

Bachelorarbeit

**Entwurf und Implementierung einer Web 2.0-
Internetpräsenz für Nationalparks**

**Design and Implementation of a Web 2.0 Site
for National Parks**

Sascha Nehls



Bachelorarbeit: Entwurf und Implementierung einer Web 2.0-Internetpräsenz für Nationalparks: Design and Implementation of a Web 2.0 Site for National Parks

Sascha Nehls

Bachelorarbeit zur Erlangung des akademischen Grades "Bachelor of Science" (B.Sc.) an der Brandenburgischen Technischen Universität Cottbus.

Institut für Informatik

Lehrstuhl Internet-Technologie

Prof. Dr. Gerd Wagner

Zusammenfassung

Diese Bachelorarbeit soll die vielfältigen Möglichkeiten und Technologien des Web 2.0 näher untersuchen und in einer Implementierung unterschiedliche Web 2.0-Anwendungen miteinander vernetzen. Aufbauend auf das Open-Source Web-Content-Management-System Drupal wird eine Internetpräsenz für Nationalparks erstellt, die den Leitgedanken des Web 2.0 gerecht wird. Eine zentrale Rolle nimmt hierbei eine Google Map ein, die mit den Diensten Flickr und YouTube vernetzt wird und wichtige Orte im Nationalpark mit Bildern und Videos illustriert. Auch soll es möglich sein, Wanderwege und Unterkünfte auf der Google Map darzustellen und mittels Drupal zu modifizieren. Die Internetpräsenz wird abgerundet durch ein Quartier-Buchungssystem sowie einen Veranstaltungskalender.

Abstract

This Bachelor Thesis examines the new technologies and possibilities of the Web 2.0 and creates an implementation that mashes up different Web 2.0 applications. This implementation will be a Web 2.0 site for a national park which is entirely powered by the open source content management system Drupal. The central role on this web site plays a Google map, which is able to show photos from Flickr and videos from YouTube as well as important sights or interesting paths. Moreover it is possible to book accommodation facilities and show events on a calendar.

Widmung

Ich möchte diese Arbeit meiner Familie, insbesondere meinen Eltern und Großeltern, widmen. Trotzdem ich eine lange Zeit meines Studiums im Ausland verbrachte und wenig zu Hause war, habe ich stets organisatorische, moralische und finanzielle Unterstützung bekommen. Dafür möchte ich mich ganz herzlich bedanken.

1. Einleitung	1
1. Motivation	1
2. Aufbau dieser Arbeit	1
2. Das Web 2.0 - Marketingwort oder tatsächliche Neuerung?	2
1. Begriffsdefinition	2
2. Prinzipien	2
2.1. Aus menschlicher Sicht	2
2.2. Aus technologischer Sicht	3
2.3. Aus wirtschaftlicher Sicht	4
3. Verwendete Technologien	5
3.1. HTML/XHTML	5
3.2. CSS	5
3.3. XML	5
3.4. JavaScript/DOM	5
3.5. JSON	6
3.6. Ajax	6
3.7. Web Services	7
4. Anwendungsformen	7
4.1. Soziale Software	7
4.1.1. Wikis	7
4.1.2. Weblogs	7
4.1.3. Soziale Netzwerke	8
4.2. Taxonomien, Folksonomien und Tag Clouds	8
4.3. RSS und Atom	9
4.4. Podcasts	9
4.5. Rich Internet Applications	10
4.6. Mashups	10
5. Probleme	10
5.1. Bedienbarkeit	10
5.2. Datenschutz und Datenklau	11
5.3. Urheberrechtsverletzungen	12
5.4. Kompatibilität	12
3. Web 2.0 bei Tourismus-Internetpräsenzen	13
1. Einführung	13
2. Besonderheiten des touristischen Produktes	14
3. Zielgruppenanalyse	14
4. Anforderungen an die Benutzeroberfläche	15
5. Anforderungen an die Informations- und Navigationsstruktur	15
6. Anforderungen an den Inhalt	16
4. Umsetzung mit Drupal	18
1. Einleitung	18
2. Was ist Drupal?	18
3. Vor- und Nachteile von Drupal	19
4. Drupal Terminologie	19
4.1. Knoten	19
4.2. Modul	19
4.3. Rolle	20
4.4. Taxonomie	20
4.5. Block	20
4.6. Theme	20
4.7. Front- und Backend	20
5. Installation von Drupal	20
6. Konfiguration von Drupal	22
6.1. Aktivierung der benötigten Module	22
6.2. Konfiguration der Benutzergruppen und Benutzerrechte	23
6.3. Konfiguration des Drupal-Kerns	23
6.4. Konfiguration des TinyMCE-Moduls	24
6.5. Konfiguration des Event-Moduls	24

6.6. Konfiguration des Contact-Moduls	25
6.7. Konfiguration des Taxonomy-Moduls	25
6.8. Konfiguration des Image- und Image Attach-Moduls	25
6.9. Konfiguration des Locale-Moduls	26
6.10. Konfiguration des Search-Moduls	26
6.11. Konfiguration der Benutzeroberfläche	26
6.11.1. Anpassung von style.css	26
5. Entwicklung neuer Drupal-Module für den Tourismus	28
1. Modul-Entwicklung im Allgemeinen	28
1.1. Coding-Standard	28
1.2. Genereller Aufbau eines Drupal-Moduls	28
1.3. Drupal-Hooks	29
1.4. Benutzergruppen und Benutzerrechte	29
1.5. Datenbankzugriff	30
1.6. Formulare und Formularverarbeitung	30
1.7. Drupal und objektorientierte Programmierung	31
2. Modul-Entwicklung im Speziellen	32
2.1. Googlemap-Modul	32
2.1.1. Anforderungen	32
2.1.2. Datenbankentwurf	33
2.1.3. Implementierung	33
2.2. Mapmarker-Modul	36
2.2.1. Datenbankentwurf	36
2.2.2. Erzeugen von Markierungen und Darstellung auf der Google Map	37
2.2.3. Allgemeiner Zugriff auf die APIs	38
2.2.4. Abfrage der Flickr-Fotos	39
2.2.5. Abfrage der YouTube-Videos	41
2.3. Mappath-Modul	41
2.3.1. Datenbankentwurf	42
2.4. Lifeform-Modul	42
2.4.1. Datenbankentwurf	43
2.4.2. Umsetzung	43
2.5. Observation-Modul	43
2.5.1. Datenbankentwurf	44
2.5.2. Verknüpfung mit dem Lifeform-Modul	44
2.6. Accommodation-Modul	45
2.6.1. Datenbankentwurf	46
2.6.2. Umsetzung	47
3. Konfiguration der neuen Module	50
3.1. Googlemap	50
3.2. Mapmarker	50
3.3. Accommodation	51
3.4. Lifeform	51
3.5. Mappath und Observation	51
6. Test der Benutzeroberfläche	52
1. Testverfahren	52
2. Aufgabenstellung "Hotel einrichten"	52
2.1. Testprotokoll 1	52
2.2. Änderungen aufgrund von Testprotokoll 1	53
2.3. Testprotokoll 2	53
2.4. Änderungen aufgrund von Testprotokoll 2	53
3. Aufgabenstellung "Hotelbuchung durchführen"	54
3.1. Testprotokoll 3	54
3.2. Änderungen aufgrund von Testprotokoll 3	54
3.3. Testprotokoll 4	54
3.4. Änderungen aufgrund von Testprotokoll 4	54
4. Aufgabenstellung "Naturbeobachtung melden"	55
4.1. Testprotokoll 5	55

4.2. Änderungen aufgrund von Testprotokoll 5	55
4.3. Testprotokoll 6	55
4.4. Änderungen aufgrund von Testprotokoll 6	56
7. Ausblick	57
1. Virtuelles World Wide Web	57
2. Mobiles World Wide Web	57
3. Semantisches Web	57
8. Fazit	59
A. Pflichtenheft	61
1. Zielbestimmung	61
1.1. Musskriterien	61
1.2. Wishkriterien	61
1.3. Abgrenzungskriterien	61
2. Produkteinsatz	61
2.1. Anwendungsbereiche	61
2.2. Zielgruppen	62
2.2.1. Unregistrierte Benutzer	62
2.2.2. Registrierte Benutzer	62
2.2.3. Systemadministrator	63
2.3. Betriebsbedingungen	63
3. Produktübersicht	64
4. Produktfunktionen	64
5. Produktdaten	71
6. Produkteleistungen	71
7. Qualitätsanforderungen	72
8. Benutzeroberfläche	72
9. Nichtfunktionale Anforderungen	74
10. Technische Produktumgebung	74
10.1. Software	74
10.2. Hardware	74
10.3. Orgware	74
10.4. Produkt-Schnittstellen	74
11. Spezielle Anforderungen an die Entwicklungs-Umgebung	75
12. Gliederung in Teilprodukte	75
13. Ergänzungen	75
B. Grundgerüst eines Drupal-Modules	76
C. Abkürzungen	80
D. Eidesstattliche Erklärung	81
Literaturverzeichnis	82

Kapitel 1. Einleitung

1. Motivation

Seit der Begriff "Web 2.0" auf einer Konferenz des O'Reilly Verlages im Jahr 2004 erschaffen wurde, erfuhr das World Wide Web einen großen Hype um neue Anwendungen wie Weblogs, RSS-Feeds, Wikis und Podcasts. Grundgedanken wie Offenheit, Interaktivität und Benutzerfreundlichkeit rückten mehr und mehr in den Vordergrund und beeinflussten die Entwicklung neuer Internetpräsenzen maßgeblich. Durch die kostenlose und unkomplizierte Verfügbarkeit von Daten und Services erreichten neue Web-Anwendungen wie Google Maps, Flickr und YouTube eine hohe Popularität. Es entstanden soziale Netzwerke mit Millionen von Mitgliedern und das Web 2.0 hielt Einzug in eine Vielzahl klassischer Internetpräsenzen.

Je mehr das Web 2.0 an Bedeutung gewann, desto interessanter wurde auch die Übertragung der neu entdeckten Möglichkeiten auf Internetpräsenzen traditioneller Industriezweige. Diese Arbeit wird dieser Entwicklung Rechnung tragen und untersucht die Möglichkeiten der Verknüpfung des Web 2.0 mit einem bedeutenden Industriezweig unserer Zeit, dem Tourismus. Nach den Angaben der Deutschen Zentrale für Tourismus betrug im Jahr 2003 der Anteil des Tourismus am Bruttoinlandsprodukt (BIP) in Deutschland alleine 8% [DZT04]. Schon anhand dieser Zahl wird deutlich, wie viel Potenzial bei der Vermischung des stark wachsenden Mediums Internet mit dem Tourismus als klassischen Industriezweig entsteht.

Doch was bringt einem die modernste Internetpräsenz, wenn sie nur schwer aktualisierbar ist und die Potenziale nicht voll ausschöpft? Nichts langweilt die Besucher mehr als veralterte Inhalte, die zudem noch schlecht aufbereitet sind. Um dies zu verhindern, können sogenannte Web-Content-Management-Systeme (WCMS) eingesetzt werden, die auch Laien die Aktualisierung und Pflege einer Internetpräsenz ermöglichen. Damit ergibt sich auf lange Sicht eine erhebliche Kostenersparnis, da die Aktualisierung der Internetpräsenz nicht mehr nur von Experten durchgeführt werden kann. Mit dem Web 2.0 ist außerdem ein weiterer Aspekt von entscheidender Bedeutung, nämlich die Verwaltung der Nutzerdaten. Mehr als je zuvor stehen die von den Benutzern erstellten Inhalte im Vordergrund und tragen zur Popularität einer Internetpräsenz bei. Um das hohe Datenaufkommen zu verwalten, ist der Einsatz eines WCMS kaum mehr zu vermeiden. Glücklicherweise gibt es eine Vielzahl frei verfügbarer Content-Management-Systeme, die nach Belieben angepasst und erweitert werden können.

2. Aufbau dieser Arbeit

Diese Arbeit geht zunächst auf den Begriff "Web 2.0" ein und erläutert die Ursprünge und Prinzipien dieser neuen Netzform aus verschiedenen Blickwinkeln. Anhand konkreter Beispiele werden außerdem die Technologien und Anwendungsformen näher untersucht und kritisch hinterfragt. Anschließend werden die gewonnenen Erkenntnisse auf die speziellen Bedürfnisse des Tourismus übertragen und beispielhaft eine Internetpräsenz für den Nationalpark Unteres Odertal entworfen.

Zur Verwaltung dieser Internetpräsenz wurde nach einem geeigneten Content-Management-System (CMS) gesucht und Drupal als geeignete Lösung vorgestellt. Für dieses CMS wurden neue Web 2.0-Module entworfen und implementiert, die die Funktionalität von Drupal drastisch erweitern und eine Darstellung von Web 2.0-Inhalten erlauben. Eine zentrale Rolle nimmt dabei eine Google Map ein, welche mit den Diensten Flickr und YouTube vernetzt wird und wichtige Orte im Nationalpark mit Bildern und Videos illustriert. Auch ist es durch die neu entwickelten Module möglich, Wanderwege und Unterkünfte auf der Google Map darzustellen und mittels Drupal zu modifizieren. Besucher der Internetpräsenz können außerdem ihre eigenen Naturbeobachtungen veröffentlichen und in einem Forum über die unterschiedlichen Aspekte des Nationalparks diskutieren. Für die Nationalparkverwaltung besteht die Möglichkeit, Neuigkeiten und interessante Berichte in einem Weblog zu veröffentlichen sowie bevorstehende Ereignisse in einem Online-Kalender anzukündigen.

Die neu entwickelten Module wurden anschließend ausgiebigen Benutzertests unterworfen, um die Funktionalität und Bedienbarkeit zu überprüfen. Die Ergebnisse dieser Tests sind direkt in die Modulentwicklung eingeflossen und halfen bei der Optimierung der Module für den Benutzer. Zu guter Letzt wurde ein Ausblick in die Zukunft des World Wide Webs gewagt und vielversprechende Neuentwicklungen näher untersucht.

Kapitel 2. Das Web 2.0 - Marketingwort oder tatsächliche Neuerung?

Eine Suche mit der Suchmaschine Google nach dem Begriff "Web 2.0" bringt nicht weniger als 678 Millionen Treffer. Zum Vergleich, eine Suche nach "Deutschland" bringt 278 Millionen, eine Suche nach "George Bush" 117 Millionen Treffer (Stand 23.November 2007). Gemessen an der Anzahl der Suchergebnisse auf Google muss das Web 2.0 also etwas sehr bedeutendes sein, dennoch ist vielen Menschen nicht klar, was das Web 2.0 eigentlich darstellt. Dieses Kapitel klärt genau diese Frage und erläutert die Ursprünge, Prinzipien, Technologien und Anwendungsformen des Web 2.0.

1. Begriffsdefinition

Um es vorweg zu nehmen, für das Web 2.0 gibt es keine klare Definition, sondern lediglich Merkmale, die das Web 2.0 charakterisieren. Der Grund dafür ist relativ einfach, denn der Begriff "Web 2.0" wurde ursprünglich als Marketingwort eingeführt. Das Web 2.0 ist somit kein Begriff für eine neue Softwareversion oder eine neue Infrastruktur, sondern lediglich ein Wort zum Vermarkten neuer Bewegungen und Tendenzen im World Wide Web, die in den letzten Jahren an Bedeutung gewonnen haben.

Entwickelt wurde der Begriff "Web 2.0" bei einem Brainstorming des O'Reilly Verlages und MediaLive International im Jahr 2004. Nach den Auffassungen des Verlagsteams war es nach dem Platzen der Dotcom-Blase nötig, einen Begriff für die neu aufkommenden Internet-Anwendungen wie Flickr, Napster oder Wikipedia zu finden. Der Begriff sollte nach außen hin zeigen, dass es sich um neue Internet-Anwendungen handelt, die die Dotcom-Blase überlebt hatten und gewissermaßen einen Wendepunkt darstellten (Vgl. [Ore05]). Aus dem ursprünglich als Marketingwort eingeführten Begriff hat sich in den letzten Jahren eine neue Netzform entwickelt, die uns bewusst oder unbewusst im Internet begleitet.

Häufig wird das Web 2.0 dafür kritisiert, dass es keine genaue Abgrenzung zum herkömmlichen World Wide Web (WWW) gibt. Einer der Mitbegründer des WWWs, Tim Berners-Lee, brachte es in einem Interview für den "IBM Developer Podcast" auf den Punkt: "I think Web 2.0 is of course a piece of jargon, nobody even knows what it means" [IBM06]. Übersetzt bedeutet dies so viel wie: "Ich denke das Web 2.0 ist ein Jargon, von dem keiner auch nur weiß, was er bedeuten soll.". Um also die Frage zu klären, was das Web 2.0 eigentlich ist und wie es sich vom herkömmlichen Web unterscheidet, muss man die Merkmale und Prinzipien näher untersuchen.

2. Prinzipien

Folgen wir den Ausführungen von [Ore05] gibt es eine Reihe von Prinzipien, die das Web 2.0 kennzeichnen.

2.1. Aus menschlicher Sicht

Eines der wesentlichen Prinzipien des Web 2.0 ist es, dass der Benutzer in den Mittelpunkt rückt. Anders als bei Internetseiten der "früheren Generation" ist der Benutzer nun nicht mehr nur Konsument von Informationen, sondern auch Produzent. Die Besucher von Internetseiten werden somit selbst zu Autoren und veröffentlichen ihre eigenen Inhalte. Die so erstellten Inhalte nennt man auch "User Generated Content" (dt. von Nutzern erstellter Inhalt). Die Idee, die dahinter steckt, hatte schon Macchiavelli vor hunderten von Jahren erkannt:

„Die beste Methode, um Informationen zu bekommen, ist die, selbst welche zu geben.“ Niccolo Macchiavelli (1469 - 1527)

Diese Auffassung spiegelt sich auch im Web 2.0 wieder: Es werden Meinungen ausgetauscht, Empfehlungen geschrieben, die Lieblings-Internetseiten publiziert, Videos ausgetauscht und viele Dinge mehr. In gewisser Weise ist dies eine natürliche Sache, die uns Menschen schon seit jeher begleitet. Mit dem Web 2.0 ist nun dieser Austausch von Informationen mit Menschen aus der ganzen Welt möglich, nicht umsonst bezeichnet man das Web 2.0 auch als "Mitmach-Netz".

Eine sehr populäre Anwendung, die genau dieses Prinzip widerspiegelt ist die freie Enzyklopädie Wikipedia¹. Die Inhalte dieser Enzyklopädie sind von herkömmlichen Internetnutzern geschaffen. Wirklich jeder kann bei Wikipedia Inhalte erstellen, bearbeiten oder löschen. Inzwischen ist die Qualität der Beiträge sehr hoch und kann sich durchaus mit kommerziellen Enzyklopädiën wie dem Brockhaus messen (Vgl. [Wol06]).

Durch immer schnellere und günstigere Internetverbindungen ist aber der Informationsaustausch nicht mehr nur auf reinen Text beschränkt. Dienste wie Flickr² oder YouTube³, die das Publizieren von Bildern bzw. Videos für jedermann erlauben, sind technologisch möglich und haben sich am Markt etabliert. Hier findet sich auch ein weiteres Prinzip des Web 2.0; die Benutzeroberflächen bestechen durch Einfachheit und Klarheit, bieten aber trotzdem einen hohen Funktionsumfang. Dieses Phänomen wird auch unter dem Begriff "Barrierefreiheit" oder "Usability" zusammengefasst. Angesichts des zunehmenden Wettbewerbs können es sich Unternehmen auch gar nicht mehr leisten, bestimmte Nutzergruppen auszuschließen. Mit dem Web 2.0 ist es zum Beispiel für Internet-Einsteiger möglich, auf Plattformen wie YouTube ihre Urlaubsvideos, Konzertausschnitte oder sonstige Videos hochzuladen und diese ihren Freunden und Verwandten auf der ganzen Welt zu zeigen. Wer hätte sich das vor 5 Jahren so vorgestellt? Dass das Ganze auch Probleme mit sich führen kann, wird später noch aufgezeigt.

Der letzte wichtige Punkt aus menschlicher Sicht sind die sogenannten "sozialen Netzwerke" (engl. Social Networks). Früher war die Vernetzung der Internetnutzer auf Gästebücher oder Foren beschränkt, zu Zeiten des Web 2.0 gibt es soziale Netze für jegliche Inhalte. Über diese Netzwerke können Meinungen ausgetauscht, Empfehlungen geschrieben, persönliche Kontakte hergestellt oder sogar Freundschaften geschlossen werden. Diese sozialen Netzwerke erfreuen sich ungemein großer Beliebtheit, alleine das im Oktober 2005 gegründete Studentennetzwerk StudiVZ⁴ hat bereits 4 Millionen aktive Nutzer (Vgl. [Stu07]). Die Benutzung solcher Netzwerke ist in der Regel kostenlos.

2.2. Aus technologischer Sicht

Aus technologischer Sicht ist das Web 2.0 keinesfalls eine Neuentwicklung. Das Web 2.0 verwendet immer noch das HTTP-Protokoll zur Datenübertragung, bedient sich zur Darstellung der Internetseiten immer noch HTML und verwendet auch immer noch JavaScript/DOM zur clientseitigen Ausführung von Funktionen. Auch die verwendeten Technologien zur Strukturierung von Daten wie z.B. XML existieren größtenteils schon seit dem Ende der 90er Jahre. Die eigentlich technische Neuerung beim Web 2.0 besteht darin, dass bereits vorhandene Technologien auf eine neue Art und Weise genutzt werden.

Ein klassisches Beispiel hierfür ist eine Technologie namens "Asynchronous JavaScript and XML" (*Ajax*), welche eine Verknüpfung von JavaScript und XML darstellt. Früher hatte man blinde Frames⁵ zur asynchronen Datenübertragung verwendet, in den Zeiten des Web 2.0 nimmt man dafür das XMLHttpRequest-Objekt⁶. Damit werden die Reaktionszeiten von Internet-Anwendungen drastisch reduziert und der Benutzer bekommt das Gefühl, die Anwendung läuft auf dem lokalen Rechner und nicht im Internet.

Bedingt durch die neuen Anwendungsformen im Web 2.0 sind die technischen Anforderungen für die Anbieter von Web 2.0-Plattformen enorm gewachsen. Unternehmen wie Google oder Yahoo haben Millionen investiert, um dem gesteigerten Datenaufkommen entgegen zu kommen (Vgl. [Ore05]). Diese Entwicklung ging einher mit einer starken Verbesserung der Infrastruktur. Die Internetverbindungen wurden immer schneller und preisgünstiger - eine Tendenz die datenintensiven Diensten zusätzlichen Auftrieb verschaffte.

Ohnehin sind beim Web 2.0 die Daten von größerer Bedeutung, seien es die Karten bei Google Maps oder die Videos bei YouTube - ohne diese Daten wären die Internetseiten nutzlos. Tim O'Reilly, einer der Ideengeber des Web 2.0, hatte dies in einem kurzen Satz zusammen gefasst:

„Data is the next Intel Inside“[Ore05]

¹<http://de.wikipedia.org/>

²<http://www.flickr.com/>

³<http://www.youtube.com/>

⁴<http://www.studivz.net/>

⁵Blinde Frames haben eine Größe von 1x1 Pixel und wurden dafür genutzt, Daten zu übertragen ohne die komplette Internetseite neu laden zu müssen.

⁶Das XMLHttpRequest-Objekt wird gerade vom W3C standardisiert und kann bereits in vielen Web-Browsern genutzt werden.

Ein weiteres Merkmal aus technischer Sicht ist die Art, wie die Daten angeboten werden. Im klassischen World Wide Web haben die Benutzer die Inhalte mit ihrem Web-Browser aufgerufen, im Web 2.0 gibt es zusätzlich sogenannte *Application Programming Interfaces* (APIs), die den Zugriff auf die Daten durch eigene Programme erlauben. So kann jeder sein persönliches Weblog mit Videos von YouTube bereichern oder aber Nachrichten von digg.com direkt im Weblog anzeigen. Die dadurch entstehende Vernetzung verschiedener Web 2.0-Anwendungen bezeichnet man auch als Mashup.

2.3. Aus wirtschaftlicher Sicht

Auch aus wirtschaftlicher Sicht werden dem Web 2.0 in [Ore05] einige Neuerungen zugeschrieben. Als Beispiel dient hier das Unternehmen DoubleClick, welches sich gegen Ende der 90er Jahre auf Online-Werbung spezialisierte. Dabei konnten Betreiber stark frequentierter Internetpräsenzen die Werbung von DoubleClick veröffentlichen, und bekamen im Gegenzug Geld ausgezahlt. Dafür musste ein Werbevertrag ausgehandelt werden, dessen Konditionen von der Popularität der Internetpräsenz abhängig war. Für Betreiber weniger frequentierter Internetpräsenzen bestand jedoch keine Möglichkeit, mit DoubleClick-Werbung Geld zu verdienen.

Diesen Umstand nutzte Google und etablierte einen eigenen Werbedienst namens AdSense, welcher auf jeder Internetpräsenz implementiert werden konnte. Anders als DoubleClick, die Werbung nur in Form von Bannern anboten, entwickelte Google jedoch verschiedene benutzerfreundliche Darstellungsformen von Werbung, wie z.B. Texte oder Verweise mit kleinen Beschreibungen. Zudem wurde die angezeigte Werbung abhängig vom Seiteninhalt und damit auch von den Interessen des Besuchers der Internetpräsenz. So konnten gezielt die verschiedenen Nutzer angesprochen werden und die Werbung wurde personalisierter.

Neben der gezielten Werbung findet sich aber noch ein weiteres wirtschaftliches Prinzip im Web 2.0, genannt "The long tail" (dt. "der lange Schwanz"). Dieser Begriff wurde von Chris Anderson, dem Chefredakteur eines US-amerikanischen Technologie-Magazins, im Jahre 2004 vorgestellt und beschreibt den Markt der Nischenprodukte. Er führte die Betrachtung ausgehend von Erkenntnissen der US-Plattenindustrie, bei denen die breite Masse an Nischenmusik genauso viel Umsatz macht, wie die relativ kleine Produktpalette von Chart-Musik. In einem konventionellen Ladengeschäft würden Nischenprodukte nur selten verkauft werden und sind wenig rentabel. Durch die neuen Vertriebswege über das Internet, wie zum Beispiel das Musikportal iTunes, kann weniger gefragte Musik global vertrieben werden und erzielt dabei einen höheren Umsatz als die wenigen Bestseller (Vgl. [And07] und [Alb07]).

Abbildung 2.1. Longtail-Kurve von Chris Anderson [And07]



3. Verwendete Technologien

Der Begriff "Web 2.0" vermittelt durch die Versionsnummer geradezu eine Überarbeitung eines bestehenden technischen Systems. Und dennoch, das Web 2.0 verwendet Technologien, die bereits seit vielen Jahren existieren. Lediglich die Art und Weise, wie die vorhandenen Technologien verwendet werden, hat sich geändert. In den folgenden Abschnitten findet sich eine Übersicht mit den Kern-Technologien des Web 2.0.

3.1. HTML/XHTML

Das World Wide Web basiert auf den Sprachen *HTML* (Hypertext Markup Language) bzw. *XHTML* (Extensible Hypertext Markup Language). Dabei handelt es sich nicht um Programmiersprachen, sondern um textbasierte Auszeichnungssprachen. Es können damit Strukturen wie Überschriften, Absätze, Tabellen, Hyperlinks usw. erzeugt und in Web-Browsern angezeigt werden. HTML wurde vom World Wide Web Consortium (W3C) standardisiert und wird ständig weiter entwickelt⁷.

3.2. CSS

CSS (Cascading Style Sheets) ist eine vom W3C normierte Sprache, um die Darstellung der (X)HTML-Dokumente zu verändern. Mit CSS können Schriftgrößen, -farben, Abstände, Navigationselemente uvm. angepasst werden. Dabei ist es möglich, die Darstellung vom Medium abhängig zu machen. Somit können Internetseiten für verschiedene Ausgabemedien wie Bildschirme oder Drucker optimiert werden, ohne dass man dafür separate Internetseiten schaffen muss.

3.3. XML

XML (Extensible Markup Language) ist eine W3C-Spezifikation⁸ zur Darstellung strukturierter Daten in Form von Textdateien. Wie (X)HTML besteht auch XML aus *Tags*, also Sprachelementen mit einer bestimmten Semantik. Jedoch sind die Namen der Tags im Gegensatz zu (X)HTML frei definierbar. XML ist die Grundlage für viele andere Auszeichnungssprachen wie RSS, XHTML oder DocBook und damit für das Web 2.0 von entscheidender Bedeutung (Vgl. [Sch04] S 11-13).

Beispiel 2.1. Aufbau einer XML-Datei

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<university>
  <title>Technical University of Cottbus</title>
  <programme>Computer Science</programme>
  <programme>Information and Media Technology</programme>
  <programme>Architecture</programme>
</university>
```

3.4. JavaScript/DOM

JavaScript ist eine clientseitige Programmiersprache, mit der man die eher beschränkten Möglichkeiten von HTML erweitern kann. Das bedeutet, JavaScript läuft im Web-Browser und wird nur auf dem lokalen Computer ausgeführt. Früher wurde JavaScript dazu verwendet, um Formulareingaben vor dem Absenden zu überprüfen oder Laufschriften anzuzeigen. Heute liegt das Haupteinsatzgebiet von JavaScript eher in der dynamischen Manipulation von Internetseiten über das *Document Object Model* (DOM). Damit ist es möglich, ganz gezielt auf die einzelnen Inhalte der Internetseite zuzugreifen und diese zu verändern - ohne dabei die Seite neu laden zu müssen. Nicht zuletzt auch durch die neuen Konzepte des Web 2.0 erhielt JavaScript einen großen Aufschwung und ist heute die grundlegende Technologie zur asynchronen Datenübertragung mit Ajax (Vgl. [Wen07] S 21).

⁷Mehr Informationen dazu unter <http://www.w3.org/TR/xhtml1/>

⁸Mehr Informationen dazu unter <http://www.w3.org/TR/xml/>

3.5. JSON

JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, welches für Computer einfach zu verarbeiten ist. Dabei werden die Objekte durch einfachen Text repräsentiert und lassen sich so leicht erstellen und verarbeiten. Die Darstellung erfolgt typisiert, das heißt JSON-Objekte sind entweder Zeichenketten, Zahlen, Felder, Wahrheitswerte oder Null. Im Gegensatz zu *XML* entfällt somit das Parsen der Daten (Vgl. [GGA06] Seite 163). Darüber hinaus hat *JSON* weniger Overhead als *XML* und wird von nahezu allen Programmiersprachen verstanden.

Das Format eines JSON-Objektes lässt sich am Besten anhand eines Beispieles erklären.

Beispiel 2.2. Beispiel eines JSON-Objektes

```
{
  "Name": "Schwedt",
  "Einwohner": 37000,
  "Arbeitslosenquote": 21.2,
  "Bürgermeister": {
    "Vorname": "Jürgen",
    "Nachname": "Polzehl",
    "Alter": "57",
    "Partei": "SPD"
  }
  "Industrien": [
    "Erdöl", "Papier", "Tourismus"
  ]
}
```

Wie im Beispiel zu erkennen ist, beginnt und endet ein JSON-Objekt mit einer geschweiften Klammer. Innerhalb der geschweiften Klammer findet sich eine ungeordnete Liste von Name/Wert-Paaren, die ihrerseits wieder Objekte, Arrays, Zeichenketten, Zahlen oder Wahrheitswerte sein können.

3.6. Ajax

Ajax steht für "Asynchronous JavaScript and XML" und ist ein Konzept zur asynchronen Datenübertragung zwischen einem Browser und einem Web-Server. *Ajax* selbst ist jedoch keine neue Technologie oder Programmiersprache, die mit dem Web 2.0 entstanden ist, sondern ein Konzept, welches mehrere klassische Technologien in eine neue Art und Weise kombiniert. Einer der Ideengeber des Web 2.0, Jesse James Garret von Adaptive Path, erklärt *Ajax* folgendermaßen.

„Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways.“[Gar05]

Er führt 5 Technologien auf, die bei Ajax vereinigt werden:

- XHTML und CSS;
- Document Object Model (*DOM*);
- XML und XSLT;
- XMLHttpRequest;
- und JavaScript.

Das XMLHttpRequest-Objekt ist hierbei der entscheidende Teil, um asynchrone Datenübertragungen zu realisieren. Asynchron heißt in diesem Fall, dass Anfragen zum Server geschickt werden können, ohne dass der Web-Browser dabei blockiert oder die Seite komplett neu geladen werden muss. Aber wo finden sich Ajax-Anwendungen und was ist das besondere Merkmal dieser Anwendungen? Nun, der wesentliche Unterschied zu herkömmlichen Web-Anwendungen liegt darin, dass die Internetseiten nicht mehr so oft komplett neu geladen werden müssen und somit die Übertragung bzw. Verarbeitung der Daten wesentlich schneller von statten geht. Möchte man zum Beispiel

bei Google Maps⁹, eine typische Ajax-Anwendung, den Ausschnitt der Karte verändern, so muss dafür nicht die komplette Internetseite neu geladen werden. Es wird lediglich die Karte neu geladen, also der Bestandteil der Seite, der sich wirklich verändert hat. Ajax-Anwendungen haben also deutlich schnellere Reaktionszeiten als herkömmliche Internet-Anwendungen. Dies klingt im ersten Moment sehr interessant, doch diese neuen Anwendungen bringen auch einige Probleme mit sich (dazu mehr im Abschnitt "Probleme").

3.7. Web Services

Auch wenn Web Services nicht unbedingt dem Web 2.0 zuzuordnen sind, so finden sie doch in einigen Web 2.0-Anwendungen Verwendung und sollen an dieser Stelle erwähnt werden. Ein Web Service ist eine Software-Anwendung, auf die über gängige Internet-Protokolle zugegriffen wird. Dabei ist jeder Web Service durch einen *Uniform Resource Identifier* (URI) eindeutig identifizierbar und verwendet XML-basierte Nachrichten zur Interaktion mit anderen Anwendungen. Dies hat den Vorteil, dass Softwaresysteme unabhängig von der verwendeten Plattform und Programmiersprache miteinander kommunizieren können (Vgl. [Web07]).

Ein klassisches Beispiel für einen Web Service liefert die Google SOAP Search API. Dieser Web Service bietet für registrierte Benutzer die Möglichkeit, mittels selbst geschriebener Programme auf die Google-Suche zuzugreifen und die Ergebnisse in strukturierter Form zurückzubekommen. Man kann also eine Suchanfrage bei Google durchführen, ohne den Web-Browser zu benutzen.

4. Anwendungsformen

Auf den vorherigen Seiten wurden schon einige Beispiele für Web 2.0-Anwendungen gegeben, die sich großer Beliebtheit erfreuen. Laut einer aktuellen Studie von ARD und ZDF [GF07] nutzen bereits 47% der Onlinenutzer die freie Enzyklopädie Wikipedia und 34% Videoportale wie YouTube - Tendenz steigend. Doch im Web 2.0 gibt es mehr als nur Audio- und Videoportale, wie die folgende Einteilung zeigt.

4.1. Soziale Software

Eines der Kernkonzepte des Web 2.0 ist es, Menschen auf verschiedenen Gebieten miteinander zu vernetzen. Die aus diesem Prinzip entstandene Software nennt man soziale Software (engl. Social Software). Bei dieser Art von Software ist der *User Generated Content* (UGC), also der von Benutzern erstellte Inhalt, von zentraler Bedeutung und macht die soziale Software so interessant. Zudem unterstützt soziale Software die menschliche Kommunikation, Interaktion und Zusammenarbeit ([Gra06] S 37). Soziale Software lässt sich in verschiedene Anwendungsformen untergliedern.

4.1.1. Wikis

Wikis stellen im Allgemeinen eine Sammlung von Internetseiten dar, die von Benutzern nicht nur gelesen, sondern auch in Echtzeit verändert werden können. Ursprünglich kommt der Name Wiki aus dem Hawaiianischen und bedeutet "schnell". Wikis werden primär zur Wissensverwaltung eingesetzt, denn Benutzer können bei Wikis selbst Informationen hinzufügen oder bereits vorhandene Informationen bearbeiten. Das so publizierte Wissen ist grundsätzlich jedem zugänglich. [Alb07] spricht in diesem Zusammenhang auch von "kollektiver Intelligenz".

Die wohl bekannteste Wiki ist die 2001 gegründete Wikipedia. Der Name Wikipedia setzt sich aus dem Wort "Wiki" und "Encyklopedia" zusammen und weist auf die Verwendung als Enzyklopädie hin. Im Laufe der Zeit hat sich Wikipedia zu einem angesehenen Nachschlagewerk entwickelt und kann sich von dem Umfang und der Qualität auch mit klassischen Enzyklopädien messen (Vgl. [Wol06]). Wie kann das funktionieren? Nun, dadurch das wirklich jeder an dieser Enzyklopädie mitarbeiten kann, entsteht die Qualität der Beiträge durch die Breite der Nutzer. Jeder Nutzer weiß etwas, und sei es nur der Name des Nachbardorfes. Falschmeldungen oder Vandalismus werden durch die Vielzahl an Benutzern innerhalb kurzer Zeit erkannt und beseitigt.

4.1.2. Weblogs

Der Begriff "Weblog" oder kurz "Blog" ist aus der Zusammenführung der englischen Begriffe "Web" und "Log" (dt. Logbuch) entstanden. Ein Blog ähnelt also in gewisser Weise einem Tagebuch, welches im Internet veröffentlicht

⁹<http://maps.google.de/>

wird. Die Einträge werden in der Regel chronologisch sortiert und regelmäßig aktualisiert. Jeder, der Zugang zum Internet hat, kann kostenlos ein Blog anlegen und über seine Erlebnisse berichten (Vgl. [Alb07] S 21-22).

Mittlerweile gibt es Weblogs mit allen möglichen Themen, von der Reise zur Internationalen Raumstation ISS¹⁰ über Erfahrungen in fremden Ländern¹¹ bis hin zu wissenschaftlichen Berichten und Studien¹².

Weblogs bestehen nicht mehr nur aus reinem Text, sondern können auch Fotos, Videos, Google Maps oder ähnliches enthalten. Auch werden Weblogs mittlerweile zur Kommunikation innerhalb von Unternehmen eingesetzt. Gerade bei der Projektarbeit können Weblogs als Informations- und Archivplattform eingesetzt werden und bieten so jedem Teilnehmer einen Überblick über die aktuellen Entscheidungen (Vgl. [Alb07] S 43).

4.1.3. Soziale Netzwerke

Sei es ein berufliches Netzwerk wie XING¹³, Nachrichtennetzwerke wie Digg¹⁴, Kontaktnetzwerke wie MySpace¹⁵ oder Lesezeichennetzwerke wie Del.Icio¹⁶ - im Web 2.0 gibt es für nahezu jeden Bereich ein passendes Netzwerk. Die Benutzer profitieren gegenseitig von den eingestellten Inhalten und es entsteht ein Gemeinschaftsgefühl. Auch ist es möglich, in den sozialen Netzwerken Anerkennung und Aufmerksamkeit zu bekommen. So hat der Fotodienst Flickr ein Bewertungssystem, in dem die am besten bewerteten Bilder ausgezeichnet werden. Bei MySpace und XING sind die Anzahl der Kontakte bzw. Freunde ein Indikator der eigenen Popularität (Vgl. [Alb07] S 108-110).

4.2. Taxonomien, Folksonomien und Tag Clouds

So wie bei den Gelben Seiten Unternehmen nach bestimmten Kategorien (Branchen) geordnet werden, können auch Internetseiten in bestimmte Kategorien eingeordnet werden. So werden Produkte beim Online-Auktionshaus ebay¹⁷ immer in bestimmte Hauptkategorien (wie z.B. Auto, Computer, Garten, Musik, Sport) und Unterkategorien (beim Beispiel Auto gibt es die Unterkategorien Reifen, Felgen, Tuning, Ersatzteile, Werkzeuge, etc.) eingeordnet. Solche Gliederungen kennt man auch aus der Biologie, wo Tiere und Pflanzen in eine bestimmte Art, Gattung und Familie eingeordnet werden. Dafür wurde der Begriff Taxonomie eingeführt.

In Anlehnung an dieses altbewährte Gliederungsprinzip wurde mit dem Web 2.0 der Begriff *Folksonomy*¹⁸ geschaffen. Im Gegensatz zu einer Taxonomie werden aber nicht nur Hierarchien zur Gliederung der Inhalte verwendet, sondern auch sogenannte *Tags*. Diese Tags sind Schlagwörter, die einen Inhalt beschreiben. So kann ein Blogeintrag über ein Fußballspiel des FC Energie Cottbus beispielsweise mit den Schlagwörtern "Fußball", "Sport", "Lausitz", "Cottbus", "Spaß", "Stimmung", "Stadion", "Frust" oder ähnliches versehen werden. Inhalte sind so nicht mehr nur über bestimmte Gliederungshierarchien auffindbar, sondern auch über bestimmte Schlagwörter.

Im Unterschied zu Taxonomien, wo Inhalte in vorgegebene Kategorien eingeordnet werden müssen, kann bei den Folksonomien jeder Nutzer selbst seine Inhalte anhand frei gewählter Schlagwörter kategorisieren. Diese Vorgehensweise bezeichnet man auch als *Tagging*. Dadurch entsteht eine sehr komplexe Kategorisierung, die zwar einiges an Präzision verloren hat, dafür aber Informationen auf sehr interessante Weise miteinander verknüpft.

Im Web 2.0 haben diese Schlagwörter eine besondere Bedeutung und werden grafisch in Form einer sogenannten "Tag Cloud" (dt. Begriffswolke) aufgearbeitet. Diese "Wolke" zeigt alle vorkommenden Schlagwörter in alphabetischer Reihenfolge und stellt die populärsten Schlagwörter am größten dar. So können Benutzer auf einem Blick die wichtigsten Schlagwörter und damit auch den Inhalt des Dokumentes erfassen. Das folgende Bild zeigt beispielhaft eine Tag Cloud vom Bilderdienst Flickr.

¹⁰<http://spaceblog.xprize.org/>

¹¹<http://lena.erasmusblog.com/>

¹²<http://wissenschafts-news.blog.de/>

¹³XING ist ein Netzwerk um Geschäftspartner, Mitarbeiter und Jobs zu finden. Internet: <http://www.xing.com>

¹⁴Digg ist ein Netzwerk, das hoch bewertete Nachrichten und Videos anzeigt. Internet: <http://www.digg.com/>

¹⁵Unter <http://www.myspace.com> können Kontakte mit Personen gleicher Interessengebiete hergestellt werden.

¹⁶Unter <http://del.icio.us> kann jeder seine Lieblings-Lesezeichen veröffentlichen.

¹⁷<http://www.ebay.de>

¹⁸Folksonomy ist eine Zusammensetzung der Wörter Folks (engl. für Menschen) und Taxonomy.

Abbildung 2.2. Tag Cloud

07 africa amsterdam animals architecture art august australia autumn baby barcelona
beach berlin birthday black blackandwhite blue boston bw california
cameraphone camping canada canon car cat chicago china christmas church
city clouds color concert day de dog england europe fall family festival film
florida flower flowers food football france friends fun garden geotagged
germany girl girls graffiti green halloween hawaii hiking holiday home honeymoon
house india ireland island italy japan july june kids la lake landscape light live
london macro may me mexico mountain mountains museum music nature new
newyork newyorkcity night nikon nyc ocean october paris park party people
photos portrait red river roadtrip rock rome san sanfrancisco scotland sea seattle show
sky snow spain spring street summer sun sunset sydney taiwan texas thailand
tokyo toronto tour travel tree trees trip uk urban usa vacation vancouver
washington water wedding white winter yellow york zoo

4.3. RSS und Atom

Tim O'Reilly, einer der Mitbegründer des Web 2.0, bezeichnete *RSS* einst als eines der wichtigsten Fortschritte in der Architektur des World Wide Webs seit Programmierer erkannt haben, dass *CGI* dazu benutzt werden kann, datenbankbasierte Internetseiten zu generieren (Vgl. [Ore05]). Heute, über 3 Jahre nach der ersten Web 2.0-Konferenz, wird das Ganze etwas nüchterner betrachtet.

RSS ist die Abkürzung für "Really Simple Syndication" (dt. "wirklich einfache Verbreitung") und ist ein elektronisches Nachrichtenformat, welches dem Nutzer ermöglicht, personalisierte Nachrichten mit dem Web-Browser oder einer speziellen Software zu empfangen. Diese Nachrichten werden in Form sogenannter *RSS-Feeds* auf verschiedenen Internetseiten bereit gestellt und können dann von dem Benutzer abonniert werden. In den meisten Fällen enthalten die Nachrichten nur die Überschrift und einen einleitenden Text, wie zum Beispiel der *RSS-Feed* der Tagesschau¹⁹.

Das *RSS*-Format wurde schon im Jahre 1997 entwickelt, also lange bevor vom Web 2.0 die Rede war. Es basiert auf *XML* und existiert in verschiedenen Versionen, die von unterschiedlichen Unternehmen entwickelt wurden und zum Teil nicht miteinander kompatibel sind. Um ein einheitliches Nachrichtenformat zu haben, wurde im Jahre 2005 ein standardisiertes Format namens *Atom* entwickelt, welches ebenfalls auf *XML* basiert und einige Verbesserungen gegenüber *RSS* besitzt. Dieses Format hat sich allerdings noch nicht durchgesetzt (Vgl. [Alb07] S 137-145).

4.4. Podcasts

Ähnlich wie der Begriff "Weblog" entwickelte sich auch der Begriff "Podcast" aus zwei zusammen gesetzten englischen Wörtern. Zum einen aus dem Namen des populären MP3-Players "iPod" und zum anderen aus dem englischen Wort "Broadcasting", welches soviel wie "Sendung" oder "Übertragung" bedeutet.

Ein Podcast ist eine Art Radiosendung im Internet, die im MP3-Format angeboten wird. Ähnlich wie bei Weblogs kann jeder Internetnutzer selbst Podcasts erstellen und veröffentlichen, wobei dem Themenbereich keine Grenzen gesetzt sind. Die meisten Podcasts sind frei im Internet verfügbar und können direkt herunter geladen oder über *RSS-Feeds* abonniert werden. Viele Fernsehsender und Nachrichtenstationen bieten im Internet ihre Sendungen

¹⁹<http://www.tagesschau.de/xml/rss2/>

als Podcast an, darunter auch die Tagesschau²⁰. In der Regel haben Podcasts eine Länge von wenigen Minuten bis zu einer Stunde.

Mittlerweile sind Podcasts nicht mehr nur ein akustisches Vergnügen, unter ²¹ gibt es Woche für Woche eine Video-Botschaft der Bundeskanzlerin Angela Merkel zum Herunterladen.

4.5. Rich Internet Applications

Rich Internet Applications (*RIA*) sind Web-Anwendungen, die im Web-Browser ausgeführt werden, aber sich dennoch ähnlich verhalten wie Desktop-Anwendungen. Diese Anwendungen bestechen durch Reaktionszeiten und Funktionsabläufen, wie man sie sonst nur von normalen Desktop-Anwendungen kennt. Im Allgemeinen hat der Nutzer von *RIA* das Gefühl, die Anwendung wird lokal auf seinem Rechner ausgeführt und nicht über das Internet.

Ein aktuelles Beispiel hierfür kommt von Google, nämlich Google Docs²². Diese Internet-Anwendung ermöglicht es jedem Benutzer, ein Textverarbeitungsprogramm, eine Tabellenkalkulation und ein Präsentationsprogramm im Web-Browser zu benutzen. Es ist keinerlei Software-Installation notwendig, die Anwendungen können einfach in jedem Web-Browser geladen werden. Der Funktionsumfang und die Funktionsweise dieser Software kommt schon sehr nahe an die klassischen Büroanwendungen wie Microsoft Office²³ heran. Die im Internet erstellten Dokumente können natürlich auch online gespeichert sowie in gängige Formate exportiert werden. Und dank Ajax ist die Reaktionszeit dieser Anwendungen auf ein Minimum geschrumpft.

4.6. Mashups

Das Wort "Mashup" kommt ursprünglich aus der Musikindustrie und bezeichnet die Vermischung von Musik unterschiedlicher Stilrichtungen. Dieser Begriff wurde im Web 2.0 auf neuartige Anwendungen übertragen, die die Daten und Services mehrerer Web 2.0-Dienste nutzen und diese miteinander vernetzen. In der Regel werden diese Daten über sogenannte Application Programming Interfaces (APIs) abgerufen und als Teil einer größeren Anwendung verwendet.

Während Mashups ursprünglich von Tim O'Reilly als Spielzeug für Hacker abgestempelt wurden (Vgl. [Ore05]), haben sich in der Zwischenzeit auch viele kommerzielle Anbieter an die Entwicklung neuer Mashups gewagt. Beinahe jede Datenquelle wird heute verwendet, seien es Fotos, Verkehrsnachrichten oder Immobilien, und es entstehen sehr interessante Anwendungen. In den meisten Fällen werden die Daten mit einer Google Map verknüpft, so bietet das Hamburger Unternehmen Ubilabs einen Service mit dem Namen `loc.alize.us`²⁴. Diese Anwendung zeigt eine Weltkarte in Form einer Google Map und stellt auf dieser Karte Fotos von Flickr dar. Dabei entspricht die Position der Fotos auf der Karte der tatsächlichen Position, wo die Fotos aufgenommen wurden.

5. Probleme

Das eine gemeinschaftliche Erstellung von Inhalten funktioniert, zeigt die Online-Ezyklopädie Wikipedia. Registrierte und unregistrierte Benutzer können dort nach Belieben Inhalte erstellen, bearbeiten oder löschen. Durch die Größe des Netzwerks und die damit verbundene Anzahl an Mitgliedern haben Vandalismus oder Falschaussagen kaum eine Chance. Dennoch treten im Web 2.0 immer wieder Probleme auf, die in den folgenden Abschnitten erläutert werden.

5.1. Bedienbarkeit

Eines der Merkmale von Web 2.0-Anwendungen ist, dass sie durch die Verwendung von *Ajax* einfach und schnell bedient werden können. Im englischen Sprachraum bezeichnet man dies auch als *usability*. Dadurch entstehen aber auch Probleme in der Bedienbarkeit, denn die Benutzer des Internets haben sich einfach an einen gewissen Ablauf gewöhnt. Im Prinzip verlaufen herkömmliche Internetanwendungen immer nach dem selben Muster:

²⁰<http://www.tagesschau.de/podcast>

²¹<http://www.bundeskanzlerin.de/Webs/BK/DE/Aktuelles/VideoPodcast/video-podcast.html>

²²<http://docs.google.com>

²³<http://office.microsoft.com/>

²⁴<http://loc.alize.us>

- Benutzeraktion (zum Beispiel Abschicken eines Formulars, Aufruf eines Verweises),
- Übertragung der Daten zum Server,
- Verarbeitung der Daten auf dem Server,
- Übertragung der Antwortdaten zum Clienten,
- und Aufbau einer neuen Internetseite im Browser.

Dadurch, dass bei jeder Benutzeraktion immer die komplette Seite neu geladen wird, versteht der Benutzer, dass seine Eingaben verarbeitet werden. Dank neuester DSL-Anschlüsse wurde die Verarbeitungs- und Ladezeit drastisch verkürzt, aber trotzdem verläuft jede klassische Internet-Anwendung immer noch nach dem oben beschriebenen Muster.

Bei sehr Ajax-intensiven Anwendungen wird dieses Muster jedoch durchbrochen, denn es werden nur die Daten neu geladen, die auch wirklich benötigt werden. Ein positiver Nebeneffekt ist hierbei, dass die Antwortzeiten deutlich schneller sind, jedoch führt dies auch zu Missverständnissen bei den Anwendern. Man stelle sich vor, ein Benutzer schickt ein Online-Formular ab. In herkömmlichen Anwendungen würde nach dem Bestätigen des Formulars die komplette Seite verschwinden und eine neue Seite im Browser-Fenster aufgebaut. Gleichzeitig zeigt der Ladebalken im Browser-Fenster an, wie weit der Ladevorgang fortgeschritten ist. Bei einer Ajax-Anwendung dagegen ist es möglich, dass die Daten einfach abgeschickt werden, ohne die Seite neu zu laden. Der Anwender bekommt also keine unmittelbare Rückmeldung mehr über die Durchführung der gewünschten Aktion. Dies führt zur Verunsicherung bei den Benutzern, ob die gewünschte Aktion ausgeführt wurde oder nicht. Deshalb ist es ungemein wichtig, dem Nutzer Rückmeldungen über den Status seiner Aktion zu geben (Vgl. [Pus07] S 160-163).

Doch dies ist nicht das einzige Problem hinsichtlich der Bedienbarkeit. Die Vor- und Zurück-Schaltflächen im Browser-Fenster werden von Ajax-Anwendungen nicht unterstützt und verlieren ihre Funktion. Verschiebt man zum Beispiel den Kartenbereich einer Google Map, so erzeugt dies keine neue Stelle in der Zurück-Liste des Web-Browsers. Wesentlich unangenehmer wird dies, wenn man eine Internetseite mit einer Google Map neu laden muss. Dann wird in jedem Fall wieder der Ursprungszustand hergestellt und alle Benutzeraktionen auf der Karte sind verloren.

Ein anderes Problem in diesem Zusammenhang ist das Erstellen von Lesezeichen²⁵. Eine Ajax-Seite kann unterschiedliche Zustände haben, je nach dem welche Aktionen der Benutzer ausgeführt hat. Ein Lesezeichen kann jedoch immer nur den Standard-Zustand, also den Zustand beim Betreten der Internetseite, abspeichern.

5.2. Datenschutz und Datenklau

So schön die neuen Möglichkeiten des Web 2.0 auch sein mögen, der Schutz der Benutzerdaten wird leider nicht immer gewährleistet. Ein populärer Fall im deutschsprachigen Raum ist beispielsweise der mangelhafte Datenschutz im Online-Studentenverzeichnis "studivZ"²⁶. Dort kann man im Sinne eines sozialen Netzwerkes Kontakt zu Kommilitonen aufnehmen und gemeinsame Interessen herausfinden. Außerdem kann man eigene Bilder hochladen und diese seinen Online-Freunden zeigen. Zum Erstaunen der Benutzer waren diese Bilder jedoch nicht nur für den Freundeskreis sichtbar, sondern prinzipiell für jeden Internetnutzer. Auch wurden in mehreren Phishing-Angriffen die E-Mail-Adressen einiger Nutzer ausgelesen und verbreitet. Das Auftauchen immer größerer Sicherheitslücken zwangen den Betreiber sogar zu mehrfachen Zwangspausen, die Sicherheit der Daten konnte einfach nicht mehr gewährleistet werden (Vgl. [Meu06]).

Hinzu kommt, dass Web 2.0-Plattformen häufig schon im Beta-Stadium für die Nutzung freigegeben werden und erst mit Hilfe der Benutzer zu einem fertigen Produkt vervollständigt werden. Für die Entwickler solcher Plattformen besteht so die Möglichkeit, direkte Rückmeldungen zu neuen Funktionen zu bekommen und die Plattform somit direkt im Sinne der Benutzer zu erweitern. Diese Herangehensweise wird in [Ore05] auch als "perpetual beta" bezeichnet. Auch hier besteht der Nachteil darin, dass durch eventuelle Sicherheitslücken die sensiblen Daten der Benutzer verraten werden könnten.

²⁵Bei Browsern aus dem Hause Microsoft werden Lesezeichen auch als Favoriten bezeichnet.

²⁶<http://www.studivz.net>

Grundsätzlich gilt natürlich, je weniger Daten ich preisgebe, desto weniger Daten können auch zweckentfremdet verwendet bzw. gestohlen werden. Gerade weil bei Web 2.0-Anwendungen aber die Daten der Nutzer im Vordergrund stehen - seien es Kommentare zu bestimmten Themen, publizierte Fotos, Videos oder Blogeinträge über den letzten Urlaub - sind Web 2.0-Anwendungen naturgemäß anfällig für Datenklau.

5.3. Urheberrechtsverletzungen

Das Web 2.0 lebt von den Inhalten der Nutzer, aber genau darin liegt auch eine Gefahrenquelle. Sehr leicht können beim Publizieren von Inhalten die Urheberrechte anderer verletzt werden, denn die juristischen Spitzfindigkeiten sind nur den wenigsten Nutzern geläufig. Vor allem die Foto- und Videoplattformen waren in der Vergangenheit häufig die Quellen für Urheberrechtsverletzungen. So hatte der US-Medienkonzern Viacom die Videoplattform YouTube wegen der Veröffentlichung urheberrechtlich geschützter Werke auf eine Milliarde US-Dollar verklagt (Vgl. [Foc07]).

5.4. Kompatibilität

Auch wenn dieser Prozentsatz in den letzten Jahren stark gesunken ist, gibt es immer noch Internetnutzer, die in ihren Web-Browsern JavaScript deaktiviert oder sogar überhaupt keine Unterstützung für JavaScript haben. Laut einer Statistik von [W3S07] betraf dies im Januar 2007 noch rund 7% der Internetnutzer. Basiert nun ein sehr bedeutender Teil der Internetpräsenz auf JavaScript, schließt man automatisch eine gewisse Gruppe an Internetnutzern aus. Dieses Problem kann man umgehen, indem man zumindest die Navigation JavaScript-frei gestaltet oder sogar eine extra Internetpräsenz ohne JavaScript anbietet. Für Anwendungen wie Google Maps ist es natürlich unmöglich, eine JavaScript-freie Version zu veröffentlichen. Andere Technologien wie zum Beispiel Macromedia Flash sind auch keine wirklichen Alternativen, da sie ebenfalls von einigen Internetnutzern nicht verstanden werden.

Kapitel 3. Web 2.0 bei Tourismus-Internetpräsenzen

1. Einführung

Das Web 2.0 als Mitmach-Netz bietet sich gerade zu an, für den Tourismus eingesetzt zu werden. Seien es einfache Blogs über die Erlebnisse im Urlaub, Empfehlungen über bestimmte Sehenswürdigkeiten oder interaktive Karten zur Darstellung von Wanderwegen - gerade im Tourismussektor stellen diese neuen Anwendungsformen eine unglaubliche Bereicherung dar. Touristen können von zu Hause ihre Reise planen, Preise vergleichen und Hotels buchen, ohne auch nur einen Schritt zum Reisebüro zu machen.

Es wurden dafür verschiedene Schlagwörter wie "Travel 2.0" oder "eTourismus" geschaffen, die als Synonym für neue Anwendungen in dem Bereich des Online-Tourismus angewendet werden. Eine genaue Abgrenzung oder Definition dieser Begriffe ist jedoch nur schwer möglich. Im Generellen sind die folgenden Szenarien beim Zusammenspiel vom Web 2.0 mit dem Tourismus von besonderer Bedeutung:

- Wissen vermitteln

Eine der Hauptaufgaben von Tourismus-Internetpräsenzen ist immer das Vermitteln von Wissen über die touristische Region und die Ausflugsmöglichkeiten vor Ort. Hierzu können Wikis verwendet werden, die im Laufe der Zeit von den Nutzern vervollständigt werden. Dies setzt allerdings voraus, dass die Internetpräsenz eine sehr große Besucheranzahl hat.

- Reiseberichte und Reisebewertungen

Empfehlungen und Meinungen anderer Menschen, die schon das touristische Ziel besucht haben, helfen zur Meinungsbildung und vervollständigen das Bild eines Interessenten. Hierbei können Weblogs oder soziale Netzwerke genutzt werden, um realitätsnahe Informationen an andere Personen weiter zu geben. Klar, man kann auch in das Reisebüro gehen und sich beraten lassen. Aber wird einem auch dort über unzufriedene Kunden oder schlechte Hotels berichtet?

- Multimedia Inhalte

Durch die Video- und Fotoplattformen im WWW können Interessenten einen soliden Eindruck von der touristischen Umgebung bekommen. Zudem ermöglichen virtuelle Welten (z.B. in VRML) einen realitätsnahen Einblick in die Umgebung. Diese Art der Darstellung ist bei weitem interessanter und aufschlußreicher als nur die Bilder in den Reiseprospekten.

- Erstellen von Reiseplänen

Aufbauend auf die Meinungen der Nutzer können Interessenten ihren Reiseplan online zusammenstellen und sich mit anderen Nutzern darüber austauschen. Das Web 2.0 wird so zum virtuellen Reiseführer (oft auch "Trip-Planer" genannt), der immer aktuelle Neuigkeiten einbezieht und bei der Urlaubsplanung behilflich ist.

- Durchführen von Buchungen

Durch das Web 2.0 im Zusammenspiel mit dem elektronischen Handel ist es möglich, Buchungen für Flüge, Hotels, Mietwagen und Veranstaltungen online durchzuführen. Nicht zuletzt auch durch die Verwendung von Web Services können Restkapazitäten für Flüge und Hotels bei den verschiedenen Reiseanbietern abgefragt und auf einer Internetpräsenz dargestellt werden.

- Durchführen von Preisvergleichen

Veranstalter von Reisen können ihre Preise über spezielle Programmierschnittstellen (APIs) anbieten, um damit anderen Web 2.0-Diensten einen Preisvergleich zu ermöglichen.

Viele dieser genannten Szenarien werden bereits bei Tourismus-Internetpräsenzen angeboten. So hat die offizielle Internetpräsenz von Holland¹ unter anderem eine Art Weblog (genannt "Triplog"), eine Tag Cloud, Google Maps, Flickr-Fotos und YouTube-Videos im Angebot. Außerdem können Benutzer dort im Sinne eines sozialen Netzwerks Kommentare und Bewertungen zu den verschiedenen touristischen Attraktionen abgeben. Der Benutzer wird hier mit allen notwendigen Informationen über Holland versorgt und bekommt die Information auf eine interaktive Art und Weise präsentiert.

Ein anderes Beispiel ist das NetHotels Travel Network²

. Diese Anwendung erlaubt das Suchen nach Hotels basierend auf eine bestimmte Region. Auf einer Google Map werden bei erfolgreicher Suche alle Hotels der Umgebung angezeigt und sogar die Hotelbuchung ist möglich. Wie viele andere Web 2.0-Anwendungen befindet sich auch diese noch im Beta-Stadium.

Aber Web 2.0-Angebote finden sich auch auf den Internetseiten klassischer Reisedienstleister wie der Österreichischen Bundesbahn³. Dort gibt es speziell für Bahnreisende verschiedene Podcasts über europäische Reiseziele zum Herunterladen. So kann die Zeit im Zug damit verbracht werden, sich die neuesten Informationen des jeweiligen Zielortes anzuhören.

2. Besonderheiten des touristischen Produktes

Aufgrund der Komplexität und Vielseitigkeit des touristischen Produkts, herrscht ein sehr hoher Informationsbedarf bei der Vermarktung des Tourismus. [Egg05] führt folgende Besonderheiten auf:

- Der Tourismus stellt ein immaterielles Produkt dar, dass sich mit seinem Gebrauch auflöst.
- Das touristische Gut kann nicht gelagert werden und besitzt nach Ablauf keinen materiellen Wert mehr.
- Das Gesamtprodukt kann in Teilleistungen zerlegt werden.
- Der Kauf der Dienstleistung erfolgt in der Regel lange vor der Nutzung.
- Zur Leistungserfüllung muss der Kunde physisch präsent sein.

Dies führt generell zu einem gewissen Maß an Unsicherheit beim Kunden. Um den Faktor Unsicherheit zu überwinden, muss der Informationsbedarf vollständig und glaubhaft erfüllt werden (Vgl. [Sch94]). Für die Internetpräsenz bedeutet dies, dass die Benutzeroberfläche und die Inhalte eine hohe Seriösität aufweisen müssen, aber gleichzeitig nicht überladen wirken dürfen. Es muss Vertrauen aufgebaut werden, um den Benutzer vom touristischen Produkt zu überzeugen.

3. Zielgruppenanalyse

Schon in der Entwurfsphase der Internetpräsenz muss geklärt werden, welche Zielgruppen damit später angesprochen werden sollen. Wichtige Punkte sind hierbei das Alter, das Geschlecht, das Bildungsniveau und die Motivation der Zielgruppe. Diese Analyse beeinflusst im Wesentlichen die Aufarbeitung der Informationen für das WWW (Vgl. [Sch03] S 52).

Die Zielgruppen einer klassischen Internetpräsenz für den Tourismus sind natürlich zukünftige Touristen, die einen Urlaub in einem bestimmten Gebiet planen und sich vorab über Urlaubs- und Ausflugsmöglichkeiten in dieser Region informieren möchten. Jedoch lässt sich diese Zielgruppe auch weiter unterteilen. Zum Beispiel in Tages-touristen und Touristen, die über mehrere Tage in der Region bleiben. Darüber hinaus ist eine Unterteilung in Erlebnistouristen und Erholungstouristen möglich. All diese Gruppen sollen sich von der Internetpräsenz gleichermaßen angesprochen fühlen. Natürlich ist es im Sinne des Web 2.0 auch überaus erwünscht, dass die Zielgruppen auch noch nach dem Urlaub die Internetpräsenz besuchen und ihre Urlaubsfotos, Urlaubsvideos oder Eindrücke für andere Benutzer veröffentlichen.

¹<http://us.holland.com>

²<http://www.nethotels.com/>

³<http://podcast.oebb.at/>

Am Rande bedient eine Tourismus-Internetpräsenz auch Biologen, die sich über die Tier- und Pflanzenwelt der Region informieren möchten, ohne sich unbedingt für einen Urlaub dort zu entscheiden. Sie möchten einfach nur auf direktem Wege an die gewünschten Informationen kommen.

Trotzdem eine Tourismus-Internetpräsenz sehr spezifische Informationen enthält, ist die Verschiedenartigkeit der Benutzer im Vergleich mit klassischer Software immer noch sehr groß. Die Ziele variieren vom reinen Informieren bis zum Interagieren oder Publizieren. Erschwerend kommt hinzu, dass bei Internetpräsenzen eine ganz andere Lernkurve als bei klassischer Software vorausgesetzt werden muss. Während Käufer klassischer Computer-Anwendungen durchaus willig sind, sich mit dem Programm auseinander zu setzen und viel Zeit damit verbringen, sind Besucher von Internetpräsenzen eher hektisch und verlassen eine Internetseite schnell wieder, wenn sie sich nicht auf Anhieb damit zurecht finden. Der Grund dafür ist relativ einfach, denn klassische Computer-Anwendungen kosten in der Regel Geld. Internetpräsenzen dagegen kann man in den meisten Fällen kostenlos benutzen (Vgl. [HV03] S 236).

4. Anforderungen an die Benutzeroberfläche

Die Wirkung der Benutzeroberfläche ist nicht zu unterschätzen. Eine gute Benutzeroberfläche schafft Vertrauen, erhöht die Glaubwürdigkeit, unterstreicht die Aussage der Internetpräsenz und unterstützt den Benutzer bei der Suche nach den gewünschten Informationen. Dies gilt für alle Internetpräsenzen, für den Tourismus aber im besonderen Maße.

[Lyo01] nennt folgende Punkte, die bei der Entwicklung einer Benutzeroberfläche beachtet werden müssen:

- Die Fülle an Informationen muss logisch gegliedert und leicht auffindbar sein.
- Die Navigationselemente müssen klar erkennbar sein und sich von dem Rest der Internetseite in irgendeiner Art und Weise hervorheben.
- Benutzer haben ein begrenztes Kurzzeitgedächtnis und können somit nur eine begrenzte Anzahl von Informationen gleichzeitig verarbeiten. Für eine Internetpräsenz macht es Sinn, die Informationen in 5-6 Kategorien zu gruppieren um den Benutzer die Möglichkeit zu geben, die angebotenen Informationen zu verinnerlichen und zu verstehen.
- Benutzer dürfen sich nicht verloren oder alleine gelassen fühlen. Ein Benutzer möchte ständig wissen: Wo bin ich? Wo kann ich hingehen? Wie kann ich dort hingehen? Wie komme ich zurück?
- Die Benutzeroberfläche darf keine Menschen ausschließen und muss barrierefrei gestaltet sein. Zu kleine Schriften, zu schwache Kontraste oder schlecht skalierte Oberflächen, die nur auf großen Monitoren angeschaut werden können, sollten vermieden werden. Gleichzeitig dürfen Benutzeroberflächen nicht für einen speziellen Browser oder eine spezielle Auflösung konzipiert werden.
- Die Verwendung exotischer Browser-Plugins, welche nicht geläufig sind und extra nachinstalliert werden müssen, ist zu unterbinden.

Bei all diesen Punkten ist zu bedenken, dass die Benutzer nicht unendlich geduldig sind und die Internetpräsenz schnell wieder verlassen, wenn sie mit der Benutzeroberfläche nicht zurechtkommen. Ebenso sollte man in Betracht ziehen, dass Benutzer Spaß beim Durchsuchen einer Internetseite haben wollen. Langweilige Designs können hier genauso negativ wirken wie lange, schier endlose Texte (Vgl. [CR03] S 57ff). Gerade im Web 2.0, wo die Benutzer dazu aufgerufen werden eigene Inhalte zu publizieren, müssen Benutzeroberflächen einfach und intuitiv bedienbar sein.

5. Anforderungen an die Informations- und Navigationsstruktur

Eine gute Informationsstruktur ist unabdingbar für eine erfolgreiche Internetpräsenz. Der Besucher einer Internetpräsenz möchte zu jedem Zeitpunkt genau wissen, wo er herkommt, wo er hingehen kann und wie er zurückkommt. Eine intuitive und überschaubare Informationsstruktur hilft dem Benutzer sich zurecht zu finden. [Lyo01] führt in diesem Zusammenhang 4 Informationsstrukturen auf, die bei Internetpräsenzen zum Einsatz kommen:

- Sequentielle Informationsstruktur

Diese Informationsstruktur ist durch einen Prozess gekennzeichnet, der Schritt für Schritt abgearbeitet wird. Typische Beispiele finden sich im Internet bei Umfrageformularen, bei denen die Umfrage auf mehrere Schritte aufgeteilt ist. Um an das Ende der Umfrage zu gelangen, muss man erst alle vorherigen Schritte durchlaufen.

- Hierarchische Informationsstruktur

Dies ist wohl die bedeutendste Informationsstruktur im Internet, gekennzeichnet durch eine Homepage (dt. Heimseite oder Startseite), die baumartig auf alle Internetseiten unterer Ebenen verweist.

- Assoziative Informationsstruktur

Eine assoziative Informationsstruktur findet man bei Verweisen in Form von Glossaren oder Tag Clouds. Dort werden ganz bestimmte Informationen mit einem bestimmten Wort assoziiert. Prinzipiell jede Informationsstruktur, die nicht sequentiell und nicht hierarchisch ist, ist assoziativ.

- Kombinierte Informationsstruktur

Dies ist eine Kombination aller vorher genannten Informationsstrukturen.

Für Navigationszwecke ist vor allem die hierarchische Informationsstruktur von besonderer Bedeutung. Inhalte können in ganz bestimmte Themengebiete geordnet werden, die dann auf der Startseite angezeigt werden. Wichtig hierbei, die Strukturierung sollte sich nicht nur grafisch auf der Internetpräsenz widerspiegeln, sondern auch technisch in Form von aussagekräftigen Verzeichnissen.

„What makes a cool URI? A cool URI is one which does not change.“

⁴Gut lesbare und klar strukturierte Verzeichnisse sind ein wesentlicher Punkt bei der Entwicklung einer benutzerfreundlichen Internetpräsenz. Diese Pfade helfen dem Benutzer, die hierarchische Gliederung der Internetpräsenz besser zu verstehen und gezielt auf bestimmte Inhalte zuzugreifen. Auch hinsichtlich der Suchmaschinen-Optimierung sind aussagekräftige Verzeichnisse von enormer Bedeutung, da die darin enthaltenen Schlüsselwörter die Thematik der Internetpräsenz zusätzlich unterstreichen. Besonders für Content-Management-Systeme (CMS) stellt dies häufig ein Problem dar, denn Inhalte werden hier meistens anhand bestimmter Nummern verwaltet und nicht anhand natürlicher Namen.

6. Anforderungen an den Inhalt

Natürlich sind konkrete Inhalte immer abhängig von den Zielen und den Nutzergruppen der Internetpräsenz, jedoch gibt es beim Online-Tourismus eine Vielzahl an Punkten, die unbedingt beachtet werden müssen:

- Die Aktualität der Inhalte muss sehr hoch und auf die Jahreszeit abgestimmt sein. Informationen über Bademöglichkeiten sind im Winter eher von untergeordneter Bedeutung. Außerdem sind aktuelle Wettermeldungen bzw. Wettervorhersagen sehr interessant, da der Tourismus auch stark wetterabhängig ist.
- Der Tourismus ist ein internationales Geschäftsfeld und es ist zu überlegen, Tourismus-Internetpräsenzen auch in mehreren Sprachen anzubieten.
- Es muss für Interessenten die Möglichkeit gegeben werden, Kontakt mit verantwortlichen Personen aufzunehmen um spezielle Fragen zu klären. Kontaktformulare oder Foren könnten hier eine Lösung sein.
- Es muss Informationen zur Anreise geben. Bei einer Natur-Internetpräsenz muss die Anreise mit öffentlichen Verkehrsmitteln hervorgehoben werden.
- Die touristische Region muss erklärt werden. Insbesondere interaktive Karten mit Übernachtungsmöglichkeiten, Ausflugszielen sowie Rad- und Wanderwegen helfen dem Benutzer bei der Reiseplanung.

⁴Zitat von <http://www.w3.org/Provider/Style/URI>

- Rezensionen und Meinungen anderer Touristen helfen bei der Entscheidungsfindung und stärken zudem noch die Glaubwürdigkeit.

Kapitel 4. Umsetzung mit Drupal

Wir haben im vorherigen Kapitel dargestellt, dass im Web 2.0 die Daten eine entscheidende Rolle spielen. Seien es die Videos von YouTube, die Karten von Google Maps oder die Fotos von Flickr - ohne diese Daten wären viele Web 2.0-Anwendungen nutzlos. Ein Problem wurde jedoch noch nicht gelöst: Wie kann man diese Daten zuverlässig auf einer Internetpräsenz anbieten und verwalten? Eine Lösung hierfür sind sogenannte Web-Content-Management-Systeme (WCMS), die auch Internet-Einsteigern die Möglichkeit geben, eine eigene Internetpräsenz zu erstellen und zu pflegen.

1. Einleitung

Seit dem Beginn der Nutzung des Internets als Informationsplattform wurden Wege gesucht, die ständig wachsenden Informationsmengen effektiv zu verwalten. Im Laufe der Zeit entstanden vielerlei Entwicklungswerkzeuge und -umgebungen, die die Verwaltung ganzer Internetprojekte ermöglichten. Auch wenn es immer leichter wurde, mit diesen Werkzeugen zu arbeiten, sind immer noch fachlich versierte Personen erforderlich, um die Software zu bedienen und die Internetpräsenzen zu erstellen. Erst durch die Entwicklung von Content-Management-Systemen (CMS) konnten auch fachfremde Personen eine Internetpräsenz erstellen und verwalten. Diese Systeme werden in der Regel auf einem Web-Server installiert und können dann von einer Vielzahl von Personen bedient werden. Mittlerweile stellen Content-Management-Systeme ein zuverlässiges Mittel dar, um die Qualität und Aktualität der Internetpräsenz zu verbessern sowie dauerhaft Kosten und Zeit zu sparen. Das Besondere an diesen Systemen ist, dass Inhalt und Layout der Internetpräsenz voneinander getrennt sind und sich die Benutzer keine Gedanken über das spätere Erscheinungsbild machen müssen (Vgl. [ZTZ02] S 73-77).

Speziell für diese Bachelorarbeit wurde ein Content-Management-System gesucht, welches leicht durch neue Module für das Web 2.0 erweitert werden kann, nichts kostet und gut dokumentiert ist. So fiel die Wahl auf das Open-Source Content-Management-System Drupal.

2. Was ist Drupal?

„Drupal ist ein datenbankbasiertes Content Management Framework, geschrieben in PHP.“ ([Gra06] S 19)

Framework heißt im Englischen so viel wie Rahmenwerk oder auch Fachwerk. In der Softwareentwicklung wird der Begriff Framework verwendet, um einen Rahmen zur Verfügung zu stellen, in dem ein Programmierer eine Anwendung erstellen kann. Übertragen auf Drupal bedeutet dies, dass Drupal vorgefertigte Funktionen zur Verfügung stellt, die der Programmierer bei der Entwicklung neuer Module verwenden kann. Diese vorgefertigten Funktionen heißen *Hooks*, dazu aber später mehr. Zunächst folgt ein kleiner Einblick in die Geschichte von Drupal.

Die Anfänge von Drupal liegen an der Universität Antwerpen, als die Studenten Hans Snijder und Dries Buytaert ein kleines Programm entwickelten, mit dem sie untereinander Nachrichten schreiben und Kurse an der Universität bewerten konnten. Die Software lief zu Beginn nur lokal im Universitätsnetzwerk, als aber Dries sein Studium beendete, wollte er weiterhin mit seinen Freunden in Kontakt bleiben. Er versuchte zunächst, die Domain dorp.org (niederl. Dorf) auf seinen Namen zu registrieren. Leider verschrieb er sich bei der Anmeldung und registrierte fälschlicherweise die Domain drop.org. Der Gruppe gefiel der Name jedoch so gut, dass sie dabei blieben und die Software unter diesem Namen weiter entwickelten.

Im Januar 2001 beschloss Dries, die Software unter der allgemeinen GNU-Lizenz zu veröffentlichen. Nun konnten auch andere Programmierer an der Software experimentieren und neue Module dafür entwickeln. Gleichzeitig wurde die Software in Drupal, eine Art Lautschrift zu dem niederländischen Wort Druppel (dt. Tropfen), umbenannt (Vgl. [Gra06] S 22-23).

Eine stetig wachsende Entwicklergemeinde arbeitet seit dem an der Verbesserung der Software und entwickelt neue Module für neue Anwendungsgebiete. Seit 2002 gibt es etwa jedes Jahr Sprünge in der Versionsnummer, die Version 6 wird Ende 2007 erwartet.

Obwohl Drupal ursprünglich nur als Community-Portal entwickelt wurde, wird es heutzutage für die Verwaltung privater und kommerzieller Internetpräsenzen jeglicher Art eingesetzt. Beispiele sind die Umweltschutzorganisa-

tion Greenpeace¹, das Medienunternehmen Warner Bros. Records² oder auch das Nachrichtenmagazin Linux Journal³. Im November 2005 wurde Drupal bereits bei über 54700 Internetpräsenzen eingesetzt⁴.

3. Vor- und Nachteile von Drupal

Aus vielerlei Hinsicht ist Drupal ein geeignetes Content-Management-System für die Verwaltung einer Nationalpark-Internetpräsenz. Hervorzuheben sind hier:

- Drupal ist eine freie Software und steht unter der GNU General Public License.
- Drupal hat eine große Entwicklergemeinde und wird ständig hinsichtlich Sicherheitslücken überarbeitet.
- Drupal ist modular aufgebaut und bietet somit die Möglichkeit, nachträglich Module einzubinden und den Funktionsumfang zu erweitern.
- Es ist sehr gut dokumentiert.
- Es bietet eine ausgereifte Benutzerverwaltung und eignet sich damit hervorragend zum Administrieren von *User Generated Content* (UGC).
- Inhalte lassen sich über das Taxonomy-Modul einfach kategorisieren, insbesondere für die Tier- und Pflanzendatenbank sowie für die Einträge im Weblog bietet dies entscheidende Vorteile.
- Es gibt ein ausgezeichnetes Logging und Statistik-System. So können fehlerhafte Logins genauso eingesehen werden wie die eingegebenen Suchbegriffe in das Suchfeld.

Dem Gegenüber steht eine relativ kleine Anzahl von Nachteilen:

- Das Hochladen von Bildern wird in der Standard-Installation nicht unterstützt. Es gibt zwar externe Module zur Bildverwaltung - die auch in dieser Arbeit verwendet wurden - jedoch sind diese wenig benutzerfreundlich.
- Es gibt in der Standard-Installation keinen *WYSIWYG*-Editor, der das Erstellen von Internetseiten erleichtern könnte.
- Es gibt keine native Unterstützung zum Importieren von Daten wie zum Beispiel Tabellen im CSV-Format. Ebenso gibt es keine Export-Funktion für die erstellten Inhalte.
- Die Internetpräsenz (*Frontend*) und die Verwaltungsoberfläche (*Backend*) sind eher unscharf voneinander getrennt. Gerade bei unerfahrenen Benutzern führt dies häufig zu Verwirrungen.

4. Drupal Terminologie

4.1. Knoten

Alle Inhalte, die mit Drupal erstellt werden, sind Knoten (engl. node). Ein Knoten ist ein strukturierter Inhaltstyp, zum Beispiel eine einfache Internetseite, ein Kalendereintrag oder eine Google Map. Alle Knoten zusammen ergeben die komplette Internetpräsenz. Jedes Drupal-Modul definiert in der Regel einen bestimmten Knotentyp.

4.2. Modul

Drupal ist streng modular aufgebaut und besteht aus einzelnen Modulen. Jedes dieser Module realisiert eine gewisse Funktionalität in Drupal. So gibt es zum Beispiel das Forum-Modul zum Erstellen und Verwalten von Foren oder das Contact-Modul zum Erstellen eines Kontakt-Formulares. Es wird hierbei unterschieden in *Core-Module*, welche Bestandteil des Drupal-Kerns sind und bei der Installation von Drupal bereits aktiviert sind, und

¹<https://www.greenpeace.org.uk>

²<http://warnerbrosrecords.com>

³<http://www.linuxjournal.com>

⁴Mehr Informationen dazu unter <http://drupal.org/node/36748>

Contrib-Module, welche meistens von Dritten angeboten werden und separat installiert werden müssen. Alle in dieser Bachelorarbeit entwickelten Module sind *Contrib-Module*.

4.3. Rolle

Jeder Benutzer von Drupal, und damit auch jeder Besucher der verwalteten Internetpräsenz, nimmt eine bestimmte Rolle ein (engl. *role*). Anhand dieser Rollen werden die Zugriffsrechte auf die verschiedenen Drupal-Funktionen festgelegt. Standardmäßig gibt es in Drupal die Rolle "anonymous user" (dt. anonymer Benutzer) und "authenticated user" (dt. registrierter Benutzer). Zur ersten Gruppe gehören alle Besucher der Internetpräsenz, die kein eigenes Benutzerkonto bei Drupal besitzen. Zur zweiten Gruppe gehören alle Benutzer der Internetpräsenz, die das Registrierungsformular auf der Internetpräsenz ausgefüllt haben und somit über ein eigenes Benutzerkonto verfügen. Für diese Arbeit wurden noch weitere Rollen definiert, mehr dazu steht im Pflichtenheft im Anhang.

4.4. Taxonomie

Unter Taxonomie (engl. *Taxonomy*) versteht man im Allgemeinen die Einteilung bzw. Kategorisierung von Dingen. In Bezug auf Drupal bedeutet dies, dass alle erstellten Inhalte einer oder mehreren Kategorien zugeordnet werden können. Diese Kategorien lassen sich auch hierarchisch gliedern, wodurch ein sehr ausgereiftes Klassifikationssystem entstehen kann (Vgl. [Gra06] S 70). Taxonomien sind in dieser Arbeit für das Lifeform-Modul von immenser Bedeutung. Erst dadurch lassen sich Tiere und Pflanzen in eine bestimmte Art, Gattung und Familie einordnen.

4.5. Block

Blöcke (engl. *blocks*) befinden sich auf der linken und rechten Seite sowie im Kopf- und Fußbereich der mit Drupal erstellten Internetpräsenz. In diesen Blöcken können Navigationselemente oder Inhalte angezeigt werden, wobei das Aussehen durch eine Vorlage verändert werden kann (Vgl. [Gra06] S 67).

4.6. Theme

Eine Theme ist eine Art Schablone, die das Erscheinungsbild der Internetpräsenz bestimmt. So werden im Theme die Farben, Schriftarten, Schriftgrößen, Hintergrundbilder und die generelle Aufteilung der Oberfläche der Internetpräsenz bestimmt. In Drupal wird das Theme durch mehrere PHP- und CSS-Dateien realisiert. Der Vorteil des Themings besteht darin, dass das Erscheinungsbild der Internetpräsenz verändert werden kann, ohne die Inhalte selbst zu verändern.

4.7. Front- und Backend

Wie in jedem anderen Content-Management-System gibt es auch in Drupal eine Unterteilung der Benutzeroberfläche in *Frontend* und *Backend*.

Das Frontend ist die eigentliche Internetpräsenz, wie sie jeder Besucher im Internet sehen kann. Dies umfasst auch Inhalte, die man erst nach einer Registrierung einsehen kann. Im Allgemeinen dient das Frontend also nur zur Darstellung der Information. Oftmals ist es für einen Laien gar nicht ersichtlich, dass die betrachteten Internetseiten von einem Content-Management-System generiert werden.

Das Backend hingegen ist die Verwaltungsoberfläche, in der die einzelnen Seiten der Internetpräsenz erstellt und bearbeitet werden können. Um zum Backend zu gelangen, benötigt man immer einen Benutzernamen und ein Passwort und muss sich unter einer bestimmten URL einloggen. Das Backend wird nur von einem bestimmten Personenkreis verwendet, normale Besucher der Internetpräsenz haben in der Regel keine Möglichkeit, zum Backend zu gelangen.

5. Installation von Drupal

Drupal wird auf einem Web-Server ausgeführt und benötigt eine Datenbankanbindung zum Speichern der erstellten Inhalte. Die aktuelle Version kann im Internet unter⁵ heruntergeladen werden. Dieses Archiv bietet eigentlich eine

⁵<http://www.drupal.org/>

lauffähige Version von Drupal, für den Zweck dieser Arbeit müssen jedoch noch andere Module heruntergeladen und installiert werden. Dies sind im Einzelnen:

- Image 5.x-1.0⁶, um den Benutzern die Möglichkeit zu geben, Bilder hochzuladen;
- Event 5.x-1.0⁷, um einen Kalender auf der Internetpräsenz anzuzeigen;
- TinyMCE 2.1.2⁸, um die Formatierung von Texten in einem WYSIWYG-Editor⁹ zu ermöglichen;
- TinyMCE 5.x-1.9¹⁰, um den TinyMCE Editor in Drupal zu aktivieren;
- IMCE 5.x-1.0¹¹, um Bilder direkt in Texte einzufügen.

Nachdem das Drupal-Archiv entpackt wurde, muss im Verzeichnis "sites/all" ein Unterverzeichnis "modules" erstellt werden, in dem alle extra heruntergeladenen Module extrahiert werden müssen. Ebenso werden dort die in dieser Bachelorarbeit entwickelten Module hinein kopiert. Die Module werden dort automatisch von Drupal erkannt und müssen später nur noch aktiviert werden. Außerdem muss ein Verzeichnis "files" im Hauptverzeichnis erstellt werden, um Platz für die Benutzerdateien zu schaffen. In diesem Verzeichnis müssen für den Web-Server Schreibrechte bestehen. Alles in allem ergibt dies folgende Verzeichnisstruktur.

```
files
includes
misc
modules
profiles
scripts
sites
  all
    modules
      accommodation
      event
      googlemap
      image
      imce
      lifeform
      tinymce
themes
.htaccess
CHANGELOG.txt
cron.php
index.php
INSTALL.mysql.txt
INSTALL.pgsql.txt
install.php
INSTALL.txt
LICENSE.txt
MAINTAINERS.txt
robots.txt
update.php
UPGRADE.txt
xmlrpc.php
```

Nun kann man das Drupal-Verzeichnis im Browser aufrufen und über einen Assistenten die Datenbankverbindung konfigurieren. Nach dem Klick auf "Save configuration" werden die notwendigen Datenbank-Tabellen angelegt und eine Datei namens settings.php im Verzeichnis "sites/default/" erstellt.

Ruft man nun Drupal erneut auf, erläutert eine Seite die nächsten Konfigurationsschritte. Wie unter Punkt 1 beschrieben, muss zunächst ein Administrator eingerichtet werden. Dafür muss ein Benutzername, ein Passwort

⁶<http://drupal.org/project/image>

⁷<http://drupal.org/project/event>

⁸<http://tinymce.moxiecode.com/download.php>

⁹WYSIWYG ("What You See Is What You Get") ist die Bezeichnung für Programme, welche die spätere Darstellung im Web simulieren und somit sehr Benutzerfreundlich sind.

¹⁰<http://drupal.org/project/tinymce>

¹¹<http://drupal.org/project/imce>

und eine E-Mail-Adresse festgelegt werden. Nach diesem Schritt ist die Installation von Drupal abgeschlossen und es kann konfiguriert werden.

6. Konfiguration von Drupal

6.1. Aktivierung der benötigten Module

Nach der erfolgreichen Installation von Drupal auf dem Web-Server muss man mit dem Administrator-Account unter "Administer > Site Building > Modules" die benötigten Module aktivieren. Für die Nationalpark-Internetpräsenz benötigen wir aus dem Drupal-Kern die folgenden Module:

- Blog
- Color
- Comment
- Contact
- Forum
- Locale
- Menu
- Path
- Search
- Taxonomy

Zusätzlich müssen aber noch die folgenden Zusatz-Module sowie die selbst entwickelten Module aktiviert werden.

- Accommodation
- Event
- Google Map
- Image
- Image Attach
- IMCE
- Lifeform
- Map Marker
- Map Path
- Observation
- TinyMCE

An diesem Punkt ist Drupal praktisch einsatzbereit, jedoch müssen noch die einzelnen Module auf die speziellen Bedürfnisse der Nationalpark-Internetpräsenz abgestimmt werden und Benutzergruppen eingerichtet werden.

6.2. Konfiguration der Benutzergruppen und Benutzerrechte

Im Pflichtenheft im Anhang dieser Arbeit sind 6 Benutzergruppen angegeben, die für die Internetpräsenz vorgesehen sind. Dies sind im Einzelnen:

- Unregistrierter Besucher
- Registrierter Besucher
- Unterkunftsanbieter
- Veranstaltungsmanager
- Nationalparkverwalter
- Superadministrator

Jede dieser Gruppen hat verschiedene Rechte und kann bestimmte Funktionen ausführen. Mit Hilfe des Administrator-Accounts werden nun unter "Administer > User Management > Roles" die oben genannten Benutzergruppen angelegt. Anschließend können unter "Administer > User Management > Access Control" die folgenden Rechte für die Benutzergruppen vergeben werden:

Anonymous User: access content, access tinymce, search content, use advanced search

Authenticated User: access content, access tinymce, access imce, search content, use advanced search, access comments, post comments, post comments without approval, create images, edit own images, create mapmarker, create observation, edit observation

Event Administrator: create events, edit own events

Accommodation Provider: create accommodation, edit own accommodation

Park Officer: edit all accommodation, administer comments, administer forums, create forum topics, edit own forum topic, administer googlemap pages, create googlemap pages, edit googlemap pages, edit all googlemap pages, create lifeform, edit all lifeforms, edit lifeform, administer mapmarker, add flickr and youtube content, create mappath, edit all mappaths, edit own mappath, administer nodes, create observation, edit observation, edit all observations, access user profiles, administer users

Die Benutzergruppen Event Administrator, Accommodation Provider und Park Officer erben automatisch die Rechte der Benutzergruppe Authenticated User. Der Superadministrator besitzt automatisch alle Rechte und muss nicht extra konfiguriert werden.

6.3. Konfiguration des Drupal-Kerns

Folgende Einstellungen müssen vorgenommen werden:

- Unter "Administer > Site configuration > Site information" müssen einige Informationen zur Internetpräsenz eingegeben werden. Für diese Bachelorarbeit wird als Name "Nationalpark Unteres Odertal", als Footer

```
Diese Internetpräsenz entstand im Rahmen einer Bachelorarbeit von Sascha Nehls.<br />
Copyright © Sascha Nehls und BTU Lehrstuhl Internet-Technologie
<a href="http://creativecommons.org/licenses/by-nc-sa/2.0/de/">Some rights reserved.</a>
```

sowie als E-Mail-Adresse die Adresse "sascha.nehls@tu-cottbus.de" verwendet. Diese E-Mail-Adresse erscheint als Absender bei Bestätigungs-E-Mails wie z.B. bei der Registrierung von neuen Nutzern oder bei der Buchung von Unterkünften (sofern der Unterkunftsanbieter keine eigene E-Mail-Adresse hinterlegt hat). Im Feld "Anonymous User" wird "Anonymer Benutzer" eingetragen und im Feld "Default Front Page" wird "willkommen" eingetragen.

- Unter "Administer > Site configuration > Input Formats" muss "Full HTML" als Standardwert gesetzt werden, anderenfalls werden die Vorschautexte im Informationsfenster der Google Map nicht korrekt formatiert.

- Unter "Administer > Content management > Post settings" muss die Länge der Teaser auf 200 Zeichen begrenzt werden, da die Vorschautexte sonst evtl. nicht in das Fenster auf der Google Map passen.
- Unter "Administer > Site configuration > Clean URLs" muss der URL-Test durchgeführt werden. Bei diesem Test wird überprüft, ob der Web-Server für diese Funktion geeignet ist. Nach erfolgreichem Test kann diese Funktionalität aktiviert werden und somit die URLs der einzelnen Inhaltsseiten manuell angepasst werden.
- Unter "Administer > Content management > Content types" sollte für die Inhaltstypen Page, Pflanze und Tier die Kommentarfunktion deaktiviert werden. Damit können Nutzer standardmäßig keine Kommentare zu Seiten dieses Typs abgeben.
- Unter "Administer > Logs > Status report" muss ein Cron manuell ausgeführt werden. Dieser Cron kann dazu verwendet werden, regelmäßig anfallende Aufgaben (wie z.B. das Löschen von temporären Dateien) automatisch auszuführen. Für diese Drupal-Konfiguration genügt es, wenn wir den Cron nur einmal ausführen.
- Unter "Administer > Site building > Themes" sollten im Abschnitt "Display post information on" nur die Knotentypen "Beobachtung", "Blog Entry" und "Forum Topic" aktiviert sein. Dies bewirkt, dass bei diesen Knotentypen ein kleiner Hinweistext zum Autor eingeblendet wird.

Mit dem Abarbeiten dieser Liste wurden die Einstellungen im Drupal-Kern vorgenommen. Nun folgt noch die Konfiguration der zusätzlich installierten Module.

6.4. Konfiguration des TinyMCE-Moduls

Das TinyMCE-Modul erlaubt die Verwendung eines WYSIWYG-Editors beim Erstellen und Bearbeiten von bestimmten Knotentypen. Zunächst muss jedoch unter "Administer > Settings > TinyMCE" ein Profil angelegt werden. Dort kann man unter anderem festlegen, bei welchen Knotentypen der WYSIWYG-Editor angezeigt werden soll. Für diese Arbeit ist es sinnvoll, den Editor für die folgenden Seiten zu deaktivieren:

```
user/*
comment/*
admin/*
contact
```

Die anderen Einstellungen können in der Standardeinstellung belassen bzw. nach belieben angepasst werden.

6.5. Konfiguration des Event-Moduls

Dieses Modul ermöglicht die Darstellung eines Kalenders und das Verwalten von Veranstaltungen. Dadurch können die Besucher der Internetpräsenz gezielt über bevorstehende Ereignisse informiert werden. Über das Taxonomy-Modul lassen sich die Veranstaltungen auch in bestimmte Kategorien einteilen und ermöglichen so das Suchen von ganz bestimmten Veranstaltungstypen.

Die Ausgabe des Event-Moduls wurde geringfügig optimiert. So wurde in der Datei "event.css" im Event-Verzeichnis die CSS-Klasse "event-calendar" modifiziert und die Textausrichtung auf zentriert gesetzt. Hier die geänderte Klassenbeschreibung:

```
.event-calendar td {
  border: 1px solid #bbb;
  color: #777;
  text-align: center;
  vertical-align: top;
  margin: 0;
  padding: 0;
}
```

Außerdem wurden die Abkürzungen der Wochentage in der Datei event.module in Zeile 1314 geändert, da diese zu lang waren und zu viel Platz auf der Internetpräsenz benötigten.

```
<?php
$weekdays = array(array('day' => 'Mon', 't' => t('M')), array('day' => 'Tue', 't' =>
```

```
t('D')), array('day' => 'M', 't' => t('M')), array('day' => 'Thu', 't' => t('D')),
array('day' => 'Fri', 't' => t('F')), array('day' => 'Sat', 't' => t('S')),
array('day' => 'Sun', 't' => t('S')));
?>
```

6.6. Konfiguration des Contact-Moduls

Das Contact-Modul dient dazu, ein Kontaktformular auf der Internetpräsenz bereitzustellen. Dieses Kontaktformular kann für Anfragen jeglicher Art verwendet werden. Zunächst muss allerdings ein Formular mit mindestens einer Kategorie unter "Administer > Site building > Contact form" angelegt werden. Für die Internetpräsenz dieser Arbeit werden die Kategorien "Allgemeine Anfrage", "Anfrage an die Nationalparkverwaltung", "Anfrage an den Webmaster", "Anfrage zur Buchung von Unterkünften" sowie "Anfrage zur Tier- und Pflanzendatenbank" mit den entsprechenden Empfänger-Adressen angelegt. Das erstellte Kontaktformular wird dann später auf der Internetpräsenz verlinkt.

6.7. Konfiguration des Taxonomy-Moduls

Eines der wichtigsten Module zur Erstellung einer Web 2.0-Internetpräsenz ist das Taxonomy-Modul. Dieses leistungsstarke Modul erlaubt die Klassifizierung sowie Kategorisierung jeglicher Inhalte. Das Besondere hierbei ist, dass auch Hierarchien zwischen den einzelnen Kategorien hergestellt werden können und somit auch sehr komplexe Taxonomien möglich sind.

Über "Administer > Content Management > Categories" lassen sich für jeden Knotentyp Taxonomien erstellen. Für das Accommodation-, Lifeform-, Event- und Blog-Modul sind solche Taxonomien zwingend erforderlich. Folgende Vokabeln wurden bei den jeweiligen Modulen angelegt:

- Accommodation-Modul: Hotel, Pension, Gästehaus, Ferienhaus, Privatzimmer und Bungalow
- Blog-Modul: Nationalpark, Forschung, Erlebnis, Tierwelt und Pflanzenwelt
- Event-Modul: Konzert, Theater, Führung und Fest
- Lifeform-Modul: Hier wurde eine hierarchische Gliederung der verschiedenen Arten, Familien und Gattungen erstellt.

6.8. Konfiguration des Image- und Image Attach-Moduls

Das Image-Modul wird zwingend benötigt, um die Bilder für die verschiedenen Internetseiten sowie für die Unterkünfte und die Tier- und Pflanzen hochladen zu können. Dabei wird automatisch eine verkleinerte Kopie des Bildes erzeugt, welche dann beim Betrachten der entsprechenden Inhalte angezeigt wird.

Zunächst muss jedoch im Administrationsbereich unter "Administer > Site configuration > Image" die maximale Größe der Bilder angepasst werden. Im Feld "Maximum Upload Size" trägt man am besten 3000 KB ein, damit auch größere Bilder direkt hochgeladen werden können. Im Feld "Original" trägt man am besten eine Größe 640x480 Pixel ein, im Feld "Thumbnail" am besten 120x80 Pixel ein und im Feld "Preview" 320x240 Pixel.

Eine besondere Bedeutung nimmt das Modul Image Attach ein. Hiermit ist es möglich, gleichzeitig mit dem Erstellen eines Knotens ein Bild anzufügen. Leider erlaubt das Modul die Verwaltung der Bilder nicht per Nutzer, weswegen die Option "Attach existing images" unter "Administer > Site configuration > Image attach" auf "disabled" gesetzt werden muss. Anderenfalls könnten Nutzer die Bilder anderer Nutzer zu ihren Knoten hinzufügen.

Außerdem muss das Image Attach-Modul noch für die Knotentypen Beobachtung, Pflanze, Tier und Unterkunft unter "Administer > Content management > Content types" aktiviert werden. Mit dieser Einstellung erscheint beim Erstellen oder Bearbeiten eines Knotens diesen Typs automatisch das Formular zum Hochladen von Bildern.

6.9. Konfiguration des Locale-Moduls

Da die Drupal Verwaltungsoberfläche standardmäßig nur in Englisch ist, muss mit dem Locale-Modul eine deutsche Sprachdatei nachinstalliert werden. Dazu muss zunächst unter "Administer > Site configuration > Localization" im Abschnitt "Add language" die Deutsche Sprache aktiviert werden. Anschließend muss von der "Drupal Translation Page"¹² ein deutsches Sprachmodul heruntergeladen werden. Dieses Archiv enthält die Übersetzung für die Verwaltungsoberfläche und für viele Standardmodule. Anschließend muss die Datei "de.po" aus diesem Archiv in Drupal hochgeladen werden. Dazu steht eine eigene Benutzeroberfläche zur Verfügung, die man im Abschnitt "Import" findet. Nachdem dort die Sprachdatei hochgeladen wurde, ist die Verwaltungsoberfläche von Drupal in deutscher Sprache.

6.10. Konfiguration des Search-Moduls

Damit das Suchfeld auf der Internetpräsenz sichtbar ist, muss zunächst unter "Administer > Site building > Blocks" das Suchformular in einem sichtbaren Bereich der Internetpräsenz aktiviert werden. Anschließend muss unter "Administer > Logs > Status report" der Inhalt der Internetpräsenz indiziert werden. Dazu richtet man entweder einen automatischen Cron ein oder führt den Cron in regelmäßigen Abständen manuell aus. Dies ist dringend notwendig, da die Suche über Schlüsselwörter funktioniert, welche in der Datenbank gespeichert werden. Ohne die Indizierung werden keine Schlüsselwörter gespeichert und die Suche bringt keine Resultate.

6.11. Konfiguration der Benutzeroberfläche

Drupal wird standardmäßig mit einer Vielzahl von verschiedenen Themes ausgeliefert. Diese Themes bestimmen das äußere Erscheinungsbild der Internetpräsenz wie Farben, Schriften oder Navigationsbereiche. Da der Schwerpunkt dieser Arbeit nicht auf der Entwicklung einer neuen Benutzeroberfläche liegt, wird nur das Standard-Theme angepasst und das Seitenlogo sowie das Favicon¹³ ausgetauscht.

Dazu muss unter "Administer > Site building > Themes" die Konfigurationsseite des Themes "Garland" aufgerufen werden. Im Abschnitt "Color Scheme" werden die folgenden Werte eingetragen:

```
Color set: Custom
Base color: #c9c497
Link color: #0c7a00
Header top: #3b5228
Header bottom: #416029
Text color: #494949
```

Mit diesen Einstellungen bekommt die Benutzeroberfläche einen grünen Farbanstrich und sieht nicht mehr so "kalt" aus. Zur Auflockerung der Benutzeroberfläche wurde ein Foto eines Eisvogels¹⁴ modifiziert, um es als Logo zu verwenden. Außerdem wurde ein neues Favicon erstellt. Beide Elemente können im Abschnitt "Logo image settings" und "Shortcut icon settings" hochgeladen werden.

6.11.1. Anpassung von style.css

Da Inhalte in verkürzter Form auf einer Google Map angezeigt werden können, mussten ein paar Änderungen an der Datei style.css im Verzeichnis "themes/garland" vorgenommen werden.

```
.node {
  /* border-bottom: 1px solid #e9eff3; removed because the border underneath each node
  doesn't look nice in the google map info window */
  margin: -1.5em -26px 1.5em;
  padding: 1.5em 26px;
}

/* add an icon before the category links */
.terms ul.links li.first {
  margin-left: 0;
```

¹²<http://drupal.org/project/translations>

¹³Favicon ist die Bezeichnung für das kleine Vorschau-Bild einer Internetpräsenz, welches in der Adressleiste des Web-Browsers angezeigt wird.

¹⁴Das Foto wurde mit freundlicher Genehmigung von Claude Ruchet (<http://www.ruchet.com>) verwendet.


```
margin-right: 0;
padding-right: 0;
background: url(images/mini-category.png) no-repeat 0 2px;
padding-left: 15px;
}

/* add an icon before the booking links */
.links ul.links li.accommodation_book {
margin-left: 0;
margin-right: 0;
padding-right: 0;
background: url(images/mini-book.png) no-repeat 0 2px;
padding-left: 15px;
}

/* add an icon before the read more links */
.links ul.links li.node_read_more {
margin-left: 0;
margin-right: 0;
padding-right: 0;
background: url(images/mini-readmore.png) no-repeat 0 2px;
padding-left: 15px;
}

/* add an icon before the comment links */
.links ul.links li.comment_add {
margin-left: 0;
margin-right: 0;
padding-right: 0;
background: url(images/mini-comment.png) no-repeat 0 2px;
padding-left: 15px;
}

/* align the preview images to the right side of the page */
div.image-attach-teaser {
float:right;
padding-left:5px;
}

/* this has been added to create a good looking border around the attached images */
div.image-attach-body img {
border: #E7EEE2 1px solid;
padding: 4px;
margin: 4px;
}

/* this has been added to create a good looking border around special images */
div.content img.bordered {
border: #E7EEE2 1px solid;
padding: 4px;
margin: 4px;
}

/* this has been added to create a good looking border for preview images */
div.content a img.bordered:hover {
border: #CFE2D0 5px solid;
padding: 0px;
margin: 4px;
}

p {
margin: 0.6em 0 1.2em;
padding: 0;
text-align:justify; /* this has been added to have a nicer alignment */
}
```

Damit ist auch der letzte Schritt der Anpassung von Drupal vorgenommen und wir können mit der Erstellung von Inhalten beginnen.

Kapitel 5. Entwicklung neuer Drupal-Module für den Tourismus

Im vorherigen Kapitel wurde die Installation und Konfiguration der frei verfügbaren Module für Drupal beschrieben. Um nun den Funktionsumfang von Drupal hinsichtlich der Prinzipien des Web 2.0 zu erweitern, wurden insgesamt 5 neue Module entwickelt. In diesem Kapitel wird die Entwicklung und Konfiguration dieser Module beschrieben.

1. Modul-Entwicklung im Allgemeinen

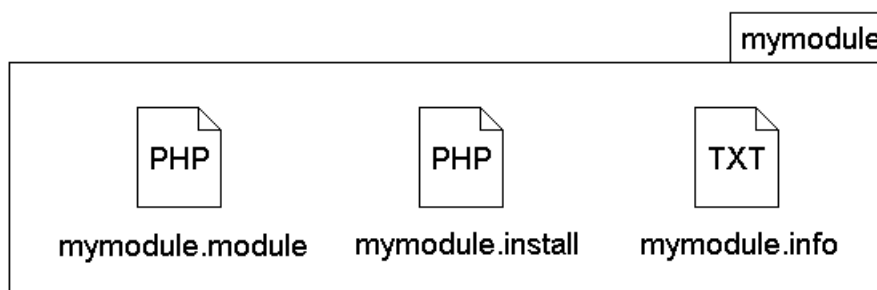
1.1. Coding-Standard

Alle entwickelten Module folgen dem Drupal Coding-Standard¹. Dies betrifft insbesondere die Formatierung von Kontrollstrukturen, die Bezeichnung der Variablen und Funktionen sowie die Kommentierung des Quelltextes. Alle Funktionen wurden mit einer kurzen Beschreibung der Parameter und der Funktionsweise ausgestattet, um das Verständnis des Quelltextes zu erhöhen. Die Variablen und Funktionsnamen sowie die Kommentare sind in Englisch, die Benutzeroberflächen wurden jedoch ins Deutsche übersetzt.

1.2. Genereller Aufbau eines Drupal-Moduls

Jedes Modul besteht mindestens aus den folgenden 3 Dateien, wobei "mymodule" jeweils durch den Namen des Moduls ersetzt werden muss. Diese 3 Dateien müssen sich in einem extra Verzeichnis befinden, welches unter "/sites/all/modules/mymodule" abgelegt wird. Gelegentlich besitzen Module auch noch externe JavaScript- oder CSS-Dateien, welche ebenfalls in diesem Verzeichnis gespeichert werden sollten.

Abbildung 5.1. Übersicht der notwendigen Dateien zur Erstellung eines Drupal-Moduls



- mymodule.info

Hier werden Angaben zum Modulnamen, zur Version sowie zur Abhängigkeit von anderen Modulen gemacht. Diese Datei ist sehr klein und umfasst nur wenige Zeilen.

- mymodule.install

In dieser Datei werden die Datenbank-Tabellen definiert, die zur Ausführung des Moduls notwendig sind. Außerdem befinden sich dort auch die Anweisungen zum Löschen der Tabellen, wenn das Modul deinstalliert wird.

- mymodule.module

Hier befindet sich der eigentliche Quelltext, welcher die Funktionsweise des Moduls bestimmt. Dieser Quelltext ist in PHP geschrieben und basiert auf *Hooks*, um die gewünschten Funktionalitäten zu implementieren. Jeder Hook stellt dabei eine PHP-Funktion in der Moduldatei (mymodule.module) dar und übernimmt bestimmte

¹<http://drupal.org/coding-standards>

Aufgaben. Im Anhang dieser Arbeit befindet sich der Quelltext eines Beispielmodes, welches die wichtigsten Hooks auf einfache Weise erklärt.

Zur Implementierung neuer Module bietet Drupal neben den Hooks eine eigene Programmierschnittstelle (API)² für das Menüsystem, für Datenbankzugriffe, für Formulare, für Datei-Uploads und weitere Komponenten an.

1.3. Drupal-Hooks

Die Implementierung eines neuen Moduls in Drupal geht immer einher mit der Implementierung der jeweiligen Hooks. Diese Hooks sind PHP-Funktionen, die in der Datei `mymodule.module` definiert werden müssen. Drupal kommuniziert über diese Hooks mit dem neu entwickelten Modul und führt die in den Hooks implementierten Aktionen aus. Jeder Hook, oder anders ausgedrückt, jede PHP-Funktion im Quelltext des Modules, hat dabei ganz bestimmte Parameter und Rückgabewerte. Die Funktionsnamen müssen immer nach dem Schema `modulname_hookname()` vergeben werden (zum Beispiel `googlemap_view()`). Unter ³ findet sich eine Übersicht mit allen Hooks in Drupal 5.

Folgende Hooks sind am wichtigsten für die Entwicklung neuer Module und wurden auch in dieser Arbeit verwendet:

- `hook_node_info()` In diesem Hook wird festgelegt, welche Inhalte man mit dem entsprechenden Modul erstellen kann.
- `hook_menu()` Dieser Hook wird benutzt, um Pfade (URLs) des Moduls zu konfigurieren und um Einträge in der Navigationsleiste zu erstellen.
- `hook_view()` Hiermit kann die Ausgabe der modulspezifischen Internetseiten angepasst werden.
- `hook_form()` Dieser Hook dient zur Erstellung von Formularen. Diese Formulare werden unter anderem angezeigt, wenn ein Knoten des spezifischen Modules erstellt oder bearbeitet wird.
- `hook_validate()` Dieser Hook ist nur im Verbund mit `hook_form()` sinnvoll und dient der Validierung der abgeschickten Formulardaten.
- `hook_insert()` Dieser Hook wird aufgerufen, wenn ein neuer modulspezifischer Knoten erstellt wird und dient zur Speicherung der Daten.
- `hook_update()` Dieser Hook wird aufgerufen, wenn ein Knoten des entsprechenden Modules aktualisiert wird.
- `hook_delete()` Dieser Hook wird aufgerufen, wenn ein Knoten des entsprechenden Modules gelöscht wird.
- `hook_load()` Hiermit werden die modulspezifischen Daten geladen und in einem Array gespeichert. Bei Anzeige des Knotens können diese Daten dann direkt ausgegeben werden.
- `hook_prepare()` Dieser Hook wird direkt nach `hook_load()` aufgerufen und dient zur Manipulation der Daten.

1.4. Benutzergruppen und Benutzerrechte

Mit Drupal lassen sich unterschiedliche Benutzergruppen verwalten und mit besonderen Rechten ausstatten. Diese Rechte können in Drupal unter "Administer > User management > Access control" vergeben werden. Typische Rechte sind zum Beispiel "Inhalt erstellen", "Eigene Inhalte löschen" oder "Fotos hochladen". Jedes Drupal-Modul kann dabei eigene Benutzerrechte definieren, um für spezifische Benutzergruppen bestimmte Funktionen zu ermöglichen. Dazu muss lediglich ein Drupal-Hook namens `hook_perm()` in der Moduldatei (z.B. `accommodation.module`) angelegt werden. Dieser Hook definiert einen Array mit allen vorhandenen Benutzerrechten des entsprechenden Modules.

Möchte man nun überprüfen, ob ein Benutzer das Recht hat, eine bestimmte Aktion durchzuführen, muss man lediglich die Drupal-Funktion `user_access()` mit dem Namen des Benutzerrechts aufrufen und es wird bestimmt, ob der Benutzer diese Aktion ausführen darf. Ist dies der Fall, wird TRUE zurückgegeben.

²<http://api.drupal.org/api/5>

³<http://api.drupal.org/api/group/hooks/5>

1.5. Datenbankzugriff

Für Zugriffe auf die Datenbank bietet die Drupal-API eigene Funktionen, welche eine komfortable Datenbankabfrage ermöglichen und gleichzeitig SQL-Injektionen verhindern. Die wichtigsten Funktionen zum Zugriff auf die Datenbank sind:

- `db_query()`

Diese Funktion wird benutzt, um eine Datenbankabfrage auszuführen. Dabei werden alle Argumente der Abfrage als gesonderte Parameter übergeben, um SQL-Injektionen zu verhindern. Ähnlich der `printf`-Funktion aus der Programmiersprache C müssen so die Platzhalter `%s` (für Zeichenketten), `%d` (für Ganze Zahlen), `%f` (für Gleitkommazahlen) oder `%b` (für Binärdaten) verwendet werden.

Beispiel: `$result = db_query("SELECT n.nid, n.created FROM {node} n WHERE n.type = 'blog' AND n.status = %d ORDER BY n.created DESC", $status);`

- `db_fetch_array()`

Hiermit wird die Antwort der Datenbankabfrage zeilenweise in einem assoziativen Array zurückgegeben.

Beispiel: `while ($blog = db_fetch_array($result)) { echo $blog['nid']." ".$blog['created']; }`

- `db_fetch_object()`

Hiermit wird die Antwort der Datenbankabfrage zeilenweise in einem Objekt zurückgegeben.

Beispiel: `while ($blog = db_fetch_object($result)) { echo $blog->nid." ".$blog->created; }`

Um auf die Datenbank zu zugreifen, muss nicht extra eine Datenbankverbindung hergestellt werden, da dies von Drupal automatisch durchgeführt wird.

1.6. Formulare und Formularverarbeitung

Drupal besitzt eine eigene Programmierschnittstelle (API) zur Darstellung und Verarbeitung von Formularen. Die einzelnen Eingabefelder des Formulars werden dabei in einem multidimensionalen Array gespeichert.

Beispiel 5.1. Formular in Drupal

```
<?php
function test_form() {
    $form['foo'] = array(
        '#type' => 'textfield',
        '#title' => t('bar'),
        '#default_value' => $object['foo'],
        '#size' => 60,
        '#maxlength' => 64,
        '#description' => t('baz'),
    );

    $form['submit'] = array(
        '#type' => 'submit',
        '#value' => t('Save')
    );

    return $form;
}

function test_page() {
    return drupal_get_form('test_form');
}
?>
```

Im Beispiel wird ein einfaches Formular mit einem Text-Eingabefeld und einem Submit-Button erstellt. Im Schlüssel "#type" des Formular-Arrays wird der Typ des Formularelementes bestimmt. Gültige Werte sind hier checkbox, date, fieldset, file, password, radio, select, textarea, textfield, hidden und submit. Dieser Array kann um beliebige Formularelemente erweitert werden und muss lediglich mit der Funktion `drupal_get_form()` ausgegeben werden, um dem Benutzer das Formular zu präsentieren.

Um ein solches Formular hinsichtlich der Benutzereingaben zu validieren, erzeugt man lediglich eine Funktion die den Namen des Formulars trägt plus die Endung `_validate`.

Beispiel 5.2. Validierung eines Formulars in Drupal

```
function test_form_validate($form_id, $form_values) {
    if ($form_values['foo'] == '') {
        form_set_error('', t('You must fill in something.'));
    }
}
```

Zur Speicherung oder Verarbeitung des abgeschickten Formulars, muss eine Funktion mit dem Formularnamen plus der Endung `_submit` erzeugt werden.

Beispiel 5.3. Verarbeitung eines abgeschickten Formulars in Drupal

```
function test_form_submit($form_id, $form_values) {
    db_query("INSERT INTO {table} (name) VALUES ('%s')", $form_values['foo']);
    drupal_set_message(t('Your form has been saved.'));
}
```

Diese Funktionen reichen aus, um in Drupal Formulare zu erzeugen und zu verarbeiten.

1.7. Drupal und objektorientierte Programmierung

Eine Antwort auf die Frage, ob Drupal wirklich objektorientiert ist, liefert der Artikel "Object-Oriented Programming in Drupal" [OOP07]. Dort heißt es, ab Version 4.7 besitzt der Quellcode von Drupal keine einzige PHP-Klasse mehr.

Die Gründe dafür liegen im Aufbau von Drupal selbst, denn Drupal ist durchgängig in Module aufgeteilt. Jedes dieser Module verwendet seine eigenen Funktionen. Nun ist es so, dass Drupal bei einer Anfrage versucht, so wenig PHP-Code wie möglich auszuführen, das heißt, nur den Code der wirklich benötigten Module auszuführen. Bei einem objektorientierten Ansatz müssten jedoch alle Module auf oberster Ebene der Klassenhierarchie eingebunden werden, was zu erheblich mehr auszuführenden Code führen und damit auch die Anfragen stark verlangsamen würde. Außerdem besitzt Drupal eigene objektorientierte Konzepte, welche sich nur umständlich in herkömmliche PHP-Klassen einbauen lassen (Vgl. [OOP07]).

Gerade diese Konzepte machen Drupal trotzdem zu einer quasi-objektorientierten Anwendung. So erfüllen die Drupal-Hooks beispielsweise das Konzept der Abstraktion. In diesen Hooks werden bestimmte Funktionsweisen der einzelnen Module definiert, welche von jedem anderen Modul verwendet werden können. Auch finden sich Objekte in Drupal. So können zum Beispiel Knoten in Drupal als Objekte gesehen werden, die bei Bedarf auch um bestimmte Inhalte erweitert werden können.

Bei Einhaltung der Drupal Coding-Standards finden sich auch Konzepte der Datenkapselung wieder. Funktionen, welche nicht von anderen Modulen aufgerufen werden sollen, beginnen mit einem Unterstrich.

2. Modul-Entwicklung im Speziellen

2.1. Googlemap-Modul

Das Googlemap-Modul erlaubt das Anzeigen von Google Maps auf der von Drupal verwalteten Internetpräsenz. Dazu wurde ein neuer Knotentyp namens "googlemap" implementiert, der zur Anzeige der Google Map verwendet wird. Im Drupal Modul-Verzeichnis⁴ findet sich bereits ein fertiges Modul zur Darstellung von Google Maps, jedoch ist dieses umständlich zu bedienen und bietet nur wenig Möglichkeiten. So können mit diesem Modul keine Markierungen zu bestehenden Inhaltsseiten verlinkt werden und auch keine Rad- und Wanderwege angezeigt werden. Somit wurde ein komplett neues Modul entworfen, das leichter zu bedienen ist und mehr Möglichkeiten bietet.

2.1.1. Anforderungen

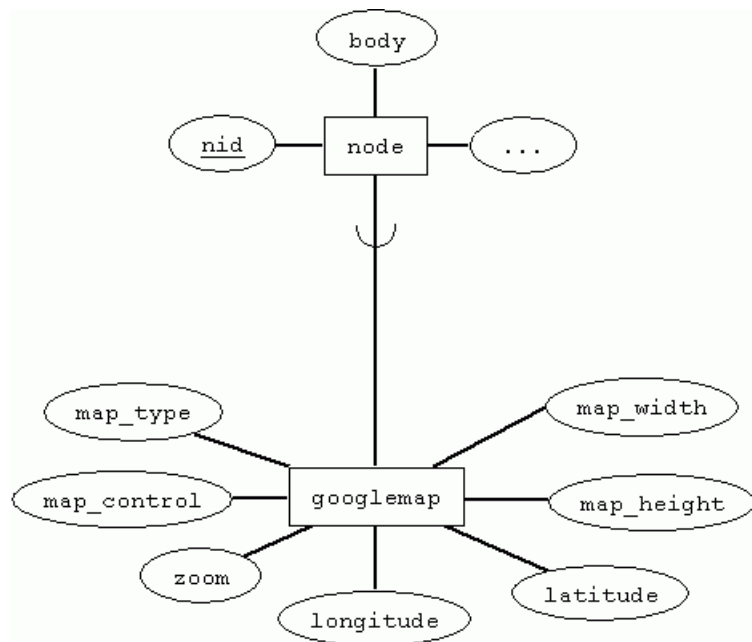
Das Modul soll den Benutzer dazu befähigen, auf einfache Weise eine Internetpräsenz mit einer oder mehrerer Google Maps zu erstellen. Dabei soll der Kartentyp, die Kartengröße und der Ausschnitt der Karte frei wählbar sowie bei Bedarf ein erläuternder Text anzeigbar sein. Außerdem soll es möglich sein, Markierungen und Wege auf dieser Karte darzustellen und diese mit den Web 2.0-Diensten YouTube und Flickr zu vernetzen. Wenn der Benutzer der Internetpräsenz eine Markierung auf einer Google Map anklickt, soll ein Informationsfenster erste Informationen über den verlinkten Inhalt anzeigen sowie bei Bedarf die Fotos von Flickr und die Videos von YouTube einblenden.

Da die Anforderungen an das Modul sehr umfangreich sind, wurde es in drei kleinere Module aufgeteilt, die alle bestimmte Funktionalitäten realisieren. Das so entstandene Googlemap-Modul dient im Wesentlichen nur zur Erstellung und Anzeige einer oder mehrerer Google Maps auf der Internetpräsenz. Dieses Modul kann auch alleine verwendet werden, jedoch ist es dann nicht möglich, Inhalte auf den Google Maps anzuzeigen. Dazu wurden die beiden Nebenmodule "Mapmarker" und "Mappath" entwickelt. Diese Module sind ohne das Googlemap-Modul nicht lauffähig und stellen eine Erweiterung dessen dar. Das Mapmarker-Modul wird dazu verwendet, um mit Drupal erstellte Inhalte auf einer Google Map zu verlinken. Das Mappath-Modul hingegen dient dazu, Wege bzw. Pfade auf der Google Map anzuzeigen. Beide Module werden in separaten Abschnitten erläutert.

⁴<http://drupal.org/project/Modules>

2.1.2. Datenbankentwurf

Abbildung 5.2. ER-Modell des Googlemap-Moduls



Die Super-Entitätenmenge "node" wurde vom Drupal-Kern übernommen und ist nicht Teil des modulspezifischen Datenbankentwurfs. Jeder erstellte Knoten, also auch Knoten vom Typ "googlemap", wird durch eine Entität in "node" und "googlemap" repräsentiert. Die Entitätenmenge "googlemap" stellt eine Sub-Entitätenmenge von "node" dar und speichert die zusätzlichen Attribute für die Google Map. Zu jeder erstellten Google Map wird so der Kartentyp, die Kontrollelemente, die Zoomstufe, die Longitude und Latitude der Kartenmitte sowie die Kartengröße gespeichert. Gleichzeitig wird auch der Seitentext, der Vorschautext, das Erstellungsdatum und einige andere Attribute in "node" gespeichert. Mit diesen beiden Entitätenmenge können alle relevanten Informationen redundanzfrei gespeichert werden.

2.1.3. Implementierung

Das Grundgerüst des Googlemap-Modules wurde vom Beispielmodul im Anhang dieser Arbeit übernommen. Dabei mussten die unterschiedlichen Nutzerrechte in dem Hook `googlemap_perm()` vergeben und die Menüeinträge in `googlemap_menu()` angepasst werden. Der wesentliche Schwerpunkt bei der Entwicklung dieses Modules lag aber in der Darstellung der Google Map.

2.1.3.1. Zugriff auf die Google Map-API

Der wichtigste Aspekt bei der Umsetzung des Googlemap-Moduls ist der Zugriff auf das Kartenmaterial von Google und die Darstellung der Karte. Google bietet dazu eine Programmierschnittstelle (*API*) an, mit der die Karten auf einfachste Weise in eine bestehende Internetseite integriert werden können. Zunächst muss jedoch unter ⁵ ein sogenannter "Google Maps API Key" beantragt werden. Danach können die Karten mit wenigen Zeilen JavaScript-Code in eine bestehende Internetseite integriert werden. Dazu muss im HTML-Kopf der Internetseite eine JavaScript-Klasse von Google eingefügt werden, dies geschieht am besten mit dem folgenden HTML-Tag:

Beispiel 5.4. Benötigte JavaScript-Klasse zur Anzeige einer Google Map

```
<script src="http://maps.google.com/maps?file=api&v=2.x&key=APIKEY"
type="text/javascript">
</script>
```

⁵<http://www.google.com/apis/maps/signup.html>

In diesem Beispiel muss "APIKEY" noch durch den tatsächlichen API Key ersetzt werden. Anschließend kann man einfach ein JavaScript-Objekt des Typs "GMap2" erstellen und die entsprechenden Eigenschaften zuweisen. Bei der Erstellung des Objekts registriert man automatisch den Container, in der Regel ein div-layer, der zur Anzeige der Google Map verwendet werden soll.

Beispiel 5.5. Festlegen der Parameter einer Google Map

```
<script type="text/javascript">
  //

  window.onload = function() {
    // first check if browser is compatible
    if (GBrowserIsCompatible()) {
      // create a new google map object and connect it to the layer with the id "map"
      var map = new GMap2(document.getElementById("map"));
      // set the default latitude, longitude and the zoom level
      map.setCenter(new GLatLng(53.0370844528,14.2884063721), 11);
      // set the default map type to hybrid
      map.setMapType(G_HYBRID_MAP);
      // add the buttons for zooming and navigating on the map
      map.addControl(new GLargeMapControl());
      // adds the buttons where the user can change the map type
      map.addControl(new GMapTypeControl());
    }
  }
  window.onunload = function() {
    GUnload(); // this function is provided by google
  }

  //]]&gt;
&lt;/script&gt;
&lt;!-- other HTML-Code --&gt;
&lt;div id="map" style="width:500px;height:300px;"&gt;&lt;/div&gt;
&lt;!-- other HTML-Code --&gt;</pre></div><div data-bbox="115 512 886 598" data-label="Text"><p>Mehr JavaScript-Code ist nicht erforderlich, um eine Google Map auf einer Internetseite anzuzeigen. Aber wie kann man einen solchen Code mit Drupal ausgeben? Nun, in der Moduldatei <code>googlemap.module</code> befindet sich ein Drupal-Hook namens <code>googlemap_view()</code>. Dieser Hook beeinflusst die Ausgabe der Knoten vom Typ Googlemap und ist damit ideal um den extra JavaScript-Code auszugeben. Dazu muss lediglich die Drupal-Funktion <code>drupal_set_html_head()</code> mit dem entsprechenden JavaScript-Code als Parameter ausgegeben werden. Die jeweiligen Parameter wie Longitude, Latitude und Kartengröße werden vorher aus der Datenbank ausgelesen.</p></div><div data-bbox="115 608 387 624" data-label="Section-Header"><h4>2.1.3.2. Konfigurationsoberfläche</h4></div><div data-bbox="115 635 886 722" data-label="Text"><p>Es wurde eine Konfigurationsoberfläche entwickelt, in dem sich der Google Map API Key sowie einige andere Optionen zur Google Map einstellen lassen. Diese Oberfläche wurde durch eine Funktion namens <code>googlemap_admin()</code> realisiert. Diese Funktion ist kein Drupal-Hook und muss daher auch extra in dem Drupal-Hook <code>googlemap_menu()</code> registriert werden. Zur Darstellung der Formularelemente auf der Konfigurationsoberfläche muss lediglich ein Array mit der in der Drupal Form API<sup>6</sup> festgelegten Struktur erzeugt werden. Die Speicherung der Eingaben wird von Drupal automatisiert und muss nicht extra implementiert werden.</p></div><div data-bbox="115 896 520 912" data-label="Footnote"><hr/><p><sup>6</sup><a href="http://api.drupal.org/api/file/developer/topics/forms_api_reference.html/5">http://api.drupal.org/api/file/developer/topics/forms_api_reference.html/5</a></p></div><div data-bbox="484 937 510 953" data-label="Page-Footer"><hr/><p>34</p></div>
```


Beispiel 5.6. Implementierung von `googlemap_menu()` und `googlemap_admin()`

```
<?php

/**
 * Implementation of hook_menu
 * This hook enables modules to register paths, which determines whose requests are to be
 * handled. Depending on the type of registration requested by each path, a link is placed
 * in the the navigation * block and/or an item appears in the menu administration page.
 * @param may_cache A boolean indicating whether cacheable menu items should be returned.
 * @return array An array of menu items
 */
function googlemap_menu($may_cache) {

    $items = array();

    $items[] = array(
        'path' => 'admin/settings/googlemap', // we register this path
        'title' => t('Google Map Optionen'),
        'description' => t('Einstellungen der Google Map.'),
        'callback' => 'drupal_get_form', // the path above shows a form
        'callback arguments' => array('googlemap_admin'), // name of the function that shows
the form
        'access' => user_access('administer googlemap pages'), // allow access only to
certain users
        'type' => MENU_NORMAL_ITEM
    );

    // some other items here ...

    return $items;
}

/**
 * This function is not a hook. This function is used to configure the Google Map
 * module and store the API key.
 */
function googlemap_admin() {

    $form['googlemap_api_key'] = array(
        '#type' => 'textfield',
        '#title' => t('Google Map API key'),
        '#default_value' => variable_get('googlemap_api_key',''), // default api key is empty
        '#size' => 105,
        '#maxlength' => 128,
        '#description' => t("Bitte den Schlüssel eingeben.")
    );
    $form['googlemap_map_width'] = array(
        '#type' => 'textfield',
        '#title' => t('Standard-Kartenbreite (in pixel)'),
        '#default_value' => variable_get('googlemap_map_width','500'),
        '#size' => 3,
        '#maxlength' => 3,
        '#description' => t("Bitte die Breite eingeben, die die Google Map Standardmäßig hat.")
    );
    $form['googlemap_map_height'] = array(
        '#type' => 'textfield',
        '#title' => t('Standard-Kartenhöhe (in pixel)'),
        '#default_value' => variable_get('googlemap_map_height','300'),
        '#size' => 3,
        '#maxlength' => 3,
        '#description' => t("Bitte die Höhe eingeben, die die Google Map Standardmäßig hat.")
    );
    $form['googlemap_default_longitude'] = array(
        '#type' => 'textfield',
        '#title' => t('Longitude'),
        '#default_value' => variable_get('googlemap_default_longitude','14.288406372070312'),
        '#size' => 30,

```

```

    '#maxlength' => 30,
    '#description' => t("Bitte die Latitude eingeben, die neue Google Maps beim erstellen
haben.")
  );
  $form['googlemap_default_latitude'] = array(
    '#type' => 'textfield',
    '#title' => t('Latitude'),
    '#default_value' => variable_get('googlemap_default_latitude', '53.03708445284038'),
    '#size' => 30,
    '#maxlength' => 30,
    '#description' => t("Bitte die Latitude eingeben, die neue Google Maps beim erstellen
haben.")
  );
  $form['googlemap_default_zoom'] = array(
    '#type' => 'select',
    '#title' => t('Standard Zoom-Stufe'),
    '#default_value' => variable_get('googlemap_default_zoom', '11'),
    '#options' => drupal_map_assoc(range(0, 19)),
    '#description' => t("Bitte die Zoom-Stufe eingeben, die bei der Erstellung neuer Google
Maps verwendet werden soll.")
  );
return system_settings_form($form);
}
?>

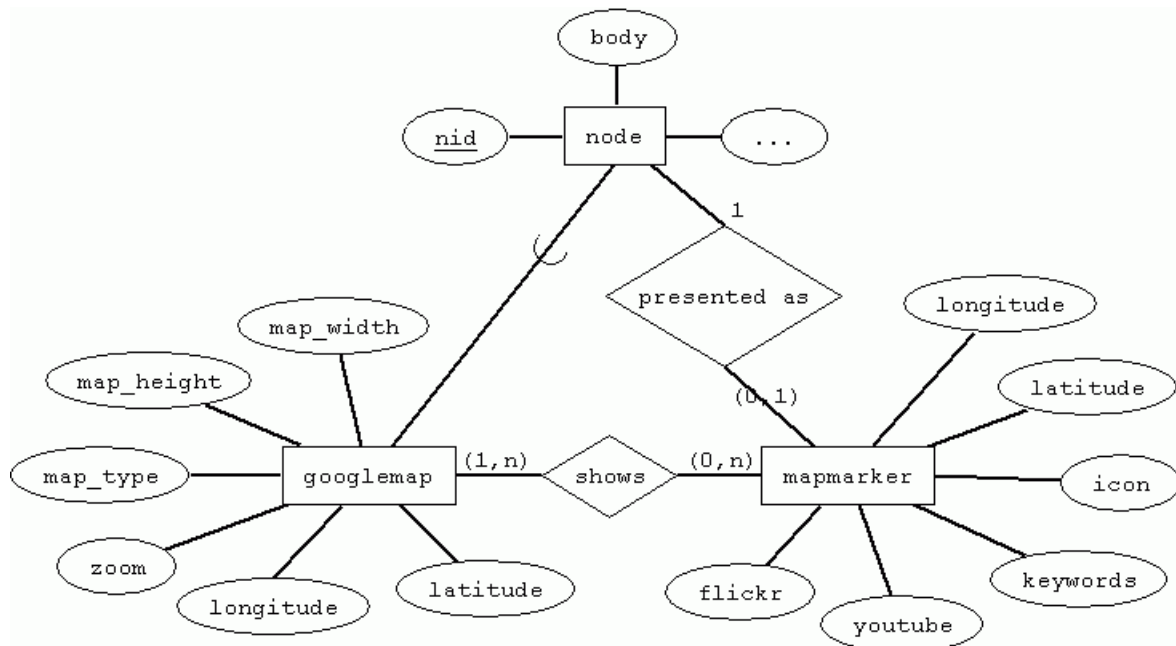
```

2.2. Mapmarker-Modul

Dieses Modul ist eine Erweiterung zum Googlemap-Modul und ist auch nur in Verbindung mit diesem lauffähig. Im Wesentlichen ermöglicht es das Erstellen von Markierungen auf beliebigen Google Maps. Dabei stehen eine Vielzahl von Symbolen⁷ zur Verfügung, die auf der Google Map angezeigt werden.

2.2.1. Datenbankentwurf

Abbildung 5.3. ER-Modell des Mapmarker-Moduls



⁷Die Symbole wurden mit einem kostenlosen Werkzeug unter http://www.gmaplive.com/marker_maker.php erstellt und mit einem Bildbearbeitungsprogramm nachbearbeitet.

Wie in dem Entwurf zu sehen ist, benötigt das Mapmarker-Modul die Entitätenmenge "googlemap" aus dem Googlemap-Modul und die Entitätenmenge "node" aus dem Drupal-Kern. Jede Markierung auf einer Karte ist durch eine Entität in "mapmarker" repräsentiert. Zu jeder Entität werden dabei die genaue Position der Markierung, das Symbol (Icon) und die Schlüsselwörter gespeichert. Die Attribute "youtube" und "flickr" sagen aus, ob in dem jeweiligen Informationsfenster auch YouTube-Videos oder Flickr-Bilder angezeigt werden sollen. Die Beschreibung, welche im Informationsfenster der Markierung erscheint, ist eine verkürzte Version des Attributes "body" der Entitätenmenge "node".

Eine Google Map zeigt 0 bis unendlich viele Markierungen an und eine Markierung kann auf einer bis unendlich vielen Google Maps angezeigt werden. Zu jedem Knoten in Drupal gibt es entweder eine Darstellung als Markierung oder keine und jede Markierung gehört immer zu genau einem Knoten. Damit ist es möglich, beliebige Inhaltsobjekte von Drupal - sei es ein Blogeintrag, ein Foto oder eine normale Internetseite - auf einer oder mehrerer Google Maps darzustellen.

2.2.2. Erzeugen von Markierungen und Darstellung auf der Google Map

Das Mapmarker-Modul ist das einzige Modul dieser Bachelorarbeit, welches keinen eigenen Knotentyp implementiert. Das heißt, man kann mit Drupal keinen Inhalt vom Typ "Mapmarker" erstellen. Wie kann man dann solche Markierungen erstellen und auf der Karte anzeigen? Nun, jede Markierung auf der Karte steht für genau einen Knoten. Wird ein solcher Knoten, zum Beispiel eine Naturbeobachtung oder eine Unterkunft, mit Drupal erstellt oder bearbeitet, kann auch gleichzeitig eine passende Markierung auf einer Google Map erstellt werden. Dazu wurde das Formular zur Erstellung bzw. Bearbeitung von Knoten erweitert, um gleichzeitig auch eine Verlinkung mit der Google Map zu ermöglichen.

Dieses Formular wird immer angezeigt, wenn man einen neuen Knoten erzeugt oder einen bestehenden Knoten bearbeitet. Man bekommt somit die Möglichkeit, diesen Knoten auf einer oder mehrerer Google Maps zu veröffentlichen. Der wesentliche programmiertechnische Teil zum Erzeugen eines solchen Formulars liegt in dem Hook `mapmarker_form_alter()` in der Datei `mapmarker.module`. Dort wird ein Array namens `$form` verwendet, um die zusätzlichen Formularelemente für die Kartenauswahl, die Schlüsselwörter usw. hinzuzufügen.

Das Anzeigen von Markierungen wird über verschiedene JavaScript-Anweisungen gesteuert, die in der Funktion `get_marker_code()` in der Datei `mapmarker.module` erzeugt werden. Diese Anweisungen werden bei Anzeige einer Google Map im HTML-Code ausgegeben und erzeugen so die einzelnen Markierungen.

Beispiel 5.7. Funktion `get_marker_code()`

```
<?php
/*
 * This function is called by the googlemap module. It is not a Drupal Hook.
 * It created the JavaScript code to show the markers on the Google Map.
 * @param googlemap_id The node_id of the google map that shows the marker
 */
function get_marker_code($googlemap_id) {
  // select all markers of the particular google map (but only the published ones)
  $queryResult = db_query("SELECT mapmarker.mid, icon, longitude, latitude, title,
youtube, flickr FROM node, mapmarker, googlemap_marker WHERE
googlemap_marker.googlemap_id=%d AND googlemap_marker.mid = mapmarker.mid AND
mapmarker.nid = node.nid AND node.status=1", $googlemap_id);
  $marker_code = '// found ' . db_num_rows($queryResult) . ' marker on this map'. "\n";
  while ($marker = db_fetch_object($queryResult)) {
    $marker_code .= "map.addOverlay(createMarker(new GLatLng(" . $marker->latitude . ", " .
$marker->longitude . " ), \" . addslashes(htmlspecialchars($marker->title, ENT_NOQUOTES)) . "\", " .
$marker->mid . "\", \" . $marker->icon . "\", " . $marker->flickr . "\", " . $marker->youtube . "));\n";
  }
  return $marker_code; // returns a few lines of JavaScript-Code
}
?>
```

Viel interessanter als die Darstellung der Symbole auf der Google Map ist jedoch die Verknüpfung mit der YouTube- und Flickr-API.

2.2.3. Allgemeiner Zugriff auf die APIs

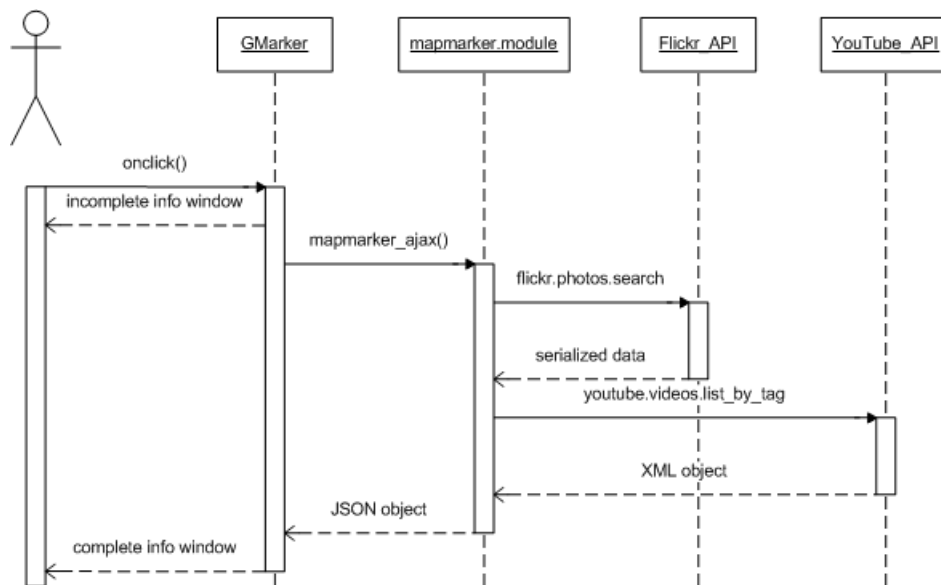
Das Besondere am Mapmarker-Modul ist die Tatsache, dass Fotos und Videos im Informationsfenster der Kartenmarkierung angezeigt werden können. Diese Fotos und Videos werden durch die *Application Programming Interfaces* (APIs), also die Programmierschnittstellen der jeweiligen Anbieter, gesucht und ohne Zwischenspeicherung direkt im Informationsfenster angezeigt. Der allgemeine Ablauf einer solchen Anfrage sieht wie folgt aus:

1. Der Benutzer klickt in der Google Map auf eine Markierung.
2. Ein Informationsfenster öffnet sich auf der Google Map. Es wird per JavaScript erkannt, ob die Markierung nur allgemeine Informationen enthält oder auch Fotos und Videos zeigen soll.
3. Es wird per JavaScript eine Ajax-Anfrage an das Mapmarker-Modul gestellt, um die Beschreibung der Markierung und die eventuellen Fotos oder Videos zu laden. Diese Ajax-Anfrage ruft eine spezielle URL des Mapmarker-Moduls auf, um die gewünschten Daten zu bekommen.
4. Ausgelöst durch die Ajax-Anfrage wird auf dem Web-Server der Vorschautext der entsprechenden Markierung aus der Drupal-Datenbank geladen. Außerdem wird entschieden, ob noch eine Ajax-Anfrage an Flickr oder YouTube gestellt werden muss. Ist dies der Fall, werden diese Daten durch einen einfachen REST-Aufruf geladen.
5. Auf dem Web-Server werden die zurückgegebenen Daten der Flickr und YouTube-API verarbeitet, in das JSON-Format konvertiert und ausgegeben. Diese Ausgabe stellt gleichzeitig die Antwort auf die ursprüngliche JavaScript-Anfrage aus Schritt 3 dar.
6. Die zurückgegebenen Daten werden per JavaScript entgegengenommen und die entsprechenden Inhalte im Informationsfenster angezeigt.
7. Es werden keine weiteren Daten geladen, bis der Benutzer auf eine weitere Markierung klickt.

Das folgende Sequenzdiagramm verdeutlicht diesen Ablauf. Wie dort ersichtlich ist, wird vom Computer des Benutzers nur eine Ajax-Anfrage gestartet, die ihrerseits weitere Anfragen durch den Web-Server auslöst. Der Web-Server ist also eine Zwischeninstanz, die die API-Anfragen an Flickr und YouTube stellt. Dieser Zwischenschritt ist insofern notwendig, um die Verarbeitung der von den APIs zurückgegebenen Daten zu erleichtern. Die von Flickr und YouTube angebotenen Antwortformate lassen sich nämlich nur sehr umständlich und ressourcenintensiv mit JavaScript verarbeiten.

Außerdem wird im Sequenzdiagramm ersichtlich, dass kurz nach dem Klick auf die Kartenmarkierung bereits ein unvollständig geladenes Fenster angezeigt wird. Dieses Fenster ist insofern von Bedeutung, da es dem Nutzer bereits eine Rückmeldung gibt, dass etwas geladen wird. Ohne diese Rückmeldung würde sich der Benutzer fragen, ob überhaupt etwas passiert ist. Sobald die restlichen Daten per Ajax geladen sind, wird das Fenster aktualisiert und mit den vollständigen Informationen gefüllt.

Abbildung 5.4. Sequenzdiagramm zum Ablauf der Ajax-Anfragen



2.2.4. Abfrage der Flickr-Fotos

Die Flickr-API bietet vielfältige Möglichkeiten, um mit Drupal auf den Bildbestand zuzugreifen. Dabei werden verschiedene Formate wie *REST*, *XML-RPC* oder *SOAP* zur Anfrage und Übertragung der Daten unterstützt. Für diese Arbeit wurde ein serielles PHP-Antwortformat ausgewählt, da es wenig Overhead hat und leicht zu verarbeiten ist.

Die Anfrage der Daten gestaltet sich außerordentlich einfach, denn es muss kein spezielles Anfrageobjekt geschaffen werden. Es muss lediglich eine spezifische URL von Flickr mit der PHP-Funktion `file_get_contents()` aufgerufen und das Rückgabeobjekt als Zeichenkette gespeichert werden. Eine solche Zeichenkette ist eine einfache serielle Darstellung eines Arrays, welche sich leicht mit der PHP-Funktion `unserialize()` in einen PHP-Array konvertieren lässt. Dieses Array enthält dann verschiedene Angaben zum Foto wie den Namen, die Größe oder die URL. Diese Angaben werden anschließend gefiltert und in ein JSON-Objekt konvertiert, damit sie auch einfach mit JavaScript ausgelesen werden können.

Hier der komplette Aufruf aus der Datei `mapmarker.module`, der für die serverseitige Anfrage der Flickr-Fotos zuständig ist.

Beispiel 5.8. Aufruf der Flickr-API in der Datei mapmarker.module

```
<?php

// some module content here

$flickr_params = array(
  'api_key' => variable_get('flickr_api_key',''), // we get the flickr key from drupal
  'method' => 'flickr.photos.search', // we want to call this method
  'tags' => $marker->keywords, // all the keywords separated by commas
  'tag_mode' => 'all', // we want an AND combination of the keywords
  'format' => 'php_serial', // the format of the answer object
  'page' => $page, // this allows us to view the photos one by one
  'per_page' => 1, // we only want to get one photo per request
);

$encoded_params = array();

// build the parameters for the API call and don't forget to encode them
foreach ($flickr_params as $key => $value) {
  $encoded_params[] = urlencode($key).'='.urlencode($value);
}

// build the complete URL of the API call
$url = "http://api.flickr.com/services/rest/?".implode('&', $encoded_params);

// do the api call and save the answer object
$response = file_get_contents($url);

// check if we got an answer
if ($response !== false) {

  // convert the string into an array
  $received_object = unserialize($response);

  // process response
  if ($received_object['stat'] == 'ok') {
    if ($received_object['photos']['total']) {
      $nextPhotoPage = $page+1;
      if ($nextPhotoPage > $received_object['photos']['total']) $nextPhotoPage = 1;
      $previousPhotoPage = $page-1;
      if ($previousPhotoPage < 1) $previousPhotoPage = $received_object['photos']['total'];

      foreach ($received_object['photos']['photo'] as $photo) {
        $return_object['flickr'] = "<strong>".$photo['title']."</strong> ".$page." / ".
        $received_object['photos']['total']. " <img src=\"".base_path()."\".drupal_get_path('module',
        'mapmarker')."\"/images/previous.gif\" style=\"cursor:pointer;\" title=\"Vorheriges Foto ("
        $previousPhotoPage.")\" onclick=\"swapPhoto(\".$marker_id.\",\".$previousPhotoPage.)\";\"
        width=\"11\" height=\"11\"> <img src=\"".base_path()."\".drupal_get_path('module',
        'mapmarker')."\"/images/next.gif\" style=\"cursor:pointer;\" title=\"Nächstes Foto ("
        $nextPhotoPage.")\" onclick=\"swapPhoto(\".$marker_id.\",\".$nextPhotoPage.)\";\" width=\"11\"
        height=\"11\"> <br />\n";
        $return_object['flickr'] .= "<img src=\"http://farm".$photo['farm']."
        static.flickr.com/\".$photo['server'].\"/\".$photo['id'].\"_\".$photo['secret'].\"_m.jpg\"
        height=\"175\" />";
      }
    }
    else {
      $return_object['flickr'] = "Keine Fotos gefunden.";
    }
  }
  else {
    $return_object['flickr'] = "Fehler beim Laden der Fotos.";
  }
}
else {
  $return_object['flickr'] = "Fehler beim Laden der Fotos.";
}
```

?>

2.2.5. Abfrage der YouTube-Videos

Ähnlich wie die Flickr-Fotos können auch die YouTube-Videos in dem Informationsfenster einer bestimmten Markierung auf einer Google Map angezeigt werden. Auch hier bietet YouTube eine ausgereifte API, die den Zugriff auf die Videos mit eigenen Programmen erlaubt. Im Unterschied zur Flickr werden allerdings nur die Antwortformate *REST* und *XML-RPC* unterstützt, was einen zusätzlichen Schritt bei der Verarbeitung der Antwort erfordert.

Die API wird, ähnlich der Flickr-API, einfach über eine URL mit den gewünschten Parametern aufgerufen. Ruft man zum Beispiel die URL

```
http://www.youtube.com/api2_rest?dev_id=_2CId_x5h0U&method=youtube.videos.list_by_tag&tag=Oder-  
tal&page=1&per_page=1
```

auf, bekommt man folgendes Objekt zurückgeliefert.

Beispiel 5.9. Rückgabeobjekt bei Zugriff auf die YouTube-API

```
<ut_response status="ok">  
  <video_list>  
    <total>2</total>  
    <video>  
      <author>netztaucherbrille</author>  
      <id>btcdl3u98tw</id>  
      <title>Kanufahrt Odertal</title>  
      <length_seconds>409</length_seconds>  
      <rating_avg>0.00</rating_avg>  
      <rating_count>0</rating_count>  
      <description>Geführte Kanufahrten im Unteren Odertal.</description>  
      <view_count>64</view_count>  
      <upload_time>1171560153</upload_time>  
      <comment_count>0</comment_count>  
      <tags>Kanu Odertal "Untere Oder" Nationalpark</tags>  
      <url>http://www.youtube.com/?v=btcdl3u98tw</url>  
      <thumbnail_url>http://img.youtube.com/vi/btcdl3u98tw/default.jpg</thumbnail_url>  
      <embed_status>ok</embed_status>  
    </video>  
  </video_list>  
</ut_response>
```

Dieses XML-Objekt muss nun mit PHP ausgelesen und weiterverarbeitet werden, dazu wurde die PHP-Klasse `SimpleXMLElement` verwendet. Erstellt man damit ein Objekt, kann man die XML-Inhalte wie einen Array ansprechen und weiterverarbeiten.

Da das Video letztendlich im Infofenster der Google Map angezeigt werden soll, müssen das Video und der Titel des Videos in ein geeignetes Format konvertiert werden, dass auch von JavaScript einfach ausgelesen werden kann. Die Wahl fiel auf JSON, da dieses Datenaustauschformat wenig Overhead besitzt und von JavaScript leicht ausgelesen werden kann. Eine Übertragung in XML wäre hier ungeeignet. Über die PHP-Funktion `json_encode()` werden schließlich die Daten ins JSON-Format gebracht und ausgegeben, so dass sie auf dem Computer des Benutzers mit JavaScript weiterverwendet werden können.

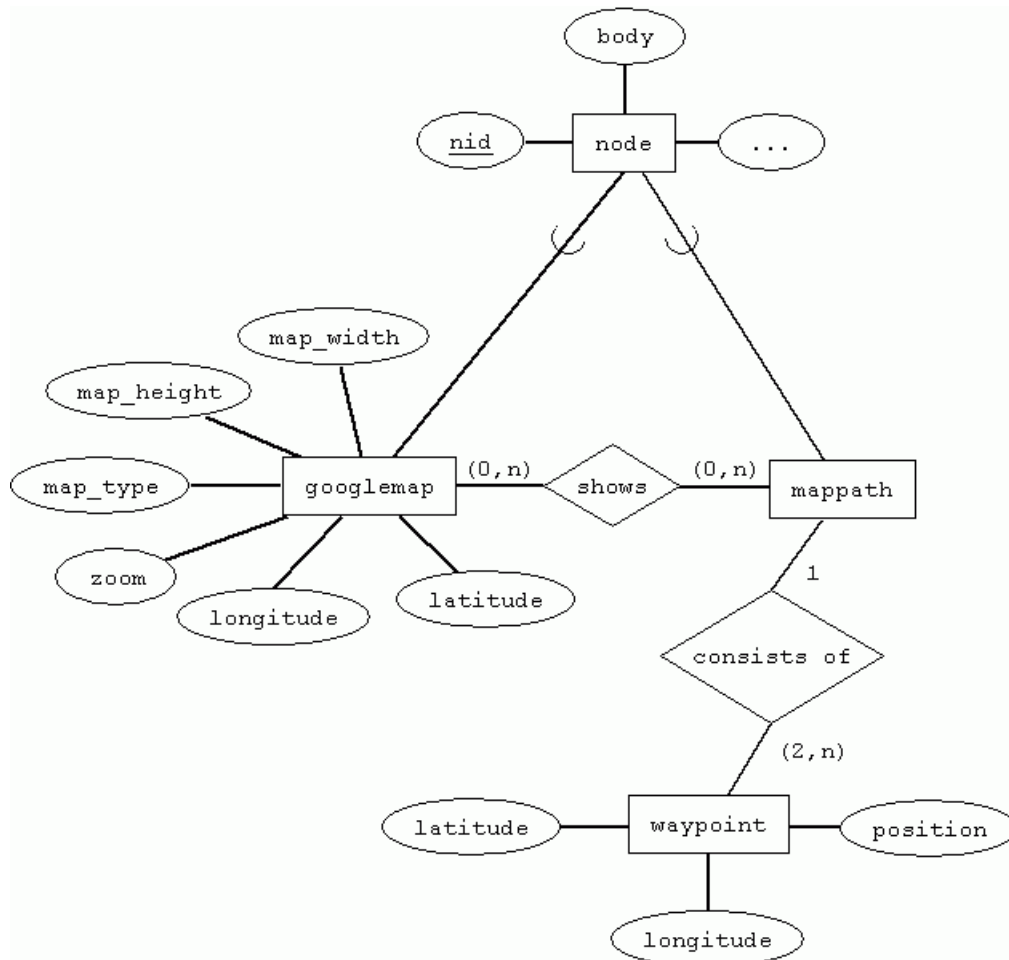
2.3. Mappath-Modul

Das Mappath-Modul ist, genauso wie das Mapmarker-Modul, eine Erweiterung zum Googlemap-Modul. Es erlaubt das Erstellen von Pfaden, wie z.B. Rad- und Wanderwege, auf einer oder mehreren Google Maps. Im Unterschied zum Mapmarker-Modul, bietet dieses Modul aber einen eigenen Inhaltstyp. Das heißt, ein Pfad kann erstellt werden, ohne ihn dabei zwangsläufig mit einer oder mehreren Google Maps zu verknüpfen. Der Grund für diese Herangehensweise liegt darin, dass so einzelne Rad- oder Wanderwege dargestellt werden können, ohne dafür extra neue Google Maps einzurichten.

Alle Pfade bestehen aus mindestens 2 Punkten, die auf direktem Wege miteinander verbunden sind. Sie haben einen eindeutigen Start- und Zielpunkt.

2.3.1. Datenbankentwurf

Abbildung 5.5. ER-Modell des Mappath-Moduls



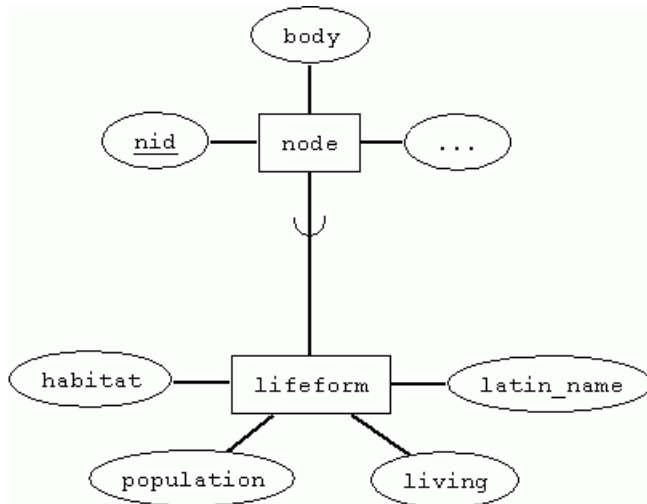
Die beiden Entitätenmengen "node" und "googlemap" sind bereits durch den Drupal-Kern bzw. das Googlemap-Modul gegeben und werden hier erweitert. So stellt jede Entität vom Typ "mappath" auch gleichzeitig eine Entität vom Typ "node" dar. In anderen Worten bedeutet dies, jeder Kartenpfad stellt auch gleichzeitig einen eigenen Drupal-Knoten dar. Jeder Kartenpfad besteht dabei aus mindestens zwei Punkten, welche durch die Entitätenmenge "waypoint" ausgedrückt werden.

2.4. Lifeform-Modul

Das Lifeform-Modul befähigt den Benutzer dazu, eine Datenbank mit Tieren und Pflanzen anzulegen und diese auf der Internetpräsenz anzuzeigen. Dabei können die Merkmale, der Lebensraum, die Lebensweise und der Bestand der jeweiligen Lebensform näher erläutert werden. Außerdem soll es möglich sein, ein Foto hinzuzufügen und dieses beim Lesen der Textbeschreibungen anzuzeigen. Da in einem Nationalpark eine hohe Artenvielfalt vorherrscht, sollen Tiere und Pflanzen in selbst gewählte Kategorien einzuordnen sein.

2.4.1. Datenbankentwurf

Abbildung 5.6. ER-Modell des Liform-Moduls



Zur Speicherung der Informationen über das Tier bzw. die Pflanze wurde eine Entitätenmenge "liform" entworfen, die eine Spezialisierung der Entitätenmenge "node" darstellt. Dort werden die spezifischen Informationen wie der Lebensraum, die Population, die Lebensweise und der lateinische Name gespeichert. Die allgemeinen Merkmale wie Größe und Erscheinungsbild des Tieres bzw. der Pflanze werden im Attribut "body" der Entitätenmenge "node" gespeichert. Jede Entität von "liform" ist auch eine Entität von "node".

2.4.2. Umsetzung

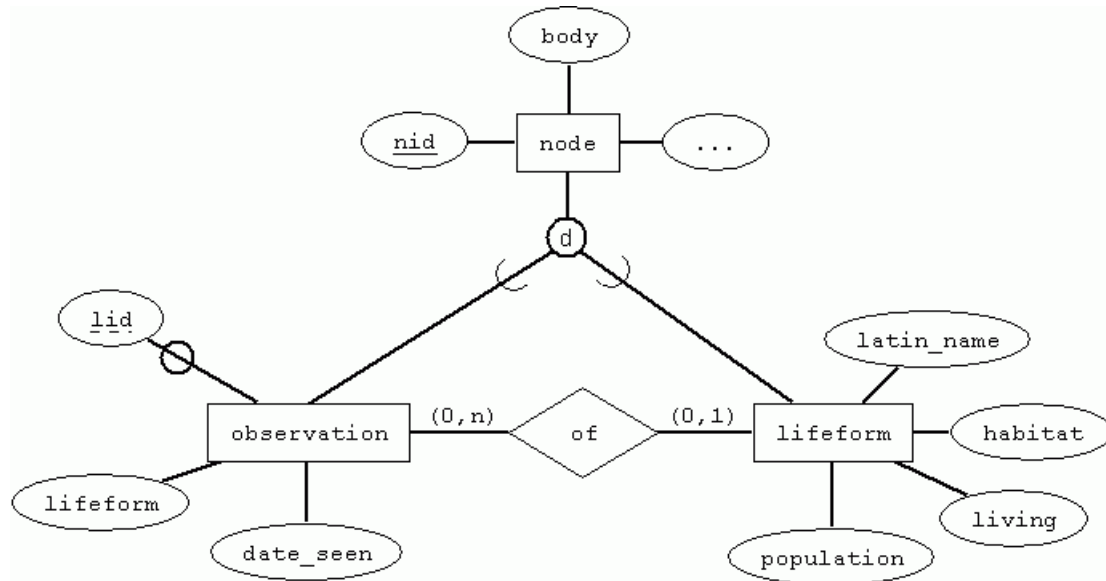
Die Kategorisierung der Inhalte geschieht über das Taxonomy-Modul. Der Administrator muss also nach der Installation des Modules die wichtigen Gliederungspunkte anlegen und hierarchisch vernetzen.

2.5. Observation-Modul

Dieses Modul stellt eine Erweiterung des Liform-Moduls dar und ist auch nur in Verbindung mit diesem Modul lauffähig. Es erlaubt registrierten Besuchern der Internetpräsenz, beobachtete Tiere oder Pflanzen zu melden und bedient damit eines der Kernkonzepte des Web 2.0. Es können auch Bilder zu den Beobachtungen hochgeladen werden, die dann später auch auf der Google Map angezeigt werden.

2.5.1. Datenbankentwurf

Abbildung 5.7. ER-Modell des Observation-Moduls



Wie im ER-Modell zu erkennen ist, stellt "node" eine Super-Entitätenmenge von "observation" und "lifeform" dar. Beide Sub-Entitätenmenge sind disjunkt, das heißt, eine Entität "node" ist gleichzeitig eine Entität von "observation" oder von "lifeform", nicht aber von beiden gleichzeitig.

Der Attribut "lid" der Entitätenmenge "observation" ist ein optionales Attribut, das nur gegeben ist, wenn sich die Naturbeobachtung auf eine bereits erstellte Entität der Entitätenmenge "lifeform" bezieht. Publiziert also ein Benutzer in Drupal eine Naturbeobachtung und gibt den Namen einer bereits existierenden Pflanze bzw. eines bereits existierenden Tieres ein, so wird das Attribut "lid" mit dem Schlüsselattribut des Tieres bzw. der Pflanze gesetzt.

2.5.2. Verknüpfung mit dem Lifeform-Modul

Beim Erstellen einer Naturbeobachtung wird automatisch nach einem passenden Tier bzw. einer passenden Pflanze im Lifeform-Modul gesucht. Schreibt man also den Namen des beobachteten Tieres bzw. der beobachteten Pflanze in das Eingabefeld, wird automatisch eine Ajax-Anfrage durchgeführt, die nach einem übereinstimmenden Lebewesen in der Naturdatenbank sucht. Wird ein Lebewesen gefunden, wird es dem Nutzer als Vorschlag angezeigt und er kann die Eingabe des Lebewesens abkürzen.

Um diese Funktionalität zu implementieren, wurde das Eingabefeld mit dem Attribut "autocomplete" versehen. Dadurch stellt Drupal automatisch eine fertige Ajax-Lösung zur Verfügung, in der nur noch die Zieladresse definiert werden muss. Wird nun ein Buchstabe ins Eingabefeld eingegeben, wird die Zieladresse per Ajax aufgerufen und die bereits eingegebenen Buchstaben als Parameter übergeben.

Beispiel 5.10. Formularfeld mit automatischer Vervollständigung

```
<?php
// the form array describes what input fields are needed when you create a new observation
$form['lifeform'] = array(
  '#type' => 'textfield',
  '#title' => t('Name des beobachteten Tiers oder der beobachteten Pflanze'),
  '#default_value' => $node->lifeform,
  '#description' => t('Bitte den Namen des beobachteten Lebewesens eingeben.'),
  '#size' => 60,
  '#maxlength' => 128,
  '#required' => TRUE,
  '#weight' => 1,
  '#autocomplete_path' => 'observation/autocomplete', // assign callback function for ajax
request
);
// some more form elements come here
?>
```

Jedes Mal, wenn ein Buchstabe im Eingabefeld eingegeben wurde, wird eine Ajax-Anfrage durchgeführt. Dabei wird eine spezielle PHP-Funktion in der Datei `observation.module` aufgerufen, die Vorschläge für bereits erstellte Tiere und Pflanzen zurückliefert.

Beispiel 5.11. Callback-Funktion zur automatischen Vervollständigung von Formularfeldern

```
<?php
// this function is used to retrieve a pipe delimited string of autocomplete
// suggestions for existing lifeform
function observation_autocomplete($string) {

  $string = trim($string); // we ignore whitespace at the beginning or end of the string

  // we try to find a lifeform that matches the users input
  $result = db_query_range("SELECT title FROM {node} WHERE (type='plant' OR type='animal')
AND LOWER(title) LIKE LOWER('%s%'", $string, 0, 10);
  $matches = array();
  while ($observation = db_fetch_object($result)) {
    // we put all the matches into an array
    $matches[$observation->title] = check_plain($observation->title);
  }
  print drupal_to_js($matches); // and output it as java script
  exit();
}
?>
```

Die über die `print`-Funktion zurückgegebenen Werte werden anschließend per JavaScript verarbeitet und im Eingabefeld angezeigt.

2.6. Accommodation-Modul

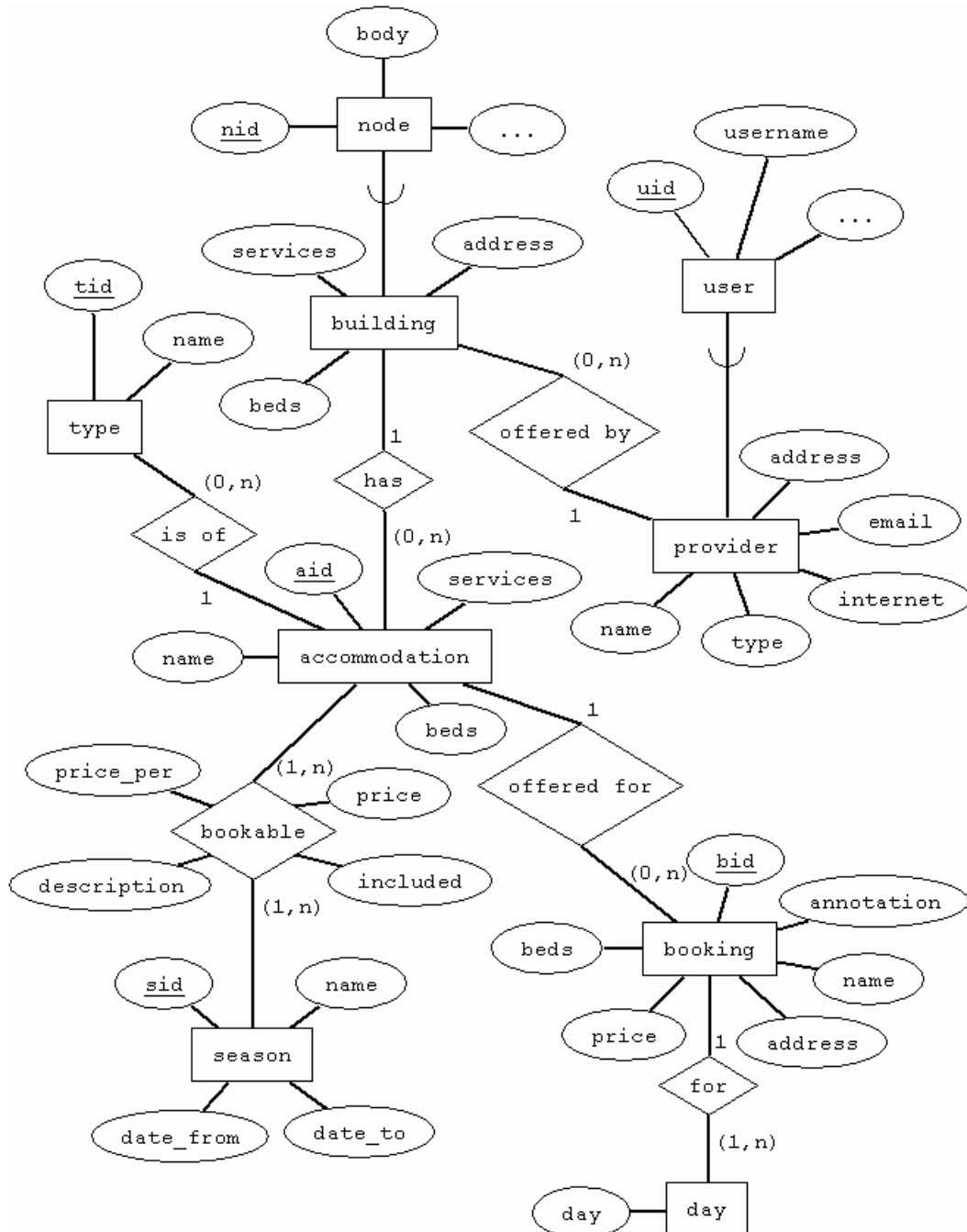
Mit dem Accommodation-Modul können Benutzer Unterkünfte jeglicher Art auf der Internetpräsenz anbieten und auf einer Google Map vernetzen. Dies soll für große Hotelanbieter genauso gut funktionieren wie für Anbieter privater Unterkünfte. Jeder Unterkunftsanbieter soll selbst in der Lage sein, seine Unterkunft mit Drupal zu erstellen und zu verwalten. Dadurch muss die Benutzeroberfläche sehr aufgeräumt und verständlich sein, darf aber auch keine wichtigen Details verbergen.

Jeder Hotelanbieter soll eine beliebige Anzahl von Unterkunftsgebäuden verwalten können, welche alle auf einer Google Map dargestellt werden können. Zu jedem Unterkunftsgebäude kann der Anbieter seine Zimmertypen und -preise sowie die Serviceeinrichtungen selbst festlegen. Die Preise sind abhängig von der Saison, es muss also möglich sein, unterschiedliche Saisonpreise für eine Unterkunft festzulegen.

Die erstellten Unterkünfte sollen auch direkt über die Internetpräsenz buchbar sein, wobei jeweils nur die freien Zimmer zur Buchung angeboten werden dürfen. Es soll für den Unterkunftsanbieter eine Buchungsübersicht geben, in der alle freien und gebuchten Zimmer des Unterkunftsgebäudes zu sehen sind. Für die Buchung selbst ist kein extra Drupal-Account erforderlich, jedoch müssen persönliche Angaben wie Name, Adresse und Telefonnummer eingegeben werden.

2.6.1. Datenbankentwurf

Abbildung 5.8. ER-Modell des Accommodation-Moduls



Für die Vielzahl an Informationen wurden unterschiedliche Entitätenmengen vorgesehen. Die logisch oberste Entitätenmenge ist "building". Diese Entitätenmenge stellt eine Spezialisierung von "node" dar, das heißt, jedes Unterkunftsgebäude, das mit Drupal erstellt wird, erzeugt eine Entität in "building" und eine in "node". Zu jedem

Unterkunftsgebäude wird die Anschrift, die Gesamtzahl an Betten, die Serviceeinrichtungen und ein eventuelles Bild gespeichert.

Jeder Anbieter von Unterkünften muss ein Benutzerkonto für Drupal besitzen. Die von Drupal geforderten Angaben zur Erstellung eines Benutzerkontos sind aber äußerst spärlich, so dass noch eine Sub-Entitätenmenge "provider" geschaffen wurde. Dort werden die zusätzlichen Kontaktdaten wie Anschrift, E-Mail, Internetadresse oder Telefonnummer der Unterkunftsanbieter gespeichert. Auch wird hier gespeichert, ob es sich um einen gewerblichen oder privaten Unterkunftsanbieter handelt. Jeder Anbieter kann beliebig viele Unterkunftsgebäude erstellen, muss aber seine Kontaktdaten nur einmal anlegen.

Die nächst kleinere Einheit bei einem Unterkunftsgebäude sind die einzelnen Zimmer, hier als "accommodation" bezeichnet. Jedes Zimmer hat einen Namen, eine bestimmte Anzahl an Betten und bestimmte Serviceeinrichtungen. Jedes Zimmer muss in eine bestimmte Kategorie eingeordnet werden wie Einzelzimmer, Doppelzimmer oder Mehrbettzimmer. Zur Speicherung der unterschiedlichen Zimmertypen wurde die Entitätenmenge "type" entworfen.

Jedes freie Zimmer kann auf der Internetpräsenz gebucht werden. Zur Speicherung der Buchungsdaten dient die Entitätenmenge "booking". Hier wird die Anschrift des Buchenden, der Preis, die Anzahl an gebuchten Betten sowie sonstige Anmerkungen gespeichert. Der Zeitraum der Buchung wird in einer extra Entitätenmenge namens "day" gespeichert. Jede Zimmerbuchung für x Tage erzeugt x Entitäten in "day". Warum wird der Buchungszeitraum nicht einfach in "booking" gespeichert? Nun, möchte man den Buchungsstatus einer Unterkunft zu einem bestimmten Tag abfragen, muss man lediglich die Entitätenmenge "day" durchsuchen und findet jeden einzelnen Buchungstag. Würde man stattdessen einen Zeitraum (von - bis) speichern, müsste man erst alle dazwischen liegenden Tage umständlich bestimmen und kann auch nicht so einfache Aussagen wie den Auslastungsgrad ableiten.

Der Preis der Unterkunft ist abhängig vom Buchungszeitraum, welcher durch die Entitätenmenge "season" ausgedrückt wird. Jeder Buchungszeitraum geht von einem bestimmten Tag bis zu einem bestimmten Tag und hat einen Namen. Anhand der Saison kann der Preis eindeutig bestimmt werden, ausgedrückt durch die Beziehung zwischen "season" und "accommodation". Dabei wird auch gespeichert, welche Leistungen im Preis inbegriffen sind (Frühstück, Halbpension oder Vollpension) und ob der Preis pro Person oder pro Zimmer gilt.

2.6.2. Umsetzung

2.6.2.1. Allgemein

Bevor die Unterkünfte auf der Internetpräsenz angezeigt werden können, müssen eine Reihe von Einstellungen wie Zimmertypen, Zimmerpreise, Kontaktadressen, Saisons usw. vorgenommen werden. Diese verschiedenen Einstellmöglichkeiten wurden auf unterschiedliche Konfigurationsseiten aufgeteilt und mit aussagekräftigen Namen versehen. So gibt es einen Zimmerkonfigurator, einen Preiskonfigurator, einen Saisonkonfigurator und einen Anbieterkonfigurator. Nur wenn alle Konfiguratoren durchlaufen und die nötigen Einstellungen gemacht wurden, wird die Unterkunft zur Buchung freigegeben. Unvollständige Informationen führen dazu, dass keine Zimmer gebucht werden können.

Die einzelnen Konfigurationsseiten können einfach über den Menüpunkt "Unterkunfts Einstellungen" aufgerufen werden. Sofern man mit dem Administrator-Account in Drupal eingewählt ist, findet man diesen Menüpunkt unter "Administer > Site configuration > Unterkunfts Einstellungen".

Das eigentliche Buchungsformular wurde aus Gründen der Übersichtlichkeit in 5 Schritte aufgeteilt, in denen jeweils spezielle Informationen abgefragt werden. Erst wenn das Buchungsformular bis Schritt 5 durchlaufen wurde, wird die Buchung tatsächlich durchgeführt. Der Buchende erhält dann automatisch eine Bestätigung per E-Mail.

2.6.2.2. Umsetzung der Formulare zum Einstellen der Unterkunft

Der wesentliche Programmieraufwand im Accommodation-Modul bestand darin, die einzelnen Formulare zum Einstellen der Unterkunft zu erstellen und benutzergerecht aufzuarbeiten. Alle Formulare arbeiten nach dem gleichen Prinzip, weshalb hier nur beispielhaft auf den Zimmertypkonfigurator eingegangen wird.

Zunächst muss im Drupal-Hook `accommodation_menu()` die Adresse (URL) des Zimmertypkonfigurators registriert werden, damit das Formular später unter einer bestimmten Adresse aufgerufen werden kann. Da es auch jeweils

ein Formular zum Bearbeiten und Löschen existierender Zimmertypen gibt, muss man insgesamt 3 Adressen registrieren. Im Parameter "callback arguments" wird jeweils der Name der PHP-Funktion angegeben, die für die Formularverarbeitung zuständig ist.

Beispiel 5.12. Registrierung der Pfade für den Zimmertypkonfigurator

```
/**
 * Implementation of hook_menu
 * This hook enables modules to register paths, which determines whose requests are to be
 * handled. Depending on the type of registration requested by each path, a link is placed
 * in the the navigation block and/or an item appears in the menu administration page.
 * @param may_cache A boolean indicating whether cacheable menu items should be returned.
 * @return array An array of menu items
 */
function accommodation_menu($may_cache) {

  global $user;
  $items = array();

  // we only allow the users to edit their own accommodations
  if (arg(0) == 'admin' && arg(1) == 'settings' && arg(2) == 'accommodation' &&
  is_numeric(arg(3)) && _accommodation_allowed(arg(3),TRUE)) {
    if (arg(4) == 'types') {
      $items[] = array(
        'path' => 'admin/settings/accommodation/'.arg(3).'/types',
        'title' => t('Zimmertypkonfigurator'),
        'description' => t('Einstellungen zu den Typen von Zimmern.'),
        'callback' => 'drupal_get_form',
        'callback arguments' => array('accommodation_type_form', arg(3)),
        'access' => user_access('create accommodation'),
        'type' => MENU_CALLBACK
      );
    }
    if (arg(4) == 'types' && is_numeric(arg(5)) && arg(6) == 'edit') {
      $items[] = array(
        'path' => 'admin/settings/accommodation/'.arg(3).'/types/'.arg(5).'/edit',
        'title' => t('Zimmertyp Bearbeiten'),
        'callback' => 'drupal_get_form',
        'callback arguments' => array('accommodation_type_form',arg(3),arg(5)),
        'type' => MENU_CALLBACK,
        'access' => user_access('create accommodation'),
      );
    }
    if (arg(4) == 'types' && is_numeric(arg(5)) && arg(6) == 'delete') {
      $items[] = array(
        'path' => 'admin/settings/accommodation/'.arg(3).'/types/'.arg(5).'/delete',
        'title' => t('Zimmertyp Löschen'),
        'callback' => 'drupal_get_form',
        'callback arguments' => array('accommodation_type_delete',arg(5),arg(3)),
        'type' => MENU_CALLBACK,
        'access' => user_access('create accommodation'),
      );
    }
  }
}
```

Um die Formularverarbeitung in Drupal zu verstehen, genügt es, einen Blick auf den Quelltext des Formulars `accommodation_type_form()` zu werfen. Dieses Formular wurde im obigen Beispiel registriert und kann nun definiert werden. Wie im folgenden Beispiel zu erkennen ist, wird ein multidimensionaler Array zum Erstellen der einzelnen Formularelemente verwendet. Anhand des Attributes "type" kann man den Typ des jeweiligen Formularelementes festlegen. Die Darstellung und Ausgabe des Formulars wird von Drupal abstrahiert und braucht nicht implementiert zu werden. Die Funktion gibt einfach nur den fertigen Formular-Array zurück.

Beispiel 5.13. Implementierung von `accommodation_type_form()`

```
/**
 * This function is used to configure the different types of rooms. It is not a Drupal
 * Hook.
 */
function accommodation_type_form($node_id,$type_id='') {

  // if the type_id is set to something, the user wants to load an accommodation
  type (in order to rename it later)
  if ($type_id) {
    $type = db_fetch_object(db_query("SELECT name from {accommodation_types} WHERE
  tid=%d", $type_id));
  }

  // this form element outputs a simple text
  $form['introduction'] = array(
    '#value' => '<p>'.t('Bitte legen Sie hier die einzelnen Zimmertypen Ihrer Unterkunft
  an. In der Liste am Ende dieser Seite sehen Sie bereits erstellte Zimmertypen. Die Anzahl
  der Betten in diesem Zimmertyp und weitere Angaben werden in einem anderen Formular
  eingestellt.').</p>',
  );

  // this one outputs a textfield
  $form['name'] = array(
    '#type' => 'textfield',
    '#title' => t('Name'),
    '#default_value' => $type->name, // show the name of the type if requested
    '#size' => 100,
    '#maxlength' => 100,
    '#required' => TRUE,
    '#description' => t('Bitte einen Namen für einen Zimmertyp angeben. Beispiele:
  Einzelzimmer, Doppelzimmer, Suite')
  );

  // if a form has a submit function, then hidden form values can be declared as such
  $form['type_id'] = array(
    '#type' => 'hidden',
    '#value' => $type_id
  );

  $form['node_id'] = array(
    '#type' => 'hidden',
    '#value' => $node_id
  );

  $form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Speichern'),
  );

  // initialise table header and variable for the table body
  $form['headline'] = array('#value' => '<h2>'.t('Bereits erstellte Zimmertypen').</h2>');
  $header = array(t('Name'),t('Aktion'));
  $rows = array();

  // select the all the rooms of this accomodation building
  $queryResult = db_query("SELECT tid,name FROM {accommodation_types} WHERE nid = %d",
  $node_id);

  // we check first if we have results, otherwise we don't display the results table
  if (db_num_rows($queryResult) > 0) {
    while ($type = db_fetch_object($queryResult)) {
      $row = array();
      $row[] = $type->name;
      $row[] = array('data' => l(t('umbenennen'), 'admin/settings/accommodation/' . $node_id .
  '/types/' . $type->tid . '/edit') . ' ' . l(t('löschen'), 'admin/settings/accommodation/' . $node_id .
  '/types/' . $type->tid . '/delete'), 'align' => 'left');
      $rows[] = $row;
    }
  }
}
```

```
    }
    $form['overview'] = array('#value' => theme('table', $header, $rows, array('class' =>
'package')));
  }
  else {
    $form['overview'] = array('#value' => '<p>'.t('Es wurde noch kein Zimmertyp erstellt.').
'</p>');
  }

  return $form;
}
```

Wenn ein solches Formular auf der Internetpräsenz abgeschickt wird, sucht Drupal automatisch nach einer Funktion, die das Formular validiert. Bezogen auf das Zimmertypformular lautet die Validierungsfunktion `accommodation_type_form_validate()`. Nach erfolgreicher Validierung können die jeweiligen Eingaben des Formulars weiterverarbeitet werden. Dazu muss eine Funktion namens `accommodation_type_form_submit()` implementiert werden.

Beispiel 5.14. Implementierung von `accommodation_type_form_submit()`

```
/**
 * This function is not a hook, but it's a function of the Drupal Forms API. It is used
 * to store the data of the accommodation type in to the database.
 * @param form_id A unique string identifying the accommodation type form
 * @param form_values An associative array containing the submitted values of the form
 */
function accommodation_type_form_submit($form_id, $form_values) {

  // that means the user added a new accommodation type
  if (empty($form_values['type_id'])) {
    db_query("INSERT INTO {accommodation_types} (name,nid) VALUES ('%s',%d)",
$form_values['name'], $form_values['node_id']);
    drupal_set_message(t('Der Zimmertyp wurde erfolgreich angelegt.'));
  }
  // the user renamed an existing accommodation type so we have to update the values
  else {
    db_query("UPDATE {accommodation_types} SET name='%s' WHERE tid=%d",
$form_values['name'], $form_values['type_id']);
    drupal_set_message(t('Der Zimmertyp wurde erfolgreich umbenannt.'));
    drupal_goto('admin/settings/accommodation/' . $form_values['node_id'] . '/types');
  }
}
```

Auf einer ähnlichen Art und Weise funktioniert auch das Formular zur Buchung von Unterkünften, nur das hier das Formular auf 5 Teilformulare aufgeteilt wurde.

3. Konfiguration der neuen Module

3.1. Googlemap

Bevor das Modul verwendet werden kann, muss in der Drupal-Verwaltungsoberfläche unter "Administer > Site configuration > Google Map Optionen" der API Key sowie die Standardeinstellungen für neue Google Maps eingestellt werden. Erst danach können Inhalte vom Typ "googlemap" mit Drupal erstellt werden.

3.2. Mapmarker

Nach der Installation des Modules muss zunächst der Flickr und YouTube-Schlüssel eingetragen werden. Ohne diese Schlüssel können keine Fotos oder Videos der beiden Dienste angezeigt werden. Gleichzeitig müssen alle Knotentypen festgelegt werden, die auf einer Google Map verlinkt werden dürfen. Die Konfigurationsoberfläche findet sich unter "Administer > Site configuration > Kartenmarkierungs Optionen".

3.3. Accommodation

Bei der Aktivierung des Moduls in Drupal wird im Taxonomy-Modul automatisch eine Kategorie namens "Unterkunftstyp" erstellt. Diese Kategorie muss anschließend noch mit Vokabeln wie Hotel, Pension, Ferienhaus, Privatunterkunft gefüllt werden. Damit wird der Benutzer beim Erstellen einer Unterkunft gezwungen, den Typ seiner Unterkunft auszuwählen.

3.4. Lifeform

Bei der Aktivierung des Moduls werden die Kategorien "Tierwelt" und "Pflanzenwelt" im Taxonomy-Modul erstellt. Damit kann man eine erste Gliederung der Lebewesen im Nationalpark vornehmen, jedoch ist dies noch sehr grob. Unter "Administer > Content-Management > Categories" können dann weitere Unterkategorien erstellt werden, die das Lebewesen noch besser kennzeichnen.

3.5. Mappath und Observation

Bei diesen beiden Modulen müssen nach der Installation keine weiteren Einstellungen vorgenommen werden.

Kapitel 6. Test der Benutzeroberfläche

Eines der Hauptanliegen des Web 2.0 ist es, die Anwendungen auf den Benutzer auszurichten. Das beinhaltet eine klare und intuitive Benutzeroberfläche genauso wie eine verständliche Menüführung. Dieses Kapitel beschreibt ein Testverfahren zur Optimierung der Benutzeroberfläche und stellt die Protokolle und Ergebnisse dieses Testverfahrens dar.

1. Testverfahren

Eine Evaluation der Benutzeroberfläche mit echten Benutzern ist unabdingbar für ein benutzerfreundliches Design. Zwar können Entwickler schon bei der Planung und Implementierung auf eine hohe Benutzerfreundlichkeit achten, jedoch wissen sie bereits, wie die fertige Anwendung funktionieren soll und können sich deshalb nur schwer in den jeweiligen Endnutzer hineinversetzen. Benutzertests mit verschiedenen Personen garantieren einen wesentlich objektiveren und kritischeren Blick (Vgl. [HV03] S 15).

Diese Bachelorarbeit nutzt die Task-Analyse, auch Aufgabenanalyse genannt, zur Evaluation der Benutzeroberfläche. Dabei bekamen Testpersonen, die nie vorher mit Drupal bzw. mit der Internetpräsenz gearbeitet haben, konkrete Aufgaben und sollten eigenständig diese Aufgaben erledigen. Die Tester wurden angewiesen, dem Testleiter jeden Schritt und jede Eingabe zu erläutern. Damit lassen sich später Rückschlüsse auf unzulängliche oder irreführende Erklärungen auf der Benutzeroberfläche ziehen. Die Rolle des Testleiters hat Sascha Nehls eingenommen. Er hat sich während der Testphase im Hintergrund gehalten und nur auf Anfrage Hinweise gegeben, um das Nutzerverhalten nicht zu manipulieren. Ziel war es, unlogische oder unverständliche Benutzeroberflächen in den neu entwickelten Modulen zu finden. Die Module des Drupal-Kerns wurden keinen Benutzertests unterworfen, genauso wie die Benutzeroberflächen, die nur dem Superadministrator zugänglich sind.

Die Testpersonen wurden ganz bewusst aus verschiedenen Alters- und Gesellschaftsgruppen rekrutiert, da die Zielgruppe der Internetpräsenz eine ähnlich breite Nutzergruppe aufweist. Alle Tester hatten bereits Erfahrungen im Umgang mit dem Internet, jedoch keine Erfahrungen mit einem Content-Management-System.

2. Aufgabenstellung "Hotel einrichten"

Aufgabenstellung: Sie sind Besitzer eines Hotels mit dem Namen "Hotel am See" und möchten nun das Hotel für die Online-Buchung einstellen. Führen Sie nun selbstständig die Konfiguration des Hotels mit seinen Zimmern und Preisen durch. Erläutern Sie dabei jeden Schritt, den Sie unternehmen und berichten Sie sofort über Dinge, die Ihnen unklar erscheinen. Die Prüfungsaufsicht wird Ihnen nur wenig Hilfestellung bieten, versuchen Sie die Aufgabe alleine zu lösen.

Das Hotel hat 2 Einzelzimmer und 2 Doppelzimmer. Die Preise der Zimmer sind von der Saison abhängig. In der Wintersaison kostet das Einzelzimmer 29 EUR pro Person, in der Sommersaison 35 EUR pro Person. Das Doppelzimmer kostet im Winter 25 EUR pro Person respektive 39 EUR im Sommer. Die Sommersaison reicht vom 1. Juli bis 31. September, die restliche Zeit ist Wintersaison. Bitte entscheiden Sie selbst, welche Serviceleistungen Ihre Unterkunft anbietet.

2.1. Testprotokoll 1

Die Testperson hier ist Ute Nehls, 51. Sie ist Angestellte bei einem Vermessungsunternehmen und nutzt das Internet beruflich und privat. Der Test begann um 20.30 Uhr und endete 21.25 Uhr.

Ute hat sich sofort zum Unterpunkt "Unterkunft anlegen" bewegt und hat alle Eingaben im Formular korrekt eingegeben. Lediglich bei den Leistungen wusste sie nichts mit den Unterpunkten "WLAN kostenlos" und "WLAN kostenpflichtig" anzufangen. Außerdem wurde ein Punkt "Shuttleservice" von der Testperson vermisst.

Im Abschnitt der Verlinkung der Unterkunft mit der Google Map, wusste Ute nichts mit den Icons anzufangen und hat keine Beschreibung zu den einzelnen Icons gefunden. Sie war nicht in der Lage, eigenständig eine Markierung auf der Google Map zu erstellen. Ebenso wurde die Verknüpfung zwischen YouTube, Flickr und den

Schlüsselwörtern nicht auf Anhieb verstanden. Der Testleiter musste hier die Zusammenhänge und die weiteren Schritte erklären.

Nachdem die Unterkunft erstellt worden war, wusste Ute zunächst nicht, was nun zu tun ist. Unter Anleitung wurde sie zum Saisonkonfigurator geleitet. Die Bedienung dort fiel ihr sehr leicht und sie konnte problemlos die einzelnen Saisons erstellen. Bei einer Saison gab es eine zeitliche Überlappung, so dass Ute diese erst korrigieren musste, bevor die Saison akzeptiert wurde.

Anschließend ging Ute selbstständig zum Zimmertypkonfigurator über und war zunächst irritiert, warum die Anzahl dieses Zimmertyps nirgendwo eingegeben werden konnte. Anschließend hat Ute den Weg zum Zimmerkonfigurator gefunden und die einzelnen Zimmer korrekt angelegt. Lediglich im Preiskonfigurator hatte Ute Probleme, wieder zurück zum Zimmerkonfigurator zu kommen.

Danach rief Ute die Buchungsübersicht auf und war über die grünen Symbole verwundert. Der Test war damit abgeschlossen.

2.2. Änderungen aufgrund von Testprotokoll 1

Bei den Serviceleistungen unter "Unterkunft anlegen" wurde ein Auswahlfeld "Shuttle Service" hinzugefügt. Außerdem wurden die Icons für die Google Map beschrieben. Die Erklärung zum Erstellen einer Markierung auf der Google Map wurde erweitert. Der Hilfetext im Zimmertypkonfigurator wurde verbessert. Beim Speichern im Preiskonfigurator wird automatisch zum Zimmerkonfigurator zurückgegangen. Der Verweis zum Anbieterkonfigurator wurde offensichtlicher gestaltet, da die Testperson den Verweis komplett übersehen hatte.

2.3. Testprotokoll 2

Die Testperson ist Bernhard Nehls, 55. Er ist Angestellter in einem Bauunternehmen und hat beruflich und privat mit dem Internet zu tun. Der Test begann um 9.45 Uhr und endete 11.30 Uhr.

Bernhard bewegte sich direkt zur Seite "Unterkunft anlegen" und hat das Formular korrekt begonnen. Er vermisste jedoch eine Erklärung, für was diese Daten verwendet werden und ob diese im Internet erscheinen. Im Feld "Straße" hatte Bernhard lediglich die Straße ohne Hausnummer eingegeben, obwohl dies im Hilfetext erwähnt wurde. Bei der Leistungsübersicht wurden Auswahlfelder für Fitness und Wellness vermisst. Außerdem wünschte er sich eine Kategorisierung der Unterkunft, zum Beispiel "Pension", "Hotel", "Bungalow".

Beim Erstellen der Verlinkung mit der Google Map hatte der Tester Probleme und konnte erst unter Anleitung die Markierung erstellen. Er wünschte sich hierbei ein Auswahlfeld, mit dem man diese Option ausschalten kann. Bernhard wusste mit dem Begriff "Icon" nichts anzufangen und wünschte sich ein deutsches Wort dafür. Ebenso wusste er nichts in das Feld Schlüsselwörter einzufügen. Er hat das Eingabefeld frei gelassen und die Eingaben gespeichert. Anschließend wurde er aufgefordert, ein Schlüsselwort einzugeben, damit die Daten gespeichert werden können.

Nach erfolgreicher Speicherung hat Bernhard den Anbieterkonfigurator besucht, obwohl er zunächst verwirrt war, was ihn unter "persönliches Profil" erwartete. Im Anbieterkonfigurator vermisste Bernhard einen Hilfstext beim Eingabefeld "Form". Das Eingabefeld "Name" verwirrte Bernhard, da er nicht wusste, was für einen Namen er dort eingeben sollte. Den Hilfetext direkt unter dem Eingabefeld hatte er nicht beachtet. Er hat Informationen darüber vermisst, ob und wie diese Daten im Internet veröffentlicht werden.

Im Saisonkonfigurator hat er einzelne Saisons erfolgreich angelegt und bearbeitet, jedoch wurde ein Saisonname doppelt vergeben.

Die Preise für die Zimmer wurden vom Tester eigenständig angelegt.

2.4. Änderungen aufgrund von Testprotokoll 2

Auf der Seite "Unterkunft erstellen" wurde ein Text eingefügt, der die Verwendung der angegebenen Daten erläutert. Außerdem wurde das Eingabefeld "Straße" in "Straße mit Hausnummer" umbenannt. Die Anleitung zum Erstellen eines Markers auf der Google Map wurde überarbeitet. Für Unterkunftsanbieter wird nun automatisch

das Hotel-Icon verwendet und der Benutzer sieht keine anderen Icons mehr. Die Überschrift "Icon" wurde zu "Symbol der Kartenmarkierung" geändert.

Der beschreibende Text zum Anbieterkonfigurator wurde überarbeitet. Im Anbieterkonfigurator wurde beim Eingabefeld "Form" ein Hilfetext hinzugefügt, die Überschrift des Eingabefeldes "Name" zu "Name des Unternehmens" umbenannt sowie ein neues Eingabefeld "Mobiltelefon" hinzugefügt.

Beim Bearbeiten einer Saison wird nach dem Speichern automatisch wieder die Ursprungsseite des Saisonkonfigurators geladen. Der Bestätigungsknopf zum Löschen einer Saison wurde umbenannt. Außerdem wurde eine Formularüberprüfung implementiert, welche doppelte Saisonnamen nicht erlaubt.

3. Aufgabenstellung "Hotelbuchung durchführen"

Aufgabenstellung: Sie haben sich auf der Internetpräsenz über das Urlaubsziel informiert und möchten nun ein Zimmer buchen. Suchen Sie bitte dazu das Hotel "Zur Müllerwiese" und buchen Sie ein Doppelzimmer für 2 Personen vom 20. bis 22. Oktober 2007.

3.1. Testprotokoll 3

Die Testperson ist Enrico Speer, ein 29-jähriger Kommissionierer aus Berlin. Er hat sich auf der Internetpräsenz ein wenig umgeschaut und das Hotel eher zufällig auf der Google Map gefunden. Über den Link "Buchen" ist er zum Buchungsformular des Hotels gekommen. Er hat das Formular zur Auswahl des Buchungszeitraums und der Unterkunft richtig ausgefüllt. Er hatte jedoch angemerkt, dass er gerne eine Kalenderansicht zur Buchung hätte, damit er auch die Wochentage einsehen kann. Nach dem Klick auf "Verfügbarkeit prüfen" hat er sich für ein freies Doppelzimmer entschieden. Bei der anschließenden Eingabe seiner Kontaktdaten hat er im Feld "Name" nur seinen Nachnamen eingetragen und er hat ein Eingabefeld für den Vornamen vermisst. Bei der Eingabe der Telefonnummer hat er angemerkt, dass viele Anbieter die Telefonnummer in ein Feld für die Vorwahl und ein Feld für die Telefonnummer aufspalten. Er hat sich gefreut, dass er noch eine Buchungsübersicht bekommen hat, bevor er die Buchung endgültig abgeschlossen hat.

3.2. Änderungen aufgrund von Testprotokoll 3

Das Eingabefeld für den Namen des Buchenden wurde von "Name" in "Vor- und Nachname" umbenannt. Für die Eingabe des Anreisedatums wurde das Modul JavaScript Tools¹ installiert, welches die Eingabe des Datums über einen Popup-Kalender ermöglicht. Dieser Kalender wurde jedoch als zu schwergewichtig empfunden und wurde daher nicht in die endgültige Version des Accommodation-Moduls eingebaut. Das Eingabefeld für die Telefonnummer wird nicht aufgespalten, da eine zu hohe Granularität die Eingabe eher erschwert als vereinfacht.

3.3. Testprotokoll 4

Susanne Gruß, eine 20-jährige Studentin der Stadt- und Regionalplanung an der TU Cottbus hat sich bereit erklärt, den Test zu machen. Der Test begann um 17.30 und endete um 17.40 Uhr.

Sie wurde auf die Internetpräsenz des Nationalparks geführt und begann sofort mit der Suche des Hotels. Sie fand das Hotel, indem sie auf den Menüpunkt "Unterkünfte" und anschließend auf "Hotels" klickte. Sie wählte das Hotel und den Buchungszeitraum korrekt aus und hat anschließend das frei verfügbare Doppelzimmer ausgewählt. Auf der Seite zu ihren Personendaten hatte sie ebenfalls alles korrekt ausgefüllt und die Buchung korrekt ausgeführt. Die Aufgabe wurde somit ohne Probleme gelöst. Bei einer anschließenden Nachfrage nach Verbesserungswünschen hat sie angemerkt, dass sie beim Buchen gerne eine Übersicht hätte, wie viele Schritte nötig sind bzw. wie viele Schritte sie noch auszuführen hat, bis die Buchung abgeschlossen ist.

3.4. Änderungen aufgrund von Testprotokoll 4

Die Buchung wurde ohne Hilfestellungen korrekt durchgeführt, es gab also keinen Handlungsbedarf, etwas zu ändern. Es wurde lediglich eine Anzeige mit den notwendigen Schritten bei dem Buchungsformular hinzugefügt. Damit ist die Entwicklung dieser speziellen Benutzeroberfläche abgeschlossen.

¹<http://drupal.org/project/jstools>

4. Aufgabenstellung "Naturbeobachtung melden"

Aufgabenstellung: Sie haben Ihren Urlaub im Nationalpark verbracht und waren von der Natur sehr begeistert. Nun möchten Sie die Beobachtung eines Weißstorchs melden und dazu gleich das passende Bild hochladen. Hinweis: Sie müssen sich zunächst auf der Internetpräsenz registrieren und können anschließend eine Naturbeobachtung melden.

4.1. Testprotokoll 5

Auch hier hat sich die 20-jährige Susanne Gruß als Testperson zur Verfügung gestellt. Sie war bereits ein wenig mit der Internetpräsenz vertraut, da sie vorher eine Zimmerbuchung durchgeführt hatte. Die Aufgabenstellung hat sie sofort verinnerlicht und ist zielstrebig auf den Registrieren-Link in der oberen Navigationsleiste gegangen. Dort wunderte sie sich zunächst, warum es eine Seite für "Anmelden" und eine andere Seite für "Registrieren" gab. Nachdem sie sich die Erklärungen durchgelesen hatte, hat sie das Registrierungsformular korrekt ausgefüllt und sich anschließend bei ihrem E-Mail-Anbieter eingewählt. Dort hat Sie das Passwort kopiert und in das Login-Formular für Drupal eingefügt.

Nach erfolgreichem Login hat sie sich zunächst mit den einzelnen Menüs vertraut gemacht und sich über die Bezeichnung "Mein Konto" gewundert. Beim Durchstöbern der Seite fand sie nun die Google Map mit den Kartenmarkierungen und hat sich gefragt, wo sie denn nun eine eigene Kartenmarkierung hinzufügen kann.

Sie durchsuchte das Menü weiter und fand den Verweis "Inhalt erstellen". Dort wurde sie auf den Verweis "Beobachtung" aufmerksam, den sie auch sogleich aufrief. Mit dem nun auszufüllenden Formular hatte sie keine Probleme. Interessanterweise sah sie aber bei Eingabe des Wortes "Weißstorch" den Vorschlag nicht und hat das komplette Wort selbst eingegeben. Beim Anfügen des Fotos wunderte sie sich, warum die Erklärungen in Englisch sind. Das Eingabefeld für die Schlüsselwörter sowie die Kontrollkästchen für Flickr und YouTube beachtete sie nicht. Nach erfolgter Speicherung sah sie sich das Ergebnis auf der Nationalparkkarte an und war erfreut. Der Test war damit abgeschlossen.

4.2. Änderungen aufgrund von Testprotokoll 5

Die teilweise verwirrenden Bezeichnungen bei der Registrierung können nur durch Änderungen im Drupal-Kern behoben werden. Dies bringt Probleme bei späteren Aktualisierungen mit sich und wird daher nicht in Betracht gezogen. Es wurde aber nach einer Übersetzung für das Image-Modul gesucht, jedoch keine gefunden. Da es sich um ein externes Modul handelt, wurde keine weitere Maßnahme unternommen, die Texte zu übersetzen. Man hätte hier zwei Möglichkeiten: Entweder man übersetzt die Moduldatei manuell ins Deutsche, oder man erstellt eine eigene Sprachdatei und installiert diese nachträglich in Drupal. Für diese Arbeit sollte die englische Version vorerst genügen.

Da die Testperson zunächst nicht wusste, wo sie ihre Naturbeobachtung erstellen kann, wurde ein Informationstext unter der Google Map eingefügt. Dieser erklärt, wie man eigene Naturbeobachtungen auf der Karte darstellen kann. Wie auch in vorherigen Tests ersichtlich wurde, wissen die meisten Benutzer nichts mit Flickr und YouTube anzufangen. Beim Erstellen von Markierungen auf der Google Map führt dies immer wieder zu Verwirrungen. Also wurde eine neue Rolle im Mapmarker-Modul eingeführt. Diese Rolle bestimmt, welche Nutzergruppen die Eingabefelder für Flickr, YouTube und die Schlüsselwörter sehen. Normale registrierte Besucher sehen nun die betreffenden Eingabefelder nicht mehr, im Gegensatz zu den anderen Gruppen wie die Nationalparkverwaltung oder die Unterkunftsanbieter.

4.3. Testprotokoll 6

Nicole Gabel, eine 21-jährige Architekturstudentin aus Cottbus, hat auf Anhieb den Verweis zum Registrieren gefunden. Sie hat die Daten im Registrierungsformular korrekt eingegeben, wunderte sich aber, warum sie kein Passwort festlegen musste. Die Registrierungsbestätigung hat sie gelesen und sich mit dem generierten Passwort in die Internetpräsenz eingewählt. Anschließend probierte Sie den Link "Meine Beobachtungen" um zu sehen, ob sie dort eine Naturbeobachtung anlegen kann. Dies war nicht der Fall, also suchte sie weiter und fand schließlich die Seite zum Erstellen einer Naturbeobachtung.

Dort hat sie alles korrekt eingegeben und ein Bild angehängen. Die Aufgabe war damit erfolgreich abgeschlossen.

4.4. Änderungen aufgrund von Testprotokoll 6

Da die Testperson zunächst "Meine Beobachtungen" besuchte, um eine Naturbeobachtung zu melden, wurde auf dieser Seite ein Verweis eingefügt. Dieser Verweis führt direkt zum Erstellen einer Naturbeobachtung. Auf die gleiche Art und Weise wurden Verweise bei "Meine Unterkünfte", "Meine Google Maps", "Meine Kartenpfade" und "Meine Lebewesen" eingefügt.

Außerdem wurde der Text "Es wurde noch keine Tier- bzw. Pflanzenbeobachtungen gemeldet" in "Sie haben noch keine Tier- bzw. Pflanzenbeobachtung gemeldet" umbenannt.

Kapitel 7. Ausblick

Die in dieser Bachelorarbeit erstellte Internetpräsenz bedient aktuelle Konzepte des Web 2.0 und vernetzt bestehende Web 2.0-Dienste miteinander. Zudem haben Benutzer die Möglichkeit bekommen, selbst Inhalte zu publizieren und zu verwalten. Damit entspricht die Internetpräsenz dem aktuellen Stand der Web-Entwicklung, jedoch werden in der Zukunft wohl auch noch andere Formen des Webs eine große Rolle spielen.

1. Virtuelles World Wide Web

In den Anfängen des World Wide Webs war häufig vom Cyberspace oder virtuellen Welten die Rede. Die virtuellen Welten zu dieser Zeit beschränkten sich jedoch auf textbasierte Rollenspiele, in denen die Benutzer mittels Textkommandos durch verschiedene Räume streifen konnten. Eine Verbesserung stellte erst die "Virtual Reality Modeling Language" (VRML) dar, die 1997 ihren Höhepunkt durch die Verwendung in den Chaträumen von CyberWorld hatte (Vgl. [Alb07] S 167).

Die Idee von damals wurde 2003 wieder aufgegriffen, als Linden Lab die virtuelle 3D-Welt Second Live¹ vorstellte. Durch schnellere Internetanbindungen und schnellere Heim-Computer ist es damit möglich, in einer virtuellen Welt umherzulaufen, auf andere Leute zu treffen und mit ihnen zu sprechen. Es ist sogar möglich, "Land" zu kaufen und somit sein eigenes Grundstück in dieser virtuellen Welt zu besitzen. Solche Welten können auch ins reale Leben übertragen werden, man stelle sich nur vor, Studenten gehen nicht körperlich zu einer Vorlesung sondern von zu Hause aus über eine virtuelle Welt. Solche Szenarien sind heute technisch möglich und es scheint nur noch eine Frage der Zeit, wann sie sich durchsetzen. Laut einer Online-Studie von ARD und ZDF nutzen erst 3% der Internetnutzer virtuelle Spielwelten wie Second Life (Vgl. [GF07]). Es bleibt abzuwarten, wie sich diese Netzform in Zukunft entwickelt.

2. Mobiles World Wide Web

Nicht erst mit der Einführung des iPhones² ist das mobile WWW Wirklichkeit geworden. Sogenannte Smart-Phones oder Blackberrys im Zusammenspiel mit UMTS erlauben schon seit geraumer Zeit die Nutzung des mobilen WWWs, bislang wird dies allerdings nur wenig verwendet. Dennoch bieten sich hier vielfältige Möglichkeiten, wie [Alb07] herausgefunden hat:

- Durch standortbezogene Dienste (engl. Location Based Services) könnten Naturliebhaber während ihres Urlaubs die Sehenswürdigkeiten, Restaurants oder Unterkünfte in ihrer näheren Umgebung herausfinden. Ebenso könnte das aktuelle Kinoprogramm oder das nächste Krankenhaus gefunden werden. Das Mobiltelefon wird so zum Fremdenführer, Navigationselement und Notfall-Wegweiser.
- Mobiles Blogging erlaubt die zeitnahe Veröffentlichung von Fotos, Videos oder Texten mit dem Mobiltelefon. Menschen könnten so Informationen in Echtzeit an bestimmte Web 2.0-Dienste übertragen, die wiederum diese Informationen an andere Benutzer weitergeben. Aktuelle Tiersichtungen könnten so sofort ins World Wide Web gespeist werden und andere Naturliebhaber informieren.

Diese und andere Szenarien sind von technischen Gesichtspunkten bereits möglich, jedoch sind die Mobiltelefone momentan noch umständlich zu bedienen und bieten vergleichsweise schlechte Qualität bei hohen Verbindungskosten. Wenn die Verbindungskosten im Laufe der Zeit sinken und die Mobiltelefone besser werden, könnte das mobile WWW eine echte Bereicherung werden.

3. Semantisches Web

Auch wenn das Semantische Web (engl. Semantic Web) nichts im direkten Sinne mit dem Web 2.0 zu tun hat, so stellt es doch eine innovative Netzform dar, die in Zukunft eine deutlich größere Rolle spielen kann. Das Semantic Web ist eine Vision von Tim Berners-Lee, dem Erfinder des World Wide Webs, und stellt eine Erweiterung des

¹<http://secondlife.com/>

²iPhone ist die Bezeichnung für ein Mobiltelefon von Apple Inc., welches die Darstellung von Internetseiten auf einem hochauflösendem Bildschirm erlaubt.

WWW's um maschinenlesbare Daten dar. Auf Internetseiten veröffentlichte Informationen können so auch von Maschinen gelesen, interpretiert und weiterverarbeitet werden. Eins muss klar sein, das volle Potential des World Wide Webs kann nur dann ausgeschöpft werden, wenn die Daten von Maschinen genauso genutzt und verarbeitet werden können wie von Menschen. Das Semantische Web stellt einen wichtigen Schritt in diese Richtung dar.

Um dies zu ermöglichen, muss die Semantik der Daten und die Beziehungen zueinander in einer formalen Sprache wie RDF beschrieben werden.

Kapitel 8. Fazit

Das Web 2.0 hat sich von einem Marketingwort zu einer neuen Netzform entwickelt, die aus dem World Wide Web nicht mehr wegzudenken ist. Es ist heutzutage allgegenwärtig, auch wenn es von vielen Personen nicht als Web 2.0 wahrgenommen wird.

Diese Bachelorarbeit hat das Web 2.0 als ein verbessertes, auf den Benutzer zugeschnittenes World Wide Web vorgestellt, in dem die Benutzer dazu animiert werden, selbst Inhalte zu erstellen und zu veröffentlichen. Der Benutzer ist somit nicht mehr nur Konsument von Informationen, sondern auch Produzent. Durch die immer schnelleren Internetverbindungen wurde es möglich, auch multimediale oder interaktive Inhalte auf Internetseiten darzustellen. Zudem spielen soziale Netzwerke beim Web 2.0 eine überaus große Rolle. All diese Ideen wurden auch in dieser Arbeit aufgegriffen und in Form einer Internetpräsenz für Nationalparks umgesetzt.

Hierzu wurden neue Module für das Content-Management-System Drupal entwickelt, welche verschiedene Web 2.0-Dienste über Programmierschnittstellen (APIs) miteinander vernetzen. Die Darstellung der Google Map nimmt hierbei eine zentrale Rolle ein, da sie einerseits über die Topografie des Nationalparks informiert, andererseits aber auch zur Vernetzung mit den Foto- und Videodiensten benötigt wird.

Trotz der vielen technischen Feinheiten auf der Internetpräsenz, ist immer noch genug Freiraum für weitere Verbesserungen. So könnte die Vernetzung von Wikipedia mit der Tier- und Pflanzendatenbank sowie mit den einzelnen Orten auf der Google Map vorangetrieben werden. Leider gibt es zur Zeit keine Programmierschnittstelle (API) zur Abfrage der Wikipedia-Datenbank und auch die Google-API bietet nicht die Möglichkeit, Orte auf der Karte als einzelne Objekte zu behandeln. Es bleibt abzuwarten, wie sich diese Dienste in Zukunft entwickeln. Gerade die Google-API entwickelt sich rasant und bietet immer mehr Möglichkeiten.

Auch wäre eine Vernetzung der Fotos und Videos über die Geo-Position sehr interessant. Der Bilderdienst Flickr bietet den Benutzern bereits die Möglichkeit, ihre Bilder mit Positionsdaten zu versehen. Da der Datenbestand dieser Bilder momentan aber noch äußerst gering ist, wurde die Vernetzung mit der Google Map über Schlüsselwörter realisiert. Wenn mehr und mehr Benutzer ihre Bilder mit Positionsdaten versehen, wird es auch für diese Internetpräsenz interessant und man könnte die Bilder direkt über die Positionsdaten, und damit ohne Schlüsselwörter, auf der Google Map anzeigen.

Ein weiterer Verbesserungsansatz liegt im Mappath-Modul. So könnten bei Rad- und Wanderwegen mehrere Zwischenstopps angezeigt werden, die ganz spezifische Informationen zu diesem Halt anzeigen. Weitere Verbesserungen betreffen vor allem das Image-Modul: Das Hochladen von Bildern funktioniert zwar tadellos, jedoch ist die Verwaltung äußerst kompliziert und bietet keine Multi-Nutzer-Funktionalität. Es bleibt abzuwarten, ob und wie die Bildverwaltung in Drupal 6.0 verbessert wird.

Eine weitere Idee zur Verbesserung der Internetpräsenz wäre die Anzeige von Entfernungen im Informationsfenster der Google Map. So könnte die Distanz zur nächsten Stadt, zum nächsten Restaurant o.ä. angezeigt werden. Basierend auf dieser Idee, kann auch eine Art Reiseplaner implementiert werden. Nutzer könnten sich dort ihre Urlaubsziele zusammenstellen und anschließend auf einer Google Map übersichtlich darstellen lassen.

Das Content-Management-System Drupal ist in jedem Fall ein geeignetes Framework, um solche Funktionen umzusetzen. Mit der neuen Version 6 wird Drupal nochmals verbessert, ohne dabei die Flexibilität und Leichtigkeit dieses Systems zu vernachlässigen. Bei der Entwicklung neuer Web 2.0-Module sei jedoch eines angemerkt; die von Web 2.0-Diensten angebotenen Daten und Schnittstellen können sich jederzeit ändern bzw. sogar ganz eingestellt werden. So hatten sich zum Ende des Bachelorprojekts die Methoden und Parameter der YouTube-API komplett geändert und eine neue Programmierschnittstelle wurde veröffentlicht. Glücklicherweise ist die vorherige Programmierschnittstelle noch bis zum August 2008 gültig, eine Änderung der erstellten Funktionen war also nicht nötig..

Zusammenfassend kann man sagen, dass das Web 2.0 nicht mehr nur ein Marketingwort, sondern tatsächlich eine neue Form des Internets darstellt. Es verbindet Technologien auf eine neue Art und Weise und bietet einen tatsächlichen Mehrwert. Allerdings benötigt das Web 2.0 dazu unbedingt die Inhalte der Benutzer. Gibt es nicht genügend Benutzer, die ihre Meinungen, Fotos, Reisetips o.ä. veröffentlichen, bietet eine Web 2.0-Internetpräsenz nur wenig Mehrwert gegenüber einer herkömmlichen Internetpräsenz. Betreiber von Web 2.0-Internetpräsenzen müssen daher immer die Benutzer dazu animieren, selbst aktiv zu werden und Inhalte zu produzieren.

Die Möglichkeiten dazu sind mit dieser Bachelorarbeit geschaffen worden. Wie sehr diese Möglichkeiten von den Besuchern der Internetpräsenz angenommen werden, wird die Praxis zeigen. In jedem Fall ist das WWW einer hohen Dynamik unterworfen und als Betreiber einer Internetpräsenz sollte man sich dieser Dynamik nicht verschließen. Auch muss es immer klar sein, man gestaltet die Internetpräsenz nicht für sich, sondern für den Benutzer. Dies gilt für das Web 2.0 mehr denn je.

Anhang A. Pflichtenheft

Das Pflichtenheft enthält eine Zusammenfassung aller fachlichen Anforderungen der Internetpräsenz. Die Gliederung erfolgt nach dem Muster von [Bal01].

1. Zielbestimmung

Es wird eine Internetpräsenz für Nationalparks erstellt, welche durch das Content-Management-System Drupal 5.3 verwaltet wird. Diese Internetpräsenz vernetzt verschiedene Web 2.0-Anwendungen und bietet registrierten Besuchern die Möglichkeit, in einem Forum zu diskutieren sowie eigene Naturbeobachtungen zu veröffentlichen. Des Weiteren können Veranstaltungen, Weblogs und Wanderrouten eingesehen sowie Unterkünfte gebucht werden.

Ein Nutzerkreis bestehend aus der Nationalparkverwaltung, Veranstaltungsmanagern und Unterkunftsanbietern kann gezielt Informationen auf der Internetpräsenz publizieren.

1.1. Musskriterien

Für die Administration der Inhaltsseiten, der Foren, der Weblogs sowie für die Benutzer- und Rechteverwaltung werden bestehende Drupal Module verwendet und ggf. geringfügig angepasst. Für die Verwaltung der Unterkünfte und der Buchungen, für die Verwaltung der Naturbeobachtungen inklusive der Tier- und Pflanzendatenbank sowie für die Verwaltung der Google Maps werden neue Module für Drupal implementiert. Insbesondere die Vernetzung der Google Map mit den anderen Modulen bedient eines der Schlüsselkonzepte des Web 2.0. So können Standorte von Unterkünften, Ausflugsziele, Naturbeobachtungen sowie Rad- und Wanderwege übersichtlich auf einer Google Map dargestellt werden. Interessante Orte im Nationalpark können mit Bildern von Flickr und Videos von YouTube bereichert werden. Die Verknüpfung zwischen einem Punkt auf der Karte und den Bildern bzw. Videos erfolgt über Schlüsselwörter.

Zur eigentlichen Erzeugung von Inhalten auf der Internetpräsenz sind 4 Benutzergruppen mit Zugang zu bestimmten Bereichen des Content-Management-Systems vorgesehen. Darüber hinaus gibt es einen Superadministrator in Drupal, der eine Kontrollfunktion einnimmt und hier nicht als eigentliche Benutzergruppe aufgeführt wird. Mehr Informationen zu den Benutzergruppen finden sich im Abschnitt 2.2 des Pflichtenheftes.

1.2. Wunschkriterien

Beim Hochladen von Fotos soll automatisch eine verkleinerte Kopie erzeugt werden. Diese Kopie kann dann als Vorschau in eine Internetseite eingebunden werden.

Sollte während der Entwicklungsphase des Projektes Drupal 6.0 veröffentlicht werden, wird geprüft, inwiefern sich die entwickelten Module auch in die neue Version von Drupal integrieren lassen. Wenn die neuen Module ohne weitere Anpassungen auch unter Drupal 6.0 lauffähig sind, wird der Wechsel von Version 5.3 zu 6.0 vollzogen.

1.3. Abgrenzungskriterien

Die Oberfläche der Internetpräsenz kann mit Drupal alleine nicht angepasst werden, da dazu das Theme der Internetpräsenz manuell geändert werden muss.

Die Verknüpfung der Google Map mit Flickr und YouTube setzt voraus, dass Schlüsselwörter zu den betreffenden Orten angegeben wurden. Es erfolgt keine Verknüpfung über die Geo-Position.

2. Produkteinsatz

2.1. Anwendungsbereiche

Das fertig implementierte und konfigurierte Content-Management-System eignet sich hervorragend zur Verwaltung einer Internetpräsenz für Nationalparks, Naturparks, Biosphärenreservate und Touristenregionen.

2.2. Zielgruppen

Die erstellte Internetpräsenz ist frei im Internet verfügbar und kann von jedem Besucher eingesehen werden. Durch die Prinzipien des Web 2.0 werden Besucher dazu ermuntert, selbst Inhalte zu Erstellen und somit zum Produzenten von Inhalten zu werden. Um dies zu ermöglichen, müssen im Content-Management-System klare Rollen definiert werden.

2.2.1. Unregistrierte Benutzer

Alle Besucher der Internetpräsenz, die kein eigenes Benutzerkonto besitzen, fallen in die Gruppe der unregistrierten Besucher. Prinzipiell gehören zu dieser Gruppe also alle Personen, die die Internetpräsenz besuchen und Informationen über den Nationalpark sammeln möchten. Dazu gehört auch das Suchen nach Ausflugszielen, Übernachtungsmöglichkeiten, Rad- und Wanderwegen sowie das Durchsuchen der Tier- und Pflanzendatenbank. Diese Gruppe kann alle publizierten Inhalte der Internetpräsenz einsehen, jedoch keine eigenen Inhalte erzeugen oder Kommentare abgeben.

Die Gruppe der unregistrierten Benutzer kann sich über ein Formular auf der Internetpräsenz registrieren lassen und bekommt damit die erweiterten Rechte eines registrierten Benutzers. Die Verarbeitung der Registrierung verläuft voll automatisch und bedarf keiner Bestätigung durch den Systemadministrator.

2.2.2. Registrierte Benutzer

Jeder Benutzer der Internetpräsenz, der sich in einem Online-Formular registrieren lässt, kommt automatisch in die Gruppe der registrierten Benutzer. Diese Benutzergruppe kann nicht nur alle Inhalte einsehen, sondern ganz im Sinne des Web 2.0 auch eigene Inhalte publizieren. Dazu gehört das Schreiben von Foreneinträgen, das Verfassen von Kommentaren, das Publizieren von Naturbeobachtungen und das Hochladen selbstgemachter Bilder.

Der Systemadministrator hat die Möglichkeit, registrierten Benutzern weitere Rechte zu ermöglichen. Dafür wurden 3 weitere Benutzergruppen geschaffen, die eine Untergruppe der registrierten Benutzer darstellen und für ganz konkrete Anwendungsfälle konzipiert wurden. Im Einzelnen sind dies folgende Benutzergruppen:

- Unterkunftsanbieter

Jeder Anbieter von Übernachtungsmöglichkeiten (privat oder geschäftlich) kann durch den Systemadministrator mit einem Benutzernamen und einem Passwort ausgestattet werden. Damit kann in der Verwaltungsoberfläche eine Beschreibung der eigenen Unterkunft erstellt werden, welche dann auf der Internetpräsenz für jeden Besucher einsehbar ist. Außerdem können Buchungsanfragen verwaltet werden, um auf der Internetpräsenz immer die tatsächlich verfügbaren Quartiere anzuzeigen. Die Beschreibung der Unterkunft kann mit Bildern aufgelockert werden und es ist möglich, die Unterkunft auf einer Google Map zu publizieren. Jeder Unterkunftsanbieter hat nur Zugriff auf die von ihm erstellten Unterkünfte und sollte den Buchungsstatus in regelmäßigen Abständen aktualisieren.

- Veranstaltungsmanager

Auch die Benutzer aus der Gruppe der Veranstaltungsmanager müssen durch den Systemadministrator erstellt werden. Sie können Veranstaltungen auf einem Kalender eintragen und mit Bildern unterlegen. Diese Veranstaltung ist dann für jeden Besucher der Internetpräsenz auf einem Kalender sichtbar. Außerdem kann der Veranstaltungsmanager, wie jeder andere registrierte Benutzer auch, Inhalte kommentieren sowie im Forum aktiv werden. Darüber hinaus besitzt der Veranstaltungsmanager aber keine Rechte.

- Nationalparkverwalter

Der Nationalparkverwalter hat weitreichende Befugnisse beim Verwalten der Internetpräsenz. Er kann Internetseiten erstellen, kann Diskussionsforen anlegen, kann Weblogs erstellen, kann die Kommentare verwalten, kann Rad- und Wanderwege erstellen sowie die Datenbank mit Pflanzen und Tieren des Nationalparks verwalten.

2.2.3. Systemadministrator

Eine zentrale Rolle bei der Installation und Konfiguration des Content-Management-Systems sowie bei der Überwachung der Benutzergruppen nimmt der Systemadministrator (auch Superadministrator genannt) ein. Er hat Zugriff auf alle Funktionen der Verwaltungsoberfläche und kann als Einziger neue Benutzer und Benutzergruppen anlegen. Benutzer können durch den Systemadministrator auch wieder gelöscht werden. Aus Sicherheitsgründen gibt es in Drupal immer nur genau einen Systemadministrator. Das Benutzerkonto des Systemadministrators sollte nicht dafür verwendet werden, Inhalte auf der Internetpräsenz zu erstellen, da gerade unerfahrene Benutzer mit diesem Zugang auch großen Schaden anrichten können.

2.3. Betriebsbedingungen

Das Content-Management-System wird auf einem zentralen Web-Server installiert und kann so von jedem Arbeitsplatzrechner mit Internetanschluss und Web-Browser aufgerufen werden. Eine extra Software-Installation auf dem Arbeitsplatzrechner ist nicht erforderlich. Der Web-Server sollte rund um die Uhr eingeschaltet sein, um eine ständige Verfügbarkeit des Systems zu gewährleisten.

Die Nutzung mit mobilen Endgeräten wie *PDA*s oder Smartphones ist nicht vorgesehen.

Beschreibung:

- 1 Internetpräsenz aufrufen
- 2 Über das Menu die gewünschte Internetseite aufrufen
- 3 Internetseite erscheint

Geschäftsprozess: Administer Content

Ziel: Inhalte jeglicher Art für die Internetpräsenz erstellen, bearbeiten oder löschen

Vorbedingung: Erfolgreich ins Content-Management-System eingeloggt

Nachbedingung Erfolg: Gewünschte Änderung wurde durchgeführt

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Internetseite wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

- 1 Formular zur Administrierung von Inhalten aufrufen
- 2 Formular zur Administrierung von Inhalten ausfüllen
- 3 Formular abschicken

Geschäftsprozess: Administer Comment

Ziel: Kommentare anlegen, ändern oder löschen

Vorbedingung: Erfolgreich ins Content-Management-System eingeloggt

Nachbedingung Erfolg: Der Kommentar ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen des Kommentars wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

- 1 Formular zur Administrierung von Kommentaren aufrufen
- 2 Formular zur Administrierung von Kommentaren ausfüllen
- 3 Formular abschicken

Geschäftsprozess: Post Comment

Ziel: Kommentar zu einer bestimmten Internetseite abgeben

Vorbedingung: Benutzerkonto zum Schreiben von Kommentaren muss vorhanden sein

Nachbedingung Erfolg: Der gewünschte Kommentar ist auf der Internetseite angelegt

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Registered Website Visitor, Event Administrator, Accommodation Provider, Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen des Events wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

- 1 Internetseite aufrufen
- 2 Zu kommentierenden Inhalt aufrufen
- 3 Kommentarformular aufrufen
- 4 Mit vorhandenem Benutzerkonto einloggen
- 5 Kommentarformular ausfüllen
- 6 Formular abschicken

Geschäftsprozess: Administer Event

Ziel: Veranstaltung im Kalender anlegen, ändern oder löschen

Vorbedingung: Erfolgreich ins Content-Management-System eingeloggt

Nachbedingung Erfolg: Die gewünschte Veranstaltung ist im Kalender angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Event Administrator

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Veranstaltung wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

- 1 Formular zur Administrierung von Veranstaltungen aufrufen
- 2 Formular zur Administrierung von Veranstaltungen ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Page

Ziel: Erstellen, Ändern oder Löschen von Internetseiten mit Texten und Bildern.

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die gewünschte Internetseite ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Internetseite wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zur Administrierung von Internetseiten aufrufen

2 Formular zur Administrierung von Internetseiten ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Weblog

Ziel: Erstellen, Ändern oder Löschen von Weblogs

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Das gewünschte Weblog ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen des Weblogs wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zur Administrierung der Weblogs aufrufen

2 Formular zur Administrierung der Weblogs ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Forum

Ziel: Erstellen, Ändern oder Löschen von Foren bzw. Foreneinträgen

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Das gewünschte Forum bzw. der gewünschte Forumseintrag ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zur Administrierung der Foren aufrufen

2 Formular zur Administrierung der Foren ausfüllen

3 Formular abschicken

Geschäftsprozess: Create Entry

Ziel: Erstellen eines Foreneintrags in ein bestehendes Forum

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Der gewünschte Forumseintrag ist angelegt und auf der Internetpräsenz sichtbar

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Registered Website Visitor, Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Internetseite wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Forum aufrufen

2 Formular zum Erstellen eines Foreneintrags aufrufen

3 Formular zum Erstellen eines Foreneintrags ausfüllen

4 Formular abschicken

Geschäftsprozess: Administer Accommodation

Ziel: Erstellen, Bearbeitung und Löschen von Unterkünften im Nationalpark. Diese Unterkünfte können auch auf einer Google Map dargestellt werden. Der Buchungsstatus der Unterkunft kann bearbeitet und

abgefragt werden.

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die Änderung ist auf der Internetpräsenz sichtbar

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Accommodation Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Unterkunft wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zur Administrierung der Unterkunft aufrufen

1 Formular zur Administrierung der Unterkunft ausfüllen

3 Formular abschicken

Geschäftsprozess: Book Accommodation

Ziel: Aus der Vielzahl der angebotenen Unterkünfte eine Unterkunft finden und eine Zimmerbuchung durchführen

Vorbedingung: Passende Unterkunft muss gefunden sein

Nachbedingung Erfolg: Die Buchungsanfrage wurde abgeschickt. Der Tourist bekommt dann per E-Mail eine Buchungsbestätigung.

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Anonymous Website Visitor, Registered Website Visitor, Event Officer, Park Officer, Accommodation Officer, System Administrator

Auslösendes Ereignis: Der Tourist möchte eine Übernachtungsmöglichkeit buchen

Beschreibung:

1 Suchen einer passenden Unterkunft

2 Verfügbarkeit des gewählten Zimmertyps prüfen

3 Kontaktdaten eingeben

4 Eingaben bestätigen

5 Buchungsbestätigung erhalten

Geschäftsprozess: Administer Image

Ziel: Hochladen und Löschen von Bildern auf den Web-Server sowie das Einbinden in eine Internetseite

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Das Bild wurde auf den Web-Server geladen bzw. gelöscht und in eine Internetseite eingebunden

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: In der Verwaltungsoberfläche wird die Seite des Image Moduls aufgerufen und die gewünschte Funktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Bildern aufrufen

2 Formular zum Administrieren von Bildern ausfüllen

3 Formular abschicken

Geschäftsprozess: Upload Image

Ziel: Hochladen von Bildern auf den Web-Server und Verknüpfung mit einem Knoten

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz. Auf dem Web-Server steht genug Speicherplatz zur Verfügung.

Nachbedingung Erfolg: Das Bild wurde auf den Web-Server geladen und kann nun für die unterschiedlichen Seiten der Internetpräsenz verwendet werden

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Registered Website Visitor, Hotel Officer, Park Officer, Hotel Officer

Auslösendes Ereignis: In der Verwaltungsoberfläche wird die Seite des Image Moduls aufgerufen und die gewünschte Funktion ausgewählt

Beschreibung:

1 Formular zum Hochladen von Bildern aufrufen

2 Formular zum Hochladen von Bildern ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Google Map

Ziel: Erstellen, Ändern oder Löschen von Internetseiten mit einer Google Map

Vorbedingung: Der Google Key muss eingestellt sein. Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz.

Nachbedingung Erfolg: Die gewünschte Google Map ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Google Map wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Google Maps aufrufen

2 Formular zum Administrieren von Google Maps ausfüllen

3 Formular abschicken

Geschäftsprozess: Create Map Marker

Ziel: Erstellen einer Markierung auf einer Google Map

Vorbedingung: Es muss eine fertig konfigurierte Google Map vorhanden sein. Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz.

Nachbedingung Erfolg: Die gewünschte Markierung ist angelegt

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Registered Website Visitor, Park Officer, Accommodation Provider

Auslösendes Ereignis: Es wird ein Knoten erstellt und bearbeitet und das Formular zur Erstellung einer Markierung angezeigt.

Beschreibung:

1 Formular zum erstellen bzw. bearbeiten eines Knotes aufrufen

2 Formular zum Erstellen einer Markierung aufrufen

3 Formular abschicken

Geschäftsprozess: Administer Map Path

Ziel: Erstellen, Ändern oder Löschen von Wegen auf einer Google Map

Vorbedingung: Es muss eine fertig konfigurierte Google Map vorhanden sein. Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz.

Nachbedingung Erfolg: Der gewünschte Weg ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen des Weges wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Wegen aufrufen

2 Formular zum Administrieren von Wegen ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Plant

Ziel: Erstellen, Ändern oder Löschen von Pflanzenbeschreibungen

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die gewünschte Pflanzenbeschreibung ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Pflanzenbeschreibung wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Pflanzenbeschreibungen aufrufen

2 Formular zum Administrieren von Pflanzenbeschreibungen ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Animal

Ziel: Erstellen, Ändern oder Löschen von Tierbeschreibungen.

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die gewünschte Tierbeschreibung ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Tierbeschreibung wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Tierbeschreibungen aufrufen

2 Formular zum Administrieren von Tierbeschreibungen ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Observation

Ziel: Erstellen, Ändern oder Löschen von Tier- oder Pflanzenbeobachtungen

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die Beobachtung ist angelegt, geändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Beobachtung wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Beobachtungen aufrufen

2 Formular zum Administrieren von Beobachtungen ausfüllen

3 Formular abschicken

Geschäftsprozess: Publish Observation

Ziel: Melden einer Tier- bzw. Pflanzenbeobachtung und Verlinkung auf einer Google Map

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz. Es muss eine Google Map erstellt worden sein.

Nachbedingung Erfolg: Die Beobachtung ist angelegt und auf einer Google Map sichtbar

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Registered Website Visitor, Park Officer

Auslösendes Ereignis: Ein gesichtetes Tier oder eine gesichtete Pflanze soll publiziert werden

Beschreibung:

1 Formular zum Erstellen von Tierbeobachtungen aufrufen

1 Formular zum Erstellen von Tierbeobachtungen ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer Taxonomy

Ziel: Erstellen, Ändern oder Löschen von Vokabeln um Inhalte zu Kategorisieren

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die gewünschte Kategorie ist angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: Park Officer

Auslösendes Ereignis: Die notwendigen Daten zum Erstellen, Ändern oder Löschen der Kategorien wurden eingegeben und die gewünschte Aktion ausgewählt

Beschreibung:

1 Formular zum Administrieren von Vokabeln aufrufen

2 Formular zum Administrieren von Vokabeln ausfüllen

3 Formular abschicken

Geschäftsprozess: Administer User

Ziel: Erstellen, Ändern oder Löschen von Benutzern für das Content-Management-System

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Der gewünschte Benutzer ist mit den definierten Rechten angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: System Administrator

Auslösendes Ereignis: Neuer Benutzer soll erstellt, modifiziert oder gelöscht werden

Beschreibung:

- 1 Formular zum Administrieren von Benutzer aufrufen
- 2 Formular zum Administrieren von Benutzer ausfüllen
- 3 Formular abschicken
- 4 Benutzer erhält Bestätigung per E-Mail

Geschäftsprozess: Administer Role

Ziel: Erstellen, Ändern oder Löschen von Benutzergruppen (Rollen) und die Vergabe von Rechten

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Die gewünschte Benutzergruppe ist mit den definierten Rechten angelegt, verändert oder gelöscht

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: System Administrator

Auslösendes Ereignis: Benutzergruppe und deren Rechte sollen erstellt, modifiziert oder gelöscht werden

Beschreibung:

- 1 Formular zum Administrieren von Benutzer aufrufen
- 2 Formular zum Administrieren von Benutzer ausfüllen
- 3 Formular abschicken
- 4 Benutzer erhält Bestätigung per E-Mail

Geschäftsprozess: Administer Module

Ziel: Installieren, Deinstallieren oder Konfigurieren von Modulen im Content-Management-System

Vorbedingung: Erfolgreicher Login in die Verwaltungsoberfläche der Internetpräsenz

Nachbedingung Erfolg: Das gewünschte Modul ist installiert, deinstalliert oder konfiguriert. Die zugehörigen Datenbank-Tabellen wurden angelegt bzw. gelöscht.

Nachbedingung Fehlschlag: Informationen über den Fehler werden angezeigt

Akteure: System Administrator

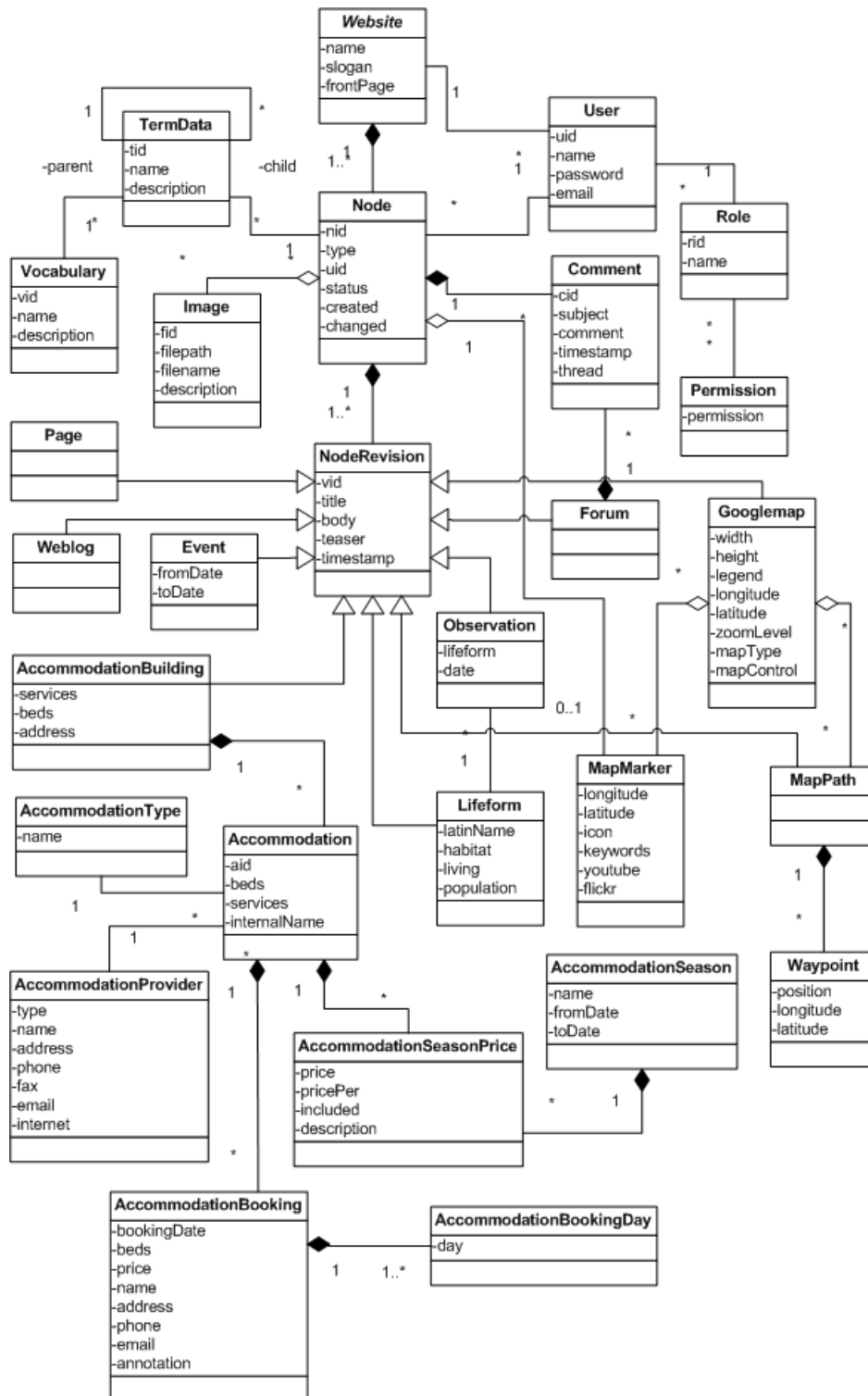
Auslösendes Ereignis: Die Funktionalität des Content-Management-System soll erweitert oder verkleinert werden

Beschreibung:

- 1 Formular zum Installieren/Deinstallieren von Modulen aufrufen
- 2 Module zur Installation oder Deinstallation auswählen
- 3 Formular abschicken
- 4 Formular zur Administration des Modules aufrufen
- 5 Formular zur Administration des Modules ausfüllen
- 6 Formular abschicken

5. Produktdaten

Abbildung A.2. Klassendiagramm



6. Produktleistungen

Bei durchschnittlicher Auslastung des Web-Servers kann mit einer 2 MBit-Internetverbindung jede Seite der Internetpräsenz, ausgenommen die Seiten mit einer Google Map, in weniger als 2 Sekunden komplett geladen

werden. Internetseiten mit einer Google Map können leicht eine Größe von mehreren hundert Kilobyte erreichen und benötigen bei einer 2 MBit Internetverbindung einige Sekunden zum vollständigen Laden.

Es gibt keine Funktion, die übermäßig viel Rechenleistung bzw. eine übermäßig starke Internetanbindung verlangt.

7. Qualitätsanforderungen

Tabelle A.1. Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		X		
Richtigkeit		X		
Interoperabilität			X	
Ordnungsmäßigkeit		X		
Sicherheit	X			
Zuverlässigkeit				
Reife	X			
Fehlertoleranz	X			
Wiederherstellbarkeit		X		
Benutzbarkeit				
Verständlichkeit	X			
Erlernbarkeit	X			
Bedienbarkeit	X			
Effizienz				
Zeitverhalten		X		
Verbrauchsverhalten			X	
Änderbarkeit				
Analysierbarkeit			X	
Modifizierbarkeit		X		
Stabilität	X			
Prüfbarkeit			X	
Übertragbarkeit				
Anpassbarkeit		X		
Installierbarkeit		X		
Konformität			X	
Austauschbarkeit			X	

8. Benutzeroberfläche

Neue Module für die Administration der Internetpräsenz werden in die bestehende Drupal-Verwaltungsoberfläche integriert und benutzen die gegebene Navigationsstruktur. Über Formularfelder können die gewünschten Änderungen an der Internetpräsenz durchgeführt werden. Fehlerhafte oder unvollständige Eingaben werden serverseitig abgefangen und automatisch farblich markiert.

Die Benutzeroberfläche der Internetpräsenz wird selbst gestaltet und soll ein hohes Maß an Übersichtlichkeit und Funktionalität bieten. Die Bedienung soll intuitiv und die Navigationsstruktur klar und verständlich sein. Alle

Navigationselemente werden mit Tooltips ausgestattet, welche beim Überfahren mit dem Mauszeiger nähere Informationen zu dem Verweis anzeigen. Die HTML-Seiten verwenden als Dokumenttyp XHTML 1.1 und werden mit dem Validator des W3C auf Richtigkeit überprüft.¹

Abbildung A.3. Entwurf der Benutzeroberfläche der Internetpräsenz



Folgende Elemente sind ständig auf der Internetpräsenz sichtbar:

1 - Schnellverweise

Diese Verweise führen zum Impressum, zur Registrierung, zum Login und zum Kontaktformular.

2 - Naturbanner

Dieser Bereich zeigt ein typisches Bild vom Nationalpark sowie das Logo des Nationalparks. Damit soll der Inhalt der Internetpräsenz für den Benutzer auf dem ersten Blick erschließbar sein sowie die Seite optisch aufgewertet werden.

3 - Breadcrumb-Navigation

Darunter versteht man eine Navigationsleiste, die anzeigt, wie tief man in die Internetpräsenz vorgedrungen ist. Damit lässt sich für den Benutzer leichter einordnen, an welcher Stelle er sich befindet.

4 - Suchfeld

Ein einfaches Eingabefeld, welches das Suchen nach Begriffen ermöglicht.

5 - Inhalt

Hier wird der eigentliche Inhalt der Internetseite angezeigt.

6 - Inhaltsblöcke

¹Der W3C Validator ist im Internet unter <http://www.validator.w3.org> zu finden.

Hier können weitere Informationen angezeigt werden. So ist vorgesehen, dort einen kleinen Kalender zu platzieren oder auch die neuesten Fotos oder Foreneinträge anzuzeigen.

7 - Primärer Navigationsbereich

In diesem Bereich finden sich alle Verweise zu den Hauptkategorien der Internetpräsenz. Auch die Unterkategorien werden dort angezeigt. Dieser Bereich ist somit essentiell zur Navigation auf der Internetpräsenz.

9. Nichtfunktionale Anforderungen

Keine

10. Technische Produktumgebung

Das Content-Management-System sowie die dazugehörige Internetpräsenz sind nur auf einem Web-Server mit Datenbankbindung lauffähig.

10.1. Software

Web-Server

Das Content-Management-System ist auf dem Apache HTTP Server sowie dem Microsoft IIS lauffähig. Es wird mindestens PHP 5.2 sowie eine Anbindung an eine mySQL-Datenbank benötigt. Aus Sicherheitsgründen sollte immer die neueste Version der jeweiligen Software verwendet werden.

Web-Client

Zum Betrachten der Internetpräsenz bzw. des Content-Management-Systems ist ein aktueller Web-Browser erforderlich. Empfohlen wird Firefox ab Version 2.0 oder Microsoft Internet Explorer ab Version 7.0. Eine fehlerfreie Darstellung unter älteren Browsern wird nicht garantiert. Ältere Browserversionen werden aus Sicherheitsgründen nicht empfohlen.

In jedem Fall muss JavaScript zur korrekten Darstellung der Google Maps und der Verwaltungsoberfläche aktiviert sein. Als Betriebssystem kann Microsoft Windows 2000/XP/Vista sowie Mac OS X verwendet werden.

10.2. Hardware

Web-Server

Das Content-Management-System ist auf jedem handelsüblichen Web-Server lauffähig. Auch ein sogenannter "Shared Server" bzw. "Virtual Server" ist ausreichend.

Web-Client

Die Internetpräsenz und die Verwaltungsoberfläche kann von jedem Arbeitsplatzrechner mit einem Internetzugang aufgerufen werden. Für einen flüssigen Seitenaufbau wird ein Computer mit einem Prozessor ab 700 MHz und 256 MB RAM sowie einer Internetanbindung von mindestens 2 MBit empfohlen.

10.3. Orgware

Web-Server und Web-Client müssen eine Verbindung zum Internet haben.

10.4. Produkt-Schnittstellen

Es sind keine besonderen Schnittstellen vorgesehen.

11. Spezielle Anforderungen an die Entwicklungs-Umgebung

Es gibt keine Abweichungen zu der Produktumgebung.

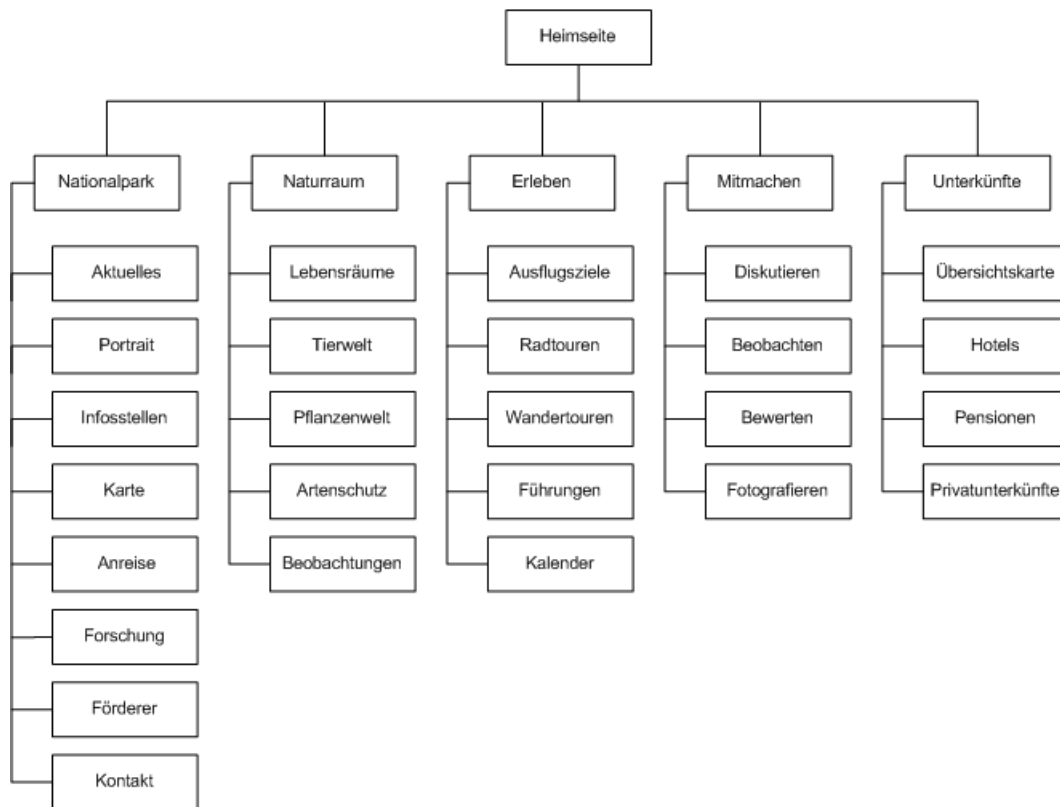
12. Gliederung in Teilprodukte

Es ist nicht geplant, verschiedene Teilprodukte zu erstellen.

13. Ergänzungen

Das Content-Management-System lässt sich für beliebige Nationalparks nutzen. Die entstehende Internetpräsenz wird jedoch mit Daten des Nationalparks "Unteres Odertal" gefüllt. Dazu ist die folgende Informationsstruktur vorgesehen.

Abbildung A.4. Informationsstruktur der Internetpräsenz



Anhang B. Grundgerüst eines Drupal-Modules

Jedes Drupal-Modul besteht aus einer Vielzahl von Hooks, welche die Benutzeroberfläche und das Verhalten des Modules bestimmen. Im Folgenden ist das Grundgerüst eines Modules mit dem Namen "node_example" abgebildet, welches direkt von ¹ übernommen wurde. Es diene als Vorlage bei der Erstellung der neuen Module.

```
// $Id: node_example.module,v 1.21 2007/01/29 17:27:13 bdragon Exp $

/**
 * @file
 * This is an example outlining how a module can be used to define a new
 * node type.
 *
 * Our example node type will allow users to specify a "color" and a "quantity"
 * for their nodes; some kind of rudimentary inventory-tracking system, perhaps?
 * To store this extra information, we need an auxiliary database table.
 *
 * Database definition:
 * @code
 * CREATE TABLE node_example (
 *   vid int(10) unsigned NOT NULL default '0',
 *   nid int(10) unsigned NOT NULL default '0',
 *   color varchar(255) NOT NULL default '',
 *   quantity int(10) unsigned NOT NULL default '0',
 *   PRIMARY KEY (vid, nid),
 *   KEY `node_example_nid` (nid)
 * )
 * @endcode
 */

/**
 * Implementation of hook_node_info(). This function replaces hook_node_name()
 * and hook_node_types() from 4.6. Drupal 5 expands this hook significantly.
 *
 * This is a required node hook. This function describes the nodes provided by
 * this module.
 *
 * The "name" value provide a human readable name for the node,
 * while the "module" value tells Drupal how the module's functions map to hooks
 * (i.e. if the module is node_example_foo then node_example_foo_insert will be
 * called when inserting the node). The "description" value provides a brief
 * description of the node type, which will show up when a user accesses the
 * "create content" page for that node type. Other attributes can also be
 * defined through this hook, but only these ones are required.
 */
function node_example_node_info() {
  return array(
    'node_example' => array(
      'name' => t('example node'),
      'module' => 'node_example',
      'description' => t("This is an example node type with a few fields."),
    )
  );
}

/**
 * Implementation of hook_access().
 *
 * Node modules may implement node_access() to determine the operations
 * users may perform on nodes. This example uses a very common access pattern.
 */
function node_example_access($op, $node) {
```

¹http://api.drupal.org/api/file/developer/examples/node_example.module/5/source

```

global $user;

if ($op == 'create') {
  // Only users with permission to do so may create this node type.
  return user_access('create example node');
}

// Users who create a node may edit or delete it later, assuming they have the
// necessary permissions.
if ($op == 'update' || $op == 'delete') {
  if (user_access('edit own example nodes') && ($user->uid == $node->uid)) {
    return TRUE;
  }
}
}

/**
 * Implementation of hook_perm().
 *
 * Since we are limiting the ability to create new nodes to certain users,
 * we need to define what those permissions are here. We also define a permission
 * to allow users to edit the nodes they created.
 */
function node_example_perm() {
  return array('create example node', 'edit own example nodes');
}

/**
 * Implementation of hook_form().
 *
 * Now it's time to describe the form for collecting the information
 * specific to this node type. This hook requires us to return an array with
 * a sub array containing information for each element in the form.
 */
function node_example_form(&$node) {
  $type = node_get_types('type', $node);

  // We need to define form elements for the node's title and body.
  $form['title'] = array(
    '#type' => 'textfield',
    '#title' => check_plain($type->title_label),
    '#required' => TRUE,
    '#default_value' => $node->title,
    '#weight' => -5
  );
  // We want the body and filter elements to be adjacent. We could try doing
  // this by setting their weights, but another module might add elements to the
  // form with the same weights and end up between ours. By putting them into a
  // sub-array together, we're able force them to be rendered together.
  $form['body_filter']['body'] = array(
    '#type' => 'textarea',
    '#title' => check_plain($type->body_label),
    '#default_value' => $node->body,
    '#required' => FALSE
  );
  $form['body_filter']['filter'] = filter_form($node->format);

  // Now we define the form elements specific to our node type.
  $form['color'] = array(
    '#type' => 'textfield',
    '#title' => t('Color'),
    '#default_value' => $node->color
  );
  $form['quantity'] = array(
    '#type' => 'textfield',
    '#title' => t('Quantity'),
    '#default_value' => $node->quantity,
    '#size' => 10,
    '#maxlength' => 10
  );
};

```

```

return $form;
}

/**
 * Implementation of hook_validate().
 *
 * Our "quantity" field requires a number to be entered. This hook lets
 * us ensure that the user entered an appropriate value before we try
 * inserting anything into the database.
 *
 * Errors should be signaled with form_set_error().
 */
function node_example_validate(&$node) {
  if ($node->quantity) {
    if (!is_numeric($node->quantity)) {
      form_set_error('quantity', t('The quantity must be a number.'));
    }
  }
  else {
    // Let an empty field mean "zero."
    $node->quantity = 0;
  }
}

/**
 * Implementation of hook_insert().
 *
 * As a new node is being inserted into the database, we need to do our own
 * database inserts.
 */
function node_example_insert($node) {
  db_query("INSERT INTO {node_example} (vid, nid, color, quantity) VALUES (%d, %d, '%s', %d)", $node->vid, $node->nid, $node->color, $node->quantity);
}

/**
 * Implementation of hook_update().
 *
 * As an existing node is being updated in the database, we need to do our own
 * database updates.
 */
function node_example_update($node) {
  // if this is a new node or we're adding a new revision,
  if ($node->revision) {
    node_example_insert($node);
  }
  else {
    db_query("UPDATE {node_example} SET color = '%s', quantity = %d WHERE vid = %d", $node->color, $node->quantity, $node->vid);
  }
}

/**
 * Implementation of hook_nodeapi().
 *
 * When a node revision is deleted, we need to remove the corresponding record
 * from our table. The only way to handle revision deletion is by implementing
 * hook_nodeapi().
 */
function node_example_nodeapi(&$node, $op, $teaser, $page) {
  switch ($op) {
    case 'delete revision':
      // Notice that we're matching a single revision based on the node's vid.
      db_query('DELETE FROM {node_example} WHERE vid = %d', $node->vid);
      break;
  }
}

```

```

* Implementation of hook_delete().
*
* When a node is deleted, we need to remove all related records from our table.
*/
function node_example_delete($node) {
  // Notice that we're matching all revisions, by using the node's nid.
  db_query('DELETE FROM {node_example} WHERE nid = %d', $node->nid);
}

/**
 * Implementation of hook_load().
 *
 * Now that we've defined how to manage the node data in the database, we
 * need to tell Drupal how to get the node back out. This hook is called
 * every time a node is loaded, and allows us to do some loading of our own.
 */
function node_example_load($node) {
  $additions = db_fetch_object(db_query('SELECT color, quantity FROM {node_example} WHERE
vid = %d', $node->vid));
  return $additions;
}

/**
 * Implementation of hook_view().
 *
 * This is a typical implementation that simply runs the node text through
 * the output filters.
 */
function node_example_view($node, $teaser = FALSE, $page = FALSE) {
  $node = node_prepare($node, $teaser);
  $node->content['myfield'] = array(
    '#value' => theme('node_example_order_info', $node),
    '#weight' => 1,
  );

  return $node;
}

/**
 * A custom theme function.
 *
 * By using this function to format our node-specific information, themes
 * can override this presentation if they wish. We also wrap the default
 * presentation in a CSS class that is prefixed by the module name. This
 * way, style sheets can modify the output without requiring theme code.
 */
function theme_node_example_order_info($node) {
  $output = '<div class="node_example_order_info">';
  $output .= t('The order is for %quantity %color items.', array('%quantity' =>
check_plain($node->quantity), '%color' => check_plain($node->color)));
  $output .= '</div>';
  return $output;
}

```

Anhang C. Abkürzungen

Ajax	Asynchronous JavaScript and XML
API	Application Programming Interface
CGI	Common Gateway Interface
CMS	Content-Management-System
CSS	Cascading Style Sheets
DSL	Digital Subscriber Line
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
PDA	Personal Digital Assistant
PHP	PHP Hypertext Preprocessor
RAM	Random Access Memory
RDF	Resource Description Framework
REST	Representational State Transfer
RIA	Rich Internet Applications
RSS	Really Simple Syndication
SOAP	Früher: Simple Object Access Protocol
UGC	User Generated Content
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WCMS	Web-Content-Management-System
WWW	World Wide Web
WYSIWYG	What you see is what you get
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XML-RPC	Extensible Markup Language Remote Procedure Call

Anhang D. Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit selbständig und ohne unerlaubte Hilfe angefertigt habe, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Cottbus, den _____ Unterschrift: _____

Literaturverzeichnis

- [Alb07] Tom Alby. Copyright © 2007 Carl Hanser Verlag. 3-4464-0931-9. *Web 2.0. Konzepte, Anwendungen, Technologien.*
- [And07] Chris Anderson. Copyright © 2007 Hanser Wirtschaft. 3-4464-0990-4. *The Long Tail - Der lange Schwanz. Nischenprodukte statt Massenmarkt – Das Geschäft der Zukunft.*
- [Bal01] Band 1. Helmut Balzert. Copyright © 2001 Spektrum Akademischer Verlag GmbH. 3-8274-0480-0. *Lehrbuch der Software-Technik. Software-Entwicklung.*
- [CR03] Alan Cooper und Robert Reimann. Copyright © 2003 John Wiley & Sons. 0-76452-641-3. *About Face 2.0. The Essentials of Interaction Design.*
- [DZT04] Deutsche Zentrale für Tourismus. Copyright © 2004 Deutsche Zentrale für Tourismus e. V. (DZT). http://www.deutschland-tourismus.de/pdf/marktforschung_04_d.pdf. *Incoming - Tourismus Deutschland. Zahlen - Fakten - Daten 2003.* Letzter Zugriff 17.10.07.
- [Egg05] Roman Egger. Copyright © 2005 Shaker Verlag. 3-8322-3663-5. *Grundlagen des eTourism. Informations- und Kommunikationstechnologien im Tourismus.*
- [Foc07] Copyright © 2007 FOCUS Online. http://www.focus.de/digital/internet/google/copyright-verletzungen_aid_50519.html. *Copyright-Verletzungen. Milliardenklage gegen Google.* Letzter Zugriff 11.11.07.
- [Gar05] Jesse James Garret. Copyright © 2005 Adaptive Path. <http://www.adaptivepath.com/ideas/essays/archives/000385.php>. *Ajax: A New Approach to Web Applications.* Letzter Zugriff 22.10.07.
- [GF07] Copyright © 2007 Media Perspektiven. http://www.ard-zdf-onlinestudie.de/fileadmin/Online07/Online07_Multimedia.pdf. *Das Mitmach-Netz im Breitbandzeitalter. Ergebnisse der ARD/ZDF Online-Studie 2007.* Letzter Zugriff 13.11.07.
- [GGA06] Justin Gehtland, Ben Galbraith und Dion Almaer. Copyright © 2006 Carl Hanser Verlag. 3-4464-0630-1. *Ajax. Eine pragmatische Einführung in Web 2.0.*
- [Gra06] Hagen Graf. Copyright © 2006 Addison-Wesley Verlag. 3-8273-2321-5. *Drupal. Community-Websites entwickeln und verwalten mit dem Open Source-CMS.*
- [HV03] Sven Heinsen und Petra Vogt. Copyright © 2003 Carl Hanser Verlag. 3-4464-0990-4. *Usability praktisch umsetzen. Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte.*
- [IBM06] Scott Laningham. Copyright © 2006 IBM DeveloperWorks Podcast. <http://www-128.ibm.com/developerworks/podcast/dwi/cm-int082206.txt>. *IBM DeveloperWorks Podcast 7-28-2006.*
- [Lyo01] Charles J. Lyonss. Copyright © 2001 Prentice Hall. 0-13-032161-3. *Essential Design for Web Professionals.*
- [Meu06] Richard Meusers. Copyright © 2006 Spiegel Online. <http://www.spiegel.de/netzwelt/web/0,1518,448340,00.html>. *Peinliche Pannen bringen StudiVZ in Verruf.*
- [Mey06] Eric A. Meyer. Copyright © 2006 O'Reilly Media Inc.. 0-596-52733-0. *CSS. The Definitve Guide.*
- [OOP07] Copyright © 2007 Drupal API. <http://api.drupal.org/?q=api/file/developer/topics/oop.html/5>. *Object-Oriented Programming in Drupal.* Letzter Zugriff 22.10.07.
- [Ore05] Tim O'Reilly. Copyright © 2005 O'Reilly. <http://www.oreilynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software.*
- [Pus07] Frank Puscher. Copyright © 2007 Heise Zeitschriften Verlag. c't 2/07. *Klarheit trotz Ajax. Web 2.0 benutzerfreundlich dosiert.*

- [Sch94] Walter Schertler. Copyright © 1994 Ueberreuter Wirtschaft. 3-8323-0005-8. *Tourismus als Informationsgeschäft*.
- [Sch03] Peter Schweizer. Copyright © 2003 Galileo Press GmbH. 3-89842-362-X. *Handbuch der Webgestaltung*. Eine konzentrierte Einführung in professionelles Webdesign.
- [Sch04] Thomas Schraitle. Copyright © 2004 Suse Press. 3-89990-078-2. *DocBook-XML*. Medienneutrales und plattformunabhängiges Publizieren.
- [Stu07] Copyright © studiVZ Ltd. http://static.ak.studivz.net/lp/svz_de/press/img/studiVZ-Faktenblatt.pdf. *studiVZ-Faktenblatt*. Letzter Zugriff 16.11.07.
- [W3S07] W3 Schools. Copyright © 2007 Refsnes Data. http://www.w3schools.com/browsers/browsers_stats.asp. *Browser Statistics*. What is the trend in browser usage?. Letzter Zugriff 13.11.07.
- [Web07] Copyright © 2007 Wikipedia. http://de.wikipedia.org/wiki/Web_Services. *Web Services*. Letzter Zugriff 28.11.07.
- [Wen07] Christian Wenz. Copyright © 2007 Galileo Press. 3-89842-859-1. *JavaScript & AJAX*. Das umfassende Handbuch.
- [Wol06] Gregor Wolkensteiner. <http://www.wi.wu-wien.ac.at/~koch/lehre/inf-sem-ss-06/wolkensteiner.pdf>. *Die Qualität von Wikipedia im Vergleich zu Traditionellen Enzyklopädien*. Letzter Zugriff 05.11.07.
- [ZTZ02] Oliver Zschau, Dennis Traub und Rik Zahradka. Copyright © 2002 Galileo Press GmbH. 3-89842-157-0. *Web Content Management*. Websites professionell planen und betreiben.