

Universität Bielefeld  
Technische Fakultät  
AG Rechnernetze und Verteilte Systeme

Diplomarbeit

# **Integration einer kollaborativen Arbeitsumgebung in die Kommunikationsplattform Worksphere**

Heiko Holtkamp

10. April 2006

Betreuer: Prof. Peter B. Ladkin PhD  
Dipl. Inform. I Made Wiryana



## **Erklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Zitate sind als solche kenntlich gemacht.

Bielefeld, 10. April 2006

Heiko Holtkamp



## Danksagung

An dieser Stelle möchte ich mich bei meinen Betreuern Prof. Peter B. Ladkin PhD und I Made Wiryana bedanken, die es mir ermöglichten zu diesem Thema meine Diplomarbeit zu verfassen und die mich in jeglicher Hinsicht unterstützt haben.

Mein Dank gilt auch den übrigen Mitarbeitern der AG Rechnernetze und Verteilte Systeme, die stets für Fragen ein offenes Ohr hatten und zu wertvollen Diskussionen bereit waren.

Ebenso möchte ich Dr. Norbert Nothbaum und Johanna Lojewski von der Nothbaum GmbH danken. Ohne Norbert wäre die Entwicklung der Worksphere niemals möglich gewesen. Seine Unterstützung und seine Anregungen haben die Worksphere zu dem gemacht, was sie ist. Einen wesentlichen Beitrag hat auch Johanna geleistet, die durch den engen Kontakt zu den Anwendern der Worksphere genau vermitteln konnte, wo es „hakt“ und was verbessert werden muss.

Ein besonderer Dank gilt meinen Eltern, meinem Bruder und Anneke für ihre unermüdliche Unterstützung und Ermunterung. Ein Danke reicht nicht aus für das, was ihr mir gegeben habt!



# **Don't Panic**

Douglas Adams [Adams1979]



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Wikis</b>	<b>3</b>
2.1. Geschichte der Wikis . . . . .	5
2.1.1. Das erste Wiki . . . . .	5
2.1.2. Per Anhalter durch die Galaxis . . . . .	7
2.1.3. Wikipedia . . . . .	8
2.2. Was ist ein Wiki? . . . . .	9
2.3. Wiki-Klone und Wikis . . . . .	13
2.3.1. Wiki-Klone . . . . .	13
2.3.2. Wikis . . . . .	16
2.4. Wiki-Syntax . . . . .	16
2.4.1. Syntax-Standards . . . . .	17
2.4.2. Texte gliedern und formatieren . . . . .	20
2.5. Anwendungsgebiete eines Wiki . . . . .	23
<b>3. Worksphere</b>	<b>27</b>
3.1. Technische Anforderungen . . . . .	29
3.2. Kernfunktionen der Worksphere . . . . .	29
3.2.1. Konfigurationsdateien . . . . .	30
3.2.2. Datenbank-Abstraktion . . . . .	30
3.2.3. Sitzungsverwaltung (Sessions) . . . . .	32
3.2.4. Authentifikation . . . . .	37
3.2.5. Zugriffsrechte (Permissions) . . . . .	39
3.3. Kommunikationsfunktionen der Worksphere . . . . .	41
3.3.1. Foren . . . . .	42
3.3.2. Chats . . . . .	43
3.3.3. Schwarze Bretter . . . . .	45
3.3.4. Dateibereiche . . . . .	45
3.3.5. Benutzerliste . . . . .	46

3.3.6.	Sofortnachrichten . . . . .	48
3.3.7.	Projektgruppen . . . . .	49
3.3.8.	Newsticker . . . . .	50
<b>4.</b>	<b>Implementation</b>	<b>53</b>
4.1.	Anforderungskatalog . . . . .	55
4.2.	Datenbankentwurf . . . . .	56
4.3.	Benutzeroberfläche . . . . .	60
4.4.	Module, Skripte und Verzeichnisse . . . . .	62
4.4.1.	Übersicht der Skripte und Verzeichnisse . . . . .	62
4.4.2.	Administrationsskripte des Wiki-Moduls . . . . .	62
4.4.3.	Aufbau des Wiki-Moduls . . . . .	63
4.4.3.1.	Konfigurationsdatei . . . . .	63
4.4.3.2.	Hilfsfunktionen . . . . .	65
4.4.3.3.	Menüleiste für das Wiki . . . . .	66
4.4.3.4.	Steuerungsskript für das Wiki . . . . .	67
4.4.3.5.	Verwaltung der Wiki-Seiten . . . . .	68
4.4.3.6.	Anzeigen von Wiki-Seiten . . . . .	71
4.4.3.7.	Bearbeiten von Wiki-Seiten . . . . .	76
4.4.3.8.	Wiederherstellen archivierter Seiten . . . . .	80
4.4.3.9.	Versionen von Wiki-Seiten . . . . .	81
4.4.3.10.	Letzte Änderungen im Wiki . . . . .	82
4.4.3.11.	Suche in Wiki-Seiten . . . . .	83
<b>5.</b>	<b>Bedrohungsanalyse</b>	<b>87</b>
5.1.	Bedrohungsanalyse mit Angriffsbäumen . . . . .	88
5.2.	Angriffsziel Wiki . . . . .	90
5.3.	Folgerungen und Empfehlungen . . . . .	97
<b>6.</b>	<b>Zusammenfassung und Ausblick</b>	<b>101</b>
<b>A.</b>	<b>Angriffsbäume</b>	<b>105</b>
<b>B.</b>	<b>Inhalt der CD-ROM</b>	<b>109</b>

# Abbildungsverzeichnis

2.1. Screenshot des WorldWideWeb Browsers . . . . .	4
2.2. Bearbeitung einer Wiki-Seite im Browser . . . . .	10
2.3. Entstehende Vernetzung bei der Bearbeitung von Wiki-Seiten . . . . .	11
2.4. Vergleich einiger Syntax-Elemente verschiedener Wikis. . . . .	19
2.5. Anwendungsgebiete eines Wiki . . . . .	24
3.1. Screenshot einer Plattform auf Basis der Worksphere . . . . .	28
3.2. HTTP-Anfrage/-Antwort. . . . .	33
3.3. Typischer Cookie-Austausch. . . . .	34
3.4. Session-Objekte als Datensammlungen . . . . .	34
3.5. Klassenschema DB_Sql, CT_Sql und Session . . . . .	35
3.6. Beispiel für das Login-Formular in der Worksphere . . . . .	37
3.7. Modell der Zugriffskontrolle der PHPLib/w . . . . .	40
3.8. Screenshot des Administrationstools. . . . .	42
3.9. Screenshot eines Forums in der Worksphere. . . . .	43
3.10. Screenshot eines Chats in der Worksphere . . . . .	44
3.11. Screenshot eines Schwarzen Bretts in der Worksphere. . . . .	45
3.12. Screenshot eines Dateibereichs in der Worksphere. . . . .	46
3.13. Screenshot der Benutzerinformationen in der Worksphere . . . . .	47
3.14. Screenshot einer Sofortnachricht in der Worksphere. . . . .	48
3.15. Screenshot einer Projektgruppe in der Worksphere. . . . .	49
3.16. Screenshot des Newstickers der Worksphere. . . . .	50
4.1. Evolutionäre Entwicklung . . . . .	53
4.2. Entwurf der Wiki-Bedienoberfläche. . . . .	60
4.3. Kontextsensitive Navigationselemente in der Menüleiste . . . . .	61
4.4. Administrationsfunktionen für das Wiki-Modul. . . . .	64
4.5. Übertragung der Parameter in der URL. . . . .	68
4.6. Darstellung eines mit der Wiki-Syntax formatierten Textes. . . . .	73
4.7. Eingabemaske des Wiki-Moduls. . . . .	77

---

4.8. Vorschau auf eine bearbeitete Seite. . . . .	80
4.9. Wiederherstellen einer archivierten Wiki-Seite. . . . .	81
4.10. Anzeige der Versionen einer Wiki-Seite und der Seiten, die auf die Seite verweisen. . . . .	82
4.11. Übersicht der letzten Änderungen im Wiki. . . . .	83
4.12. Erweitertes Suchformular des Wiki-Moduls. . . . .	85
5.1. Grafische und textuelle Repräsentation einer UND-Verknüpfung. . . . .	88
5.2. Grafische und textuelle Repräsentation einer ODER-Verknüpfung. . . . .	89
5.3. Angriffsszenarien eines Beispiel-Angriffsbaums. . . . .	89
5.4. Angriffsbaum zum unberechtigten Lesen von Wiki-Seiten. . . . .	92
5.5. Angriffsbaum zum unberechtigten Bearbeiten von Wiki-Seiten. . . . .	93
5.6. Angriffsbaum zum Herbeiführen eines Systemausfalls. . . . .	94
5.7. Angriffsbaum zum Erlangen fremder Zugangsdaten. . . . .	96
5.8. Angriffsbaum zum Zugriff auf den Datenbankserver. . . . .	96
5.9. Angriffsbaum zum Zugriff auf den Webserver. . . . .	97
A.1. Angriffsbaum zum Angriff auf den Webserver. . . . .	105
A.2. Angriffsbaum zum Angriff auf Kommunikationsstrukturen. . . . .	106
A.3. Angriffsbaum zum Angriff auf den Datenbankserver. . . . .	107

# 1. Einleitung

Den Anstoß zu dieser Arbeit habe ich dem Artikel „Kollaboratives Lernen mit Wiki-WikiWebs“ von Tobias Thelen und Clemens Gruber, der in [Kerres2003] erschienen ist, zu verdanken. Der Artikel war meine erste nähere Begegnung mit dem Konzept der Wikis abseits der Online Enzyklopädie Wikipedia [WikipediaDE2006]. In dem Artikel skizzieren Gruber und Thelen ihre Erfahrungen mit dem „Einsatz eines Wikis in einem standortübergreifenden Seminar“. Ein Wiki wurde in diesem Seminar als zentrales Kommunikationselement eingesetzt. Als Fazit ihrer Erfahrungen mit dem Einsatz eines Wikis in dem Seminar kommen die Autoren zu dem Schluss, dass „durch den Einsatz eines WikiWikiWebs in diesem standortübergreifenden Szenario kollaborative Kommunikations- und Schreibprozesse ermöglicht und gefördert werden, die mit anderen Werkzeugen nur schwer zu erreichen sind.“.

Ein Wiki ist eine Sammlung von Webseiten, die von den Benutzern nicht nur gelesen, sondern auch direkt im Browser über ein Formular, ohne HTML-Kenntnisse, bearbeitet werden können. D.h., jeder kann jeden Text im Wiki ändern. Mit einer, gegenüber HTML-Seiten, stark vereinfachte Link-Syntax können auf einfache Weise Hypertexte in einem Wiki entstehen.

Nach einigen Überlegungen und Gesprächen, vor allem mit Norbert Nothbaum und Mitarbeitern der AG Rechnernetze und Verteilte Systeme an der Technischen Fakultät der Universität Bielefeld, reifte in mir die Idee, ein Wiki als Werkzeug zur kollaborativen Arbeit in die Kommunikationsplattform Worksphere zu integrieren. Die Worksphere wird seit einiger Zeit in verschiedenen Projekten und (Fern-)Studiengängen als Kommunikationsplattform (bzw. CSCW/CSCL-System<sup>1</sup>) eingesetzt. Diese Arbeit setzt die Idee, ein Wiki in die Worksphere als kollaborative Arbeitsumgebung zu integrieren, um. Das Wiki-Modul ist als Ergänzung der vorhandenen Kommunikationsmöglichkeiten wie Chats, Foren, Schwarzen Brettern, Sofortnachrichten (Instant Messages) etc. gedacht.

---

<sup>1</sup>Computer Supported Collaborative/Cooperative Work, Computer Supported Collaborative/Cooperative Learning

Die Arbeit setzt dabei den Schwerpunkt auf eine rein technische Umsetzung und betrachtet nicht, inwieweit sich ein Wiki tatsächlich für die kollaborative Arbeit eignet oder welche Konsequenzen der Einsatz eines Wikis für die Zusammenarbeit hat bzw. welche positiven oder negativen Effekte es mit sich bringt.

## **Gliederung**

Kapitel 2 stellt das Wiki-Konzept vor. Daneben werden die Entwicklungsgeschichte der Wikis, die ihren Ursprung in der Software-Entwicklung als Ideen- und Informationssammlung zu Software-Patterns haben [Cunningham2006d], und verschiedene Wiki-Klone dargestellt. Die Darstellung dient als Basis für die spätere Implementation des Wiki-Moduls.

Kapitel 3 stellt die Kommunikationsplattform Worksphere vor und gibt eine Einführung in die Umgebung, in die das Wiki integriert werden soll.

Die Implementation des Wiki-Moduls wird in Kapitel 4 beschrieben.

In Kapitel 5 wird eine Bedrohungsanalyse für das in der Arbeit implementierte Wiki-Modul durchgeführt. Die Bedrohungsanalyse zeigt mögliche Angriffsszenarien gegen das Wiki-Modul auf. Aus den ermittelten Szenarien werden Empfehlungen zur Absicherung des Wiki-Moduls abgeleitet.

Zum Schluss folgt eine Zusammenfassung der Ergebnisse der Arbeit und ein Ausblick auf mögliche Erweiterungen.

## 2. Wikis

ATTENTION WIKIPEDIA, THERE IS A MAJOR BUG ON YOUR SITE ALLOWING ME TO DO THIS (WRITE ON YOUR ARTICLES). I HAVE DONE NO DAMAGE BUT AM TRYING TO ALERT YOU BEFORE SOMEONE DOES.

(Anonymer Kommentar auf den Seiten der englischen Wikipedia)<sup>1</sup>

In diesem Zitat aus einem Artikel der englischsprachigen Wikipedia [WikipediaEN2006], wird das grundlegende Prinzip eines Wikis – das offene und kollektive Erstellen eines (Hyper)Textes – als Fehler in der Software gedeutet. Das ist eine häufige Reaktion bei der ersten Begegnung mit einer Website auf Basis eines Wikis. Warum sollte jeder die Möglichkeit haben, den Inhalt einer im Web veröffentlichten Seite zu bearbeiten?

Erik Möller schreibt in [Möller2005a] dazu:

*„Wer darüber stolperte (...) las vielleicht noch eine kurze Zusammenfassung des Prinzips – ‚jeder kann alles editieren‘ – und surfte dann kopfschüttelnd weiter.“*

Doch genau dieses Prinzip führt einen Grundgedanken, den Sir Tim Berners-Lee beim Entwurf des World Wide Web (WWW) hatte, wieder zum WWW zurück: Webseiten einfach und unkompliziert lesen *und* bearbeiten zu können [Berners-Lee2000]:

*„The fundamental principle behind the Web was that once someone somewhere made available a document, database, graphic, sound, video, or screen at some stage in an interactive dialogue, it should be accessible (subject to authorization, of course) by anyone, with any type of computer, in any*

---

<sup>1</sup>12. April 2004, <http://en.wikipedia.org/w/wiki.phtml?title=Image:Soapbubble.sky.250px.jpg&oldid=3158755>  
Die Idee zu diesem Zitat stammt aus [Möller2005a].

country.“ (S. 37)

„That meant that anyone (authorized) should be able to publish and correct information, and anyone (authorized) should be able to read it.“ (S. 38)

Die Nutzer sollten nicht nur Leser sein, sondern auch aktive Autoren. Geblieben ist von dieser Idee im Wesentlichen nur, dass Webseiten heruntergeladen bzw. gelesen werden können. Das Web ist mehr und mehr zu einem reinen Lesemedium geworden. Doch der erste von Berner-Lee entwickelte Browser war gleichzeitig auch ein Editor [Berners-Lee].

„The WorldWideWeb browser/editor was working on my machine an Robert's, communicating over the Internet with the info.cern.ch server by Christmas Day 1990.“ ([Berners-Lee2000], S. 30)



Abbildung 2.1.: Screenshot des WorldWideWeb Browsers [Berners-Lee].

Aber mit der Zeit, als auch andere Personen, Firmen und Institutionen begannen Browser für das WWW zu programmieren, verschwand die Editor-Funktion aus den Browsern.

„Although browsers were starting to spread, no one working on them tried to include writing and editing functions.“ ([Berners-Lee2000], S. 57)

Die Präsentation bzw. das Layout des Inhalts rückte immer stärker in den Vordergrund und die Entwickler arbeiteten verstärkt an neuen Funktionen, die die Darstellung betrafen.

*„But it was also true that most were more excited about putting fancy display features into the browser – multimedia, different colors and fonts – which took much less work and created much more buzz among users.“*  
([Berners-Lee2000], S. 71)

Tim Berners-Lee hat diese Entwicklung von Beginn an kritisiert, konnte sie dennoch nicht aufhalten. Er hat immer wieder versucht, die Entwickler dazu zu bringen, Editor-Funktionalität in ihre Browser zu integrieren.

*„(... ) and made my now-standard case for making the Mosaic browser an editor, too. Marc and Eric explained that they had looked at that option and concluded that it was just impossible. It couldn't be done. This was news to me, since I had already done it with the WorldWideWeb on the NeXT.“*  
([Berners-Lee2000], S.70)

Wikis bringen die Grundideen von Tim Berners-Lee, dass Web als „kreatives, flexibles Medium, mit Anmerkungen auf Webseiten, Gruppeneditoren, und so weiter.“<sup>2</sup> wieder zurück. Mit der Änderbarkeit der einzelnen Seiten durch jedermann – oder nahezu jedermann – wird eine ursprüngliche Idee für das World Wide Web in Wikis realisiert.

## 2.1. Geschichte der Wikis

### 2.1.1. Das erste Wiki

Die Geschichte der Wikis beginnt im März 1995 mit einer E-Mail von Ward Cunningham, dem Entwickler des ersten Wikis [Cunningham2006b]:

*„ Steve – I've put up a new database on my web server and I'd like you to take a look. It's a web of people, projects and patterns accessed through a*

---

<sup>2</sup>Tim Berners-Lee in einem Interview mit Mark Frauenfelder: „I have always wanted the Web to be a more creative, flexible medium, with annotation systems and group editors and so on.“ [Frauenfelder2004].

*cgi-bin script. It has a forms based authoring capability that doesn't require familiarity with html. I'd be very pleased if you would get on and at least enter your name in RecentVisitors. I'm asking you because I think you might also add some interesting content. I'm going to advertise this a little more widely in a week or so. The URL is <http://c2.com/cgi-bin/wiki>. Thanks and best regards. – Ward “*

Kurz darauf, im Mai 1995, veröffentlichte Cunningham als bekanntes Mitglied auf der Patterns List<sup>3</sup> eine Einladung, sein Wiki als Ergänzung zu der von ihm gepflegten Portland Pattern Repository Website<sup>4</sup> zu besuchen [Cunningham2006a]:

*"(...) The database is made available to the web through a 'common gateway interface' that includes forms for authoring. You don't need to know HTML to add or modify pages. (...) Think of it as a moderated list where anyone can be moderator and everything is archived. It's not quite a chat, still, but conversation is possible."*

Diese Einladung führte dazu, dass sein Wiki immer stärker genutzt und immer mehr Seiten im Wiki geschrieben wurden [Cunningham2006c].

Cunningham spricht davon, dass er die ersten Ideen für ein Wiki bereits in der 1980er Jahren hatte, als er mit HyperCard<sup>5</sup> arbeitete [Cunningham2006c]. Er nutzte HyperCard bis dahin als Ideen- und Informationssammlung zu Software Patterns [Cunningham2006d]. HyperCard funktionierte laut Cunningham sehr gut für diesen Zweck, das einzige Problem war, dass es auf einen Benutzer beschränkt war und die Ideensammlung nicht gemeinsam mit anderen Personen bearbeitet werden konnte [Venners2003]. Ausgehend von der grundlegenden Funktionsweise eines HyperCard-Stack schrieb Cunningham ein Perl-Skript, mit dem es möglich war Texte, die in einzelnen Dateien gespeichert sind, direkt in einem Webbrowser mit HTML-Formularen über das WWW zu bearbeiten. Er schuf mit diesem ersten implementierten Wiki eine einfache und schnelle Möglichkeit, Texte online im WWW zu verfassen und zu veröffentlichen

---

<sup>3</sup>patterns@cs.uiuc.edu

<sup>4</sup><http://c2.com/ppr>

<sup>5</sup>HyperCard wurde 1985 von Bill Atkinson für Apple Macintosh-Computer entwickelt. HyperCard ist ein Hyperkarteikartensystem zur Verwaltung persönlicher Datenbasen. Ein HyperCard-Knoten ist eine frei formatierbare Karte, die neben Text auch Grafiken, Videos und Audiodaten enthalten kann. Karten sind zu Karteien zusammengefasst, die als Stacks (dt. Stapel) bezeichnet werden. Als Quellanker für Verweise kann jedes Objekt auf einer Karte dienen. Zielanker ist immer eine Karte in einem beliebigen Stack. (vgl. [Schnupp1992], Kapitel 5.5, S. 103ff.).

sowie dritten Personen das Kommentieren, Korrigieren, Ändern und Ergänzen der Texte zu erlauben. Dieses einfache Prinzip hat sich beim Portland Pattern Repository und später auch in anderen Einsatzbereichen bewährt.

Mit ein wenig Verwunderung schreibt ein Besucher des Wikis von Ward Cunningham folgenden Kommentar zur Darstellung Cunninghams, dass seine Ideen auf HyperCard zurück gehen:

*„This all makes me wonder if wiki, rather than being a brilliant new idea, is actually just a (brilliant) new implementation of an old brilliant idea. Perhaps the idea has never had an implementation before. Or maybe it’s an idea which is so old and common that I don’t recognize it as a single idea. – TomAnderson“* (nach [Cunningham2006d])

Und in der Tat, Ward Cunningham war nicht der erste, der diese Idee hatte. Er hat sie nur mit einfachsten Mitteln neu umgesetzt. Die Idee geht vielmehr auf eine Reihe von Personen zurück, angefangen bei Vannevar Bush mit seinem Aufsatz „As we may think“, in dem er die Vision eines universellen Dokumentationssystems Memex vorstellt [Bush1945], über Ted Nelson, der 1965 den Begriff Hypertext einführte und das Hypertextsystem Xanadu entwickelte [Nelson2006], Douglas Engelbart, dem Entwickler der ersten Computer-Maus und der kollaborativen Arbeitsumgebung NLS (oN Line System) / Augment [MouseSite2006], Bill Atkinson mit HyperCard für den Apple Macintosh [Schnupp1992] bis hin zu Tim Berners-Lee, dem Entwickler des World Wide Web [Berners-Lee2000].

### 2.1.2. Per Anhalter durch die Galaxis

Der Roman „Hitchhiker’s Guide to the Galaxy“ von Douglas Adams aus dem Jahr 1979 wird oft als literarische Vorlage des Wiki-Prinzips dargestellt (vgl. [Möller2005a]), obwohl Ward Cunningham nie einen Bezug zu diesem Roman hergestellt hat. Der „Anhalter“ ist eine Mischung aus Lexikon, Enzyklopädie und Ratgeber und wird von mehreren Autoren, die durch das Universum reisen, gemeinsam geschrieben und aktualisiert<sup>6</sup>.

---

<sup>6</sup>Und der große Vorteil des Anhalters ist, es „stehen auf seinem Umschlag in großen, freundlichen Buchstaben die Worte KEINE PANIK.“. Damit ist auch geklärt, warum diese Arbeit mit eben diesen Worten aus der englischen Originalausgabe des Anhalters überschrieben ist.

*„Der Reiseführer Per Anhalter durch die Galaxis ist ein wirklich sehr interessantes Buch. Er ist über viele Jahre von vielen verschiedenen Redakteuren zusammengetragen und viele Male umgearbeitet worden. Er enthält die Beiträge unzähliger Reisender und Forscher.“* ([Adams1981], S. 75)

Im April 1999 (also lange nachdem Ward Cunningham das erste Wiki veröffentlichte) startete Douglas Adams mit seiner Firma „The Digital Village (TDV)“ die Website h2g2 (<http://www.h2g2.com>), mit der die Idee des „Anhalters“ im Web umgesetzt werden sollte [Adams2000]. h2g2 unterscheidet zwischen einem offenen und einem redaktionell gepflegten Teil sowie freiwilligen, festen und bezahlten Redakteuren. Um einen Artikel auf h2g2 zu veröffentlichen, müssen sich die Benutzer anmelden. Artikel durchlaufen einen mehrstufigen Prozess von Peer-Reviews bis zur Veröffentlichung. Nach Adams Tod im Mai 2001 wurde h2g2 von der BBC übernommen, die schon in den 1970er Jahren eine Radioserie finanzierte, auf der Adams Bücher basieren [Möller2005a].

### **2.1.3. Wikipedia**

Das bekannteste und wohl auch größte öffentliche Wiki ist die freie Enzyklopädie *Wikipedia*. Mit dem Erfolg der Wikipedia wurden Wikis zunehmend in der Öffentlichkeit bekannt. Am 20. September 2004 erreichte die Wikipedia die Marke von mehr als 1 Million Artikeln in über 100 Sprachen [Fiebig2005]. In einem Artikel der Tageszeitung *Neue Westfälische* vom 14./15. Januar 2006 wird berichtet, dass die deutschsprachige Version der Wikipedia 300.000 Artikel, die englischsprachige Version 850.000 Artikel umfasst [NW2006]. Die Idee zur Wikipedia hat ihre Wurzeln im Nupedia-Projekt. Die Nupedia wurde im März 2000 von Jimmy Wales und Larry Sanger gegründet [Fiebig2005]. Artikel für die Nupedia wurden vor der Veröffentlichung einer Prüfung durch „Peers“, qualifizierten Experten und Fachleuten aus den jeweiligen Fachgebieten, unterzogen. Durch das Redaktionssystem sollte die Qualität der Artikel sichergestellt werden. Aber genau dieses System behinderte die Nupedia, *„anstatt neue Artikel zu verfassen, stritten sich die Benutzer über die Qualität der schon bestehenden Artikel so lang, bis das Projekt beendet wurde“* [Fiebig2005]. Die Nupedia hat in den Jahren ihrer Existenz lediglich ca. 30 Artikel hervorgebracht [Möller2005a].

Während der Arbeit an der Nupedia kamen Sanger und Wales auf die Idee, ein Wiki als eine Art Vorstufe zum Erstellen von Artikeln für die Nupedia zu nutzen. Am 15. Januar 2001 wurde das Projekt unter dem Namen „Wikipedia“ gestartet. Die Wikipedia zog so viele Nutzer an, dass sie bereits einen Monat nach dem Start ca. 1000 Artikel umfas-

ste [Fiebig2005]. Das Projekt entwickelte eine große Eigendynamik, so dass bald Ableger in anderen Sprachen gegründet wurden, darunter: Französisch, Spanisch, Deutsch, Dänisch, Schwedisch und andere. Die Nupedia brachte die Wikipedia allerdings nicht weiter; kaum Artikel fanden den Weg aus der Wikipedia in die Nupedia, so dass sie eingestellt wurde. Das rasante Wachstum der Wikipedia hält an und mit ihm steigt die Bekanntheit sowie das allgemeine Interesse an Wikis (vgl. [Ebersbach2005]).

Neben einer zunehmenden Anzahl von Fachpublikationen zu Themen Rund um Wikis und der immer größeren Erwähnung auch in der allgemeinen Presse, äußert sich das steigende Interesse an Wikis auch in einer ersten internationalen Konferenz unter dem Titel „Wikimania“, die im August 2005 in Frankfurt/Main stattfand [Wikimania2005]. Die nächste Wikimania soll im August 2006 in Cambridge, Massachusetts stattfinden.

## 2.2. Was ist ein Wiki?

Um zu beschreiben, was ein Wiki ist, müssen zwei Aspekte berücksichtigt werden: erstens Wiki als Konzept und zweitens Wiki als Software (vgl. [Leuf2001], [Burry2005]). Das Konzept steht für die Idee des einfachen und unkomplizierten Bearbeitens eines (Hyper)Textes im Browser; die Software setzt diese Idee um.

Das Wort *Wiki* stammt aus dem Hawaiianischen von „*wikiwiki*“ und bedeutet soviel wie „*schnell*“<sup>7</sup>.

„*Wikiwiki (stative verb). Fast, speedy; to hurry, hasten; quick, fast, swift.*“  
(nach [Leuf2001])

Wiki steht somit auch für einen Teil des Konzeptes, Inhalte schnell und unkompliziert zur Verfügung zu stellen und bearbeiten zu können.

„*I chose wiki-wiki as an alliterative substitute for quick and thereby avoided naming this stuff quick-web.*“ (Ward Cunningham [Cunningham2006c])

**Wiki als Konzept** Ein *Wiki*, auch *WikiWiki*, *WikiWeb* oder *WikiWikiWeb*, ist eine Sammlung von Webseiten, die von den Benutzern nicht nur gelesen, sondern auch unmittelbar im Browser bearbeitet – ergänzt, kommentiert, verändert – werden können.

---

<sup>7</sup>Siehe auch [http://de.wikipedia.org/wiki/Hawaiianische\\_Sprache](http://de.wikipedia.org/wiki/Hawaiianische_Sprache) und <http://wehewehe.org> (Hawaiian Dictionaries).

**Wiki als Software** Ein *wiki*<sup>8</sup>, auch *WikiEngine*, ist eine (webbasierte) Software, die es den Benutzern einer Webseite erlaubt, den Inhalt der Seite zu bearbeiten – ergänzen, kommentieren, verändern –, indem sie die Seite über ein Formular direkt im Browser editieren.

Die Funktionsweise eines Wiki ist einfach: Auf jeder Seite gibt es einen Link oder eine Schaltfläche zum Bearbeiten der Seite. Der Inhalt kann direkt im Browser bearbeitet werden (vgl. Abbildung 2.2). Für die Gestaltung des Textes müssen die Bearbeiter nicht HTML verwenden. Text kann unformatiert eingegeben werden. Für bestimmte Formatierungen steht eine vereinfachte Syntax zur Verfügung, z.B. "text", um ein Wort in Kursivschrift darzustellen oder `==text==`, um den Text als Überschrift zweiter Ordnung zu formatieren. Die Gesamtheit der vorhandenen Formatierungen bzw. Auszeichnungen wird als *Wiki-Syntax* oder *Wiki-Markup* bezeichnet. Im allgemeinen wird eine einfache Menge an Auszeichnungen angeboten, die sich auf das Wesentliche beschränkt. Nach dem Speichern des bearbeiteten Seiteninhalts werden die Formatierungen für die Anzeige im Browser in HTML umgesetzt.

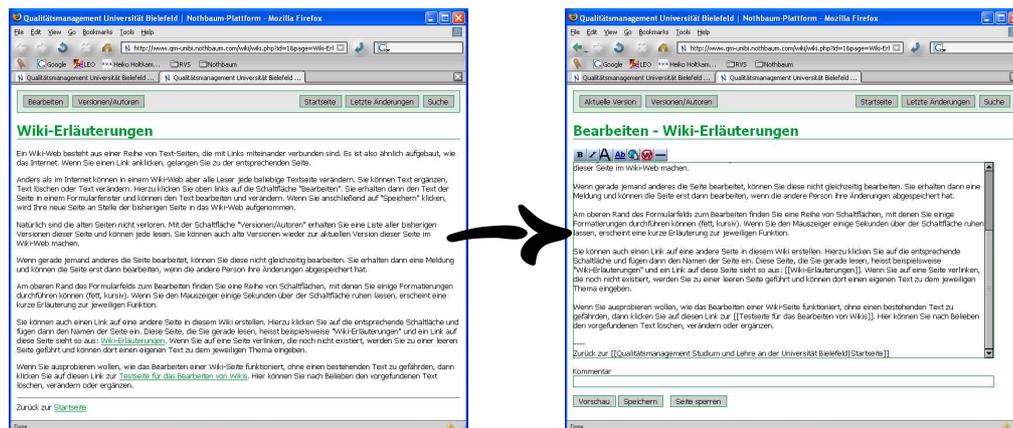


Abbildung 2.2.: Bearbeitung einer Wiki-Seite im Browser: links die Wiki-Seite, rechts das Formular zum Ändern derselben.

Das Erstellen von Verweisen folgt einem ebenfalls sehr einfachen Schema, den sogenannten *Wiki-Wörtern* (*WikiWords*). Wörter oder Zeichenfolgen, die mit einem Groß-

<sup>8</sup>„The proper term "Wiki" is used in this book to refer more to the essential concept than to any particular implementation, the latter being called simply a "wiki". (This is similar to the distinction between "Perl" the language and "perl" the implementation.“ ([Leuf2001], S. 14).

Im Weiteren wird diese Unterscheidung zwischen Wiki und wiki nicht getroffen; es sollte an den entsprechenden Stellen klar sein, ob es um das Konzept oder die Implementierung geht.

buchstaben beginnen und einen Großbuchstaben innerhalb des Wortes haben, werden als Verweise auf eine Seite mit diesem Namen interpretiert, z.B. „WikiWiki“ oder „EineNeueSeite“ (diese Schreibweise wird wegen der hervorstehenden Großbuchstaben auch als „*CamelCase*“ bezeichnet (vgl. [Möller2005a]). Existiert eine Seite mit dem Namen noch nicht, kann sie durch Anklicken des Links angelegt werden, andernfalls verweist der Link auf die entsprechende Seite. Im Original-Wiki von Ward Cunningham wird neben dem Link ein Fragezeichen angezeigt, falls die Seite noch nicht existiert. Wiki-Klone unterscheiden sich hier zum Teil in der Darstellung, z.B. wird bei einigen zwischen einem Link auf eine existierende oder nicht existierende Seite über die Farbe des Linktextes unterschieden: rot für eine nicht vorhandene Seite, blau für eine vorhandene Seite.

Eine neue Seite in einem Wiki wird immer dadurch erstellt, dass sie zuerst als Link auf einer bereits vorhandenen Seite eingetragen wird. Dadurch entsteht eine direkte Vernetzung der Seiten im Wiki. Abbildung 2.3 zeigt die Konstruktion eines einfachen Wikis. Sie zeigt das Anwachsen ausgehend von einer bestehenden Seite und die implizite Erstellung von Seiten („Strukturen entstehen bottom-up“ [Thelen2003]). Jede Seite wird in zwei Schritten erzeugt: im ersten Schritt wird der Name der Seite als Verweis auf einer existierenden Seite eingetragen. Im zweiten Schritt wird die Seite beim ersten Klick auf den zu ihr verweisenden Link erstellt, indem dem Benutzer das Formular zum Bearbeiten des Seiteninhalts angezeigt wird und er den initialen Seiteninhalt eingeben kann.

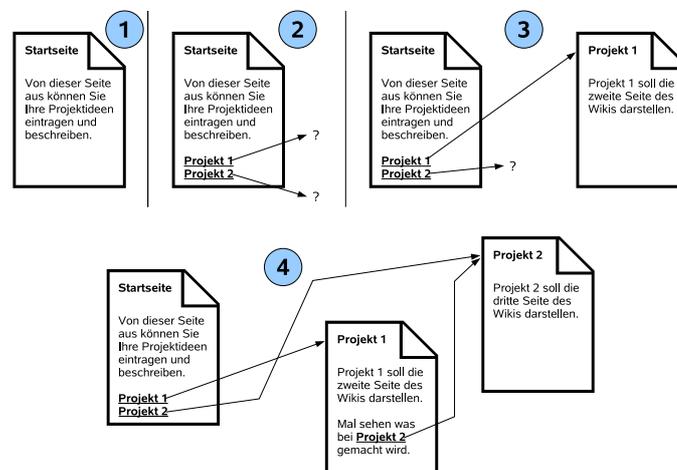


Abbildung 2.3.: Entstehende Vernetzung bei der Bearbeitung von Wiki-Seiten: (1) Initiales Wiki, mit nur einer Seite; (2) ein Benutzer bearbeitet die Seite und erstellt zwei Verweise; (3) ein Benutzer erstellt die Seite „Projekt 1“; (4) ein Benutzer erstellt die Seite „Projekt 2“.

Das Setzen von Verweisen mit Wörtern in CamelCase gilt jedoch als unschön (vgl. [Möller2005a]). Die Lesbarkeit der Seiten wird durch die Groß-/Kleinschreibung und zum Teil komische Schreibweise gestört. Einher geht dies mit unterschiedlichen Schreibweisen ein und desselben Begriffs: SchreibWeise, SchreibweisE etc. Da diese Wörter auch für die Titel der Seiten stehen, sind „SchreibWeise“ und „SchreibweisE“ zwei unterschiedlichen Seiten. Aus diesem Grund hat sich zunehmend eine zusätzliche bzw. CamelCase ablösende Variante herausgebildet, die sogenannten *Free-Links* (vgl. [Möller2003]). Um einen Link zu setzen, werden um einen Text doppelte eckige Klammern gesetzt: [[Ein Free Link]].

Ähnlich einfach, wie das Bearbeiten und Erstellen einer Wiki-Seite, ist die Navigation in einem Wiki. Sie folgt im Wesentlichen über die im Inhalt der Wiki-Seiten zu findenden Verweise auf andere Wiki-Seiten oder externe Seiten im WWW. Mit der u.U. starken Vernetzung der Wiki-Seiten untereinander und auf andere externe Seiten kann bzgl. der Navigation im Wiki bzw. im Hypertext ein Problem auftreten, dass in der Literatur oft als *lost in hyperspace* („Verloren im Hyperraum“) bezeichnet wird, einer Desorientierung in den miteinander verlinkten Texten (vgl. [Schnupp1992], [Ernst2005], [Schulmeister2002]). Schulmeister hebt aber hervor, dass dies auch ein – in gewissen Grenzen – gewünschter Effekt sein kann:

„(...) *ich stimme seinem Wunsch nach aufregenderen Anwendungen zu: ‚The complete absence of orientational challenges is dull and uncomfortable. A boring hypertext is every bit as bad as a confusing one.‘*“ ([Schulmeister2002], Seite 59)

Eine Alternative These zum „lost in hyperspace“ ist laut Schulmeister [Schulmeister2002] „*serendipity*“ (zu deutsch soviel wie „*Spiürsinn*“, „*Mehr Glück als Verstand*“ – Collins German Dictionary 3. Auflage 1998). „*Zuweilen geschieht es, dass man auf der Suche nach einer bestimmten Information so ‚beschlagahmt‘ wird, dass über deren aktueller Dominanz das ursprüngliche Ziel irrelevant oder vergessen wird. Dies bezeichnet man als ‚Serendipity-Effekt‘.*“ (zitiert nach [Schulmeister2002], Seite 59). In entdeckenden oder explorativen Lernsituationen kann dies ein gewünschter Effekt sein.

Wikis weisen bezüglich der Navigation oft noch weitere Funktionen auf (nicht alle Wikis setzen diese Funktionen um):

**Letzte Änderungen** (Recent Changes). Eine automatisch erzeugte Liste über kürzlich oder innerhalb eines bestimmten Zeitraums geänderter oder neu erzeugter Seiten.

**Verweisende Seiten** (Referer). Eine Liste aller Seiten, die auf die aktuell angezeigte Seite verweisen.

**Historie/Versionen** (History/Versions). Diese Funktion speichert alle alten Versionen bzw. Veränderungen einer Seite. Damit ist es möglich, den Bearbeitungsprozess einer Seite genau nachzuverfolgen. Eine ältere Version kann wieder geöffnet werden und zur aktuellen Version gemacht werden. Auf diese Weise können ungewünschte Änderungen an einer Seite ohne großen Aufwand rückgängig gemacht werden. Die Historie kann bis zur ersten Version einer Seite zurückreichen, ist aber aus Speicherplatzgründen oft auf eine bestimmte Anzahl oder einen bestimmten Zeitraum begrenzt.

Einige Wiki-Engines bieten auch eine Vergleichsfunktion – eine sogenannte Diff-Funktion<sup>9</sup> –, die die Änderungen/Differenzen zwischen zwei Versionen einer Seite anzeigt.

**Suchfunktionen** Der Großteil der Wiki-Engines bietet zudem eine Suchfunktion an, mit denen eine Titelsuche (nach den Namen einer Seite) oder eine Volltextsuche durchgeführt werden kann. Die Suche ist eine der wichtigsten Recherche-Funktionen in einem Wiki („*A decent search function is one of the strengths of a wiki and a valuable complement to whatever structure you give the page database.*“ [Leuf2001], Seite 92).

## 2.3. Wiki-Klone und Wikis

### 2.3.1. Wiki-Klone

Insgesamt gibt es hunderte verschiedener Wiki-Engines. Die meisten davon sind Open-Source-Software und stehen unter einer entsprechenden Lizenz (wie. z.B. der GNU General Public License (GPL), <http://www.gnu.org>). Ausführliche Listen mit Wiki-Klonen werden in Ward Cunninghams Wiki<sup>10</sup> und in der Wikipedia<sup>11</sup> geführt. Zu den bekanntesten und verbreitetsten Wiki-Engines gehören (vgl. [Huhmann2002], [Huhmann2004], [Klein2003], [Möller2003]):

---

<sup>9</sup>Die Bezeichnung geht auf das von Unix bekannte diff-Tool zurück

<sup>10</sup><http://c2.com/cgi/wiki?WikiEngines>

<sup>11</sup>[http://en.wikipedia.org/wiki/List\\_of\\_wiki\\_software](http://en.wikipedia.org/wiki/List_of_wiki_software), [http://en.wikipedia.org/Comparison\\_of\\_wiki\\_software](http://en.wikipedia.org/Comparison_of_wiki_software)

**UseModWiki, kurz UseMod** (<http://www.usemod.com>). UseMod ist ein einfaches, dateibasiertes Wiki, das von Clifford Adams in Perl geschrieben wurde. UseMod (der Name leitet sich von Usenet Moderation Project ab) ist eine der ältesten und verbreitetsten Wiki-Engines und basiert auf Ward Cunninghams Ursprungs Wiki. Unter <http://usemod.com/cgi-bin/wiki.pl?SiteList> steht eine Liste der mit UseMod betriebenen Wikis. UseMod dient vielen anderen Wikis als Vorbild, insbesondere die Markup-Elemente bzw. deren Syntax finden sich in vielen anderen Wikis wieder. UseMod war auch lange Zeit die Wiki-Engine der Wikipedia, bis sie durch MediaWiki ersetzt wurde.

**MediaWiki** (<http://www.mediawiki.org>). MediaWiki wurde speziell für die Bedürfnisse der Online-Enzyklopädie Wikipedia entwickelt. Der Code basiert auf der Skriptsprache PHP. Zur Speicherung der Inhalte nutzt MediaWiki eine MySQL-Datenbank. Die Hauptentwickler sind Magnus Manske, Lee Daniel Crocker, Tim Starling und Erik Möller. Neben einigen Erweiterungen für die Wikipedia (wie Namensräume, Behandlung von Versionskonflikten etc.) wurde MediaWiki vor allem mit einem Schwerpunkt auf Skalierbarkeit, wegen der großen Datenmengen der Wikipedia, und Stabilität entworfen. MediaWiki geht auf UseMod zurück.

**TikiWiki** (<http://tikiwiki.org>). TikiWiki ist laut den Entwicklern nicht nur eine Wiki-Engine, sondern eine Web-Applikation, die unter anderem auch die Funktionen eines Content Management Systems (CMS) und Groupware-Software implementiert. Die Wiki-Engine ist nur ein Bestandteil unter weiteren Modulen für Foren, Blogs, Newsletter, Dateibereiche, Bildergalerien, Chats, Kalendern und Umfragen. TikiWiki ist in PHP programmiert und erfordert zum Speichern der Inhalte ein DBMS, das von der Bibliothek ADODB für PHP unterstützt wird. TikiWiki bringt eine umfangreiche Benutzerverwaltung und Zugriffskontrolle mit.

**TWiki („tweaky“)** (<http://twiki.org>). TWiki ist neben TikiWiki einer der umfangreichsten Wiki-Klone und bietet ebenso zahlreiche Groupware-Funktionen. TWiki ist stark auf die Erweiterbarkeit mit Plugins ausgerichtet. Unter <http://twiki.org/cgi-bin/view/Plugins/PluginPackage> wird eine ausführliche Übersicht von verfügbaren Plugins geführt. TWiki ist in Perl geschrieben und speichert Inhalte im Dateisystem. Die Versionsverwaltung der Inhalte wird über das Versionierungssystem *Revision Control System (RCS)* durchgeführt. TWiki wird laut der TWiki-Website in mehreren Installationen von namhaften Firmen genutzt (u.a. Motorola, Texas Instruments, Yahoo, SAP, Wind River, Disney). [Huhmann2002] nennt Motorola mit „*mittlerweile mindestens fünf Installationen*“ als einen der Anwender<sup>12</sup>.

---

<sup>12</sup>[Leuf2001] führt aus, dass TWiki 1999 von Crawford Currie als „Team Communication Tool“ im Be-

**MoinMoin** (<http://moinmoin.wikiwikiweb.de>). MoinMoin ist eine einfache, von Thomas Waldmann und Jürgen Hermann in Python programmierte Wiki-Engine. MoinMoin speichert die Inhalte im Dateisystem. MoinMoin-eigene Makros, die ebenfalls in Python programmiert werden, bieten eine einfache Möglichkeit die Funktionalität der Wiki-Engine zu erweitern. Mit Access Control Lists (ACLs) können Zugriffsrechte für Benutzer im Wiki fein steuerbar vergeben werden. Für die Behandlung von Versionskonflikten bietet MoinMoin drei verschiedene Methoden: (i) keine Behandlung von Versionskonflikten, (ii) Benutzer werden gewarnt, falls jemand anderes die Seite bereits bearbeitet und (iii), eine Seite die gerade bearbeitet wird ist gesperrt, so dass es nicht möglich ist, dass zwei Nutzer zur gleichen Zeit an der Seite arbeiten.

**PmWiki** (<http://pmwiki.org>). PmWiki ist nach Patrick Michaud benannt, der die Entwicklung begonnen hat; mittlerweile wird diese von einer Entwicklergruppe voran getrieben. PmWiki ist in PHP programmiert und speichert die Inhalte im Dateisystem. PmWiki zeichnet sich vor allem durch eine sehr gute und ausführliche Dokumentation aus. Zwei der wesentlichen Grundsätze der Entwickler von PmWiki lauten[Michaud2005]: „*Don't try to replace HTML*“ – nicht jede Formatierungsmöglichkeit von HTML soll in die Wiki-Syntax integriert werden – und „*Avoid gratuitous features*“ – überflüssige Funktionen sollen vermieden und Erweiterungen gründlich durchdacht werden.

**FlexWiki** (<http://www.flexwiki.org>). FlexWiki wird hier weniger wegen seiner Verbreitung aufgeführt, sondern weil es gegenüber den anderen Wiki-Engines quasi eine Sonderstellung einnimmt. FlexWiki ist ein ehemaliges Mitarbeiter-Projekt von Microsoft, das im Rahmen der Shared Source Initiative maßgeblich von David Ornstein entwickelt wurde. FlexWiki nutzt das .NET-Framework (ASP.NET) von Microsoft und speichert Daten wahlweise im Dateisystem oder in einer SQL-Datenbank. Als Web-Server erfordert FlexWiki Microsofts Internet Information Server 5.0 (IIS). Als Besonderheit bietet FlexWiki eine eigene Skriptsprache namens FlexTalk, die auf Smalltalk basiert. WikiTalk erlaubt das Einbinden dynamischer Inhalte in Wiki-Seiten.

---

reich „Motorola Systems-on-Chip Design and Technology“ in Großbritannien eingeführt wurde. Durch den Erfolg des Einsatzes eines Wikis als neue Kommunikationslösung im Projekt- und Teamarbeitsbereich wurden auch andere Arbeitsgruppen darauf aufmerksam und begannen eigene Wikis aufzubauen oder an den bestehenden Wikis teilzunehmen: „*We are now hosting 7 different webs, extending the 'team' from an on-site project team to a virtual team including members in Germany, UK, France, Australia, Russia and the US, with about 60 regular contributors (and growing).*“ [Currie2000].

### 2.3.2. Wikis

Neben einer Vielzahl verschiedener Wiki-Engines gibt es auch Wikis zu fast jedem erdenklichen Thema (vgl. [Möller2005a], [Ebersbach2005], [Leuf2001]):

- Portland Pattern Repository, das Ursprungs-Wiki (<http://c2.com>);
- die Online-Enzyklopädie Wikipedia als eines der größten Wikis (<http://www.wikipedia.org>);
- Wikitionary ein mehrsprachiges Wörterbuch (<http://www.wikitionary.org>);
- Wikitravel ein Reiseführer (<http://www.wikitravel.org>);
- Wikibooks, eine Sammlung freier (Lehr)Bücher (<http://www.wikibooks.org>);
- Wikiweise, eine Alternative zur Wikipedia (<http://www.wikiweise.de>);
- LinuxWiki, eine Plattform für deutschsprachige Linux-Anwender und -Entwickler (<http://linuxwiki.org>);
- JuraWiki, das Wiki für Rechtsfragen (<http://www.jurawiki.de>) etc.

Ein sehr interessantes allgemeines Wiki ist der „virtuelle Reisebus“: TourBus (<http://www.usemod.com/cgi-bin/mb.pl?TourBusStop>). Die TourBus-Karte führt eine Liste mit verschiedenen Buslinien, die einzelne (thematisch zusammengefasste) Wiki-Reiseziele als Haltepunkte haben. Fast jedes bekanntere Wiki ist mittlerweile als Reiseziel aufgenommen und fast jedes bekanntere Wiki hat mittlerweile eine sogenannte Haltestelle für den TourBus (dies ist zumeist eine spezielle Seite, die den Titel „TourBusStop“ oder deutsch „TourBusHaltestelle“ trägt).

## 2.4. Wiki-Syntax

Obwohl Wikis darauf ausgelegt sind, dass sich die Nutzer weitestgehend nicht um das Layout des Textes kümmern müssen bzw. kümmern sollten, ist es oft wünschenswert und auch für das Verständnis eines Textes erleichternd, wenn Möglichkeiten zur Formatierung des Textes zur Verfügung stehen.

Cunningham wollte die Barrieren für die Teilnahme an einem Wiki so gering wie möglich halten (vgl. Zitate auf Seite 5 und 6), weshalb er für die Auszeichnung von Formatierungen im Text nicht HTML wählte, um das Bearbeiten des Textes einfach zu gestalten.

*„So, the Wiki rule for editing is simple: enter your notes as paragraphs of text, written as clearly as possible. Focus on content, not styling. Plain text can go a long way as typewriters have shown over the past century or so and much e-mail continues to in our time.“* ([Leuf2001], S. 97)

Es muss aber auch gesagt werden, dass ein Grund für die Nicht-Nutzung von HTML zur Auszeichnung von Texten im Wiki darin bestand, dass die Versionen der Browser 1995 zum Teil nicht dazu in der Lage waren, HTML-Code in den Texteingabefeldern eines HTML-Formulars zu handhaben (vgl. [Leuf2001], S. 403). Cunningham entschloss sich dazu, spezielle Textmuster zu verwenden, um Formatierungen kenntlich zu machen. Diese Textmuster sollten (i) den einzugebenden Text so nah wie möglich an reinem Text (plain text) belassen, (ii) einfach einzugeben sein und (iii) sich auf ein Minimum beschränken (vgl. [Leuf2001]).

- (i) *„The chosen starting set of content markup options implemented as simple text patterns allows meaningful structuring of content. It aims to enhance browsing readability without degrading editing readability. Heavily tagged HTML source is by contrast hardly easy to read.“* (S. 98)
- (ii) *„A wiki should avoid typographically unclear patterns and potentially hard-to-find or difficult-to-type characters.“* (S. 111)
- (iii) *„(...) This leads to the second statement about markup: Less is more.“* (S. 114)

Zur Darstellung der Inhalte der Seite, werden die Textmuster von der Wiki-Engine in HTML umgesetzt und dann im Browser dargestellt (vgl. Abbildung 2.2).

### 2.4.1. Syntax-Standards

Für die in Wikis verwendeten Textmuster gibt es keinen einheitlichen Standard, nicht einmal einen einheitlichen Namen. Einige Quellen nennen die Menge an Textmustern *Wiki-Syntax*, andere Quellen *Wiki-Markup* oder *Wiki-Text*. Abbildung 2.4 stellt für die in Kapitel 2.3.1 vorgestellten Wikis einen Auszug der Textmuster in einer Tabelle dar.

Verschiedene Wiki-Engines unterscheiden sich zum Teil erheblich bei den verwendeten Textmustern sowie im Umfang der angebotenen Formatierungen für den Text. Während sich einige Wikis, wie das Ursprungs-Wiki von Ward Cunningham, auf einen Basissatz von Textformatierungen beschränken, erweitern andere die Möglichkeiten auf Tabellen, die Formatierung von mathematischen Formeln, interaktive Elemente wie Umfragen, Makros etc. Viele Wiki-Engines orientieren sich aber weiterhin stark an der von Ward Cunningham eingeführten Syntax für die Textmuster.

Es gibt einige Bemühungen, die Wiki-Syntax zu standardisieren. Im Meatball-Wiki wird zu diesem Thema eine allgemeine und mehrere Wiki-Engines umfassende – zum Teil sehr kontroverse – Diskussion geführt (vgl. [Meatball2006a], [Meatball2006b]).

Befürworter einer standardisierten Syntax argumentieren, dass es für Personen, die an mehreren Wikis mit unterschiedlicher Syntax teilnehmen, schwer ist, immer wieder die verschiedenen Syntaxvarianten zu erinnern bzw. zu lernen. Ebenso erschwert es den Austausch von Informationen zwischen Wikis.

Kritiker heben hervor, dass es einen Grund gibt, warum die Syntax verschieden sein sollte: die Wiki-Syntax sei als Teil des Benutzerinterfaces zu verstehen („*There's little consistency across WikiEngines for a variety of reasons. One, there should be no standard syntax. The WikiSyntax is the UserInterface, it is not the data encoding.*“ [Meatball2006a]). Wikis müssen ihre Syntax für die Textmuster ändern können, um den Bedürfnissen der Benutzer entgegen zu kommen. So kann es von dem einen Anwenderkreis bevorzugt werden, hervorzuhebenden Text in Sternchen (\* \*) einzuklammern, während ein anderer vielleicht Unterstriche (\_ \_) oder wie bei Ward Cunninghams Ur-Wiki doppelte einfache Anführungszeichen (" ") vorzieht.

Mit dem RFCWiki-Dokument [Martin2004] wird von der Tiki Working Group versucht einen „IETF (Internet Engineering Task Force) RFC Draft for Standardization of the Wiki Syntax“ zu entwickeln. Bisher ist aber noch kein Vorschlag als RFC bei der IETF aufgenommen. Die Diskussion um ein Wiki-Austauschformat (Wiki Interchange Format) versucht eine andere Alternative auf den Weg zu bringen [Meatball2006c]. Mit einem Austauschformat soll es möglich werden, Daten zwischen verschiedenen Wikis auszutauschen und erleichtert werden, Wiki-Inhalte in andere Formate wie PDF, RTF etc. zu konvertieren.



## 2.4.2. Texte gliedern und formatieren

Nachfolgend sollen einige Möglichkeiten in Kürze vorgestellt werden, Texte in einem Wiki zu formatieren. Die Darstellungen halten sich dabei im Wesentlichen an die Syntax der Wiki-Engines UseMod und MediaWiki (vgl. Kapitel 2.3.1). Beide orientieren sich an der Syntax des Ur-Wikis von Ward Cunningham. Damit wird auch in Grundzügen die Syntax dieses Wiki-Veterans vorgestellt. Ausführliche Angaben zu den vollständigen Formatierungsmöglichkeiten bieten die Webseiten zu den jeweiligen Wiki-Engines.

### Absätze, Zeilenumbrüche und Leerräume

Absätze werden in allen Wikis durch einen doppelten Zeilenumbruch – dem Einfügen einer extra Leerzeile – erzeugt. Ein Zeilenumbruch wird bei UseMod und bei MediaWiki durch das Einfügen des HTML-Elements `<BR />` erreicht. Einige andere Wikis erzwingen einen einfachen Zeilenumbruch über das Einfügen zweier Backslashes `\` am Zeilenende, wie dies auch von  $\text{\LaTeX}$  bekannt ist. Whitespaces (Leerzeichen, Zeilenvorschübe etc.) werden, wie auch in HTML üblich, als einzelnes Leerzeichen behandelt.

```
Hier beginnt der erste Absatz auf der Seite, \\  
mit einem eingefügten Zeilenumbruch.
```

```
Weiter geht es mit Absatz 2.
```

### Überschriften

Zur Formatierung von Überschriften haben sich in Wikis zwei Schreibweisen durchgesetzt. Verschiedene Ebenen von Überschriften werden formatiert, indem sie mit der entsprechenden Zahl von Gleichheitszeichen eingeklammert werden (`== Überschrift Ebene 2 ==`) oder eine entsprechende Anzahl Ausrufezeichen vor den Text der Überschrift gesetzt wird (`!!! Überschrift Ebene 3`). Überschriften beginnen stets am Anfang einer Zeile.

```
= Beispielseite =  
Und weiter geht es mit dem nächsten Absatz...
```

## Listen

Die Syntax für Listen ist in so gut wie allen Wikis gleich. Die Elemente einer Aufzählungsliste beginnen mit einem Stern (\*) am Anfang einer Zeile; für geschachtelte Listen werden mehrere Sterne an den Anfang der Zeile geschrieben, entsprechend der Tiefe der Schachtelung. Nummerierte Listen werden anstelle des Sterns mit dem Doppelkreuzzeichen (#) begonnen. Im Allgemeinen besteht kein Einfluß auf die Art die Nummerierung, es werden immer arabische Ziffern verwendet. Einige Wiki-Engines erlauben auch die Mischung von nummerierten Listen und Aufzählungslisten.

```
Diese Seite hat den Zweck
* Beispiele für die Formatierungen in einem Wiki zu zeigen
* und diese Beispiel zu erläutern.
```

Die Syntax für Definitionslisten unterscheidet sich bei verschiedenen Wiki-Engines. Bei UseMod wird eine Definitionsliste mit einem Semikolon (;) am Anfang der Zeile eingeleitet, ein Doppelpunkt (:) trennt den zu definierenden Begriff vom Text der Definition ab. Schachtelungen werden über eine entsprechende Anzahl Semikola angegeben. MediaWiki verwendet für Schachtelungen eine andere Syntax, hier muss über eine entsprechende Anzahl Doppelpunkte vor dem Semikolon die Schachtelungstiefe angegeben werden.

```
;ARPA:Advanced Research Projects Agency
;IAB:Internet Architecture Board
;IETF:Internet Engineering Task Force
```

## Logische Auszeichnungen im Text

Logische Auszeichnungen haben Bedeutungen wie „betont“ oder „emphatisch“ (wie dies z.B. aus HTML mit `<strong>` und `<emph>` oder in  $\text{\LaTeX}$  mit `\emph` bekannt ist). Bei logischen Auszeichnungen entscheidet das Programm zur Darstellung des Textes, wie ein solcher Text hervorgehoben wird. Die logischen Textauszeichnungen bzw. -muster in Wikis werden zumeist auf physische Textauszeichnungen von HTML abgebildet (z.B. fett, `<b>`, oder kursiv, `<i>`). In den Beschreibungen der Textmuster für Textauszeichnungen werden deshalb oft die physischen Textauszeichnungen von HTML als Bedeutung angegeben. Bei UseMod und MediaWiki wird Text, der von je zwei einfachen Anführungszeichen umschlossen ist, in Kursivschrift bzw. hervorgehoben dargestellt ("*kursiv*"); Text in drei einfachen Anführungszeichen fett bzw. stark hervorgehoben ("**fett**"). Eine häufige alternative Syntax bei anderen Wiki-Engines ist, für Kursivschrift den Text in zwei

Unterstrichen einzuklammern (`_kursiv_`) und für Fettschrift in zwei Sternchen (`*fett*`).

## Verweise

Grundsätzlich können zwei Verweistypen unterschieden werden: Wiki-interne und -externe Links.

### Wiki-interne Verweise

Wiki-interne Links verweisen auf Seiten innerhalb eines Wikis. Der Seitentitel ist in einem Wiki immer eindeutig, es kann keine zwei Seiten mit dem gleichen Titel geben. Es reicht daher für einen Wiki-internen Link aus, als Ziel den Seitentitel anzugeben.

In den frühen Wikis gab es nur eine Möglichkeit, Verweise zu machen: die Wiki-Wörter, WikiWords (vgl. Seite 11). Ein Wort in gemischter Groß-/Kleinschreibung (CamelCase-Schreibweise) wird als Link auf eine Seite mit dem entsprechenden Titel interpretiert. Der Vorteil dieser Art der Verlinkung ist ihre Einfachheit. Nachteile sind aber die zum Teil künstliche Schreibweise eines Wortes, um daraus einen Verweis zu machen (z.B. „Buch“); ein und derselbe Begriff kann unterschiedlich geschrieben werden und ist damit auch ein Verweis zu einer anderen Seite (z.B. „BuCh“, „BUCH“ etc.); in langen Texten kann die Lesbarkeit durch die CamelCase-Schreibweise gestört werden; und ein Wort, das normal in gemischter Groß-/Kleinschreibung vorkommt, kann fälschlicherweise als Verweis interpretiert werden (z.B. „MwSt.“).

Aufgrund der Nachteile der Wiki-Wörter haben sich daher die sogenannten Free-Links (vgl. Seite 11) etabliert. Der Titel der Seite, auf die verlinkt werden soll, wird dabei in doppelte eckige Klammern gesetzt (`[[Free Link]]`). Der Seitentitel wird dabei als Verweisbezeichnung genommen. Manchmal ist es wünschenswert, eine andere Beschreibung als Verweisbezeichnung angeben zu können (z.B. um vom Wort „Bücher“ auf die Seite „Buch“ verlinken zu können. Dies ist mit den sogenannten „benannten Free-Links“ (*itled links*<sup>13</sup> oder auch *piped links*<sup>14</sup>) möglich: `[[Seitentitel | Beschreibung]]`. Hinter dem Seitentitel steht ein senkrechter Strich (die „pipe“) gefolgt von einer Beschreibung, die als Verweisbezeichnung angezeigt wird (`[[Buch | Bücher]]`).

### Wiki-externe Verweise

Die einfachste Möglichkeit, von einer Seite in einem Wiki auf eine andere Adresse im WWW oder Internet zu verweisen ist, direkt die URL anzugeben. Der Link wird

---

<sup>13</sup>vgl. [Meatball2006b]

<sup>14</sup>vgl. [http://en.wikipedia.org/wiki/Wikipedia:How\\_to\\_edit\\_a\\_page](http://en.wikipedia.org/wiki/Wikipedia:How_to_edit_a_page)

von der Wiki-Engine entsprechend formatiert, wobei die URL als Verweistext angezeigt wird. Soll der externe Verweis nicht mit seiner URL, sondern mit einer alternativen Beschreibung dargestellt werden, kann dies zumeist mit einer den Freien-Links ähnlichen Syntax erreicht werden. Bei MediaWiki werden externe Links im Gegensatz zu den internen Links in einfachen eckigen Klammern notiert. Die URL des Links wird vom Beschreibungstext mit einem Leerzeichen statt des Pipe-Zeichens getrennt ([<http://www.beispiel.org> Beispiel]). Wird kein Beschreibungstext angegeben, werden die Links in der Reihenfolge ihres Auftretens im Text durchnummeriert und die Nummerierung als Verweistext in eckigen Klammern angezeigt.

## Tabellen

Einige Wiki-Engines führen eine eigene Syntax für die Darstellung von Tabellen bzw. tabellarischen Daten ein. Diese Syntax ist meistens nicht so mächtig und flexibel wie die von HTML. So ist es oft nicht möglich Einfluss auf die Textausrichtung (links, zentriert, rechts) in einer Tabellenspalte zu nehmen oder ineinander geschachtelte Tabellen zu definieren. Einfache Tabellen werden in UseMod wie in dem folgenden Beispiel gezeigt formatiert:

```
||Wiki||URL||  
||UseMod||http://www.usemod.com||  
||MediaWiki||http://www.mediawiki.org||  
||TikiWiki||http://tikiwiki.org. ||
```

Die Syntax für die Formatierung von Tabellen unterscheidet sich zwischen verschiedenen Wiki-Engines aber zum Teil erheblich. Auf diese Unterschiede soll bei dieser kurzen Darstellung nicht weiter eingegangen werden.

## 2.5. Anwendungsgebiete eines Wiki

Ebersbach, Glaser und Heigl stellen in [Ebersbach2005] die möglichen Anwendungsgebiete plakativ in einer Grafik dar. Tatsächlich sind die einzelnen Einsatzmöglichkeiten für ein Wiki schwer allgemein zu fassen und abzugrenzen. Der Einsatz ist im wesentlichen abhängig von den Zielen der Personen, Unternehmen oder Organisationen, die ein Wiki einsetzen wollen.

Grundsätzlich können zwei verschiedene Anwendungsmöglichkeiten von Wikis unter-

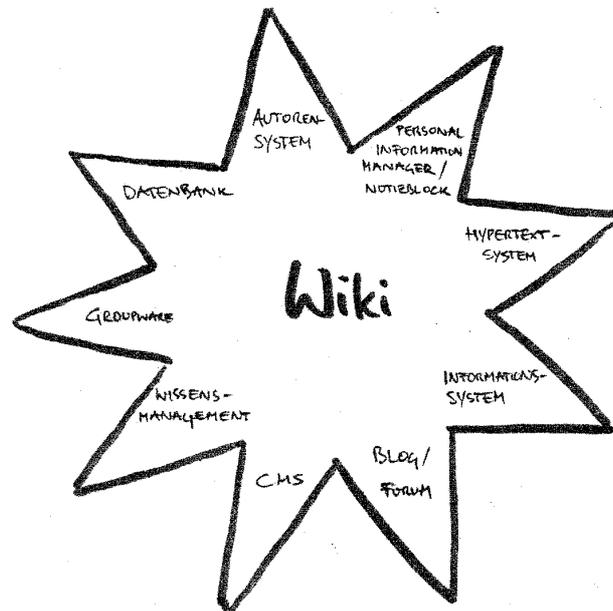


Abbildung 2.5.: Anwendungsgebiete eines Wiki (nach [Ebersbach2005]).

schieden werden: Wikis als Werkzeuge für geschlossene Arbeitsgruppen und Wikis, die sich potenziell über das WWW an alle wenden.

Wikis können als einfache, webbasierte Content Management Systeme (CMS) zur Verwaltung von Internet-Auftritten genutzt werden. Sie können im Projektmanagement zur Planung und Koordination angewendet werden oder als Wissensmanagement-Werkzeuge und Informationssysteme Anwendung finden. Als webbasierter Notizblock und zum Brainstorming eingesetzt werden. In E-Learning-Umgebungen werden Wikis zur kollaborativen Textproduktion und zur Gruppenarbeit genutzt. Die Anwendungsmöglichkeiten sind zahlreich.

Vor dem Einsatz eines Wikis für einen Anwendungsbereich ist von den Betreibern zu klären, ob das Wiki-Konzept tatsächlich für den von ihnen gedachten Einsatzzweck geeignet ist.

*„Not everyone needs a wiki. Not everyone wants a wiki. Not every situation benefits from becoming an open discussion or collaboration forum.“*  
 ([Leuf2001], S. 30)

Im Weiteren werden drei verschiedene Anwendungsszenarien eines Wikis kurz skizziert.

Die Liste möglicher Anwendungsszenarien ließe sich beliebig fortsetzen.

### **Wikis im Projektmanagement**

Im Bereich Projektmanagement können Wikis z.B. genutzt werden, um

- **Projektdokumentationen zu erstellen**  
Innerhalb eines Projektes schreibt im allgemeinen jedes der Projektmitglieder immer nur einen Teil der Projektdokumentation, den Teil an dem er beteiligt war. Bei größeren Projekten führt dies schnell dazu, dass die Dokumentationen an verschiedenen Orten verteilt sind. Selbst wenn ein zentraler Ort zum Speichern der Dokumentationen vorhanden ist, ist es oftmals schwer die verschiedenen Teildokumentationen zu verwalten und zu einem Ganzen zusammen zu fügen. Wikis bieten sich hier als Problemlösung an. Ein Wiki ist per se ein zentraler Speicherort. Durch die Möglichkeit die einzelnen Projektdokumentation miteinander zu verlinken entsteht schnell eine Gesamtdokumentation. Über die Versionsverwaltung können die Bearbeitungsschritte an einem Teildokument nachvollzogen werden; der gesamte Projektverlauf kann über die Recent Changes im Wiki nachverfolgt werden.
- **Projekttreffen planen**  
Die Planung von Projekttreffen und des Tagesordnungspunkte eines Treffens ist i.a. keine sehr beliebte Aufgabe bei den Projektteilnehmern. Die Person, die die Organisation des Treffens übernimmt wird oft mit zahlreichen E-Mails, Telefonanrufen, Faxen etc. von allen anderen Teilnehmern über deren Wünsche und Vorschläge informiert. Da niemand weiß, welche Vorschläge bereits beim Organisator eingereicht wurden, werden Vorschläge oft mehrfach eingereicht. Diese Organisation könnte auch über ein Wiki unter den Teilnehmern stattfinden. Jeder Teilnehmer kann seine Vorschläge direkt in das Wiki eingeben, wobei jeder sehen könnte, welche Vorschläge bereits von jemand anderem notiert wurden. Die Vorbereitung könnte dabei auf mehrere Personen verteilt werden und jeder Teilnehmer könnte schon vorab aktiv an der Gestaltung des Treffens teilnehmen.

### **Ein Wiki als Wissensmanagementsystem**

Wissensmanagement besteht aus drei Teilen, dem Erfassen von Wissen, dem Teilen von Wissen und dem Schaffen neuen Wissens aus der Kombination schon vorhandener Erkenntnisse (vgl. [Kluge2002]). Der Prozess des Wissensmanagements kann damit sehr

gut über ein Wiki abgebildet werden. Einzelne tragen ihr Wissen bei, in dem sie es in das Wiki auf einer Seite einpflegen. Die Informationen stehen anderen sofort zur Verfügung und über die Hypertextstruktur eines Wikis kann das vorhandene Wissen verknüpft werden, um daraus neues Wissen aus der Kombination zu schaffen.

### **Wikis in der Software-Entwicklung**

Wikis sind stark im Bereich Software-Entwicklung verbreitet. Viele OpenSource-Projekte (z.B. <http://www.opensuse.org>, <http://www.ubuntu.org>) aber auch kommerzielle Anbieter (z.B. <http://wiki.novell.com>) nutzen Wikis zur Planung von Features, zur Projektdokumentation, für Anwenderdokumentationen und zur Interaktion (Fehlerberichte, Support etc.) mit den Benutzern.

Panagiotis Louridas führt in [Louridas2006] aus, dass sich Wikis bei der Software-Entwicklung vor allem für Teams, die über verschiedene Orte verteilt sind zur Organisation, Planung und zur Veröffentlichung der Arbeiten eignen.

## 3. Worksphere

Die *Worksphere OpenSource* (kurz *Worksphere*) ist ein Framework zur Erstellung von webbasierten Kommunikationsplattformen. Der Begriff Framework soll ausdrücken, dass es sich bei der Worksphere nicht um eine vordefinierte webbasierte Plattform handelt, die in engen Grenzen über Administrationswerkzeuge angepasst werden kann, sondern um ein aus Modulen bestehendes System: ein Baukasten für Kommunikationsplattformen. Mit den Modulen der Worksphere ist es möglich, eine Plattform komplett nach den eigenen Wünschen und Bedürfnissen zusammenzustellen (hierzu sind zum Teil Kenntnisse in der Programmierung mit der Skriptsprache PHP notwendig). Ebenso ist es möglich nur einzelne Teile der Worksphere zu nutzen und diese in andere Anwendungen einzubinden. Die Worksphere bietet Basismodule für Datenbankzugriffe, Sitzungsverwaltung, Authentifikation und Rechteverwaltung sowie Komponenten, die verschiedene Varianten computervermittelter Kommunikation realisieren: Schwarze Bretter, Foren, Chats, Sofortnachrichten, Dateibereiche und Newsticker. Ergänzt werden diese Module durch einen Administrationsbereich, mit dem sie bequem konfiguriert werden können. Die Worksphere wird in einer Standardkonfiguration ausgeliefert, die die einzelnen Teile beispielhaft zu einer sinnvollen Konfiguration zusammenfasst und als Grundlage eigener Anpassungen dienen kann (siehe Abbildung 3.1.).

Die derzeit aktuelle Version der Worksphere ist die Version 1.0preX (Stand April 2006). Die Versionsnummer 1.0preX drückt aus, dass es sich um die Vorversion 10 der Version 1.0 der Worksphere handelt. Eine fertige Version 1.0 der Worksphere ist noch nicht verfügbar. Die zum Zeitpunkt des Schreibens dieser Arbeit aktuellste öffentlich verfügbare Version der Worksphere ist 1.0pre4 (Download unter <http://www.worksphere.de>).

Die Entwicklung der Worksphere wurde initiiert durch ein Telelearning Projekt an der Fakultät für Gesundheitswissenschaften der Universität Bielefeld innerhalb des Fernstudiengangs Angewandte Gesundheitswissenschaften (vgl. [Nandelstädt2001]). Die ursprüngliche Planung in diesem Projekt hatte vorgesehen, eine vollkommen neue Plattform zu entwickeln, die speziell auf die bestehenden Anforderungen und Zielvorstellungen des Fernstudiengangs abgestimmt ist. Diese Planung wurde jedoch zugunsten einer

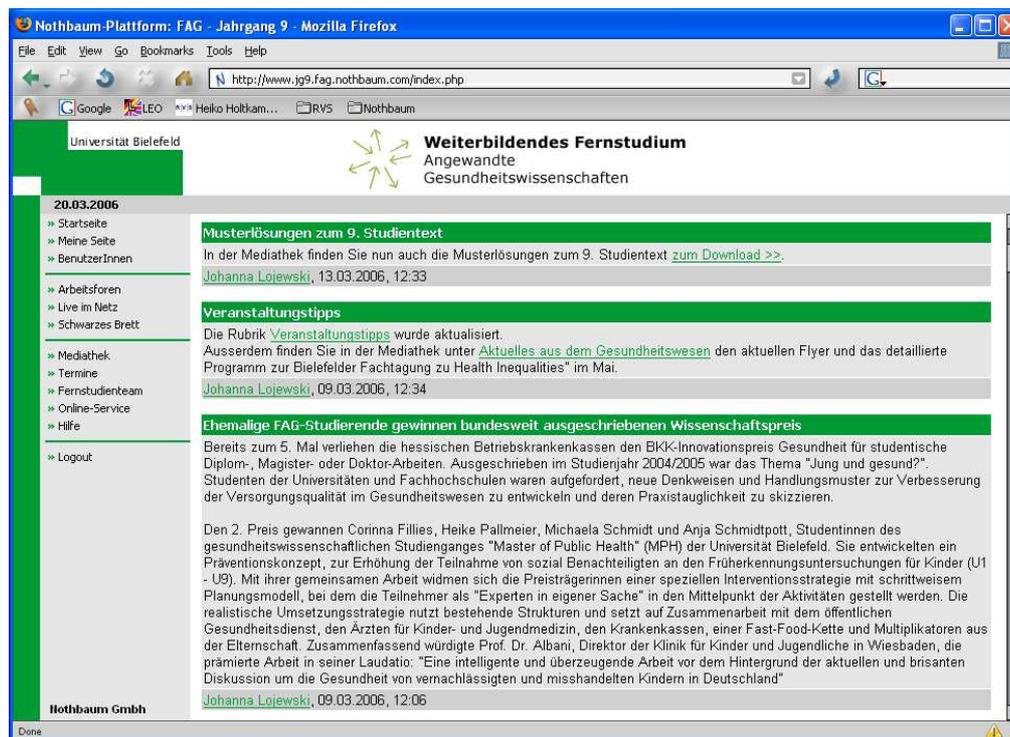


Abbildung 3.1.: Screenshot einer Plattform auf Basis der Standardkonfiguration der Worksphere.

vorhandenen externen Software-Lösung, dem IBT-Server der Firma time4you, aufgegeben [Nandelstädt2001]. Der IBT-Server wurde gemäß den Anforderungen und Zielen des Fernstudiums angepasst. Nach internen Probeläufen im September 2000 ging die auf Basis des IBT-Servers erstellte Kommunikationsplattform des Fernstudiums im Oktober 2000 online [Nandelstädt2001].

Im Laufe der Zeit stellte sich der Ansatz, eine externe Software-Lösung einzusetzen, auf deren Programmierung kein Zugriff besteht, jedoch als zu unflexibel dar. Es war nur schwer möglich, den IBT-Server neuen Anforderungen und Bedürfnissen des Fernstudiums anzupassen. Aus diesem Grund wurde deshalb im Sommer 2001 damit begonnen eine neue Kommunikationsplattform zu entwickeln (siehe [Nothbaum2006]). Die Idee zur Worksphere war geboren. Die Entwicklung wurde von der Nothbaum GmbH, die an der Planung und Konzipierung der Kommunikationsplattform beteiligt war, initiiert. Unterstützt wurde die Entwicklung von der Arbeitsgruppe Rechnernetze und Verteilte Systeme, Technische Fakultät, Universität Bielefeld. Die Implementierung der neuen

Plattform wurde als OpenSource Software<sup>1</sup> konzipiert. Es sollte ein System entstehen, das flexibel einsetzbar ist und den Wünschen der Betreiber individuell angepasst werden kann. Aus diesem Grund sollten die Betreiber einer Plattform auf Basis der Worksphere auch Zugriff auf den Quellcode der Programmierung haben, um eigene Anpassungen und Erweiterungen umsetzen zu können.

## 3.1. Technische Anforderungen

Die Worksphere läuft serverseitig auf jedem Betriebssystem, das die Skriptsprache PHP4 unterstützt. PHP (PHP steht für *PHP Hypertext Preprocessor* (vgl. [Zic2005]))<sup>2</sup> ist eine Skriptsprache, die eine an C und Perl angelehnte Syntax hat und sich vor allem für die Programmierung von Webanwendungen bzw. dynamischen Webseiten eignet. Sie ist leicht zu erlernen, als OpenSource verfügbar, lizenzfrei einzusetzen und für eine Vielzahl unterschiedlicher Betriebssysteme verfügbar, wie z.B. Unix-Varianten, Linux, Windows und Mac OS (vgl. [Welling2001]).

Weitere Grundbausteine bei der Entwicklung bildeten die OpenSource Produkte Linux (Betriebssystem), Apache (Webserver) und MySQL (Relationales Datenbanksystem). Eine Umgebung aus diesen Systemen, in Zusammenspiel mit der Skriptsprache PHP, wird häufig als *LAMP-System* bezeichnet<sup>3</sup>. Die Worksphere kann aber auch auf anderen Betriebssystemen und mit anderen Webservern betrieben werden. Der Wechsel auf ein anderes Datenbanksystem erfordert evtl. Anpassungen.

Auf Seiten der Clients sind die einzigen Anforderungen ein Browser (für eine vollständige Funktionalität der Worksphere mit Unterstützung von Cascading Style Sheets (CSS) und JavaScript) und ein Internetzugang. Damit kann eine Plattform auf Basis der Worksphere auf so gut wie jedem System genutzt werden.

## 3.2. Kernfunktionen der Worksphere

Den Kern der Worksphere bildet eine angepasste und erweiterte Version der PHPLib 7.2d [PHPLib], im Weiteren kurz als PHPLib/w bezeichnet. Die PHPLib ist eine Samm-

---

<sup>1</sup>Siehe auch <http://www.opensource.org>

<sup>2</sup>Für eine ausführliche Darstellung der Skriptsprache PHP sei an dieser Stelle auf entsprechende Literatur verwiesen, z.B. [Krause2001], [Ratschiller2000], [Welling2001] oder die offizielle PHP-Webseite <http://www.php.net>.

<sup>3</sup>Vergleiche <http://www.onlamp.com> und <http://de.wikipedia.org/wiki/LAMP>.

lung von PHP-Skripten, die Funktionen bieten, welche Programmierer beim Entwickeln von Anwendungen in PHP häufig benötigen. Die PHPLib steht unter der GNU Library Public License (LGPL) und kann daher sowohl in nichtkommerziellen als auch in kommerziellen Projekten kostenfrei eingesetzt werden (Einzelheiten zur LGPL sind im Download-Paket der PHPLib enthalten [PHPLib] sowie unter <http://www.gnu.org>). Die ersten Versionen der PHPLib wurden von Boris Erdmann und Kristian Köhntopp entwickelt [Köhntopp1999]. Bei weiteren Versionen der PHPLib kamen neue Entwickler hinzu, unter ihnen sind vor allem Sascha Schuhmann und Jay Bloodworth zu erwähnen, die wesentlichen Anteil an den Funktionen zur Sitzungsverwaltung und anderen Klassen hatten [Köhntopp1999]. Die Weiterentwicklung der PHPLib ist seit einiger Zeit eingestellt und Teile der PHPLib sind im PEAR-Projekt aufgegangen (PHP Extension and Application Repository, <http://pear.php.net>). Im Folgenden wird die Funktionalität der PHPLib/w umrissen, da sie die Grundlage der Programmierung von Modulen für die Worksphere bildet. Für eine ausführliche Dokumentation sei auf [Köhntopp2002], [Köhntopp1999] und [Ratschiller2000] verwiesen.

### 3.2.1. Konfigurationsdateien

Die PHPLib/w hat zwei Dateien<sup>4</sup>, die Konfigurationsinformationen enthalten und die für jede Installation der Worksphere angepasst werden müssen: `prepend.php` und `local.inc`.

Die Datei `prepend.php` lädt die Skripte mit den Definitionen der Klassen der PHPLib/w, die in den verschiedenen Skripten verfügbar sein sollen. Bei den Skripten der Worksphere geschieht die Einbindung der Datei `prepend.php` über die Anweisung `include ("prepend.php")`.

Zur Anpassung der Basisklassen der PHPLib/w werden in der Datei `local.inc` eigene Klassen, die auf die Umgebung abgestimmt sind, abgeleitet.

### 3.2.2. Datenbank-Abstraktion

Die PHPLib/w stellt mit der Klasse `DB_Sql` ein API mit Funktionen zur Verfügung, die den Zugriff auf Datenbanken weitestgehend unabhängig vom zugrunde liegenden relationalen Datenbankmanagementsystem (RDBMS) erlauben. Durch das Ändern des sog. Backends der Datenbank-Abstraktionsschicht ist es leicht möglich, das RDBMS zu

---

<sup>4</sup>Die Dateien befinden sich im Unterverzeichnis `/conf` einer Worksphere-Installation

wechseln, ohne die Datenbankzugriffe in den einzelnen Programmteilen zu ändern, die die Funktionen der Abstraktionsschicht verwenden<sup>5</sup>. Die Worksphere verwendet bisher nur das Backend für MySQL-Datenbanken und wurde nicht mit anderen RDBMS getestet (abgesehen von Tests bei einer sehr frühen Version mit Postgres).

Die Verwendung der Datenbankabstraktion lässt sich am einfachsten an einem kurzen Beispiel veranschaulichen:

```
1  <?php

    // Datei mit Voreinstellungen fuer die Worksphere laden
    include( "prepend.php" );

6  // Neues Datenbank-Objekt erzeugen
    $db = new Worksphere_DB;

    // Verbindung zur Datenbank herstellen
    $db->connect( "worksphere", "localhost", "db_user", "
        db_passwd" );

11

    // SQL-Statement erstellen
    $sql = "SELECT * FROM test_table";

    // Query ausfuehren
16  $db->query($sql);

    // Ergebnis mit einer Schleife durchlaufen und ausgeben
    while ( $db->next_record() )
    {
21        $spalte1 = $db->f("col1");
        $spalte2 = $db->f("col2");
        $spalte3 = $db->f("col3");

        echo "<p>$spalte1 | $spalte2 | $spalte3</p>";
26    }

    ?>
```

Listing 3.1: Ein Beispiel für die Verwendung von DB\_Sql.

---

<sup>5</sup>Selbst mit der PHPLib/w ist die Portierbarkeit zwischen einzelnen RDBMS aber nur theoretischer Natur, wenn nicht von vornherein bei der Programmentwicklung auch portierbarer Datenbankcode in Form von SQL-Anweisungen geschrieben wird. Werden hier datenbankspezifische Funktionen verwendet, müssen diese bei der Portierung auf ein neues RDBMS umgeschrieben werden (Zur Standardisierung von SQL und den Unterschieden verschiedener SQL-Dialekte siehe auch [Elmasri2002]).

Die erste Zeile erzeugt ein neues Datenbankobjekt. Als nächstes wird eine Verbindung zur Datenbank `worksphere` auf `localhost` mit dem Benutzernamen `db_user` und dem Kennwort `db_passwd` hergestellt. Diese Parameter können auch direkt über die entsprechenden Konfigurationsvariablen in der Datei `prepend.php` gesetzt werden (dies ist die bei der PHPLib/w bevorzugte Variante):

```
$_WS_CONF["DB_HOST"]      = "localhost";
$_WS_CONF["DB_DATABASE"] = "worksphere";
$_WS_CONF["DB_USER"]     = "db_user";
$_WS_CONF["DB_PASSWORD"] = "db_passwd";
```

Mit dem Aufruf von `query()` wird eine Anfrage an die Datenbank abgesendet. Die Methode kümmert sich um alle relevanten Funktionsaufrufe zur Verarbeitung und zum Senden der Abfrage, die je nach Datenbanksystem unterschiedlich sind. Konnte die Abfrage erfolgreich ausgeführt werden, steht das Ergebnis bereit und kann innerhalb eines Skriptes weiter verarbeitet werden.

Die Methode `next_record()` liefert einen Boolean-Wert, ob ein Datensatz gelesen werden konnte oder nicht und stellt ggf. den gelesenen Datensatz in einer Ergebniszeile zur Verfügung. Auf die einzelnen Einträge der Ergebniszeile (die die Namen der Attribute der Relation haben) kann über die Methode `f()` zugegriffen werden. Im Beispiel werden so die Werte der Attribute `col1`, `col2` und `col3` aller gelesenen Datensätze in einer Schleife ausgegeben.

### 3.2.3. Sitzungsverwaltung (Sessions)

Die Klasse zur Sitzungsverwaltung ist der eigentliche Grund, warum die PHPLib von Kristian Köhntopp und Boris Erdmann entwickelt wurde. Ursprünglich wurde die PHPLib für PHP3 geschrieben, um Variablen und deren Werte über den Aufruf mehrerer Webseiten hinweg gültig und verfügbar zu machen<sup>6</sup> [Köhntopp1999].

#### **Exkurs: HTTP und Cookies**

Das Hypertext Transfer Protocol (HTTP)<sup>7</sup> ist ein zustandsloses Protokoll (vgl. [Kurose2002] und [Tanenbaum2003]). Verschiedene Anfragen

---

<sup>6</sup>Seit PHP4 ist die Sitzungsverwaltung Bestandteil von PHP

<sup>7</sup>Version 1.0 von HTTP ist definiert in RFC1945, Version 1.1 in RFC2616 [RFC]

eines gleichen Clients werden nicht miteinander in Zusammenhang gebracht. Wenn ein Benutzer eine Webseite über seinen Browser aufruft, sendet der Browser eine HTTP-Anfrage (HTTP request) an einen Webserver, der mit einer HTTP-Antwort (HTTP response) auf die Anfrage des Browsers reagiert. Danach „vergisst“ der Server, dass er den Client „jemals“ gesehen hat.

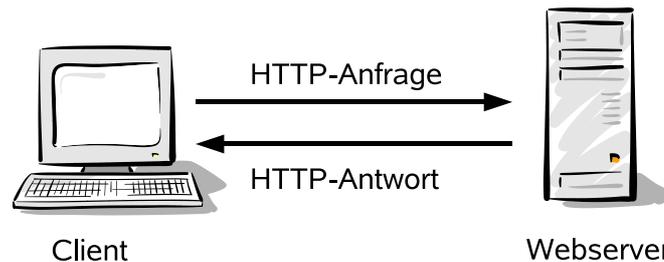


Abbildung 3.2.: HTTP-Anfrage/-Antwort.

Oft ist es aber gewünscht, den Zustand zwischen Anfragen festzuhalten (z.B. bei Benutzeranmeldungen auf einer Webseite). HTTP bietet einen Mechanismus zur Identifizierung von Clients. Die Technik hierfür, die als *Cookies* bekannt ist, wurde von Netscape eingeführt [Tanenbaum2003] und ist in RFC2109 [RFC] spezifiziert.

Cookies sind allerdings eine viel kritisierte und kontrovers diskutierte Technik, da es damit auch möglich ist Web-Nutzungsgewohnheiten eines Benutzers zusammenzustellen. Auf diese Diskussion, auch in Zusammenhang mit möglichem Missbrauch von Cookies durch Sicherheitslücken soll im Rahmen dieser kurzen Darstellung nicht weiter eingegangen werden. Informationen hierzu sind z.B. auf den Seiten der Cookie Central zu finden<sup>8</sup>.

Wenn ein Client eine Webseite anfordert, die Cookies einsetzt, antwortet der Server mit einer HTTP-Antwort, deren Header die Zeile **Set-Cookie:** beinhaltet (siehe Abbildung 3.3). Der Wert des so vom Server übermittelten Cookie ist oft eine Identifikationsnummer (z.B. **Set-Cookie: 12345**). Empfängt der Client die Antwortnachricht, erkennt er die Set-Cookie-Zeile und kann den Wert bzw. Inhalt des Cookies speichern, sofern der Client Cookies akzeptiert. Zusammen mit dem Wert des Cookies werden üblicherweise die Domäne des Cookie-sendenden Servers und eine Gültigkeitsdauer für den

<sup>8</sup><http://www.cookiecentral.com> [Kurose2002]

Cookie abgespeichert. Ist keine Gültigkeitsdauer angegeben, bedeutet dies, dass das Cookie nur so lange leben soll, wie die Browser-Instanz läuft, ein sog. nicht-persistentes Cookie. In späteren Anfragen an den gleichen Server bindet der Client die Header-Zeile **Cookie**: zusammen mit dem gespeicherten Wert des Cookie in den HTTP-Anfrage-Header ein (z.B. **Cookie: 12345**). Auf diese Weise kann der Server den Client wiedererkennen.

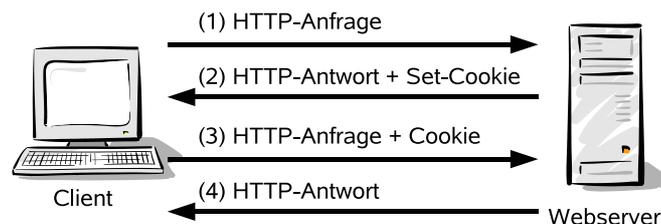


Abbildung 3.3.: Typischer Cookie-Austausch.

Cookies bieten eine einfache Möglichkeit, den Zustand von Anfragen zu speichern. Sie haben allerdings den Nachteil, dass sie a) auf Client-Seite gespeichert werden, also keine Kontrolle auf Seiten des Servers über die Daten besteht und b), dass sie nur eine begrenzte Größe für zu speichernde Daten haben. Es ist deshalb wünschenswert, die Zustandsinformationen auf Serverseite verwalten zu können. Genau dies erlauben *Sessions* (*Sitzungen*) oder genauer *Session-Objekte*. Sessions sind serverseitige Sammlungen von Variablen, die zusammen einen Zustand repräsentieren. Über eine sog. *Session-ID* wird eine Datensammlung auf dem Server identifiziert (vgl. Abbildung 3.4). Die Session-ID wird dem Client als Cookie übertragen.

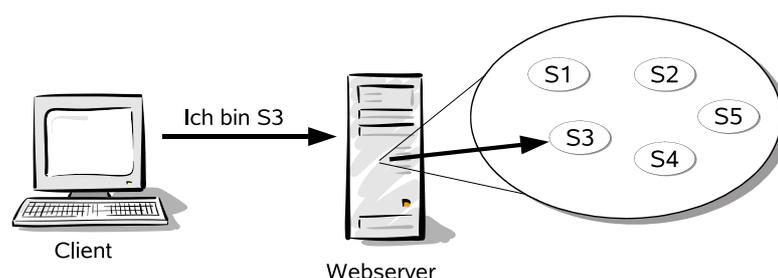


Abbildung 3.4.: Session-Objekte können als Datensammlungen auf dem Webserver betrachtet werden.

In der Standardeinstellung arbeitet die Sitzungsverwaltung der PHPLib/w mit Cookies. Es ist jedoch möglich dieses Verhalten auf die sog. GET-Methode umzustellen. Dabei

wird die Session-ID als Teil der URL<sup>9</sup> kodiert. Ebenso bietet die PHPLib/w einen Rückfallmodus. Ist die Klassenvariable `$fallback_mode` der Klasse `Session` bzw. der sie erweiternden Klasse `WorkspHERE_Session` auf `get` gesetzt, wird der GET-Mode verwendet, falls Cookies vom anfragenden Client nicht unterstützt werden. Die Nutzung der GET-Methode wird allerdings nicht empfohlen, da sie ein Sicherheitsrisiko darstellt. Die Daten werden direkt sichtbar mit der URL übertragen und können auf diese Weise auch in den Logs von Webservern gespeichert werden oder im Referer-Header<sup>10</sup> auftauchen (vgl. dazu [Huseby2004]).

Die PHPLib/w verwendet zur Speicherung der Session-Daten sogenannte *Container*. Die Container-Klassen beginnen mit dem Präfix `CT_`. Zur Auswahl stehen `CT_Sql` (Speicherung in einer Datenbank), `CT_File` (Speicherung im Dateisystem) und `CT_Shm` (Shared Memory) [Köhntopp1999]. Der Standard-Container ist `CT_Sql` (die Sessin-Daten werden intern in der Relation `active_sessions` gespeichert. Bei der Verwendung von `CT_Sql` ergibt sich das in Abbildung 3.5 dargestellte Klassensystem.

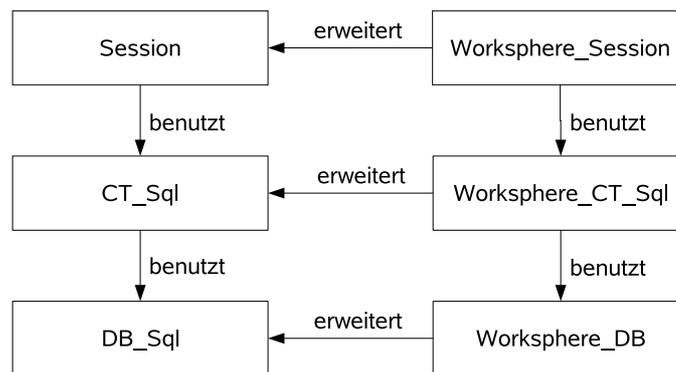


Abbildung 3.5.: Klassenschema `DB_Sql`, `CT_Sql` und `Session` bzw. deren erweiterter Klassen (nach [Köhntopp2002]).

Die Nutzung der Sessionverwaltung lässt sich wiederum am einfachsten an einem Beispiel erläutern:

```

1  <?php

    // Datei mit Voreinstellungen fuer die WorkspHERE laden
  
```

<sup>9</sup>URLs (Uniform Resource Locators) sind eine Teilmenge von URIs (Uniform Resource Identifier). In RFCs und offiziellen Dokumenten wird deshalb normalerweise von URIs gesprochen (vgl. RFC1630 und RFC2396). Der Begriff URL ist allerdings bekannter und wird deshalb hier verwendet.

<sup>10</sup>Die meisten Browser senden den Referer-Header im HTTP-Header mit. Er enthält die URL des Dokuments, von dem die Anfrage ausging.

```
include("prepend.php");

6 // Aufruf von page_open mit der Session-Funktionalität
  // => Laden der ggf. bisher gespeicherten Session-Variablen
  // aus dem Container
  page_open("array("sess" => "Worksphere_Session");

11 if ( !isset($s) ) { $s = 0; }

  // Die globale Variable $s bei der Session registrieren
  // (ohne(!) vorangestelltes $-Zeichen)
  $sess->register("s");

16 echo "s hat den Wert $s\n";

  $s++;

21 // Zurueckspeichern der registrierten Session-Variablen in
  // den Container
  page_close();

?>
```

Listing 3.2: Ein Beispiel für die Verwendung der Session-Funktionalität.

Ein PHP-Skript, das die Sessions der PHPLib/w nutzt, hat mindestens zwei Funktionsaufrufe: `page_open()` (am Anfang des Skripts) `page_close()` (am Ende des Skripts). Beim Aufruf von `page_open()` wird mitgeteilt, dass die Seite die Session-Funktionalität verwenden soll, so wie sie in der erweiterten Klasse `Worksphere_Session` konfiguriert ist. Die beiden `page`-Funktionen werden in der PHPLib-Dokumentation [Köhntopp2002] auch als Seitenverwaltungsfunktionen bezeichnet. Zu beachten ist, dass vor dem Aufruf von `page_open()` keine Ausgaben im Skript gemacht werden dürfen, da die Funktion den Cookie-Header und weitere Header setzt. Die Funktion `page_close()` sollte immer am Ende eines Skriptes aufgerufen werden, damit die registrierten Session-Variablen im Container gespeichert - persistent gemacht - werden. Mit dem Aufruf der Methode `register()` in Zeile 15 des Beispielskripts wird die Variable `$s` bei der Session angemeldet. Von nun an wird der aktuelle Wert der Variablen `$s` beim Aufruf von `page_close()` gespeichert und bei einem Aufruf von `page_open()` wieder hergestellt. Mit dem Aufruf der Methode `unregister()` kann eine Variable von der Session abgemeldet werden.

### 3.2.4. Authentifikation

Mit der Klasse Auth bzw. `WorkspHERE_Auth` wird ein einfacher Mechanismus zur Authentifikation bereitgestellt: Benutzer melden sich mit einem Benutzernamen und Kennwort an<sup>11</sup>. Die Klasse bietet Methoden, mit denen auf jeder Seite bzw. in jedem Skript geprüft werden kann, ob sich ein Benutzer bereits angemeldet hat. Ist dies der Fall, wird der Inhalt der Seite dargestellt, andernfalls ein frei konfigurierbares Login-Formular (siehe Abbildung 3.6). Ob sich ein Benutzer bereits angemeldet hat wird über die Sitzungsverwaltung der PHPLib/w festgehalten.

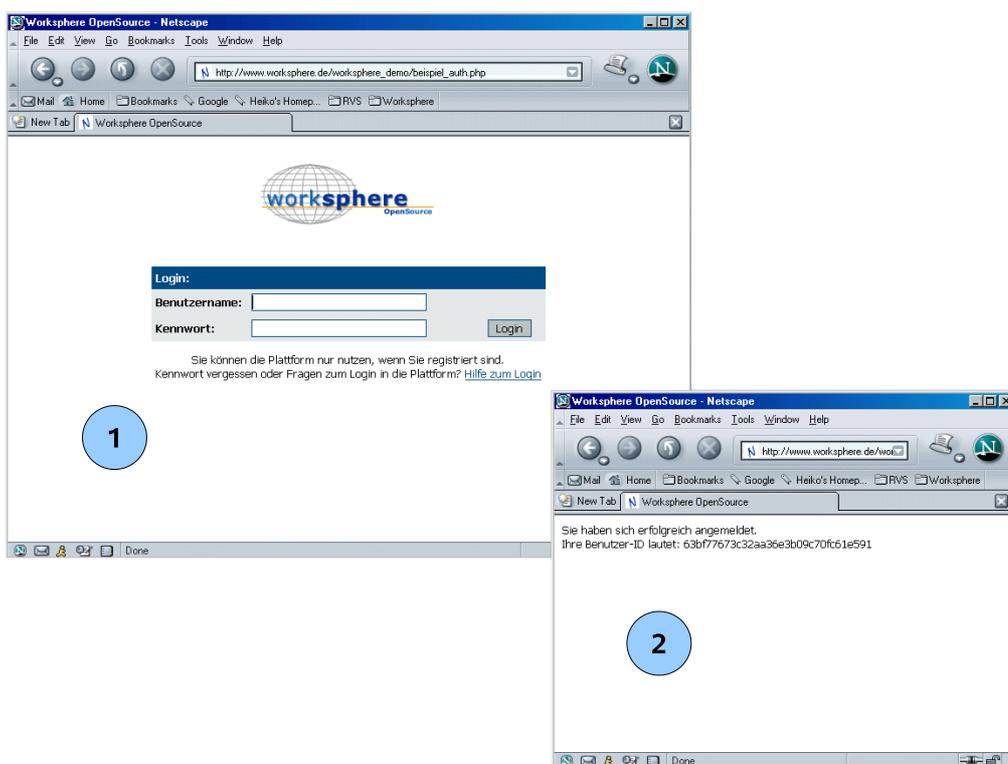


Abbildung 3.6.: Beispiel für das frei konfigurierbare Login-Formular (1), Anzeige des Seiteninhalts nach erfolgreichem Login (2).

Alle Seiten, die die Authentifikation der PHPLib/w verwenden, haben die im Beispieldokument dargestellte allgemeine Struktur:

```
<?php
```

<sup>11</sup>Zur Problematik dieses einfachen Ansatzes siehe auch [Schneier2001], Kapitel 9: Identifizierung und Authentifikation

```
// Datei mit Voreinstellungen fuer die Worksphere laden
include("prepend.php");
5
// Aufruf von page_open mit der Session- und
// und der Authentifikationsfunktionalitaet
page_open("array("sess" => "Worksphere_Session",
                "auth" => "Worksphere_Auth"));
10

echo "Sie haben sich erfolgreich angemeldet.\n";
echo "Ihre Benutzer-ID lautet: ".$auth->auth["uid"]."\n";

15 page_close();

?>
```

Listing 3.3: Ein Beispiel für die Verwendung der Authentifikation.

Intern arbeitet die Authentifikation mit User-IDs. User-IDs haben ein festgelegtes Format, sie werden beim Anlegen eines neuen Benutzers über das Administrationstool der Worksphere mit den PHP-Funktionen `uniqid()`<sup>12</sup> und `md5()`<sup>13</sup> erzeugt: `$uid=md5(uniqid())`.

Mit der Klassenvariablen `$lifetime` kann die Gültigkeitsdauer einer Anmeldung in Minuten festgelegt werden. Ist ein Benutzer für eine längere Zeit als die festgelegte Zeit inaktiv<sup>14</sup>, muss er sich erneut authentifizieren.

Gegenüber der PHPLib ist die PHPLib/w um die Protokollierung von Zugriffsdaten der Benutzer in einer Log-Tabelle (**logs**) ergänzt. Es wird protokolliert, welche Seiten der Worksphere der Benutzer mit welchen Parametern, zu welchem Zeitpunkt von welchem Host (bzw. mit welcher IP-Adresse) aus abgerufen hat. Optional kann beim Aufruf der Funktion `page_open()` ein Kommentar zum Seitenaufruf in der Log-Tabelle gespeichert werden:

---

<sup>12</sup>Die Funktion `uniqid()` erzeugt eine ID aus der Systemzeit in Millisekunden; mit dem optionalen Parameter `prefix` kann ein Präfix für die ID festgelegt werden (vgl. [Krause2001]).

<sup>13</sup>Die Funktion `md5()` einen Hash-Code aus dem übergebenen Parameter nach dem MD5 Message-Digest Algorithmus (vgl. [Krause2001]).

<sup>14</sup>Aktivität bzw. Inaktivität bezieht sich auf die Interaktion des Benutzers mit dem Server. Dazu zählen Anfragen nach einer neuen Seite, das Speichern von Daten etc. Aktivitäten des Benutzers, die nur auf dem Client-System ausgeführt werden, z.B. das Schreiben eines Beitrags in ein Formularfeld, können nicht als Aktivität gewertet werden, da der Server hiervon nichts mitbekommt.

```
page_open("array("sess" => "Worksphere_Session",  
                "auth" => "Worksphere_Auth",  
                "page" => "Kommentar zur Seite"));
```

Die Log-Tabelle liefert wertvolle Informationen, falls Unregelmäßigkeiten festgestellt werden oder Fehler nach verfolgt werden müssen. Es kann gut eingegrenzt werden, welcher Benutzer bzw. welcher Account für die Unregelmäßigkeiten verantwortlich ist bzw. bei welcher Folge von Seitenaufrufen der Fehler aufgetreten ist.

### 3.2.5. Zugriffsrechte (Permissions)

Mit den Methoden der Klasse `Perm` bzw. der sie erweiternden Klasse `Worksphere_Perm` können Zugriffsrechte für einzelne Benutzer verwaltet werden. Die `PHPLib/w` verwaltet die Zugriffsrechte in der Relation `permissions` in einer Datenbank, statt wie die `PHP-Lib` die Zugriffsrechte fest in einem Array in der Klasse zu kodieren. Es muss hier auch anstatt von Zugriffsrechten von *Gruppen* oder *Rollen* gesprochen werden. Die Zugehörigkeit eines Benutzers zu einer Gruppe bzw. das Innehaben einer Rolle führt dazu, dass der Benutzer für ein Objekt (z.B. ein Forum in der Worksphere) ein bestimmtes Zugriffsrecht hat (z.B. Lesen, Schreiben). Diese Art der Zugriffskontrolle wird auch als *Role-based Access Control (RBAC)* bezeichnet.

„*Role-based access control (RBAC)* – Rather than to subjects, permissions are assigned to roles. A subject always acts according to the currently delegated role and therefore acquires the appropriate permissions relevant to the current role. The subject can hold different permissions to objects depending on the role assigned to it.“ ([Pieprzyk2003], Kapitel 17, S. 566)

„The RBAC model streamlines the access control by first defining roles which are given access permissions to objects and later assigning roles to subjects.“ ([Pieprzyk2003], Kapitel 17, S. 576f.)

Die Gruppenzugehörigkeiten eines Benutzers werden in der Relation `permissions` gespeichert. Ein Benutzer kann beliebig viele Gruppen zugewiesen bekommen. Ein bestimmter Zugriff eines Benutzers auf ein Objekt ist dann erlaubt, wenn der Benutzer zu einer Gruppe gehört, die das Recht eines solchen Zugriffs auf das Objekt hat (vgl. Abbildung 3.7). Um die Klasse für Zugriffsrechte zu nutzen, muss die Funktion `page_open()`

mit einem weiteren Element aufgerufen werden (siehe Beispielskript 3.4, Zeilen 11-13).

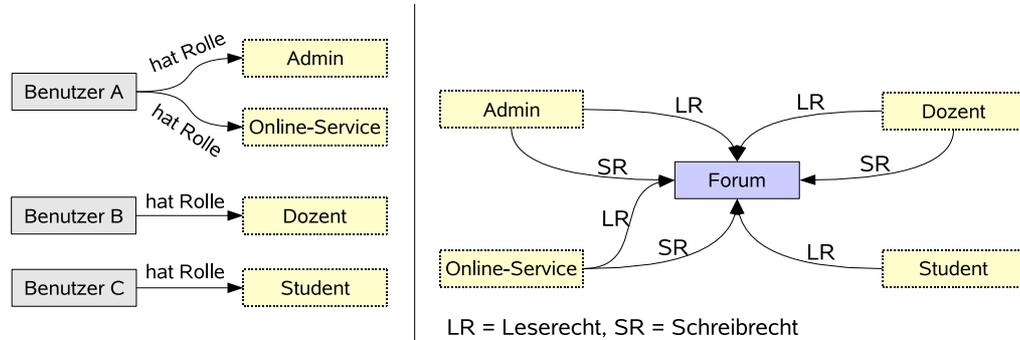


Abbildung 3.7.: Modell der Zugriffskontrolle der PHPLib/w. Linke Seite: Zuordnung von Rollen zu Benutzern. Rechte Seite: Zuordnung von Berechtigungen zu Gruppen. In dem Beispiel haben Benutzer A und Benutzer B durch ihre Gruppenzugehörigkeit Lese- und Schreibrechte in dem Forum; Benutzer C hat nur Leserechte in dem Forum.

Die Klasse `Worksphere_Perm` bietet für diese Überprüfung die Methoden `have_dperm()` und `have_dpermid()` (Skript 3.4, Zeile 30). Das folgende Beispielskript veranschaulicht die Verwendung der Klasse für Zugriffsrechte:

```

<?php
2
// Die Foren-ID - $f_ID - wird dem Skript als URL-Parameter
// uebergeben

// Datei mit Voreinstellungen fuer die Worksphere laden
7 include ("prepend.php");

// Aufruf von page_open mit der Session-, der
// Authentifikations- und der ZugriffsrechtEFunktionalitaet
page_open("array("sess" => "Worksphere_Session",
12           "auth" => "Worksphere_Auth",
           "perm" => "Worksphere_Perm"));

$db = new Worksphere_DB;

17 // Testen, ob das Recht zum Lesen im Forum vorhanden ist
    $leserecht = false;

```

```
// Eintraege der Tabelle foren_rechte_lesen auslesen, die
// die IDs der Gruppen speichert, die Leserecht fuer das
22 // Forum haben
$query = "SELECT lesen FROM foren_rechte_lesen WHERE ID = '
    $f_ID'";
$db->query($query);

while ( $db->next_record() )
27 {
    // Testen, ob der angemeldete Benutzer in der
    // aus der Tabelle gelesenen Gruppe ist
    if ( $perm->have_dpermid($db->f("lesen")))
    {
32         $leserecht = true;
    }
}

if ( $leserecht )
37 {
    // Code zum Anzeigen der Formulareintraege

    (...)
}
42 else
{
    echo "<P>Sie haben keine Berechtigung Eintr&auml;ge
        in diesem Forum zu lesen.</P>";
}

47 page_close();
?>
```

Listing 3.4: Ein Beispiel für die Verwendung der Zugriffsrechte.

### 3.3. Kommunikationsfunktionen der Worksphere

Die Worksphere bietet verschiedene Kommunikationsmöglichkeiten, die es den Benutzern erlauben, sich untereinander auszutauschen. Sie geben auch die Möglichkeit, Dozenten und Experten zu bestimmten Themen zu befragen und Meinungen einzuholen. Die Kommunikationsfunktionen der Worksphere sind als Module realisiert, die sich beliebig miteinander kombinieren lassen und frei eingesetzt werden können. Auf diese Wei-

se kann eine Plattform erstellt werden, die sich den individuellen Anforderungen eines Kurses genau anpassen lässt. Um eine Plattform auf Basis der Worksphere zusammen zu stellen, sind Programmierkenntnisse (in PHP) erforderlich. Die Verwaltung und Konfiguration der einzelnen Module wird durch ein Administrationswerkzeug unterstützt (siehe Abbildung 3.8). Im Weiteren werden einige der Kommunikationsfunktionen der Worksphere kurz dargestellt, ohne dabei auf technische Details einzugehen.

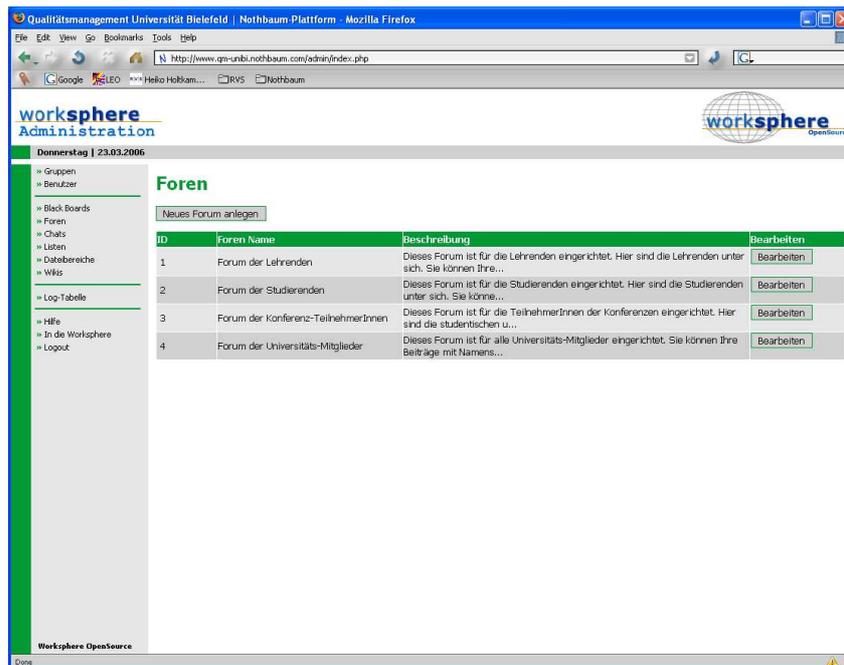


Abbildung 3.8.: Screenshot des Administrationstools (Dargestellt ist die Liste aller eingerichteten Foren).

### 3.3.1. Foren

Die Foren der Worksphere sind sogenannte digitale Foren (oft auch als *Newsgroups* bezeichnet), in denen Benutzer eigene Nachrichten veröffentlichen und andere Nachrichten beantworten können. Die Nachrichten in einem Forum werden in einer hierarchischen Struktur dargestellt, die den Verlauf einer Diskussion wieder spiegelt. Es lassen sich sehr schnell Diskussionsbeiträge sowie die zugehörigen Antworten und Folgebeiträge finden und lesen. So kann eine Diskussion im Forum leicht nachvollzogen werden. Neue Beiträge werden in einem Forum besonders hervorgehoben. Der Benutzer kann einen Zeitraum

in Tagen festlegen, in dem neue Beiträge kenntlich gemacht werden sollen. Die Startseite

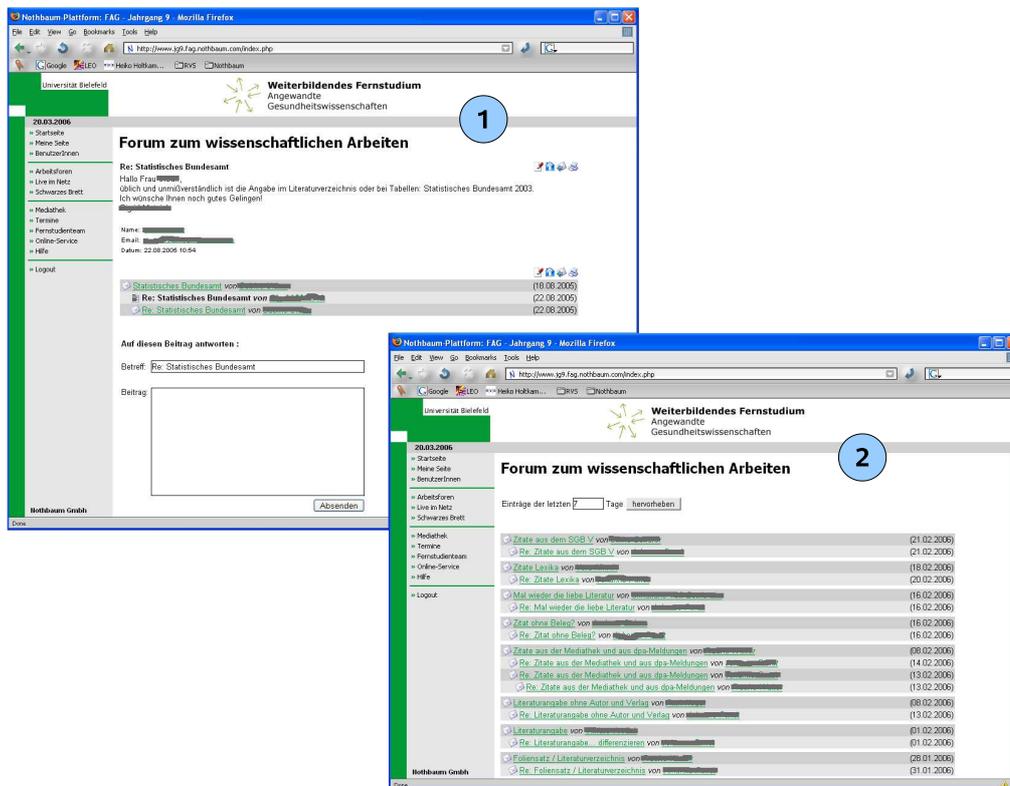


Abbildung 3.9.: Screenshot eines Forums in der Worksphere. Startseite des Forums (2), Nachrichtenanzeige im Forum (1).

eines Forums präsentiert dem Benutzer eine Übersicht aller Diskussionen in dem Forum (vgl. Abbildung 3.9). Wählt der Benutzer eine bestimmte Nachricht aus, so wird der vollständige Text dieser Mitteilung angezeigt sowie der Diskussionsstrang (auch *thread* genannt), zu dem sie gehört (siehe Abbildung 3.9).

Über das Administrationstool der Worksphere können beliebig viele Foren eingerichtet werden. Für jedes Forum kann eingestellt werden, welche Gruppen in der Worksphere Lese- und/oder Schreibzugriff auf das Forum haben.

### 3.3.2. Chats

Die Worksphere bietet ein einfaches Chat-Modul. Ein Chat ist eine synchrone und textgebundene Kommunikationsmöglichkeit. Chats ermöglichen es, in Echtzeit mit anderen

Chat-Teilnehmern zu kommunizieren. Der Begriff „Chat“ (engl. für „Plausch“, „Plaudelei“, „Schwätzchen“) macht deutlich, dass diese computervermittelte Kommunikationsform dem mündlichen Gespräch zugeordnet wird, auch wenn die Teilnehmer eines Chats dabei nicht sprechen, sondern tippen [e-teaching2006]. Neben dem eigentlichen Nach-

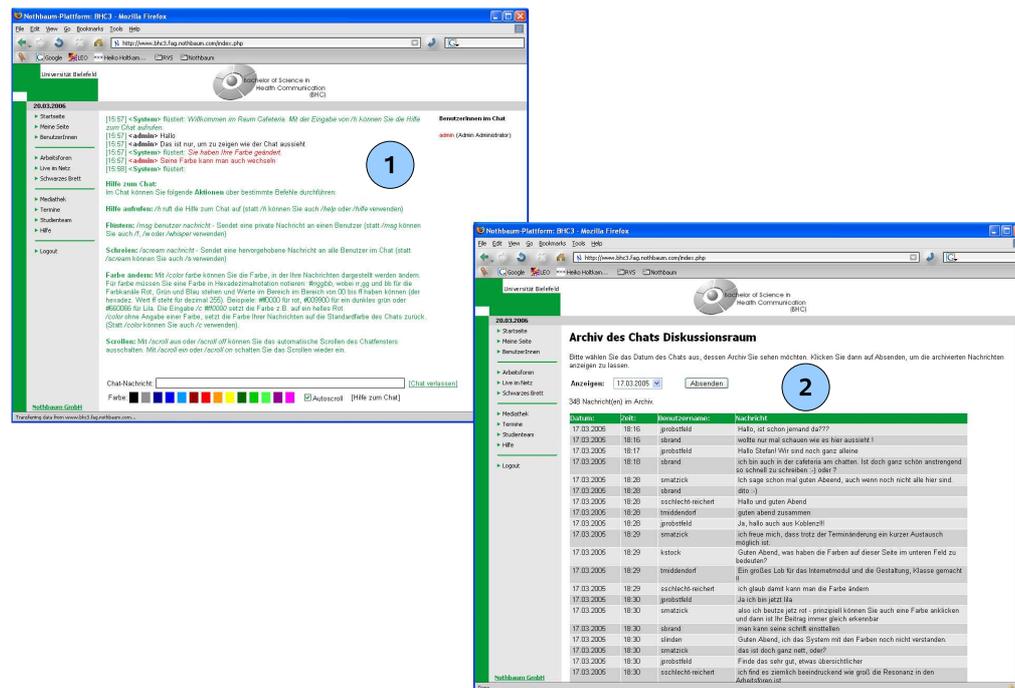


Abbildung 3.10.: Screenshot eines Chat-Raums (1) und eines Chat-Archivs (2) in der Worksphere.

richtenfenster, in dem die Eingaben der Benutzer erscheinen, wird am rechten Rand eine Benutzerliste angezeigt, die alle Benutzer führt, die im Chat-Raum anwesend sind (siehe Abbildung 3.10). Teilnehmer haben die Möglichkeit, die Farbe des Textes zu ändern, in dem ihre Mitteilungen dargestellt werden. Das automatische Scrollen des Nachrichtenfensters kann abgeschaltet werden, um längere Nachrichten und Diskussionen in Ruhe verfolgen zu können. Für private Nachrichten zwischen zwei Benutzern, die nicht in der öffentlichen Diskussion erscheinen sollen, steht eine Flüsterfunktion zur Verfügung. Beim Anlegen eines Chats kann eingestellt werden, ob die geführten Diskussionen archiviert werden. Das Chat-Archiv erlaubt es, die Inhalte vergangener Chats zu einem späteren Zeitpunkt nachzulesen (siehe Abbildung 3.10).

### 3.3.3. Schwarze Bretter

Die Schwarzen Bretter in der Worksphere sind einfache Nachrichtenbretter (*Bulletin Boards*). Je nach Einsatz haben alle Nutzer oder nur bestimmte Gruppen, die Möglichkeit Aushänge in Textform auf einem Schwarzen Brett zu machen oder zu lesen. Schwarze Bretter können auch als Newsticker genutzt werden, um die Benutzer einer Plattform über Neuigkeiten zu informieren. Die Administratoren haben beim Anlegen eines Schwarzen Bretts vielfältige Konfigurationsmöglichkeiten. Es kann festgelegt werden, welche Gruppen das Recht zum Lesen und/oder Schreiben von Beiträgen haben, nach welcher Sortierreihenfolge Beiträge angezeigt werden, ob anonyme Beiträge erlaubt sind etc.



Abbildung 3.11.: Screenshot eines Schwarzen Bretts in der Worksphere.

### 3.3.4. Dateibereiche

Dateibereiche sind ein webbasiertes Dateiarchiv innerhalb der Worksphere. Benutzer können Dateien speichern und laden. Jede Datei kann mit einer Kurzbeschreibung und einem ausführlichen Kommentar versehen werden (vgl. Abbildung 3.12). So können z.B. unterschiedliche Versionen einer Datei genauer beschrieben und der Arbeitsstand erläu-

tert werden. Mit dem Rechtesystem der Worksphere ist es möglich, genau festzulegen, welche Benutzergruppen Dateien eines Dateibereichs lesen dürfen oder Dateien hinein- stellen dürfen.



Abbildung 3.12.: Screenshot eines Dateibereichs in der Worksphere.

### 3.3.5. Benutzerliste

Die Benutzerliste liefert Informationen über die Nutzer der Plattform. Name, Benutzername, E-Mailadresse und der Status, ob der Benutzer aktuell in der Plattform eingeloggt ist oder nicht, werden in einer alphabetisch nach dem Nachnamen sortierten Tabelle dargestellt (siehe Abbildung 3.13 (1)). Die Namen sind verlinkt auf eine Informationsseite des Nutzers.

Benutzer haben mit der Informationsseite die Möglichkeit eine Selbstbeschreibung sowie private und berufliche Adress- und Kontaktdaten zu veröffentlichen (vgl. Abbildung 3.13 (2)). Daneben können die Nutzer auch ein Foto in die Plattform stellen. All diese Angaben stellt der Benutzer freiwillig zur Verfügung, es werden nur die Informationen angezeigt, die der Teilnehmer eingepflegt hat.

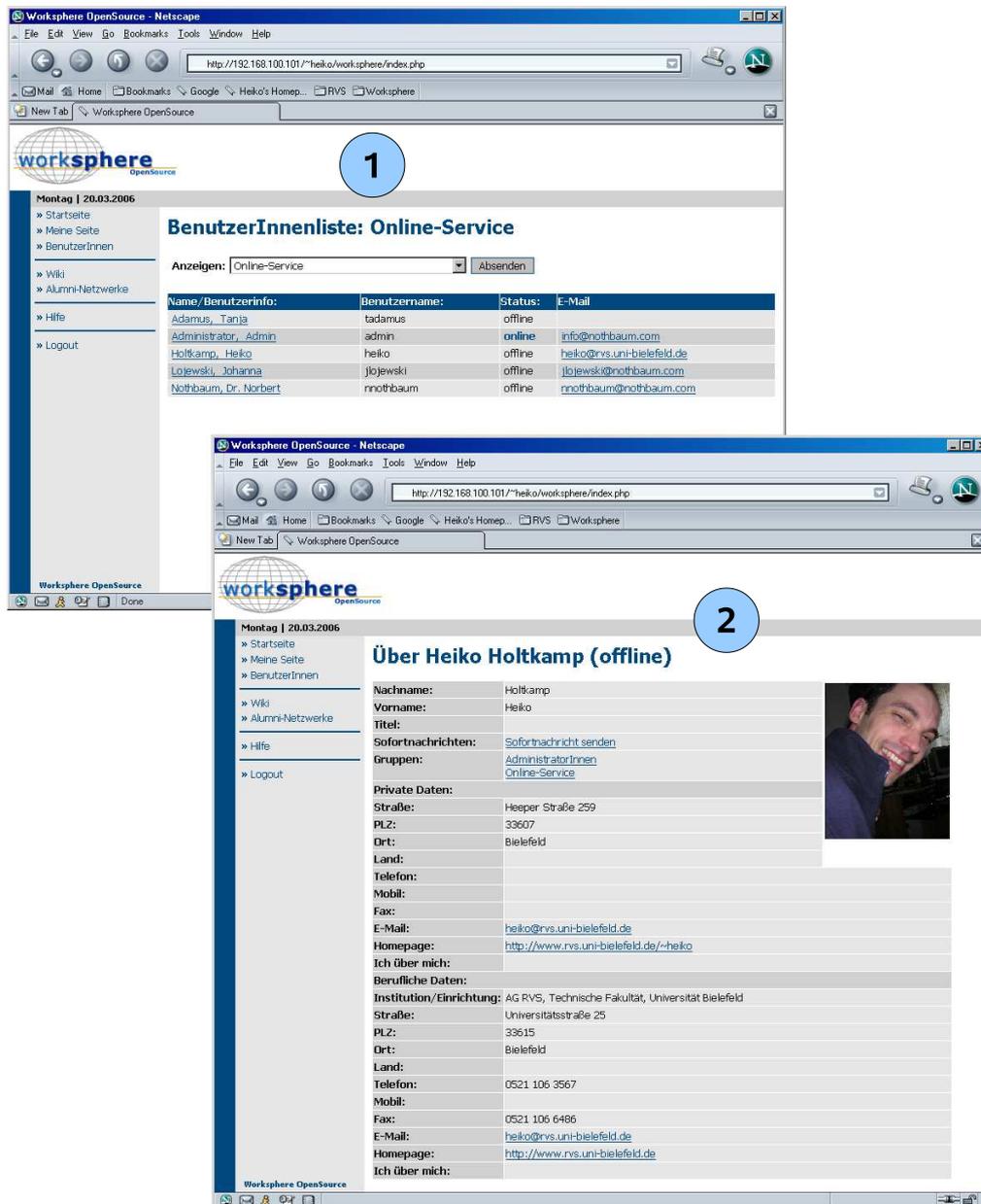


Abbildung 3.13.: Screenshot der Benutzerliste (1) und der Benutzerinfo-Seite (2).

### 3.3.6. Sofortnachrichten

Die Sofortnachrichten (auch oft als *Instant Messages* bezeichnet) erlauben es Benutzern, private Nachrichten untereinander zu senden. Die Funktion kann von der Informationsseite eines Benutzers aus aufgerufen werden. Benutzer können selber festlegen, ob sie den Empfang von Sofortnachrichten wünschen oder nicht. Erlaubt ein Benutzer den Empfang, ist auf seiner Informationsseite der Link „Sofortnachricht senden“ freigeschaltet, ansonsten werden andere Benutzer darüber informiert, dass der Teilnehmer keine Sofortnachrichten wünscht („Empfängt keine Sofortnachrichten“). Sendet ein Nut-

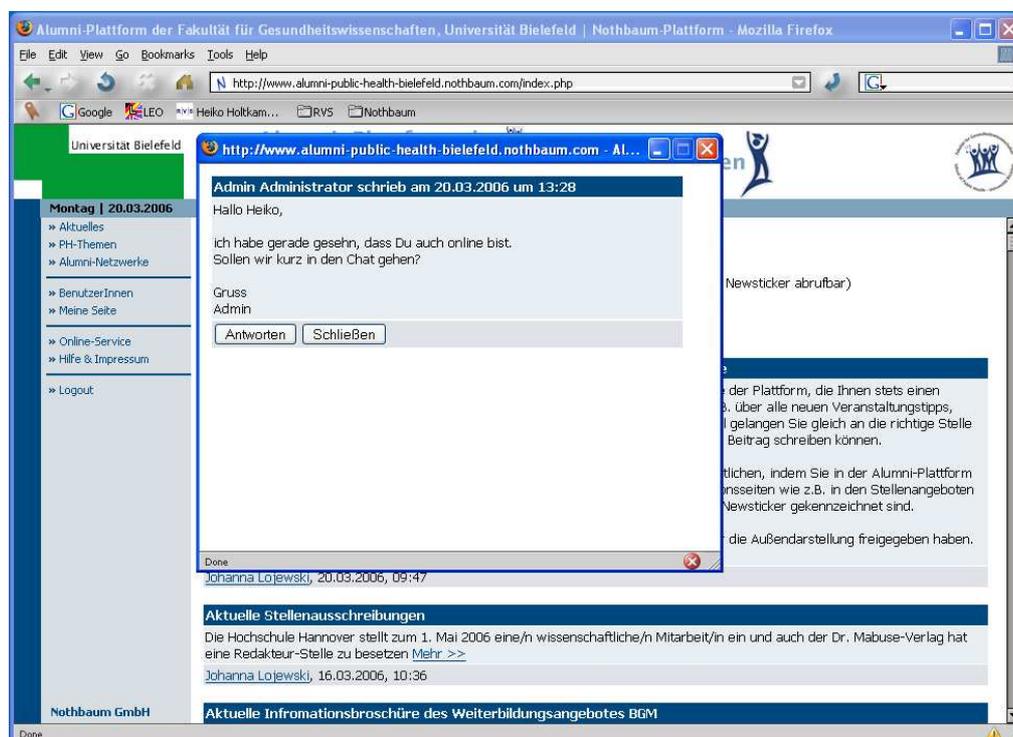


Abbildung 3.14.: Screenshot einer Sofortnachricht in der Worksphere.

zer eine Sofortnachricht an einen anderen Benutzer, erscheint diese Nachricht innerhalb weniger Sekunden auf seinem Bildschirm, wenn er in der Plattform eingeloggt ist (siehe Abbildung 3.14). Ansonsten wird die Nachricht zwischengespeichert und dem Nutzer angezeigt, sobald er sich das nächste mal in der Plattform einloggt. Benutzer haben über ein Archiv Zugriff auf ältere gesendete und empfangene Nachrichten. Der Zeitraum, in dem Nachrichten archiviert werden, kann von den Administratoren der Plattform festgelegt werden. Benutzer haben auf diesen Zeitraum keinen Einfluß.

### 3.3.7. Projektgruppen

Wie bereits in Kapitel 3.2.5 dargestellt, werden die Plattformbenutzer einer oder mehreren Gruppen zugeordnet; jeder Nutzer kann Mitglied beliebig vieler Gruppen sein. Der Bereich Projektgruppen fasst die Kommunikationsangebote einer Gruppe auf einer Übersichtsseite zusammen (siehe Abbildung 3.15). Der Nutzer kann sehen, welchen Projektgruppen er zugeordnet ist und welche Plattformangebote innerhalb dieser Gruppe zur Verfügung stehen: Chats, Schwarze Bretter, Foren, Dateibereiche etc. (alle diese Angebote funktionieren genau so, wie in den vorhergehenden Abschnitten beschrieben). Projektgruppen stellen keine neuen Angebote dar, sondern sind ein zweiter Weg, alle verfügbaren Angebote übersichtlich und gruppenorientiert darzustellen.

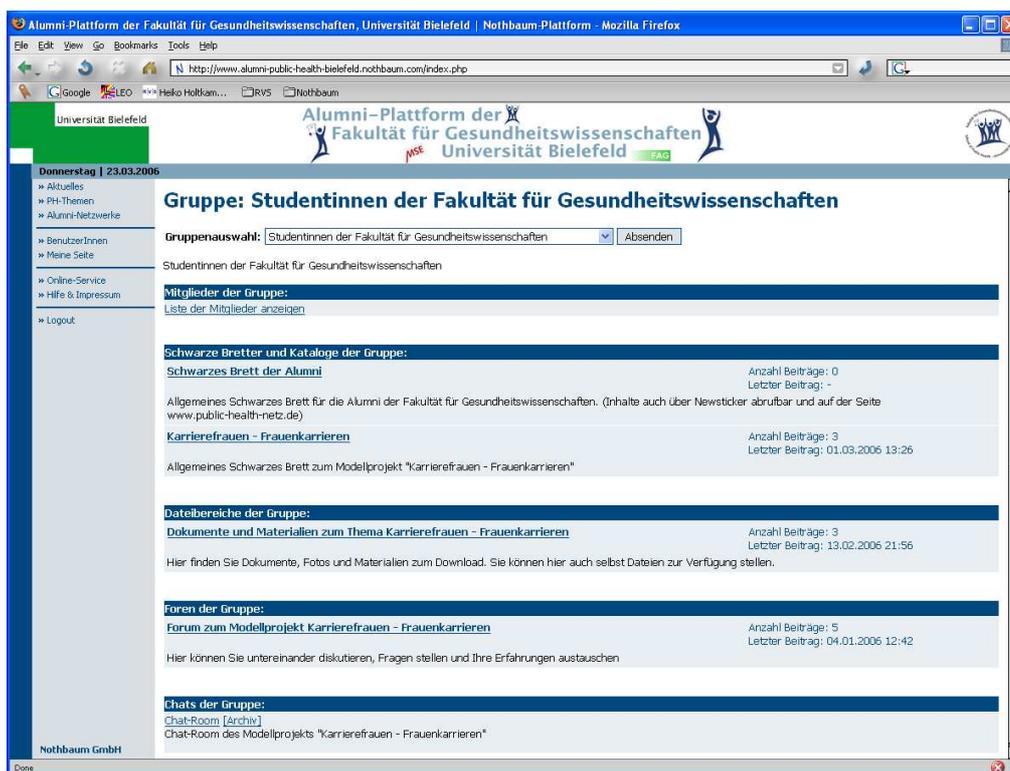


Abbildung 3.15.: Screenshot einer Projektgruppe in der Worksphere.

### 3.3.8. Newsticker

Der Newsticker basiert auf einem sogenannten *RSS-Feed*. RSS ist ein plattform-unabhängiges XML-basiertes Datenformat und wurde entwickelt, um Nachrichten und andere Web-Inhalte unkompliziert auszutauschen. Die Worksphere generiert RSS-Feeds nach der RSS 2.0 Spezifikation. In dieser Spezifikation steht RSS für *Really Simple Syndication*<sup>15</sup>. Die RSS-Feeds können mit jedem üblichen RSS-Reader (wie z.B. Sage, ein Plu-

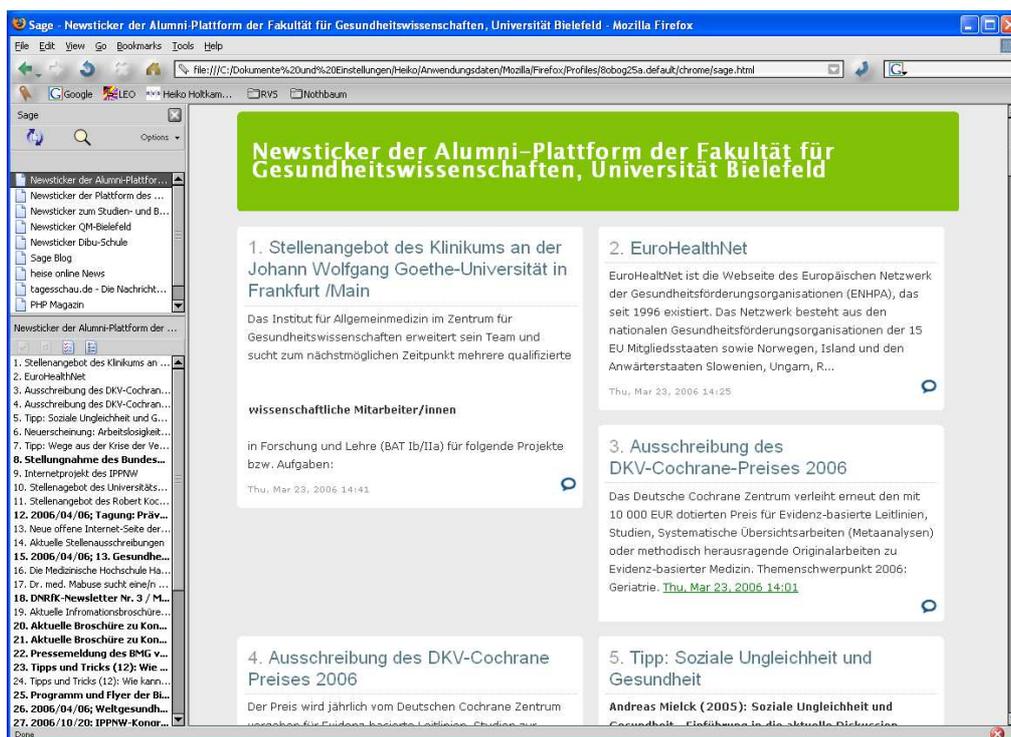


Abbildung 3.16.: Screenshot des Newstickers der Worksphere.

gin für Mozilla-basierte Browser, dem E-Mail-Programm Thunderbird etc.) abonniert werden. Nutzer haben damit die Möglichkeit, sich über Neuigkeiten in der Plattform zu informieren, ohne sich in die Worksphere einloggen zu müssen. Der Newsticker der Worksphere fasst Neuigkeiten in Foren, Schwarzen Brettern und Dateibereichen zusam-

<sup>15</sup>Die Abkürzung RSS hat in den verschiedenen technischen Spezifikationen eine unterschiedliche Bedeutung [WikipediaRSS]:

RSS 0.9x	Rich Site Summary	<a href="http://backend.userland.com/rss091">http://backend.userland.com/rss091</a>
		<a href="http://backend.userland.com/rss092">http://backend.userland.com/rss092</a>
RSS 1.0	RDF Site Summary	<a href="http://web.resource.org/rss/1.0/spec">http://web.resource.org/rss/1.0/spec</a>
RSS 2.0	Really Simple Syndication	<a href="http://blogs.law.harvard.edu/tech/rss">http://blogs.law.harvard.edu/tech/rss</a>

men (vgl. Abbildung 3.16). Es ist konfigurierbar, von welchen Kommunikationsmodulen Nachrichten in dem Feed gesammelt werden sollen, wie viele Einträge angezeigt werden sollen und innerhalb welchen Zeitraums. In der Version 1.0preX der Worksphere kann der Newsticker noch nicht über das Administrationstool verwaltet werden. Anpassungen und Konfigurationen werden direkt in dem PHP-Skript zum Erstellen eines Nachrichtenfeeds gemacht.



## 4. Implementation

„Functional, then Optimized“

– Paul Weston [Weston2004]

Als Vorgehensmodell zum Entwurf bzw. zur Implementierung des Wiki-Moduls in der Worksphere wurde die *Evolutionäre Entwicklung* gewählt. Bei diesem Ansatz überlappen die Phasen der Spezifikation, der eigentlichen Entwicklung/Implementierung und der Validierung; sie werden nicht als separate Abläufe bei der Software-Entwicklung behandelt (wie z.B. beim *Wasserfallmodell*). Mit Hilfe einer sehr abstrakten Spezifikation bzw. eines Anforderungskatalogs wird schnell ein erstes System als Prototyp entwickelt, das dann im Weiteren (durch Rückmeldungen der Benutzer/der Kunden) über mehrere Versionen hinweg verbessert und verfeinert wird, bis ein System entsteht, das den Anforderungen und Bedürfnissen der Anwender entspricht.

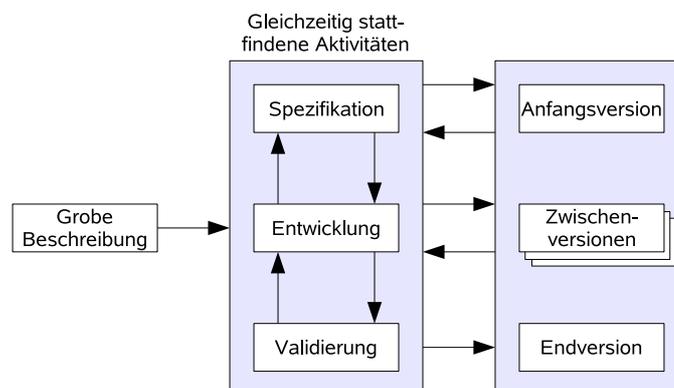


Abbildung 4.1.: Evolutionäre Entwicklung (nach [Sommerville2001]).

Vorteil bei der evolutionären Entwicklung ist, dass eine Spezifikation nach und nach während der Implementation entsteht. Diese Spezifikation kann am Ende der Entwick-

lung als Basis einer Neuentwicklung durch ein strukturierteres Entwurfsmodells dienen. Klare Nachteile der evolutionären Entwicklung sind (vgl. [Sommerville2001]):

- Der Entwicklungsprozess ist nicht unmittelbar sichtbar, da bei der schnellen Entwicklung des Prototypen nicht jeder Zwischenschritt dokumentiert wird und
- die Software ist oft schlecht strukturiert, da sie in einem schnellen Entwurfsprozess entsteht und stetige Veränderungen die Struktur der Software beeinflussen.

[Sommerville2001] führt aus, dass er die evolutionäre Entwicklung für kleine oder mittlere Systeme (bis zu 500.000 Codezeilen) mit einer kurzen Lebensdauer als den besten Ansatz erachtet. Für größere Systeme mit einer längeren Lebensdauer ist die evolutionäre Entwicklung ungeeignet, da hier die vorher genannten Probleme schnell sichtbar werden.

Die folgende Beschreibung der Entwicklung des Wiki-Moduls für die Worksphere ist keine Beschreibung des Softwareentwicklungsprozesses. Sie stellt nur die Informationen dar, die für das Verständnis der Form der Implementierung vonnöten sind und erläutert die Funktionsweise des Wiki-Moduls. Insbesondere wird darauf verzichtet, die einzelnen Zwischenschritte bei der evolutionären Entwicklung aufzuführen, sondern die Entwicklung in ihrer Gesamtheit beschrieben: wie ist die Implementation aufgebaut, welche Skripte wurden geschrieben, welche Funktion haben die Skripte. Die Implementierung erfolgt in der Skriptsprache PHP4, die auch zur Programmierung der Worksphere verwendet wird; als Datenbanksystem wird die relationale Datenbank MySQL genutzt (vgl. Kapitel 3). Eine erste Betrachtung des Anforderungskatalogs (siehe Kapitel 4.1) hat zu der Entscheidung geführt, das Wiki-Modul von Grund auf als eigene Programmierung neu zu entwickeln, anstatt zu versuchen die Skripte eines der zahlreich vorhandenen Wiki-Klone in die Worksphere zu integrieren (vgl. Kapitel 2.3). Die Betrachtung der Programmierung verschiedener Wiki-Klone hat gezeigt, dass der Aufwand, deren Programmierung anzupassen, vergleichbar ist mit der Neuprogrammierung einer Wiki-Engine<sup>1</sup>. Die Vorteile einer Neuentwicklung sind, dass die Wiki-Engine von vornherein auf die

---

<sup>1</sup>Betrachtet wurden die in PHP implementierten Wiki-Klone PmWiki, MediaWiki, TWiki und TikiWiki (vgl. Kapitel 2.3) sowie DokuWiki (<http://wiki.splitbrain.org/wiki:dokuwiki>), PHP-Wiki (<http://phpwiki.sourceforge.net>), QWikiWiki (<http://www.qwikiwiki.com>), WikkaWiki (<http://wikkawiki.org>) und WikkiTikkiTavi (<http://tavi.sourceforge.net>). Dabei hat sich gezeigt, dass es mit erheblichem Aufwand verbunden ist, die Wiki-Engines in die Rechteverwaltung der Worksphere zu integrieren und das Layout bzw. die Benutzeroberflächen der Wiki-Klone an die Worksphere anzupassen.

WorkspHERE zugeschnitten und angepasst werden kann und ein wesentlich größerer Einfluss auf die Implementierung besteht.

Das in der Arbeit entwickelte Wiki-Modul ist auf der dieser Arbeit beiliegenden CD-ROM enthalten (vgl. Anhang B).

## 4.1. Anforderungskatalog

Als Grundlage für die Implementierung des Wiki-Moduls dient der folgende abstrakte Anforderungskatalog. Alle genannten Anforderungen sind bei der Implementation des Wiki-Moduls umzusetzen.

- Das Wiki-Modul fügt sich in die WorkspHERE ein, d.h. die Authentifikation und Rechteverwaltung sowie die Stylesheets der WorkspHERE werden genutzt, damit das Layout dem der WorkspHERE angepasst ist.
- Es können beliebig viele, voneinander unabhängige Wikis in der WorkspHERE konfiguriert werden (wie z.B. auch beliebig viele verschiedenen Foren, Chats, Schwarze Bretter etc. in der WorkspHERE angelegt werden können).
- Die Daten des Wikis werden in der Datenbank der WorkspHERE gespeichert.
- Die grundlegenden Konzepte eines Wikis (vgl. Kapitel 2.2) werden umgesetzt, d.h. Benutzer können Seiten im Wiki (sofern berechtigt) lesen und (sofern berechtigt) bearbeiten, in dem sie die Seite direkt im Browser über ein HTML-Formular editieren.
- Es kann festgelegt werden, welche Benutzergruppen Lese- und Bearbeitungsrechte in einem Wiki haben. D.h. die Wikis der WorkspHERE sind u.U. nicht für alle Benutzer offen, sondern es können auch Wikis eingerichtet werden, in denen nur bestimmte Benutzergruppen Lese- und Schreibrechte haben.
- Einzelne Seiten eines Wikis können von berechtigten Benutzergruppen gesperrt werden, d.h. die Seiten sind für die Bearbeitung gesperrt. Die Sperrung kann von berechtigten Benutzergruppen wieder aufgehoben werden.
- Das Wiki bietet eine vereinfachte Syntax für die Formatierung des Textes, die sich an der Syntax von UseMod und MediaWiki orientieren soll (vgl. Kapitel 2.4.2).

- Für die Verlinkung von Wiki-Seiten wird die Free Link Syntax verwendet; externe Links werden unterstützt (vgl. Kapitel 2.4.2).
- Das Wiki unterstützt eine einfache Verlinkung auf andere Elemente in der Worksphere (Foren, Chats, Schwarze Bretter), um z.B. von einer Wiki-Seite auf ein Forum in der Worksphere verweisen zu können.
- Das Wiki bietet eine einfache Möglichkeit, die letzten Änderungen (Recent Changes) im Wiki nachzuvollziehen (vgl. Kapitel 2.2).
- Von jeder Seite des Wikis aus, kann eine Übersicht aufgerufen werden, die die Seiten auflistet, die auf die Seite verweisen (Referer) (vgl. Kapitel 2.2).
- Die Seiten des Wikis werden versioniert, d.h. alte Versionen einer Seite werden archiviert und sind jederzeit abrufbar.
- Das Wiki unterstützt eine Suchfunktion.

## 4.2. Datenbankentwurf

Die Relationen sind ad hoc und nicht über einen formalen Datenbankentwurfsprozess (z.B. über das Entity/Relationship-Modell) modelliert worden. Dieses Vorgehen ist zum einen durch den evolutionären Entwicklungsprozess gerechtfertigt, dessen Ziel die Entwicklung eines Prototypen ist, und zum anderen dadurch, dass für die bereits vorhandenen Module in der Worksphere (z.B. Foren, Dateibereiche etc.) eine geeignete Datenbankmodellierung durchgeführt wurde, an deren Ergebnis sich die Modellierung der Relationen für das Wiki-Modul orientiert.

In der Datenbank der Worksphere sind vier Verwaltungstabellen für das Wiki-Modul hinzugefügt:

- (1) `wiki_lists`: Speichert Informationen zu den in der Worksphere verfügbaren Wikis (siehe Tabelle 4.1). Es kann mehr als ein Wiki in der Worksphere angelegt werden.
- (2) `wiki_permissions_read`: Speichert die Gruppen, die das Recht zum Lesen von Seiten eines Wikis haben (siehe Tabelle 4.2).
- (3) `wiki_permissions_write`: Speichert die Gruppen, die das Recht zum Bearbeiten und Anlegen von Seiten eines Wikis haben (siehe Tabelle 4.3).

- (4) `wiki_permissions_lock`: Speichert die Gruppen, die das Recht haben, eine Seite eines Wikis dauerhaft für die Bearbeitung zu sperren oder diese Sperrung aufzuheben (siehe Tabelle 4.4).

Ferner werden für jedes Wiki in der Worksphere drei Tabellen erstellt. Die Tabellen werden von der Skripten zur Administration der Wikis automatisch erzeugt, wenn ein neues Wiki über das Administrationstool der Worksphere angelegt wird.

- (1) `wiki_<id>_current`: Speichert die jeweils aktuellen Versionen der Seiten in einem Wiki (siehe Tabelle 4.6). `<id>` gibt die ID des Wikis an, zu der die Tabelle gehört. In der Tabellenbezeichnung entspricht `<id>` der ID des Wikis in der Tabelle `wiki_lists`. So gehört die Tabelle `wiki_1_current` zum Wiki mit der ID 1 in der Tabelle `wiki_lists`.
- (2) `wiki_<id>_archive`: Archiviert die alten Versionen der Seiten eines Wikis (siehe Tabelle 4.5). Für `<id>` gilt das entsprechende wie bei den `current`-Tabellen.
- (3) `wiki_<id>_lock`: Speichert, ob die Seite für die Bearbeitung durch Benutzer gesperrt ist. Ein Eintrag in dieser Tabelle gibt an, zu welchem Zeitpunkt (Feld `timestamp`) die Seite von einem Benutzer gesperrt wurde. Ist in `timestamp` der Wert „-1“ eingetragen, kennzeichnet dies, dass die Seite dauerhaft für die Bearbeitung gesperrt ist, bis die Sperre durch einen berechtigten Benutzer aufgehoben wird (siehe Tabelle 4.7). Für `<id>` gilt das entsprechende wie bei den `current`-Tabellen.

Die Trennung in zwei Relationen „`current`“ und „`archive`“ zum Speichern der Seiten hat den Vorteil, dass bei größer werdenden Datenbeständen die Zugriffe auf die aktuellen Versionen einer Seite aufgrund des geringeren Datenumfangs in der Tabelle `current` effizienter sind<sup>2</sup>. Desweiteren hat die Trennung den Vorteil, dass die Archivtabelle zu beliebigen Zeitpunkten unabhängig von den aktuellen Versionen einer Seite geleert werden kann, so dass nur noch die aktuellen Versionen von Seiten in einem Wiki vorhanden sind.

Andere Relationen in der Datenbank der Worksphere mussten für die Implementation des Wiki-Moduls nicht verändert oder erstellt werden.

---

<sup>2</sup>Auf den Attributen `title` und `data` der Relation `current` liegt z.B. ein Volltext-Index, dessen Verwaltung durch das DBMS bei großen Datenbeständen natürlich immer aufwendiger wird.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Wikis, Primärschlüssel
timestamp	varchar(14)	Zeitstempel, wann das Wiki angelegt bzw. geändert wurde
name	tinytext	Name des Wikis
description	text	Beschreibung des Wikis
status	char(1)	0 = Wiki ist als gelöscht markiert, 1 = Wiki ist aktiv
allow_html	char(1)	0 = kein HTML erlaubt, 1 = HTML erlaubt, 2 = nur bestimmte HTML-Tags erlaubt

Tabelle 4.1.: Relation wiki\_lists.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Wikis, Primärschlüssel
timestamp	varchar(14)	Zeitstempel, wann das Recht gesetzt bzw. geändert wurde
<b>group_id</b>	int(32)	ID der Gruppe, Primärschlüssel

Tabelle 4.2.: Relation wiki\_permissions\_read.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Wikis, Primärschlüssel
timestamp	varchar(14)	Zeitstempel, wann das Recht gesetzt bzw. geändert wurde
<b>group_id</b>	int(32)	ID der Gruppe, Primärschlüssel

Tabelle 4.3.: Relation wiki\_permissions\_write.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Wikis, Primärschlüssel
timestamp	varchar(14)	Zeitstempel, wann das Recht gesetzt bzw. geändert wurde
<b>group_id</b>	int(32)	ID der Gruppe, Primärschlüssel

Tabelle 4.4.: Relation wiki\_permissions\_lock.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Eintrags (Auto-Inkrement), Primärschlüssel
<b>title</b>	tinytext	Titel der Wiki-Seite, Primärschlüssel
data	longtext	Inhalt der Wiki-Seite
comment	text	Kommentar zur Bearbeitung der Wiki-Seite
<b>timestamp</b>	varchar(14)	Zeitstempel, wann die Wiki-Seite gespeichert wurde, Primärschlüssel
user_id	varchar(32)	ID des Benutzers, der die Wiki-Seite gespeichert hat
ip	varchar(16)	IP-Adresse, von der aus die Wiki-Seite gespeichert wurde

Tabelle 4.5.: Relation wiki\_&lt;id&gt;\_archiv.

Feldname	Typ	Beschreibung
<b>id</b>	int(32)	ID des Eintrags (Auto-Inkrement), Primärschlüssel
<i>title</i>	tinytext	Titel der Wiki-Seite (auf dem Feld liegt ein Volltextindex)
<i>data</i>	longtext	Inhalt der Wiki-Seite (auf dem Feld liegt ein Volltextindex)
comment	text	Kommentar zur Bearbeitung der Wiki-Seite
timestamp	varchar(14)	Zeitstempel, wann die Wiki-Seite gespeichert wurde
user_id	varchar(32)	ID des Benutzers, der die Wiki-Seite gespeichert hat
ip	varchar(16)	IP-Adresse, von der aus die Wiki-Seite gespeichert wurde

Tabelle 4.6.: Relation wiki\_&lt;id&gt;\_current.

Feldname	Typ	Beschreibung
<b>title</b>	tinytext	Titel der Wiki-Seite, Primärschlüssel
timestamp	varchar(14)	Zeitstempel, wann die Seite gesperrt wurde. Ist in diesem Feld der Wert „-1“ gespeichert, ist die Seite permanent gesperrt.
user_id	varchar(32)	ID des Benutzers, der die Seite gesperrt hat

Tabelle 4.7.: Relation wiki\_&lt;id&gt;\_lock.

### 4.3. Benutzeroberfläche

Eines der grundlegenden Prinzipien eines Wikis ist die einfache Benutzbarkeit, die deshalb natürlich auch Ausgangspunkt der Implementation des Wiki-Moduls für die Worksphere sein soll. Einen wesentlichen Beitrag zur einfachen Bedienbarkeit liefert die Benutzeroberfläche (*User Interface*) eines Programms. Einfache Bedienbarkeit („*ease of use*“) wird in der Literatur oft mit dem Begriff *Usability* beschrieben (vgl. [Raskin2000], [Dix1998]). Linda Macaulay definiert Usability als:

„*Usability is the ease of learning, the ease of use, the flexibility of use, the effectiveness of use, and the user satisfaction with the system.*“ (zitiert nach [Paul-Stüwe2005], Seite 5)

Die Benutzeroberfläche für das Wiki-Modul soll sich vollständig in die Worksphere integrieren. Aus diesem Grund nutzt das Wiki-Modul die gleichen Stylesheets wie die übrigen Module der Worksphere; dies führt zu einem einheitlichen „*look & feel*“. Abbildung 4.2 zeigt den Entwurf der Bedienoberfläche für das Wiki.

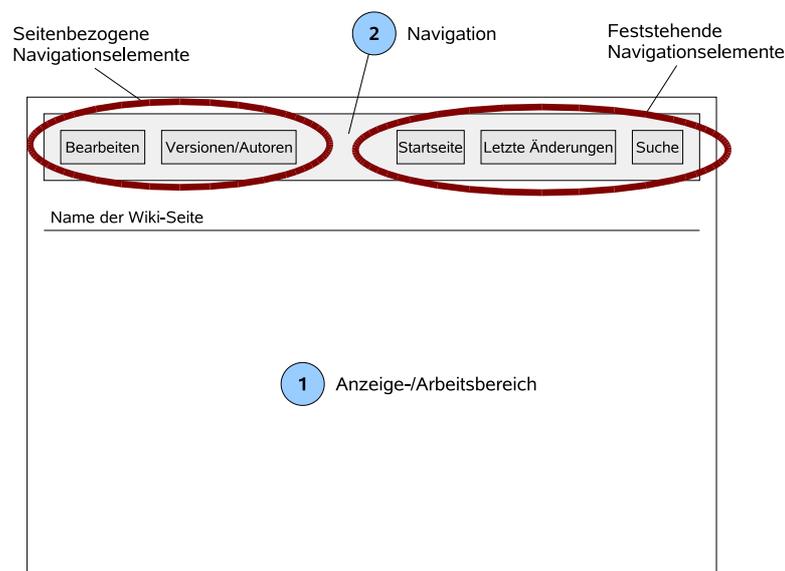


Abbildung 4.2.: Entwurf der Wiki-Benutzeroberfläche.

Die Oberfläche ist in zwei Hauptbereiche geteilt: (1) dem Anzeige- und Arbeitsbereich und (2) dem Navigationsbereich. Die Navigation findet in einem Wiki im Wesentlichen

über die Verweise im Text einer Seite statt (vgl. Kapitel 2.2, Seite 11). Für die weitere Navigation und den Zugriff auf Funktionen des Wikis (wie Suche, Bearbeiten einer Seite etc.) wird immer am oberen Rand der Benutzeroberfläche eine Menüleiste angezeigt. Die Menüleiste ist in zwei Bereiche getrennt. Rechts befinden sich die Menüpunkte der Funktionen, die in jedem Zustand des Wikis zur Verfügung stehen.

- **Startseite**, mit diesem Menüpunkt kann der Benutzer jederzeit wieder zur Startseite (der ersten Seite) des Wikis zurückspringen.
- **Letzte Änderungen**, zeigt eine Übersicht über die letzten Bearbeitungen von Seiten im Wiki. Hiermit kann sich der Benutzer schnell einen Überblick über die zuletzt veränderten oder neu angelegten Seiten im Wiki verschaffen.
- **Suche**, ermöglicht es dem Benutzer eine Volltextsuche nach Seitentiteln oder Stichworten im Wiki durchzuführen.

Im linken Bereich der Menüleiste befinden sich die Funktionen, die sich immer auf die aktuell angezeigte Seite im Anzeige-/Arbeitsbereich beziehen (vgl. Abbildung 4.3).



Abbildung 4.3.: Kontextsensitive Navigationselemente im linken Bereich der Menüleiste: (1) Elemente bei der normalen Anzeige einer Seite, (2) Elemente bei der Anzeige einer Seite im Bearbeitungsmodus, (3) Elemente bei der Anzeige einer archivierten Version einer Seite.

Die Benutzeroberfläche ist bewusst einfach gehalten, damit Nutzer die angebotenen Funktionen schnell erkennen und einfach nutzen können.

## 4.4. Module, Skripte und Verzeichnisse

### 4.4.1. Übersicht der Skripte und Verzeichnisse

Die folgenden Verzeichnisse und Skripte wurden für die Implementation des Wiki-Moduls der Worksphere hinzugefügt:

Skript/Verzeichnis:	Funktion:
./conf	Enthält die Konfigurationsdateien zur Worksphere
wiki.css.php	Datei mit speziellen Stylesheet-Angaben für das Wiki
./admin	Enthält die Administrationskripte für die Worksphere
wiki.php	Startseite der Wiki-Administration
wiki_aendern.php	Wiki-Einstellungen bearbeiten
wiki_bearbeiten.php	Zwischenskript für die Bearbeitungsfunktionen
wiki_loeschen.php	Wiki loeschen
wiki_neu.php	Neues Wiki einrichten
./wiki	Enthält die Skripte und Dateien für das Wiki-Modul
wiki.php	Steuerungsdatei für das Wiki
conf.inc.php	Konfigurationsdaten für das Wiki
edit.inc.php	Bearbeitungsfunktionen
lib_wiki.inc.php	Hilfsfunktionen
menu.inc.php	Funktionen zum Darstellen des Menüs
page.inc.php	Seitenverwaltungsfunktionen
render.inc.php	Darstellungsfunktionen ("Parser")
search.inc.php	Suchfunktionen
wiki.js	Bearbeitungsleiste für Wiki-Seiten (Javascript)

### 4.4.2. Administrationskripte des Wiki-Moduls

Die Skripte zur Administration des Wiki-Moduls sind im Verzeichnis ./admin enthalten, in dem auch die übrigen Administrationskripte der Worksphere gespeichert sind. Alle Skripte in diesem Verzeichnis dürfen nur von Benutzern mit Administratorrech-

ten ausgeführt werden (dies ist fest in den Skripten kodiert). Mit den Skripten zur Administration des Wiki-Moduls können neue Wikis angelegt werden (`wiki_neu.php`), die Einstellungen eines bestehenden Wikis geändert werden (`wiki_aendern.php`) oder ein Wiki als gelöscht markiert werden<sup>3</sup> (`wiki_loeschen.php`). Die Skripte arbeiten auf den Tabellen `wiki_lists`, `wiki_permissions_read`, `wiki_permissions_write` und `wiki_permissions_lock`. Beim Erstellen eines neuen Wikis werden vom Skript `wiki_neu.php` drei neue Tabellen in der Datenbank angelegt (`wiki_<id>_current`, `wiki_<id>_archive` und `wiki_<id>_lock`; `<id>` entspricht der internen ID des neu angelegten Wikis). Für die Wikis kann festgelegt werden, welche Benutzergruppen Leserechte und/oder Schreibrechte und welche Benutzergruppen das Recht zum Sperren einzelner Seiten im Wiki haben. Ferner kann festgelegt werden, ob in dem Wiki HTML zugelassen ist (der volle Sprachumfang von HTML), nur bestimmte HTML-Tags zugelassen sind (die zugelassenen Tags werden für alle Wikis in einer Konfigurationsdatei (s.u.) angegeben) oder generell keine HTML-Tags zugelassen sind (die Benutzer haben dann nur über die Wiki-Syntax die Möglichkeit den Text zu formatieren). Die Administrations-skripte sind in das Administrationstool der Worksphere integriert und können von dort aus aufgerufen werden (siehe Abbildung 4.4).

### 4.4.3. Aufbau des Wiki-Moduls

#### 4.4.3.1. Konfigurationsdatei

Die Konfigurationsdatei enthält eine ganze Reihe von Konstanten, die im Wiki-Modul verwendet werden. Der folgende Auszug aus der Datei macht im wesentlichen schon klar, welche Bedeutung die Konstanten im einzelnen haben. In der Datei ist auch die Liste erlaubter HTML-Tags enthalten, die Benutzer in einem Wiki zur Formatierung nutzen können, wenn dies über das Administrationstool konfiguriert wurde (s.o.).

```
<?php
2
  // Timestamp-Wert, der einen permanenten Lock signalisiert
  define('PERMANENT_LOCK', -1);

  // Maximale Tiefe fuer Ueberschriften
7  define('MAX_HEADING', 6);
```

---

<sup>3</sup>Die Worksphere hat das Konzept, dass keine Inhalte gelöscht werden, sondern nur mit Löschmarken versehen werden. Ausnahme davon sind Einträge auf Schwarzen Brettern, Dateibereichen etc., die von den Benutzern gelöscht wurden.

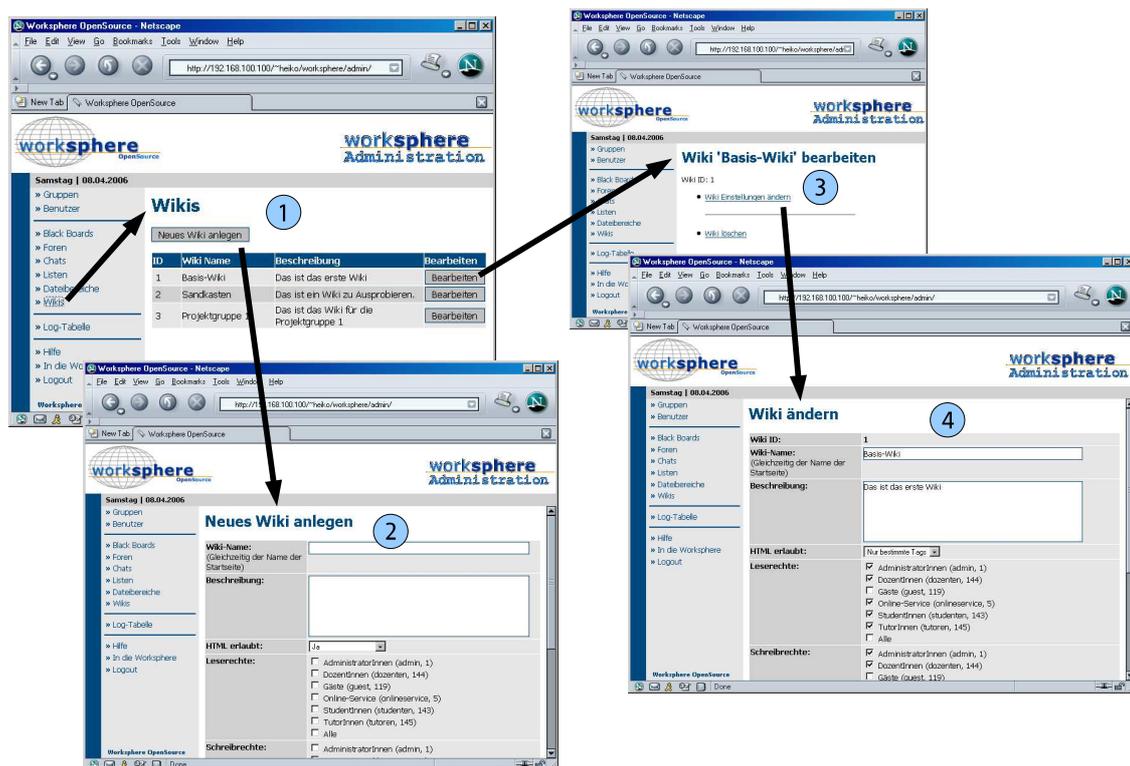


Abbildung 4.4.: Administrationsfunktionen für das Wiki-Modul. (1) Übersicht der eingerichteten Wikis, (2) Anlegen eines neuen Wikis; (3) Bearbeiten eines Wikis (Ändern/Löschen), (4) Ändern der Einstellungen eines Wikis.

```

// Basispfad fuer die Wiki-Skripte
define('WIKI_BASE', get_ws_baseuri()."/wiki/wiki.php");

12 // Format-String fuer Zeitangaben
define('TIME_STRING', 'd.m.Y H:i');

// Liste erlaubter HTML-Tags
define('ALLOWED_HTML_TAGS',
17     "<a>, <abbr>, <acronym>, <b>, <big>, ".
     "<blockquote>, <br>, <caption>, <center>, ".
     "<cite>, <code>, <col>, <colgroup>, <dd>, ".
     "<del>, <dfn>, <dir>, <div>, <dl>, <dt>, ".
     "<em>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, ".
22     "<hr>, <i>, <ins>, <isindex>, <kbd>, <legend>, ".
     "<li>, <map>, <menu>, <ol>, <p>, <pre>, <q>, ".
     "<s>, <samp>, <small>, <span>, <strike>, ".
     "<strong>, <sub>, <sup>, <table>, <tbody>, ".
     "<td>, <tfoot>, <th>, <thead>, <tr>, <tt>, ".
27     "<u>, <ul>, <var>, <img>"
);

// Bezeichner fuer Schaltflaechen im Wiki

32 // Startseite
define('START_LABEL', "Startseite");
define('START_TOOLTIP', "Startseite des Wiki anzeigen");

// Wiki-Historie
37 define('HISTORY_CHANGES', 20);
define('HISTORY_ACTION', "recentchanges");
define('HISTORY_LABEL', "Letzte &Auml;nderungen");
define('HISTORY_TOOLTIP', HISTORY_CHANGES." letzte &Auml;
    nderungen anzeigen");

42 ?>

```

Listing 4.1: Auszug aus der Datei conf.inc.php.

#### 4.4.3.2. Hilfsfunktionen

Das Skript `lib_wiki.inc.php` enthält eine Reihe von Hilfsfunktionen, u.a. zum Zugriff auf GET- und POST-Parameter sowie Funktionen, die die Zugriffe auf die in der Datenbank

gespeicherten Lese-, Schreib- und Sperrrechte abstrahieren (siehe Tabelle 4.8).

<b>Funktionsname:</b>	<b>Rückgabewert:</b>
<code>get_wiki_id()</code>	GET-Parameter id
<code>get_page()</code>	GET-Parameter page
<code>get_version()</code>	GET-Parameter version
<code>get_action()</code>	GET-Parameter action
<code>is_preview()</code>	POST-Parameter edit_preview
<code>is_save()</code>	POST-Parameter edit_save
<code>is_edit_lock()</code>	POST-Parameter edit_lock
<code>is_edit_unlock()</code>	POST-Parameter edit_unlock
<code>get_data()</code>	POST-Parameter edit_data
<code>get_comment()</code>	POST-Parameter edit_comment
<code>get_search_mode()</code>	GET-Parameter mode
<code>get_search_phrase()</code>	POST-Parameter search_phrase
<code>get_search_ordering()</code>	POST-Parameter search_ordering
<code>get_search_orderby()</code>	POST-Parameter search_orderby
<code>get_search_in()</code>	POST-Parameter search_in
<code>get_search_method()</code>	POST-Parameter search_method
<code>no_slashes(\$text)</code>	\$text ohne Maskierungen
<code>mask_quotes(\$text)</code>	\$text mit Maskierungen
<code>format_time(\$timestamp)</code>	\$timestamp als Datum formatiert
<code>print_wikiheader(\$text)</code>	Ausgabe von \$text als Wiki-Überschrift
<code>wiki_read_permission(\$wiki_id)</code>	true/false; Benutzer hat Leseberechtigung im Wiki \$wiki_id
<code>wiki_write_permission(\$wiki_id)</code>	true/false; Benutzer hat Recht zum Bearbeiten im Wiki \$wiki_id
<code>wiki_lock_permission(\$wiki_id)</code>	true/false; Benutzer hat Recht zum Sperren im Wiki \$wiki_id
<code>wiki_html_allowed(\$wiki_id)</code>	0/1/2; HTML nicht erlaubt, erlaubt, nur bestimmte Tags

Tabelle 4.8.: Liste der Hilfsfunktionen in `lib_wiki.inc.php`

#### 4.4.3.3. Menüleiste für das Wiki

Das Navigationsmenü des Wiki-Moduls ist als einfache HTML-Tabelle realisiert, die mit Stylesheet-Angaben formatiert wird. Das Menü beinhaltet drei feste Navigationspunkte: Startseite, Letzte Änderungen und Suche (vgl. Kapitel 4.3). Diese Navigationselemente werden am rechten Rand des Menüs dargestellt (vgl. Abbildung 4.2). Für die Darstellung des Menüs wird die Funktion `print_menu()` aufgerufen. Als Parameter kann der

Funktion ein assoziatives Array übergeben werden, das die darzustellenden Navigationselemente für die linke Seite des Menüs enthält. Der Index des Array-Elements wird dabei als Text in der Navigationsleiste angezeigt, der Wert des Array-Elements wird als „Aktion“ für das Navigationselement übernommen (der Wert sollte immer ein HTML-Link sein). Das folgende Skript zeigt einen beispielhaften Aufruf:

```
$menu = array();

$menu["Beispiel 1"] = "<a href='beispiel1.php' </a>";
$menu["Beispiel 2"] = "<a href='beispiel2.php' </a>";

print_menu($menu);
```

#### 4.4.3.4. Steuerungsskript für das Wiki

Die Skriptdatei `wiki.php` ist das Hauptskript des Wiki-Moduls; die Interaktionen des Benutzers mit dem Wiki werden über dieses Skript gesteuert. Das Skript bindet die Konfigurationsdatei der Worksphere ein (`./conf/prepend.php`) und lädt die Konfigurationsdaten für das Wiki-Modul sowie die Skripte, die die Hilfsfunktionen, Menüfunktionen, Bearbeitungsfunktionen, Suchfunktionen etc. enthalten.

Das Skript wird mit dem URL-Parameter `id` (in PHP als *GET-Parameter*<sup>4</sup> bezeichnet) aufgerufen, der die ID des Wikis angibt, in dem gearbeitet werden soll. Der Aufruf `wiki.php?id=1` ruft über das Wiki-Modul das Wiki mit der ID 1 auf. Wird keine ID oder eine fehlerhafte ID übergeben, liefert das Skript eine Fehlermeldung zurück.

Das Skript ist an die Sessionverwaltung, Authentifizierung und Rechteverwaltung der Worksphere gebunden. D.h., bevor ein Anwender Zugriff auf ein Wiki erhält, muss er sich mit Benutzername und Kennwort authentifizieren. Ferner prüft das Skript die Rechte, die der Benutzer in dem Wiki hat. Hat der Benutzer keine Leserechte, erhält er eine Rückmeldung, dass keine Inhalte angezeigt werden. Das Leserecht ist eine zwingende Voraussetzung dafür, dass ein Benutzer in einem Wiki arbeiten kann; hat er keine Leserechte, wird das Wiki nicht angezeigt, unabhängig davon ob er evtl. Schreibrechte hat.

Ein einfacher Handler in Form einer `switch`-Anweisung steuert die Funktionen des Wiki-Moduls abhängig von den Aktionen des Benutzers. Die Aktionen wer-

---

<sup>4</sup>Die Bezeichnung rührt von den Übertragungsmethoden her, die bei einem HTML-Formular für die HTTP-Übertragung – GET oder POST – verwendet werden können.

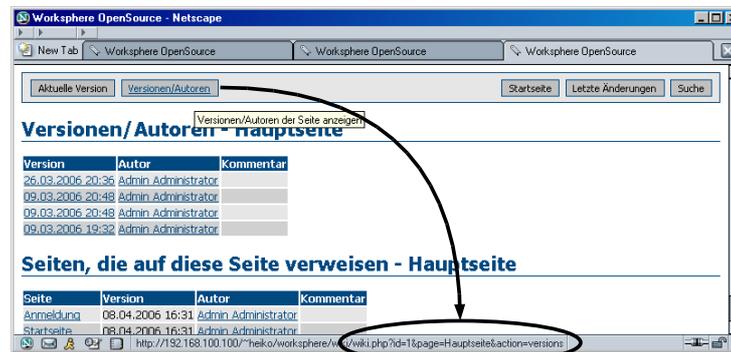


Abbildung 4.5.: Übertragung der Parameter in der URL.

den wie die ID des Wiki als Parameter in der URL übertragen: `action`. Mit weiteren Parametern wie z.B. `page` und `version` wird angegeben, auf welche Seite und welche Version der Seite eine Aktion ausgeführt werden soll. Der Aufruf `wiki.php?id=1&page=Hauptseite&action=versions` ruft z.B. die Versionsübersicht für die Seite „Hauptseite“ aus dem Wiki mit der ID 1 auf (vgl. Abbildung 4.5). Tabelle 4.9 zeigt eine Liste der verwendeten GET-Parameter zur Steuerung eines Wikis. Abhängig von der durchzuführenden Aktion (eine konkrete Aktion ist bestimmt durch eine Kombination von GET- und POST-Parametern) ruft das Skript die entsprechenden Funktionen auf und stellt das Ergebnis im Browser dar. Das Navigationsmenu des Wikis wird dabei immer dem dargestellten Ergebnis angepasst.

Vorteil dieses einfachen Steuerungsskripts ist es, dass neue Aktionen bzw. Funktionen sehr einfach dem Wiki-Modul hinzugefügt werden können. Es genügt, im Handler eine weitere Prüfung auf die Parameterkombination hinzuzufügen, die die neue Aktion auslöst, und die entsprechenden Funktionsaufrufe als Reaktion durchzuführen.

#### 4.4.3.5. Verwaltung der Wiki-Seiten

Das PHP-Skript `pages.inc.php` beinhaltet die Funktionen zum Zugriff auf die Seiten eines Wikis. Die Funktionen kapseln die Zugriffe auf die Datenbank. Tabelle 4.10 gibt eine Übersicht über die implementierten Funktionen.

Die Funktionen `update_page()` und `restore()` bedürfen einer weiteren Erläuterung. Die Funktion `update_page($page, $data, $comment)` erstellt eine neue Version der Seite `$page` mit den Daten `$data` und dem Bearbeitungskommentar `$comment`. Dabei wird

<b>URL/GET-Parameter:</b>	<b>Werte:</b>	<b>Bedeutung:</b>
id	Zahl (Integer)	ID des Wikis. Wird keine ID oder eine fehlerhafte ID übergeben, liefert das Skript eine Fehlermeldung zurück
page	Titel einer Wiki-Seite (String)	Titel der Seite, auf die sich eine Aktion bezieht. Wird der Parameter nicht angegeben, wird die Startseite (die erste Seite des Wikis angezeigt); wird ein Seitentitel einer nicht vorhandenen Seite übergeben, liefert das Skript eine Fehlermeldung zurück
version	Timestamp	Timestamp einer archivierten Seite (die Versionen einer Seite werden über Timestamps (UNIX-Timestamps) verwaltet). Wird der Parameter version nicht angegeben bezieht sich die eine Aktion immer auf die aktuelle, d.h. die letzte Version einer Seite.
action		Wird der Parameter action nicht angegeben, wird als Default-Aktion immer das Anzeigen einer Wiki-Seite ausgeführt. Das Anzeigen einer Seite ist abhängig von den Parametern page und version
	edit	Aufrufen des Formulars zum Bearbeiten einer Seite. Abhängig vom Wert des URL-Parameters page.
	edit_submit	Bearbeitung der Wiki-Seite durchführen. Abhängig von weiteren POST-Parametern (siehe Kapitel).
	versions	Anzeigen der gespeicherten Versionen einer Seite. Abhängig vom Wert des URL-Parameters page.
	restore	Wiederherstellen einer alten Version einer Seite. Abhängig von den Werten der URL-Parameter page und versions.
	recentchanges	Liste der letzten Änderungen im Wiki aufrufen.
	search	Suchformular für das Wiki aufrufen. Das angezeigte Suchformular ist abhängig vom Wert des URL-Parameters mode.
	search_submit	Durchführen der Suche und Darstellen der Suchergebnisse. Abhängig von weiteren POST-Parametern (siehe Kapitel).
mode		Der Parameter mode ist nur in Verbindung mit dem Parameter action=search relevant. Wird der Parameter mode nicht angegeben, wird ein Standardsuchformular angezeigt.
	extended	Es wird ein Formular zu einer erweiterten Suche, angezeigt.
	allpages	Es wird eine Liste aller Seiten im Wiki angezeigt.

Tabelle 4.9.: Liste der GET-Parameter zur Steuerung eines Wikis.

<b>Funktion:</b>	<b>Beschreibung:</b>
<code>exists_version(\$page, \$version)</code>	true/false. Ermittelt, ob die Seite \$page in der Version \$version existiert.
<code>exists_page(\$page)</code>	true/false. Ermittelt, ob die Seite \$page existiert.
<code>fetch_page(\$page, \$version)</code>	Liefert die Daten der Seite \$page in der Version \$version zurück, falls vorhanden.
<code>update_page(\$page,\$data,\$comment)</code>	Erstellt eine neue Version der Seite \$page mit den Daten \$data und dem Bearbeitungskommentar \$comment.
<code>restore_page(\$page,\$version)</code>	Die archivierte Seite \$page in der Version \$page wird wieder zur aktuellsten Version gemacht.
<code>lock_page(\$page)</code>	Sperrt die Seite \$page für die Bearbeitung.
<code>unlock_page(\$page)</code>	Hebt die Sperrung der Seite \$page auf.
<code>is_locked(\$page)</code>	true/false. Ermittelt, ob für die Seite \$page eine Bearbeitungssperre existiert.
<code>locked_by(\$page)</code>	Liefert die User-ID des Benutzers zurück, der die Sperre auf die Seite \$page gesetzt hat
<code>unlock_my_pagelock()</code>	Eigene Seitensperre aufheben.
<code>is_permanent_locked(\$page)</code>	true/false. Ermittelt, ob für die Seite \$page eine permanente Bearbeitungssperre existiert.
<code>permanent_unlock(\$page)</code>	Hebt die permanente Sperre der Seite \$page auf.
<code>permanent_lock(\$page)</code>	Setzt eine permanente Sperrung auf die Seite \$page.
<code>print_versions(\$page)</code>	Gibt eine Liste der Versionen der Seite \$page als HTML-Liste aus.
<code>print_recentchanges()</code>	Gibt eine Liste der letzten Änderungen im Wiki als HTML-Liste aus.
<code>print_referredby(\$page)</code>	Gibt eine HTML-Liste der Seiten aus, die auf die Seite \$page verweisen.
<code>print_all_pages()</code>	Gibt eine Liste aller Seiten des Wikis als HTML-Liste aus.

Tabelle 4.10.: Liste der Seitenverwaltungsfunktionen.

eine evtl. bereits vorhandene Version der Seite in der Archiv-Tabelle `wiki_<id>_archive` des entsprechenden Wikis gesichert und die neue Version der Seite in die Tabelle `wiki_<id>_current` geschrieben. Die Funktion `restore($page, $version)` macht die Seite `$page` in der Version `$version` wieder zur aktuellsten Version der Seite. Dazu wird die bisher aktuelle Version in die Archiv-Tabelle verschoben und die archivierte Seitenversion in die Tabelle `wiki_<id>_current` zurückgeschrieben. Der Seite wird als Bearbeitungskommentar hinzugefügt, dass es sich um die Wiederherstellung der archivierten Version einer Seite handelt.

#### 4.4.3.6. Anzeigen von Wiki-Seiten

Die Funktionen zur Darstellung von Wiki-Seiten sind im Skript `render.inc.php` implementiert. Die zentralen Funktionen sind die Funktionen `print_page($page, $version)` und `render($text)`.

```

function print_page($page, $version) {
    if (!exists_version($page, $version))
        return false;
4
    // richtige Version finden
    $text = fetch_page($page, $version);

    // Ausgabe der formatierten Seite
9    render(explode("\n", $text));
}

```

Die Funktion `print_page()` wird mit den Parametern `$page` und `$version` aufgerufen. Falls die Seite in der angegebenen Version vorhanden ist, wird über die Funktion `fetch_page()` der Inhalt der Seite aus der Datenbank gelesen<sup>5</sup>. Der aus der Datenbank gelesene Text der Seite wird über die PHP-Funktion `explode()` mit dem Trennzeichen „newline“ `\n` in ein Array zerlegt, das Trennzeichen wird dabei entfernt. Das Array enthält die einzelnen Zeilen des Textes (eine Zeile ist durch „newline“ am Ende gekennzeichnet). Die Funktion `render()` wird mit dem Array als Parameter aufgerufen.

Die Funktion `render()` arbeitet die Array-Elemente ab und wandelt die im Text enthaltenen Formatierungen in passenden HTML-Code zur Darstellung im Browser um. Das folgende Listing zeigt einen Auszug aus der Funktion `render()`:

<sup>5</sup>Wird der Parameter `$version` nicht übergeben, prüft die Funktion `exists_version()`, ob die Seite in der aktuellen Version, d.h. ohne Versionsangabe verfügbar ist. Gleiches gilt für die Funktion `fetch_page()`, sie liefert in diesem Fall die Daten der aktuellen Version der Seite zurück.

```

function render($text) {
    (...)
    // Zeile fuer Zeile von $text durchlaufen
    foreach ($text as $line) {
5       $line = parse_html($line);
        $line = parse_links($line);
        $line = parse_headings($line);
        $line = parse_newline($line);
        (...)
10      }
        (...)
    }
}

```

Mit der Funktion `render()` wird ein einfacher zeilenbasierter Parser für die Wiki-Syntax umgesetzt. Für die Erkennung der Auszeichnungen werden die *Perl Compatible Regular Expressions (PCRE)* von PHP genutzt. Das folgende Listing zeigt die Verwendung eines PCRE für das Erkennen der Formatierung für Kursivtext in der Funktion `parse_italic`:

```

function parse_italic ($text)
{
3   return preg_replace("/'(.*)'"/, "<i>\\1</i>", $text);
}

```

Die Syntax des implementierten Wiki-Moduls orientiert sich, wie im Anforderungskatalog gefordert, an der Syntax der Wiki-Engines UseMod und MediaWiki. Abbildung 4.6 zeigt die Anwendung der Wiki-Syntax in einem Text und das Ergebnis der Darstellung. Die Wiki-Syntax beschränkt sich auf ein Minimum an Formatierungselementen („*Less is more*“, vgl. Kapitel 2.4, S. 17). Es wird z.B. keine spezielle Syntax für die Formatierung von Tabellen im Wiki unterstützt. Sollen komplexere Formatierungen auf Seiten in einem Wiki vorgenommen werden, kann hierzu die Eingabe von HTML-Tags in dem Wiki freigeschaltet werden (vgl. Kapitel 4.4.2). Die Implementierung des „Renderers“ als einfacher zeilenbasierter Parser ermöglicht es aber auch, auf einfache Weise neue Wiki-Syntax-Elemente aufzunehmen.

**Absätze** Absätze werden durch einen doppelten Zeilenumbruch – eine extra Leerzeile – gekennzeichnet. Die Funktion `parse_new_paragraph($line)` implementiert einen sehr einfachen Automaten, der Absätze im Wiki-Text erkennt und durch die entsprechenden HTML-Tags zur Kennzeichnung eines Absatzes ersetzt (`<p> </p>`).

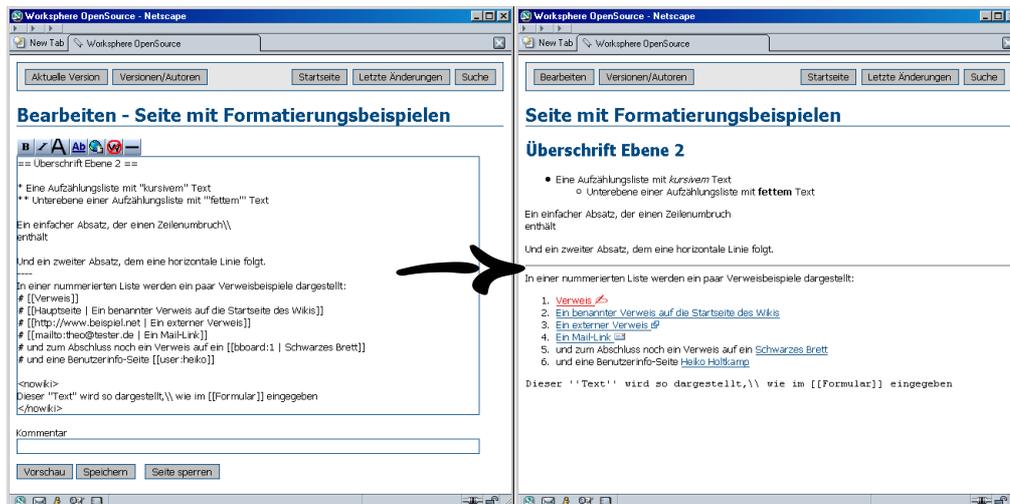


Abbildung 4.6.: Darstellung eines mit der Wiki-Syntax formatierten Textes.

**Zeilenumbrüche** Zeilenumbrüche werden durch zwei Backslashes `\\` gekennzeichnet. Die Funktion `parse_newline($text)` ersetzt die Vorkommen von `\\` durch den HTML-Tag `<br>`.

**Horizontale Linie** Horizontale Linien werden durch vier (oder mehr) Bindestriche (`-`) in einer Zeile im Text formatiert. Vor und nach den Bindestrichen darf kein weiterer Text folgen, d.h. die Bindestriche stehen alleine in einer Zeile. Die Funktion `parse_hr($text)` sucht nach den Vorkommen von vier Bindestrichen und ersetzt sie durch den HTML-Tag `<hr>`.

**Fett und Kursiv** Die Funktionen `parse_italic($text)` und `parse_bold($text)` wandeln die Wiki-Formatierungen für Fett- und Kursivschrift in die entsprechenden HTML-Tags `<b>` `</b>` und `<i>` `</i>` um. Text der kursiv dargestellt werden soll, muss in zwei einfache Anführungszeichen ("kursiv") eingeschlossen werden, Text der in Fettschrift dargestellt werden soll, in drei einfachen Anführungszeichen ("fett"). Die Wiki-Auszeichnungen für Fett- und Kursivschrift können verschachtelt werden (z.b. zu "'fett und 'kursiv'").

**Überschriften** Um Text als Überschrift zu formatieren wird er in Gleichheitszeichen eingeschlossen. Die Anzahl der Gleichheitszeichen gibt die Ebene der Überschrift an.

Der Text, der als Überschrift formatiert werden soll, muss alleine in einer Zeile stehen, d.h. kein weiterer Text darf vor und nach dem Text stehen, und er muss links und rechts von der gleichen Anzahl „=“ umgeben sein. == Überschrift Ebene 2 == formatiert z.B. einen Text als Überschrift der Ebene 2. Die Funktion `parse_headings($text)` setzt die Wiki-Syntax für Überschriften in die entsprechenden HTML-Tags `<h1>` bis `<h6>` um. Tiefere Überschriftenebenen als 6 werden nicht unterstützt.

**Listen** Die Elemente einer Aufzählungsliste beginnen mit einem Stern (\*) am Anfang einer Zeile; für geschachtelte Listen werden mehrere Sternchen an den Anfang der Zeile geschrieben, entsprechend der Tiefe der Schachtelungstiefe. Nummerierte Listen werden anstelle des Sterns mit dem Doppelkreuzzeichen (#) begonnen. Aufzählungslisten und nummerierte Listen können miteinander verschachtelt werden (z.B. \*# Aufzählungselement). Die Wiki-Formatierungen für Listen werden mit der Funktion `parse_lists($text)` in die entsprechenden HTML-Listenumgebungen umgesetzt.

**Verweise** Verweise werden über eine Variante der Free Link Syntax formatiert. Alle Verweise, sowohl Wiki-interne als auch -externe Links, müssen von doppelten eckigen Klammern umschlossen werden. Die Syntax für Verweise ist für das in dieser Arbeit implementierte Wiki-Modul ergänzt worden, um auch die Möglichkeit zu bieten, aus einem Wiki einfach auf andere Plattformelemente zu verlinken. Abbildung 4.11 stellt die Syntaxvarianten für Verweise dar. Die Funktion `parse_links($text)` ersetzt die Vorkommen von Wiki-Verweisen durch die entsprechenden HTML-Verweise (`<a href> </a>`). Für Wiki-interne Verweise prüft die Funktion, ob die Seite bereits existiert oder nicht. Existiert die Seite, wird der Link anders dargestellt als bei einer Seite, die nicht existiert (die entsprechende Darstellung der Links wird in der Stylesheet-Datei (`./conf/wiki.css.php`) für das Wiki-Modul festgelegt). Zeigt der Link auf eine Seite, die noch nicht existiert, wird dem Link zusätzlich der URL-Parameter `action=edit` zugefügt, so dass die Seite bei Bedarf durch einfaches Anklicken des Links erstellt werden kann.

**HTML-Code** Die Funktion `parse_html($text)` entfernt, je nachdem wie das Wiki konfiguriert ist (vgl. 4.4.2), entweder alle HTML-Tags, nur bestimmte HTML-Tags oder keine HTML-Tags aus dem übergebenen Text. Das Entfernen der Tags wird mit der PHP-Funktion `strip_tags($string, $allowed_tags)` durchgeführt, die intern von der Funktion aufgerufen wird. Der Funktion `strip_tags()` wird eine Zeichenkette (`$string`) und eine Liste erlaubter Tags (`$allowed_tags`) übergeben. Alle Tags, die nicht in der Liste vorkommen, werden aus dem String entfernt.

<b>Wiki-Syntax:</b>	<b>Bedeutung:</b>
[[Verweise]]	Wiki-interner Verweis auf die Seite „Verweise“
[[http://www.besipiel.org]]	Externer Verweis auf die URL http://www.besipiel.org
[[ftp://ftp.beispiel.org]]	Externer Verweis auf die URL http://www.besipiel.org
[[mailto:theo@tester.de]]	Verweis auf die E-Mail-Adresse theo@tester.de
[[bboard:<id>]]	Verweis auf das Schwarze Brett mit der ID <id> in der Work- sphere
[[forum:<id>]]	Verweis auf das Forum mit der ID <id> in der Worksphere
[[chat:<id>]]	Verweis auf den Chat mit der ID <id> in der Worksphere
[[chat_archiv:<id>]]	Verweis auf das Chat-Archiv mit der ID <id> in der Work- sphere
[[file:<id>]]	Verweis auf den Dateibereich mit der ID <id> in der Work- sphere
[[wiki:<id>]]	Verweis auf das Wiki mit der ID <id> in der Worksphere
[[group:<id>]]	Verweis auf die Projektgruppe mit der ID <id> in der Work- sphere
[[user:<username>]]	Verweis auf die Benutzerinfo-Seite des Benutzer mit dem Benutzernamen <username> in der Worksphere
<b>Für alle Link-Varianten ist die folgende Variante als sogenannter „benannter Link“ (piped Link) möglich:</b>	
[[Verweis   Anderer Text]]	Der Verweis wird als Link-Text mit dem Text „Anderer Text“ dargestellt.

Tabelle 4.11.: Wiki-Syntax für Verweise.

**No-Wiki Modus** Über die Auszeichnung `<nowiki> text </nowiki>` können Textstellen gekennzeichnet werden, die zur Anzeige nicht formatiert werden sollen – präformatierte Textstellen. Der Text, der von `<nowiki>` umschlossen ist, wird so dargestellt wie eingegeben. Die Tags `<nowiki>` und `</nowiki>` müssen jeweils zu Beginn einer Zeile stehen. Hiermit ist es z.B. möglich, Wiki-Formatierungen als präformatierten Text zu kennzeichnen. Die Formatierungen werden zur Darstellung dann nicht geparsed und in die entsprechenden HTML-Tags zur Anzeige umgesetzt.

#### 4.4.3.7. Bearbeiten von Wiki-Seiten

Bei der Bearbeitung von Seiten müssen zwei Fälle unterschieden werden. Zum einen muss das Eingabeformular zum Bearbeiten einer Seite aufgerufen werden und zum anderen müssen die Eingaben des Formulars verarbeitet werden.

Die Datei `edit.inc.php` enthält die Funktionen zum Bearbeiten einer Seite. Das Skript stellt die folgenden Funktionen zur Verfügung:

- `edit_page($page)`
- `print_edit_form($page, $data, $comment)`
- `print_edit_preview($page)`

Die Eingabemaske unterscheidet sich von der Darstellung einer Seite dadurch, dass ein HTML-Formular zur Bearbeitung des Quelltextes der Seite an die Stelle der Ausgabe der Seite tritt. Die Funktion `edit_page($page)` ist für den Aufruf des Eingabeformulars verantwortlich. Aufgerufen wird die Funktion von der Steuerungsdatei des Wikis. Das folgende Listing zeigt eine vereinfachte Version des Quellcodes zum Aufruf der Funktion in der Steuerungsdatei `wiki.php`.

```
1  case 'edit':
    unlock_my_pagelock();
    $menu = array();
    if ( exists_page($page) )
    {
2      (...)
3      print_menu($menu);
4      print_wikiheader(EDIT_LABEL." - ".$page);

5      // Seite bearbeiten
```

```

11         edit_page($page) ;

         break ;

```

Die Funktion `edit_page($page)` prüft, ob der Benutzer Bearbeitungsrechte in dem Wiki hat und ob die in dem Parameter `$page` übergebene Seite für die Bearbeitung gesperrt ist. Eine Seite ist für die Bearbeitung gesperrt, wenn (1) gerade ein anderer Benutzer die Seite bearbeitet (`is_locked($page)`) oder (2), eine permanente Sperre auf die Seite gesetzt ist (`is_permanent_locked($page)`).

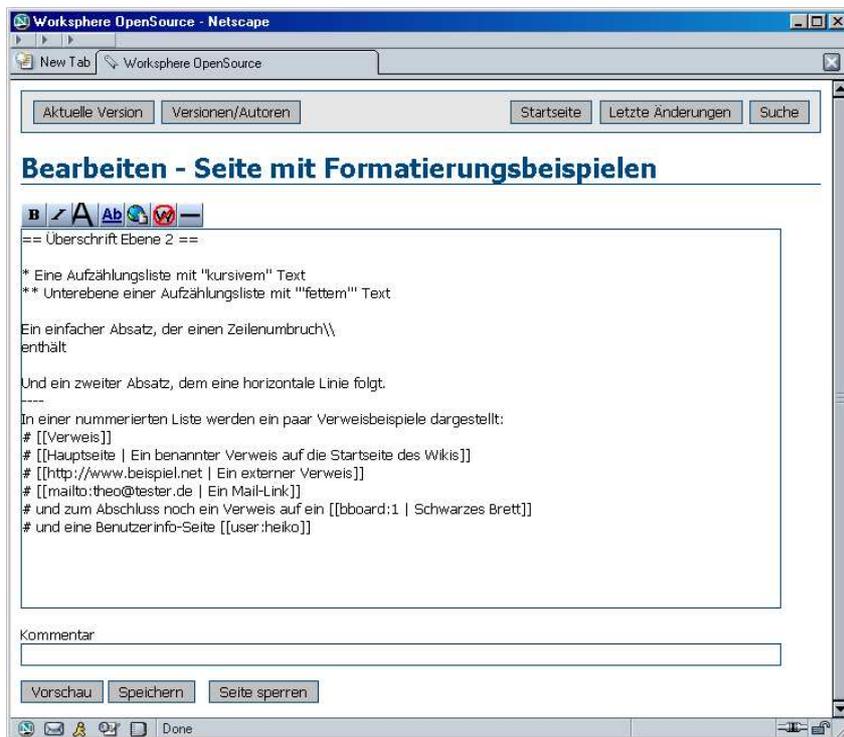


Abbildung 4.7.: Eingabemaske des Wiki-Moduls.

Mit diesem einfachen Prinzip werden Versionskonflikte bei der Bearbeitung von Seiten vermieden: Wenn eine Seite von einem Benutzer bearbeitet wird, kann sie von anderen Benutzern weiterhin gelesen werden, aber nicht mehr bearbeitet, bis der Benutzer die Bearbeitung beendet hat. Was passiert, wenn ein Benutzer vergisst, die Bearbeitung zu beenden? Über die Auto-Logout-Funktion der Worksphere wird der Benutzer nach einer bestimmten Zeit der Inaktivität aus der Worksphere abgemeldet, damit wird auch die Bearbeitungssperre aufgehoben.

Liegt keine Bearbeitungssperre auf der Seite `$page`, liest die Funktion `edit_page` die Daten der zu bearbeitenden Seite über den Funktionsaufruf `fetch_page($page, null)` (d.h. die Daten der aktuellsten Version der Seite `$page` werden gelesen) auf und ruft anschließend die Funktion `print_edit_form($page, $data, $comment)` auf. Die Funktion `print_edit_form()` stellt die Eingabemaske für die Bearbeitung der Seite dar. Die Eingabemaske wird zusätzlich um eine Bearbeitungsleiste ergänzt, falls es über die Browser Internet Explorer, Opera oder Gecko-basierte Browser (Mozilla, Firefox) aufgerufen wird<sup>6</sup>. Die Icons der Bearbeitungsleiste sind von der Wiki-Engine MediaWiki übernommen. Die Eingabemaske ist in Abbildung 4.7 dargestellt.

Bei der Verarbeitung der über das Formular übertragenen Daten müssen vier Fälle unterschieden werden. Das folgende Listing zeigt eine vereinfachte Version der Fallunterscheidung in der Datei `wiki.php`

```

    case 'edit_submit':
2      // Seite soll in der Vorschau angezeigt werden
        if ( is_preview() )
        {
            print_menu($menu);
            print_wikiheader(EDIT_PREVIEW_LABEL." - ".$page);
7          print_edit_preview($page);
        }

        // Seite soll gespeichert werden
        if ( is_save() )
12       {
            if (!update_page($page, get_data(), get_comment()))
            {
                // Update gescheitert... -> Fehlermeldung
            }
17         else
            {
                print_menu($menu);
                print_wikiheader($page);
                print_page($page, null);
22         }
        }

        // Seite soll gesperrt werden
        if ( is_edit_lock() )

```

---

<sup>6</sup>Die Bearbeitungsleiste basiert auf Javascript und kann aufgrund von unterschiedlichen Implementierungen des Document Object Model (DOM) in den Browsern noch nicht für alle Browser verwendet werden. Die Javascript-Funktionen für die Funktionen der Bearbeitungsleiste sind in der Datei `wiki.js` implementiert.

```
27     {
        print_menu($menu);
        print_wikiheader($page);
        permanent_lock($page)
        print_page($page, null);
32     }

    // Seite soll entsperrt werden
    if ( is_edit_unlock() )
    {
37         if ( wiki_lock_permission($wiki_id) AND
            permanent_unlock($page) )
            {
                print_menu($menu);
                print_wikiheader(EDIT_LABEL." - ".$page);
                print_page($page, null);
42            }
            else
            {
                print_menu($menu);
                print_wikiheader(EDIT_LABEL." - ".$page);
47                ws_errormsg(ERR_PERMANENT_NO_UNLOCK);
            }
    }

    break;
```

Bei der Vorschau werden die Eingaben in das Bearbeitungsformular durch den Aufruf der Funktion `edit_print_preview` wie bei der Darstellung einer Seite ausgegeben und zusätzlich unterhalb der Ausgabe auch die Eingabemaske, damit noch Korrekturen an der Seite vorgenommen werden können (vgl. Abbildung 4.8). Die Übertragung der Seitendaten geschieht über POST-Parameter bzw. die POST-Methode des HTTP-Protokolls. Der Zugriff auf die POST-Parameter wird über die in Kapitel 4.4.3.2 vorgestellten Hilfsfunktionen gekapselt.

Bei dem Speichern der Seite werden die Eingaben in das Bearbeitungsformular als neue Version der Seite gespeichert. Dies geschieht über den Aufruf der Funktion `update_page` (siehe Kapitel 4.4.3.5). Anschließend wird die Seite dargestellt.

Die beiden anderen Fälle betreffen das Setzen bzw. Entfernen einer permanenten Bearbeitungssperre für die Seite. Zum Setzen einer permanenten Sperre auf die Seite wird die Funktion `permanent_lock($page)` aufgerufen, zum Entfernen der Sperre die Funktion `permanent_unlock($page)`. Voraussetzungen für die (Ent-)Sperrung einer Seite sind,

dass der Benutzer die erforderlichen Rechte besitzt und das die Seite aktuell nicht von einem anderen Benutzer bearbeitet wird.

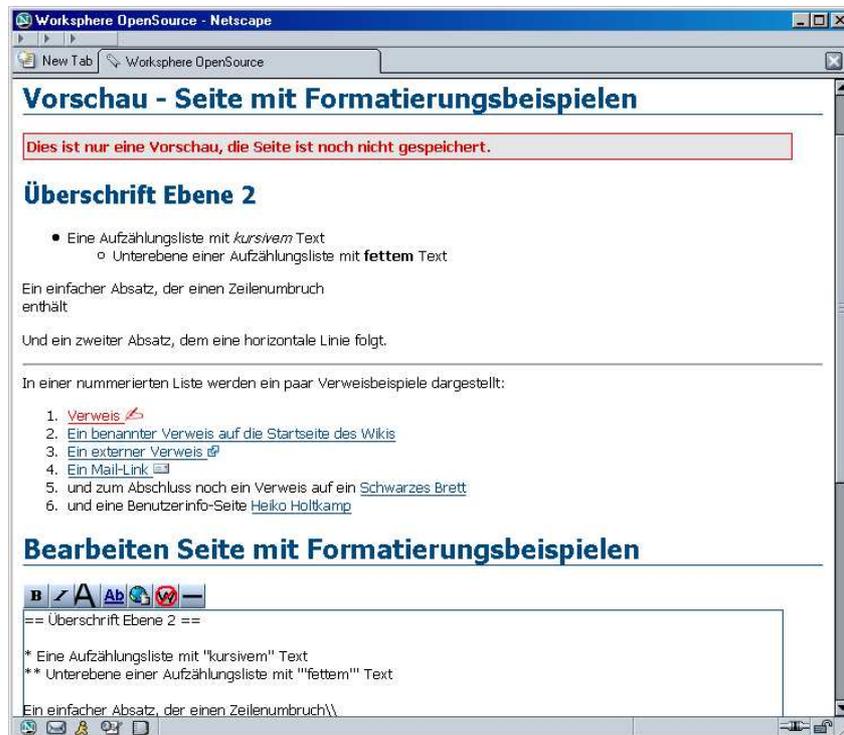


Abbildung 4.8.: Vorschau auf eine bearbeitete Seite.

#### 4.4.3.8. Wiederherstellen archivierter Seiten

Es kann immer nur die aktuelle Version einer Seite bearbeitet werden. Soll eine archivierte Version einer Seite bearbeitet werden, muss diese zunächst wieder zur aktuellen Version der Seite gemacht werden. Über die Funktion `restore($page, $version)` kann die Version `$version` der Seite `$page` wieder zur aktuellen Version gemacht werden (vgl. Kapitel 4.4.3.5). Der Seite wird dabei als Bearbeitungskommentar hinzugefügt, dass es sich um die Wiederherstellung einer Seite handelt (siehe Abbildung 4.9). Die Funktion `restore` wird von der Steuerungsdatei des Wikis bei Übergabe des Wertes `restore` für den GET-Parameter `action` aufgerufen. Vor dem Wiederherstellen der Seite wird geprüft, ob eine Bearbeitungssperre auf der Seite liegt und ob der Benutzer das Recht zum Bearbeiten von Wiki-Seiten hat.

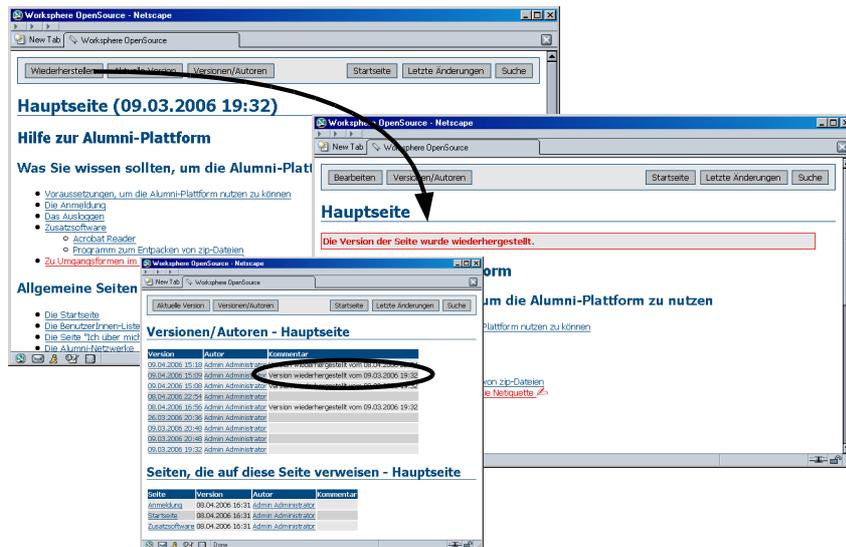


Abbildung 4.9.: Wiederherstellen einer archivierten Wiki-Seite.

#### 4.4.3.9. Versionen von Wiki-Seiten

Die Funktion `print_versions($page)` gibt eine Übersicht der gespeicherten Versionen und der Autoren der Seite `$page` aus. Die Funktion `print_referredby($page)` erstellt eine Liste der Seiten, die auf die Seite `$page` mit einem Link verweisen (siehe Abbildung 4.10). Beide Funktionen sind in der Datei `lib_wiki.inc.php` definiert (siehe Kapitel 4.4.3.2). Aufgerufen werden die Funktionen von der Steuerungsdatei des Wikis. Das folgende Listing zeigt eine vereinfachte Version des Quellcodes zum Aufruf der Funktionen in der Steuerungsdatei `wiki.php`.

```

case 'versions':
    print_menu($menu);

4    // Seitentitel ausgeben
    print_wikiheader(VERSIONS_LABEL." - ".$page);

    if (exists_page($page))
    {
9        print_versions($page);
        print_wikiheader(VERSIONS_REFEREDBY_LABEL." - ".$page);
        print_referredby($page);
    }
    else

```

```

14      {
            ws_errormsg( ERR_PAGE_NOT_FOUND );
        }
    }
    break ;

```

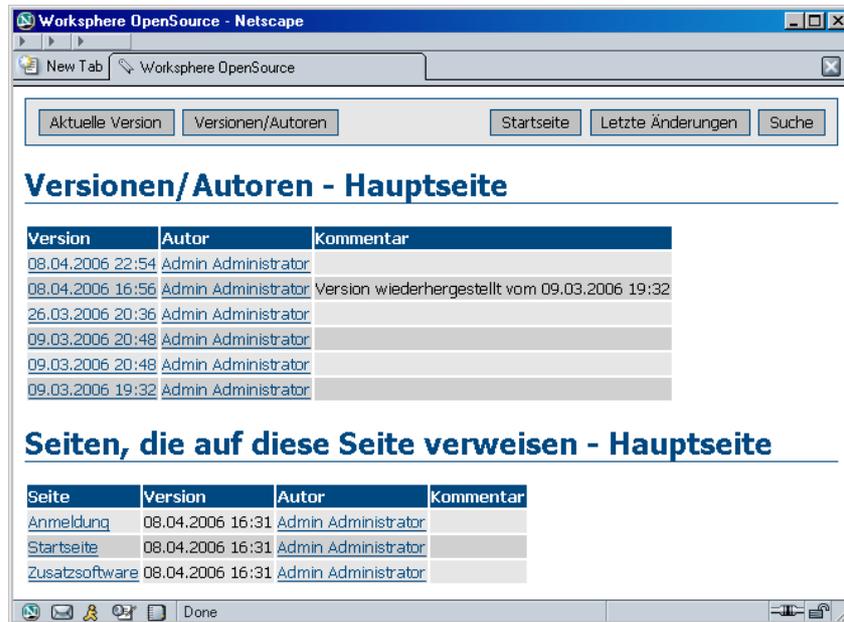


Abbildung 4.10.: Anzeige der Versionen einer Wiki-Seite und der Seiten, die auf die Seite verweisen.

#### 4.4.3.10. Letzte Änderungen im Wiki

Über den Aufruf der Funktion `print_recentchanges()` (definiert in der Datei `lib_wiki.inc.php`, vgl. Kapitel 4.4.3.2) wird eine Übersicht der letzten Änderungen im Wiki angezeigt (siehe Abbildung 4.11). Die Anzahl der dargestellten Einträge wird über die Konstante `HISTORY_CHANGES` in der Konfigurationsdatei `conf.inc.php` des Wiki-Moduls festgelegt. Die Funktion wird über die Steuerungsdatei des Wikis aufgerufen. Im nachfolgenden Listing ist der Aufruf der Funktion skizziert.

```

    case 'recentchanges' :
        print_menu($menu) ;

3

        print_wikiheader(HISTORY_LABEL) ;

        print_recentchanges() ;

```

8

**break ;**

Seite	Version	Autor	Kommentar
Seite mit Formatierungsbeispielen	08.04.2006 23:03	Admin Administrator	
Hauptseite	08.04.2006 22:54	Admin Administrator	
Hauptseite	08.04.2006 16:56	Admin Administrator	Version wiederhergestellt vom 09.03.2006 19:32
Startseite	08.04.2006 16:31	Admin Administrator	
Anmeldung	08.04.2006 16:31	Admin Administrator	
Zusatzsoftware	08.04.2006 16:31	Admin Administrator	
Zusatzsoftware	08.04.2006 16:30	Admin Administrator	
Hauptseite	26.03.2006 20:36	Admin Administrator	
Ein Test	18.03.2006 22:00	Admin Administrator	
Ein Test	18.03.2006 22:00	Admin Administrator	Version wiederhergestellt vom 18.03.2006 21:54
ZIP-Programme	18.03.2006 21:59	Norbert Nothbaum	Version wiederhergestellt vom 18.03.2006 21:57
ZIP-Programme	18.03.2006 21:58	Norbert Nothbaum	
ZIP-Programme	18.03.2006 21:57	Norbert Nothbaum	
Ein Test	18.03.2006 21:56	Admin Administrator	
Ein Test	18.03.2006 21:54	Admin Administrator	
ZIP-Programme	18.03.2006 21:52	Admin Administrator	Version wiederhergestellt vom 18.03.2006 21:34
ZIP-Programme	18.03.2006 21:51	Admin Administrator	RESTORED_COMMENT 18.03.2006 21:34
ZIP-Programme	18.03.2006 21:34	Admin Administrator	
ZIP-Programme	18.03.2006 21:34	Admin Administrator	
ZIP-Programme	18.03.2006 21:30	Admin Administrator	

Abbildung 4.11.: Übersicht der letzten Änderungen im Wiki.

#### 4.4.3.11. Suche in Wiki-Seiten

Bei der Suche werden zwei Fälle unterschieden. Zum einen das Darstellen des Suchformulars und zum anderen das Durchführen der Suche, abhängig von den Eingaben im Suchformular. Die Datei `search.inc.php` enthält die Funktionen, die die Suche und das Suchformular implementieren. Das Skript stellt die folgenden Funktionen zur Verfügung:

- `print_search_form($mode)`: Funktion zur Darstellung des Suchformulars.
- `search()`: Funktion zum Durchführen der Suche.

Die Funktionen werden abhängig vom Wert des GET-Parameters `action` in der Steuerungsdatei `wiki.php` des Wikis aufgerufen.

```
    case 'search':  
2     print_menu($menu);  
      print_wikiheader(SEARCH_LABEL);  
      print_search_form(get_search_mode());  
      break;  
  
7  case 'search_submit':  
    print_menu($menu);  
    print_wikiheader(SEARCH_LABEL);  
    search();  
    break;
```

Das Wiki-Modul bietet drei verschiedene Suchfunktionen, die abhängig vom GET-Parameter `mode` aufgerufen werden.

- `mode=allpages`: Ausgabe aller Seiten des Wikis. Dies ist eigentlich nicht als Suche bzw. Suchfunktion zu bezeichnen, da einfach eine Liste aller im Wiki vorhandenen Seiten ausgegeben wird. Die Funktion ist aber dem Menüpunkt „Suche“ zugeordnet.
- `mode=simple`: Einfache Suchmaske. Enthält ein Eingabefeld, in das ein Suchmuster eingegeben werden kann, nach dem im Seitentitel und -inhalt von Wiki-Seiten gesucht werden soll.
- `mode=extended`: Erweiterte Suchmaske. Neben dem Eingabefeld für das Suchmuster können mit der erweiterten Suche noch Angaben zur Darstellung der Suchergebnisse gemacht werden.

Statt eine eigene Volltextsuche zu implementieren, wird die Suche mit Hilfe der Volltextsuchfunktionen von MySQL durchgeführt. Die Volltextsuchfunktionen stehen seit Version 3.23.23 der MySQL-Datenbank zur Verfügung [MySQL2006]. Eine Volltextsuche kann nur auf die Feldtypen Char, Varchar und Text einer MyISAM-Tabelle ausgeführt werden.

Die Volltextsuche wird über die MySQL-Anweisung `MATCH() ... AGAINST()` durchgeführt. `MATCH()` nimmt als Eingabeparameter eine kommaseparierte Liste von Spaltennamen der Spalten, die durchsucht werden sollen. `AGAINST()` nimmt als Parameter das Suchmuster, nachdem die Spalten durchsucht werden sollen und einen optionalen Parameter, der die Suchmethode festlegt.

```
SELECT * FROM wiki_1_current MATCH(title, data)  
      AGAINST('suchmuster' IN BOOLEAN MODE);
```

Sucht in der Tabelle `wiki_1_current` in den Spalten `title` und `data` nach dem Suchmuster „suchmuster“ mit der Suchmethode **BOOLEAN MODE**.

Das Wiki-Modul verwendet nur die Suchmethode **BOOLEAN MODE** (zu Details der anderen Suchmethoden sei auf die MySQL-Referenz [MySQL2006] verwiesen). Die **BOOLEAN MODE**-Suchmethode unterstützt verschiedene Operatoren:

- + : Ein Pluszeichen vor einem Wort zeigt an, dass das Wort in dem Suchergebnis vorkommen muss.
- : Ein Minuszeichen vor einem Wort zeigt an, dass das Wort in dem Suchergebnis nicht vorkommen darf.
- \* : Der Stern dient als Wildcard-Zeichen. „Start\*“ sucht nach allen vorkommen von Anfang, gefolgt von beliebigem Text: Start, Startzeit, Starter etc.

Daneben wird noch eine Vielzahl weiterer Operatoren unterstützt, die hier nicht alle aufgeführt werden sollen, weitere Informationen zu den Operatoren sind in der MySQL-Referenz zur Volltextsuche zu finden ([MySQL2006]).

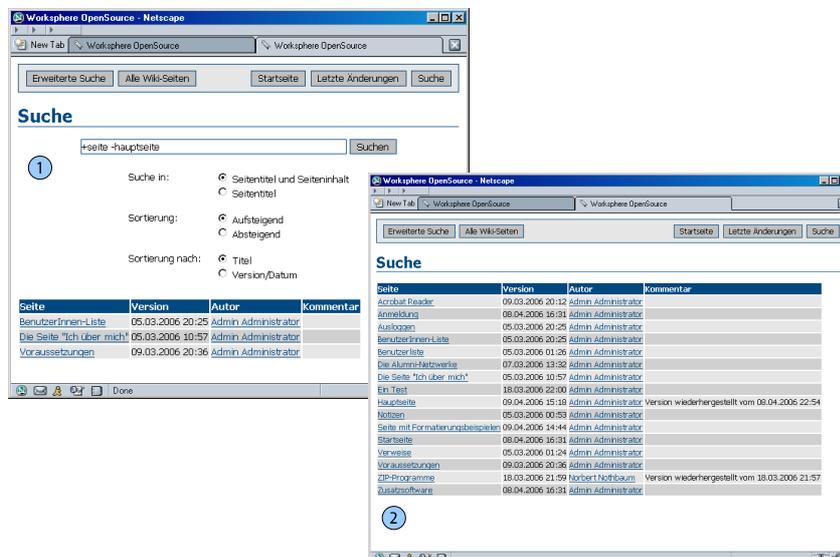


Abbildung 4.12.: Erweitertes Suchformular des Wiki-Moduls.



## 5. Bedrohungsanalyse

Mit einer Bedrohungsanalyse ist die systematische Identifikation und Abschätzung von Bedrohungen, denen ein System ausgesetzt ist, möglich. Eine Bedrohungsanalyse sollte kein einmaliger Prozess sein, sondern ein sich wiederholender, permanenter Prozess. Die Bedrohungsanalyse sollte die Entwicklungsphasen des Systems begleiten und während des gesamten Lebenszyklus des Systems fortgesetzt werden. Es ist nicht möglich, alle möglichen Bedrohungen gegen ein System in einem Durchgang zu identifizieren. Während des Lebenszyklus eines Systems können neue Bedrohungen hinzukommen, die bei einer anfänglichen Analyse nicht bedacht wurden oder bedacht werden konnten. Das Ergebnis einer Bedrohungsanalyse kann als eine Dokumentation angesehen werden, die die Bedrohungen gegen das System enthält und aus der sich Maßnahmen zur Absicherung des Systems gegen die Bedrohungen ableiten lassen. Das Risiko<sup>1</sup> eines Angriffs gegen das System kann entschärft werden, die tatsächliche Bedrohung aber nicht. Die Bedrohung ist immer vorhanden, ungeachtet der Maßnahmen die gegen sie getroffen wurden. Eine Bedrohungsanalyse kann bei der Bewertung der Risiken und der Wahl geeigneter Gegenmaßnahmen eine wertvolle Hilfe sein (vgl. [Schneier2001]).

Die im Rahmen dieser Arbeit durchgeführte Bedrohungsanalyse stützt sich auf die Sicherheitsanalysen von Martin Ellermann und Andre Döring, die sie in ihren Diplomarbeiten durchgeführt haben (siehe [Ellermann2002] und [Döring2003a]). Die Erkenntnisse aus den durchgeführten Analysen von Döring und Ellermann lassen sich auf eine Bedrohungsanalyse für das in dieser Arbeit entwickelte und in die Worksphere integrierte Wiki-Modul übertragen<sup>2</sup>.

---

<sup>1</sup>Risiko sei hier definiert als: Risiko = Wahrscheinlichkeit \* Schadenspotential (vgl. [WikipediaRisiko]). Das Risiko, das von einer bestimmten Bedrohung ausgeht ist gleich dem möglichen Schaden, der im Falle eines erfolgreichen Angriffs auf das System eintritt, multipliziert mit der Wahrscheinlichkeit, dass dieser Angriff eintritt.

<sup>2</sup>Im Gegensatz zu Döring und Ellermann bezeichne ich die hier durchgeführte Analyse als Bedrohungsanalyse und nicht als Sicherheitsanalyse, da die Analyse die einzelnen Bedrohungen nicht weitergehend bewertet, um damit letztendlich die Sicherheit des System zu bestimmen (vgl. dazu [Schneier2001]). Aus der Analyse gehen lediglich die Bedrohungen gegen das System hervor, daraus werden wiederum *Empfehlungen* abgeleitet, wie das System sicherer gemacht werden kann.

## 5.1. Bedrohungsanalyse mit Angriffsbäumen

Der Begriff *Angriffsbaum* (*Attack Tree*) wurde von Bruce Schneier geprägt. „Angriffsbäume bieten eine methodische Möglichkeit, Bedrohungen gegen ein System und Maßnahmen für dessen Schutz zu erkennen und damit die Sicherheit von Systemen zu beschreiben.“ [Schneier2001]. Angriffsbäume sind in ähnlicher Form als Konzept sogenannter *Fehlerbäume* (*Fault Trees*) schon seit langem bekannt und werden erfolgreich bei der Zuverlässigkeitsanalyse (Dependability Analysis) von Systemen eingesetzt (vgl. [Stamatelatos2002], [Moore2001]).

Angriffsbäume stellen Angriffe gegen ein System in einer Baumstruktur dar. Die Analyse beginnt mit einem Angriffsziel – das als Wurzelknoten des Baums dargestellt wird – und arbeitet rückwärts die Möglichkeiten heraus, dieses Ziel zu erreichen. Die Knoten unterhalb des Angriffsziels repräsentieren die Möglichkeiten, wie das Ziel erreicht werden kann. Die Möglichkeiten, die zum Erreichen des Ziels führen, werden durch logische Operatoren miteinander verknüpft; Angriffsbäume sind UND/ODER-Bäume [Schneier2001]. Mit ODER-Knoten werden alternative Möglichkeiten, ein Ziel zu erreichen, dargestellt. UND-Knoten stellen verschiedene notwendige und zusammenhängende Schritte dar, um ein Ziel zu erreichen. Die Möglichkeiten zur Zielerreichung können entweder elementare Methoden sein, die nicht weiter untergliedert werden, oder können Möglichkeiten sein, die ein Unterziel darstellen, zu dessen Erreichung verschiedene Untermöglichkeiten führen. Der Baum wird so lange fortgeführt bzw. verfeinert, bis einzelne Möglichkeiten nicht weiter verfeinert werden können (oder eine weitere Verfeinerung für die Analyse nicht notwendig ist).

Angriffsbäume können grafisch oder textuell, mittels einer Gliederung, repräsentiert werden (vgl. Abbildung 5.1 und Abbildung 5.2).

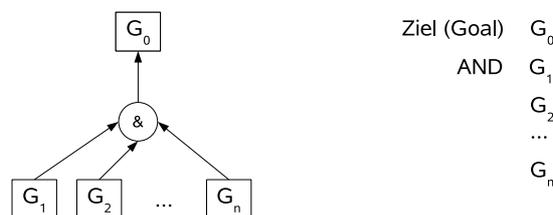


Abbildung 5.1.: Grafische und textuelle Repräsentation einer UND-Verknüpfung.

Angriffsbäume bestehen aus einer beliebigen Kombination von UND- und ODER-Verknüpfungen. Angriffsszenarien werden aus einem Angriffsbaum abgeleitet, in dem der

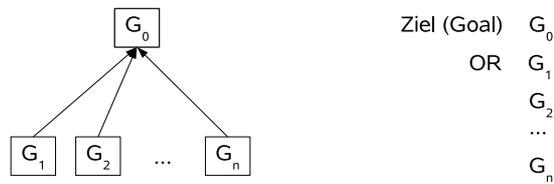


Abbildung 5.2.: Grafische und textuelle Repräsentation einer ODER-Verknüpfung.

Baum in Art einer Tiefensuche durchlaufen wird. Ausgehen vom Angriffsziel werden die Knoten des Baums bis zu einem Blattknoten durchlaufen. Die logische Verknüpfung der Blattknoten bis zum Ziel des Angriffs stellt ein Angriffsszenario dar [Moore2001]. Alle in einem Baum dargestellten Angriffsszenarien können zu einem globalen Angriffsszenario verknüpft werden. Das globale Angriffsszenario beinhaltet alle Möglichkeiten, einen Angriff auf das Ziel durchzuführen (vgl. Abbildung 5.3).

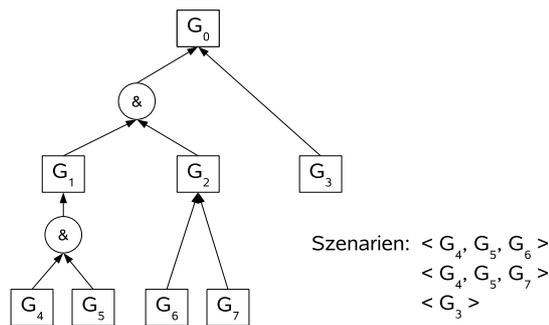


Abbildung 5.3.: Angriffsszenarien eines Beispiel-Angriffsbaums.

Zusätzlich können den Knoten des Angriffsbaums Werte bzw. Gewichte zugeordnet werden. Dies können Kosten zum Erreichen des entsprechenden Unterziels sein oder Kosten, die das Erreichen des Unterziels verursacht (z.B. Kosten in Geldeinheiten oder Zeiteinheiten). Werte können auch die Eintrittswahrscheinlichkeit des Unterziels oder Attribute wie, dieser Knoten ist „möglich“ oder „unmöglich“, „erfordert spezielle Geräte“, oder „erfordert keine speziellen Geräte“ etc., sein (vgl. [Schneier2001]). Diese zusätzlichen Angaben helfen, die einzelnen Angriffsszenarien zu beurteilen und abzuschätzen, wie das System (sinnvoll) gegen Angriffe gesichert werden kann. Aus den Gewichten können so z.B. die durchführbaren Angriffe gegen das System bestimmt werden oder die Angriffe, die mit dem geringsten Risiko durchzuführen sind, die Angriffe, bei denen die wenigsten Sachkenntnisse notwendig sind etc.

Angriffsbäume stellen aber keine beweisbar vollständige oder beweisbar korrekte Analyse der Bedrohungen dar. Die Vollständigkeit eines Angriffsbaums ist im Wesentlichen abhängig von der Expertise des Erstellers und seiner Kreativität beim Erdenken der Angriffsmöglichkeiten. Bruce Schneier beschreibt dies sehr anschaulich mit dem folgenden Satz:

*„Bedrohungsanalyse ist größtenteils eine Ad-hoc-Angelegenheit. Sie denken über die Bedrohungen nach, bis Ihnen keine mehr einfällt. Dann sind sie verärgert und überrascht, wenn ein Angreifer sich eine Attacke ausgedacht hat, auf die sie nicht gekommen sind.“ ([Schneier2001], S. 309)*

Der Analyst lässt so viele mögliche Attacken gegen das Angriffsziel in die Analyse einfließen, bis ihm keine weiteren Möglichkeiten mehr einfallen oder eine gewünschte Detailebene erreicht ist. Damit deckt ein Angriffsbaum nicht zwangsläufig alle denkbaren Bedrohungen gegen das System ab. Gerade auch aus diesem Grund muss die Bedrohungsanalyse als ein permanenter Prozess angesehen werden. Um den Prozess der Bedrohungsanalyse zu unterstützen, können Angriffsbäume eingesetzt werden. Sie helfen, Sicherheitsentscheidungen zu treffen und neue Erkenntnisse über die Sicherheit eines Systems zu erhalten. Schneier führt aus, dass *„eine der Überraschungen, die sich bei dieser Art von Analyse ergeben, ist, dass gerade die Bereiche, welche die meisten für besonders verwundbar halten, dies normalerweise nicht sind.“* ([Schneier2001], S. 322), andere Arten von Bedrohungen dafür aber vor dem Erstellen eines Angriffsbaums bei der Betrachtung vernachlässigt wurden. Mit Angriffsbäumen können potentielle Attacken gegen ein System in einer strukturierten Form gesammelt und dokumentiert werden. Mit ihrer Hilfe kann das Wissen über Angriffe in einer wieder verwendbaren Form erfasst werden (vgl. [Schneier2001]). Das Wissen kann für weitere Analysen genutzt werden sowie bei der (Weiter)Entwicklung des Systems eingesetzt werden, um den Entwicklern potentielle Schwachstellen aufzuzeigen, und letztendlich auch zur Sensibilisierung der späteren Nutzer des Systems gegenüber Sicherheitsproblemen.

## 5.2. Angriffsziel Wiki

Für das in die Worksphere integrierte Wiki-Modul sind drei allgemeine Angriffsziele vorhanden:

1. Unberechtigtes Lesen von Wiki-Seiten.

2. Unberechtigtes Bearbeiten (Ergänzen, Ändern, Überschreiben, Löschen) von Wiki-Seiten.
3. Herbeiführen einer Systemstörung.

Angreifer können sowohl interne Angreifer sein – Nutzer und Administratoren des Wikis bzw. der Worksphere – als auch externe Angreifer, die versuchen Zugriff auf das System zu erhalten (um Daten unberechtigt zu lesen oder zu manipulieren) oder es zu stören.

Die Angriffe beeinflussen die Sicherheitsbedürfnisse Vertraulichkeit, Authentizität, Integrität und Verfügbarkeit (vgl. [Schneier2001] und [Döring2003a]).

- Vertraulichkeit:** Vertraulichkeit schützt Informationen/Daten vor dem unberechtigten Einblick durch Dritte. Vertraulichkeit kann so weit gehen, dass bereits die Existenz bestimmter Informationen für Außenstehende nicht bekannt sein darf.
- Authentizität:** Ist die Übereinstimmung der behaupteten Identität eines Kommunikationspartners mit seiner wahren Identität. „Der Benutzer joe-user ist tatsächlich die Person Joe-User“. Authentizität hat mit dem Ursprung der Daten zu tun (vgl. [Schneier2001]). Wer hat die Information veröffentlicht oder geändert? Kann ich der Person bzw. ihrer Identität vertrauen oder nicht?
- Integrität:** Bedeutet, dass Daten nicht unbefugt und unerkannt verändert werden dürfen. Integrität betrifft die Gültigkeit der Daten. Sind die Daten seit der Erstellung (unberechtigt) verändert worden?
- Verfügbarkeit:** Nutzungsberechtigte Personen können zur rechten Zeit am rechten Ort auf das System zugreifen – eine Kommunikation/Interaktion kann mit dem System stattfinden. Ist das System nicht oder nur eingeschränkt verfügbar, werde ich ganz oder teilweise an meiner Arbeit gehindert.

## **Angriffsziel: Unberechtigtes Lesen von Wiki-Seiten**

Ziel dieses Angriffs ist es, Zugriff auf das Wiki zu erhalten, um die im Wiki gespeicherten Informationen (unberechtigt) zu lesen oder weitere Aktionen mit den Daten auszuführen (sie z.B. an Dritte weiterzugeben). Der Angriffsbaum zu diesem Angriffsziel ist in

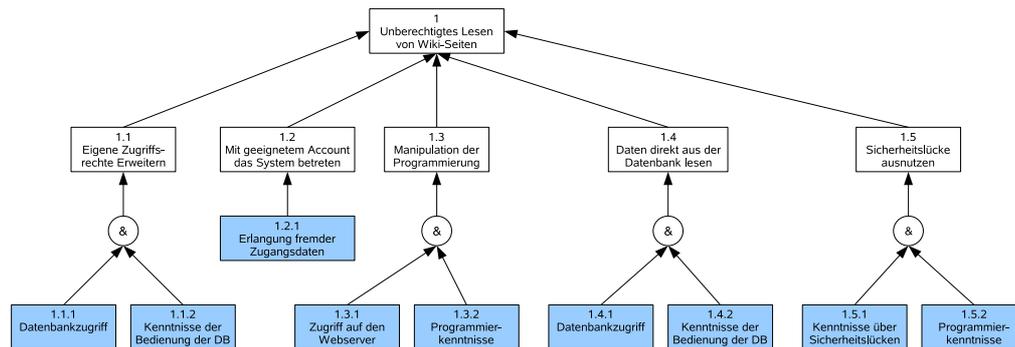


Abbildung 5.4.: Angriffsbau zum unberechtigten Lesen von Wiki-Seiten.

Abbildung 5.4 dargestellt. Ein erfolgreicher Angriff verletzt die Sicherheitsbedürfnisse Vertraulichkeit und Authentizität der Informationen/Daten des Wikis bzw. der Worksphere. Es ist nicht mehr sichergestellt, dass nur berechtigten Personen bestimmte Informationen zugänglich sind. Der Angriffsbau zeigt, dass der Angreifer zum Erreichen des Ziels vier verschiedene Möglichkeiten hat: (i) er versucht Zugang zum System über fremde Zugangsdaten zu erhalten (verfeinert im Angriffsbau 5.7); (ii) er versucht direkten Zugriff auf die Datenbank zu erhalten (verfeinert im Angriffsbau 5.8) und dort die Daten zu lesen oder seine Zugriffsrechte zu erweitern, dazu benötigt der Angreifer zusätzlich Kenntnisse über die Bedienung der Datenbank; (iii) er versucht über eine Manipulation der Programmierung Informationen im Wiki lesen zu können, dafür muss der Angreifer Zugang zum Webserver (verfeinert im Angriffsbau 5.9) und Kenntnisse der Programmierung haben; (iv) er versucht über die Ausnutzung einer Sicherheitslücke unberechtigt Seiten im Wiki zu lesen, hierzu muss der Angreifer Kenntnis entsprechender Sicherheitslücken haben und das Wissen, wie diese auszunutzen sind (im Baum als Programmierkenntnisse dargestellt).

## Angriffsziel: Unberechtigtes Bearbeiten von Wiki-Seiten

Der Angreifer versucht Zugriff auf das Wiki zu erhalten, mit dem Ziel, die im Wiki gespeicherten Informationen (unberechtigt) zu verändern, zu ergänzen oder zu zerstören. Abbildung 5.5 zeigt den Angriffsbau zu diesem Angriffsziel. Ein erfolgreicher Angriff verletzt neben der Vertraulichkeit und der Authentizität auch die Integrität und im Falle einer Zerstörung (Löschen, Überschreiben) von Daten die Verfügbarkeit des Wikis bzw. der Worksphere. Es ist nicht mehr sichergestellt, dass nur autorisierte Personen Daten bearbeiten können. Durch das Löschen bzw. die Zerstörung von Daten ist deren Verfüg-

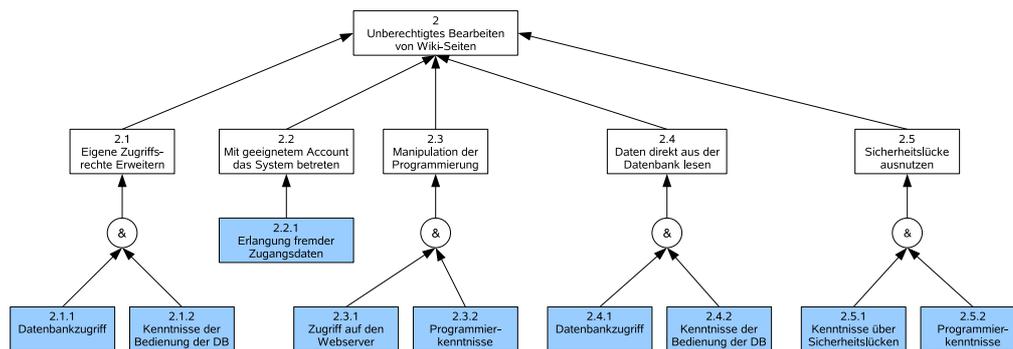


Abbildung 5.5.: Angriffsbaum zum unberechtigten Bearbeiten von Wiki-Seiten.

barkeit gestört. Die Möglichkeiten, dieses Ziel zu erreichen, sind die gleichen, wie beim Angriffsziel „Unberechtigtes Lesen von Wiki-Seiten“.

## Angriffsziel: Systemstörung

Ziel des Angriffs ist es, durch die Störung eines Teilsystems bis hin zur Überführung des Teilsystems in einen Fehlerzustand bzw. Ausfall oder durch das Einschleusen schadhaften Codes, die Verfügbarkeit des Wikis bzw. der Worksphere zu beeinflussen. Die Analyse betrachtet die Teilsysteme Webserver, Datenbankserver und Kommunikationsstruktur, da der Ausfall einer dieser Komponenten den Ausfall des Gesamtsystems nach sich zieht und den Fall, dass ein Angreifer versucht, schadhaften Code in das Wiki einzufügen. Der Angriffsbaum ist für die Teilsysteme Webserver, Datenbankserver und Kommunikationsstruktur nur in einer abstrakten Form dargestellt (siehe Abbildung 5.6). In Anhang A sind die Angriffsbäume enthalten, die Andre Döring im Rahmen seiner Diplomarbeit ([Döring2003a]) erstellt hat. In diesen Angriffsbäumen werden die Angriffsmöglichkeiten auf die Teilsysteme weiter ausgeführt. Für das Einschleusen fehlerhaften Codes bestehen drei Möglichkeiten: (i) Ein Angreifer nutzt eine Sicherheitslücke aus, hierzu muss er Kenntnis entsprechender Sicherheitslücken haben und das Wissen, wie diese ausgenutzt werden können (im Baum als Programmierkenntnisse dargestellt); und er muss Kenntnis haben, welchen schadhaften Code er in das Wiki einfügen kann; (ii) Ein Benutzer mit Rechten für das Bearbeiten des Wikis schleust schadhaften Code in das Wiki ein; (iii) Ein Benutzer, der nicht berechtigt ist Seiten des Wikis zu bearbeiten, schleust schadhaften Code ein, hierzu muss er sich unberechtigten Zugang zum Bearbeiten des Wikis verschaffen (verfeinert in Angriffsbaum 5.5).

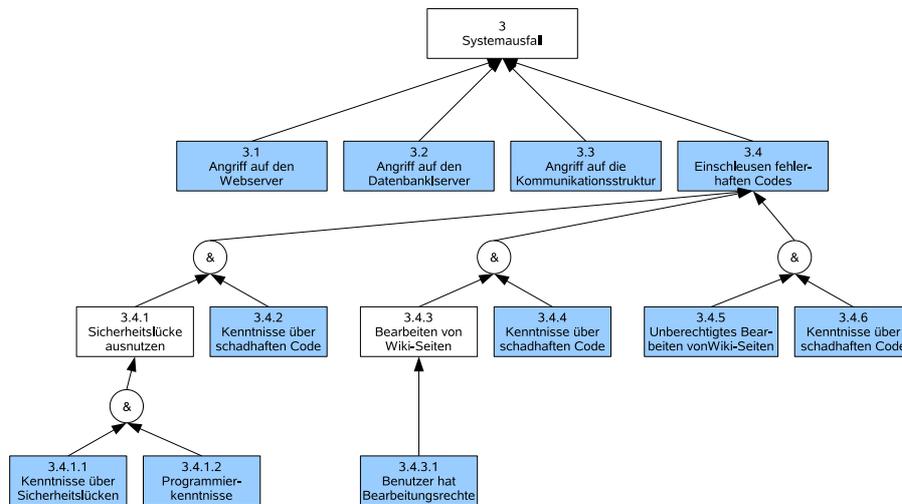


Abbildung 5.6.: Angriffsbaum zum Herbeiführen eines Systemausfalls.

## Angriffsziel: Erlangen fremder Zugangsdaten

Bei den vorhergehenden Angriffsbäumen taucht immer wieder als eine Bedrohungsmöglichkeit das „Erlangen fremder Benutzerdaten auf“. Daher wird diese Bedrohungsmöglichkeit hier als separater Angriffsbaum dargestellt (Abbildung 5.7) und weiter verfeinert. In den Angriffsbäumen ist an den entsprechenden Stellen der Knoten „Erlangen fremder Zugangsdaten“ durch den hier gezeigten Angriffsbaum zu ersetzen. Um fremde Zugangsdaten zu erlangen hat der Angreifer die folgenden Möglichkeiten:

- **Social Engineering:**  
Über *Social Engineering* versucht der Angreifer die Zugangsdaten eines Benutzers zu erhalten. Es wird versucht, den Benutzer dazu zu verleiten oder ihn zu überreden, die Zugangsdaten preiszugeben. Social Engineering ist eine sehr wirksame und oft unterschätzte Methode, Informationen zu sammeln. Bruce Schneier schreibt in [Schneier2001] dazu: „*Social Engineering ist sehr wirksam; es umgeht Kryptographie, Computersicherheit, Netzwerksicherheit und alles andere, was an Technologie aufgeboten wird. Es geht direkt auf das schwächste Glied in jedem Sicherheitssystem zu: den armen Menschen, der lediglich seine Arbeit verrichtet und gerne aushilft, wenn er kann.*“.
- **Zugangsdaten ausspionieren:**  
Der Angreifer kann versuchen auf Client-Seite (dem PC, an dem der Benutzer arbeitet) die Daten auszuspionieren. Hier sind vielfältige Möglichkeiten denkbar. Im

Angriffsbaum sind die Möglichkeiten, dem Benutzer einfach über die Schulter zu schauen oder die Tastaturanschläge zu überwachen (über spezielle Vorrichtungen oder eine Kamera, die die Eingaben des Benutzers aufzeichnet) aufgeführt. Es sind viele weitere Möglichkeiten denkbar, die sich nicht nur auf eine „physikalische“ Spionage beschränken. Ein Angreifer kann versuchen den Client-Computer zu manipulieren (über Trojaner, Viren etc.), um so über ein Programm Daten auf dem Client-PC zu sammeln. Die Vielzahl denkbarer Möglichkeiten ist in dem Angriffsbaum nicht explizit aufgeführt. Wichtig ist die Feststellung, dass es eine große Zahl Möglichkeiten gibt, die Zugangsdaten auszuspionieren.

- Offene Benutzersitzung übernehmen:  
Der Benutzer vergisst, sich aus der Workspace auszuloggen, so dass der Angreifer direkten Zugriff über den Benutzeraccount auf das System hat.
- Datenbankzugriff:  
Der Angreifer versucht Zugriff auf die Datenbank zu erhalten (verfeinert im Angriffbaum 5.8), um dort die Zugangsdaten eines Benutzers auszulesen. Dazu benötigt der Angreifer zusätzlich Kenntnisse über die Bedienung der Datenbank.
- Brute Force Angriff:  
Mit einem Brute Force Angriff kann der Angreifer versuchen, die Zugangsdaten eines Benutzers zu ermitteln. Zu unterscheiden ist ein Brute Force Angriff mit Kenntniss eines Accountnamens – der Angreifer versucht dann, das Passwort des Benutzers zu ermitteln (z.B. über das Generieren verschiedener Passwörter oder einen Lexikonangriff) – oder ohne Kenntnis eines Accountnamens – der Angreifer versucht einen Accountnamen und das zugehörige Kennwort zu ermitteln.
- Ausnutzen einer Sicherheitslücke:  
Über die Ausnutzung einer Sicherheitslücke kann versucht werden, an die Zugangsdaten eines Benutzers zu gelangen. Hierzu muss der Angreifer Kenntnis entsprechender Sicherheitslücken haben und das Wissen, wie diese auszunutzen sind (im Baum als Programmierkenntnisse dargestellt).
- Netzwerkverkehr abhören:  
Der Angreifer kann versuchen, den Netzwerkverkehr abzuhören (z.B. mit Netzwerk-Sniffen) und aus den Daten die Zugangsinformationen eines Benutzers zu ermitteln. Das Abhören des Netzwerks erfordert physikalischen Zugang zum Netzwerk.

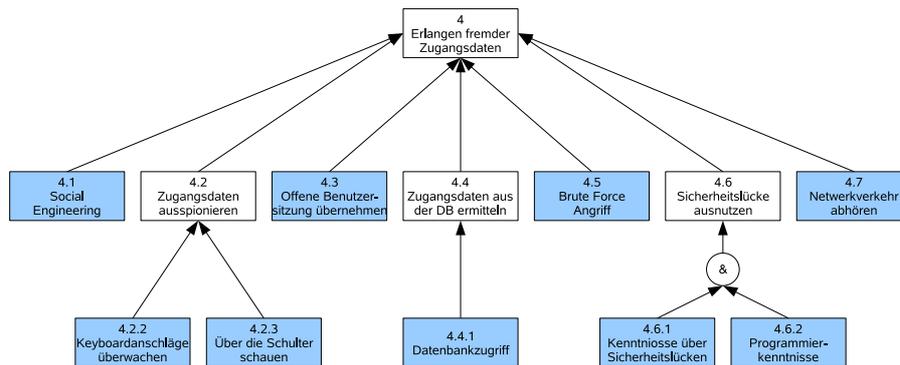


Abbildung 5.7.: Angriffsbaum zum Erlangen fremder Zugangsdaten.

## Angriffsziele: Datenbankzugriff und Webserverzugriff

Die Angriffsziele Datenbankzugriff und Webzugriff sind Unterziele in den Angriffsbäumen zum unberechtigten Lesen und Bearbeiten von Wiki-Seiten. Die hier skizzierten Angriffsbäume (Abbildung 5.8 und Abbildung 5.9) sind in den Angriffsbäumen an den entsprechenden Stellen einzusetzen. Der Angreifer hat verschiedene Möglichkeiten zum Erreichen des Ziels: (i) er versucht Zugang zum System über fremde Zugangsdaten zu erhalten (verfeinert im Angriffsbaum 5.7); (ii) er versucht über die Ausnutzung einer Sicherheitslücke Zugang zu erhalten, hierzu muss der Angreifer Kenntnis entsprechender Sicherheitslücken haben und das Know-how haben, diese ausnutzen zu können; (iii) der Benutzer ist bereits Administrator und hat in seiner Rolle per se Zugriff auf das System<sup>3</sup>.

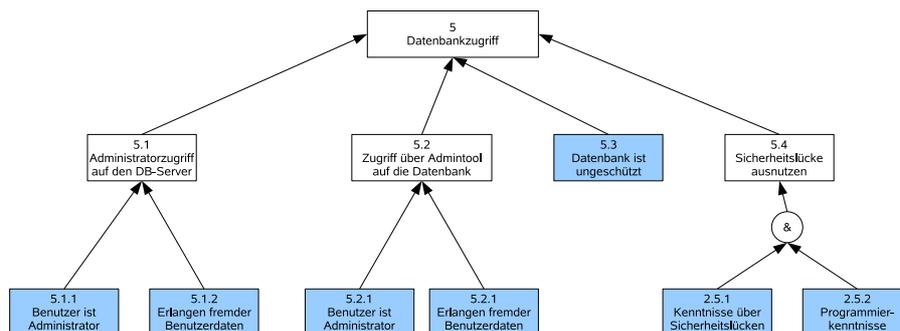


Abbildung 5.8.: Angriffsbaum zum Zugriff auf den Datenbankserver.

<sup>3</sup>Dies mag als Angriffsmöglichkeit seltsam erscheinen, hat aber den Hintergrund, dass auch die Administratoren unberechtigten Zugriff auf die Systeme nehmen können. Über den Datenbankzugriff hat der Administrator Zugang zu Informationen, die er evtl. gar nicht lesen bzw. bearbeiten darf.

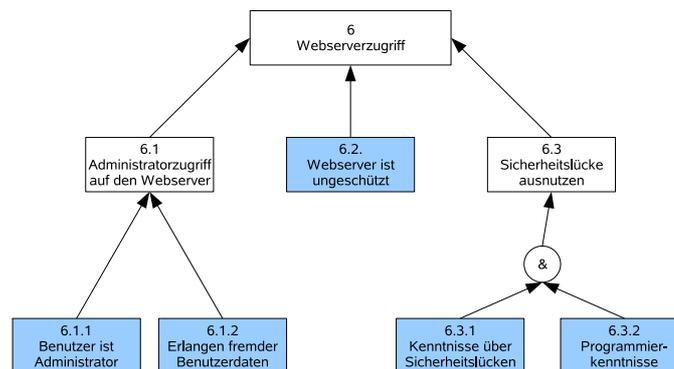


Abbildung 5.9.: Angriffsbau zum Zugriff auf den Webserver.

### 5.3. Folgerungen und Empfehlungen

Aus den zur Bedrohungsanalyse erstellten Angriffsbäumen (Abbildungen 5.4, 5.5 und 5.6) können die folgenden Angriffsszenarien abgeleitet werden:

- (1) Erlangen fremder Zugangsdaten
- (2) Datenbankzugriff (in Zusammenhang mit Programmierkenntnissen)
- (3) Zugriff auf den Webserver (in Zusammenhang mit Programmierkenntnissen)
- (4) Ausnutzen von Sicherheitslücken (d.h. die Kenntnis von Sicherheitslücken in Zusammenhang mit Programmierkenntnissen)
- (5) Einfügen schadhaften Codes. Hierzu muss der Angreifer berechtigt oder unberechtigt Bearbeitungsrechte für das Wiki haben oder eine Sicherheitslücke ausnutzen und zusätzlich Kenntnisse über schadhaften Code haben.
- (6) Unter Beachtung der Angriffsbäume, die zum Ausfall eines Teilsystems führen (siehe Anhang A): Physikalischer Zugang zum Webserver, zum Datenbankserver oder zum Netzwerk.

Eine weitere Betrachtung der Angriffsmöglichkeiten (1), (2), (3) und (5) zeigt, dass diese Angriffe über die folgenden Möglichkeiten durchführbar sind:

- (i) Social Engineering

- (ii) Ausspionieren der Zugangsdaten
- (iii) Offene Benutzersitzung übernehmen
- (iv) Netzwerkverkehr abhören
- (v) Brute Force Angriff
- (vi) Ausnutzen von Sicherheitslücken (d.h. die Kenntnis von Sicherheitslücken in Zusammenhang mit Programmierkenntnissen)
- (vii) Benutzer hat Bearbeitungsrechte
- (viii) Benutzer ist Administrator (bzw. hat Administratorrechte)

Aus diesen Kenntnissen können die folgenden allgemeinen Empfehlungen abgeleitet werden:

**Die Nutzer des Wikis bzw. der Worksphere müssen gegenüber Sicherheitsproblemen sensibilisiert werden.**

Die Benutzer sollten über mögliche Sicherheitsprobleme informiert und sensibilisiert werden. Dazu gehört auch, ihnen Informationen zu geben, wie sie die Sicherheitsprobleme vermeiden können (dies schließt *alle* Benutzer ein, also auch die Administratoren). Wichtigster Punkt ist die Geheimhaltung der Zugangsdaten. Benutzer müssen über Social Engineering unterrichtet werden und über die Möglichkeiten, die vorhanden sind, ihre Zugangsdaten auszuspionieren, um selber bewusst darauf achten zu können, Sicherheitsprobleme die hierdurch entstehen, zu vermeiden. Das Problem der offenen Benutzersitzungen versucht die Worksphere über den in Kapitel 3.2.4 erwähnten Auto-Logout-Mechanismus abzuschwächen: ist ein Benutzer für eine gewisse (festlegbare) Zeitdauer inaktiv, muss er sich erneut authentifizieren. Der Mechanismus stellt aber bei weitem keinen ausreichenden Schutz für das Problem der offenen Benutzersitzung dar. Den Benutzern muss klar sein, dass sie sich zum Ende jeder Sitzung explizit aus der Worksphere ausloggen sollten, um zu verhindern, dass ein Dritter über ihren Account Zugriff auf die Worksphere hat.

**Zur Absicherung des Systems müssen geeignete Maßnahmen getroffen werden.**

Es empfiehlt sich, die Daten verschlüsselt zu übertragen, d.h. der Webserver, über den die Skripte der Worksphere aufgerufen werden, sollte Verschlüsselung (z.B. Secure-HTTP) unterstützen, damit Daten nicht unverschlüsselt an Dritte gelangen. Dies betrifft im Fall

der Worksphere insbesondere die Zugangsdaten, die ohne Einsatz einer Verschlüsselung im Klartext übertragen werden. Ein einfaches Abhören der Netzwerkkommunikation reicht also aus, um an gültige Zugangsdaten zu gelangen. Daneben sollten weitere Maßnahmen getroffen werden, wie z.B. der Einsatz einer Firewall, um die Server (Webserver, Datenbankserver) zu schützen. Zur Sicherstellung der Verfügbarkeit sollten regelmäßige Backups der Daten durchgeführt werden und evtl. auch für Redundanz gesorgt werden. Zu den Maßnahmen gehört auch, dass die Administratoren der Systeme regelmäßig Informationen zu relevanten Sicherheitslücken (dies betrifft Sicherheitslücken des eingesetzten Server-Betriebssystems, der eingesetzten Webserver-Software, der Datenbank, der Skriptsprache PHP etc.) einholen *und* diese auch schließen. Die Maßnahmen, die getroffen werden können, sind noch wesentlich umfangreicher und weitreichender, daher soll an dieser Stelle nur eine kleine Auswahl skizziert werden. Wesentlich ist aber, zu verstehen, dass Angriffe *nicht* verhindert werden können. Es wird immer eine findige Person geben, die die getroffenen Sicherheitsmaßnahmen umgehen kann. Es reicht nicht aus, eine Firewall zu installieren, kryptographische Protokolle zur verschlüsselten Übertragung der Daten zu nutzen und sich dann in Sicherheit zu wägen. Angriffe und Angriffsversuche müssen auch erkannt werden. Das heißt, dass die Logs, die die Firewall, der Webserver, der Datenbankserver, der Router, die Worksphere etc. produzieren, auch angeschaut und interpretiert werden müssen. Angriffe hinterlassen immer Spuren (und seien sie noch so klein). Diese müssen erkannt und aufgespürt werden. Versuche, Sicherheitslücken auszunutzen, und Brute Force Angriffe lassen sich im allgemeinen gut durch eine Analyse der Log-Dateien erkennen. Es gilt auch, je eher ein Angriff erkannt wird, desto schneller kann darauf reagiert werden. Eine Hilfe bei der Erkennung von Angriffen können Einbruchserkennungssysteme (Intrusion Detection Systems – IDS) sein, die das Netzwerk und die Systeme überwachen und versuchen Angriffe automatisiert zu entdecken. Festzuhalten gilt trotz aller technischen Sicherungsmaßnahmen aber: *„Sicherheit ist ein Prozess und kein Produkt“* [Schneier2001]. Sicherheit umfasst vorbeugende technische Maßnahmen, aber auch Erkennungs- und Reaktionsprozesse bezüglich neuer und alter Sicherheitslücken und (versuchten) Angriffen.

**Die „inneren“ Bedrohungen durch Benutzer und Administratoren dürfen nicht vernachlässigt werden.**

Wie die Punkte (vii) und (viii) zeigen, gehen auch von den Benutzern und den Administratoren des Wikis bzw. der Worksphere Bedrohungen aus. Bruce Schneier beschreibt dies mit: *„Ein böswilliger Insider ist ein gefährlicher und heimtückischer Gegner. Er befindet sich bereits im Inneren des Systems, das er anzugreifen gedenkt, sodass er eventuell vorhandene äußere Abwehrmaßnahmen rund um das System ignorieren kann.“*

[Schneier2001]. Ein Administrator hat per se sehr weitreichende Zugangsberechtigungen und gilt im allgemeinen auch als vertrauenswürdig; diese privilegierte Rolle kann er auch missbrauchen. Mit Sicherheitsmaßnahmen wie der Einrichtung einer Firewall oder eines Einbruchserkennungssystems etc. wird im Wesentlichen versucht, externe Angreifer abzuwehren bzw. externe Angriffe zu erkennen. Gegen Insider sind sie aber so gut wie machtlos. Die Wahrscheinlichkeit, dass ein Insider das System angreift ist vielleicht geringer, dafür ist das System aber gegenüber solchen Angriffen verwundbarer (insbesondere gegenüber Angriffen, die von einem Nutzer mit Administratorrechten ausgehen). Für Insider ist es einfacher Kenntnisse über Sicherheitslücken und über die Systemarchitektur zu erlangen und diese Kenntnisse dann für Angriffe auszunutzen.

### **Sicherheitsschwächen bei der Implementierung der Software müssen bedacht werden**

Ein Großteil der Sicherheitsprobleme basiert auf fehlerhaften Programmen bzw. fehlerhaften Implementierungen (vgl. [Schneier2001]). Fehler in Programmen bilden oft die Sicherheitslücken, die zum Angriff auf ein System ausgenutzt werden können. Klassisches Beispiel sind die sogenannten Pufferüberläufe (Buffer Overflows), die immer wieder Sicherheitslücken in Programmen öffnen. Die Worksphere und das in dieser Arbeit implementierte Wiki-Modul bilden keine Ausnahme. Die Worksphere ist z.B. nicht in besonderem Maße gegen das sogenannte *Session-Hijacking* (das Entführen einer Session) geschützt<sup>4</sup>. Der in dieser Arbeit implementierte Prototyp des Wiki-Moduls enthält ebenfalls eine bereits bekannte Sicherheitslücke: Benutzer haben die Möglichkeit auch HTML-Code in das Wiki einzugeben (vgl. Kapitel 4.4.2, 4.4.3.6). Diese Möglichkeit ist den Benutzern eingeräumt bzw. bei der Konfiguration eines Wikis kann festgelegt werden, ob in dem Wiki HTML-Code zugelassen ist oder nicht. Wird das Wiki so konfiguriert, dass HTML-Code zugelassen ist, können die Benutzer ohne weiteres schadhafte Code (z.B. mittels Java-Script) in das Wiki einschleusen. Die Betreiber bzw. Administratoren der Worksphere müssen sich also im Klaren darüber sein, dass sie explizit eine Sicherheitslücke öffnen, wenn sie Benutzern die Möglichkeit einräumen HTML-Code in das Wiki einzugeben. Des Weiteren findet bisher noch keine Validierung der Eingaben über das Eingabeformular für Wiki-Seiten und im Suchformular des Wiki-Moduls statt, so dass auch hier noch weitere Möglichkeiten bestehen, schadhafte Code (z.B. über SQL-Injection<sup>5</sup>) einzuschleusen.

---

<sup>4</sup>Die Worksphere nutzt Sessions, um die Authentifizierung eines Benutzers über verschiedene Seitenaufrufe hinweg zu transportieren (vgl. Kapitel 3.2.3 und 3.2.4).

<sup>5</sup>Mit SQL-Injection kann der Angreifer Anfragen, die an die Datenbank gesendet werden, dadurch modifizieren, dass er mit den Eingaben in das Eingabefeld „experimentiert“

## 6. Zusammenfassung und Ausblick

Der in dieser Arbeit implementierte Prototyp eines Wiki-Moduls für die Kommunikationsplattform Worksphere setzt alle im Anforderungskatalog (siehe Kapitel 4.1) aufgestellten Anforderungen um. Ein besonderer Augenmerk wurde auf eine schlanke und übersichtliche Implementation gelegt, die nur die wesentlichen Funktionen eines Wikis umsetzt. Der Prototyp kann damit als ideale Basis für Weiterentwicklungen in Absprache mit den Anwendern dienen.

Die gewählte Form der Implementierung schränkt das Wiki-Modul nicht auf bestimmte Anwendungsgebiete ein. Durch die vollständige Integration des Wiki-Moduls in die Authentifikation und Rechteverwaltung der Worksphere können sowohl für alle Benutzer offene Wikis als auch nur für bestimmte Benutzergruppen zugängliche Wikis innerhalb der Plattform eingerichtet werden. Damit ist das Wiki-Modul für die verschiedensten Einsatzmöglichkeiten geeignet, als Arbeitswerkzeug für Projektgruppen zur Dokumentation des Projektes oder zum projektinternen Austausch von Informationen, als Wissenssammlung oder Wissensmanagementtool, als Content Management System, um Kursmaterialien in einem E-Learning-Einsatz zur Verfügung zu stellen, als Gruppenarbeitstool zur gemeinsamen Arbeit an einem Text etc.

Durch die Kapselung der Darstellungsfunktionen bzw. des einfachen zeilenbasierten Parsers für die Wiki-Syntax in die Datei `render.inc.php`, kann der Parser auch für andere Elemente in der Worksphere genutzt werden. Damit ist es möglich, dass Anwender auch in Schwarzen Brettern, Foren, Dateibereichen etc. die Wiki-Syntax nutzen. Damit stehen neue Formatierungsmöglichkeiten für Beiträge in diesen Plattformelementen zur Verfügung.

Seit dem 17. März 2006 wird der Prototyp des Wiki-Moduls in einer Installation der Worksphere zur Dokumentation der Konzeptentwicklung eines universitären Qualitätsmanagement Projekts<sup>1</sup> eingesetzt. Das Wiki befindet sich noch im Aufbau bzw. die ersten Inhalte der Dokumentation werden noch eingepflegt. Es wird zur Zeit von drei Personen

---

<sup>1</sup>Qualität in der Lehre, Universität Bielefeld, <http://www.qm-unibi.nothbaum.com>

intensiv genutzt. Erste Rückmeldungen dieser drei Anwender sind durchweg als positiv zu bewerten: „liegt gut in der Hand“, „ist einfach zu bedienen“, „zeigt Potential für andere Anwendungen“<sup>2</sup>.

Die Integration des Wiki-Moduls in weitere Installationen der Worksphere ist für Ende April 2006 geplant. So soll das Wiki-Modul in die Alumni-Plattform des Fernstudiengangs Angewandte Gesundheitswissenschaften, Universität Bielefeld<sup>3</sup> und in die auf der Worksphere basierenden E-Learning-Plattformen des Fernstudiengangs Angewandte Gesundheitswissenschaften<sup>4</sup> eingefügt werden. Weiterhin ist geplant, die Online-Hilfe der Worksphere, die bisher über statische Webseiten realisiert ist, in allen Installationen der Worksphere in ein Wiki zu überführen.

Ein geäußerter Erweiterungswunsch von Testern des Wiki-Moduls ist die Integration einer Druckfunktion, mit der es möglich ist, über eine Schaltfläche im Wiki (z.B. in der Navigationsleiste) den Inhalt einer Wiki-Seite auszudrucken, anstatt die Druckfunktion des Browsers über dessen Menü aufzurufen.

Andere interessante Erweiterungen und Weiterentwicklungen für das Wiki-Modul sind die Integration einer Funktion, mit der sich Bilder komfortabel in eine Wiki-Seite einbinden lassen. Der umgesetzte Prototyp des Wiki-Moduls ist zur Zeit auf reinen Text beschränkt. Bilder können bisher nur über den „Umweg“ der Einbindung mit dem HTML-Tag `<img>` in eine Wiki-Seite eingefügt werden. Diese Erweiterung ist bereits für die nahe Zukunft geplant und wird voraussichtlich bis Mai/Juni 2006 umgesetzt sein. Auch überlegt werden kann die Integration einer sogenannten *Diff-Funktion*, die die Unterschiede zweier Versionen einer Wiki-Seite anzeigt.

In Zusammenhang mit der Diff-Funktion ist ein interessantes Forschungsprojekt von IBM Research zu erwähnen: History Flow<sup>5</sup>. Die Visualisierungsapplikation History Flow stellt die Entstehungsgeschichte von Wiki-Seiten in Form von Diagrammen dar. Mit History Flow ist es möglich, über einen längeren Zeitraum zu untersuchen, welchen Beitrag einzelne Autoren zu einer Seite geleistet haben, welchen zeitlichen Bestand einzelne Beiträge zu einer Seite haben etc. History Flow ist in Java implementiert und kann über Plug-ins an verschiedene Wiki-Engines angepasst werden. Seit 2005 steht History Flow zum Download zur Verfügung<sup>6</sup>.

---

<sup>2</sup>Zitate aus einem persönlichen Gespräch mit Norbert Nothbaum am 22. März 2006

<sup>3</sup><http://www.alumni-public-health-bielefeld.nothbaum.com>

<sup>4</sup><http://www.jg9.fag.nothbaum.com>, <http://www.jg10.fag.nothbaum.com>

<sup>5</sup><http://www.research.ibm.com/history>

<sup>6</sup><http://www.alphaworks.ibm.com/tech/historyflow>

Die in dieser Arbeit durchgeführte Bedrohungsanalyse für das Wiki-Modul zeigt mögliche Angriffsszenarien gegen das implementierte Wiki-Modul bzw. die Worksphere. Mit Kenntnis dieser Angriffsszenarien können und sollen geeignete Maßnahmen getroffen werden, um das Wiki-Modul bzw. die Worksphere sicherer zu machen. Kapitel 5.3 gibt hierzu Hinweise und Empfehlungen. Insbesondere gilt es aber auch, die offen gelegten Sicherheitsschwächen in der Implementierung des Wiki-Modul-Prototypen zu beheben. Dies ist für die nächsten Zwischenversionen des Wiki-Moduls geplant. Die durchgeführte Bedrohungsanalyse ist eine gute Grundlage für weitere detaillierte Bedrohungsanalysen und kann als Basis einer Sicherheitsanalyse dienen. Die durchgeführte Analyse lässt sich auf die anderen Module der Worksphere (Chats, Schwarze Bretter, Foren etc.) übertragen. Damit ist es möglich in mehreren Schritten eine Bedrohungsanalyse für das Gesamtsystem Worksphere durchzuführen.



# A. Angriffsbäume

Dieser Anhang enthält leicht modifizierte Versionen der Angriffsbäume aus der Diplomarbeit von Andre Döring [Döring2003a], auf die in Kapitel 5.2 verwiesen wird.

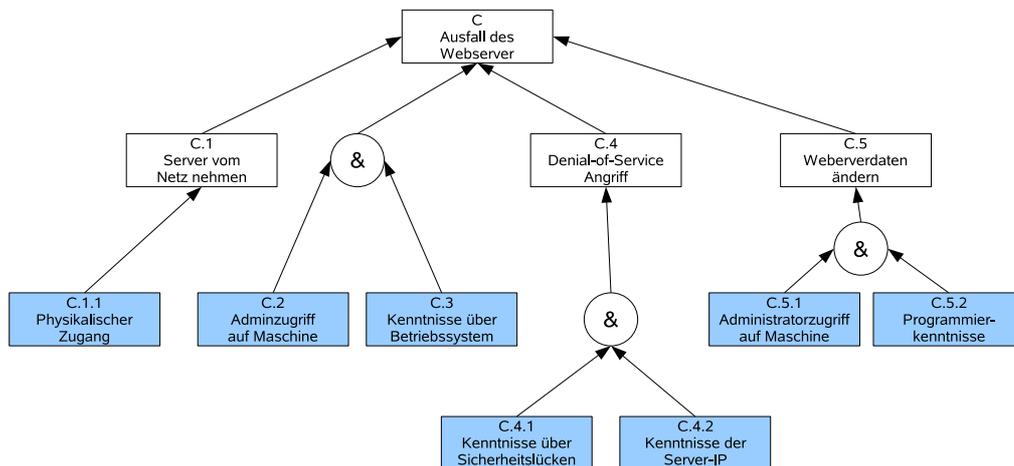


Abbildung A.1.: Angriffsbau zum Angriff auf den Webserver.

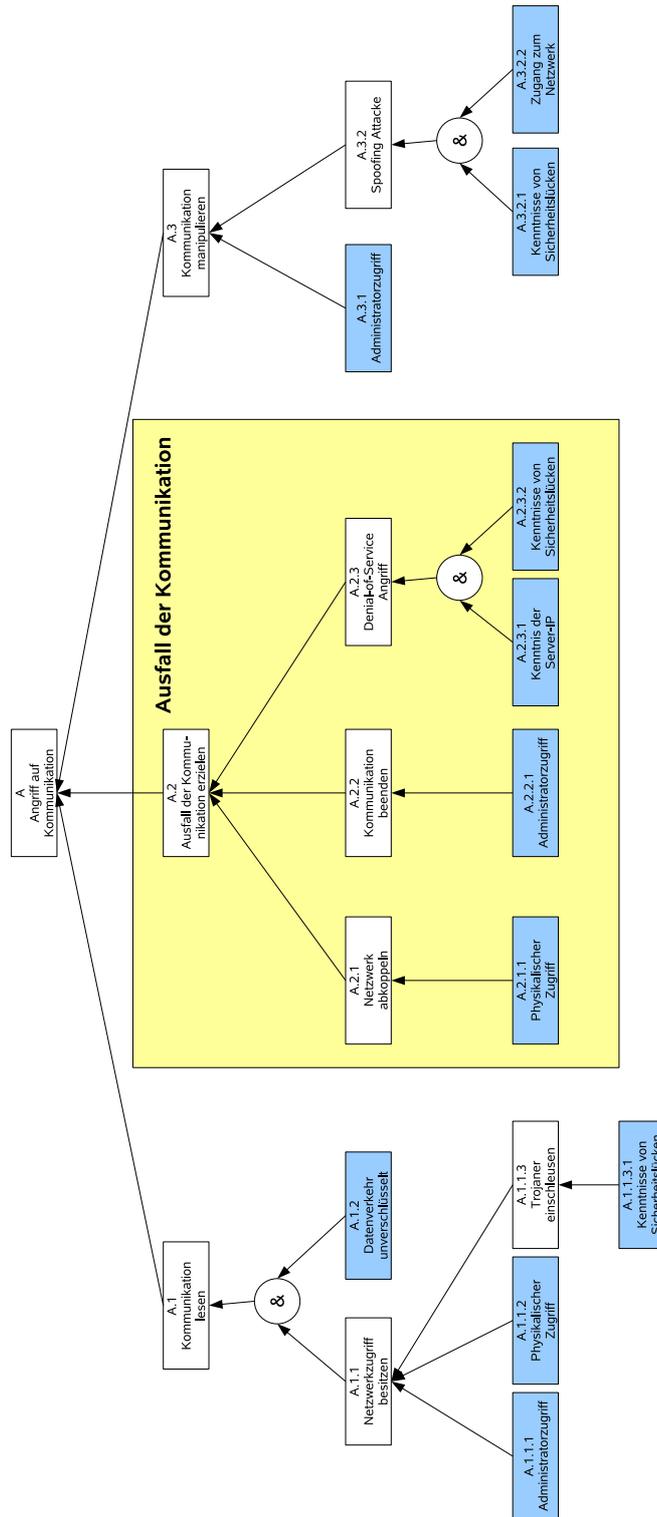


Abbildung A.2.: Angriffsbaum zum Angriff auf Kommunikationsstrukturen.

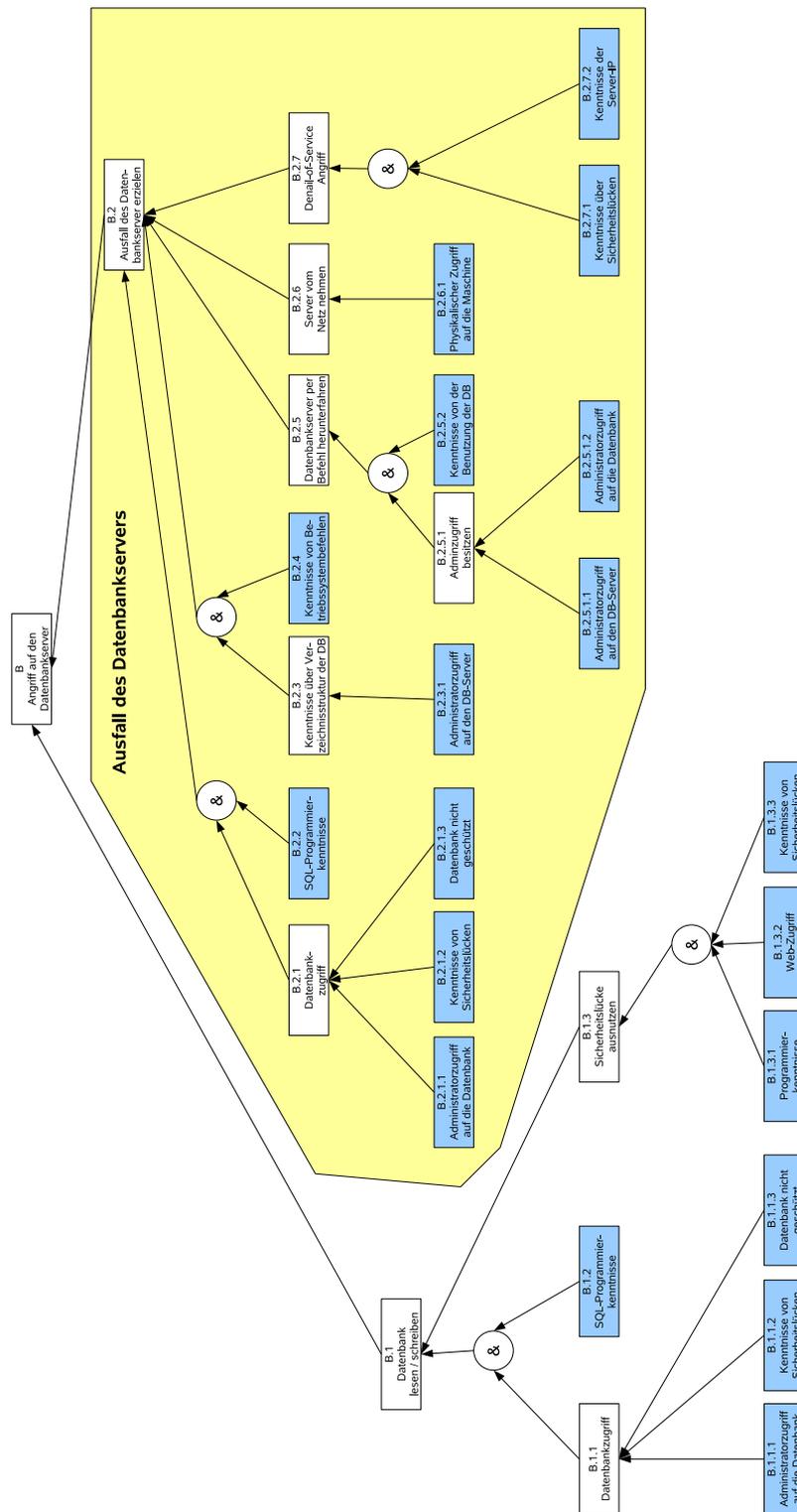


Abbildung A.3.: Angriffsbaum zum Angriff auf den Datenbankserver.



## B. Inhalt der CD-ROM

Auf der dieser Arbeit beiliegenden CD sind enthalten:

- Der Text der Arbeit als PostScript- und PDF-Datei (im Verzeichnis `Diploarbeit`)
- Die in dieser Arbeit erstellte neue Version der Worksphere mit integriertem Wiki (im Verzeichnis `Worksphere_mit_Wiki`).
- Kurzanleitung zur Installation der Worksphere als PDF-Datei (im Verzeichnis `Dokumentation`).
- Die originale Version 7.2d der PHPLib (im Verzeichnis `PHPLib`).



# Literaturverzeichnis

- [Adams1979] Douglas Adams. *The Hitchhiker's Guide to the Galaxy*. Pan Books Ltd., 1979.
- [Adams1981] Douglas Adams. *Per Anhalter durch die Galaxis*. Roger und Bernhard, 1981.
- [Adams2000] Douglas Adams. *My Vision for h2g2*. <http://bbc.co.uk/dna/h2g2/A550955>, April 2000 (zuletzt eingesehen März 2006).
- [Berners-Lee] Tim Berners-Lee. *The WorlWideWeb browser*. <http://www.w3.org/People/Berners-Lee/WorldWideWeb.html> (zuletzt eingesehen, März 2006).
- [Berners-Lee2000] Tim Berners-Lee & Mark Fischetti. *Weaving the Web. The original design and ultimate destiny of the World Wide Web*. Harper Business, 2000.
- [Burry2005] Jane Burry, Andrew Burrow, Robert Amor & Mark Burry. *Shared Design Space. The contribution of augmented wiki hypertext to design collaboration*. CAAD Futures. Wien, Österreich. 20-22. Juni 2005. Seite 217-226. <http://www.cs.auckland.ac.nz/~trebor/papers/BURR05.pdf> (zuletzt eingesehen März 2006).
- [Bush1945] Vannevar Bush. *As we may think*. Atlantic Monthly. Juli 1945. <http://www.theatlantic.com/194507/bush> (zuletzt eingesehen März 2006)
- [Cunningham2006a] Ward Cunningham. *WikiWikiWeb. Invitation to the Patterns List*. <http://c2.com/cgi/wiki?InvitationToThePatternsList> (zuletzt eingesehen März 2006).

- [Cunningham2006b] Ward Cunningham. *WikiWikiWeb. Wards Wiki Tenth Anniversary*. <http://c2.com/cgi/wiki?WardsWikiTenthAnniversary> (zuletzt eingesehen März 2006).
- [Cunningham2006c] Ward Cunningham. *WikiWikiWeb. Wiki History*. <http://c2.com/cgi/wiki?WikiHistory> (zuletzt eingesehen März 2006).
- [Cunningham2006d] Ward Cunningham. *WikiWikiWeb. Wiki Wiki Hyper Card*. <http://c2.com/cgi/wiki?WikiWikiHyperCard> (zuletzt eingesehen März 2006).
- [Currie2000] Crawford Currie. *TWiki success story of Motorola*. 1. Juni 2000. <http://twiki.org/cgi-bin/view/Main/TWikiSuccessStoryOfMotorola> (zuletzt eingesehen März 2006).
- [Dix1998] Alan Dix, Janet Finlay, Gregory Abowd & Russel Beale. *Human-Computer Interaction*. Prentice Hall, 2nd edition, 1998.
- [Döring2003a] Andre Döring. *tQuest. Spezifikation und Implementation eines sicheren Lernerfolgskontrollmoduls für CSCL-Werkzeuge*. Diplomarbeit, Universität Bielefeld, Technische Fakultät, Oktober 2003. RVS-Dip-03-03, <http://www.rvs.uni-bielefeld.de/publications/>.
- [Ebersbach2005] Anja Ebersbach, Markus Glaser, & Richard Heigl. *WikiTools. Kooperation im Web*. Springer, Berlin Heidelberg, 2005.
- [Ellermann2002] Martin Ellermann. *Design und Entwicklung einer vorlesungsbegleitenden Übungsplattform*. Diplomarbeit, Universität Bielefeld, Technische Fakultät, September 2002. RVS-Dip-02-01, <http://www.rvs.uni-bielefeld.de/publications/>.
- [Elmasri2002] Ramez Elmasri & Shamkant S. Navathe. *Grundlagen von Datenbanksystemen*. Pearson Studium, 3. überarbeitete Auflage, 2002.
- [Ernst2005] Astrid Ernst. *Konstruktivistisch orientierte Inhaltsaufbereitung für Hypermedia: Entwicklung einer modellhaften Vorgehensweise*. Verlag Franzbecker, 2005.
- [e-teaching2006] e-teaching.org. *Definition des Begriffs Chat auf e-teaching.org*. 2006. <http://www.e-teaching.org/didaktik/gestaltung/kommunikation/chat/> (zuletzt eingesehen März 2006).

- [Fiebig2005] Henriette Fiebig (Hrsg.). *Wikipedia. Das Buch*. Wiki Press 1, Zenodot Verlagsgesellschaft, Berlin. 2. Auflage, 2005.
- [Frauenfelder2004] Mark Frauenfelder. *Sir Tim Berners-Lee*. Interview, in *Technology Review*, Oktober 2004.  
<http://www.technologyreview/articles/04/10/frauenfelder1004.aps>  
(zuletzt eingesehen März 2006).
- [Huhmann2002] Jochem Huhmann. *Schnell, schnell. Content Management für Eilige*. *iX*, 10:84ff., 2002. <http://www.heise.de/ix/artikel/2002/10/084>,  
(zuletzt eingesehen März 2006).
- [Huhmann2004] Jochem Huhmann. *Noch schneller. Welches Wiki für welchen Zweck?* *iX*, 4:74–78, 2004.
- [Huseby2004] Sverre H. Huseby. *Sicherheitsrisiko Webanwendung. Wie Web-Programmierer Sicherheitslücken erkennen und vermeiden*. dPunkt.Verlag, 2004.
- [Kerres2003] Michael Kerres & Britta Voß (Hrsg.). *Digitaler Campus. Vom Medienprojekt zum nachhaltigen Medieneinsatz in der Hochschule*. *Medien in der Wissenschaft*, Band 24. Waxmann Verlag, Münster, 2003.
- [Kleinz2003] Torsten Kleinz. *Schreibrecht für alle! Mit Wikis Sites gemeinschaftlich betreuen*. *c't – Magazin für Computertechnik*, 2:176–181, 2003.
- [Kluge2002] Jürgen Kluge, McKinsey & Wolfram Stein. *Wissen entscheidet*. Redline Wirtschaft bei Ueberreuter, 2002.
- [Koschmann1996] Timothy Koschmann (Hrsg.). *CSCL: Theory and practice of an emerging paradigm*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 1996.
- [Krause2001] Jörg Krause. *PHP – Grundlagen und Profiwissen*. Hanser, 2001.
- [Köhntopp1999] Kristian Köhntopp. *PHPLIB*. In [Schmid1999], 1999. <http://kris.koehntopp.de/artikel/php-lib-deutsch> (zuletzt eingesehen März 2006).

- [Köhntopp2002] Kristian Köhntopp, Boris Erdmann, Sascha Schuhmann, & Tarique Sani (Hrsg.). *PHPLib Dokumentation*, 2002. <http://www.sanisoft.com/phplib/manual> (zuletzt eingesehen März 2006).
- [Kurose2002] James F. Kurose & Keith W. Ross. *Computernetze. Ein Top-Down-Ansatz mit Schwerpunkt Internet*. Pearson Studium, 2002.
- [Leuf2001] Bo Leuf & Ward Cunningham. *The Wiki Way*. Addison-Wesley, 2001.
- [Louridas2006] Panagiotis Louridas. *Using Wikis in Software Development*. IEEE Software. März/April 2006, S. 88ff.
- [Macaulay1995] Linda Macaulay. *Human computer interaction for software designers*. Thomson Computer Press, 1995.
- [Martin2004] Franck Martin. *RFCWiki. IETF RFC Draft for Standardization of the Wiki Syntax*. 1. September 2004. <http://tikiwiki.org/tiki-index.php?page=RFCWiki> (zuletzt eingesehen März 2006).
- [Meatball2006a] Meatball Wiki. *Wiki Syntax*. <http://www.usemod.com/cgi-bin/mb.pl?WikiSyntax> (zuletzt eingesehen März 2006).
- [Meatball2006b] Meatball Wiki. *Wiki Markup Standard*. <http://www.usemod.com/cgi-bin/mb.pl?WikiMarkupStandard> (zuletzt eingesehen März 2006).
- [Meatball2006c] Meatball Wiki. *Wiki Interchange Format*. <http://www.usemod.com/cgi-bin/mb.pl?WikiInterChangeFormat> (zuletzt eingesehen März 2006).
- [Michaud2005] Patrick Michaud. *PmWiki. PmWikiPhilosophy*. <http://www.pmichaud.com/wiki/PmWiki/PmWikiPhilosophy>, letzte Änderung 24.11.2005 (zuletzt eingesehen März 2006).
- [Möller2003] Erik Möller. *Das Wiki-Prinzip. Tanz der Gehirne Teil 1*. Telepolis, Magazin der Netzkultur. 9. September 2003. <http://www.telepolis.de/r4/artikel/14/14736/1.html> (zuletzt eingesehen März 2006)

- [Möller2005a] Erik Möller. *Die heimliche Medienrevolution*. Heise, 2005.
- [Moore2001] Andrew P. Moore, Robert J. Ellsion & Richard C. Linger. *Attack Modelling for Information Security and Survivability*. Technical Note CMU/SEI-2001-TN-001, Carnegie Mellon University, Pittsburgh, März 2001.
- [MouseSite2006] MouseSite. *The Demo. Video of the live public demonstration of the On Line System*.  
<http://sloan.stanford.edu/mousesite/1968Demo.html> (zuletzt eingesehen März 2006).
- [MySQL2006] MySQL. *MySQL Reference Manual*.  
<http://dev.mysql.com/doc/refman/4.1/en/fulltext-search.html> (zuletzt eingesehen März 2006).
- [Nandelstädt2001] Susanne Nandelstädt. *Computervermittelte Kommunikation: Formen und Eigenschaften. Am Beispiel einer webbasierten Telelearning- und Kommunikationsplattform im Bielefelder „Weiterbildenden Fernstudium Angewandte Gesundheitswissenschaften“*. Magisterarbeit, Universität Bielefeld, Fakultät für Linguistik und Literaturwissenschaften, Bielefeld, April 2001.
- [Nelson2006] Ted H. Nelson. *Project Xanadu. The Original Hypertext Project*.  
<http://xanadu.com> (zuletzt eingesehen März 2006).
- [Nothbaum2006] Norbert Nothbaum, Johanna Lojewski & Heiko Holtkamp. *Präsentation zur Worksphere. Vorstellung der BGM-Plattform. Einführungswoche des Studiengangs „Betriebliches Gesundheitsmanagement“*. Februar 2006, Verfügbar über die Nothbaum GmbH (<http://www.nothbaum.com>).
- [NW2006] *Auf dem Weg zur größten Wissenssammlung der Welt*. Neue Westfälische. Ausgabe Nr. 12, 14./15. Januar 2006.
- [Paul-Stüwe2005] Thilo Paul-Stüwe. *Formal Task Analysis of Graphical System Engineering Software Use*. Diplomarbeit, Universität Bielefeld, Technische Fakultät, März 2005. RVS-Dip-05-01, <http://www.rvs.uni-bielefeld.de/publications/>.
- [PHPLib] PHPLib. *PHPLib-Quellen*. <http://phplib.sourceforge.net>, (zuletzt eingesehen März 2006).

- [Pieprzyk2003] Josef Pieprzyk, Thomas Hardjono & Jennifer Seberry. *Fundamentals of Computer Security*. Springer, 2003.
- [Raskin2000] Jeff Raskin. *The Humane Interface. New directions for designing interactive systems*. Addison-Wesley, 2000.
- [Ratschiller2000] Tobias Ratschiller & Till Gerken. *Web Application Development with PHP 4.0*. New Riders Publishing, 2000.
- [RFC] RFC-Editor. *RFCs – Request for Comments. Liste aller RFCs*. <http://www.rfc-editor.org> (zuletzt eingesehen März 2006).
- [Schmid1999] Egon Schmid, Christian Cartus, & Richard Blume. *PHP, dynamische Webauftritte professionell realisieren*. Markt und Technik, 1999.
- [Schneier2001] Bruce Schneier. *Secrets & Lies. IT-Sicherheit in einer vernetzten Welt*. dPunkt.Verlag, Wiley, 2001.
- [Schnupp1992] Peter Schnupp. *Hypertext*. Oldenbourg, 1992.
- [Schulmeister2002] Rolf Schulmeister. *Grundlagen hypermedialer Lernsysteme. Theorie, Didaktik, Design*. Oldenbourg, 3. Auflage, 2002.
- [Sommerville2001] Ian Sommerville. *Software Engineering*. Pearson Studium, München, 6. Auflage, 2001.
- [Stamatelatos2002] Michael Stamatelatos, William Vesely, Joanne Dugan, Joseph Fragola, Joseph Minarick & Jan Railsback. *Fault Tree Handbook with Aerospace Applications, Version 1.1*. NASA Office of Safety and Mission Assurance, Washington DC, 2002. <http://www.hq.nasa.gov/offoce/codeq/doctree/fthb.pdf> (zuletzt eingesehen März 2006).
- [Tanenbaum2003] Andrew S. Tanenbaum. *Computernetzwerke*. Pearson Studium, 4. überarbeitete Auflage, 2003.
- [Thelen2003] Tobias Thelen & Clemens Gruber. *Kollaboratives Lernen mit WikiWikiWebs*, Seiten 356–365. In Kerres & Voß [Kerres2003], 2003.
- [Venners2003] Bill Venners. *Exploring with Wiki. A Conversation with Ward Cunningham, Part I*. Oktober 2003.

- <http://www.artima.com/intv/wikiP.html> (zuletzt eingesehen März 2006).
- [Welling2001] Luke Welling & Laura Thomson. *PHP and MySQL Web Development*. Sams Publishing, 2001.
- [Weston2004] Paul Weston. *Bioinformatics Software Engineering*. John Wiley & Sons Ltd., 2004.
- [Wikimania2005] *Wikimania 2005. The first international Wikimedia Conference*. 4. – 8. August 2005. <http://wikimania.wikimedia.org>, (zuletzt eingesehen März 2006).
- [WikipediaDE2006] Wikipedia. *Wikipedia. Die freie Enzyklopädie*. <http://de.wikipedia.org> (zuletzt eingesehen März 2006).
- [WikipediaEN2006] Wikipedia. *Wikipedia. The free encyclopedia*. <http://en.wikipedia.org> (zuletzt eingesehen März 2006).
- [WikipediaRisiko] *Artikel Risiko*. In: Wikipedia, Die freie Enzyklopädie [WikipediaDE2006]. <http://de.wikipedia.org/wiki/Risiko> (zuletzt eingesehen März 2006).
- [WikipediaRSS] *Artikel RSS*. In: Wikipedia, Die freie Enzyklopädie [WikipediaDE2006]. <http://de.wikipedia.org/wiki/RSS> (zuletzt eingesehen März 2006).
- [Zic2005] Sandro Zic. *Das Problemkind. Die Geschichte von PHP*. *PHP Magazin – PHPstarter*, Seite 13ff., 2005.