

Universität Karlsruhe (TH)

STUDIENARBEIT

**Integration von Wiki-Inhalten -  
Entwurf und Implementierung einer  
Schnittstelle zur Verarbeitung der  
Inhalte von Wiki-Seiten**

von

Andreas Kurz

eingereicht am 19.12.2006 am  
Institut für Angewandte Informatik  
und Formale Beschreibungsverfahren (AIFB)  
der Universität Karlsruhe

Referent: Prof. Dr. Studer  
Betreuer: Dipl. Inform. Max Voelkel

# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einleitung</b>  | <b>2</b>  |
| 1.1      | Motivation . . . . .                                     | 2         |
| 1.2      | Idee . . . . .   | 3         |
| 1.3      | Aufgabenstellung . . . . .                               | 3         |
| 1.4      | Anwendungsfälle . . . . .                                | 4         |
| 1.5      | Definitionen . . . . .                                   | 6         |
| <b>2</b> | <b>Schnittstellen und Datenformate</b>                   | <b>8</b>  |
| 2.1      | Bestehende Schnittstellen einiger Wikis . . . . .        | 8         |
| 2.1.1    | MediaWiki . . . . .                                      | 8         |
| 2.1.2    | JspWikiApi . . . . .                                     | 8         |
| 2.1.3    | Fazit zu den Schnittstellen . . . . .                    | 9         |
| 2.2      | Verwendete Datenstrukturen . . . . .                     | 9         |
| 2.2.1    | Wiki Interchange Format . . . . .                        | 9         |
| 2.2.2    | RDF . . . . .  | 9         |
| <b>3</b> | <b>Entwurf</b>   | <b>11</b> |
| 3.1      | Integrationsmöglichkeiten . . . . .                      | 11        |
| 3.2      | Modellierung der zu integrierenden Daten . . . . .       | 11        |
| 3.3      | Architektur . . . . .                                    | 12        |
| 3.4      | Datenstrukturen . . . . .                                | 15        |
| 3.4.1    | Zugriff der Adapter auf die Seiten eines Wikis . . . . . | 15        |
| 3.4.2    | Vorteile eines Zwischenformats . . . . .                 | 15        |
| 3.4.3    | Zwischenformat . . . . .                                 | 16        |
| 3.4.4    | Metadaten und RDFa . . . . .                             | 18        |
| 3.4.5    | Transformation . . . . .                                 | 22        |
| <b>4</b> | <b>Implementierung</b>                                   | <b>24</b> |
| 4.1      | Architektur . . . . .                                    | 24        |
| 4.1.1    | Repository . . . . .                                     | 25        |
| 4.1.2    | WifPage . . . . .  | 25        |
| 4.2      | Transformation . . . . .                                 | 26        |
| <b>5</b> | <b>Evaluation</b>  | <b>27</b> |
| <b>6</b> | <b>Zusammenfassung und Ausblick</b>                      | <b>29</b> |

# 1 Einleitung

## 1.1 Motivation

Ein Wiki ist eine im World Wide Web, im Intranet eines Unternehmens oder auch nur auf einem Rechner verfügbare Seitensammlung. Die Seiten sind untereinander verlinkt. Der Inhalt kann entweder von allen oder nur von den Nutzern mit bestimmten Rechten verändert oder hinzugefügt werden. Dabei sind keine HTML-Kenntnisse notwendig, sondern nur die der Syntax der jeweiligen Wiki-Software. Weitere Merkmale eines Wikis sind Funktionen wie die Seitensuche, die Auflistung aller Seiten und das Anzeigen der Änderungen einer Seite<sup>1</sup>.

Aus allen Wikis ragt die *Wikipedia* heraus. Allein die deutsche Version umfasst mehr als 450 Tausend Artikel<sup>2</sup>. Ausserdem wird sie von *Alexa.com* auf Platz 17 der meistbesuchten Internetseiten gelistet<sup>3</sup>, was den Schluss zulässt, dass sie sehr vielen Internetnutzern bekannt sein muss. Es gibt noch unzählige kleinere Wikis wie zum Beispiel das *Stadtwiki Karlsruhe*<sup>4</sup>.

Die Wikis laufen auf vielen verschiedenen Wiki-Softwares (für Definitionen siehe Kapitel 1.5), wobei jede Wiki-Software eine andere Syntax besitzt. Die am weitesten verbreitete ist die Syntax von *MediaWiki*. Das resultiert aus der Tatsache, dass die *Wikipedia* auf ihr läuft. Andere bekannte Wiki-Softwares sind *TWiki*, oft in Intranets eingesetzt, *PukiWiki*, *TikiWiki*, *PhpWiki* und *JspWiki*. Insgesamt gibt es zurzeit mehr als 100 verschiedene Wiki-Softwares, wobei es immer mehr werden<sup>5</sup>.

Ein Grund für die Beliebtheit der Wikis ist die Tatsache, dass die Nutzer den Inhalt der Seiten in ihrem Wiki selbst hinzufügen oder verändern können. Möchte man zum Beispiel eine Seite aus einem Wiki in ein anderes Wiki kopieren, so muss man meistens die Syntax dieser Seite komplett umschreiben, sofern die zwei Wikis nicht auf der gleichen Wiki-Software laufen.

Was ist wenn man mehrere Seiten transferieren muss? Dann muss man das Kopieren und Anpassen für jede dieser Seiten ausführen. Hinzu kommt der Suchaufwand für jede Seite. Also hat man eine langwierige monotone Arbeit vor sich, die man nur allzu gerne automatisieren würde.

Desweiteren wäre es praktisch zwei Wikis synchronisieren zu können. Idealerweise gibt man einfach einen Zeitpunkt an und dann werden alle Seiten, die seit diesem Zeitpunkt geändert wurden, in dem anderen Wiki ak-

---

<sup>1</sup><http://c2.com/cgi/wiki?WikiPrinciples>

<sup>2</sup><http://de.wikipedia.org/wiki/Hauptseite>

<sup>3</sup>[http://www.alexa.com/site/ds/top\\_500](http://www.alexa.com/site/ds/top_500)

<sup>4</sup><http://ka.stadtwiki.net>

<sup>5</sup><http://c2.com/cgi/wiki?WikiEngines>

tualisiert. Natürlich kann man auf diese Weise ein ganzes Wiki in ein anderes transferieren, indem man zum Beispiel einen Zeitpunkt angibt, der vor der Entstehung der ältesten Seite des zu kopierenden Wikis liegt.

## 1.2 Idee

Die Idee ist nun ein System zu entwerfen und zu implementieren, das imstande ist, diese Aufgaben für die Nutzer automatisch zu übernehmen. Es soll aus einem Wiki mit Syntax A Seiten auslesen, diese dann in einem Zwischenspeicherformat ablegen, sie in die Seiten des Wiki mit der Syntax B umwandeln und sie dann in dieses Wiki einspeisen können. Das Zwischenspeicherformat soll auf dem WIF [1] aufgebaut werden.

Der Vorteil der Verwendung eines Zwischenformats ist: man braucht nur  $n$  verschiedene Adapter-Schnittstellen bei  $n$  verschiedenen Wiki-Syntaxen. Ohne eines Zwischenformats werden  $n^2$  verschiedene Adapter-Schnittstellen benötigt. Dadurch erreicht man eine Reduktion der Komplexität von  $n^2$  auf  $n$ , da somit nicht für jede zwei verschiedene Wiki-Softwares eine eigene Transformationsimplementierung gebraucht wird. Das System soll natürlich nicht nur mit einer einzigen Seite, sondern auch mit einer Menge von Seiten operieren können.

## 1.3 Aufgabenstellung

Als erste Teilaufgabe soll das Softwaresystem entworfen werden. Die wichtigsten Anwendungsfälle für das System sollen erkannt und beschrieben werden. Die Architektur soll entworfen werden. Das Hauptziel ist: Die Architektur sollte gut erweiterbar sein.

Es sollen zwei Wiki-Softwares ausgewählt werden. Zu diesen sollen die Adapterschnittstellen entworfen werden. Es soll die Programmschnittstelle entworfen werden. Anschliessend sollen die Anwendungsfälle als JUnit-Testfälle modelliert werden.

Die zweite Teilaufgabe ist die Implementierung des Entwurfs. Es soll die Programmschnittstelle implementiert werden. Die erste Adapterschnittstelle soll entstehen. Diese Adapterschnittstelle soll

- mit der Wikianwendung kommunizieren können (lesen, schreiben, login).
- den Seiteninhalt aus der Wiki-Syntax oder aus der HTML-Seite herauslesen können.
- bei Bedarf diesen Inhalt in bestimmte Wiki-Syntax transformieren können.

Es folgt die Implementierung des zweiten Adapters mit den gleichen Funktionalitäten wie der erste. Ein Zwischenformat zur Speicherung des Backups und als Zwischenschritt von einer Wiki-Syntax zur anderen soll entworfen werden. Anschliessend sollen Gesamttests inklusive Fehlerbeseitigung und Optimierungen durchgeführt werden.

## 1.4 Anwendungsfälle

In diesem Abschnitt werden die wichtigsten Anwendungsfälle der Integration von Wikis vorgestellt und veranschaulicht.

**Anwendungsfall 1 (Abbildung 1):** Bob hat ein Wiki. Er stellt es über einen Provider im Internet der Welt zur Verfügung. Bob vertraut seinem Provider nicht ganz. Er möchte die Seiten in seinem Wiki einer Datensicherung unterziehen. Diese Datensicherung will er auf lokalen Speichermedien, zum Beispiel auf seiner Festplatte, ablegen. Es sollte eine Möglichkeit geben, die gesicherten Daten in sein Wiki zurückzuschreiben.

**Anwendungsfall 2 (Abbildung 2):** Bob hat ein Wiki. Es läuft auf MediaWiki. Sein Kollege Peter hat auch ein Wiki. Es läuft auf JspWiki. Peter möchte alle Seiten, die Bobs Wiki hat, in sein Wiki kopieren.

**Anwendungsfall 3 (Abbildung 3):** Bob hat ein Wiki. Es läuft auf MediaWiki. Sein Kollege Peter hat auch ein Wiki. Es läuft auf JspWiki. Peter möchte alle Seiten, die in Bobs Wiki aber nicht in seinem Wiki sind, in sein Wiki kopieren. Die Seiten die es sowohl in Bobs Wiki als auch in Peters Wiki gibt, sollen im lokalen Speicher abgelegt werden. Peter kann sich dann überlegen, wie er mit diesen Seiten verfahren möchte.

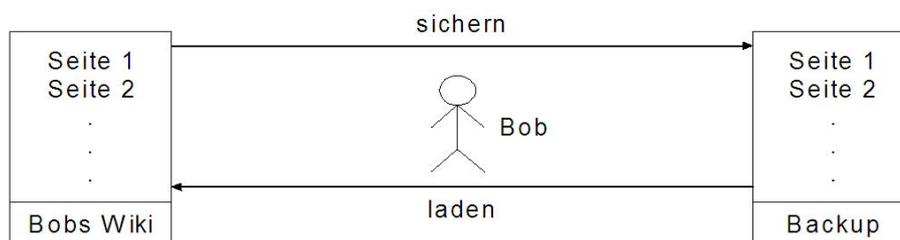


Abbildung 1: Anlegen eines Backups eines Wikis

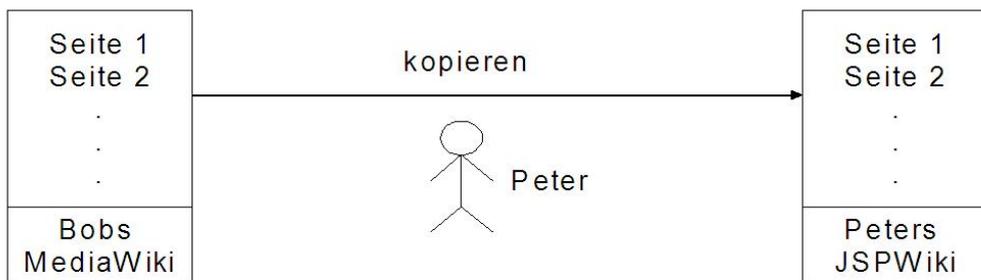


Abbildung 2: Kopieren aller Seiten eines Wikis in ein anderes

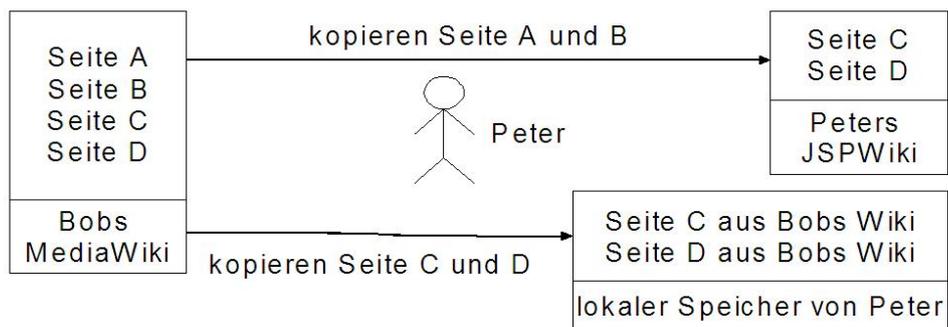


Abbildung 3: Kopieren fehlender Seiten eines Wikis in ein anderes

**Anwendungsfall 4 (Abbildung 4):** Bob hat ein Wiki. Peter hat Bobs Wiki kopiert und so sein eigenes Wiki erstellt. Einige Zeit später möchte Peter alle Seiten aus Bobs Wiki, die seit dem Kopieren verändert wurde oder neu hinzugekommen sind, in sein Wiki kopieren.



Abbildung 4: Kopieren aktualisierter Seiten

**Anwendungsfall 5 (Abbildung 5):** Bob hat ein Wiki. Peter hat ein Wiki. Peter stellt fest, dass die Suche nach 'XML' in seinem Wiki erfolglos bleibt. Er möchte alle Seiten aus Bobs Wiki kopieren, die bei einer Suchanfrage von 'XML' in Bobs Wiki in der Trefferliste zurückgegeben werden.

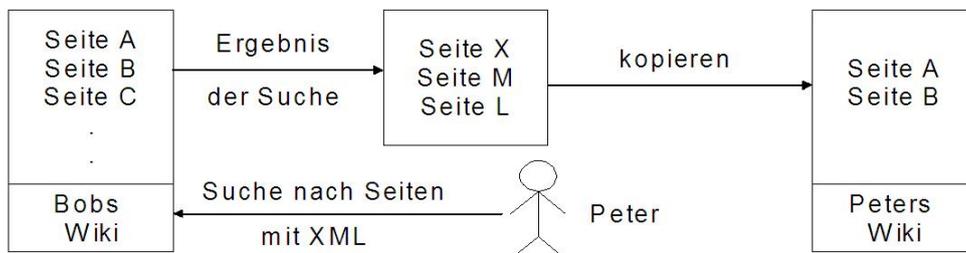


Abbildung 5: Kopieren der Seiten aus dem Ergebnis der Suche

## 1.5 Definitionen

In diesem Abschnitt werden einige Begriffe definiert, die das Verständnis der Arbeit erleichtern.

**Wiki-Seite:** Eine Seite in einem Wiki. Sie hat einen eindeutigen Titel. Der Inhalt der Seite kann von den Benutzern geändert werden.

**Wiki-Software:** Eine Anwendung, die ein Wiki verwaltet. Die Wiki-Softwares unterscheiden sich meistens in der Syntax und in der Visualisierung der Seiten.

**Wiki-Instanz:** Ein konkretes Wiki, welches auf einer bestimmten Wiki-Software läuft.

**Beispiel:** Wikipedia ist eine Wiki-Instanz. Wikipedia läuft auf der Media-Wiki. MediaWiki ist eine Wiki-Software.

## 2 Schnittstellen und Datenformate

In diesem Kapitel wird auf die Schnittstellen einiger Wiki-Softwares eingegangen. Ausserdem werden die Konzepte der in der Arbeit verwendeten Datenstrukturen vorgestellt.

### 2.1 Bestehende Schnittstellen einiger Wikis

In diesem Abschnitt werden die Schnittstellen von MediaWiki-Software und JspWiki-Software vorgestellt.

#### 2.1.1 MediaWiki

Die MediaWiki-Software hat eine XML-Schnittstelle. Sie heisst dort 'Export-XML Funktion'. Ein Artikel wird dem Benutzer in XML über diese Schnittstelle übergeben. Der Benutzer konstruiert hierzu eine URL. Diese sieht im Falle der deutschen Wikipedia so aus: *http://de.wikipedia.org/wiki/Spezial:Exportieren/Artikelname*. Nach der Eingabe dieser URL in der Adressleiste des Browsers bekommt der Benutzer den Artikel mit dem Titel '*Artikelname*' in XML zurück.

Es gibt eine Möglichkeit, die gesamten Seiten eines Wikis als nur eine XML-Datei zu bekommen.

Der Text des Artikels liegt als ein XML-Element mit dem Namen 'text' vor. Der Text ist in der MediaWiki-Syntax. Das ist nicht wirklich brauchbar für die Integration in Wikis mit anderer Syntax. Die Metadaten wie z.B. der letzte Author oder der letzte Änderungszeitpunkt sind in der Export-XML-Datei als XML-Elemente enthalten.

#### 2.1.2 JspWikiApi

Die JspWiki-Software besitzt eine XML-RPC-Schnittstelle. In dieser Arbeit [2] beschreibt J. Jalkanen eine ähnliche Schnittstelle. Über diese Schnittstelle kann der Benutzer eine JspWiki-Seite in der JspWiki-Syntax oder in HTML bekommen. Auch ist es möglich ältere Versionen einer JspWiki-Seite in der JspWiki-Syntax oder in HTML zu bekommen.

Weitere Funktionalitäten der XML-RPC-Schnittstelle sind:

- Der Benutzer kann eine Liste aller Seiten des JspWikis abrufen.
- Es gibt die Möglichkeit eine Seiteninfo zu einer Wiki-Seite zu erfragen. Diese Seiteninfo besteht aus dem Seitennamen, dem letzten Änderungszeitpunkt, dem Author und der aktuellen Version. Zudem ist es möglich

die Seiteninfo mit expliziter Versionsangabe abzufragen, um die Seiteninfo zu einer bestimmten Vorgängerversion einer Seite zu erhalten.

- Der Benutzer kann eine Liste der Links, die von einer Seite ausgehen, abrufen.

### **2.1.3 Fazit zu den Schnittstellen**

Beide Schnittstellen bieten die Zugriffsmöglichkeiten auf die Wiki-Seiten. Die Text der Wiki-Seiten wird in der jeweiligen Wiki-Syntax erhalten. Bei Jsp-Wiki hat man die Möglichkeit den Text in HTML zu erhalten. Sowohl in der Wiki-Syntax als auch in HTML ist der Text umständlich in eine andere Wiki-Syntax zu transformieren. Die Transformation ist aber notwendig für die Integration der Seiten zwischen verschiedenen Wiki-Softwares.

## **2.2 Verwendete Datenstrukturen**

In diesem Abschnitt werden die für die Arbeit wichtigen Datenstrukturen vorgestellt.

### **2.2.1 Wiki Interchange Format**

Die Idee eines Austauschformats zwecks Austausch der Wiki-Inhalte zwischen Wikis, die auf verschiedenen Wiki-Softwares laufen, gab es schon früher. So wurde in diesem Papier [1] das Wiki Interchange Format(WIF) umrissen. Die Grundidee war ein Format zu entwerfen, mit dem man Wiki-Inhalte unabhängig der Wiki-Software speichern kann. Dieses Format sollte folgende Anforderungen erfüllen: Die Wiki-Inhalte sollten nach WIF exportierbar und von WIF importierbar sein. Das Format sollte einfach zu implementieren sein. Das Format sollte von gängigen Browsern gelesen werden können und somit die im WIF gespeicherten Wiki-Inhalte von den Benutzern im Browsern betrachtet werden können. Die generischen Inhalte eines Wikis sollten in die statischen Inhalte umgewandelt werden. Die Struktur der Seite sollte kompakt sein. Das Format sollte nicht nur einfach zu generieren, sondern auch einfach zu handhaben sein.

### **2.2.2 RDF**

RDF ist eine Sprache zur Darstellung von Metainformation. Das RDF-Modell besteht aus Tripel. Ein Tripel setzt sich jeweils aus Subjekt, Prädikat und Objekt zusammen. Das Subjekt ist eine Ressource, die beschrieben wird. Das Subjekt hat eine eindeutige Bezeichnung. Das Prädikat beschreibt die

Beziehung zwischen dem Subjekt und dem Objekt. Das Objekt beschreibt den Wert des Prädikats.

RDFa schliesst RDF in XHTML ein, ohne inhaltliche Redundanz sofern die Daten Metadaten sind. Dabei werden XHTML-kompatible Konstrukte zur Spezialisierung des RDF-Inhaltes verwendet. RDF bzw. XML selbst wird nicht in XHTML eingebettet.

### 3 Entwurf

In diesem Kapitel wird der Entwurf des Systems vorgestellt, welches die generische Integration der Wiki-Inhalte ermöglicht. Diese Entwurf ist unabhängig von der Wiki-Software.

#### 3.1 Integrationsmöglichkeiten

Als erstes wird entschieden, ob das zu entwickelnde System beim Benutzer oder in eine Wiki-Software integriert werden soll (siehe Abbildung 6). Möchte man das System in eine Wiki-Software integrieren, so muss man, falls es sich nicht um sein eigenes Wiki handelt, den Eigentümer überzeugen, das System bei sich zu installieren. Da das System unter anderem auch das Kopieren eines Wikis ermöglichen soll, hat der Eigentümer eigentlich keinen guten Grund die Entwicklung eines solchen Systems zu unterstützen. Deshalb wurde entschieden, das System beim Benutzer zu integrieren. Wenn man das System braucht, soll man es bei sich auf dem Rechner installieren und verwenden. Eine Installation auf einem Wiki-Server wird nicht erforderlich sein.

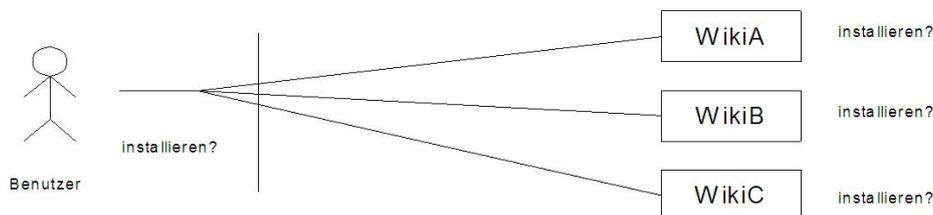


Abbildung 6: Integrationsfrage

#### 3.2 Modellierung der zu integrierenden Daten

Nachdem man die Frage geklärt hat, wo das System laufen soll, muss man entscheiden, welche zu einer Seite gehörenden Inhalte man aus dem Wiki herauslesen möchte. Zweifellos ist der Text der wichtigste Bestandteil einer Seite. Einige Wiki-Seiten enthalten Bilder, die den Text veranschaulichen. Diese Bilder sollten bei einem Backup oder bei einem Kopiervorgang in ein anderes Wiki nicht fehlen (siehe Abbildung 7). Auch gibt es die Möglichkeit, dass eine Seite angehängte Dateien hat, zum Beispiel eine Audiodatei bei einem Artikel aus der Kategorie Musik oder eine Pdf-Datei mit einer Präsentation

zu diesem Thema. Diese sind für viele Benutzer auch von Interesse. Ausserdem hat eine Seite einige interessante Metadaten, die eventuell gar nicht auf der Seite selbst erscheinen, die man aber gerne auch haben möchte. Diese Metadaten sind zum Beispiel der Name des Authors der Seite, der letzte Änderungszeitpunkt der Seite oder die Namen der Unterseiten.

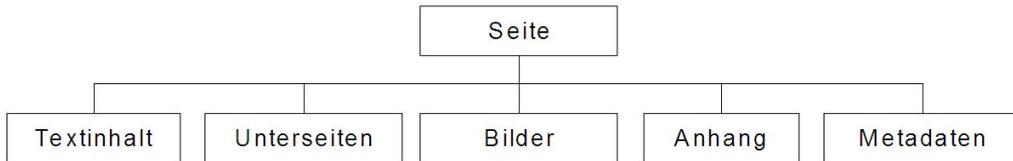


Abbildung 7: Was ist eine Seite

### 3.3 Architektur

Die meisten Wikis stellen den Benutzern ihren Textinhalt in HTML zur Verfügung. Die Bilder und die Links zu den angehängten Dateien werden in HTML eingebettet. Die Wiki-Seiten wurden in einer Wiki-Syntax erstellt. Die Wiki-Syntax ist von Wiki-Software zu Wiki-Software verschieden. Es wurden zwar einige Versuche zur Standardisierung der Wiki-Syntaxen unternommen, wie zum Beispiel von Althelm [5], doch gab es für sie bei den Betreibern der Wiki-Softwares nur wenig Zustimmung. Die HTML-Seiten werden aus den in Wiki-Syntax vorliegenden Informationen vorgeneriert oder bei der Benutzerabfrage erzeugt (siehe Abbildung 8). Die Benutzer greifen mit Hilfe eines Browsers über das HTTP auf die HTML-Seiten ein.

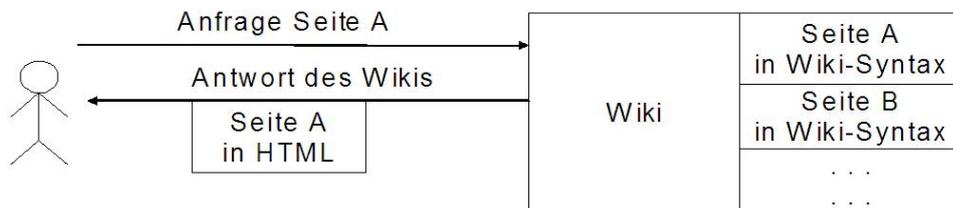


Abbildung 8: Anfrage einer Seite

Die Grundidee ist für verschiedene Wiki-Softwares jeweils einen Adapter zu bauen (siehe Abbildung 9 und 10). Das System besteht aus den Adaptern

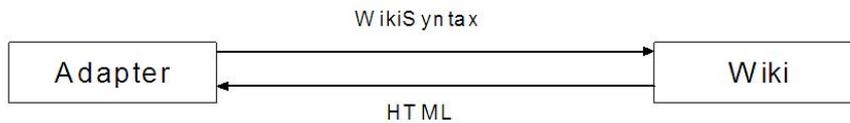


Abbildung 9: Kommunikation zwischen Adapter und Wiki

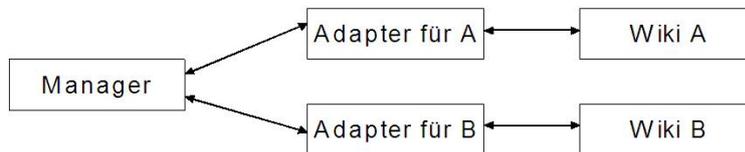


Abbildung 10: Architektur

und einem Manager. Der Aufgabe des Managers ist die Initialisierung und Verwaltung der Adapter. Ein Adapter ist das Bindeglied zwischen dem Benutzer und einer Wiki-Software. Der Adapter besteht aus einem Behälter und den dazugehörigen Seiten. Eine Seite ist die Modellierung einer bestimmten Wiki-Seite, einer Unterseite, eines Bildes, einer angehängten Datei oder der Metadaten. Ein Behälter stellt die benötigten Funktionalitäten zur Kommunikation mit einem bestimmten Wiki bereit.

Um die Anwendungsfälle erfolgreich abarbeiten zu können, weist das Softwaresystem folgende Funktionalitäten auf.

**Anwendungsfall 1, Backup eines Wikis im lokalen Dateisystem:** Das System baut eine Verbindung zur Wiki-Instanz auf. Die Liste aller zu diesem Zeitpunkt im Wiki vorhandenen Seiten wird abgefragt. Die Seiteninhalte werden geholt. Es wird ein Adapter erstellt, der für die Kommunikation mit dem lokalen Dateisystem verantwortlich ist. Über diesen Dateisystem-Adapter wird das Backup im lokalen Speicher abgelegt. Wenn das Backup zur Wiki-Instanz übermittelt werden soll, wird über es mit Hilfe des Dateisystem-Adapters geladen und an den Adapter für die jeweilige

Wiki-Instanz übergeben. Dieser baut eine Verbindung auf und übermittelt das Backup an die Wiki-Instanz.

**Anwendungsfall 2, Kopieren eines Wikis in ein anderes:** Das System baut eine Verbindung zum Wiki A auf. Die Liste aller zu diesem Zeitpunkt im Wiki A vorhandenen Seiten wird abgefragt. Die Seiten werden geholt und in die Syntax der Wiki-Software, auf der das Wiki B läuft, transformiert. Anschliessend wird eine Verbindung zum Wiki B aufgebaut und die Seiten übermittelt.

**Anwendungsfall 3, Kopieren der Seiten aus Wiki A nach Wiki B, die in Wiki B fehlen:** Ähnlich dem Fall 2. Zusätzlich werden noch die Namen aller Seiten aus dem Wiki B abgefragt. Die Listen der Seiten aus den beiden Wikis werden durchgegangen. Nur die in Wiki B fehlenden Seiten werden zum Wiki B übermittelt.

**Anwendungsfall 4, Kopieren der Seiten aus Wiki A nach Wiki B, die seit einem bestimmten Zeitpunkt verändert wurden:** Es wird eine Verbindung zum Wiki A aufgebaut. Das System fragt die Liste der seit einem bestimmten Zeitpunkt veränderten Seiten ab. Diese Seiten werden aus dem Wiki A geholt, bei Bedarf in die Wiki-Syntax von Wiki-Software auf der Wiki B läuft transformiert und zum Wiki B übermittelt, wo sie schon bestehende Seiten überschreiben oder neue anlegen, falls diese noch nicht existieren.

**Anwendungsfall 5, Kopieren der Seiten aus Wiki A nach Wiki B, die bei einer Suchanfrage im Wiki A im Ergebnis erscheinen:** Es wird eine Verbindung zum Wiki A aufgebaut. Die Suche mit einem oder mehreren Suchbegriffen wird initialisiert. Die Seiten, die im Ergebnis enthalten sind, werden aus dem Wiki A geholt. Bei Bedarf werden sie in die Wiki-Syntax von Wiki-Software auf der Wiki B läuft transformiert. Anschliessend werden die Seiten in das Wiki B geschrieben.

In allen Anwendungsfällen wird der Aufbau einer Verbindung zu einer Wiki-Instanz benötigt. Über diese Verbindung werden die Seiten aus einem Wiki geholt und zum Wiki übermittelt. Die jeweils von Fall zu Fall unterschiedliche Funktionalitäten sind: die Seiten einer Wiki-Instanz in eine andere Wiki-Syntax zu transformieren, eine Liste eine Liste aller in einem Wiki existierenden Seiten holen, eine Suche im Wiki durchführen mit einer Liste der

Seite als Ergebnis, die Seiten lokal im Speicher ablegen und sie wieder laden können soll.

Durch diesen Adapter-Entwurf wird das Software-System für beliebig viele Wiki-Softwares gut erweiterbar.

## **3.4 Datenstrukturen**

### **3.4.1 Zugriff der Adapter auf die Seiten eines Wikis**

Der Textinhalt der Seiten liegt bei den meisten Wiki-Softwares in der entsprechenden Wiki-Syntax vor. Die Benutzer können auf diese Daten zugreifen, wenn sie zum Ändern freigegeben sind. Die meisten Wikis haben eine HTML-Ausgabe, auf welche die Benutzer mittels eines Browsers zugreifen. Es gibt zwei Möglichkeiten für das Softwaresystem die Seiten aus einem Wiki zu holen. Die erste ist, man holt sie in Wiki-Syntax über die Änderungsschnittstelle ('edit this page'). Auf der Idee, die Änderungsschnittstelle zu benutzen, basiert zum Beispiel WikiGateway [4]. Dies hat den Vorteil, dass im Falle eines Backups (Anwendungsfall 1) die Seiten in der Wiki-Syntax lokal gespeichert werden und beim Rückspielen des Backups zum Wiki einfach ohne Transformationen übermittelt werden können. Die zweite Möglichkeit ist, man verwendet die HTML-Schnittstelle der Wiki-Instanz. Somit bekommt man die Seiten in HTML. Es gibt viele Werkzeuge mit denen man HTML verarbeiten kann.

Es ist schwieriger die Seiten aus einer Wiki-Syntax in eine andere Wiki-Syntax zu transformieren. Wenn die Seiten in HTML sind, geht die Transformation in eine Wiki-Syntax einfacher von statten. Eine Transformation wird zum Beispiel beim Kopieren eines Wikis in ein anderes benötigt (Anwendungsfall 2).

Ausserdem kann man das Backup aus dem Anwendungsfall 1 in einem Browser betrachten, wenn man das Backup in HTML ablegt.

Somit überwiegen die Gründe die HTML-Schnittstelle der Wiki-Instanz zu verwenden, um die Seiten zu holen. Das Softwaresystem wird auf die HTML-Schnittstelle zugreifen.

### **3.4.2 Vorteile eines Zwischenformats**

Nachdem die Entscheidung für den Zugriff über die HTML-Schnittstelle auf die Seiten der Wikis gefallen ist, werden die Gründe für die Verwendung eines Zwischenformats dargelegt. Die verschiedenen Wiki-Softwares generieren aus ihren Syntaxen etwas anderen HTML-Code zur Darstellung der Seiten. So können die gleichen Attribute von gleichen HTML-Elemente von den

verschiedenen Wiki-Softwares verschiedene Werte gesetzt bekommen. So bekommt zum Beispiel das Element `<div>`, welches den eigentlichen Textinhalt einer Wiki-Seite enthält, von der Mediawiki-Software das Attribut 'id' auf den Wert 'content' gesetzt, von der JspWiki-Software aber auf den Wert 'pagecontent'. Deshalb braucht man bei  $n$  verschiedenen Wiki-Softwares  $n^2$  verschiedene Transformationsregeln, da man für jede HTML-Ausgabe der verschiedenen Wiki-Softwares nach jede beliebige Wiki-Syntax eine neue Regel braucht (siehe Abbildung 11a). Verwendet man aber ein Zwischenformat, so reduziert sich die Anzahl der Regeln auf  $2n$  bei  $n$  verschiedenen Wiki-Softwares: Eine Regel von HTML-Ausgabe einer Wiki-Software nach Zwischenformat und eine für Zwischenformat nach Wiki-Syntax für jede Wiki-Software (siehe Abbildung 11b).

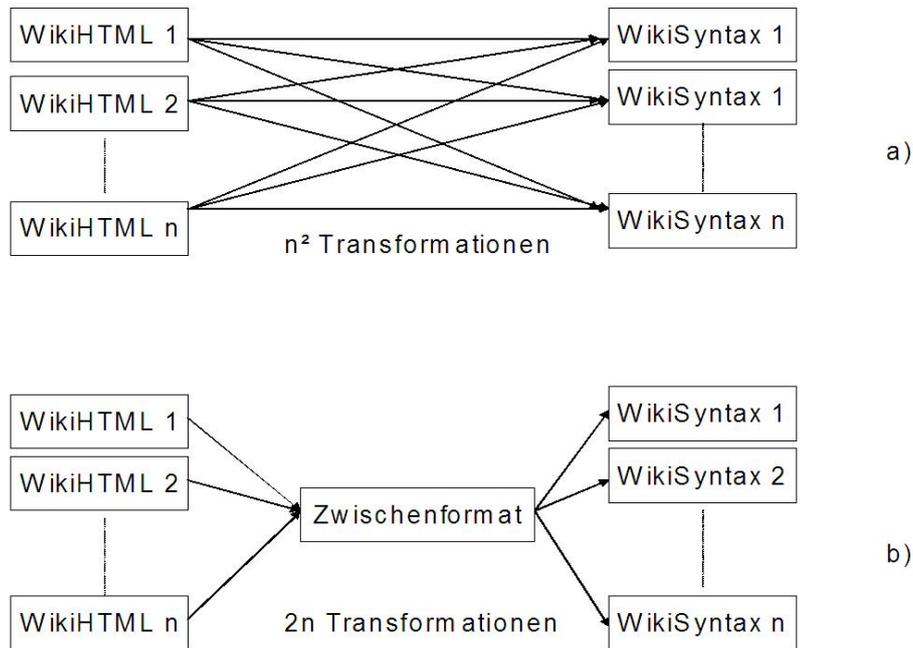


Abbildung 11: Vergleich der Anzahl der Transformationen

### 3.4.3 Zwischenformat

Nun stellt sich die Frage, welches Zwischenformat man für das Software-System wählen sollte. Wenn das System auf die Wiki-Seiten zugreift, liegen die Seiten im 'unsauberen' HTML vors. Um die Verarbeitung der Seiten mit



Abbildung 12: Von HTML nach WIF

einem Werkzeug zu erleichtern, sollte man daraus ein 'sauberes' HTML generieren. Dieses kann man mit Hilfe eines freien Werkzeugs, wie es viele gibt, problemlos nach XHTML konvertieren. Das XHTML macht durch die XML-Eigenschaften die Transformation in eine beliebige Wiki-Syntax einfacher.

Es wurde entschieden, auf der Idee des Wiki Interchange Formats ein Zwischenformat zu entwerfen. Das Zwischenformat wurde WIF genannt und ist eine Untermenge von XHTML. Es wird auf die HTML-Schnittstelle der Wiki-Software zugegriffen. Die geholte Seite in HTML wird bereinigt und nach XHTML transformiert. Dieses XHTML wird durch das Entfernen für den Inhalt nicht relevanter Elemente zum gültigen WIF (siehe Abbildung 12). Wie man XML aus HTML extrahieren kann, wurde zum Beispiel in dieser Arbeit [3] gezeigt.

**Ausgewählte Elemente:** HTML hat viele verschiedene Elemente. WIF ist nur eine Untermenge von XHTML, um die Verarbeitung von WIF-Dateien zu erleichtern. Auf diese Weise braucht man bei der Transformation für weniger Elemente Regeln erstellen.

Es wurde eine Auswahl getroffen, welche Elemente in WIF übernommen werden. Das entscheidende Kriterium ist, dass der Inhalt der Wiki-Seite durch diese Auswahl gar nicht oder so gut wie nicht verändert wird. So haben die Mitglieder von WikiCreole untersucht, welche HTML-Elemente in Wikis am häufigsten verwendet werden. Diese Elemente wurden auch von uns als wichtig eingestuft.

Die in WIF aus XHTML übernommenen Elemente sind:

- Dokumentkontext: <html>, <head>, <title>, <body>
- Überschriften: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- Blockelemente: <p>, <pre>, <hr>
- Listen: <dl>, <dd>, <dt>, <ol>, <ul>, <li>
- Tabellen: <table>, <tr>, <th>, <td>

- Inline-Elemente: `<i>`, `<b>`, `<sub>`, `<sup>`, `<tt>`
- Besondere Inline-Elemente: `<img>` mit dem Attribut `'src'`, `<a>` mit dem Attribut `'href'`.

Beim Element `<a>` wurde beschlossen ein Attribut `'wif'` hinzuzufügen. Dieses Attribut kann auf den Wert `'external'` oder `'internal'` gesetzt werden. `'internal'` wird gesetzt, falls es sich um einen Link innerhalb des Wikis handelt. Also z.B. zu einer anderen Seite in dem selben Wiki. `'external'` wird gesetzt, falls der Link ausserhalb des Wikis zeigt. Z.B. zu einer Seite eines anderen Wikis. Die Auswahl dieser Elemente ermöglicht es, den Inhalt eines Wiki darstellungskonform zu kopieren.

### **Eigenschaften und Vorteile von WIF**

- WIF kann mit geringem Aufwand erweitert werden. Da WIF gültiges XML ist, kann man ohne Komplikationen weitere Elemente hinzufügen und akzeptieren.
- WIF ermöglicht einfache Transformationen durchzuführen. Mit Hilfe von zum Beispiel XSLT kann mit der Formulierung einiger Transformationsregeln der Inhalt der WIF-Datei in die erforderliche Wiki-Syntax transformiert werden.
- WIF kann in einem Browser dargestellt werden, da XHTML von den Browsern gelesen werden kann. Dies ist sehr nützlich, wenn man sich zum Beispiel die lokal gespeicherten Backups eines Wiki ansehen möchte.
- Schliesslich erlaubt WIF die Anwendung eines XML-Parsers, um den Inhalt zu verarbeiten. Sei es um vielleicht bestimmte Inhalte herauszufiltern oder um irgendwelche Statistiken über Die Seiten eines Wikis zu erstellen.

#### **3.4.4 Metadaten und RDFa**

Die Übermittlung der Metadaten unterscheidet sich von Wiki zu Wiki und von der Art der Metadaten. So lassen sich beim Wiki A zum Beispiel sowohl der Zeitpunkt als auch der Author der letzten Änderung über die HTML-Daten, die bei der Seitenabfrage kommen, herausfinden, beim Wiki B kann dagegen nur der Zeitpunkt gefunden werden. Also muss man diese Daten auf eine andere Weise bekommen. Dazu verwendet man die dafür vorgesehene Schnittstelle der jeweiligen Wiki-Software.

Die Metadaten werden in RDF durch ein RDF-Schema definiert. Ein RDF-Tripel setzt sich wie folgt zusammen: Das Subjekt ist der Seitenname. Das Prädikat drückt die Beziehung des Subjekts zum Objekt aus. Das Objekt ist der konkrete Wert eines Attributs.

Als Prädikate gibt es

- `hasUserPage`: der Name der Seite des Authors dieser Seite
- `hasMimeType`: der MimeTyp der Seite
- `hasName`: Name der Seite
- `hasSize`: Größe der Seite
- `hasPreviousVersion`: Version der Seite, die vor der letzten Änderung aktuell war
- `hasAuthor`: der Name des Authors, der diese Seite zuletzt bearbeitet hat
- `hasChangeDate`: die Zeitangabe der letzten an dieser Seite vorgenommenen Änderung
- `hasSubpage`: der Name einer Unterseite von dieser Seite, kann auch eine angehängte Datei sein.

**Es ergibt sich folgendes RDF-Schema (Abbildung 13):**

**Klassen:**

**Klasse WikiUser**

URI: *http://wif.ontoware.org/2005/04WikiUser*

Kommentar: eine Person dem das Element gehört.

**Klasse Page**

URI: *http://wif.ontoware.org/2005/04Page*

Kommentar: das Grundelement von Wikis, modelliert Wiki-Seiten, Bilder und angehängte Dateien.

## Eigenschaften:

### **Eigenschaft hasUserPage**

URI: *http://wif.ontoware.org/2005/04hasUserPage*

Kommentar: Modelliert die Benutzerseiten in einem Wiki. Gewöhnlich hat jeder registrierte Benutzer in einem Wiki eine eigene Seite.

Bereich: WikiUser

Wert: Page

### **Eigenschaft hasMimeType**

URI: *http://wif.ontoware.org/2005/04hasMimeType*

Kommentar: Weist einer 'Page' einen MIME-Typ zu. Das erlaubt die Modellierung von Wiki-Seiten, Bildern und angehängten Dateien mit der gleichen Klasse. Aus dem MIME-Typ erkennt die Anwendung wie sie mit dem Inhalt verfahren soll. Das gleiche Vorgehen benutzt das World Wide Web.

Bereich: Page

### **Eigenschaft hasName**

URI: *http://wif.ontoware.org/2005/04hasName*

Kommentar: Weist einen einzigartigen Namen einer Seite oder einem Benutzer zu.

Bereich: Page, WikiUser

### **Eigenschaft hasSize**

URI: *http://wif.ontoware.org/2005/04hasSize*

Label: size

Kommentar: Bei Bildern die Grösse der Datei

Bereich: Page

### **Eigenschaft hasAuthor**

URI: *http://wif.ontoware.org/2005/04hasAuthor*

Label: author

Kommentar: Jede Seite hat einen Author, der ein Benutzer, 'WikiUser', ist.

Bereich: Page

Wert: WikiUser

### **Eigenschaft hasPreviousVersion**

URI: *http://wif.ontoware.org/2005/04hasPreviousVersion*

Label: previousversion

Kommentar: Minimale Versinierung weist jeder Seite ihre Vorgängerseite zu. Transitive Verwendung dieser Eigenschaft erlaubt es, die ganze Versionierung zu rekonstruieren.

Bereich: Page

Wert: Page

**Eigenschaft hasSubpage**

URI: *http://wif.ontoware.org/2005/04hasSubpage*

Label: subpages

Kommentar: Verlinkt eine Seite mit ihren Bildern, angehängten Dateien und Unterseiten

Bereich: Page

Wert: Page

**Eigenschaft hasChangeDate**

URI: *http://wif.ontoware.org/2005/04hasChangeDate*

Label: changedate

Kommentar: Jede Änderung impliziert eine neue Version der Seite. Diese Eigenschaft modelliert das Erstellungsdatum dieser Version.

Bereich: Page

Wert: xsd:date

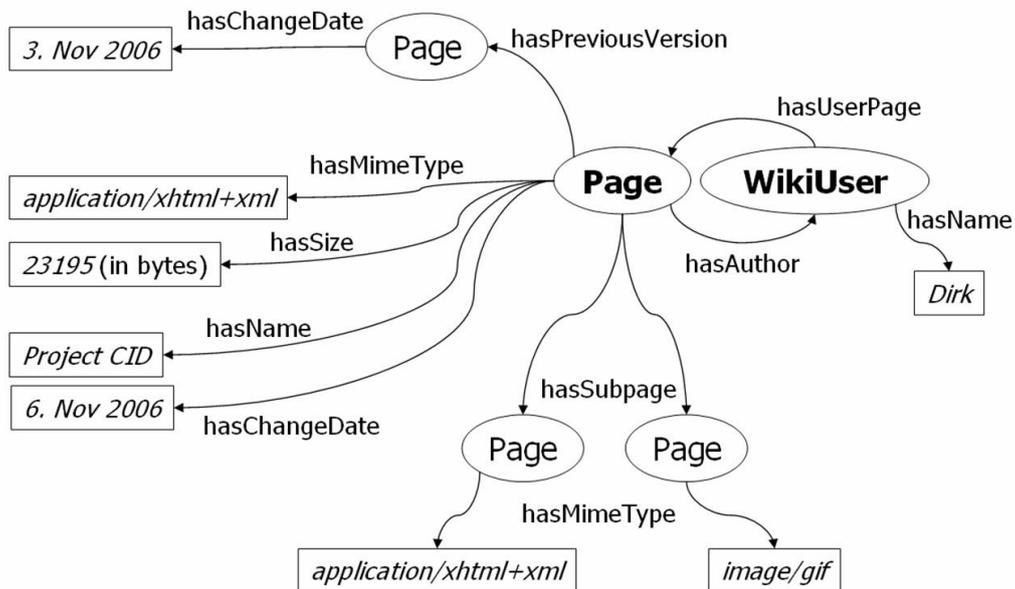


Abbildung 13: Seite in RDFa

**Beispiel (siehe Abbildung 14):** Seite A hat zwei Unterseiten Seite B und Seite C, wurde zuletzt geändert von User D zum Zeitpunkt E. Es gibt noch eine angehängte pdf-Datei F. Die Version vor der letzten Änderung dieser Seite ist Seite A0. Dann würde das wie folgt in Rdf aussehen:

A: hasName A.  
A: hasMimeType wif.  
A: hasAuthor D.  
A: hasChangeDate E.  
A: hasSubpage B.  
A: hasSubpage C.  
A: hasSubpage F.  
A: hasPreviousVersion A0.  
B: hasName B.  
B: hasMimeType wif.  
C: hasName C.  
C: hasMimeType wif.  
A0: hasName A0.  
A0: hasMimeType wif.  
F: hasName F.  
F: hasMimeType pdf-file.

### 3.4.5 Transformation

Die Wiki-Seiten bekommt man in HTML von der Wiki-Software. Diese werden in gültiges XHTML umgewandelt. Im nächsten Schritt erfolgt die Bereinigung des Dokuments, damit die Daten im WIF vorliegen. Alle Elemente, die für das WIF als XHTML-Teilmenge im Kapitel 3.4.3 ausgewählt wurden, werden auch unterstützt. Die anderen Elemente werden aus dem Dokument entfernt.

So entsteht der Inhalt der ursprünglichen Wiki-Seiten im WIF. Dieser kann nun entweder im lokalen Dateisystem mit Hilfe des Dateisystems-Adapters abgelegt werden oder muss gleich in die entsprechende Wiki-Syntax transformiert werden, damit man ihn an das Wiki übermitteln kann (siehe Abbildung 15).

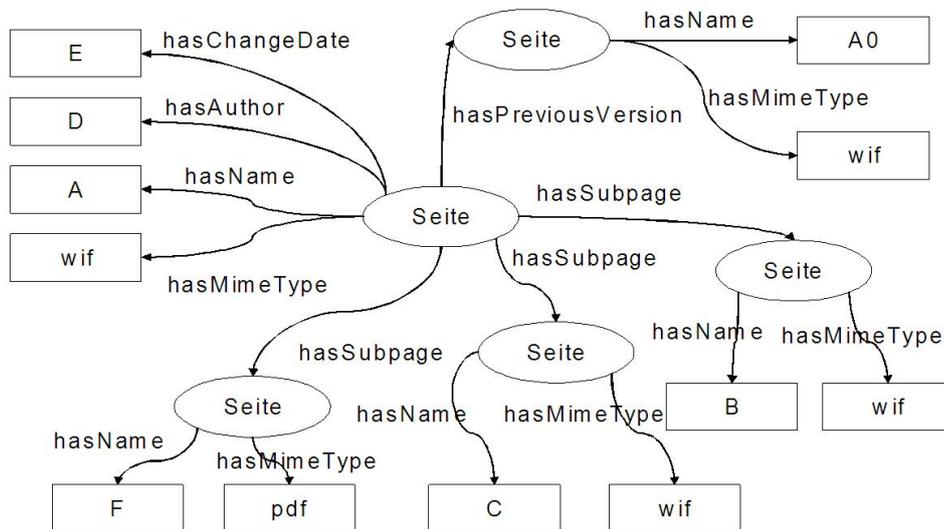


Abbildung 14: Darstellung des Beispiels

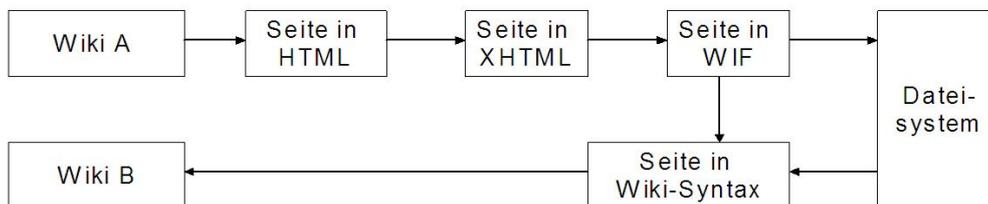


Abbildung 15: Transformationen

## 4 Implementierung

Das System wurde in Java implementiert. Das bietet zum einen die Plattformunabhängigkeit und zum anderen die Objektorientierung, mit der man gut die Architektur verwirklichen kann. Im Rahmen der Studienarbeit wurden drei Adapter erstellt. Zwei davon für bestimmte Wiki-Softwares, nämlich für die MediaWiki und für die JspWiki, und einer für das Dateisystem, da dieser bei den Anwendungsfällen 1 (Datensicherung) und 3 (Kopieren eines bestehenden Wikis in ein anderes bestehendes Wiki) gebraucht wurde. Ausserdem hat sich der Dateisystem-Adapter gut im Testen bewiesen, da man den Inhalt, der damit lokal abgelegt wurde, vor dem Hochladen zum Wiki noch manuell verändern konnte.

### 4.1 Architektur

Ein Adapter besteht aus einer Repository-Klasse und einer WifPage-Klasse. Mit der Repository-Klasse wird eine Wiki-Instanz modelliert. Mit der WifPage-Klasse wird eine Wiki-Seite modelliert. Jede Repository-Klasse und WifPage-Klasse erbt von dem entsprechenden Interface, IWifPageRepository oder IWifPage. Abbildung 16 zeigt die abstrakten Interfaces und die davon abgeleiteten implementierten Repository- und WifPage-Klassen in UML.

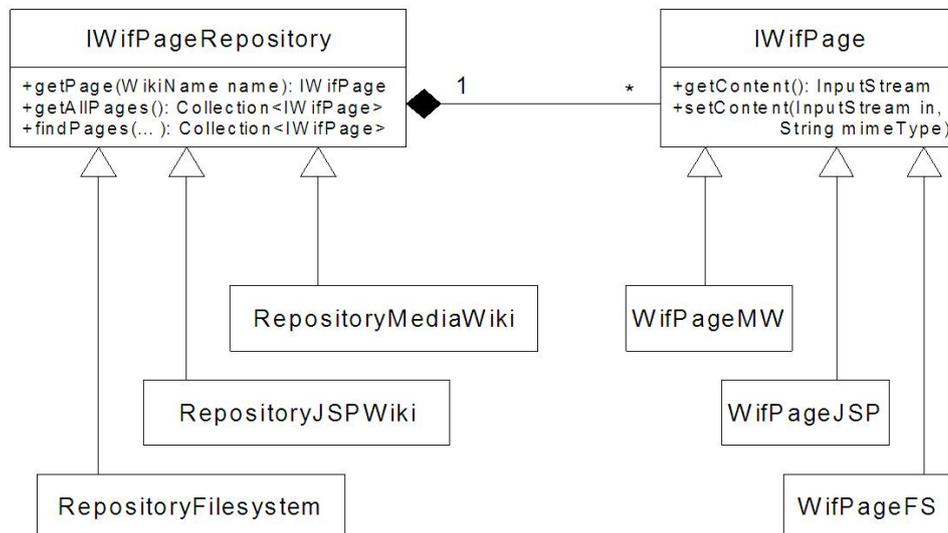


Abbildung 16: UML-Klassenmodell der Architektur

### 4.1.1 Repository

Die wichtigsten Methoden der Repository-Klasse sind:

`IWifPage getPage(WikiName wikiname)`: Damit bekommt man eine Objektinstanz einer Seite, mit der man dann weiterarbeiten kann.

`Collection<IWifPage> getAllWifPages()`: Hier werden alle Seiten einer Wiki-Instanz als IWifPage-Objekte zurückgegeben.

`Collection<IWifPage> findPages(Map<String, String> query, Date changedAfter, Date changedBefore)`: In dieser Methode wurde die Suche im Wiki realisiert. Sollten die String-Schlüssel und die String-Werte im Map gleich sein, so wird es erkannt, dass es um ein Suchbegriff handelt. Die Suchbegriffe werden mit einer Suchanfrage an die Wiki-Instanz gegeben. Als Ergebnis gibt die Methode eine Menge der Seiten zurück, die bei der Suchanfrage von der Wiki-Instanz ausgegeben werden. Die Seitensuche wurde so realisiert, dass eine gültige URL aus den Suchbegriffen generiert wird. Durch den Zugriff darauf bekommt man die Ergebnisliste der Suche. Diese Ergebnisse sollte mit den Metadaten-Parametern gefiltert werden, wonach dann die endgültige Ergebnisliste zurückgegeben wird.

### 4.1.2 WifPage

Die wichtigsten Methoden der WifPage-Klasse sind:

`InputStream getContent()`: Der Inhalt der Wiki-Seite, welche auf diesem Objekt abgebildet ist, wird durch diese Methode als ein eingehender Stream zurückgegeben.

`void setContent(InputStream in, String mimeType)`: Mit dieser Methode wird eine Wiki-Seite in die Wiki-Instanz geschrieben. Mit dem Parameter 'InputStream in' wird der zu schreibende Inhalt der Methode übergeben. Beim hochladen einer Seite zum Wiki ist besonders darauf zu achten, dass alle erforderlichen Daten mit übermittelt werden. So braucht die MediaWiki-Software diese Parameter bei der Post-Methode: wpSection, wpStarttime, wpEdittime, wpScrolltop, wpTextbox1, wpSummary, wpSave, wpAutoSummary. Die JspWiki-Software hingegen diese: page, action, edittime, \_editedtext, changenote, ok. Ausserdem muss man sowohl bei der MediaWiki-Software als auch bei der JspWiki-Software einen gültigen Edit-Zeitwert übermitteln, damit die Übermittelten Daten angenommen werden und in der Seite abgelegt werden. Dieser Edit-Zeitwert wird in der Implementierung unmittelbar vor der Übermittlung abgefragt und als Parameter richtig gesetzt. Noch eine Besonderheit besitzt die MediaWiki-Software. Für die Übermittlung der Parameter sollte man keine einfache HTTP-Post-Methode verwenden, sondern eine Multipart-Post-Methode, sonst bleibt der Hochladevorgang erfolglos.

## 4.2 Transformation

Die Transformationen auf XHTML- und WIF-Dateien erfolgen mit XSLT. WIF-Dateien können mit XSLT transformiert werden, da die WIF-Dateien gültige XML-Dokumente sind. Für die Verwendung von XSLT wurde entschieden, weil XSLT ein wichtiger W3C-Standard für Transformation ist. XSLT läuft in allen Sprachen. Deswegen ist sie auf vielen verschiedenen Systemen gut zu implementieren.

Für jede Wiki-Software gibt es zwei XSLT-Dateien.

Die erste Datei ist für das Erzeugen von WIF-Dateien. Dies geschieht nachdem man die entsprechende Wiki-Seite aus HTML in XHTML überführt. Das Erzeugen von gültigem WIF daraus ist als Bereinigung der Seite in XHTML zu verstehen.

Die zweite XSLT-Datei transformiert WIF-Dokumente in die Wiki-Syntax. So wird die Seite zur Übermittlung an das Wiki bereitgestellt.

Zu beachten ist, dass die Links ein neues Attribut bekommen. Dieses muss bei der Transformation entstehen. Bei der Rücktransformation in die entsprechende Wiki-Syntax wird dieses neu Attribut zur Hilfe genommen, um festzustellen, ob ein interner oder externer Link generiert werden soll.

## 5 Evaluation

In diesem Kapitel werden die erreichten Ergebnisse präsentiert. Diese werden anhand der Anwendungsfälle aufgeteilt.

Zunächst müssen einigen Einschränkungen erwähnt werden, mit denen das System im Moment funktioniert. Das System kann momentan nur den Textinhalt der Seiten verarbeiten. Die Bilder und angehängte Dateien werden noch nicht berücksichtigt. Auch wurde der RDFa-Teil des Systems noch nicht implementiert.

**Anwendungsfall 1, Datensicherung:** Es ist möglich den Textinhalt einer Wiki-Seite aus einem MediaWiki oder aus einem JspWiki zu lesen und ihn im lokalen Speicher als WIF-Datei abzulegen. Diese WIF-Dateien kann das System nach MediaWiki-Syntax oder JspWiki-Syntax transformieren. Nach der Transformierung kann man eine Seite zu einem MediaWiki oder JspWiki übertragen.

**Anwendungsfall 2, Kopieren eines Wikis:** Die Methoden zur Durchführung dieses Anwendungsfalles wurden implementiert. Es ist sowohl für MediaWikis als auch für JspWikis möglich, eine Liste aller im Wiki existierenden Artikel abzufragen. Für jeden Eintrag in dieser Liste kann dann der Textinhalt der Wiki-Seite geholt werden. Dieser liegt im WIF-Format vor. Der Inhalt kann in eine MediaWiki oder in eine JspWiki-Syntax transformiert werden und in das entsprechende Wiki hochgeladen werden.

**Anwendungsfall 3, Kopieren eines Wikis, existierende Seiten sollen im Speicher abgelegt werden:** Die Durchführung dieses Anwendungsfalles ist auch möglich. Hierzu sollte der Anwendungsfall 2 wie folgt ergänzt werden. Es wird eine Liste aller Seiten, die im Zielwiki existieren, geholt. Sollte eine WikiSeite dort vorkommen, wird diese Seite über den Dateisystem-Adapter im lokalen Speicher als WIF-Datei abgelegt. Sollte eine WikiSeite dort nicht vorkommen, wird sie in die Syntax des Zielwikis transformiert und in das entsprechende Wiki hochgeladen.

**Anwendungsfall 4, Synchronisation eines Wikis mit einem anderen:** Die Methoden zur Transformation und Integration der Textinhalte einer Wiki-Seite sind implementiert. Da der RDFa-Teil des Systems noch fehlt, kann nicht auf die Metadaten zugegriffen werden. Somit wird die Synchronisation zwischen zwei Wikis erst von dem System ermöglicht, wenn der RDFa-Teil implementiert wird.

**Anwendungsfall 5, Kopieren mehrerer Wiki-Seiten, die bei einer Wikisuchanfrage als Ergebnis kommen, in ein anderes Wiki:** Die Durchführung dieses Anwendungsfalles ist ebenfalls möglich. Der Zugriff auf die Suche der MediaWikis und der JspWikis ist implementiert. Die Suche funktioniert sowohl mit einem als auch mit mehreren Begriffen. Die Ergebnisliste kann abgearbeitet werden, indem man die in dieser Liste vorkommenden Wiki-Seiten aus dem Wiki holt, diese in den WIF-Format transferiert, dann in die entsprechende Wiki-Syntax und anschliessend in das Zielwiki hochgeladen werden.

## 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Software-System zur Integration der Wiki-Inhalte zwischen verschiedenen Wikis, die auf unterschiedlichen Wiki-Softwares laufen können, vorgestellt. Das System ist gut erweiterbar für beliebig viele Wiki-Softwares.

Zuerst wurden die bedeutendsten Anwendungsfälle eines solchen Systems erkannt und beschrieben. Für jeden Anwendungsfall wurden die Anforderungen an das Software-System ermittelt.

Es wurden bestehende Schnittstellen einiger Wiki-Softwares untersucht und bewertet. Die für die Realisierung des Software-Systems benötigten Datenstrukturen wurden gefunden.

Beim Entwurf des Systems wurden die Integrationsmöglichkeiten des Systems untersucht und die zu integrierenden Daten modelliert. Es wurde eine Architektur für das Software-System entworfen. Um gute Erweiterbarkeit des Systems zu ermöglichen, wurden passende Datenstrukturen, wie die Transformationen mit XSLT, ausgewählt. Ausserdem wurde ein Zwischenformat, das Wiki Interchange Format, konzipiert.

Schliesslich wurde das Software-System auf der Basis des Entwurfs implementiert. Es wurden drei Adapter erstellt, ein MediaWiki-, ein JspWiki- und ein Dateisystemadapter. Für die Transformation der MediaWiki-Syntax und der JspWiki-Syntax nach WIF und zurück wurden XSLT-Transformationsregeln formuliert.

Zusammengefasst lässt sich sagen, dass das vorliegende System ein bereits benutzbares Werkzeug darstellt. Die Textinhalte der Wiki-Seiten können zwischen den Wikis transferiert werden. Die Voraussetzung ist: Die Wikis sollten auf MediaWiki-Software oder JspWiki-Software laufen. Sollten die noch nicht implementierten Funktionen, wie zum Beispiel die Transformation der Bilder, realisiert werden, wird das Software-System ein hilfreiches Werkzeug bei den Tranfers der Wiki-Inhalte zwischen verschiedenen Wikis sein.

## Literatur

- [1] Max Völkel, Eyal Oren 'Towards a Wiki Interchange Format (WIF)' In Proceedings of the First Workshop on Semantic Wikis – From Wiki To Semantics. 2006.
- [2] J. Jalkanen. Davwiki – the next step of wikirpcinterfaces? In Proceedings of Wikimania 2005 - The First International Wikimedia Conference. Wikimedia Foundation, JUL 2005.
- [3] M. Völkel. Extraktion von XML aus HTML-seiten – das WYSIWYG-werkzeug d2c. Diplomarbeit, May 2003.
- [4] B. Shanks. Wikigateway: a library for interoperability and accelerated wiki development. In WikiSym '05: Proceedings of the 2005 international symposium on Wikis, pages 53–66, New York, NY, USA, 2005. ACM Press.
- [5] M. Altheim. Inter wiki markup language (iwml), 03 2004.

# Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den 19. Dezember 2006