quelle: XML
<sitename>Wiki</sitename>

<sitename relation="extracted-from" result="correct">Wiki</sitename>

Version

Der Name der Software inklusive der Versionsnummer des Wiki-Systems.

Quelle: XML
<generator>Version</generator>

<generator relation="generated-from" result="correct">Version</generator>

Autor

Der Autor oder die Autoren, die an der Seite mitgewirkt haben.

1. Quelle: HTML
   (http://Host/Verzeichnis/index.php?title=Seitenname&action=history)
   <a href="/wiki/Benutzer:Benutzername"
   title="Benutzer:Benutzername">Benutzername

2. Quelle: XML
   <contributor>
   <username> </username> und/oder
   <id> </id> und/oder
   <ip> </ip>
   </contributor>

<authors>
<author relation="wrote" result="correct">Autor</author>
</authors>

Datum

Das Datum der letzten Änderung.

1. Quelle: XML
   <timestamp>Version</timestamp>
2. Quelle: HTML
   Diese Seite wurde zuletzt geändert um 14:44, 28. Jul 2006.

<timestamps>
<timestamp relation="edited-on" result="correct">Datum</timestamp>
</timestamps>

4.4.2 Heuristiken

Heuristiken bezeichnen Strategien, die das Finden von unsicheren semantischen Informationen ermöglichen. Sie kommen dann zum Einsatz, wenn kein mit Sicherheit zum Erfolg führender Algorithmus bekannt ist.

Synonym

Synonyme (siehe 2.4.2) sind Wörter, die eine identische oder ähnliche Bedeutung haben.
Wenn bei einem Link (siehe 3.3) auf einer Seite der Linktext nicht identisch mit dem Seitennamen ist, kann dies ein Hinweis auf ein Synonym sein. Eine Weiterleitung (siehe 3.2) von einer Seite auf eine andere kann auch auf ein Synonym hinweisen.

1. Quelle: HTML
   Die Zielbeschreibung eines Links ist ungleich dem Seitennamen.
   anderer Name
   oder
   Quelle: Wiki-Text
   [[Seite|anderer Name]]

2. Quelle: Wiki-Text *#REDIRECT[[Begriff, auf den weitergeleitet wird]]*

```
<relations_between_terms>
    <relation_between_terms relation="synonym" result="possibly-correct">
        <term1>Begriff 1</term1>
        <term2>Begriff 2</term2>
    <relation_between_terms>
</relations_between_terms>
```

Akronym

Akronyme (siehe 3.4.3) sind Abkürzungen, die aus den Anfangsbuchstaben mehrerer (Teil-) Wörter gebildet werden.
In den meisten Fällen wird das Akronym, bevor es benutzt wird, einmal als kompletter Ausdruck geschrieben.
Akronyme lassen sich im Wiki-Text oder im HTML-Dokument finden, wenn nach Mustern gesucht wird. Links sind eine andere Möglichkeit, um Akronyme zu finden.

1. Nach [SY00] sind die häufigsten Schreibweisen für Akronyme.
   Ausdruck (Akronym) oder
   Akronym (Ausdruck) oder
   Akronym - Ausdruck.

2. Die Zielbeschreibung eines Links ist ungleich dem Seitennamen.
   Quelle: HTML
   Ausdruck oder
   Akronym oder
   Quelle: Wiki-Text
   [[Akronym|Ausdruck]] oder
   [[Ausdruck|Akronym]].

```
<relations_between_terms>
    <relation_between_terms relation="acronym" result="possibly-correct">
        <term1>Begriff 1</term1>
        <term2>Begriff 2</term2>
    <relation_between_terms>
</relations_between_terms>
```

Homonym

Als Homonym (siehe 3.4.4) bezeichnet man ein Wort, das unterschiedliche Bedeutungen hat. Der Seitenname eines Dokuments sollte eindeutig sein, damit er über die Suchfunktion eines Wikis gefunden werden kann. Eine feste Regel für die Namensgebung einer Seite gibt es nicht. Der Name sollte kurz sein und den Inhalt gut beschreiben (vgl. [CL01]). Insbesondere Abkürzungen sind nicht eindeutig, so dass aus einer Liste von Möglichkeiten die gewünschte Seite ausgewählt werden muss (siehe Abbildung 4.9) (vgl. [Lan05]). Wenn der Name der Seite ein Homonym ist, gibt es mehrere Möglichkeiten, die Seiten zu unterscheiden. Links sind eine andere Möglichkeit um Homonyme zu finden.

1. Quelle: HTML
   Seitenname (Bereich) - Beispiel: Bank (Kreditinstitut)

2. Quelle: HTML
   Seitenname führt auf eine Begriffserklärungsseite

3. Quelle: HTML
   Homonym oder
   Quelle: Wiki-Text
   [[Homonym (Bereich)|Homonym]]

```
<relations_between_terms>
    <relation_between_terms relation="homonym" result="possibly-correct">
        <term1>Begriff 1</term1>
        <term2>Begriff 2</term2>
    <relation_between_terms>
</relations_between_terms>
```

Sucht man in Wikipedia nach dem Artikel Content Management System und gibt in der Suchfunktion CMS ein, so kann kein Artikel eindeutig identifiziert werden. CMS ist sowohl eine Abkürzung für Content Management System als auch für Card-Management System, Cash-Management System, Chipmesssystem, Chronisches Müdigkeitssyndrom usw. In diesem Fall wird eine Liste aller möglichen Artikel angeboten, wie die Abbildung 4.9 zeigt.

CMS

(Weitergeleitet von Cms)

Die Abkürzung **CMS** steht für:

- Card-Management-System in der Informationstechnik zur Verwaltung von ausgegebenen C
- Cash-Management-System in der Geldbearbeitung und im Bankwesen
- Chip-Messsystem, Messsystem der Firma Dräger für Momentankonzentrationen in der Ur
- Chronisches Müdigkeitssyndrom in der Medizin
- „Color-Management-System" in der hochwertigen Display- und Drucktechnik, um gleiche F können, siehe Farbmanagement
- Compact Muon Solenoid, Detektor bei der Europäischen Organisation für Kernforschung
- Concerned Member State im Bereich der europäischen Arzneimittelzulassung
- Constant Maturity Swap
- Content-Management-System in der Informationstechnik
- „Convention on the Conservation of Migratory Species of Wild Animals", siehe Bonner Kon
- Conversational Monitor System, entwickelt von IBM
- Cryptographic Message Syntax in der Informatik
- Cytoplasmatic male sterility in der Genetik

Abbildung 4.9: Screenshot einer Begriffserklärungsseite. Quelle: [wik06e]

Links

In einem Wiki lassen sich verschiedene Linktypen extrahieren (siehe 3.3). Die Semantik eines Links kann nicht automatisch zugewiesen werden. Durch den Typ (Wikilink, Externer Link usw.) des Links erhält man lediglich die Information, um welche Art von Link es sich handelt. Ein externer Link beispielsweise wird mit sehr großer Wahrscheinlichkeit einen inhaltlichen Zusammenhang haben und nicht der Navigation dienen.

Quelle: XML

- Wikilink *[[]]*
- Externer Link *[]*
- Bildlink *[[Bild:]]*

Quelle: Spezialfunktion
(http://Host/Verzeichnis/index.php?title=Spezial:Whatlinkshere
&target=Seitenname&limit=Limit&offset=0) und HTML

- Backlinks
  Seitenname

```
<links>
    <link relation="" result="">
        <type>Linktyp</type>
        <name>Seitenname</name>
    </link>
</links>

<externallinks>
    <externallink relation="" result="">
        <url>URL</url>
        <description>Beschreibung</description>
    </externallink>
</externallinks>

<imagelinks>
    <imagelink relation="" result="">
        <name>Name</name>
        <description>Beschreibung</description>
    </imagelink>
</imagelinks>
```

```
<backlinks>
      <backlink relation="" result="">Seitenname</backlink>
</backlinks>

<compare_wikilinks_backlinks>
      <linkname relation="" result="">Seitenname</link>
</compare_wikilinks_backlinks>
```

Kategorie

Eine Seite kann in eine oder auch in mehrere Kategorien eingeordnet werden. Kategorien ermöglichen eine Klassifizierung der Seiten in verschiedene Themengebiete (vgl. 3.3.6). Eine Semantik kann nicht automatisch zugewiesen werden.

1. Quelle: Wiki-Text
   [[Kategorie:]]

2. Quelle: HTML
   <a href="/Verzeichnis/Kategorie:Kategoriename"
   title="Kategorie:Kategoriename">Kategoriename

```
<categories>
     <category relation="" result="">Kategorie<category>
<categories>
```

Oberbegriff

Hyponymie ist eine Relation, die zwischen einem allgemeinen und einem spezifischen Begriff besteht (siehe 2.4). Der allgemeine Begriff ist der Oberbegriff. Eine Kategorie selbst kann auch wiederum einer anderen Kategorie zugeordnet sein. Das Konzept der Kategorien bildet somit eine hierarchische Struktur, d.h. die Oberkategorie kann ein Oberbegriff sein. Das Attribut „level“ gibt die Rekursionstiefe an.

Quelle: HTML (http://Host/Verzeichnis/Kategorie:Seitenname)
[[Kategorie:]]

<majorcategories>
<majorcategory relation="major-term" result="possibly-correct"
level="Rekursionstiefe">Oberbegriff</majorcategory>
</majorcategories>

Unterbegriff

Hyponymie ist eine Relation, die zwischen einem allgemeinen und einem spezifischen Begriff besteht (siehe 2.4). Der spezifische Begriff ist der Unterbegriff. Eine Kategorie kann selbst auch wiederum eine Unterkategorie besitzen. Das Konzept der Kategorien bildet eine hierarchische Struktur, d.h. die Unterkategorien können Unterbegriffe sein. Das Attribut „level“ gibt die Rekursionstiefe an.

Quelle: HTML (http://Host/Verzeichnis/Kategorie:Seitenname)
<a . href="/Verzeichnis/Kategorie:Seitenname">Seitenname*

<subcategories>
<subcategory relation="sub-term" result="possibly-correct"
level="Rekursionstiefe">Unterbegriff</subcategory>
</subcategories>

Seiten innerhalb einer Kategorie

Eine Seite kann in eine oder auch in mehrere Kategorien eingeordnet werden. Kategorien ermöglichen eine Klassifizierung der Seiten in verschiedene Themengebiete (vgl. 3.3.6). Seiten innerhalb einer Kategorie stehen in einem inhaltlichen Zusammenhang.

Quelle: HTML (http://Host/Verzeichnis/Kategorie:Kategoriename)

```
<a href="http://Host/Verzeichnis/Seitenname"
title="Seitenname">Seitenname</a>

<pages_in_category>
    <categorypage relation="" result="">
        <categoryname>Kategoriename</categoryname>
        <page>Seitenname</page>
    </categorypage>
</pages_in_category>
```

Schlüsselwörter

In jedem Text lassen sich Wörter finden, die charakteristisch für den Inhalt sind. Um die Worthäufigkeiten in einem Text zu bestimmen, wird eine Stopwortliste benötigt (siehe 3.4.2). Gegeben ist der Text in Wiki-Syntax und eine Stopwortliste.

Folgende Schritte werden nacheinander ausgeführt.

1. entferne alle Sonderzeichen (!, ", &, /, [,], (,), ', #, , {, }, _ usw.),
2. entferne alle Zahlen,
3. ersetze ein Leerzeichen durch zwei,
4. wandle den Text in Kleinbuchstaben um,
5. bilde die Grundform,
6. Für alle Wörter in der Stopwortliste:
    6.1 ändere die Zeichenkodierung von ISO-8859-1 in UTF-8,
    6.2 entferne alle Wörter im Text, die gleich dem Wort aus der Stopwortliste sind,
7. zähle wie oft jedes Wort vorkommt.

Im Schritt 3 werden alle Wörter durch zwei Leerzeichen getrennt, die später wieder entfernt werden. Dieser kleine Trick stellt sicher, dass direkt hintereinander folgende Stopwörter auch gelöscht werden und das Wörter, die eines der Stopwörter enthalten, nicht getrennt werden.

<keywords>
    <frequency>Anzahl</frequency>
    <word>Wort</word>
</keywords>

Hervorgehobene Wörter

Wichtige Wörter werden hervorgehoben. Das kann sowohl durch die Textformatierung (fett, kursiv, unterstrichen) als auch durch die Kennzeichnung als Überschrift geschehen (vgl. [Bur04]).

1. Quelle: HTML
   Wort (fett), Wort (kursiv), <u>Wort</u> (unterstrichen), <h1>Überschrift</h1> (Überschrift 1 <h1>, Überschrift 2 <h2>,Überschrift 3 <h3>, ...)

2. Quelle: Wiki-Text
   "'Wort"' (fett), "Wort" (kursiv), <u>Wort</u> (Unterstrichen), =Überschrift= (Überschrift 1, Überschrift 2 ==, Überschrift 3 ===)

<markup_words>
<type>Formatierung</type>
<word>Wort</word>
</markup_words>

Letzte Änderung

Letzte Änderung bezieht sich auf das Datum an dem die Seite das letzte Mal geändert wurde. Die Semantik ist eine andere als die des Algorithmus mit dem Namen „Datum“. Die Information des Datums der letzten Änderung kann eine Aussage über die Aktualität des Themas sein.

1. Quelle: XML
   <timestamp>Version</timestamp>

2. Quelle: HTML
   Diese Seite wurde zuletzt geändert um 14:44, 28. Jul 2006.

<last_modified>lezte Änderung</last_modified>

In MediaWiki findet man unter der Rubrik „Versionen/Autoren“ eine Liste aller Änderungen der gerade aufgerufenen Seite, die so genannte Versionshistorie (siehe 2.1.2). Alte Versionen können über einen Klick auf den Link mit Datum und Uhrzeit wieder aufgerufen und ältere Beiträge wiederhergestellt werden. Jede Änderung an der Seite wird in die Versionshistorie aufgenommen, falls beim Speichern im Feld „Zusammenfassung“ ein Kommentar eingegeben wurde. Über

Details der Änderungen kann man sich, durch Klicken auf „Aktuell“ oder „Letzte“ in der Liste, informieren. Die Unterschiede zwischen zwei Versionen werden ebenfalls angezeigt. Hierdurch ist es möglich, zurückzuverfolgen, wie eine Seite entstanden ist oder welcher Benutzer in letzter Zeit Änderungen an der betreffenden Seite vorgenommen hat (vgl. [wik05b]). Einen Ausschnitt aus der HTML-Seite zeigt die Abbildung 4.10.

Abbildung 4.10: Screenshot: Versionen/Autoren. Quelle: [wik06a]

Anzahl der Änderungen einer Seite

Unter der Rubrik „Versionen/Autoren“ findet man die Versionshistorie (vgl. [wik05b]) einer Seite. Dort kann man zurückverfolgen, wann und von welchem Benutzer eine Änderung an der Seite vorgenommen wurde. Die Spezialfunktion „Seiten exportieren“ kann auf Verlangen alle Versionen einer Seite exportieren. Eine Seite, die oft geändert wurde, deutet auf eine hohe Qualität hin.

1. Quelle: HTML
   (http://Host/Verzeichnis/index.php?title=Seitenname&action=history)
   Anzahl der *<li>-Tags* zwischen den *<form></form>*

2. Quelle: XML
   Anzahl des *<revision>-Tags*

<num_ changes>Anzahl der Änderungen<num_ changes>

Anzahl der Autoren, die eine Seite bearbeitet haben

Eine Seite, die von mehreren Autoren bearbeitet wurde, enthält weniger Fehler als eine Seite, die nur von einem Autor geschrieben wurde. Es gibt zwei Möglichkeiten, die Anzahl der verschiedenen Autoren einer Seite festzustellen. Zum einen kann man diese Information, wie bei der Heuristik „Anzahl der Änderungen einer Seite“, über die Rubrik „Versionen/Autoren“ herausfinden und zum anderen aus dem XML-Dokument, das durch die Funktion „Seiten exportieren“ (siehe 4.1.2.1) erzeugt wird.

1. Quelle: HTML
   (http://Host/Verzeichnis/index.php?title=Seitenname&action=history)
   <a href="/wiki/Benutzer:Benutzername"
   title="Benutzer:Benutzername">Benutzername

2. Quelle: XML (http://Host/Verzeichnis/Spezial:Export/Seitenname)
   <contributor>
   <username> </username> und/oder
   <id> </id> und/oder
   <ip> </ip>
   </contributor>

<num_ authors>Anzahl der verschiedenen Autoren</num_ authors>

Bearbeitete Seiten eines Autors

Ein aktiver Autor wirkt in der Regel an mehreren Seiten mit. Des Weiteren lässt sich beobachten, dass ein Autor zu bestimmten Themen Beiträge abgibt. In den meisten Fällen betreffen diese thematisch den Beruf oder auch das Hobby des Autors (vgl. 3.1.3). Weitere Seiten eines Autors können interessant sein, da sie ähnliche Themen behandeln. Um den Benutzernamen eines Autors zu finden, gibt es zwei Möglichkeiten:

1. Quelle: HTML
   <a href="http://Host/Verzeichnis/Benutzer:Benutzername"
   title="Benutzer:Benutzername">Benutzername
   (http://Host/Verzeichnis/index.php?title=Seitenname&action=history)

2. Quelle: XML
   <contributor>
   <username> </username> und/oder
   <id> </id> und/oder
   <ip> </ip>
   </contributor>

Im nächsten Schritt muss von der Benutzerseite (Quelle: HTML http://Host/Verzeichnis/Benutzer:Benutzername) aus, die Spezialfunktion (Quelle: http://Host/Verzeichnis/index.php?title=Spezial:Whatlinkshere&target =Benutzer:Benutzername) aufgerufen werden. Die daraufhin erzeugte Liste enthält alle Seiten, an denen der Benuzter mitgewirkt hat.

<authorpages>
<authorname relation="wrote" result="correct">
<name>Autor</name>
<page>Seitenname</name>
</author>
</authorpages>

Meta-Daten

Die meisten Wikis benutzen Textbausteine, um Textabschnitte wiederzuverwenden, die auf vielen Seiten vorkommen (vgl. 3.4.1). MediaWiki verwendet beispielsweise einen Textbaustein dafür, um eine Seite als sehr lesenswert, exzellent, informativ oder auch als mangelhaft auszuzeichnen.

Quelle: HTML
{{}}

<metadata>
<meta>Meta-Daten</meta>
</metadata>

Kapitel 5

Implementation

In diesem Kapitel wird die prototypische Implementation des Konzeptes vorgestellt. Die Umsetzung erfolgte auf der Basis des MediaWikis. Das Unterkapitel (5.1) beschreibt die verwendeten Technologien und die realisierten Funktionen. Im Anschluss wird eine Auswahl an Problemen (5.2) und Lösungsansätzen vorgestellt, die sich während der Implementierungsphase ergeben haben. Anhand eines Beispiels (5.3) wird im letzten Unterkapitel der gesamte Prozess der „**Extraktion von semantischen Informationen**“ demonstriert.

5.1 Umsetzung

Folgende Technologien wurden verwendet:

Web-Anwendung	Browser: Microsoft Internet Explorer 6.0, HTML, CSS, JavaScript, DOM, XSLT: Transformation des XML-Dokuments in HTML
Mediator	PHP Version 5.1.1
Datenbank	MySQL
Konfigurationsdatei	XML-Datei
Wörterbuch	Wortschatz Universität Leipzig http://wortschatz.informatik.uni-leipzig.de/cgi-bin/wort_www.exe?Wort=Suchwort&site=1&cs=1
Stopwortliste	Textdatei
Wrapper	PHP Version 5.1.1
Wiki	MediaWiki 1.5.6, XAMPP für Windows Version 1.5.1, Apache 2.2.0, MySQL

Tabelle 5.1: Eingesetzte Technologien

MediaWiki verwendet die Skriptsprache PHP. Aus diesem Grund wurde diese Programmiersprache für die Implementation des Prototyps gewählt. Die Abbil-

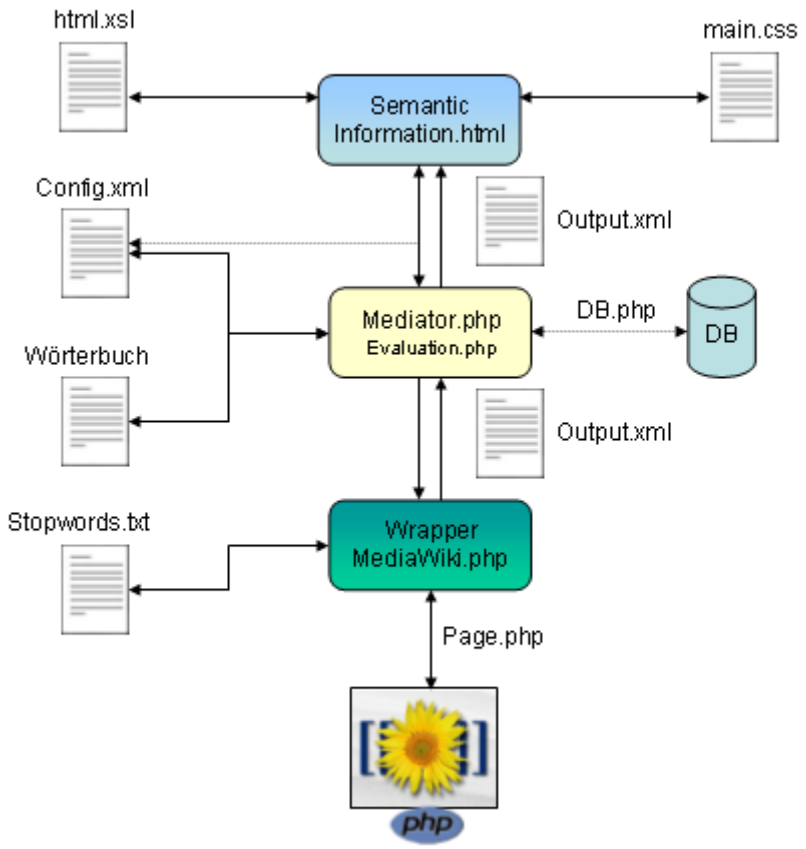


Abbildung 5.1: Prototyp im Überblick

dung 5.1 zeigt, wie die einzelnen Systemkomponenten des Konzeptes umgesetzt wurden. In den nächsten Unterkapiteln werden diese kurz erläutert.

5.1.1 SemanticInformation.html

Als Anwendung wurde für den Prototyp eine Ajax-Anwendung[1] gewählt. Ajax bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und einem Server, welches ermöglicht, dass die HTML-Seite nicht mit jeder HTTP-Anfrage komplett neu geladen werden muss. Es werden nur Teile einer HTML-Seite bei Bedarf nachgeladen. Um eine interaktive, Desktop-ähnliche Web-Anwendung zu realisieren, wurden verschiedene Technologien eingesetzt: HTML, CSS und DOM zur Repräsentation der Inhalte sowie JavaScript zur Manipulation des DOM und zur dynamischen Darstellung der Inhalte. JavaScript dient gleichzeitig als Schnittstelle zwischen den einzelnen Komponenten. Das XMLHttpRequest-Objekt ist Bestandteil vieler Browser und ermöglicht den

[1] Apronym für Asynchronous JavaScript and XML.

Austausch von Daten auf asynchroner Basis. Als Ausgabe wurde das erzeugte XML-Dokument mit XSLT in HTML transformiert.

Über ein Formular kann ein Benutzer die Übergabeparameter auswählen, die an den Mediator übergeben werden.

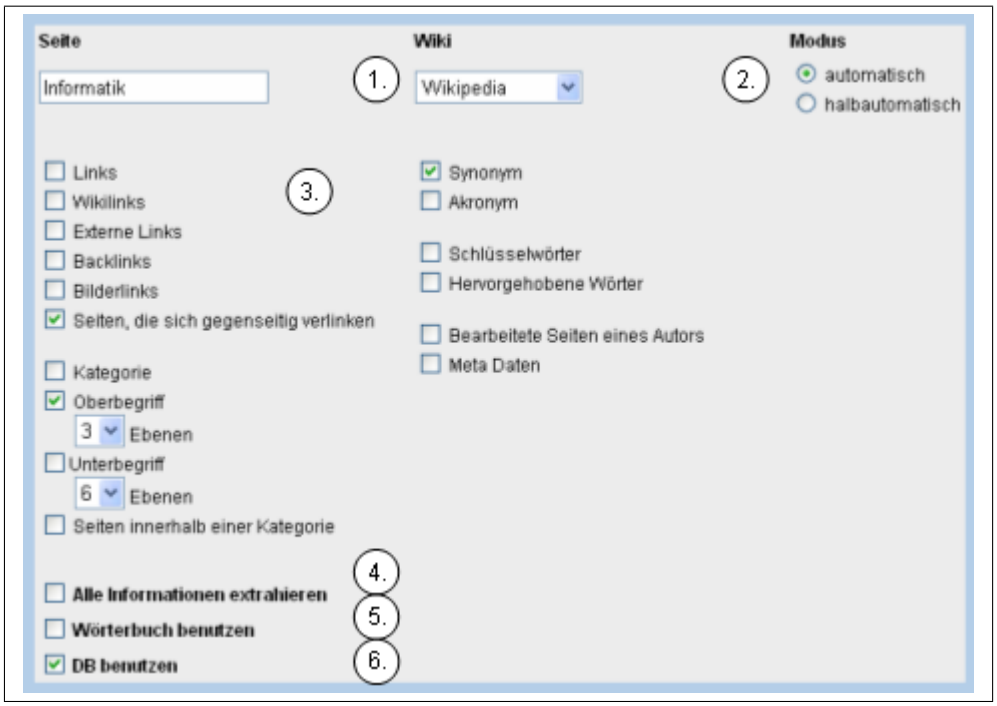


Abbildung 5.2: Prototyp: Implementierte Funktionen

Folgende Funktionen wurden implementiert:

- [1.] Eine Liste der Wiki-Systeme, die als Datenquelle genutzt werden können, wird aus der Konfigurationsdatei erstellt.
- [2.] Die Auswahl zwischen einem automatischen oder halb-automatischen Modus.
- [3] Die Extraktion folgender semantischer Informationen:
  - Wiki,
  - Version,
  - Autor,
  - Datum,
  - Links,

  - Kategorie,
  - Oberbegriff,
  - Unterbegriff,
  - Seiten innerhalb einer Kategorie,
  - Synonym,
  - Akronym,
  - Schlüsselwörter,
  - Hervorgehobene Wörter,
  - Bearbeitete Seiten eines Autors,
  - Meta-Daten.

- (4.) Alle Informationen werden extrahiert,
- (5.) Die Einbindung eines Wörterbuches,
- (6.) Die Einbindung einer Datenbank.

5.1.2 Mediator.php

Alle festen und optionalen Aufgaben sind implementiert (siehe 4.1.3). Als Konfigurationsdatei wird eine XML-Datei verwendet, da DOM das Auslesen der einzelnen Elemente erleichtert. Alle eingetragenen Wiki-Systeme können als Datenquelle benutzt werden[2]. Die Einbindung einer MySQL Datenbank ist auch funktionsfähig. Für die Kommunikation mit der Datenbank sind die Funktionen im Skript DB.php zuständig.

5.1.3 DB.php

Bei Bedarf stellt das Skript DB.php eine Datenbankverbindung her. Innerhalb einer Tabelle sollten keine Einträge doppelt vorkommen. Aus diesem Grund wird vor jedem Hinzufügen überprüft, ob die Elemente schon existieren. Folgende Funktionen wurden implementiert:

- connect()
  Eine Verbindung zur Datenbank wird hergestellt.
- disconnect()
  Die Verbindung zur Datenbank wird beendet.

[2]Die verwendete Konfigurationsdatei befindet sich im Anhang auf Seite 129.

- insertIntoRelation($relation)
  Eine neu definierte Relation ($relation) wird in die Tabelle „relation“ eingefügt.

- insertXML($name, $text)
  Der Seitenname ($name) und das gesamte XML-Dokument als String ($text) wird in die Tabelle „page“ eingefügt.

- insertIntoInformation($term1, $term2, $type, $relation)
  Die Semantik ($relation) zwischen zwei Links wird in der Tabelle „information“ gespeichert. Ebenso: der Seitenname ($term1), der Seitenname auf den verlinkt wird ($term2) und der Linktyp ($type).

- insertInto($table, $text1, $text2)
  Für die Begriffsrelationen gibt es jeweils eine Tabelle (synonym, acronym, homonym). Je nachdem um welche Beziehung es sich bei zwei Begriffen ($term1, $term2) handelt, werden diese in der Tabelle ($table) gespeichert.

- getRelations()
  Erzeugt im halb-automatischen Modus aus den definierten Relationen der Tabelle „relation“ ein Array.

- getRelation($term1, $term2)
  Für den Fall, dass bereits eine Semantik zwischen zwei Links ($term1 und $term2) definiert wurde, gibt die Funktion diese aus.

- valuesExist($table, $value1, $value2)
  Wenn zwei Werte ($value1 und $value) in der Tabelle „$table“ existieren, gibt die Funktion *exist* aus, wenn nicht *existnot*.

- relationExist($relation)
  Wenn die Relation $relation in der Tabelle „relation“ gefunden wird, dann gibt die Funktion *exist*, wenn nicht *existnot*, zurück.

- searchTable($word1, $word2)
  Die Funktion gibt im positiven Fall, d.h. wenn die Einträge gefunden werden, den Tabellennamen (synonym, homonym, acronym) aus. Im negativen Fall wird der Wert *existnot* ausgegeben.

5.1.4 WrapperMediaWiki.php

Die Klasse im Skript WrapperMediaWiki.php erzeugt eine Instanz von Page (siehe Tabelle 5.2). Die Extraktion der semantischen Informationen erfolgte nach den definierten Algorithmen und Heuristiken im Kapitel 4.4. Das aus den extrahierten Daten erzeugte XML-DomDocument wird an den Mediator zurückgegeben. Die Funktion getKeywords() benutzt zur Bestimmung der Schlüsselwörter eine Stopwortliste[3].

[3]Die Stopwortliste aus dem Anhang auf Seite 116 wurde verwendet.

Page
getXML(): return String (Wiki-Text)
pageExist(): return Boolean
getText()
getSitename(): return String
getGenerator(): return String
getAuthor(): return String
getTimestamp(): return String
getAllLinks(): return Array
getLinks(): return Array
getWikilinks(): return Array
getLinksDescription(): return Array
getImagelinks(): return Array
getImagelinksDescription(): return Array
getCategories(): return Array
getExternalLinks(): return Array
getExternalLinksDescription(): return Array
getBacklinks(): return Array
getEmphasizedWords(): return Array
getRedirect(): return Array
getMetaData(): return Array
getAcronym(): return Array
getKeywords(): return Array
getSynonym(): return Array
getCategoryPage(): return String
getMajorCategories: return Array
getSubCategories: return Array
getCategoriesPages: return Array
getPagesFromAuthor: return Array

Tabelle 5.2: Klasse: page.php

5.2 Probleme und Lösungsansätze

5.2.1 Akronyme

Bei der Extraktion von Akronymen (nach der Heuristik Akronym) hat sich herausgestellt, dass dies nicht in allen Fällen automatisch funktioniert. Es gibt zwei verschiedene Möglichkeiten Akronyme zu finden. Die erste Möglichkeit verwendet reguläre Ausdrücke. Wenn die Anzahl der Zeichen der Abkürzung identisch mit der Anzahl der Wörter sind, wie z.B. bei DOM - Document Object Model oder CMS - Content Management System, ist das Finden der Akronyme kein Problem. Bei Akronymen, wie z.B. HTTP - Hypertext Transfer Protocol ist das nicht so einfach. In diesem Fall würde man ein Wort zu viel extrahieren und nicht auf das richtige Ergebnis kommen. Dieses Problem lässt sich nur lösen, wenn man davon ausgeht, dass alle Buchstaben, die das Akronym bilden, großgeschrieben werden - HTTP - HyperText Transfer Protocol.

5.2.2 Wörterbuch

Wie im Konzept beschrieben, kann der Mediator zur Verifikation von Begriffsrelationen ein Wörterbuch (4.1.3.2) einbeziehen. Bevor zwei Wörter jedoch verifiziert werden können, müssen sie in ihrer Grundform vorliegen. Das bedeutet, dass jeweils für ein Wortpaar mindestens drei Wörterbuchabfragen durchgeführt werden müssen. Eine HTTP-Anfrage an ein Wörterbuch hat sich als zeitaufwendig herausgestellt. Des Weiteren wurden in vielen Fällen die Einträge im Wörterbuch nicht gefunden.

5.2.3 Zeichenkodierung

Wiki-Systeme, die auf MediaWiki beruhen, benutzen als Zeichenkodierung UTF-8 (siehe 3.2.3). Die XML-Daten, die vom Webserver zum Browser geschickt werden, sind auch in UTF-8 kodiert. Wenn die Webseite aber eine andere Zeichenkodierung benutzt, kann dies zu Konflikten führen. Aus diesem Grund wurde in der HTML-Seite folgender String *<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>* geändert.

5.2.4 JavaScript

Bei der Benutzung von JavaScript ergaben sich folgende Probleme. Jeder Browser reagiert auf JavaScript anders. Aus diesem Grund wurde der Prototyp für den Microsoft Internet Explorer 6.0 optimiert. Ein zweites Problem ergab sich bei der Verwendung eines Formulares. Die Übergabe der Daten an eine Funktion per HTTP-GET erfolgt durch einen String, dessen Länge - bei den meisten Systemen auf 1024 Bytes - beschränkt ist. Diese Methode kann sinnvoll sein, wenn überprüft

werden soll, welche Daten übertragen worden sind. Da es sich im Worst-Case um sehr viele Informationen handelt, die mittels eines Formulars annotiert werden, bietet sich diese Methode nicht an. Die Übergabe per HTTP-POST liefert ein assoziatives Array. Der Vorteil ist, dass die Länge nicht beschränkt ist. Aus diesem Grund wurden die Daten bei der Evaluation mit der Methode HTTP-POST übergeben.

5.2.5 Speicherung

Das gesamte XML-Dokument wird als String in der Datenbank gespeichert. Diese Methode ist jedoch nicht geeignet, weil dadurch nicht auf die einzelnen Elemente zugegriffen werden kann. Eine elementbasierte Speicherung wie im Kapitel 4.1.3.3 beschrieben, sollte bevorzugt werden.

5.3 Beispiel

Der Prototyp wird mittels Screenshots, zunächst im automatischen Modus ohne Einbindung der Datenbank und anschließend im halb-automatischen Modus mit Speicherung der Daten dargestellt. Angesichts der vielen verschiedenen Informationen, wird eine kleine Auswahl getroffen, die alle Fälle in Bezug auf die Semantik abdecken: Wiki, Version, Autor und Datum haben eine eindeutige Semantik (siehe 4.4.1), bei den Synonymen und Akronymen ist es möglich eine Annahme über die Semantik zu treffen und bei den Links kann keine Semantik automatisch zugeordnet werden. Die Ergebnisse zeigen auch nur einen Teil der Daten. Als Datenquelle wurde ein Artikel über „Informatik“ aus einem Test-Wiki verwendet.

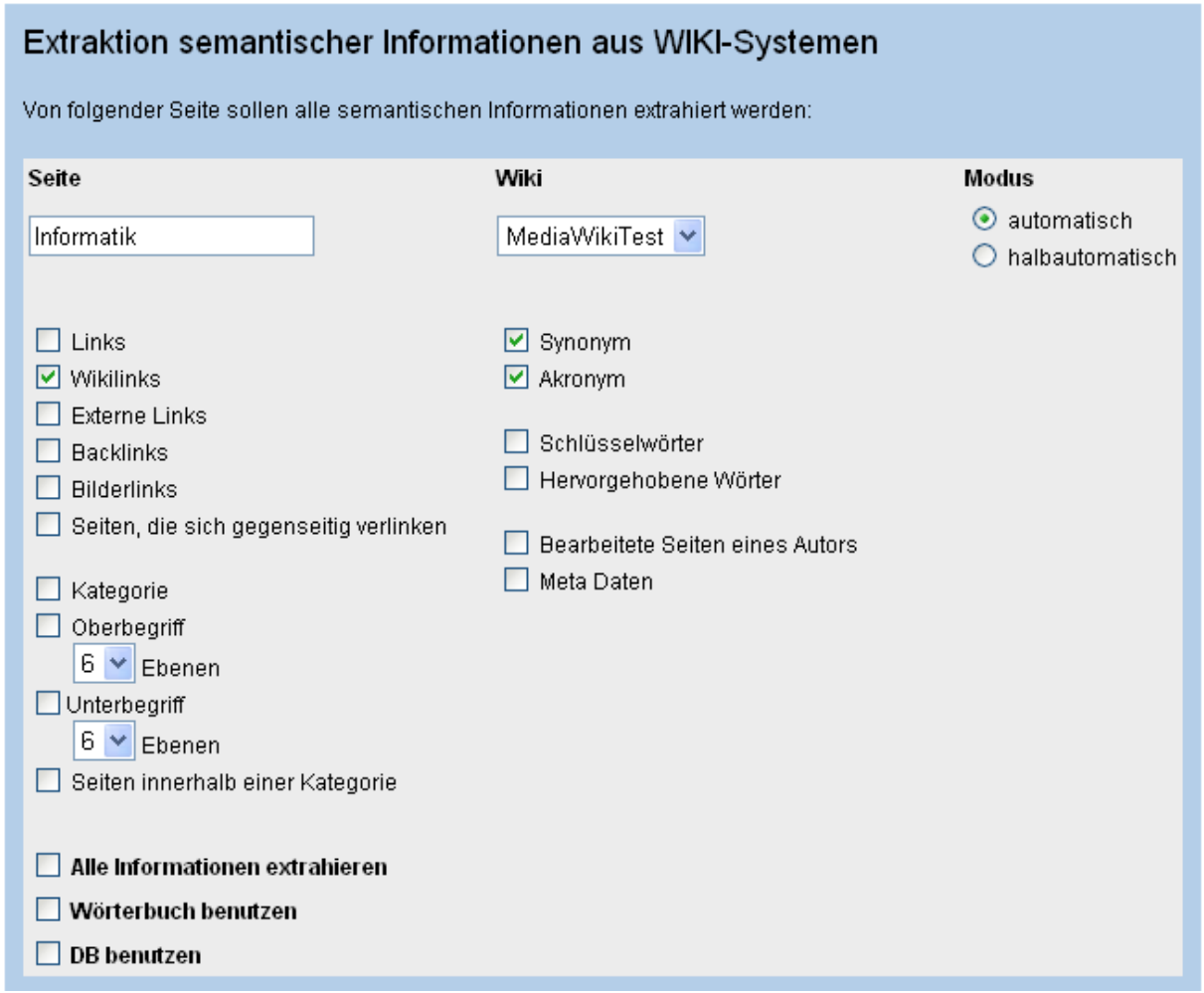


Abbildung 5.3: Auswahlmaske.

Informatik

Wiki		Semantik	Bewertung
MediaWikiTest		extracted-from	correct
Version			
MediaWiki 1.5.6		generated-from	correct
Autor			
127.0.0.1		wrote	correct
Erstellt am			
2006-09-21T10:45:41Z		edited-on	correct
Linktype	**Link**	**Semantik**	**Bewertung**
wikilink	C (Programmiersprache)		
wikilink	Transmission Control Protocol		
wikilink	Landau-Symbol		
wikilink	Krypthographie		
wikilink	Algorithmus		
wikilink	Konrad Zuse		
Synonyme, Akronyme oder Homonyme			
Begriff 1	**Begriff 2**	**Semantik**	**Bewertung**
C (Programmiersprache)	C	synonym	possibly-correct
Krypthographie	Verschlüsselungslehre	synonym	possibly-correct
Landau-Symbol	O(1)	synonym	possibly-correct
Transmission Control Protocol	TCP	synonym	possibly-correct
Personal Computer	PC	acronym	possibly-correct

Abbildung 5.4: Das Ergebnis nach automatischer Extraktion. Das Wiki, die Version, der Autor und das Datum sind sichere Informationen, die automatisch extrahiert werden können. Den Links konnte keine Semantik automatisch zugewiesen werden. Bei den Wortbeziehungen können nur Annahmen getroffen werden.

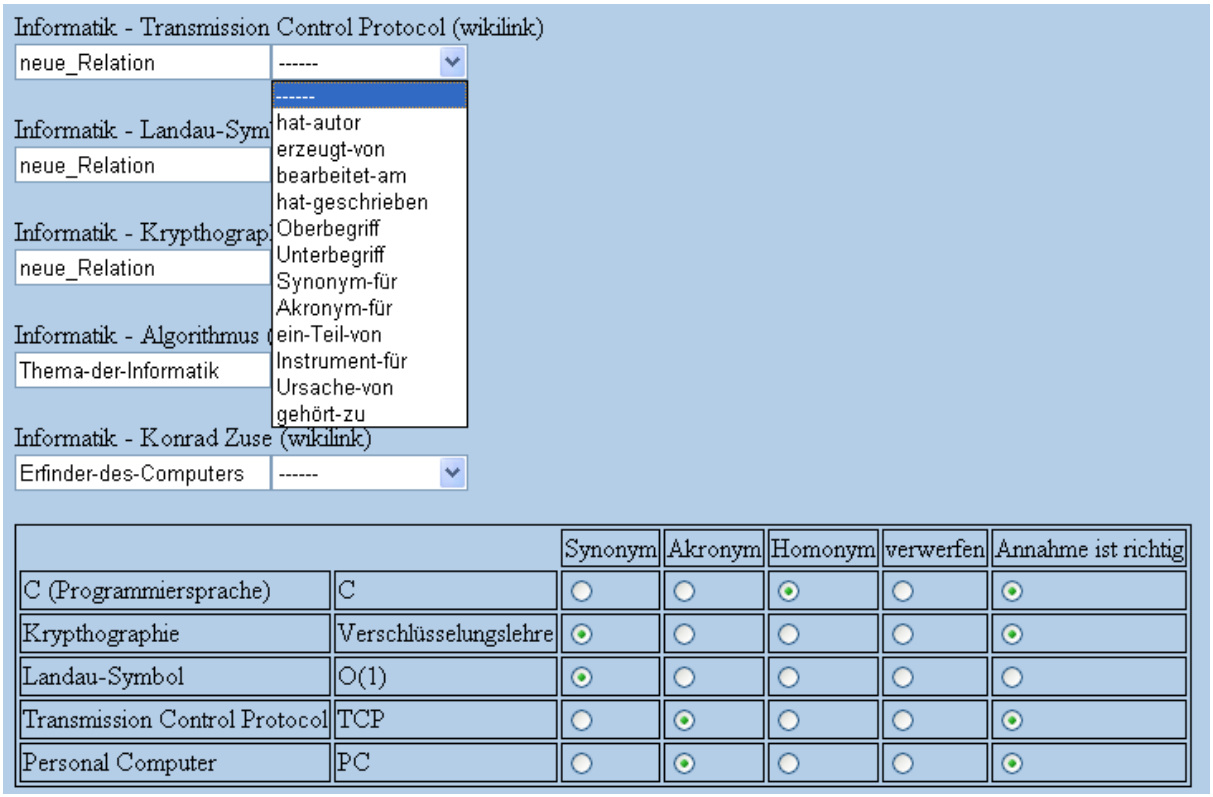


Abbildung 5.5: Im halb-automatischen Modus kann der Benutzer die Informationen annotieren. Bei Verwendung einer Datenbank wird eine Liste von Relationen erstellt, die zur Auswahl stehen.

Informatik

Wiki		Semantik	Bewertung
MediaWikiTest		extracted-from	correct
Version			
MediaWiki 1.5.6		generated-from	correct
Autor			
127.0.0.1		wrote	correct
Erstellt am			
2006-09-21T10:45:41Z		edited-on	correct
Linktype	**Link**	**Semantik**	**Bewertung**
wikilink	C (Programmiersprache)		
wikilink	Transmission Control Protocol		
wikilink	Landau-Symbol		
wikilink	Krypthographie		
wikilink	Algorithmus	Thema-der-Informatik	correct
wikilink	Konrad Zuse	Erfinder-des-Computers	correct
Synonyme, Akronyme oder Homonyme			
Begriff 1	**Begriff 2**	**Semantik**	**Bewertung**
C (Programmiersprache)	C	homonym	correct
Krypthographie	Verschlüsselungslehre	synonym	correct
Landau-Symbol	O(1)	synonym	possibly-correct
Transmission Control Protocol	TCP	acronym	correct
Personal Computer	PC	acronym	correct

Abbildung 5.6: Das Ergebnis nach halb-automatischer Extraktion. Die Semantik des Benutzers wird übernommen.

Das Ergebnis als XML zeigt der folgende Quellcode.

```
<?xml version="1.0" encoding="Utf-8"?>
<page name="Informatik">
    <siteinfo>
        <sitename relation="extracted-from" result="correct">
            MediaWikiTest</sitename>
        <generator relation="generated-from" result="correct">
            MediaWiki 1.5.6</generator>
        <authors>
            <author relation="wrote" result="correct">
                127.0.0.1
            </author>
        </authors>
        <timestamps>
            <timestamp relation="edited-on" result="correct">
                2006-09-21T10:45:41Z
            </timestamp>
        </timestamps>
    </siteinfo>
    <links>
        <link relation="" result="">
            <type>wikilink</type>
            <name>C (Programmiersprache)</name>
        </link>
        <link relation="" result="">
            <type>wikilink</type>
            <name>Transmission Control Protocol</name>
        </link>
        <link relation="" result="">
            <type>wikilink</type>
            <name>Landau-Symbol</name></link>
        <link relation="" result="">
            <type>wikilink</type>
            <name>Krypthographie</name>
        </link>
        <link relation="Thema-der-Informatik"
          result="correct">
            <type>wikilink</type>
            <name>Algorithmus</name>
        </link>
        <link relation="Erfinder-des-Computers"
          result="correct">
            <type>wikilink</type>
            <name>Konrad Zuse</name>
        </link>
```

```
    </links>
    <relations_between_terms>
        <relation_between_terms relation="homonym"
          result="correct">
            <term1>C (Programmiersprache)</term1>
            <term2>C</term2>
        </relation_between_terms>
        <relation_between_terms relation="synonym"
          result="correct">
            <term1>Krypthographie</term1>
            <term2>VerschlÃ¼sselungslehre</term2>
        </relation_between_terms>
        <relation_between_terms relation="synonym"
          result="possibly-correct">
            <term1>Landau-Symbol</term1>
            <term2>O(1)</term2>
        </relation_between_terms>
        <relation_between_terms relation="acronym"
          result="correct">
            <term1>Transmission Control Protocol</term1>
            <term2>TCP</term2>
        </relation_between_terms>
        <relation_between_terms relation="acronym"
          result="correct">
            <term1>Personal Computer</term1>
            <term2>PC</term2>
        </relation_between_terms>
    </relations_between_terms>
</page>
```

Quellcode 5.1: Erzeugtes XML-Dokument

Kapitel 6

Zusammenfassung und Ausblick

Zu Beginn der Arbeit wurde untersucht, welche Informationen sich in einem Wiki finden lassen und welche Semantik ihnen zugeordnet werden kann. Die Analyse ergab, dass drei Fälle zu unterscheiden sind: 1. eine Information hat eine eindeutige Semantik, 2. über die Semantik einer Information kann nur eine Annahme getroffen werden oder 3. es kann keine Semantik automatisch zugewiesen werden. Angesichts dieser Unterschiede in der Bestimmung der Semantik wird zwischen Algorithmen und Heuristiken differenziert. Um die verschiedenen Informationen im Detail zu beschreiben, wurde eine Schablone definiert. Durch die Schablonen kann man sich einen guten Überblick über die verschiedenen Informationen verschaffen. Darüber hinaus erleichtert die einheitliche Darstellung neue Algorithmen oder Heuristiken zu definieren und bietet Entwicklern eine Hilfestellung für die Implementierung, beispielsweise von weiteren Wrappern. Das Problem, jeder Information nicht automatisch eine Semantik zuordnen zu können, wurde gelöst, indem ein halb-automatischer Modus eingeführt wurde. Im halb-automatischen Modus kann ein Benutzer die Daten semantisch annotieren. Damit die annotierten Daten nicht verloren gehen, ist die Speicherung in einer Datenbank möglich. Der Vorteil ist, dass bei einer täglichen Extraktion von semantischen Informationen einer Seite nur die neu hinzugekommenen Informationen annotiert und verifiziert werden müssen. Alle anderen Informationen und deren Semantik können bei jedem Durchlauf aus der Datenbank entnommen werden. Dies bedeutet eine erhebliche Aufwandsreduzierung.

Ausgangspunkt der Arbeit war ebenfalls, dass alle Wiki-Systeme unabhängig von der Programmiersprache und dem Anwendungsgebiet als Datenquelle dienen sollen und dass im Hinblick auf die Visualisierung und Weiterverarbeitung ein Format gewählt wird, das anwendungsunabhängig ist. Um diese beiden Anforderungen zu erfüllen, wurden ein Wrapper und ein Mediator eingeführt. Der Prototyp zeigt, dass alle Wikis im WWW oder auf dem eigenen Rechner als Datenquelle verwendet werden können. Einzige Voraussetzung ist ein Eintrag in der Konfigurationsdatei. Das endgültig erzeugte XML-Dokument ermöglicht nicht nur die Transformation in verschiedene Ausgabeformate, sondern auch die Weiterverarbeitung durch die Anwendung. Darüber hinaus zeigt es alle extrahierten Informationen, deren Semantik und eine Bewertung der Aussage.

Mit der Problematik, dass inhaltliche Zusammenhänge zwischen Seiten nicht immer automatisch hergestellt werden können, beschäftigen sich auch die aktuellen Entwicklungen. Um die Inhalte eines Wikis besser verknüpfen zu können, führen die Entwicklungen zu den Semantic Wikis. Das Ziel der verschiedenen Projekte im Bereich der Semantic Wikis ist es, den Datenbestand mit einer Semantik anzureichern, der maschinell beispielsweise von Web-Robots verarbeitet werden kann. Die semantischen Verknüpfungen der Seiten ergeben eine Ontologie[1], die bestimmte Wissensbereiche mehr oder weniger genau abbildet. Aus diesem Netz lässt sich neues Wissen ableiten, das so noch nie explizit formuliert wurde. Dadurch werden kontextbezogene Abfragen ermöglicht wie z.B. „Gebe mir eine Liste aller Filme eines italienischen Regisseurs, die seit 1960 entstanden sind“ (vgl. [VKV+06]).

Die Idee des „Semantic MediaWiki“ wurde erstmals auf der Wikimania-Tagung 2005 in Frankfurt vorgestellt und ist auf große Zustimmung gestoßen. Das Projekt wird nun von der Universität Karlsruhe vorangetrieben und befasst sich mit der Konzeption und Entwicklung semantischer Erweiterungen der Software MediaWiki. Ziel ist es, eine einfache maschinengestützte Verarbeitung von Inhalten zu ermöglichen, indem Nutzern erlaubt wird, semantische Annotationen in den Quelltext einzufügen. Zwei Methoden sollen Informationen explizit machen: typisierte Links und typisierte Attribute. Die Seite „Berlin“ verlinkt auf die Seite „Deutschland“. Dem Link kann beispielsweise die Semantik *capital-of* gegeben werden. Die Syntax in MediaWiki ist folgende *[[capital of::Deutschland]]*. Ein Beispiel für ein typisiertes Attribut ist die Einwohnerzahl - Einwohnerzahl von Berlin *[[population:=3,993,933]]*. Die Art der Beziehung ist dabei frei formulierbar. Das birgt allerdings die Gefahr, dass ein Benutzer einem Link die Relation *wrote* gibt und ein anderer sie wiederum in *created-by* ändert. In diesem Fall, wie auch bei der Bearbeitung des Inhalts und bei der Einordnung in Kategorien, wird darauf vertraut, dass die Gemeinschaft eine Einigung findet. Dabei müssen die zusätzlichen Anforderungen der speziellen Wiki-Umgebung und der vielfältigen angestrebten Anwendungen berücksichtigt werden. Für das Konzept dieser Diplomarbeit ist die Entwicklung der Semantic Wikis eine große Bereicherung. Durch die semantischen Annotationen direkt im Quelltext sind weniger Benutzerinteraktionen im Extraktionssystem notwendig. Der Aufwand für den Benutzer[2] wird somit reduziert. Die Semantik der Links und Attribute können mittels regulärer Ausdrücke (z.B.:[[.*::.*]]|[[.*:=.*]]) automatisch vom Wrapper extrahiert werden. Auf das Wissen des Benutzers kann man nicht ganz verzichten, da auch andere Informationen wie zum Beispiel die Synonyme, Akronyme

[1] „Ontologien entstammen dem Bereich der Künstlichen Intelligenz. Ihr Ziel ist es, Wissen einheitlich in konzeptualisierter Form zu repräsentieren und damit dessen Wiederverwendbarkeit zu gewährleisten.“ [Dro05].

[2] Benutzer bezieht sich hier auf die Benutzer des Extraktionssystems und nicht auf die Benutzer (Autoren) des Wiki-Systems.

oder auch Homonyme extrahiert werden, denen durch das Konzept des Semantic MediaWikis nicht automatisch eine Semantik gegeben wird. Informationen zum aktuellen Stand der Semantic Wikis und eine Auflistung einiger Wikis, die das Konzept des Semantic MediaWikis umgesetzt haben, verwaltet „ontoworld.org“, das Wiki für die Semantic Web Community (siehe [ont06]).

In der vorliegenden Arbeit wurden zur Extraktion von semantischen Informationen Techniken aus dem Web-Content-Mining und Web-Structure-Mining verwendet. Web-Usage-Mining, das Data-Mining-Techniken zum Beispiel auf Log-Files anwendet, die das Benutzerverhalten speichern, wurde nicht mit einbezogen. Ein Log-File enthält Informationen über alle oder nur bestimmte Aktionen von einem oder mehreren Nutzern an einem Rechner, ohne dass diese davon etwas mitbekommen oder ihre Arbeit davon beeinflusst wird. Amazon, ein amerikanisches Internet Online-Versandhaus, wertet zum Beispiel die Log-Files danach aus, welche Bücher ein Kunde noch gekauft oder angesehen hat, um anderen Kunden diese zu empfehlen. „Kunden, die diesen Artikel angesehen haben, haben auch angesehen: [...]“[3]. Diesen Ansatz verfolgt auch Maik Braun in seiner Diplomarbeit zum Thema „Ermittlung nutzergruppenspezifischer Navigationspfade in Wiki-Systemen“ (siehe [Bra06]). Es wird ein Konzept erstellt, wie aus den Navigationspfaden von Nutzern eine unterstützende Nutzerhilfe generiert werden kann, die das Auffinden von weiteren Inhalten erleichtert. In aufbauenden Arbeiten könnten diese Ergebnisse, zusätzlich zu den Informationen, die aus dem Inhalt und der Verlinkung der Seiten bei der Extraktion von semantischen Informationen entstehen, hinzugezogen werden. Denn aus dem Benutzerverhalten lassen sich ebenfalls eine Menge an semantischen Beziehungen ableiten, wie das Beispiel von Amazon zeigt. Weitere Seiten, die sich der Benutzer angeschaut hat, müssen nicht unbedingt über Links aufgerufen worden sein. Sie können aber inhaltlich zusammenhängen. Als Erweiterung zu dieser Arbeit kann analysiert werden, anhand welcher Kriterien aus den Navigationspfaden zusätzliche Beziehungen ableitbar sind.

In der gesamten Arbeit wird immer von einem Anwendungskontext im Allgemeinen gesprochen, in dem semantische Informationen extrahiert werden. Ein Beispiel wäre die Integration in ein Wiki als PlugIn. Für jede Seite könnten folglich alle semantischen Beziehungen zu anderen Seiten oder zwischen Wörtern auf einen Blick angezeigt werden. Eine andere Möglichkeit wäre eine Liste jeglicher Seiten und deren Semantik, die in Beziehung zu der aktuell betrachteten stehen, anzuzeigen. Diese Liste würde nicht, wie oben beschrieben, aus den Navigationspfaden von Nutzern, sondern aus den Informationen, die im Rahmen dieser Arbeit identifiziert wurden, erstellt werden. Das sind z.B. Seiten in den selben Kategorien, weitere Seiten eines Autors oder Seiten, die sich gegenseitig verlinken.

[3] http://www.amazon.de/.

Ein anderer Anwendungsfall kann die Ergebnisse dieser Arbeit verwenden, um eine Wörterdatenbank aufzubauen. Da die Extraktion von Wortbeziehungen nicht vollständig automatisch funktioniert, kann man allerdings nicht auf die Verifikation des Benutzers und/oder eines Wörterbuches verzichten.

Literaturverzeichnis

[Ada01] Adams, Katherine C.: The Web as Database: New Extraction Technologies and Content Management. In: *ONLINE* (2001), March

[AGM03] Arasu, Arvind ; Garcia-Molina, Hector: Extracting Structured Data from Web Pages, SIGMOD 2003, 2003

[Amm04] Ammelburger, Dirk: *XML. Grundlagen der Sprache und Anwendung in der Praxis.* Hanser, 2004. – ISBN 3–446–22562–5

[App99] Appelt, Douglas E.: *Introduction to Information Extraction Technology.* 1999

[AS98] Albrecht, Schmidt ; Specht, Günther: Java, XML und Servlets zur Integration datenbankbasierter Applikationen im Web. In: *Informatik aktuell* (1998), S. 269–268

[Aum05] Aumüller, David: SHAWN: Structure helps a wiki navigate, In: W. Mueller and R. Schenkel, editors, Proceedings of the BTW-Workshop WebDB Meets IR, March 2005

[Bir06] Birnbacher, Eva: *WIKI - Back to the future.* `http://www.univie.ac.at/comment/06-1/Comment_06-1_Infodienste.pdf`. Version: 2006

[BL06] Berners-Lee, Tim: *The WorldWideWeb Browser.* `http://www.w3.org/People/Berners-Lee/WorldWideWeb.html`. Version: 06.05.2006

[Bra06] Braun, Maik: *Ermittlung nutzergruppenspezifischer Navigationspfade in Wiki-Systemen.* Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main, April 2006

[Bur04] Burget, Radek: Hierarchies in HTML Documents: Linking Text to Concepts, In: 15th International Workshop on Database and Expert Systems Applications, 2004

[Car97] Cardie, Claire: Empirical Methods in Information Extraction, In: Working Papers of ACL-97 Workshop on Natural Language Learning, 1997

[CL01] Cunningham, Ward ; Leuf, Bo: *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley Verlag, 2001. – ISBN 0–201–71499–X

[cun06] *Portland Pattern Repository*. `http://c2.com`. Version: 24.07.2006

[Cyg02] Cyganiak, Richard: *Wiki und WCMS: Ein Vergleich*. `http://richard.cyganiak.de/2002/wiki_und_wcms/wiki_und_wcms.pdf`. Version: Mai 2002

[Dav00] Davison, Brian D.: Topical Locality in the Web, In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, S. 272–279

[der06] *Politische Schlammschlacht bei Wikipedia*. `http://derstandard.at/?url=/?id=2431770`. Version: 15.05.2006

[deu03] *DEUTSCH, Neues grosses Wörterbuch, Fremdwörter*. Buch und Zeit Verlagsgesellschaft mbH, 2003. – ISBN 3–8166–0506–0

[Dro05] Drobnik, Prof. Dr. O.: Vorlesungsskript: Verteilte Systeme und Telematik II, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Fachbereich Informatik und Mathematik, WS 2004/05

[Dro05] Drobnik, Prof. Dr. O.: Vorlesungsskript: Softwaretechnik, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Fachbereich Informatik und Mathematik, SS 2005

[dud93] *DUDEN Informatik*. 2.Aufl. Dudenverlag, 1993. – ISBN 3–411–052325

[EGH05] Ebersbach, Anja ; Glaser, Markus ; Heigl, Richard: *Wiki Tools Kooperation im Web*. SpringerVerlag, 2005. – ISBN 3–540–22939–6

[Eik99] Eikvil, Line: Information Extraction from World Wide Web: a Survey, Norwegian Computing Center, July 1999

[Fer03] Ferber, Reginald: *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. 1.Aufl. dpunkt.verlag, 2003. – ISBN 3–89864–213–5

[Für02] Fürnkranz, J.: Web Structure Mining. Exploiting the Graph Structure of the Worl-Wide Web, Österreichische Gesellschaft für Artificial Intelligence (ÖGAI), 2002, S. 17–26

[Fre98] Freitag, Dayne: Information extraction from HTML, In: Proceedings of the Fifteenth Conference on Artificial Intelligence, 1998

[gal06] *Galileo Computing: Glossar Web Services.* `http://www.galileocomputing.de/glossar/gp/anzeige-8845/FirstLetter-W`. Version: 12.08.2006

[gnu06] *GNU General Public License(GPL).* `http://www.gnu.org/copyleft/fdl.html`. Version: 24.07.2006

[GY98] Gaizauskas, Robert ; Yorick, Wilks: Information Extraction: Beyond Document Retrieval, Journal of Documentation, January 1998

[HAB+97] Hobbs, Jerry R. ; Appelt, Douglas ; Bear, John ; Israel, David ; Kameyama, Megumi ; Stickel, Mark ; Tyson, Mabry: FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text, In: E. Roche & Y. Schabes (eds.) Finite State Devices for Natural Language Processing. MIT Press, 1997, S. 383–406

[HQW06] Heyer, Gerhard ; Quasthoff, Uwe ; Wittig, Thomas: *Text-Mining: Wissensrohstoff Text, Konzepte, Algorithmen, Ergebnisse.* W3L-Verlag, 2006. – ISBN 3–937137–30–0

[HZCC04] He, Bin ; Zhang, Zhen ; Chen-Chuan, Kevin C.: Knocking the Door to the Deep Web: Integrating Web Query Interaces, In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, 2004, S. 913 – 914

[JFP00] Johannes, Henkel ; Fankhauser Peter, Macherius I.: Komposition von XML Dataflows zur Integration heterogener WWW-Informationsdienste. (2000), 16. Mai

[KAS05] Krátký, Michal ; Andrt, Marek ; Svátek, Vojtěch: XML Query Support for Web Information Extraction: A Study on HTML Element Depth Distribution, In: First International Workshop on Representation and Analysis of Web Space (RAWS-05), 2005

[KS06] Kuhlins Stefan, Tredwell R.: *Wrapper Development Tools.* `www.wifo.uni-mannheim.de/~kuhlins/wrappertools/`. Version: 30.05.2006

[KT02] Kuhlins, Stefan ; Tredwell, Ross: Toolkits for Generating Wrappers. A Survey of Software Toolkits for Automated Data Extraction from Web Sites, In: Net.ObjectDays, 2002

[Lan05] Lange, Christoph: *Wiki Planen, Einrichten, Verwalten.* Computer & Literatur Verlag GmbH, 2005. – ISBN 3–936546–28–2

[Loc02] Lochbrunner, Michael: *XML als Mittler im Web Content Management.* http://www.contentmanager.de/magazin/artikel_141_xml_mittler_web_content_management.html. Version: 2002

[LT03] Lerdorf, Rasmus ; Tatroe, Kevin: *Programmieren mit PHP.* 1.Aufl. O'Reilly, 2003. – ISBN 3–89721–177–7

[Mac67] MacQueen, J.B.: Some methods for classification and analysis of multivariate observations, In: Proceedings of the 5th Berkeley Symposium on mathematical statistics and probability, 1967

[med06a] *Help:Link.* http://meta.wikimedia.org/wiki/Help:Link. Version: 14.04.2006

[med06b] *MediaWiki Architecture.* http://meta.wikimedia.org/wiki/Help:MediaWiki_architecture. Version: 17.08.2006

[MM99] Malhotra, Ashok ; Maloney, Murray: *XML Schema Requirements.* http://www.w3.org/TR/NOTE-xml-schema-req. Version: 15.02.1999

[MS99] Manning, Christopher D. ; Schütze, Hinrich: Foundations of Statistical Natural Language, In: Proceedings MIT Press, Cambridge Massachusetts, 1999

[ont06] *ontoworld.org.* http://wiki.ontoworld.org/wiki/Main_Page. Version: 01.09.2006

[RPSA99] Ravi, Kumar ; Prabhakar, Raghavan ; Sridhar, Rajagopalan ; Andrew, Tomkins: Extracting large-scale knowledge bases from the web, In: Proceedings of the 25th VLDB Conference, Edinburth, Scotland 1999

[SBP98] Soumen, Chakrabarti ; Byron, Dom ; Piotr, Indyk: Enhanced hypertext categorization using hyperlinks, In: Proceedings of the ACM SIGMOD International Conference on Management on Data, Seattle 1998

[Sch92] Schippan, Thea: *Lexikologie der deutschen Gegenwartssprache.* Max Niemeyer Verlag, 1992. – ISBN 3–484–73002–1

[Sch02] Schnitger, Prof. Dr. G.: Vorlesungsskript: Theoretische Informatik 2, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Fachbereich Informatik und Mathematik, SS 2002

[Sie92] Sieber, Helmut: *Mathematische FORMELSAMMLUNG für Gymnasien.* 1.Aufl. Ernst Klett Schulbuchverlag GmbH, 1992. – ISBN 3–12–718010–1

[SPR04] Stefano, Emilio ; Paolo, Castagna ; Roberto, Tazzoli: Playtypus Wiki: a Semantic Wiki Wiki Web, In: 1st Italian Semantic Web Workshop, 10th December 2004

[ST02] Schmidt-Thieme, Dr. L.: *KDD, Data Mining und Web Mining.* http://www.informatik.uni-freiburg.de/cgnm/lehre/wm-02w/webmining-1.pdf. Version: 2002

[SW05] Seeboerger-Weichselbaum, Michael: *PHP & XML für Web Developer, Praxis und Referenz.* 2.Aufl. Software und Support Verlag GmbH, Frankfurt 2005. – ISBN 3–935042–507

[SY00] Sundaresan, Neel ; Yi, Jeonghee: Mining the Web for Relations, In: 9th International World Wide Web Conference (WWW9), 2000

[uni06] *Universität Dortmund: Glossar.* http://www.hrz.uni-dortmund.de/docs/Glossar.html. Version: 21.09.2006

[VKV+06] Völkel, M. ; Krötzsch, M. ; Vrandecic, D. ; Haller, H. ; Studer, R.: Semantic Wikipedia, In: Proceedings of the 15th international conference on World Wide Web, May 2006

[w3ca] *Document Object Model (DOM).* http://www.w3.org/DOM/

[w3cb] *Extensible Markup Language (XML).* http://www.w3c.org/XML

[w3cc] *XSL Transformations (XSLT).* http://www.w3.org/TR/xslt

[w3c06] *XML Schema.* http://www.w3.org/XML/Schema. Version: 21.07.2006

[wik05a] *Hilfe:Spezialseiten.* http://meta.wikimedia.org/wiki/Hilfe:Spezialseiten. Version: 20.10.2005

[wik05b] *Hilfe:Versionen.* http://meta.wikimedia.org/wiki/Hilfe:Versionen. Version: 20.10.2005

[wik05c] *Hilfe:Weiterleitung.* http://meta.wikimedia.org/wiki/Hilfe:Weiterleitung. Version: 21.10.2005

[wik06a] *Wikipedia: Neue deutsche Rechtschreibung.* http://de.wikipedia.org/w/index.php?title=Neue_deutsche_Rechtschreibung\&action=history. Version: 02.08.2006

[wik06b] *Wikipedia:Formatierung.* http://de.wikipedia.org/wiki/Wikipedia:Formatierung. Version: 03.07.2006

[wik06c] *WikiEngines.* http://c2.com/cgi/wiki?WikiEngines. Version: 20.04.2006

[wik06d] *WikiMarkupStandard*. http://www.usemod.com/cgi-bin/mb.pl?WikiMarkupStandard. Version: 20.04.2006

[wik06e] *Wikipedia: CMS*. http://de.wikipedia.org/wiki/Cms. Version: 21.09.2006

[wik06f] *help:Database layout*. http://meta.wikimedia.org/wiki/Database_layout. Version: 24.07.2006

[wik06g] *WikiMatrix*. http://www.wikimatrix.org/wizard.php. Version: 24.07.2006

[wik06h] *Hilfe:Namensräume*. http://meta.wikimedia.org/wiki/Hilfe:Namensraum. Version: 25.07.2006

[wik06i] *Hilfe:Export*. http://meta.wikimedia.org/wiki/Hilfe:Export. Version: 31.07.2006

[wor06a] *Verfügbaren Webservices des Wortschatzprojektes der Universität Leipzig*. http://wortschatz.uni-leipzig.de/axis/servlet/ServiceOverviewServlet. Version: 18.08.2006

[wor06b] *Nutzungsbedingungen des Wortschatzprojektes der Universität Leipzig*. http://wortschatz.uni-leipzig.de/use.html. Version: 28.07.2006

[WS01] Winkler, Karsten ; Spiliopoulou, Myra: Extraction of Semantic XML DTDs from Texts Using Data Mining Techniques, In: Proceedings of the K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation, 2001, S. 59–68

Anhang A

Vergleich der Syntaxen

Syntax	MediaWiki	TWiki	MoinMoin
fett	'''Text'''	*Text*	'''Text'''
kursiv	''Text''	_Text_	''Text''
Unterstrichen	<u>Text</u>		__Text__
technisch	<tt>technisch</tt>		
gelöscht	<strike>gelöscht</strike>		
Überschrift	=Überschrift=	—Überschrift	=Überschrift=
Link	[[Link]]	CamelCase	CamelCase
externer Link	[http://www.link.de Beschreibung]		[url]
Bild	[[Bild:name.jpg]]	%ATTACHURL%/bild.jpg	[[Bild.name.jpg]]
No Wiki	<nowiki>Befehl</nowiki>	<nop>	<nowiki>Befehl</nowiki>

Tabelle A.1: Vergleich der Syntaxen

Anhang B

Spezialseiten

Die auf der nächsten Seite aufgeführte Liste bietet einen Überblick über alle Spezialseiten der Wikipedia[1].

[1] http://de.wikipedia.org/wiki/Spezial:Specialpages. Stand: 28.07.2006.

Alle Seiten	Meist kategorisierte Artikel
Anmelden	Meistbenutzte Bilder
Artikel mit Links in andere Namensräume	Meistbenutzte Kategorien
Artikel mit den meisten Versionen	Neue Artikel
Begriffsklärungsseiten	Neue Bilder
Benutzerverzeichnis	Nicht benutzte Vorlagen
Benutzte, aber nicht angelegte Kategorien	Nicht kategorisierte Dateien
Beobachtungsliste	Nicht kategorisierte Kategorien
Dateiliste	Nicht kategorisierte Seiten
Dateipfad	Präfixindex
Doppelte Weiterleitungen	Sackgassenartikel
Einstellungen	Seiten exportieren
Gewünschte Seiten	Seiten im MediaWiki-Namensraum
Hochladen	Statistik
Häufig verlinkte Seiten	Suche
ISBN-Suche	Suche nach MIME-Typ
Kaputte Weiterleitungen	Version
Kategorien	Verwaiste Dateien
Kurze Artikel	Verwaiste Kategorien
Lange Artikel	Verwaiste Seiten
Lange unbearbeitete Artikel	Wahlen zum Wikimedia Board of Trustees
Letzte Änderungen	Weblink-Suche
Liste blockierter IP-Adressen	Weiterleitungsliste
Liste der Wikimedia-Wikis	Zitierhilfe
Logbücher	Zufällige Weiterleitung
	Zufälliger Artikel

Tabelle B.1: Spezialseiten

Anhang C

Stopwortliste

Eine Stopwortliste im Deutschen kann zum Beispiel wie unten aufgelistet aussehen. Durch Hinzufügung oder Löschung von Wörtern ist sie an eigene Bedürfnisse anpassbar.

ab aber ähnlich alle allein allem aller alles allgemein als also am an andere anderes auch auf aus außer bei beim besonders bevor bietet bis bzw da dabei dadurch dafür daher dann daran darauf daraus das dass davon davor dazu dem den denen denn dennoch der deren des deshalb die dies diese diesem diesen dieser dieses doch dort durch eben ein eine einem einen einer eines einfach er es etc etwa etwas findet für ganz ganze ganzem ganzen ganzer ganzes gar gehen gilt gleich gute hat hinter ihm ihr ihre ihrem ihren ihrer ihres im immer in ist ja je jede jedem jeden jeder jedes jedoch jene jenem jenen jener jenes jetzt kann kein keine keinem keinen keiner keines kommen kommt können leicht machen mal man mehr mehrere meist mit muss nach neu neue neuem neuen neuer neues nicht noch nur ob oder oft ohne schließlich schon schwierig sehr sein seine seinem seinen seiner seines seit selbst sich sie sind so sodass solch solche solchem solchen solcher solches sollte sollten soviel sowieso statt stehen über um und uns unser unsere unseren unseres unter viel viele vom von vor wann war waren was welche welcher wenig wenige weniger wenn wer werden wie wieder wird wirklich wo wurde wurden zu zum zur zwar zwischen

Anhang D

Output.xsd

Das unten abgebildtet XML-Schema definiert das endgültig erzeugte XML-Dokument (Output.xml).

```
1  <?xml version="1.0" encoding="Utf-8"?>
2  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3          xmlns  ="http://localhost/mediawiki/includes/SemInfo"
4          elementFormDefault="qualified"
5          attributeFormDefault="qualified"
6          targetNamespace="http://localhost/mediawiki/includes/
               SemInfo">
7
8  <!-- Our root element -->
9  <element name="page" type="page"/>
10
11 <xsd:complexType name="page">
12 <xsd:sequence>
13   <xsd:complexType name="siteinfo" type="siteinfo"/>
14   <xsd:complexType name="relations_between_terms" type="
         relations_between_terms" minOccurs="0" maxOccurs="1"/>
15     <xsd:complexType name="links" type="links" minOccurs="0"
           maxOccurs="1"/>
16     <xsd:complexType name="externallinks" type="externallinks"
           minOccurs="0" maxOccurs="1"/>
17     <xsd:complexType name="imagelinks" type="imagelinks"
           minOccurs="0" maxOccurs="1"/>
18     <xsd:complexType name="backlinks" type="backlinks"
           minOccurs="0" maxOccurs="1"/>
19     <xsd:complexType name="compare_wikilinks_backlinks"
20         type="compare_wikilinks_backlinks" minOccurs="0"
               maxOccurs="1"/>
21     <xsd:complexType name="categories" type="categories"
           minOccurs="0"
22         maxOccurs="1"/>
23     <xsd:complexType name="majorcategories" type="
           majorcategories" minOccurs="0" maxOccurs="1"/>
```

```
24      <xsd:complexType name="subcategories" type="subcategories"
            minOccurs="0" maxOccurs="1"/>
25      <xsd:complexType name="pages_in_category"
26          type="pages_in_category" minOccurs="0" maxOccurs="1"/>
27    <xsd:complexType name="keywords" type="keywords"
28       minOccurs="0" maxOccurs="1"/>
29    <xsd:complexType name="markup_words" type="markup_words"
30       minOccurs="0" maxOccurs="1"/>
31    <xsd:element name="last_modified" type="xsd:date"
32       minOccurs="0" maxOccurs="1"/>
33    <xsd:element name="num_changes" type="xsd:integer"
34       minOccurs="0" maxOccurs="1"/>
35    <xsd:element name="num_authors" type="xsd:integer"
36       minOccurs="0" maxOccurs="1"/>
37    <xsd:complexType name="authorpages" type="authorpages"
          minOccurs="0" maxOccurs="1"/>
38    <xsd:complexType name="metadata" type="metadata"
39       minOccurs="0" maxOccurs="1"/>
40  </xsd:sequence>
41  <xsd:attribute name="name" type="xsd:string"/>
42  </xsd:complexType>
43
44  <xsd:complexType name="siteinfo">
45  <xsd:sequence>
46    <xsd:complexType name="sitename" type="sitename"/>
47    <xsd:complexType name="generator" type="generator"/>
48    <xsd:complexType name="authors" type="authors"/>
49    <xsd:complexType name="timestamps" type="timestamps"/>
50  </xsd:sequence>
51  </xsd:complexType>
52
53  <xsd:complexType name="sitename">
54  <xsd:sequence>
55     <xsd:element name="sitename" type="xsd:string"/>
56  </xsd:sequence>
57   <xsd:attribute name="relation" type="xsd:string"
58      default="extracted-from" />
59   <xsd:attribute name="result" type="xsd:string"
60      default="correct" />
61  </xsd:complexType>
62
63  <xsd:complexType name="generator">
64  <xsd:sequence>
65     <xsd:element name="generator" type="xsd:string"/>
66  </xsd:sequence>
67  <xsd:attribute name="relation" type="xsd:string"
```

```
68      default="generated-from"/>
69  <xsd:attribute name="result" type="xsd:string"
70      default="correct"/>
71  </xsd:complexType>
72
73  <xsd:complexType name="authors">
74  <xsd:sequence>
75    <xsd:complexType name="author" type="author" minOccurs="1"
          maxOccurs="unbounded"/>
76  </xsd:sequence>
77  </xsd:complexType>
78
79  <xsd:complexType name="author">
80  <xsd:sequence>
81    <xsd:element name="author" type="xsd:string" minOccurs="1"
          maxOccurs="unbounded"/>
82  </xsd:sequence>
83  <xsd:attribute name="relation" type="xsd:string"
84      default="wrote"/>
85  <xsd:attribute name="result" type="xsd:string"
86      default="correct"/>
87  </xsd:complexType>
88
89  <xsd:complexType name="timestamps">
90  <xsd:sequence>
91    <xsd:complexType name="timestamp" type="timestamp"
92        minOccurs="1" maxOccurs="unbounded"/>
93  </xsd:sequence>
94  </xsd:complexType>
95
96  <xsd:complexType name="timestamp">
97  <xsd:sequence>
98    <xsd:element name="timestamp" type="xsd:string" minOccurs="1"
          maxOccurs="unbounded"/>
99  </xsd:sequence>
100 <xsd:attribute name="relation" type="xsd:string"
101     default="edited-on"/>
102 <xsd:attribute name="result" type="xsd:string"
103     default="correct"/>
104 </xsd:complexType>
105
106 <xsd:complexType name="relations_between_terms">
107 <xsd:sequence>
108   <xsd:complexType name="relation_between_terms"
109       type="relation_between_terms" maxOccurs="unbounded"/>
110 </xsd:sequence>
```

```
111 </xsd:complexType>
112
113 <xsd:complexType name="relation_between_terms">
114 <xsd:sequence>
115   <xsd:element name="term1" type="xsd:string" />
116   <xsd:element name="term2" type="xsd:string" />
117 </xsd:sequence>
118 <xsd:attribute name="relation" type="xsd:string"/>
119 <xsd:attribute name="result" type="xsd:string"
120     default="possibly-correct" />
121 </xsd:complexType>
122
123 <xsd:complexType name="links">
124 <xsd:sequence>
125   <xsd:complexType name="link" type="link" minOccurs="1"
        maxOccurs="unbounded"/>
126 </xsd:sequence>
127 </xsd:complexType>
128
129 <xsd:complexType name="link">
130 <xsd:sequence>
131   <xsd:element name="type" type="xsd:string"/>
132   <xsd:element name="name" type="xsd:string"/>
133 </xsd:sequence>
134 <xsd:attribute name="relation" type="xsd:string"/>
135 <xsd:attribute name="result" type="xsd:string"/>
136 </xsd:complexType>
137
138 <xsd:complexType name="externallinks">
139 <xsd:sequence>
140   <xsd:complexType name="externallink" type="externallink"
        maxOccurs="unbounded"/>
141 </xsd:sequence>
142 </xsd:complexType>
143
144 <xsd:complexType name="externallink">
145 <xsd:sequence>
146   <xsd:element name="url" type="xsd:anyURI" />
147   <xsd:element name="description" type="xsd:string" />
148 </xsd:sequence>
149 <xsd:attribute name="relation" type="xsd:string"/>
150 <xsd:attribute name="result" type="xsd:string"/>
151 </xsd:complexType>
152
153 <xsd:complexType name="imagelinks">
154 <xsd:sequence>
```

```
155   <xsd:complexType name="imagelink" type="imagelink"
156       maxOccurs="unbounded"/>
157 </xsd:sequence>
158 </xsd:complexType>
159
160 <xsd:complexType name="imagelink">
161 <xsd:sequence>
162   <xsd:element name="name" type="xsd:string"/>
163   <xsd:element name="description" type="xsd:string"/>
164 </xsd:sequence>
165 <xsd:attribute name="relation" type="xsd:string"/>
166 <xsd:attribute name="result" type="xsd:string"/>
167 </xsd:complexType>
168
169 <xsd:complexType name="backlinks">
170 <xsd:sequence>
171   <xsd:complexType name="backlink" type="backlink"
172       maxOccurs="unbounded"/>
173 </xsd:sequence>
174 </xsd:complexType>
175
176 <xsd:complexType name="backlink">
177 <xsd:sequence>
178   <xsd:element name="link" type="xsd:string"
179       maxOccurs="unbounded" />
180 </xsd:sequence>
181 <xsd:attribute name="relation" type="xsd:string"/>
182 <xsd:attribute name="result" type="xsd:string"/>
183 </xsd:complexType>
184
185 <xsd:complexType name="compare_wikilinks_backlinks">
186 <xsd:sequence>
187   <xsd:complexType name="linkname" type="linkname"
188       maxOccurs="unbounded" />
189 </xsd:sequence>
190 </xsd:complexType>
191
192 <xsd:complexType name="linkname">
193 <xsd:sequence>
194   <xsd:element name="linkname" type="xsd:string"
195       maxOccurs="unbounded" />
196 </xsd:sequence>
197 <xsd:attribute name="relation" type="xsd:string"/>
198 <xsd:attribute name="result" type="xsd:string"/>
199 </xsd:complexType>
200
```

```
<xsd:complexType name="categories">
<xsd:sequence>
  <xsd:complexType name="category" type="category"
      maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="category">
<xsd:sequence>
  <xsd:element name="category" type="xsd:string"
      maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="relation" type="xsd:string"/>
<xsd:attribute name="result" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="majorcategories">
<xsd:sequence>
  <xsd:complexType name="majorcategory" type="majorcategory"
      maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="majorcategory">
<xsd:sequence>
  <xsd:element name="majorcategory" type="xsd:string"
      maxOccurs="unbounded"/>
</xsd:sequence>
  <xsd:attribute name="relation" type="xsd:string"
      default="major-term"/>
  <xsd:attribute name="result" type="xsd:string"
      default="possibly-correct"/>
</xsd:complexType>

<xsd:complexType name="subcategories">
<xsd:sequence>
  <xsd:complexType name="subcategory" type="subcategory"
      maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="subcategory">
<xsd:sequence>
  <xsd:element name="subcategory" type="xsd:string"
      maxOccurs="unbounded"/>
</xsd:sequence>
```

```
247   <xsd:attribute name="relation" type="xsd:string"
248       default="sub-term"/>
249   <xsd:attribute name="result" type="xsd:string"
250       default="possibly-correct"/>
251 </xsd:complexType>
252
253 <xsd:complexType name="pages_in_category">
254 <xsd:sequence>
255   <xsd:complexType name="categorypage" type="categorypage"
        maxOccurs="unbounded"/>
256 </xsd:sequence>
257 </xsd:complexType>
258
259 <xsd:complexType name="categorypage">
260 <xsd:sequence>
261   <xsd:element name="categoryname" type="xsd:string"/>
262   <xsd:element name="page" type="xsd:string"/>
263 </xsd:sequence>
264 <xsd:attribute name="relation" type="xsd:string"/>
265 <xsd:attribute name="result" type="xsd:string"/>
266 </xsd:complexType>
267
268 <xsd:complexType name="keywords">
269 <xsd:sequence>
270   <xsd:complexType name="keyword" type="keyword"
271       maxOccurs="unbounded"/>
272 </xsd:sequence>
273 </xsd:complexType>
274
275 <xsd:complexType name="keyword">
276 <xsd:sequence>
277   <xsd:element name="frequency" type="xsd:integer"/>
278   <xsd:element name="word" type="xsd:string"/>
279 </xsd:sequence>
280 </xsd:complexType>
281
282 <xsd:complexType name="markup_words">
283 <xsd:sequence>
284   <xsd:complexType name="markup_word" type="markup_word"
        maxOccurs="unbounded"/>
285 </xsd:sequence>
286 </xsd:complexType>
287
288 <xsd:complexType name="markup_word">
289 <xsd:sequence>
290   <xsd:element name="type" type="xsd:string"/>
```

```
291   <xsd:element name="word" type="xsd:string"/>
292 </xsd:sequence>
293 </xsd:complexType>
294
295
296 <xsd:complexType name="authorpages">
297 <xsd:sequence>
298   <xsd:complexType name="authorname" type="authorname"
          maxOccurs="unbounded"/>
299 </xsd:sequence>
300 </xsd:complexType>
301
302 <xsd:complexType name="authorname">
303 <xsd:sequence>
304   <xsd:element name="name" type="xsd:string"/>
305   <xsd:element name="page" type="xsd:string"/>
306 </xsd:sequence>
307 <xsd:attribute name="relation" type="xsd:string"
308     default="wrote"/>
309 <xsd:attribute name="result" type="xsd:string"
310     default="correct"/>
311 </xsd:complexType>
312
313 <xsd:complexType name="metadata">
314 <xsd:sequence>
315   <xsd:element name="meta" type="xsd:string"
316       maxOccurs="unbounded"/>
317 </xsd:sequence>
318 </xsd:complexType>
319
320 </xsd:schema>
```

Quellcode D.1: XML-Schema: Output.xsd

Anhang E

export-03.xsd

Das unten abgebildete XML-Schema[1] definiert das XML-Dokument, das durch die Spezialfunktion „Seiten exportieren “ erzeugt wird.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
   This is an XML Schema description of the format
   output by MediaWikis Special:Export system.

   Version 0.2 adds optional basic file upload info support,
   which is used by our OAI export/import submodule.

   Version 0.3 adds some site configuration information such
   as a list of defined namespaces.

   The canonical URL to the schema document is:
   http://www.mediawiki.org/xml/export-0.3.xsd

   Use the namespace:
   http://www.mediawiki.org/xml/export-0.3/
-->

<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:mw="http://www.mediawiki.org/xml/export-0.3/"
        targetNamespace="http://www.mediawiki.org/xml/export
            -0.3/"
        elementFormDefault="qualified">

  <annotation>
    <documentation xml:lang="en">
      MediaWikis page export format
    </documentation>
  </annotation>

```

[1] http://mediawiki.org/xml/export-0.3/ Stand: 06.08.2006.

```
30  <!-- Need this to reference xml:lang -->
31  <import namespace="http://www.w3.org/XML/1998/namespace"
32    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
33
34  <!-- Our root element -->
35  <element name="mediawiki" type="mw:MediaWikiType"/>
36
37  <complexType name="MediaWikiType">
38    <sequence>
39      <element name="siteinfo" type="mw:SiteInfoType"
40        minOccurs="0" maxOccurs="1"/>
41      <element name="page" type="mw:PageType"
42        minOccurs="0" maxOccurs="unbounded"/>
43    </sequence>
44    <attribute name="version" type="string" use="required"/>
45    <attribute ref="xml:lang" use="required"/>
46  </complexType>
47
48  <complexType name="SiteInfoType">
49    <sequence>
50      <element name="sitename" type="string" minOccurs="0"/>
51      <element name="base" type="anyURI" minOccurs="0"/>
52      <element name="generator" type="string" minOccurs="0"/>
53      <element name="case" type="mw:CaseType" minOccurs="0"/>
54      <element name="namespaces" type="mw:NamespacesType"
          minOccurs="0"/>
55    </sequence>
56  </complexType>
57
58  <simpleType name="CaseType">
59    <restriction base="NMTOKEN">
60      <!-- Cannot have two titles differing only by case of
          first letter. -->
61      <!-- Default behavior through 1.5, \$wgCapitalLinks =
          true -->
62      <enumeration value="first-letter"/>
63
64      <!-- Complete title is case-sensitive -->
65      <!-- Behavior when \$wgCapitalLinks = false -->
66      <enumeration value="case-sensitive"/>
67
68      <!-- Cannot have two titles differing only by case. -->
69      <!-- Not yet implemented as of MediaWiki 1.5 -->
70      <enumeration value="case-insensitive"/>
71    </restriction>
72  </simpleType>
```

```
73
74  <complexType name="NamespacesType">
75    <sequence>
76      <element name="namespace" type="mw:NamespaceType"
77        minOccurs="0" maxOccurs="unbounded"/>
78    </sequence>
79  </complexType>
80
81  <complexType name="NamespaceType">
82    <simpleContent>
83      <extension base="string">
84        <attribute name="key" type="integer"/>
85      </extension>
86    </simpleContent>
87  </complexType>
88
89  <complexType name="PageType">
90    <sequence>
91      <!-- Title in text form. (Using spaces, not underscores;
            with namespace) -->
92      <element name="title" type="string"/>
93
94      <!-- optional page ID number -->
95      <element name="id" type="positiveInteger" minOccurs="0"/>
96
97      <!-- comma-separated list of string tokens, if present --
            >
98      <element name="restrictions" type="string"
99        minOccurs="0"/>
100
101     <!-- Zero or more sets of revision or upload data -->
102     <choice minOccurs="0" maxOccurs="unbounded">
103       <element name="revision" type="mw:RevisionType"/>
104       <element name="upload" type="mw:UploadType"/>
105     </choice>
106   </sequence>
107 </complexType>
108
109 <complexType name="RevisionType">
110   <sequence>
111     <element name="id" type="positiveInteger" minOccurs="0"/>
112     <element name="timestamp" type="dateTime"/>
113     <element name="contributor" type="mw:ContributorType"/>
114     <element name="minor" minOccurs="0"/>
115     <element name="comment" type="string" minOccurs="0"/>
116     <element name="text" type="mw:TextType"/>
```

```
117       </sequence>
118     </complexType>
119
120     <complexType name="TextType">
121       <simpleContent>
122         <extension base="string">
123           <attribute ref="xml:space" use="optional"
124             default="preserve"/>
125         </extension>
126       </simpleContent>
127     </complexType>
128
129     <complexType name="ContributorType">
130       <sequence>
131         <element name="username" type="string" minOccurs="0"/>
132         <element name="id" type="positiveInteger" minOccurs="0"/>
133
134         <element name="ip" type="string" minOccurs="0"/>
135       </sequence>
136     </complexType>
137
138     <complexType name="UploadType">
139       <sequence>
140         <!-- Revision-style data... -->
141         <element name="timestamp" type="dateTime"/>
142         <element name="contributor" type="mw:ContributorType"/>
143         <element name="comment" type="string" minOccurs="0"/>
144
145         <!-- Filename. (Using underscores, not spaces. No
146             'Image:' namespace marker.) -->
147         <element name="filename" type="string"/>
148
149         <!-- URI at which this resource can be obtained -->
150         <element name="src" type="anyURI"/>
151
152         <element name="size" type="positiveInteger" />
153
154         <!-- TODO: add other metadata fields -->
155       </sequence>
156     </complexType>
157
158 </schema>
```

Quellcode E.1: XML-Schema: export-0.3.xsd

Anhang F

Config.xml

```
<wikis>
  <wiki>
    <name>MediaWikiTest</name>
    <host>localhost</host>
    <url>/mediawiki/index.php</url>
    <xml>?title=Spezial:Export/</xml>
  </wiki>
  <wiki>
    <name>Wikipedia</name>
    <host>de.wikipedia.org</host>
    <url>/wiki/</url>
    <xml>Spezial:Export/</xml>
  </wiki>
  <wiki>
    <name>WikiBooks</name>
    <host>de.wikibooks.org</host>
    <url>/wiki/</url>
    <xml>Spezial:Export/</xml>
  </wiki>
  <wiki>
    <name>Wiktionary</name>
    <host>de.wiktionary.org</host>
    <url>/wiki/</url>
    <xml>Spezial:Export/</xml>
  </wiki>
</wikis>
```

Quellcode F.1: Konfigurationsdatei: Config.xml

Anhang G

Prototyp auf CD-ROM

Auf der beiliegenden CD-ROM befindet sich der Prototyp. Die Readme-Datei gibt eine Anleitung zur Installation.