

Diplomarbeit

zur Erlangung des akademischen Grades
"Diplomingenieur (FH)"

Web 2.0 in der Praxis

Vor- und Nachteile der Web 2.0-Technologie im
Bereich der Webentwicklung

ausgeführt von

Markus Mayer, BSc
Schiffmühlenstr. 116/8/39
1220 Wien

Begutachter:

1. Begutachter: Dr. Georg Hüttenegger
2. Begutachter: Mag. Clemens Appl

Wien, 20.02.2008

Ausgeführt an der Fachhochschule Technikum Wien
Studiengang Wirtschaftsinformatik



Kurzfassung und Abstract

Die vorliegende Arbeit behandelt die Vor- und Nachteile von AJAX, welches als die revolutionäre Technologie des Web 2.0 angesehen wird. Dabei wird zuerst auf das Web 2.0 selbst eingegangen und erläutert, worum es sich bei diesem Begriff überhaupt handelt und was das Web 2.0 auszeichnet. Anschließend wird die AJAX-Technologie erklärt und mittels Beispielen deren Vor- und Nachteile diskutiert. Anhand von Umfragen und Studien wird weiters darauf eingegangen, wie das derzeitige Nutzungsverhalten von AJAX und Web 2.0 aussieht.

Das Ergebnis der Arbeit zeigt, dass Web 2.0 in Unternehmen derzeit noch wenig verwendet wird, viele Unternehmen planen jedoch einen Einsatz in den nächsten Jahren. Die Verwendung der AJAX-Technologie ist aber nicht immer sinnvoll und sollte vorher gut überlegt und die Vorteile gegenüber den Nachteilen abgewogen werden.

This thesis is about the advantages and disadvantages of AJAX which is seen as the revolutionary technology of Web 2.0. Firstly the focus will be on Web 2.0 itself, a definition of the term and what the distinguishing features of Web 2.0 are. Then the AJAX technology is explained and pros and cons are discussed with the aid of some examples. Using the results of surveys and studies the current utilization of AJAX and Web 2.0 is presented.

The conclusion of this work shows that Web 2.0 is not yet widely used in companies but many are planning its deployment in a few years. Using AJAX technology does not always make sense and a thorough consideration of benefits and drawbacks is essential.

Eidesstattliche Erklärung

„Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.“

Wien, Februar 2008

Danksagung

Hiermit möchte ich meinen Eltern danken, dass sie mir dieses Studium ermöglicht und mich dabei finanziell unterstützt haben. Danke für eure Geduld und dass ihr immer für mich da wart.

Ein weiterer Dank gebührt meinem Diplomarbeitsbetreuer Dr. Georg Hüttenegger und Firmenbetreuer Mag. Clemens Appl. Danke, dass Sie sich die Mühe gemacht und mich bei dieser Arbeit betreut und unterstützt haben!

Ich danke auch meinem besten Freund Tim Simpson, mit dem ich schon so viele Webprojekte durchführen durfte. Danke, dass du mich immer ermutigt hast. Danke auch an deine Frau Kathrin, dass sie Verständnis dafür hatte, wenn wir wieder einmal bis tief in die Nacht vor dem Computer gesessen sind.

Abschließend möchte ich noch allen Freunden in meiner christlichen Jugendgruppe danken. Danke, dass ihr mich im Gebet unterstützt habt.

Inhaltsverzeichnis

1. Problem- und Aufgabenstellung	3
2. Einleitung	4
3. Grundlagen	5
3.1. Die Entstehung des Web 2.0-Begriffes	5
3.2. Was ist Web 2.0?	5
3.3. Die Entwicklung zum Web 2.0.....	6
3.3.1. Höhere Downloadgeschwindigkeiten	6
3.3.2. Statisch versus dynamisch.....	7
3.3.3. Vom Desktop zum Webtop.....	7
3.4. Was zeichnet das Web 2.0 aus?	9
3.4.1. User Generated Content	9
3.4.2. Blogs.....	10
3.4.3. Wikis	11
3.4.4. RSS-Feeds	12
3.4.5. Perpetual Beta	14
3.4.6. Der Long Tail-Effekt	14
3.4.7. Folksonomy	16
3.4.8. Mashups	17
3.4.9. Der Web 2.0-Benutzer	19
3.5. AJAX – Entstehung und Definition	20
3.5.1. XHTML	20
3.5.2. CSS	21
3.5.3. DOM	21
3.5.4. XML	22
3.5.5. JavaScript	23
3.5.6. Traditionelle Anwendungen versus AJAX.....	23
3.5.7. Wie funktioniert AJAX?	25
3.5.8. Wo funktioniert AJAX?	26
3.6. Ein AJAX-Beispiel	26
3.6.1. Erzeugen des XMLHttpRequest-Objekts.....	27
3.6.2. Verbindung mit dem Webserver	27
3.6.3. Antwort des Webserverns verarbeiten	28
4. Praktischer Teil	31
4.1. Vorteile von AJAX	31
4.1.1. Geringerer Traffic bei der Datenübertragung.....	31
4.1.2. Verwendung von Standard-Technologien	32
4.1.3. Asynchrone Datenübertragung	32
4.1.4. Keine Verwendung eines Browser-Plugins.....	33
4.1.5. Fazit.....	33
4.2. Nachteile von AJAX	34
4.2.1. Abgeschaltetes JavaScript	34
4.2.2. Mangelnde Funktion der Zurück-Schaltfläche	35
4.2.3. Alte Webbrowser und XMLHttpRequest	36
4.2.4. Eingeschränkte Bookmark-Funktion.....	37

4.2.5. Anzeige von Aktivitätsindikatoren.....	38
4.2.6. Hohe Serverbelastung durch Event-Handler	39
4.2.7. Höherer Aufwand bei der Entwicklung	40
4.2.8. Barrierefreiheit	40
4.2.9. Suchmaschinenindizierung	43
4.3. Web 2.0 Akzeptanz-Umfrage	45
4.3.1. Frage 1: Haben Sie einen öffentlichen Webauftritt?	45
4.3.2. Frage 2: Wie sieht die künftige Entwicklung aus?	46
5. Diskussion.....	48
5.1. AJAX sinnvoll oder nicht?	48
5.2. Verwendung von Frameworks.....	49
5.3. Ausblick auf das Web 3.0.....	50
5.4. Fazit.....	52

Die Bezeichnungen sollen immer geschlechtsneutral verstanden werden.

1. Problem- und Aufgabenstellung

Web 2.0 ist eines der häufigst genannten Schlagworte, wenn es um Neuigkeiten im Internet geht. Dieser Begriff, welcher unweigerlich mit der AJAX-Technologie verbunden ist, soll das alte Web 1.0 ablösen und neue, interaktive Webanwendungen hervorbringen, welche Desktopanwendungen immer ähnlicher werden.

In der Hoffnung, diesen neuen Hype nicht zu verpassen, werden täglich tausende neue Webseiten veröffentlicht, welche AJAX als Grundtechnologie einsetzen. Aber AJAX ist kein Wundermittel und so wird diese Technologie oft unüberlegt verwendet. Denn was passiert, wenn der User in seinem Webbrowser JavaScript deaktiviert hat und er AJAX somit nicht mehr verwenden kann? Wie sieht es mit Suchmaschinenfreundlichkeit und Barrierefreiheit aus? Können die Vorteile dieser Technologie ihre Nachteile aufwiegen?

Weiters stellt sich die Frage, wie es in österreichischen Unternehmen aussieht. Wird Web 2.0 bereits eingesetzt bzw. planen die Unternehmen überhaupt dessen Verwendung? Diese Arbeit soll darüber Auskunft geben.

2. Einleitung

„Ajax ist sicherlich eine der spannendsten neuen Techniken der Web-Programmierung“, schreibt Walter (2008, S. 400) in seinem Kompendium der Web-Programmierung. Mit dieser Aussage beschreibt er einen neuen Internet-Hype, der im Jahr 2005 angebrochen ist. In diesem Jahr wurden der Begriff Web 2.0 und das Akronym AJAX erfunden. Seitdem haben sich Internetseiten wesentlich verändert und beide Begriffe sind aus dem Sprachgebrauch nicht mehr wegzudenken.

Spricht man heutzutage von Technologie im Web 2.0-Bereich, so wird dabei an Webservices, RSS und AJAX gedacht, wobei Letzteres als die Revolution des Internets gilt. Auch Walter (2008, S. 387) ist der Meinung, dass AJAX eine der populärsten Techniken in der Webentwicklung des 21. Jahrhunderts ist. In dieser Arbeit wird daher hauptsächlich auf AJAX eingegangen, da sie als die Web 2.0-Technologie angesehen wird.

Der Autor dieser Arbeit beschäftigt sich seit über sechs Jahren mit der Erstellung von Webseiten und Webanwendungen, unter anderem mit der Implementierung von Content Management Systemen (CMS) und kann eine Reihe von Referenzen in diesem Bereich aufweisen (Mayer, 2008). Um die Entwicklung des Web 2.0 nicht zu verschlafen, hat sich der Autor intensiv mit der neuen AJAX-Technologie auseinandergesetzt und will mit dieser Arbeit die Allgemeinheit auf deren Besonderheiten aufmerksam machen.

Am Anfang der Arbeit wird die Entstehung des Web 2.0-Begriffes beschrieben und die Funktionsweise von AJAX erläutert. Zu diesem Zweck wird ein AJAX-Beispiel erstellt und dessen Quellcode und Funktionalität werden erklärt. Weiters wird darauf eingegangen, was das Web 2.0 auszeichnet und wodurch es sich vom Web 1.0 unterscheidet.

Anschließend werden anhand von Beispielen die Vor- und Nachteile der AJAX-Technologie betrachtet und dem Webentwickler Tipps für die Erstellung von Web 2.0-Seiten mit auf den Weg geben. Auf die Frage, ob AJAX noch bei ausgeschaltetem JavaScript funktioniert, wird ebenso eingegangen wie auf die Verträglichkeit der AJAX-Technologie mit Suchmaschinen und Barrierefreiheit.

Anhand einer Umfrage bei österreichischen Unternehmen wird weiters die Frage geklärt, ob die Web 2.0-Technologie bereits im Alltag eingesetzt wird und wie die künftige Entwicklung auf diesem Gebiet aussieht.

Abschließend erfolgt eine Reflexion über die gewonnenen Erkenntnisse und ein Ausblick auf das Web 3.0.

3. Grundlagen

In diesem Kapitel werden die Grundlagen gelegt, um ein Verständnis für das Diplomarbeitsthema zu bekommen. Anfangs wird die Entstehung und Entwicklung von Web 2.0 erläutert, danach wird anhand von Beispielen diskutiert, was das Web 2.0 auszeichnet und was seine Besonderheiten sind. Im Anschluss wird auf die damit verbundene AJAX-Technologie eingegangen.

3.1. Die Entstehung des Web 2.0-Begriffes

Der Begriff Web 2.0 entstand während eines Brainstormings für eine Konferenz, welche der O'Reilly-Verlag und die Firma MediaLive International im Herbst 2004 veranstalteten. Dale Dougherty und Tim O'Reilly kamen dabei auf die Idee, das World Wide Web (WWW) wie eine Software zu betrachten (Kienitz, 2007, S. 12). Dabei erstellten sie eine Liste von Webseiten und Internetbegriffen und ordneten diese dem Begriff Web 1.0 und Web 2.0 zu.

So wurden zum Beispiel DoubleClick, Britannica Online, persönliche Webseiten und Content Management Systeme dem Web 1.0 zugeordnet, Google AdSense, Wikipedia, Blogs und Wikis hingegen dem Web 2.0.

Der O'Reilly Verlag und MediaLive veranstalteten anschließend im Oktober 2004 die Konferenz für neue Techniken und Trends im Web unter dem Namen „Web 2.0 Konferenz“. Der Begriff Web 2.0 etablierte sich somit als Schlagwort für neue Technologien, Trends und Webseiten.

3.2. Was ist Web 2.0?

Tim Berners-Lee, welcher als Begründer des WWW gilt, äußert sich dazu in einem IBM Developer Works Interview folgendermaßen: „*I think Web 2.0 is of course a piece of jargon, nobody even knows what it means*“ (IBM, 2006). Damit behauptet er, dass keiner genau wissen könne, was Web 2.0 eigentlich bedeutet und in einer gewissen Weise hat er damit auch recht. Befragt man Webentwickler und Web-Gurus, was sie unter Web 2.0 verstehen, so bekommt man jedes Mal eine andere Antwort.

Auch das Unternehmen eResult, welches im Dezember 2007 eine Umfrage (eResult, 2007b) zum Thema „Web 2.0- und E-Commerce-Begriffe“ durchführte, kam zu dem Ergebnis, dass sich nur ein Fünftel der 400 befragten Personen etwas unter dem Begriff Web 2.0 vorstellen konnten. (Umfrageergebnis siehe Abb. 27)

Da Web 2.0 anfangs nur ein Schlagwort war und es auch noch keine Definition dafür gab, veröffentlichte Tim O'Reilly circa ein Jahr nach der Web 2.0-Konferenz am 30.09.2005 einen Artikel mit dem Titel „*What is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software*“ (O'Reilly, 2005). In diesem Artikel beschreibt Tim O'Reilly, was er unter dem Begriff Web 2.0 versteht und zeigt dabei einige Beispiele für Web 2.0 auf.

Im Folgenden findet sich eine Zusammenfassung davon, was Tim O'Reilly als Kernkompetenzen des Web 2.0 sieht (Holz, 2006):

- Dienste, keine Paketsoftware, mit kosteneffizienter Skalierbarkeit
- Kontrolle über einzigartige, schwer nachzubildende Datenquellen, deren Wert proportional zur Nutzungshäufigkeit steigt
- Vertrauen in Anwender als Mitentwickler
- Nutzung kollektiver Intelligenz
- Erreichen des "Long Tail" mittels Bildung von Communities etc.
- Erstellung von Software über die Grenzen einzelner Geräte hinaus
- Leichtgewichtige User Interfaces, Entwicklungs- und Geschäftsmodelle

Trotz dieser Zusammenfassung ist es schwer, eine einheitliche Definition für das Web 2.0 zu finden. Aus diesem Grund veröffentlichte Tim O'Reilly am 12.10.2006 nochmals einen Artikel, um den Begriff Web 2.0 genauer zu definieren:

Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: Build applications that harness network effects to get better the more people use them (O'Reilly, 2006).

Grundlegend kann nun gesagt werden, dass das Web 2.0 eine Neuentwicklung des Internets hin zu einer Art Plattform beschreibt und hauptsächlich Webseiten mit Community-Charakter und User Generated Content (siehe Kapitel 3.4.1 - User Generated Content) diesen Titel tragen. Die kollektive Intelligenz, also das vernetzte Wissen der User, welche sich in Web 2.0-Seiten (wie zum Beispiel Wikipedia) widerspiegelt, ist weiters ein Grundbestandteil dieser neuen Entwicklung. Technologien wie RSS, AJAX und Webservices unterstützen dabei diesen Trend und bieten dem User neue Möglichkeiten, das Web 2.0 zu nutzen.

3.3. Die Entwicklung zum Web 2.0

Das Web 2.0 hat sich nach dem Platzen der Dotcom-Blase allmählich zu entwickeln begonnen. Höhere Downloadgeschwindigkeiten, dynamische Programmiersprachen und die Entwicklung von interaktiven, desktopähnlichen Webanwendungen wie zum Beispiel Outlook Web Access haben zu dessen Entwicklung positiv beigetragen.

3.3.1. Höhere Downloadgeschwindigkeiten

Es sind nicht nur neue Technologien, die das Web 2.0 auszeichnen. Viele der Technologien, welche derzeit in Web 2.0-Seiten verwendet werden, gibt es schon seit einigen Jahren. So wurde JavaScript schon im Jahr 1995 entwickelt und die erste XML-Spezifikation erschien im Jahr 1998. Ein Grund, warum Web 2.0 jedoch erst in den letzten Jahren seinen Siegeszug begann, liegt sicherlich an der Bandbreitenbeschränkung der 90er Jahre. So war zu Beginn der WWW-Ära ein Internetbenutzer mit einer Geschwindigkeit von 9,6 bzw. 14,4 kbit/s unterwegs. Selbst eine ISDN-Leitung brachte es auf nur 64 bzw. 128 kbit/s, wenn beide Leitungen gekoppelt wurden.

Diese Beschränkung der Downloadgeschwindigkeit und die damals noch hohen Kosten für eine Internetverbindung verhinderten eine effiziente Nutzung und Entwicklung der Online-Dienste. So stellte ein Musik-Download eine große und langwierige Herausforderung dar und ein Video-Download war anfangs praktisch unmöglich. Erst schnellere Bandbreiten und neue Übertragungstechnologien wie DSL und ADSL brachten Ende der 90er Schwung in das Internet (siehe Abb. 1). Erst so konnten sich die ersten Vorreiter von Web 2.0 wie Napster, mp3.com, Ofoto u. Ä. etablieren.

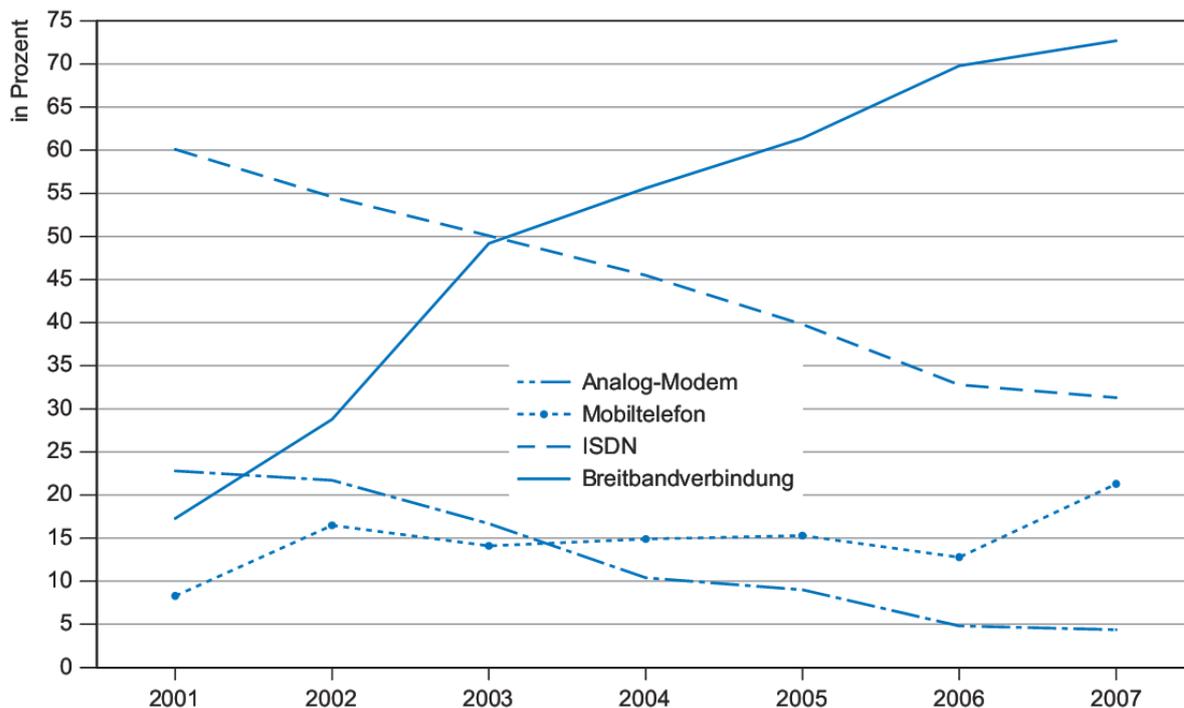


Abb. 1: Verbindungstechniken der Unternehmen für den Internetzugang 2001-2007 (Djahangiri & Edelhofer, 2007)

3.3.2. Statisch versus dynamisch

Am Anfang seiner Entwicklung bestand das WWW lediglich aus statischen Hypertext-Dokumenten, welche untereinander verlinkt waren (Kollmann & Hänsel, 2007, S. 22). Wurde eine Webseite aufgerufen, so lag diese fertig am Webserver und wurde so eins zu eins vom Webserver ausgeliefert und im Webbrowser angezeigt, bis eine andere Webseite aufgerufen wurde. Erst durch die Entwicklung von neuen Skriptsprachen wie zum Beispiel JavaScript und VBScript als clientseitige Skriptsprachen und Perl, PHP und ASP.NET als serverseitige Skriptsprachen konnten dynamische Webseiten generiert werden.

Diese zeichnen sich dadurch aus, dass ihr endgültiger Inhalt und ihre endgültige Form erst im Augenblick des Seitenaufufes entstehen (Krause, 2003). So liefert zum Beispiel eine Suchmaschine immer dynamische Webseiten aus, da das Ergebnis der Suchanfrage erst aus dem Datenbestand der Suchmaschine generiert wird. Noch dynamischer werden Webseiten durch die Verwendung der AJAX-Technologie, hier erfolgt die Datenübertragung zusätzlich asynchron. Dieses asynchrone Verhalten von AJAX wird in den nächsten Kapiteln noch genauer erläutert.

3.3.3. Vom Desktop zum Webtop

In den letzten Jahren bemerkt man eine immer stärker werdende Verlagerung der Softwareanwendungen vom Desktop in das Internet. So hatte der User früher seine Fotos fix auf der Festplatte gespeichert, heutzutage kann er diese kostenlos in das Internet laden, dort verschlagworten und anderen Usern zugänglich machen. Yahoos Flickr und Googles Picasa Web-Album bieten diese Möglichkeiten, eine Bildverwaltungssoftware ist somit nicht mehr

unbedingt nötig. Selbst die Bildbearbeitung kann mittlerweile online geschehen. So bietet beispielsweise nexImage (<http://demo.neximage.com>) bereits eine auf AJAX basierende Bildbearbeitungssoftware im Web an. Auch Adobe, der Marktführer im Bereich der professionellen Bildbearbeitung, kündigte im September 2007 eine Online-Version seines Flaggschiffes Photoshop an (Nack, 2007). Die erste Version von „Photoshop Express“ soll noch dieses Jahr im Web veröffentlicht werden.

Selbst das Office Paket von Microsoft bekommt bereits Konkurrenz aus dem Internet. So verzichten derzeit schon einige User auf das kostenpflichtige Microsoft Outlook und verwenden stattdessen das kostenlose Google Mail. Das zusätzliche Installieren eines Virenschanners und Spamfilters ist nicht mehr unbedingt nötig, da Google Mail diese Funktionen bereits integriert hat. Zoho Writer und ajaxWrite bieten ähnliche Funktionen wie Microsoft Word und mit Google Docs können verschiedene Personen gleichzeitig über das Internet an einer Text-Datei arbeiten.

Vorteile von webbasierten Office Anwendungen:

- Keine Lizenzkosten
- Mehrsprachige Benutzeroberfläche
- Kollaboratives Arbeiten (Arbeiten in Gruppen)
- Vergabe von speziellen Berechtigungen (auch Gruppenberechtigungen)
- Betriebssystemunabhängigkeit (es wird nur einen Webbrowser benötigt)
- Standortunabhängigkeit (weltweiter Zugang zu den eigenen Dokumenten)
- Backup der Dateien übernimmt der Anbieter
- Keine Updates erforderlich (siehe auch Kapitel 3.4.5 - Perpetual Beta)
- Keine Probleme mit Computerviren

Der Trend geht mittlerweile immer mehr in Richtung Webanwendungen. Eine Ablösung der Desktopanwendungen durch Anwendungen im Internet in naher Zukunft ist dadurch keine reine Utopie mehr. Auch werden Webanwendungen, die AJAX-basiert sind, vom Benutzer immer mehr wie Desktopapplikationen wahrgenommen (so auch Gratzner, 2006).

Desktopanwendung	Kostenlose Webanwendung
Microsoft Word	Zoho Writer, ajaxWrite, Google Docs
Microsoft Excel	Zoho Sheet, NumSum iRows, Google Spreadsheets
Microsoft Outlook	Google Mail, Zimbra
Microsoft Outlook Kalender	Kiko, Trumba
Microsoft MapPoint	Google Maps, Yahoo Local Maps
Windows Media Player	Deezer, Last.fm
ICQ, MSN-Messenger	Meebo
Photoshop, Gimp	nexImage, Snipshot, Picresize, Photoshop Express
Desktop-Simulationen	OOS, eyeOS, YouOS, PHP Explorer

Tabelle 1: Vergleich - Desktopanwendungen versus Webanwendungen

3.4. Was zeichnet das Web 2.0 aus?

Dass sich der Begriff Web 2.0 schwer definieren lässt, wurde bereits in Kapitel 3.2 erwähnt. Trotzdem gibt es einige Entwicklungen, Phänomene und Techniken die besonders im Web 2.0 vorkommen und mit ihm in Verbindung gebracht werden. Im Folgenden werden einige davon vorgestellt.

3.4.1. User Generated Content

Ein Erfolgskonzept von Web 2.0-Seiten ist die extrem starke Einbeziehung des Users. So sieht dies auch O'Reilly (2005) und schreibt, dass Netzwerk-Effekte durch Nutzerbeteiligung der Schlüssel zur Marktdominanz in der Web 2.0-Ära sind und trifft mit dieser Aussage den Nagel auf den Kopf. Webseiten wie Flickr, YouTube oder Wikipedia wären längst nicht so erfolgreich, würden die User nicht den Content dazu beitragen. Auf YouTube werden täglich Tausende neue Videos veröffentlicht und auch das deutschsprachige Wikipedia wächst um etwa 500 Artikel täglich (Zachte, 2007). Abb. 2 zeigt, welchen enormen Zuwachs diese Webseiten in den letzten drei Jahren bekommen haben.

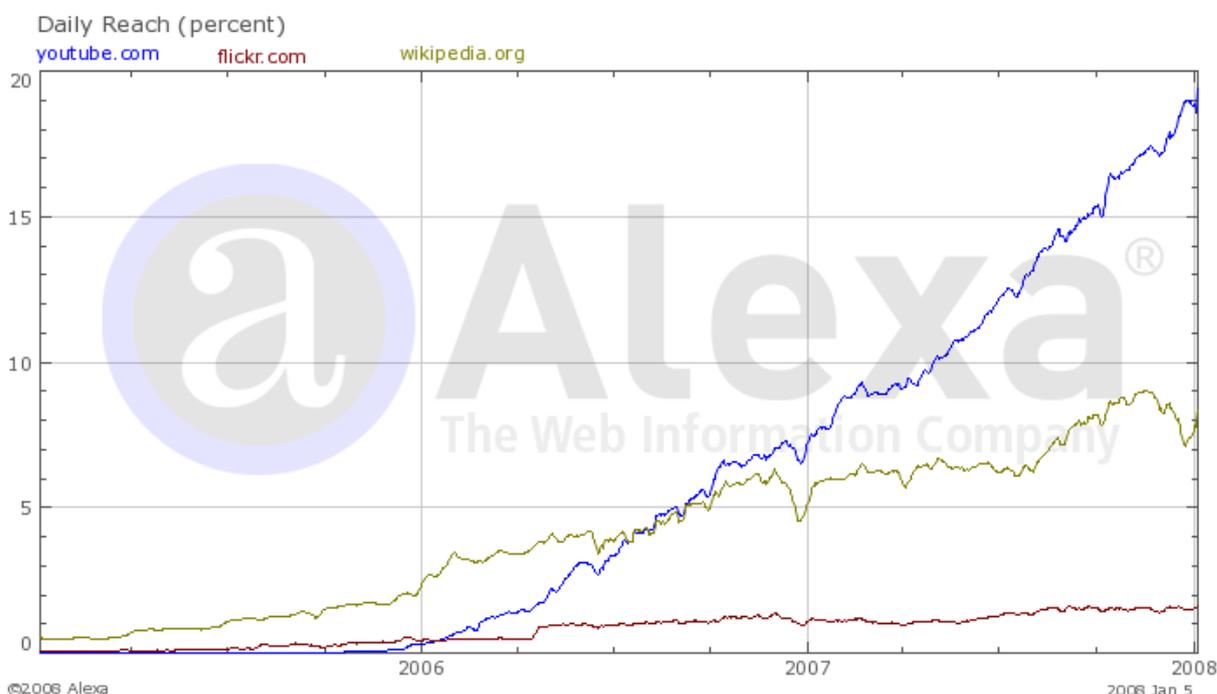


Abb. 2: Prozent der weltweiten Internet-User welche diese Webseiten besuchten

Der Grundgedanke des User Generated Content (UGC) ist dabei eigentlich ganz einfach. Der Betreiber eines UGC-Angebots stellt lediglich eine Idee und den notwendigen Speicherplatz zur Verfügung und hält ein funktionierendes Software-System im Web bereit. Den Rest besorgen eifrige Nutzer, die neue Communities rasch mit Leben und Inhalten füllen (Kienitz, 2007, S. 35).

Ganz unproblematisch sind solche Angebote jedoch nicht, es gibt immer wieder Probleme durch Rechtsverletzungen. So klagte das amerikanische Medienunternehmen Viacom, zu dem unter anderem MTV, Dreamworks und Paramount gehören, im März 2007 YouTube auf eine Milliarde US-Dollar Schadensersatz wegen Copyrightverletzungen (Wilkins, 2007). Auch das Videoportal Veoh.com, welches von Time Warner finanziert wird, blieb davon nicht

verschont und wurde im September 2007 von Universal Music aufgrund von Urheberrechtsverletzung verklagt (Briegleb, 2007).

Im deutschen und österreichischen Recht sieht die Gesetzeslage jedoch so aus, dass der Anbieter für den Content nicht strafrechtlich belangt werden kann, sofern er von der Urheberrechtsverletzung nichts wusste und nach Kenntnisnahme der Gesetzesübertretung den Content von seinem Angebot entfernt.

Zanger, Rechtsanwalt und gerichtlich beeideter Sachverständiger in Urheberrechtsfragen, definiert die Sachlage folgendermaßen: An sich haftet der Host Service Provider für die Urheberrechtsverletzung nicht. Sobald er allerdings vom rechtswidrigen Handeln Dritter Kenntnis erlangt, trifft ihn die Verpflichtung zur Löschung der entsprechenden Zugänge, sofern ihm dies technisch möglich ist. (Zanger, 2000, S. 53)

Auch Boßmanns belegt dies in ihrer Dissertation: Es bleibt festzuhalten, dass der Host-Provider im Regelfall nicht, sondern im Ausnahmefall nur dann strafrechtlich zur Verantwortung gezogen werden kann, wenn er einen besonders gefährlichen Dienst, zum Beispiel einen Filesharing-Dienst anbietet. Unterlässt er es in diesem Fall, stichprobenartige Kontrollmaßnahmen vorzunehmen, urheberrechtswidrige Datenübermittlungsvorgänge abubrechen bzw. die urheberrechtswidrig gespeicherten Daten zu löschen oder den Zugang zu diesen Daten zu sperren, so ist er nach Beihilferegeln zu bestrafen (Boßmanns, 2003, S. 148).

UGC-Anbieter sollten somit einerseits versuchen, Urheberrechtsverletzungen im Vorhinein zu unterbieten und, falls dies nicht möglich ist, im Falle einer Urheberrechtsverletzung den Content nach kurzer Prüfung auf Plausibilität der Rechtsverletzung aus dem Angebot entfernen (siehe dazu auch Ulbricht, 2007).

3.4.2. Blogs

Blog ist die Abkürzung von Weblog und bezeichnet eine Art Tagebuch oder Journal im Internet. Darin werden die unterschiedlichsten Artikel veröffentlicht, so gibt es private Blogs, wo Personen einfach aus ihrem Leben erzählen, politische Blogs, Urlaubsblogs, Technikblogs und Blogs zu den verschiedensten Themen. Die Beiträge werden von sogenannten Bloggern geschrieben. Der Begriff, welcher mittlerweile sogar im Duden vertreten ist, bezeichnet einfach Personen, die ihre Meinung in Blogs veröffentlichen. Das Veröffentlichen ist dabei ziemlich simpel und wird in Form eines vereinfachten Content Management Systems realisiert. User können sich bei ihrem Blog einloggen, Artikel schreiben, bearbeiten und mit Medien, wie zum Beispiel Bilder, Videos oder Audiodateien, versehen. Die veröffentlichten Inhalte werden dabei fast immer in chronologischer Reihenfolge angezeigt.

Für den User gibt es zwei Möglichkeiten, wie er einen eigenen Blog veröffentlichen kann. Entweder er richtet sich einen Account bei einem Blog-Provider ein oder er installiert eine Blog-Software (Blog-Publishing-System) auf einem eigenen Webserver. Mittlerweile gibt es unzählige Blog-Provider, wo sich der User kostenlos einen Blog einrichten kann. Bekannte Blog-Provider sind blogger.com, blogger.de und blog.de. Der Vorteil hierbei ist, dass der User kein technisches Hintergrundwissen benötigt und der erste Blog innerhalb kürzester Zeit erstellt werden kann. Blog-Publishing-System hingegen bieten den Vorteil, dass sie flexibler in der Anwendung sind, so kann zum Beispiel das Layout, anhand von Templates, selber angepasst werden. Der Nachteil ist jedoch, dass der User ein gewisses Know-how mitbringen muss, einen eignen Webspace und einige Zeit für das Einarbeiten benötigt. Die bekanntesten Blog-Publishing-Systeme sind:

- WordPress (<http://www.wordpress.de>)
- Serendipity (<http://www.s9y.org>)
- Movable Type (<http://www.movabletype.org>)
- ExpressionEngine (<http://www.expressionengine.com>)

Im Web 2.0 genießen Blogs eine weite Verbreitung, so bieten fast alle großen Player wie zum Beispiel Microsoft, Google oder IBM einen Firmen- oder Mitarbeiterblog an.

Typisch für Blogs sind auch die im Web 2.0 verbreiteten Elemente wie RSS-Feeds, Trackbacks, Permalinks, Tag clouds oder die Userbeteiligung in Form von Kommentaren, welche auch in dieser Arbeit teilweise betrachtet werden.

3.4.3. Wikis

Im Gegensatz zu Blogs, wo ein User die Artikel veröffentlicht und die anderen User die Beiträge nur lesen können, bieten Wikis die Möglichkeit, dass jeder User seinen Beitrag zu einem Artikel leisten kann, indem die Artikel online durch die User geändert werden können. So besteht die Möglichkeit, dass ein Artikel hinsichtlich Qualität und Quantität wächst, da sich mehrere Autoren daran beteiligen.

Das Erfolgskonzept wurde von Ward Cunningham im Jahr 1994 entwickelt. Das Wort Wiki, welches auf hawaiianisch „schnell“ bedeutet, hat sich Cunningham auf Hawaii von den dortigen Shuttlebussen abgeschaut, welche die Bezeichnung Wiki Wiki also „schnell schnell“ oder „sehr schnell“ trugen. Dass dieses Konzept funktioniert, zeigt Wikipedia seit einigen Jahren. Selbst die Qualität und Genauigkeit des Online-Lexikons sind hervorragend, was mehrfach durch Tests bestätigt wurde.

Laut einem aktuellen Test im Auftrag der Zeitschrift stern schneidet das Mitmach-Lexikon, das von Internetnutzern gestaltet wird, insgesamt weit besser ab als der von professionellen Redakteuren betreute Brockhaus [...] Wikipedia erreichte bei dem Test die Durchschnittsnote 1,7 und liegt damit um eine ganze Stufe besser als der Online-Brockhaus, der nur mit 2,7 bewertet wurde. (Zettel, 2007)

Möglich wird dies durch die kollektive Intelligenz und die Selbstregulierung der User. Postet ein User absichtlich falsche Informationen, so wird dies durch die große Anzahl an Usern innerhalb kürzester Zeit erkannt und bereinigt. Abb. 3 zeigt, wie schnell solche Änderungen durchgeführt werden, so wurde am 4. Februar 2008 um 00:59 Uhr der Artikel auf Wikipedia über den Musiker Falco mit einer offensichtlichen Falschmeldung erweitert. Schon nach nur einer Minute wurde dieser Vandalismus bemerkt und die vorige Version des Artikels wieder hergestellt.

Dies ist auf Wikipedia kein Einzelfall, sondern zeigt, dass die User innerhalb weniger Minuten auf Falschmeldungen reagieren und die Artikel bereinigen.

Abb. 3: Vandalismus beim Falco-Artikel auf Wikipedia (2008)

Wikis sind wie Blogs im Web 2.0 weit verbreitet. Eine Studie der Firma eResult (2007a) hat gezeigt, dass fast die Hälfte der befragten Teilnehmer Wikis im Web benutzen. Auch in Unternehmen werden Wikis gerne im Rahmen des Wissensmanagements eingesetzt. Zurawski (2007) unterscheidet dabei zwei Typen, die Unternehmens- bzw. Abteilungswikis und die projektbezogenen Wikis. Beide dienen dazu, das Wissen, welches im Unternehmen anfällt, sei es durch die Abwicklung von Projekten oder den alltäglichen Umgang mit Kunden, schriftlich festzuhalten. Dabei sorgen sie zusätzlich für eine höhere Transparenz im Unternehmen.

3.4.4. RSS-Feeds

RSS steht für Really Simple Syndication und bezeichnet die Möglichkeit Inhalte einer Webseite zu abonnieren. Dazu werden bestimmte Meta-Daten der Inhalte in einer RSS-Datei, welche vom Aufbau her einer XML-Datei entspricht, abgelegt. Der User kann nun diese Datei mit einem sogenannten RSS-Reader oder FeedReader abonnieren. Werden neue Inhalte veröffentlicht oder Inhalte verändert, wird der User darüber informiert. Dies geschieht, im Gegensatz zu einem Newsletter, welcher vom Webseitenbetreiber dem User zugeschickt wird, vom User aus. Technisch bedeutet dies, dass der FeedReader in bestimmten Abständen die RSS-Datei herunterlädt und auf Aktualisierungen überprüft. Hat sich die Datei verändert, zum Beispiel weil ein neuer Artikel veröffentlicht wurde, so wird dies dem User mitgeteilt. Dazu werden ihm meistens der Titel, eventuell ein Kurztext und der Link zu dem aktualisierten Inhalt mitgeteilt.

Als FeedReader können Webbrowser (Internet Explorer ab Version 7, Mozilla Firefox ab Version 1), Microsoft Outlook (ab Version 2007) oder eigene Reader-Programme wie zum Beispiel Omea Reader, Awasu oder RSS Bandit verwendet werden.

Der Vorteil von RSS ist, dass der User eine beliebige Anzahl an Webseiten beobachten kann und über Änderungen automatisch informiert wird. Er muss nicht jedes Mal die Webseiten ansurfen, um zu sehen, ob eine Änderung durchgeführt wurde. Ob eine Webseite abonniert werden kann, erkennt der User dabei an dem RSS-Logo (siehe Abb. 4).



Abb. 4: RSS-Logo

Wie eine RSS-Datei aussehen kann, zeigt Listing 1. Eingeleitet wird die Datei mit einem Statement, welches darauf hinweist, dass es sich im Grunde um eine XML-Datei handelt. Danach wird angegeben, welche RSS-Version verwendet wird. Anschließend wird der Channel-Bereich definiert, welcher allgemeine Angaben, wie zum Beispiel den Herausgeber der Datei oder Sprachdefinitionen, enthält. Im Anschluss an den Channel-Bereich können beliebig viele Item-Elemente angeführt werden, welche den eigentlichen Inhalt wie zum Beispiel Titel, Link und Beschreibung enthalten.

```

01. <?xml version="1.0" encoding="ISO-8859-1" ?>
02. <rss version="2.0">
03.   <channel>
04.     <title>ZDNet.de News</title>
05.     <link>http://www.zdnet.de/news/</link>
06.     <description>ZDNet.de - Alle Seiten der IT</description>
07.     <language>de-de</language>
08.     <item>
09.       <title>LG bringt 1650 Gramm schweres 13-Zoll-Notebook</title>
10.      <link>http://www.zdnet.de/news/hardware/0,39023109,39186605,00.htm
11.       <description>LG hat ein 13,3-Zoll-Notebook vorgestellt, das
12.       nach Herstellerangaben mit einem Gewicht von 1650 Gramm zu den
13.       leichtesten seiner Klasse zählt.</description>
14.       <pubDate>Thu, 14 Feb 2008 16:27:00 +0100</pubDate>
15.     </item>
16.     <item>
17.       <title>Amazon testet neues Werbesystem</title>
18.       <link>http://www.zdnet.de/news/tkomm/0,39023151,39186607,00.htm
19.       <description>Wie The Sydney Morning Herald berichtet, wird
20.       Amazon schon bald ein neues Bezahl-Werbesystem testen. Auf seinen
21.       Webseiten will der Online-Händler künftig Werbeeinblendungen von
22.       Produkten anderer Verkäufer schalten. </description>
23.       <pubDate>Thu, 14 Feb 2008 15:59:00 +0100</pubDate>
24.     </item>
25.   </channel>
26. </rss>

```

Listing 1: Beispiel einer RSS-Datei von zdnet.de

RSS hat sich im Web 2.0 vor allem durch Blogs etabliert und findet mittlerweile auch in Video- und Bildportalen seine Anwendung. So kann der User bestimmte Themen oder Suchbegriffe abonnieren und wird anschließend automatisch informiert, wenn neues Material zu diesem Suchbegriff oder Thema upgeloadet wurde.

3.4.5. Perpetual Beta

Perpetual Beta, also ein immerwährender Beta-Zustand, ist ein weiteres Schlagwort, welches Web 2.0-Seiten auszeichnet, und beschreibt Webseiten, welche regelmäßig weiterentwickelt und aktualisiert werden. Dieser Beta-Zustand ist dabei keineswegs, so wie in der Softwareentwicklung, als negativ oder unfertig zu verstehen. Webseiten in einem Perpetual Beta Zustand stehen dabei in ständiger Weiterentwicklung und veröffentlichen teilweise sogar täglich neue Features. Aufgrund der hohen Community-Beteiligung (wie zum Beispiel bei Flickr) können solche Features schon einige Stunden nach ihrer Veröffentlichung analysiert und die Akzeptanz der User beurteilt werden. Wird das Feature von den Usern nicht gut aufgenommen und wenig genutzt, kann es noch am selben Tag wieder verschwinden. Der User wird so, ohne es zu wissen, zu einem Black-Box Beta Tester.

Musser und O'Reilly (2006) vergleichen in ihrem Buch „*Web 2.0 Principles and Best Practices*“ den Unterschied zwischen den Veröffentlichungszyklen von Microsoft und Flickr (siehe Abb. 5) und kommen dabei zu der Erkenntnis, dass ein frühes und schnelles Veröffentlichen ganz im Sinne von Web 2.0 ist. Dabei kommt auch der Grundgedanke der Open Source Philosophie zum Vorschein: „Release early and release often.“

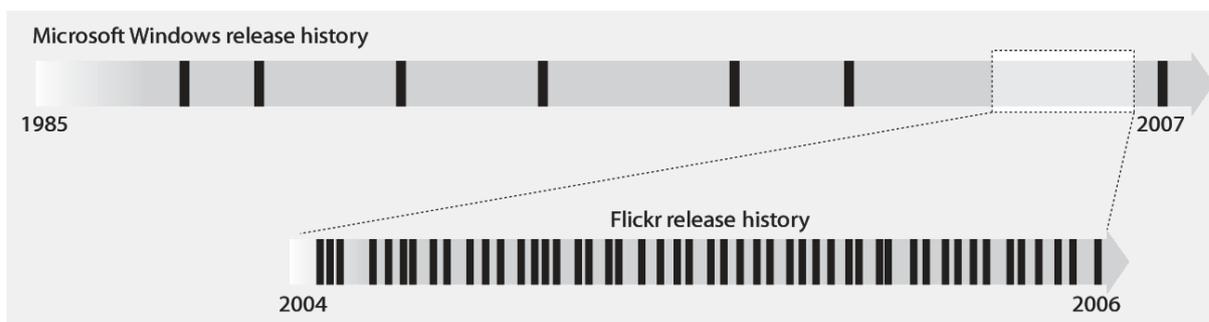


Abb. 5: Veröffentlichungszyklus Flickr Versus Microsoft (Musser & O'Reilly, 2006)

Viele der Web 2.0-Seiten tragen den Beta Status schon seit ihrer Veröffentlichung und werden ihn auch noch die nächsten Monate wenn nicht sogar Jahre tragen. Den User stört dies dabei wenig, er kann die Webseite normal nutzen und weiß gleichzeitig, dass sie weiterentwickelt wird und er in Zukunft mit neuen Features rechnen kann.

3.4.6. Der Long Tail-Effekt

Long Tail ist ein wesentlicher Bestandteil des Web 2.0 im Bereich des Verkaufs von Produkten über das Internet. Viele neue Geschäftsmodelle in der Web 2.0-Ära basieren auf diesem Prinzip und haben sich als lukrativ erwiesen.

Der Begriff wurde von Chris Anderson im Oktober 2004 durch einen Artikel im Wired Magazin geprägt. Er definiert den Begriff folgendermaßen:

The theory of the Long Tail is that our culture and economy is increasingly shifting away from a focus on a relatively small number of “hits” (mainstream products and markets) at the head of the demand curve and toward a huge number of niches in the tail (Anderson, 2006).

Long Tail, was auf Deutsch so viel wie „langer Schwanz“ oder „langer Anhang“ bedeutet, bezeichnet den Effekt, dass für Nischenprodukte ebenso eine hohe Nachfrage besteht wie für populäre Produkte. Anderson verdeutlicht dies grafisch anhand einer Kurve (siehe Abb.

6), wo er zwei Arten von Produkten erläutert: die Short Heads, jene Produkte, welche populär sind und oft verkauft werden und die Long Tails, jene Produkte mit geringer Popularität, welche dafür zahlenmäßig überlegen sind. Dabei ist ein Angebot von alternativen Produkten (Long Tails) eine Voraussetzung für den Long Tail Effekt (so auch Gratzner, 2006).

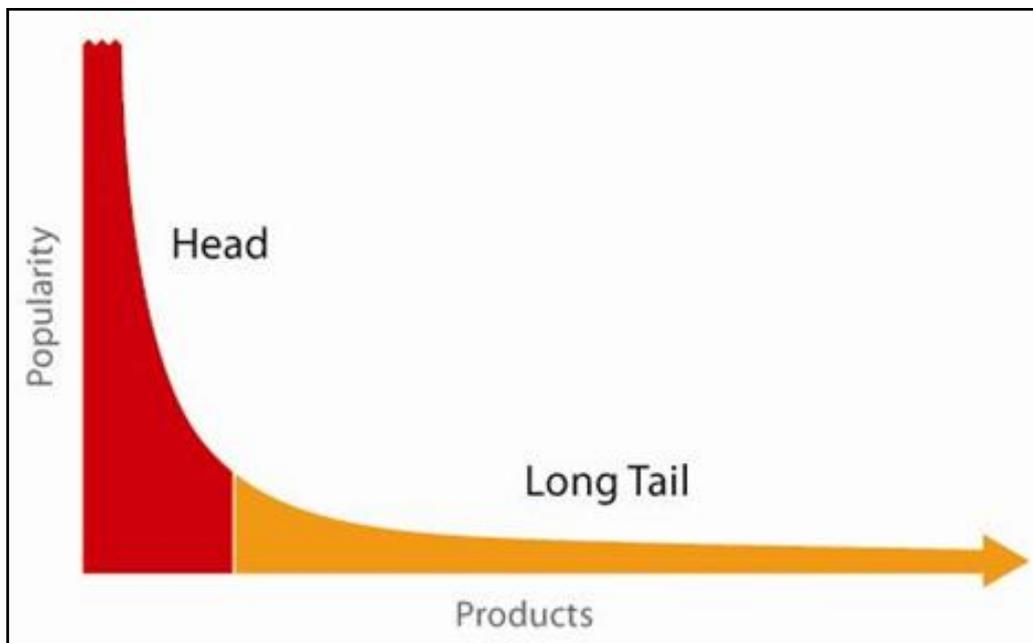


Abb. 6: Die Long Tail Kurve (Anderson, 2006)

In einem normalen Unternehmen müssen Kosten für Lager und Vertrieb in das Verkaufskonzept mit einkalkuliert werden. So wird ein normaler Buchhändler hauptsächlich Bestseller und Bücher mit hoher Nachfrage anbieten, um seinen Lagerbestand so gering wie möglich zu halten. Der Online-Buchhändler Amazon kann hingegen auch Nischenprodukte und Bücher mit geringer Nachfrage in seinen Katalog mit aufnehmen. Eine Bestellung dieser Bücher dauert dafür einige Zeit länger, da Amazon die Bestellung direkt an den Herstellerverlag weiterreicht und der Verlag zuerst das Buch an Amazon schickt, bevor es Amazon an den Kunden weiterleitet. Der Kunde nimmt eine längere Lieferzeit jedoch gerne in Kauf, da er den Titel in einem normalen Buchladen oft nicht bekommt.

Ein weiteres Beispiel wäre der iTunes Store von Apple. Ob Apple 1000-mal dasselbe populäre Lied verkauft oder jeweils ein Lied von 1000 unterschiedlichen Interpreten spielt für den Online-Musikanbieter keine Rolle. Er verdient mit dem Verkauf von Long-Tail Produkten meist noch mehr als beim Verkauf seiner Short Heads.

Der reine Verkauf von Nischenprodukten ist aus ökonomischer Sicht jedoch nicht ratsam. So sind für ein florierendes Geschäft immer noch Bestseller notwendig, um eine gewisse Attraktivität zu erzielen. Diese Bestseller sollen einen Einstiegspunkt in den Online-Store bieten. Durch geschicktes Cross-Marketing wie zum Beispiel Empfehlungssysteme können anschließend die Nischenprodukte angepriesen werden. Der Online-Versandriese Amazon, der dieses Konzept schon vor der Erfindung des Long Tail-Begriffes eingeführt hat, verzeichnet damit einen riesigen Erfolg.

3.4.7. Folksonomy

Der Begriff Folksonomy ist ein Neologismus (eine sprachliche Neubildung), welcher sich aus den englischen Wörtern *folks* und *taxonomy* zusammensetzt. Dabei bezeichnet *folks* den Menschen bzw. das Volk und *taxonomy* die Klassifizierung. Somit ist unter Folksonomy eine von Menschen geschaffene Klassifizierung zu verstehen.

Thomas Vander, welcher diesen Begriff geprägt hat, definiert ihn folgendermaßen: „Folksonomy is the result of personal free tagging of information and objects.“ (Vander, 2005)

Als tagging wird dabei das Verschlagworten von Objekten verstanden. Dabei werden einem Objekt frei wählbare Schlüsselwörter (Tags) zugeordnet, um dieses zu kategorisieren. Durch diese Vorgehensweise wird der Begriff mit zusätzlicher Semantik angereichert. Ein Tag ist dabei ein Begriff, welcher dem User zu dem Objekt einfällt. Objekte können Bilder, Webseiten, Videos, Musik o. Ä. sein.

Tagging wird bei einem Großteil der bekannten Web 2.0-Seiten wie Flickr, YouTube, del.icio.us oder last.fm eingesetzt. Visualisiert werden die Tags dabei meist mittels einer Wortwolke (engl. Tag cloud). Ein Beispiel für Tagging bietet Abb. 7. Es zeigt, welche Begriffe die User mit dem Musikstil von Falco verbinden. Durch Anklicken eines Tags kann der User weitere Interpreten aufrufen, denen dieser Tag ebenfalls zugeordnet wurde. Somit entsteht eine semantische Vernetzung der Musikinterpreten.

Umso größer ein Tag innerhalb einer Wortwolke ist, umso repräsentativer bzw. wichtiger ist dieser Begriff. Kleine Tags sind oft weit entfernt vom ursprünglichen Sinn des Objekts und können oft nicht ganz nachvollzogen werden. Ein Beispiel wäre der Begriff „electronica“ (siehe Abb. 7), welcher dem Musikstil von Falco zugeordnet wurde.

The screenshot shows the last.fm profile for Falco. The top navigation bar includes 'Musik', 'Benutzer', 'Radio', 'Events', 'Widgets', and 'Downloads'. The main content area for Falco includes a bio, a list of similar artists (Herbert Grönemeyer, STS, Drahdwaberl), and a music player for 'Falco - Rock Me Amadeus'. The right sidebar features a 'Tags' section with a word cloud of tags: '80s', 'austrian', 'electronic', 'electronica', 'german', 'new wave', 'pop', and 'rock'. A red box highlights these tags, and a red arrow points to the 'electronic' tag.

Abb. 7: Tagging am Beispiel des Musikers Falco auf der Webseite last.fm

3.4.8. Mashups

Ein aktueller Trend im Web 2.0 sind sogenannte Mashups. Der Begriff kommt eigentlich aus der Musikszene und bezeichnet einen Remix, der aus anderen Musikstücken zusammengemischt wird. Im Deutschen bedeutet es so viel wie etwas miteinander zu vermischen.

Gartner definiert den Begriff folgendermaßen: „*A mashup is a lightweight tactical integration of multi-sourced applications or content into a single offering.*“ (Gartner, 2006)

Mashups sind somit Anwendungen, welche den frei verfügbaren Inhalt anderer Webseiten oder Datenquellen in ihre eigene Webseite integrieren, um damit ein eigenes Angebot zu erstellen. Die Integration der fremden Inhalte geschieht dabei meist mittels offener Programmierschnittstellen (APIs), welche vor allem von großen Anbietern wie Google, Amazon, Flickr und eBay zur Verfügung gestellt werden. Eine weitere Möglichkeit zur Einbindung von fremdem Inhalt ist die Nutzung von News-Feeds (Göhring et al., 2006). Diese News-Feeds, besser bekannt als RSS-Feeds, sind Dateien in einer XML-Struktur, welche auf den meisten Newsseiten und Blogs zum Einsatz kommen. Mit ihnen können Inhalte von Webseiten wie zum Beispiel die neuesten Artikel oder Blogbeiträge abonniert werden.

ProgrammableWeb.com, eine Webseite welche eine Übersicht über die meisten derzeit verfügbaren APIs und ihre Nutzung anbietet, veröffentlicht täglich eine repräsentative Grafik (Abb. 8) über die Nutzungshäufigkeit der APIs.

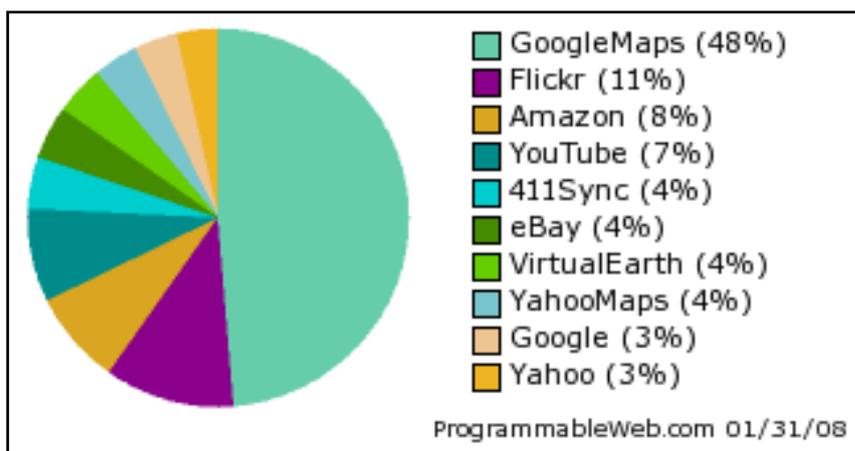


Abb. 8: Statistik der meist genutzten APIs auf programmableweb.com

Aus ihr wird ersichtlich, dass die meisten Mashups die API von Google Maps verwenden. Unter <http://code.google.com/apis/maps/> findet der Webentwickler umfangreiche Erläuterungen und Dokumentationen zu der Google Maps API. Alles ist, wie von Google gewohnt, völlig kostenlos. Diese Infos und die einfache Möglichkeit der Implementierung sind wohl dafür verantwortlich, dass die Google Maps API hier unangefochten an erster Stelle liegt. Auch der Autor dieser Arbeit verwendet Google Maps für eine seiner Kundenwebseiten (siehe Abb. 9).

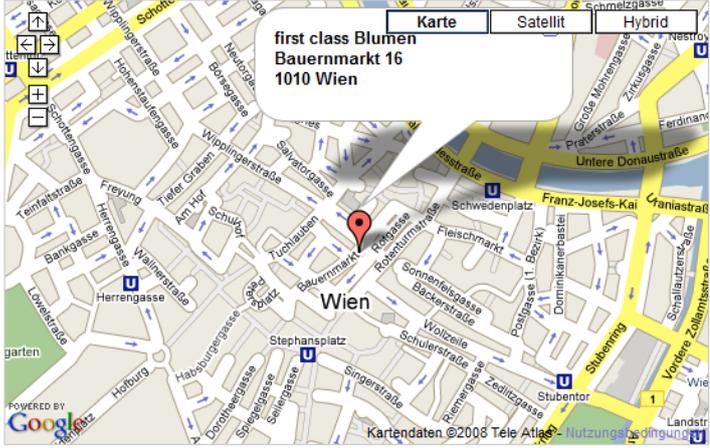


**first class
Blumen**

Kontakt

Meisterbetrieb
Sandra Holzinger-Derer
 Bauernmarkt 16, 1010 Wien
 Tel.: 01 / 907 60 20
 Fax: 01 / 907 60 39
 Mail: info@firstclassblumen.at
 UID: ATU 630 79 117
 Gewerbebuchnummer: Reg.Zl. 102 228 R01/08

Öffnungszeiten:
 Montag: 13.00 - 18.30 Uhr
 Dienstag - Freitag: 9.00 - 18.30 Uhr
 Samstag: 9.00 - 15.00 Uhr
 Montag Vormittag nach telefonischer Vereinbarung



info@firstclassblumen.at | Design by Life-is-More Multimedia | Impressum

Home
 Firmenprofil
 Floristik
 Blumenbestellung
 Events
 Kontakt

Abb. 9: Beispiel für ein Mashup – Verwendung von Google Maps in einer Webseite

Hilfestellung für das Generieren von Mashups bieten hier wieder einmal die großen Anbieter selbst. Google bietet den Google Mashup Editor (<http://editor.googlemashups.com>), Microsoft bietet Popfly (<http://www.popfly.com>) und Yahoo bringt mit Yahoo Pipes (<http://pipes.yahoo.com>) eine mächtige AJAX-Applikation für die einfache Erstellung von Mashups.

Mashups bieten sehr interessante Möglichkeiten, um Webseiten mit zusätzlichen Features auszustatten, jedoch birgt dies auch ein gewisses Risiko in sich. Das größte Risiko ist die Abhängigkeit vom Content-Lieferanten oder Dienste-Anbieter. Würde zum Beispiel Google seine API zusperren oder auf einmal Geld dafür verlangen, so müssten ebenfalls tausende Webseiten, welche auf Google Maps aufbauen (wie zum Beispiel <http://wikimapia.org> oder <http://www.placeopedia.com>), ihren Dienst einstellen. Auch Veränderungen an der API oder am Dienst selber könnten ein Problem darstellen. So behalten sich zum Beispiel Microsoft und Google das Recht vor, auf ihren Maps Werbung einzublenden. Sollte dies tatsächlich eintreffen, könnte die Nutzung der darauf aufbauenden Mashups aufgrund der Werbeeinblendungen massiv zurückgehen.

3.4.9. Der Web 2.0-Benutzer

Bei Web 2.0 steht der Benutzer im Mittelpunkt. Er kann sich aktiv am Geschehen beteiligen und seinen Beitrag zur Entwicklung des Internets beitragen, ohne dass er dabei besondere technische Fähigkeiten oder Kenntnisse über Webtechnologien benötigt.

Hier eine kleine Anekdote, wie der ultimative Web 2.0-Benutzer beschrieben werden kann: Der Web 2.0-Benutzer startet seine Reise ins Internet über eine personalisierte Startseite wie pagefalkes.com, er erweitert sein Wissen auf Wikipedia.org und veröffentlicht anschließend seine Meinung in Blogs auf Blogger.de. Er abonniert Webseiten und News als RSS, speichert interessante Webseiten bei del.icio.us und hört seine Lieblingsmusik bei last.fm. Seine E-Mails ruft er über Gmail.com ab und seine To-do-Liste schreibt er online bei Zoho.com. Seine fehlenden Sammlerobjekte kauft er auf eBay.de und die neuesten Bücher auf Amazon.de. Und anstatt den Fernseher einzuschalten, schaut er sich lieber Videos und Nachrichten auf YouTube.com an.

Doch wie sieht der Web 2.0-Benutzer in der Realität aus? Die Firma eResult GmbH hat dazu im Auftrag der Boogie Medien GmbH von Februar bis März 2007 eine Studie (eResult, 2007a) durchgeführt, wobei 1.000 Teilnehmer zur Nutzung von Web 2.0-Anwendungen befragt wurden. Die Studie ergab, dass der typische Web 2.0-Benutzer zwischen 28 und 37 Jahren alt und vorwiegend männlich ist. Fast die Hälfte der befragten Teilnehmer nutzen Wikis zur Informationsgewinnung, Foren hingegen nur ein Drittel. Die entsprechenden Umfrage-Diagramme finden sich im Anhang dieser Arbeit.

Der Web 2.0-Benutzer kann auch anhand seines Verhaltens in aktiv und passiv unterteilt werden. Aktive Web 2.0-Benutzer generieren Content, indem sie sich aktiv im Web beteiligen, zum Beispiel durch Veröffentlichen ihrer Meinungen in Blogs, Wissensweitergabe bei Wikipedia oder Upload von Bildern und Videos bei Flickr und YouTube. Der passive Benutzer hingegen ist reiner Konsument und trägt nicht zur Contentenerweiterung bei, er liest dafür die Blogbeiträge und schaut sich Bilder und Videos auf den jeweiligen Webseiten an. Auch dieser Benutzer ist wichtig für das Web, denn was würde ein Überangebot von Content bringen, wenn es keine Nachfrage danach gäbe? Beide Benutzer tragen somit zur Entwicklung des Web 2.0 bei, indem sie für ein ausgeglichenes Angebot- und Nachfrageverhältnis sorgen.

3.5. AJAX – Entstehung und Definition

Wird von Technologie im Web 2.0 gesprochen, so fällt dabei unweigerlich der Begriff AJAX. Damit ist jedoch nicht der Fußballverein Ajax Amsterdam oder Ajax der Große, Sohn des Königs Telamon von Salamis, gemeint. Die Abkürzung steht für „Asynchronous JavaScript and XML“ und bezieht sich auf die asynchrone Datenübertragung zwischen dem Webbrowser und dem Webserver. Der Begriff wurde im Jahr 2005 von Jesse James Garrett, dem Gründer der Adaptive Path Agentur, geprägt (Steyer, 2006, S. 33). Er veröffentlichte am 18. Februar 2005 einen Artikel mit dem Titel „*Ajax: A New Approach to Web Applications*“, in welchem er den Begriff AJAX definierte. Dieser Beitrag wird bis heute als eine Art „Geburtsstunde“ von AJAX angesehen (Amruth, 2006). Jesse James Garrett definiert darin AJAX folgendermaßen:

Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways. (Garrett, 2005)

Dabei zählt er folgende Techniken auf, die für AJAX maßgebend sind:

- XHTML und CSS als standardisierte Präsentationstechniken
- DOM für die dynamische Wiedergabe und Interaktion
- XML und XSLT für den Datenaustausch
- XMLHttpRequest für die Client-Server Kommunikation
- JavaScript, um die aufgelisteten Techniken zu vereinen

Hier wird auch ersichtlich, dass es sich, wie in Kapitel 3.3 erwähnt, nicht um neuartige Technologien handelt. Das Neuartige daran ist jedoch die Verknüpfung und Interaktion der jeweiligen Technologien. Nicht umsonst wird bei AJAX von „altem Wein in neuen Schläuchen“ gesprochen.

Im Folgenden sollen die einzelnen Technologien kurz erläutert werden.

3.5.1. XHTML

XHTML steht für Extensible HyperText Markup Language und ist eine von der W3C empfohlene Auszeichnungssprache für Webseiten, welche auf der Syntax von XML basiert. XHTML kann dabei auch als Erweiterung von HTML gesehen werden, es soll damit eine strengere Strukturierung der Webseiten geschaffen werden.

Die Unterschiede finden sich jedoch im Detail, so sollte bei XHTML jeder Tag einen Endtag besitzen, Attribute in Anführungszeichen gesetzt und alle Tags klein geschrieben werden. Konnte der Webentwickler bei HTML einen Zeilenumbruch mit dem `
`-Tag angeben, so soll bei XHTML die Formatierung folgendermaßen aussehen: `
</br>` oder die Kurzform `
`.

Ziel von XHTML ist es, in Zukunft XML und HTML miteinander zu vereinigen, um klare Strukturen im Web zu schaffen. Mit XHTML 2.0 soll auch das Layout einer Webseite nicht mehr in (X)HTML definiert werden, sondern nur mehr mittels CSS.

3.5.2. CSS

CSS steht für Cascading Style Sheets und bezeichnet Stilvorlagen für die Darstellung von Webseiten. Dabei wird versucht Stil (Layout) und Inhalt voneinander zu trennen, indem die Stilbeschreibung in einer eigenen CSS-Datei abgelegt wird. Im HTML-Dokument selbst werden die Bereiche, die einen eigenen Stil bekommen sollen, mittels CSS-Klassen formatiert.

Hier ein Beispiel für die CSS-Formatierung einer Überschrift:

```
01. <div class="ueberschrift">Ich bin eine rote Überschrift</div>
```

Listing 2: HTML-Code (Dateiname test.html)

```
01. .ueberschrift {
02.     font-family: Arial, Verdana;
03.     font-size: 16px;
04.     font-color: red;
05. }
```

Listing 3: Style Sheet (Dateiname style.css)

Die Referenz auf das Style Sheet geschieht im HTML Code über folgendes Statement:

```
01. <link href="style.css" rel="stylesheet" type="text/css">
```

Listing 4: Referenz auf das Style Sheet

Der Vorteil von CSS liegt auf der Hand; durch die Trennung von Inhalt und Layout können Änderungen leicht durchgeführt werden. Will der Webentwickler zum Beispiel die Farbe der Überschrift von Rot auf Grün ändern, so muss er dies nur in der CSS-Datei ausbessern. Eine Änderung sämtlicher HTML-Seiten, welche die Überschrift beinhalten, ist dadurch nicht notwendig.

Früher wurde die Grundstruktur von Webseiten mittels Tabellen gestaltet. CSS soll in Zukunft Tabellen ablösen und, auch im Sinne der Barrierefreiheit, die Gestaltung von Webseiten übernehmen.

3.5.3. DOM

DOM steht für Document Object Model und bezeichnet einen weiteren Standard des W3C (Le Hégarét, 2005). Es ist eine plattform- und sprachunabhängige Schnittstelle für den Zugriff auf XML- und XHTML-Dokumente (Gamperl, 2007, S. 21). Über DOM lässt sich ein Dokument als Baumstruktur mit hierarchischer Verschachtelung der Elemente auffassen (Steyer & Fuchs, 2006). Über die DOM-API des Webbrowsers ist es möglich, auf die einzelnen Elemente zuzugreifen und diese dynamisch zu manipulieren (bearbeiten, auswerten oder löschen). Die Manipulation geschieht dabei mittels der Skriptsprache JavaScript.

Abb. 10 zeigt die DOM-Struktur des AJAX-Beispiels aus Kapitel 3.6.

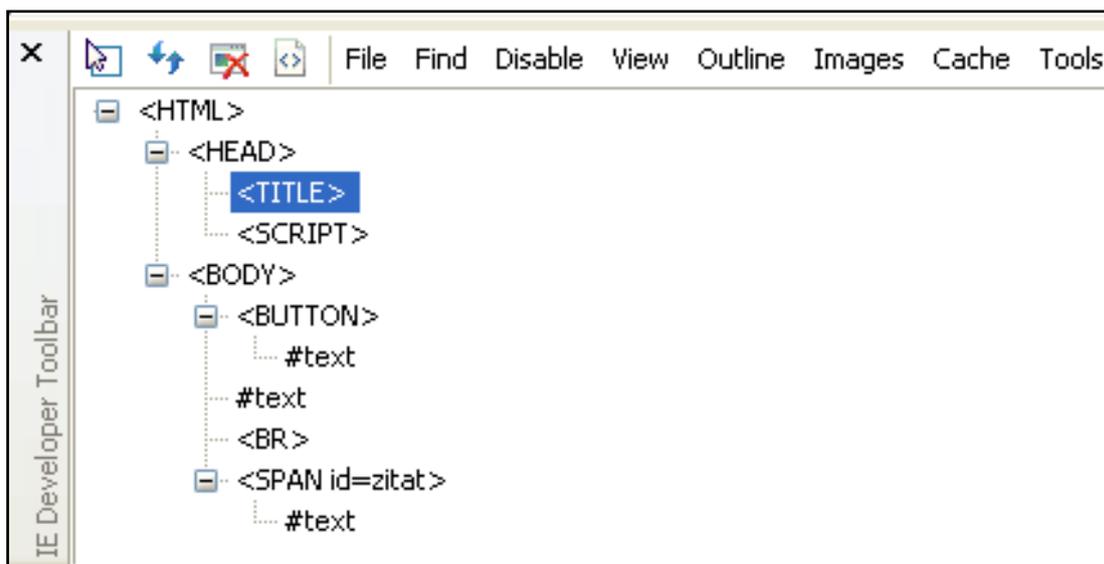


Abb. 10: Beispiel DOM-Struktur ausgelesen mit der IE Developer Toolbar

3.5.4. XML

XML steht für Extensible Markup Language und ist eine Auszeichnungssprache für die Abbildung von hierarchisch strukturierten Daten, welche hauptsächlich für den Austausch von Daten verwendet wird. Die XML-Datei weist dabei einen logischen Aufbau in Form einer Baumstruktur auf. Die Nutzdaten werden dabei von Tags eingeschlossen, um diese zu klassifizieren. Dadurch wird auch gleich ein Nachteil von XML ersichtlich: XML-Dokumente besitzen aufgrund der vielen Tags einen riesigen Overhead, welcher oftmals mehr Speicher in Anspruch nimmt als die Nutzdaten selbst. Trotzdem erfreut sich XML in der IT-Branche großer Beliebtheit und findet auch im Web einen breit gefächerten Einsatz wie zum Beispiel bei RSS-Feeds oder Webservices.

```

01. <?xml version="1.0" encoding="ISO-8859-1" ?>
02. <zitate>
03.   <zitat id="881">
04.     <text>Der Mensch ist das einzige Lebewesen, das weiß,
05.       dass es sterben wird.</text>
06.     <autor id="194">
07.       <vorname>Friedrich</vorname>
08.       <nachname>Dürrenmatt</nachname>
09.       <zusatz>schweizer Schriftsteller, Dramatiker
10.         und Maler</zusatz>
11.       <gelebt>05.01.1921 bis 14.12.1990</gelebt>
12.     </autor>
13.   </zitat>
14. </zitate>

```

Listing 5: Beispiel für eine XML-Datei

3.5.5. JavaScript

JavaScript ist eine von der Firma Sun Microsystems entwickelte, clientseitige Programmiersprache, welche in Webseiten zur Anwendung kommt und im Webbrowser ausgeführt wird. Popups, Rollover-Effekte bei Bildern und Validierung von Benutzereingaben zählen dabei zu den Hauptanwendungen von JavaScript. Es lassen sich jedoch auch einfache Spiele und komplexe Navigationselemente in JavaScript realisieren.

JavaScript weist eine Namensähnlichkeit mit der objektorientierten Programmiersprache Java auf, sollte jedoch nicht mit dieser verwechselt werden, da diese nur geringe Gemeinsamkeiten besitzen.

Im Folgenden findet sich ein Beispiel für ein JavaScript, welches in HTML eingebettet wird. Bei einem Klick auf den „Klick mich an“-Button wird die Nachricht „Du hast mich angeklickt“ im Webbrowser ausgegeben.

```
13. <script language="javascript" type="text/javascript">
14.   function zeigeMeldung()
15.   {
16.     window.alert("Du hast mich angeklickt");
17.   }
18. </script>
19.
20. <button onclick="zeigeMeldung();">Klick mich an</button>
```

Listing 6: Beispiel für ein JavaScript

Ein Nachteil von JavaScript ist, dass der User es im Webbrowser deaktivieren kann. Dadurch können Probleme mit AJAX-Anwendungen entstehen (siehe dazu Kapitel 4.2.1 - Abgeschaltetes JavaScript).

3.5.6. Traditionelle Anwendungen versus AJAX

Traditionelle Webanwendungen übermitteln ihre Daten synchron. Dies bedeutet, dass der User eine (HTTP-)Anfrage sendet und danach wartet, bis der Webserver seine Antwort an den Webbrowser schickt, erst dann kann wieder eine erneute Anfrage vom User verschickt werden. Dadurch wirken traditionelle Webseiten eher langsam, statisch und wenig interaktiv.

Moderne Web 2.0-Anwendungen hingegen benutzen AJAX, um die Seiten interaktiver zu gestalten. Das Besondere dabei ist die asynchrone Datenübertragung. Dabei können innerhalb einer Webseite Daten angefordert und angezeigt werden, ohne die gesamte Seite neu laden zu müssen. Durch diese Möglichkeit wirken Webseiten wesentlich dynamischer als traditionelle Webseiten ohne AJAX. Webanwendungen werden dadurch immer ähnlicher zu Desktop-Anwendungen und bieten diesen immer mehr Konkurrenz (siehe Kapitel 3.3.3 - Vom Desktop zum Webtop).

Ein bekanntes Beispiel für eine der ersten AJAX-Webanwendungen ist Google Suggest (siehe Abb. 11). Hier werden, nachdem der User einen Buchstaben eines Suchbegriffes eingetippt hat, automatisch (dynamisch) entsprechende Suchvorschläge angezeigt. Die Technik dahinter ist recht simpel. Jeder Tastendruck wird mit dem JavaScript Event-Handler `onkeyup` abgefangen, per XMLHttpRequest wird ein Objekt generiert und mit dem Wert der gedrückten Taste an den Google Server geschickt. Dieser durchsucht seine Datenbank und

schickt das Ergebnis an den Webbrowser zurück. Der Webbrowser aktualisiert daraufhin mittels JavaScript, CSS und Document Object Model (DOM) das Listenfeld, in welchem die Suchvorschläge angezeigt werden. Wie AJAX im Detail funktioniert, wird im nächsten Kapitel noch genauer erläutert.

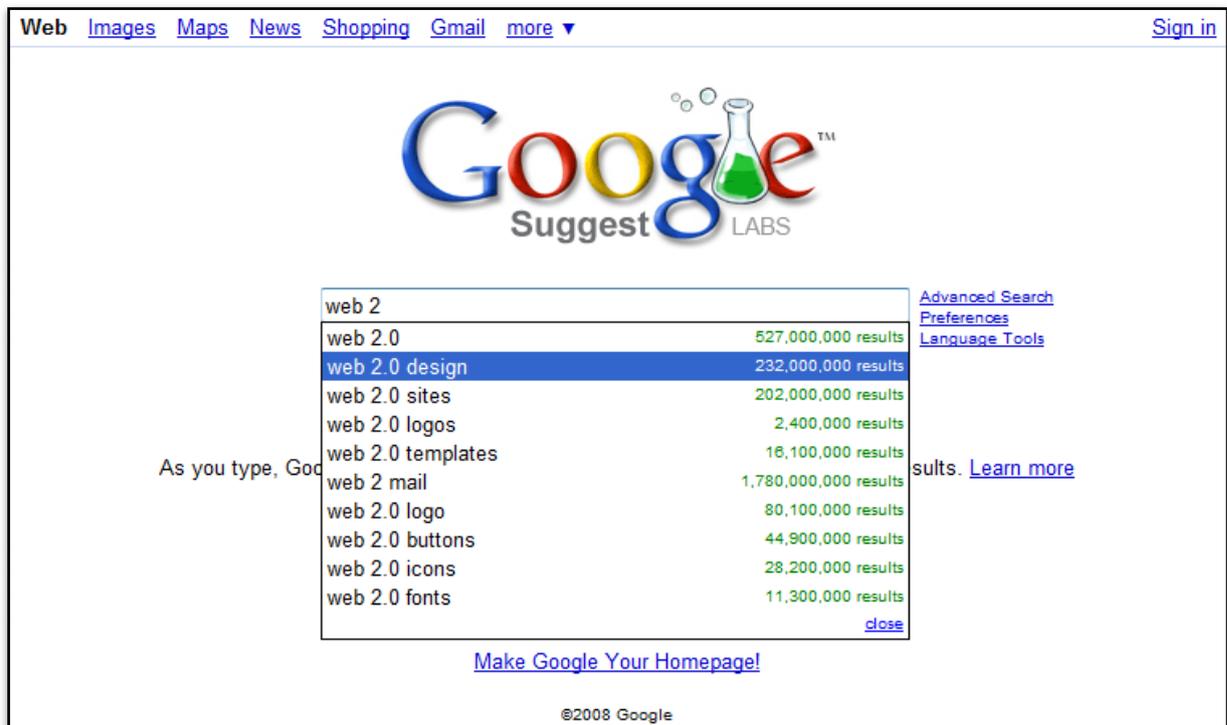


Abb. 11: Beispiel von Google Suggest (Google, 2008)

3.5.7. Wie funktioniert AJAX?

Die Grundidee hinter AJAX ist eine zusätzliche Schicht im Gegensatz zum traditionellen Modell von Webanwendungen. Diese Schicht ist die sogenannte AJAX-Engine (siehe Abb. 12). Sie soll die Interaktion der Besucher beobachten, bei Bedarf Anfragen an den Server schicken, dessen Antworten interpretieren und die Seitendarstellung entsprechend anpassen – und zwar direkt und im Hintergrund, ohne dass der Benutzer etwas von den Client-Server-Prozessen merkt. HTTP-Anfragen werden demnach durch Engine-Anfragen ersetzt, wobei Inhalte der Webseite immer automatisch aktualisiert werden, sodass ein erneuter Aufbauvorgang der kompletten Seite nicht mehr notwendig ist (Friedmann, 2007, S. 578).

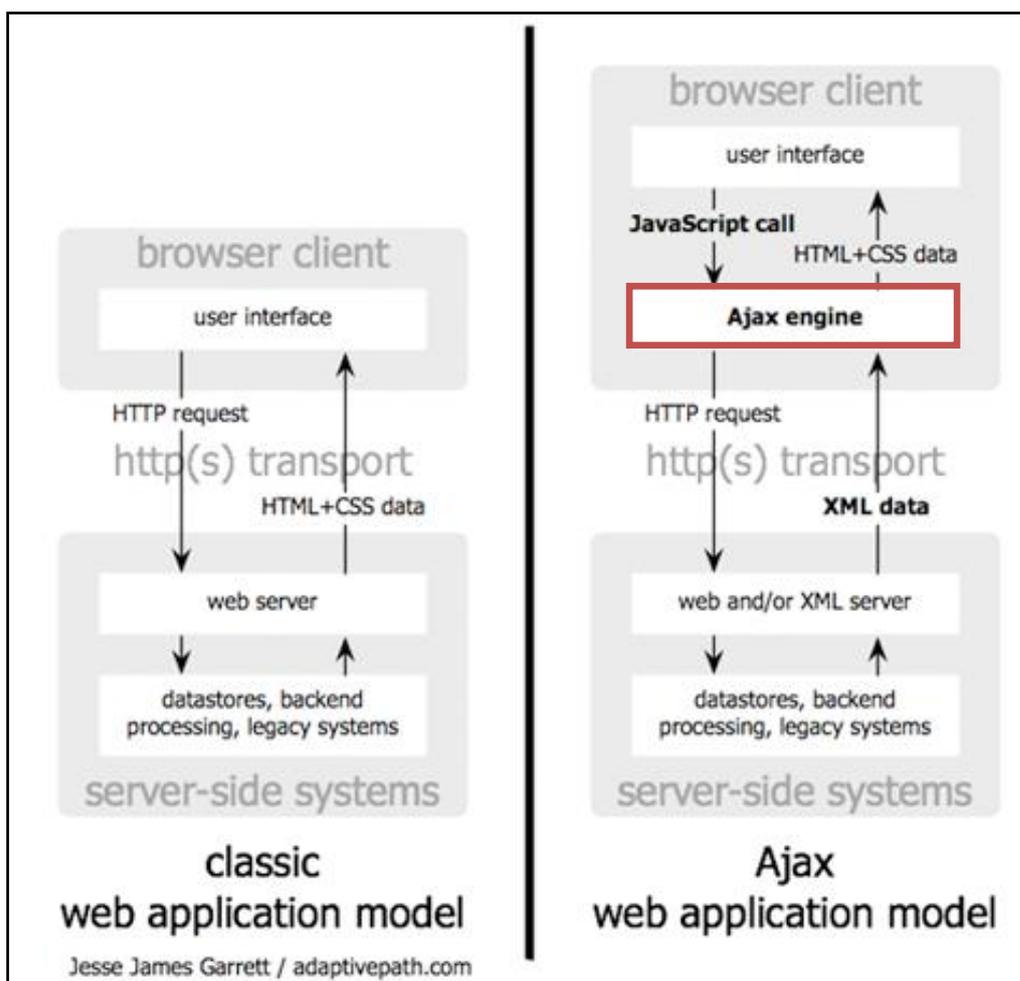


Abb. 12: Traditionelles Modell von Webanwendungen im Vergleich zum AJAX-Modell (Garrett, 2005)

Die AJAX-Engine verwendet für die asynchrone Datenübermittlung die XMLHttpRequest-API, welche im Webbrowser integriert ist. Dabei erstellt sie beim Seitenaufbau ein XMLHttpRequest-Objekt, welches, genauso wie Objekte von objektorientierten Programmiersprachen, Attribute und Methoden besitzt. Über diese Methoden können dann JavaScript Funktionen im Hintergrund auf Serverdaten zugreifen und die Seitendarstellung über DOM manipulieren (Friedmann, 2007, S. 580).

Diese AJAX-Engine ist somit nichts anderes als JavaScript Code, welcher ein XMLHttpRequest-Objekt erzeugt und verwaltet. Dieser Code kann entweder selber geschrieben (siehe Kapitel 3.6.1 - Erzeugen des XMLHttpRequest-Objekts), oder aus vorgefertigten Frameworks implementiert werden (siehe Kapitel 5.2 - Verwendung von Frameworks).

3.5.8. Wo funktioniert AJAX?

AJAX bietet den Vorteil, dass es betriebssystemunabhängig ist, es funktioniert auf einem Apple System genauso wie unter einem Windows oder Linux System. Für die Verwendung ist lediglich ein gängiger Webbrowser mit eingeschaltetem JavaScript (siehe dazu Kapitel 4.2.1 - Abgeschaltetes JavaScript) und einer XMLHttpRequest Unterstützung (siehe dazu Kapitel 4.2.3 - Alte Webbrowser und XMLHttpRequest) erforderlich.

Nicht alle Webbrowser unterstützen AJAX, so können etwa ältere Webbrowser noch kein XMLHttpRequest-Objekt erzeugen. Die nachfolgende Tabelle liefert darum eine Liste der gängigsten Browser, welche AJAX unterstützen:

Browsername:	Unterstützung ab Version:
MS Internet Explorer	5.0 als ActiveX, ab 7.0 nativ
Mozilla Firefox	1.0
Apple Safari	1.2
Opera	8.0
Konqueror	3.2
Netscape	7.1
iCab	3.0b

Tabelle 2: AJAX Browserunterstützung

Der Microsoft Internet Explorer kann XMLHttpRequest erst ab der Version 5 interpretieren. Dabei bedient sich der Webbrowser einer ActiveX-Komponente, welche jedoch bei deaktiviertem ActiveX Scripting nicht mehr funktioniert. Eine native Unterstützung liefert erst der Internet Explorer ab der Version 7.

3.6. Ein AJAX-Beispiel

Im Folgenden findet sich ein Beispiel für die Verwendung der AJAX-Technologie. Dabei wird anhand des Quellcodes die Funktionsweise erläutert. Ziel ist es, eine Webseite zu erstellen, welche ein Zitatsatz ausgibt, nachdem auf einen Button geklickt wurde. Das Zitat soll dabei dynamisch vom Webserver nachgeladen werden, ohne dass die Webseite neu geladen wird.

Zeile 1-13 aus Listing 7 geben die HTML-Grundstruktur der Webseite wieder. In Zeile 5 wird der JavaScript-Code aus den nächsten Listings eingefügt. Dieser Code ist die AJAX-Engine aus dem vorigen Kapitel. Der vollständige Quellcode des Beispiels befindet sich im Anhang dieser Arbeit.

```

1. <html>
2. <head>
3. <title>AJAX Testbeispiel</title>
4. <script language="javascript" type="text/javascript">
5. // Hier kommt der Code der nachfolgenden Listings hinein.
6. </script>
7. </head>
8. <body>
9. <button onClick="getZitat();">Neues Zitatsatz
   anzeigen</button>
10. <br />

```

```
11. <span id="zitat"></span>
12. </body>
13. </html>
```

Listing 7: AJAX-Beispiel – Darstellung der HTML-Grundstruktur

3.6.1. Erzeugen des XMLHttpRequest-Objekts

Als erster Schritt wird versucht, das XMLHttpRequest-Objekt zu erzeugen, welches grundlegend für den asynchronen Datenaustausch zwischen Webbrowser und Webserver ist. Dies geschieht mittels der Skriptsprache JavaScript folgendermaßen:

```
01. var xhrObjekt = null;
02. try {
03.   xhrObjekt = new XMLHttpRequest();
04. }
05. catch(error) {
06.   try {
07.     xhrObjekt = new ActiveXObject("Msxml2.XMLHTTP");
08.   }
09.   catch(error) {
10.     try {
11.       xhrObjekt = new ActiveXObject("Microsoft.XMLHTTP");
12.     }
13.     catch(error) {
14.       alert("Fehler beim Erzeugen des XMLHttpRequest-Objekts");
15.     }
16.   }
17. }
```

Listing 8: AJAX-Beispiel – Erzeugen des XMLHttpRequest-Objekts

Beim Aufruf des JavaScripts wird zuerst die Variable `xhrObjekt` deklariert und ihr der Wert `null` zugewiesen. In dieser Variablen wird im weiteren Verlauf das XMLHttpRequest-Objekt gespeichert. Nach der Variablendeklaration wird versucht, das Objekt zu erzeugen. Dabei werden verschiedene Varianten durchgespielt. Zuerst wird versucht, das Objekt mittels `new XMLHttpRequest()` zu erzeugen. Dies gelingt jedoch nur in den neueren Browsern (Internet Explorer ab Version 7, Firefox ab Version 1, siehe dazu auch Kapitel 3.5.8 - Wo funktioniert AJAX?). Wird ein älterer Webbrowser verwendet, so wird ein Error erzeugt, welcher über den `catch`-Block abgefangen wird. Daraufhin wird versucht, das Objekt mittels einer anderen Funktion, welche in älteren Webbrowsern (zum Beispiel IE 5 und IE 6) funktioniert, zu erzeugen. Sollte auch dies nicht erfolgreich sein, wird im Webbrowser die Fehlermeldung „Fehler beim Erzeugen des XMLHttpRequest-Objekts“ ausgegeben.

3.6.2. Verbindung mit dem Webserver

Als Nächstes wird die Funktion geschrieben, welche eine Verbindung zum Webserver herstellt und eine Anfrage an diesen schickt.

```
01. function getZitat()
02. {
03.     var url = "zitat.php";
04.     url = url + "?dummy=" + new Date().getTime();
05.     xhrObjekt.open("GET", url, true);
06.     xhrObjekt.onreadystatechange = updateSeite;
07.     xhrObjekt.send(null);
08. }
```

Listing 9: AJAX-Beispiel – Herstellen der asynchronen Verbindung

In Zeile 3 wird die URL der PHP-Datei gespeichert, welches ein neues Zitat zurückgeben soll. Hier können auch andere serverseitige Programmiersprachen wie zum Beispiel ASP.NET, Perl oder JSP zum Einsatz kommen. In dieser URL wird zusätzlich ein Dummy-Parameter übermittelt (Zeile 4), der bei jedem Aufruf einen anderen Wert hat. Der Grund dafür ist, dass der Webbrowser beim Aufruf einer URL die Antwort des Webserver in seinem Cache abspeichert. Würde die URL nicht bei jedem Aufruf unterschiedlich sein, würde beim Anfordern des Zitates immer dasselbe Zitat vom Webbrowser zurückgegeben werden.

In Zeile 5 wird nun die asynchrone Verbindung über die `open`-Methode des XMLHttpRequest-Objekt aufgerufen und dabei die URL `zitat.php` übergeben. Dies geschieht über die HTTP-Methode GET, welche es erlaubt, Daten über die URL zu übermitteln. In diesem Beispiel könnte ebenso gut die POST-Methode verwendet werden. Da hier keine Daten an die PHP-Datei übermittelt werden, sondern nur die URL aufgerufen wird, wäre das Ergebnis dasselbe. Der Parameter `true` der `open`-Methode gibt an, dass die Verbindung asynchron erfolgen soll, `false` würde hier für synchron stehen, was jedoch hier nicht gewünscht ist (siehe dazu auch Kapitel 4.1.3).

In Zeile 6 wird definiert, was bei einer Antwort des Webserver passieren soll. In diesem Fall wird die JavaScript-Funktion `updateSeite` aufgerufen, welche im übernächsten Listing beschrieben wird. Die `send`-Methode in Zeile 7 wird zum Abschicken der Anfrage an den Webserver verwendet. Der übermittelte Parameter ist entweder `null` bei GET-Anfragen wie in diesem Beispiel oder eine Variable mit den zu übermittelnden Daten bei einer POST-Anfrage.

3.6.3. Antwort des Webserver verarbeiten

Im folgenden Listing findet sich der Code der Datei `zitate.php`.

```
01. <?
02. srand(time());
03. $zufallszahl = rand(1,5);
04. switch($zufallszahl)
05. {
06.     case 1: echo "Zitat Nr. 1"; break;
07.     case 2: echo "Zitat Nr. 2"; break;
08.     case 3: echo "Zitat Nr. 3"; break;
09.     case 4: echo "Zitat Nr. 4"; break;
10.     case 5: echo "Zitat Nr. 5"; break;
11. }
12. ?>
```

Listing 10: AJAX-Beispiel – Quellcode der PHP-Datei

Dieses PHP-Skript erzeugt eine Zufallszahl zwischen 1 und 5 und gibt dementsprechend ein Zitat über die Switch-Case-Funktion aus. Dabei sei erwähnt, dass es sich hierbei um ein sehr vereinfachtes Beispielskript handelt, welches nur zur Veranschaulichung der AJAX-Funktionalität verwendet wird. In der Praxis würde das PHP-Skript auf eine Datenbank zugreifen, in welcher die Zitate gespeichert sind und von dort aus ein Zitat zurückgeben.

Im Folgenden findet sich die Funktion, welche aufgerufen wird, nachdem der Webserver eine Antwort an den Webbrowser geschickt hat.

```
01. function updateSeite()
02. {
03.     if (xhrObjekt.readyState == 4 && xhrObjekt.status == 200)
04.     {
05.         document.getElementById("zitat").innerHTML =
           xhrObjekt.responseText;
06.     }
07. }
```

Listing 11: AJAX-Beispiel – Verarbeiten der Antwort des Servers

Beim Aufruf der Funktion `updateSeite` wird zuerst überprüft, ob die Kommunikation mit dem Webserver auch abgeschlossen ist und er eine Antwort an den Webbrowser zurückgeschickt hat. Dies geschieht durch Überprüfung der Statuscodes in Zeile 3 (der Wert 4 steht dabei für „completed“, also vollständig). Ist eine Antwort erfolgreich beim Webbrowser angekommen, so wird mittels `document.getElementById(„zitat“)` auf das Element zugegriffen, dem die ID `zitat` zugeordnet ist. In diesem Fall ist das der ``-Container (Zeile 11 aus Listing 7). Über das Attribut `innerHTML` wird nun auf den ``-Container zugegriffen und der Inhalt des Containers mit der Antwort des Webserver versehen. Die Antwort des Webserver wird dabei über die Eigenschaft `xhrObjekt.responseText` abgerufen.

Abb. 13 zeigt die Ausgabe eines Testzitates, nachdem auf den Button geklickt wurde.

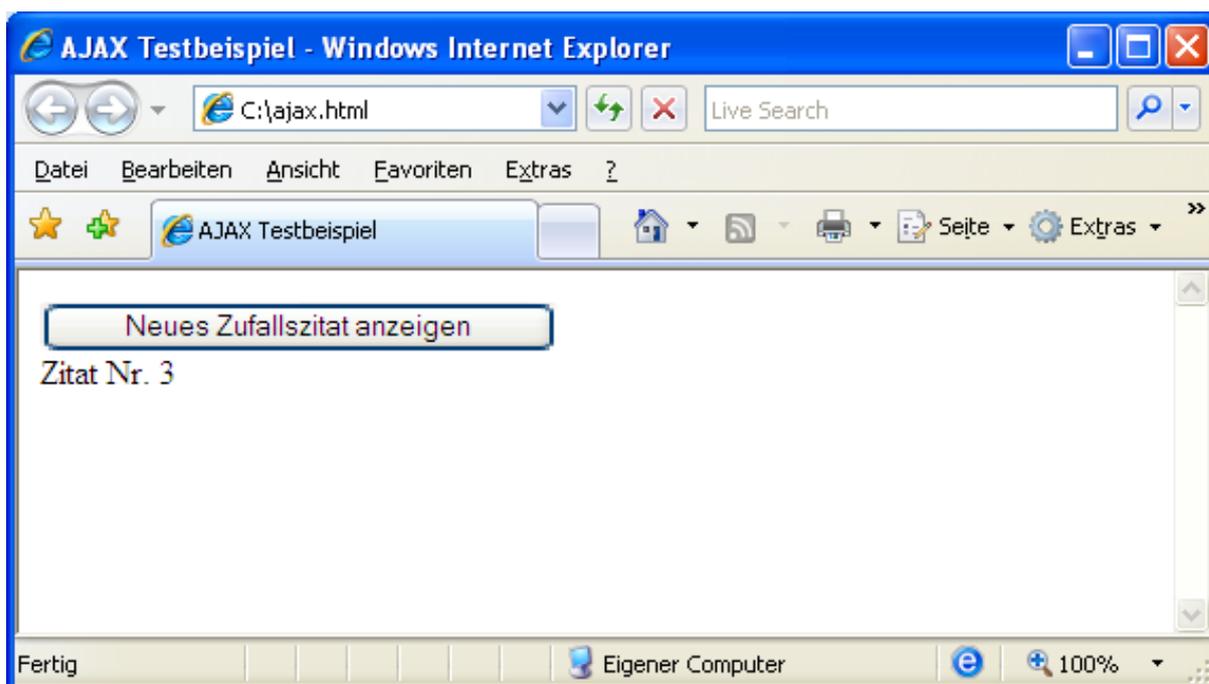


Abb. 13: Screenshot des AJAX-Beispiels

In diesem Testbeispiel wurde aufgrund der Einfachheit des Beispiels auf den Einsatz von XML verzichtet. In der Regel würde jedoch der Server nicht das Zufallszitat als unformatierten Text zurückgeben, sondern eine XML-Datei, wie zum Beispiel jene in Listing 5, an den Webbrowser zurückschicken. Dieser kann die XML-Datei parsen, auf die einzelnen Werte zugreifen und auf der Webseite ausgeben.

Für weitere Beispiele sei hier auf folgende Bücher verwiesen, welche Schritt für Schritt die Erstellung einer AJAX-Anwendung beschreiben:

- Ajax – Grundlagen, Frameworks und Praxislösungen (Mintert & Leisegang, 2007)
- AJAX – Modernes Webscripting (Pfeiffer, 2006)
- Ajax von Kopf bis Fuß (McLaughlin, 2006)
- Ajax für Dummies (Holzner, 2006)
- AJAX mit PHP (Steyer, 2006)

4. Praktischer Teil

In diesem Kapitel werden die Vor- und Nachteile der AJAX-Technologie beschrieben und anhand von Beispielen erklärt. Anschließend wird eine Umfrage ausgewertet, welche einen Einblick in das Web 2.0-Nutzungsverhalten von österreichischen Unternehmen bietet.

4.1. Vorteile von AJAX

4.1.1. Geringerer Traffic bei der Datenübertragung

Einer der Hauptvorteile von AJAX ist die geringere Datenübertragung, da bei dem asynchronen Datenaustausch nur mehr jene Daten übertragen werden, die wirklich benötigt werden. Ein gutes Beispiel bietet die Zitate-Datenbank Evangeliums.net (siehe Abb. 14), wo der User Zufallszitate dynamisch nachladen kann. Klickt der User auf den „Neues Zufallszitat“-Button, so wird ein neues Zitat vom Webserver abgerufen und in der Webseite angezeigt. Der einzige Traffic, der über die Leitung geschickt wird, ist eine XML-Datei, welche das neue Zitat und den Autor beinhaltet. Das Design-Template, also die Grafiken und der ganze HTML-Code, werden dabei nicht neu übertragen.

Hierin besteht auch das große Einsparungspotenzial der AJAX-Technologie. In diesem Fall hätte allein der HTML-Code der Webseite eine Größe von 102 KB, die XML-Datei hingegen nur circa 0,5 KB. Der Traffic beim dynamischen Nachladen eines Zitates ist hier etwa 200-mal kleiner als bei einem kompletten Neuladen der Webseite.

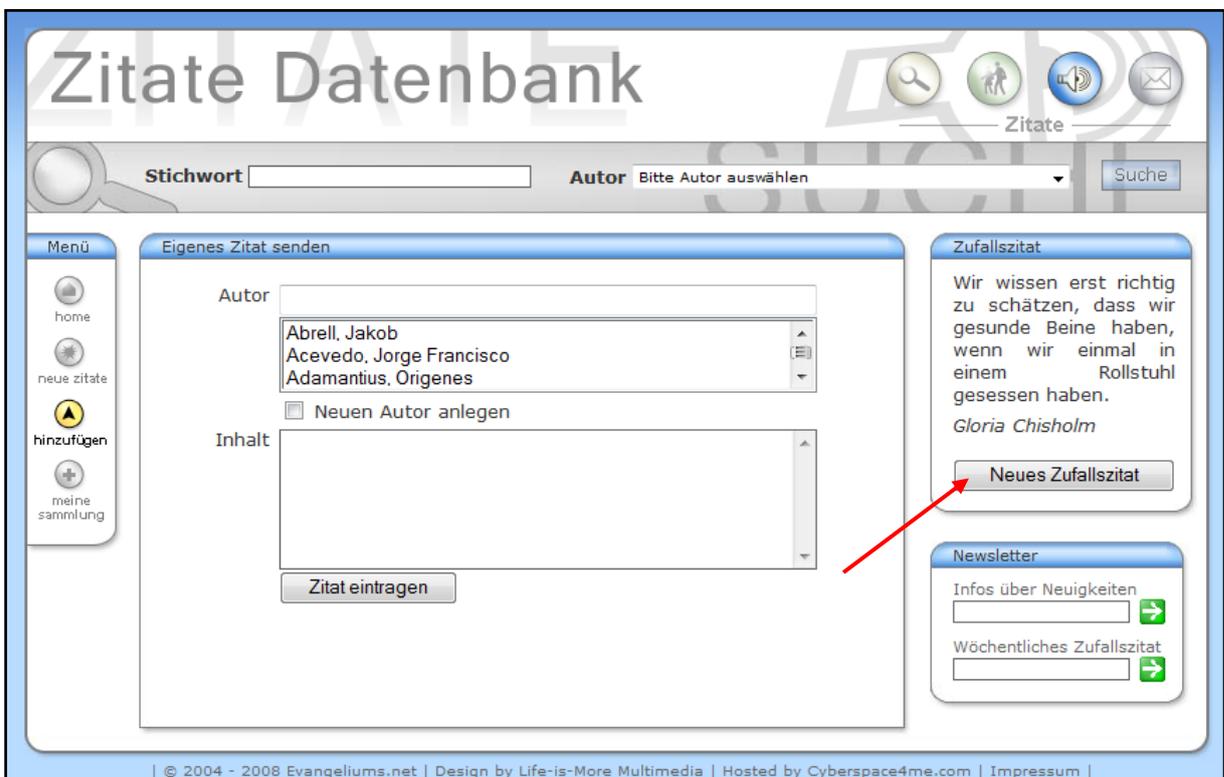


Abb. 14: Zitate Datenbank (<http://www.evangeliums.net>)

Da hier nur Nutzdaten übertragen werden, welche einen wesentlich geringeren Traffic verursachen als ein komplett neuer Aufruf der Webseite, können die Daten auch schneller angezeigt werden. Von Vorteil ist dies vor allem bei Bildergalerien, wenn sich der User Bild für Bild ansieht und die Bilder dabei dynamisch nachgeladen werden.

4.1.2. Verwendung von Standard-Technologien

Wie in Kapitel 3.3.1 erwähnt, ist AJAX eine Zusammensetzung von mehreren Technologien, welche sich in den letzten Jahren im Internet etabliert haben. XML, DOM, CSS und XHTML wurden alle zusammen von dem World Wide Web-Konsortium (W3C) entwickelt und sind De-facto-Standards im Internet. Auch JavaScript von der Firma Sun Microsystems hat sich im Web als Standard etabliert. Diese Standards sind von ihren Entwicklern auf hunderten Seiten genauestens dokumentiert und können anhand von Spezifikationen validiert und somit auf Gültigkeit überprüft werden.

Ein Vorteil von standardisierter Technologie ist, dass es im Internet unzählige Informationen, Anleitungen und Codebeispiele dazu gibt. In Newsgroups und Foren findet der Entwickler rasch Lösungen zu aktuellen Problemen. Weiters können Softwareentwickler ihr bestehendes Wissen und ihre Fähigkeiten wiederverwenden und müssen keine komplett neue Technologie lernen. Die Einstiegshürde in die Programmierung von AJAX-Anwendungen wird minimiert, da die einzelnen Technologien bereits bekannt sind und sich der nötige Lernaufwand somit in Grenzen hält.

4.1.3. Asynchrone Datenübertragung

Unter der asynchronen Datenübertragung bei AJAX wird ein zeitlich versetztes Senden und Empfangen von Daten verstanden. Dies bedeutet, dass der User, während er eine Anfrage an den Server gerichtet hat (zum Beispiel durch Klicken eines Buttons, um Daten nachzuladen) die restliche Webseite weiterhin benutzen kann. Er muss nicht erst warten, bis der Webserver die Daten liefert, um die Webseite weiterhin benutzen zu können, wie dies bei einer synchronen Datenübertragung der Fall wäre.

Der Unterschied zwischen synchroner und asynchroner Datenübertragung kann leicht festgestellt werden. Bei der Initialisierung der Verbindung durch die `open`-Methode des XMLHttpRequest-Objekts kann je nach Parameterangabe ein synchrones oder asynchrones Verhalten hervorgerufen werden. Dabei wird als dritter Parameter entweder `false` (für synchron) oder `true` (für asynchron) angegeben.

```
01. xhrObjekt.open("GET", "zitat.php", false);
```

Listing 12: Aufruf der `open`-Methode des XMLHttpRequest-Objekts

Wird der Parameter auf `false` geändert und eine Anfrage durchgeführt, so kann die Webseite erst wieder benutzt werden, wenn die Anfrage vom Webserver ausgeführt wurde. In der Zwischenzeit wird der Browser-Prozess blockiert und die Webseite bzw. der Webbrowser kann nicht mehr verwendet werden.

AJAX-Anwendungen wirken daher durch ihr asynchrones Verhalten beim User viel dynamischer und interaktiver als herkömmliche Webanwendungen. Zu beachten ist jedoch, dass der User darauf hingewiesen wird, wenn eine Interaktion mit dem Webserver stattfindet (siehe dazu Kapitel 4.2.5 - Anzeige von Aktivitätsindikatoren).

4.1.4. Keine Verwendung eines Browser-Plugins

Ein weiterer Vorteil der AJAX-Technologie ist, dass sie in fast jedem Webbrowser funktioniert, ohne dass ein entsprechendes Browser-Plugin nachinstalliert werden muss. Wird der Prozentsatz der User weggerechnet, welche JavaScript deaktiviert haben und keinen XMLHttpRequest-fähigen Webbrowser besitzen, so kommt man auf circa 97% (webhits, 2008). Ein Prozentsatz, den gerade einmal Adobe mit seinem Flash Player zustande bringt (Adobe, 2007), welcher jedoch schon seit mehr als 10 Jahren auf dem Markt ist.

Im Vergleich zu den anderen Technologien, wie zum Beispiel Visual Basic Scripting, Shockwave oder Silverlight, welche als Plugin nachinstalliert werden müssen, ist dies ein enorm hoher Anteil. Microsofts Silverlight, ehemals Windows Presentation Foundation, welches als Konkurrenzprodukt zu Flash entwickelt wurde, bringt es gerade einmal auf etwa zwei Prozent.

Der hohe Prozentsatz und der Vorteil, dass kein Plugin notwendig ist, ist sicherlich der Grund für die rasche Ausbreitung und Entwicklung von AJAX-Anwendungen.

4.1.5. Fazit

Die Vorteile von AJAX mögen anzahlmäßig den Nachteilen im nächsten Kapitel unterliegen, jedoch können diese für den User so wertvoll sein, dass der Webentwickler dafür gerne ein paar Nachteile in Kauf nimmt. Vor allem der Effekt, dass Webanwendungen im Zusammenspiel mit AJAX ein ähnliches Verhalten wie Desktopanwendungen aufweisen, bringt für den User einen entscheidenden Vorteil, welchen dieser nicht mehr missen möchte.

4.2. Nachteile von AJAX

Keine Programmiersprache oder Technologie ist hundertprozentig perfekt. So bietet auch AJAX einige Stolpersteine und Probleme in der Webentwicklung. Webseitenbetreiber sollten daher abwägen, inwieweit und in welchen Bereichen sie die neue Technologie einsetzen wollen. In diesem Kapitel werden daher die Nachteile der AJAX-Technologie beleuchtet und es wird versucht, Lösungswege aufzuzeigen.

4.2.1. Abgeschaltetes JavaScript

Laut einer aktuellen Statistik von WebHits.de (siehe Abb. 15), welche täglich fast eine Million Seitenzugriffe analysiert, haben 2,2% der Internetuser JavaScript in ihrem Webbrowser deaktiviert.

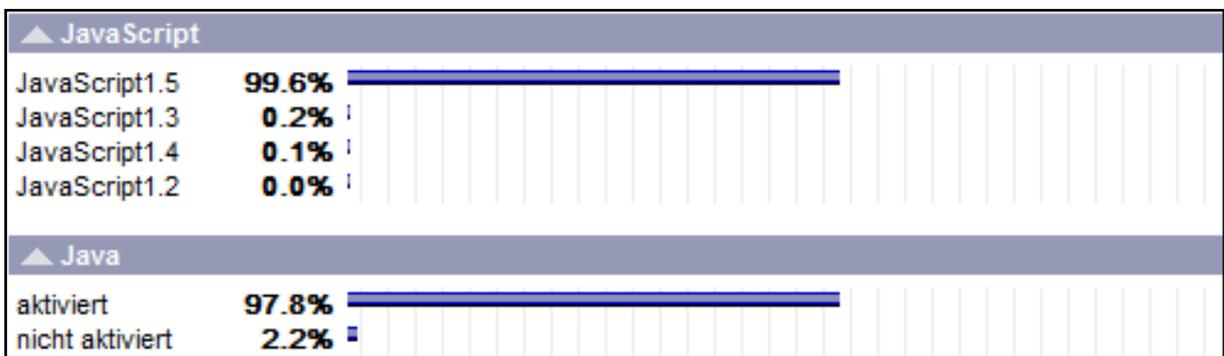


Abb. 15: Verwendung von JavaScript (webhits, 2008)

Dies ist kein besonders hoher Anteil, aber immerhin ein Prozentsatz, welcher nicht ganz außer Acht gelassen werden sollte. Das Problem bei deaktiviertem JavaScript ist, dass die verwendete AJAX-Engine (siehe Kapitel 3.5.7 - Wie funktioniert AJAX?) nicht funktioniert. Dadurch kann keine asynchrone Verbindung zum Webserver hergestellt werden und es können auch keine Elemente auf der Webseite mittels JavaScript dynamisch verändert werden.

Deaktiviert der User testweise in seinem Webbrowser JavaScript (im Internet Explorer 7 über *Extras* → *Internetoptionen* → *Sicherheit* → *Stufe anpassen* → *Scripting* → *Active Scripting* → *Deaktivieren*) und surft auf einigen bekannten Webseiten, so merkt er schnell, dass dies bei vielen Seiten nicht mehr problemlos möglich ist. So bekommt der User zum Beispiel bei Google Mail die Meldung: „JavaScript muss aktiviert sein, damit Sie Gmail in der Standardansicht verwenden können“. Bei pageflakes.com, einer bekannten Startseite, bekommt der User die Meldung: „This page is designed for JavaScript-only browsers“. Auch Google Suggest (siehe Kapitel 3.5.6 - Traditionelle Anwendungen versus AJAX), Google Maps, YouTube und ein Großteil der Web 2.0-Seiten funktionieren nur mehr eingeschränkt bzw. gar nicht mehr.

Eine Lösung des Problems ist nur teilweise möglich und auch dann nur durch zeitaufwendige Zusatzprogrammierung machbar. So muss schon beim Aufruf der Webseite überprüft werden, ob beim Webbrowser JavaScript deaktiviert ist. Ist dies der Fall, sollte eine Webseite geladen werden, die auch mit deaktiviertem JavaScript noch funktioniert. Als Vorbild sei hier eBay genannt, hier funktioniert das Beobachten der Artikel oder das Aktualisieren der Auktionsrestzeit auch bei deaktiviertem JavaScript. Bei eingeschaltetem JavaScript würden diese Funktionen ein asynchrones Verhalten aufweisen. Klickt der User in einem Angebot

bei eBay auf den „Diesen Artikel beobachten“-Button, so wird nicht die komplette Webseite neu geladen, sondern nur der Bereich um den Button herum aktualisiert.

Verwendet ein Webentwickler nun AJAX oder JavaScript auf einer Webseite, hat aber nicht die Zeit, zusätzlich eine Seite zu programmieren, die auch ohne JavaScript läuft, so sollte er zumindest den `noscript` HTML-Tag verwenden. Alles, was im HTML-Code zwischen `<noscript>` und `</noscript>` definiert wird, wird nur dann im Webbrowser angezeigt, wenn der User JavaScript deaktiviert hat. Nähere Informationen dazu gibt es auf der Webseite des World Wide Web Konsortiums (W3C, 1999).

4.2.2. Mangelnde Funktion der Zurück-Schaltfläche

Eines der wesentlichen Hauptprobleme von modernen Webseiten, welche AJAX verwenden, ist die mangelnde Funktion der Vor- und Zurück-Schaltflächen. Die meisten User hoffen, durch ein Drücken des Zurück-Buttons den vorherigen Zustand einer Webseite wiederherzustellen. Das Problem dabei ist jedoch, dass dynamisch generierte Webseiten nicht in der Historie des Webbrowsers abgespeichert werden.

Ein Beispiel dafür ist der AJAX-Onlineshop (<http://shop.palinka.net/shop.htm>) der Firma Zimek. Wird die Startseite des Onlineshops geladen, erscheint ein Hinweisenfenster mit folgendem Warnhinweis: „*Sie betreten gerade einen Shop der in der AJAX Technologie programmiert wurde. Bitte verwenden Sie keine Navigationselemente (Zurück / Vorwärts Buttons) Ihres Browsers, weil Sie dann den Shop verlassen und die bisher gespeicherten Daten verloren wären.*“ (siehe Abb. 16). Eine solche Meldung wirkt eher abschreckend und die meisten User werden den Onlineshop verlassen, ohne die tollen Funktionen auszuprobieren, die ihnen der neue Shop bietet.



Abb. 16: Warnhinweis im Onlineshop der Firma Zimek

Ein weiteres Beispiel ist das Planespotting Network, welches unter <http://www.fysb.de> erreichbar ist. Hier können Planespotter (Personen, die Flugzeuge beobachten und fotografieren) ihre Fotos uploaden und Fotos anderer User betrachten. Die Fotos werden dabei dynamisch mittels AJAX nachgeladen. Betrachtet der User einige Fotos und klickt anschließend auf die Zurück-Schaltfläche des Webbrowsers, so wird allerdings nicht, wie er vermuten würde, das vorherige Foto angezeigt, sondern die letzte vollständig geladene Webseite angezeigt. Oft ist dies eine Suchmaschine wie Google, über die der User auf die Webseite gekommen ist.

Abhilfe schaffen hier eigene History Logger Tools wie zum Beispiel der StateManager (<http://exanimo.com/actionscript/statemanager>) oder Really Simple History (RSH), eine AJAX History und Bookmark Library (<http://code.google.com/p/reallysimplehistory>).

Technisch möglich wird dies durch die Verwendung von Iframes oder manipulierten URLs mittels eines HTML-Ankers. Eine ausführliche Erklärung dazu liefert Tobias Hauser in seinem Artikel „*Rettungsanker*“, welcher im Linux-Magazin 2007/02 erschienen ist (Hauser, 2007) oder Christian Wenz in seinem Buch „*JavaScript und AJAX*“ (Wenz, 2006, S. 406).

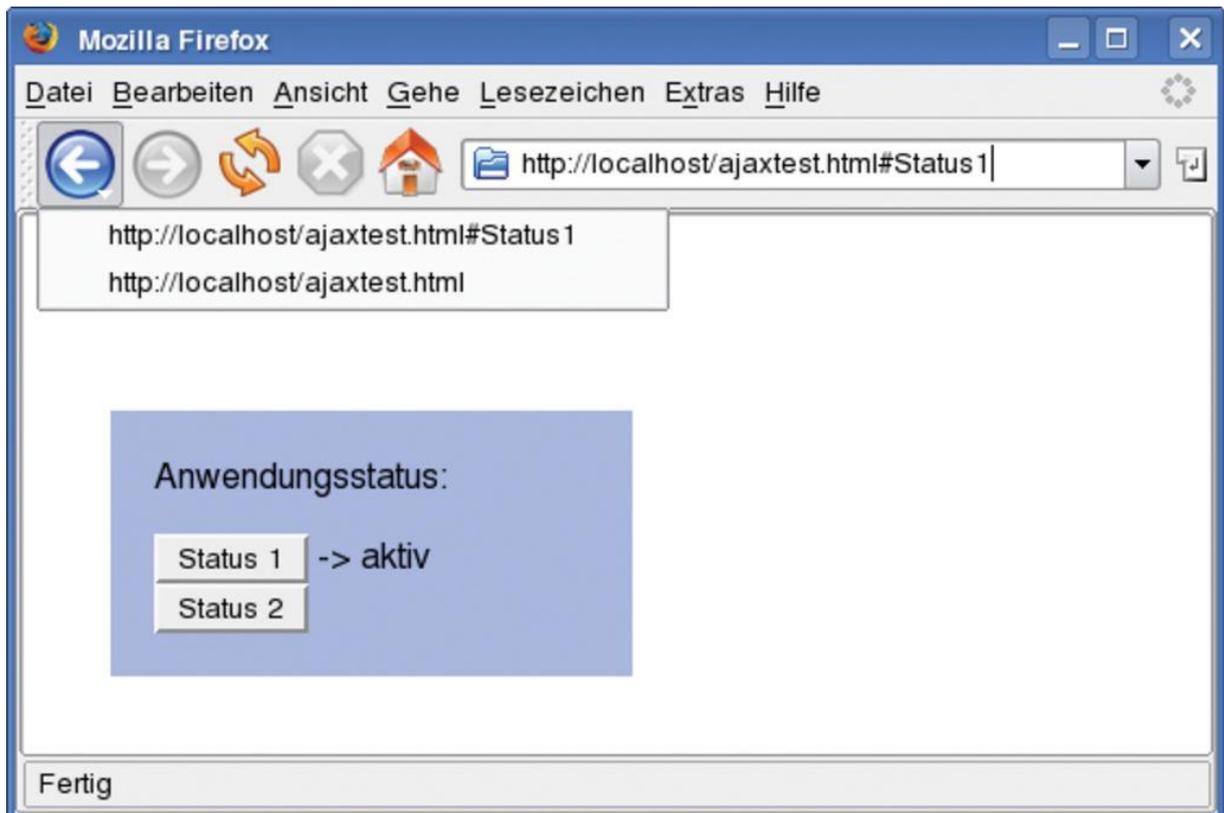


Abb. 17: Der HTML-Anker nach dem #-Zeichen in URLs lässt sich als Status-ID missbrauchen, sodass sich deren Zustand in einer Bookmark erfassen lässt (Hauser, 2007).

4.2.3. Alte Webbrowser und XMLHttpRequest

Wie in Kapitel 3.5.7 erwähnt, verwendet die AJAX-Engine die XMLHttpRequest-API des Webbrowsers. Diese ist jedoch erst im Internet Explorer ab Version 5 und im Mozilla Firefox ab Version 1 enthalten (siehe dazu Tabelle 2: AJAX Browserunterstützung). Ältere Browserversionen können daher die XMLHttpRequest-API nicht verwenden und somit auch kein XMLHttpRequest-Objekt erstellen, was aber für die asynchrone Datenübertragung nötig ist. Der Prozentsatz der User, welche derart alte Webbrowser benutzen, ist zwar sehr gering (circa 1%), jedoch sollte auch an diese gedacht werden und zumindest eine Fehlermeldung ausgegeben werden.

Bevor der User vor einer fehlerhaften Webseite steht und verzweifelt versucht, nicht funktionierende AJAX-Funktionen zu verwenden, sollte er darauf hingewiesen werden, dass sein Webbrowser zu alt ist und er darum einige Funktionen nicht benutzen kann. Bei der Erstellung des XMLHttpRequest-Objekts sollte mittels einer try-catch-Anweisung ein Fehlschlag abgefangen und stattdessen ein Warnhinweis ausgegeben werden. Der Code dafür könnte folgendermaßen aussehen:

```

01. var xhrObjekt = null;
02. try {
03.     xhrObjekt = new XMLHttpRequest();
04. }
05. catch (error) {
06.     xhrObjekt = null;
07. }

```

```
08. if (xhrObjekt == null) alert("Ihr Webbrowser ist zu alt um die
Funktionen dieser Webseite zu nutzen. Bitte installieren Sie
einen aktuelleren Browser! ");
```

Listing 13: Beispiel für die Initialisierung des XMLHttpRequest-Objekts

Abb. 18 zeigt die Ausgabe des Warnhinweises aus Listing 13, welcher ausgegeben wird, wenn das XMLHttpRequest-Objekt nicht erstellt werden kann.



Abb. 18: Warnhinweis beim Erzeugen des XMLHttpRequest-Objektes (getestet im Opera 7.52)

4.2.4. Eingeschränkte Bookmark-Funktion

Was eine Webseite auszeichnet, ist unter anderem, dass der User sie über eine bestimmte URL aufrufen kann. Bei Verwendung von AJAX ist es jedoch meistens so, dass sich der Inhalt einer Webseite ändert, die URL jedoch in der Browserleiste gleich bleibt. Der Grund dafür ist, dass der Inhalt der Seite dynamisch nachgeladen wird und kein komplett neuer Seitenaufruf stattfindet. Dadurch werden Änderungen nicht in der Historie des Webbrowsers vermerkt. Dies ist im Übrigen nicht nur ein Problem von AJAX, auch Webseiten, welche in Adobe Flash gestaltet werden, haben mit demselben Problem zu kämpfen.

Lässt der User sich zum Beispiel auf Google Maps eine Route berechnen und will das Ergebnis in seinen Favoriten (bei Mozilla Firefox Lesezeichen genannt) abspeichern, so wird leider nur die URL <http://maps.google.at> gespeichert. Ruft der User die gespeicherte Webseite über die Favoriten wieder auf, so landet er auf der Startseite von Google Maps – die gespeicherte Route wird jedoch nicht mehr angezeigt.

Eine Abhilfe schaffen sogenannte Permalinks. Dieses Wort, welches sich aus permanent und Hyperlink zusammensetzt, bezeichnet einen eindeutigen, sich nicht ändernden Link zu einer bestimmten Webseite samt Zustand. Diese Permalinks müssen allerdings erst generiert werden und stellen somit wieder einen zusätzlichen Programmieraufwand dar.

Auch Google Maps bietet Permalinks an. Klickt der User oben rechts auf den Link „URL zu dieser Seite“, so wird der generierte Link eingeblendet (siehe Abb. 19). Diesen kann der User kopieren und manuell zu den Favoriten hinzufügen.



Abb. 19: Eingblendeter Permalink bei Google Maps

4.2.5. Anzeige von Aktivitätsindikatoren

Dynamische Webseiten, welche AJAX verwenden, bieten dem User oft Interaktionen an, welche diesem aufgrund der Neuartigkeit der Technologie noch ungewohnt sind. Ein Bildupload mittels AJAX ohne einen Aktivitätsindikator (also eine Rückmeldung an den User, dass die Daten gerade übertragen werden) würde den Bildupload als nicht funktionierend erscheinen lassen, da die Daten im Hintergrund über die AJAX-Engine übertragen werden und standardmäßig erst bei erfolgreicher Übertragung eine Rückmeldung erzeugt wird. Darum ist es bei der Verwendung von AJAX immer erforderlich, den User darauf hinzuweisen, wenn eine Interaktion zwischen dem Webbrowser und dem Webserver stattfindet. Für solche Fälle gibt es unterschiedliche Grafiken oder Texte, sogenannte Aktivitätsindikatoren oder visuelle Hinweise, wie Gethland und Gabrait (2006, S. 125) sie nennen, an die sich der User bereits gewöhnt hat und die weitgehend etabliert sind.

Als textuelle Aktivitätsindikatoren kann die Phrase „*Einen Moment bitte, Daten werden geladen*“ oder einfach das Wort „*Loading...*“ verwendet werden. Als grafische Aktivitätsindikatoren eignen sich Fortschrittsbalken bzw. Fortschrittskreise oder die Sanduhr, welche bereits in Desktopanwendungen zum Einsatz kommt.



Abb. 20: Beispiel für grafische Aktivitätsindikatoren

Die grafischen Aktivitätsindikatoren können beim Laden der Webseite schon mitgeladen werden, wobei ihre Sichtbarkeit (*visibility*) auf Versteckt (*hidden*) eingestellt werden. Über das Document Objekt Model (DOM) kann anschließend auf den Aktivitätsindikator zugegriffen werden und die Sichtbarkeit mittels JavaScript eingeschaltet werden (siehe Zeile 3 in Listing 14). Dadurch kommt es bei der Darstellung nicht zu Verzögerungen, da der Aktivitätsindikator nicht erst geladen werden muss, sondern sich schon im Cache des Webbrowsers befindet.

Das folgende Listing zeigt, wie das Einblenden von versteckten Aktivitätsindikatoren realisiert werden kann. Zu diesem Zweck muss lediglich die Funktion `indikatorEinblenden` aufgerufen werden.

```
1. <script language="javascript" type="text/javascript">
2. function indikatorEinblenden() {
3.     document.getElementById("sanduhr").style.visibility =
4.         "visible";
5. }
6. </script>
7. 
```

Listing 14: Beispiel für das Einblenden eines Aktivitätsindikators

4.2.6. Hohe Serverbelastung durch Event-Handler

JavaScript bietet sehr viele Möglichkeiten, auf Benutzereingaben zu reagieren und dadurch eine Interaktion mit dem Webserver zu erzeugen. Dafür sorgen Event-Handler, wie zum Beispiel `onkeydown`, `onkeyup`, `onmouseover` oder `onfocus`, welche bei bestimmten Mausbewegungen oder Tastatureingaben eine Aktion ausführen. Eine vollständige Liste der Event-Handler gibt es unter anderem auf der Webseite des W3C unter <http://www.w3.org/TR/html401/interact/scripts.html#h-18.2.3>.

Wie in Kapitel 3.5.6 erwähnt, verwendet auch Google Suggest einen JavaScript Event-Handler, um nach Drücken eines Buchstabens eine Verbindung mit dem Webserver herzustellen und einen Suchvorschlag anzuzeigen. Im Gegenzug zur klassischen Suche, wo nur eine Anfrage an den Webserver gestellt wird, wird bei der Eingabe des Suchwortes „webdevelopment“ bei Google Suggest gleich vierzehn Mal (für jeden Buchstaben des Suchwortes) eine Verbindung zum Google-Webserver aufgebaut. Für den Suchgiganten Google, der pro Sekunde hunderttausende Anfragen verarbeiten kann, wird diese zusätzliche Serverbelastung kein Problem darstellen. Normale Webserver jedoch können durch massiven Einsatz von unüberlegten Event-Handlern in die Knie gezwungen werden.

Event-Handler, welche Serveranfragen generieren, sollten daher immer mit Bedacht eingesetzt werden, vor allem, wenn der Webserver nicht leistungsstark genug ist. Als unbedenklich können jedoch die Event-Handler `onload`, `onsubmit` und `onclick` gesehen werden, da diese nur einmalig bzw. nur nach einem Mausklick eine Aktion ausführen und somit den Webserver nicht sonderlich mehr belasten, als eine normale Anwendung dies tun würde.

Problematisch sind auch Skripte, welche zyklische Anfragen an den Webserver stellen (sogenanntes „Polling“). Dies trifft vor allem auf Chat- und Webmail-Skripte zu, welche die AJAX-Technologie verwenden. Auch hier sollte der Webentwickler vorsichtig sein und die Serverlast genauestens im Auge behalten, da diese Skripte im Hintergrund alle paar Sekunden (oft sogar in kürzeren Abständen) eine Verbindung mit dem Webserver herstellen, um zu überprüfen, ob neuer Content vorhanden ist. Abhilfe könnte hier die Comet Technologie schaffen (siehe Schiemann, 2007), welche eine permanente HTTP-Verbindung zwischen dem Webserver und dem Webbrowser herstellt.

4.2.7. Höherer Aufwand bei der Entwicklung

Wurden früher Experten für serverseitige Skriptsprachen benötigt, so werden heute für AJAX-Anwendungen noch zusätzlich Experten für clientseitige Sprachen gebraucht, vor allem für JavaScript. Die Entwicklung von AJAX-Anwendungen geschieht dabei mit normalen Text- oder Webeditoren wie zum Beispiel Adobe Dreamweaver, Microsoft Visual Studio .NET oder Eclipse. Einige davon versuchen zwar den Entwickler mit mitgelieferten Frameworks (wie zum Beispiel Adobe Spry Framework oder Eclipse Ajax Toolkit Framework) zu unterstützen, trotzdem muss dieser einen erheblichen Mehraufwand in die Entwicklung einer Webseite hineinstecken. So muss der Code umfangreich getestet werden, damit er auch in den unterschiedlichen Webbrowsern ausführbar ist (Stichwort: Cross Browser Compatibility). Soll die Webseite dazu noch barrierefrei sein, muss meist eine alternative Seite erstellt werden, welche auch bei abgeschaltetem JavaScript noch funktioniert. Dies bedeutet neben einer längeren Entwicklungsdauer auch einen finanziellen Mehraufwand, welcher sich in höheren Kosten der Webseite niederschlägt.

Wird ein traditionelles Webmail wie zum Beispiel Horde oder GMX mit Google Mail verglichen, so wird schnell klar, dass in die Entwicklung von Google Mail, welches die AJAX-Technologie verwendet, ein erheblich höherer Anteil an Entwicklungskosten und Mannstunden geflossen sind.

Webseiten, die AJAX-Funktionalitäten verwenden, werden somit in der Entwicklung immer teurer sein als Webseiten ohne AJAX. Hier gilt es dann den Mehrwert, den AJAX bietet, gegen die höheren Entwicklungskosten abzuwiegen und zu entscheiden, ob AJAX überhaupt eingesetzt werden soll (siehe auch Kapitel 5.1 - AJAX sinnvoll oder nicht?).

4.2.8. Barrierefreiheit

Barrierefreiheit ist ein aktuelles Thema im Bereich der Webentwicklung, wird jedoch aus Funktions- und Kostengründen leider viel zu wenig beachtet. Doch was ist Barrierefreiheit überhaupt? Die Web Accessibility Initiative (WAI) definiert den Begriff folgendermaßen:

Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging (Henry, 2005).

Auch im österreichischen Recht gibt es eine Definition dafür. So steht im Bundesgesetz über die Gleichstellung von Menschen mit Behinderungen (Bundes-Behindertengleichstellungsgesetz – BGStG) Folgendes:

Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, **Systeme der Informationsverarbeitung** sowie andere gestaltete Lebensbereiche, wenn sie für Menschen mit Behinderungen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind (BGStG §6 Abs. 5).

Eine Behinderung im Sinne dieses Bundesgesetzes ist „die Auswirkung einer nicht nur vorübergehenden körperlichen, geistigen oder psychischen Funktionsbeeinträchtigung oder Beeinträchtigung der Sinnesfunktionen, die geeignet ist, die Teilhabe am Leben in der Gesellschaft zu erschweren. Als nicht nur vorübergehend gilt ein Zeitraum von mehr als voraussichtlich sechs Monaten“ (BGStG §3).

Barrierefreie Webseiten sind somit Seiten, welche auch für Personen mit Behinderungen oder Einschränkungen zugänglich und benutzbar sein müssen. So soll auch Personen mit Sehbehinderungen (wie zum Beispiel Blinde oder Farbenblinde), motorischen Störungen (zum Beispiel Spastiker), invaliden oder älteren Personen der Zugang zum Internet nicht verwehrt bleiben.

Das österreichische Recht geht sogar so weit, dass es Barrierefreiheit für Behörden im E-Government-Gesetzes (E-GovG) gesetzlich vorschreibt. Der Gesetzestext besagt dabei Folgendes:

Bei der Umsetzung der Ziele dieses Bundesgesetzes ist Vorsorge dafür zu treffen, dass behördliche Internetauftritte, die Informationen anbieten oder Verfahren elektronisch unterstützen, spätestens bis 1. Jänner 2008 so gestaltet sind, dass internationale Standards über die Web-Zugänglichkeit auch hinsichtlich des barrierefreien Zugangs für behinderte Menschen eingehalten werden (E-GovG §1 Abs. 3).

Öffentlich zugängliche Webseiten von Behörden müssen somit ab 01.01.2008 barrierefrei sein. Dass dies in Österreich auch größtenteils umgesetzt wurde, zeigt zum Beispiel die Webseite des Bundeskanzleramts (<http://www.bka.gv.at>), welche sich seit Beginn des Jahres in einem neuen Design zeigt. Die Webseite funktioniert auch bei abgeschaltetem JavaScript und verwendet für das Design nun DIV-Elemente anstatt Tabellen. Die Verwendung der Webseite mit dem textbasierten Webbrowser Lynx ist ohne Weiteres möglich und zeigt, dass auch blinde Personen den Inhalt der Webseite über eine Braillezeile erfassen können.

Doch was bedeutet dies nun für den Einsatz von AJAX?

Werden barrierefreie Webseiten betrachtet, so kann festgestellt werden, dass diese ohne jeglichen „Schnickschnack“ auskommen. Dynamische Effekte und Interaktionen werden hier vergebens gesucht. Der Grund dafür ist, dass barrierefreie Webseiten ohne JavaScript funktionieren müssen. Laut einer Richtlinie von Wendy Chisholm (2000) muss bei der Verwendung von Skripten auf einer Webseite darauf geachtet werden, dass die Informationen auch bei deaktiviertem Scripting zugänglich bzw. auf einer alternativen Webseite erreichbar sind. Auch die Barrierefreie Informationstechnik-Verordnung (BITV) der Bundesrepublik Deutschland besagt in der Anlage zu §§ 3 und 4 Abs. 1 der BITV, dass sichergestellt sein muss, dass mittels Markup-Sprachen geschaffene Dokumente verwendbar sein müssen, wenn Script, Applets oder andere programmierte Objekte deaktiviert sind. Da AJAX jedoch eingeschaltetes JavaScript voraussetzt (siehe Kapitel 4.2.1 - Abgeschaltetes JavaScript), bedeutet dies wohl auch das Aus für den Einsatz in barrierefreien Webseiten.

Bedient eine invalide Person beispielsweise ihren Computer mittels Sprachsteuerung, so können Probleme bei AJAX-Anwendungen auftreten. Beim Ausfüllen von Formularfeldern wird hier nicht mehr die Tastatur verwendet, sondern eine eigene Sprachsteuerungssoftware, daher sind JavaScript Event-Handler, welche Tastaturbefehle abfangen, wirkungslos. Google Suggest, welches, wie schon in Kapitel 3.5.6 erwähnt, den Event-Handler `onkeyup` verwendet, funktioniert bei der Verwendung einer Sprachsteuerung nicht mehr. Einen Suchvorschlag von Google wird der User hier ergebnislos erwarten.

Eine Möglichkeit, AJAX doch einzusetzen, wäre, vorher zu überprüfen, ob JavaScript im Webbrowser deaktiviert ist oder nicht und dementsprechend eine Webseite mit AJAX- oder ohne AJAX-Funktionalität zur Verfügung zu stellen. Amazon (2008) ist hier vorbildlich und realisiert auf diesem Weg zum Beispiel den Verkauf von Diamant-Ringen. Abb. 21 zeigt die Webseite bei aktiviertem JavaScript und Abb. 22 bei deaktiviertem JavaScript. Hat der Benutzer JavaScript aktiviert, so kann er die AJAX-Funktionalität der Webseite nutzen und die Suche nach einem Diamantring dynamisch anpassen. Bei deaktiviertem JavaScript kann

er die Suche jedoch noch immer benutzen, indem er seine Suchbegriffe in normale Formularfelder eingibt.

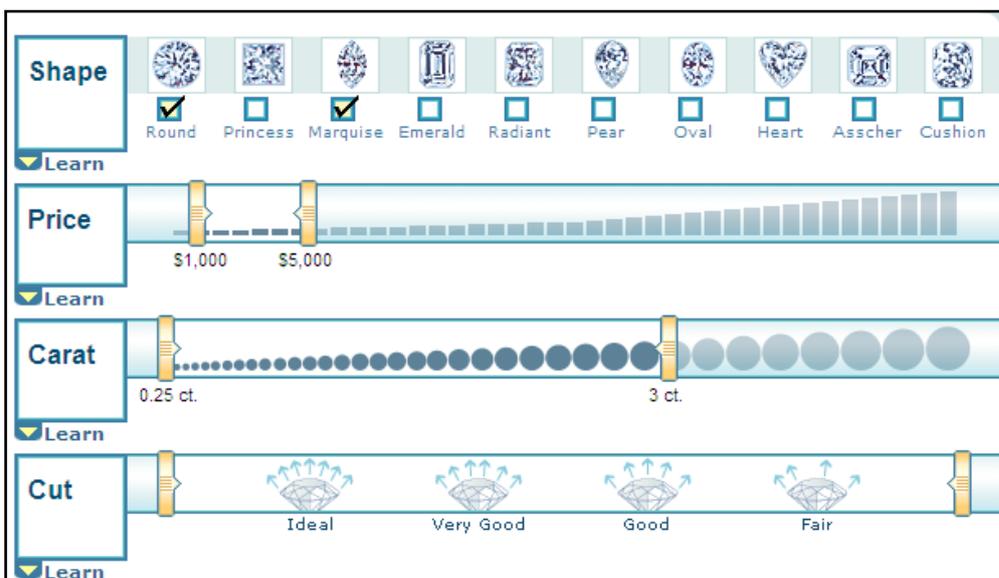


Abb. 21: Amazon.com Diamond Search bei aktiviertem JavaScript

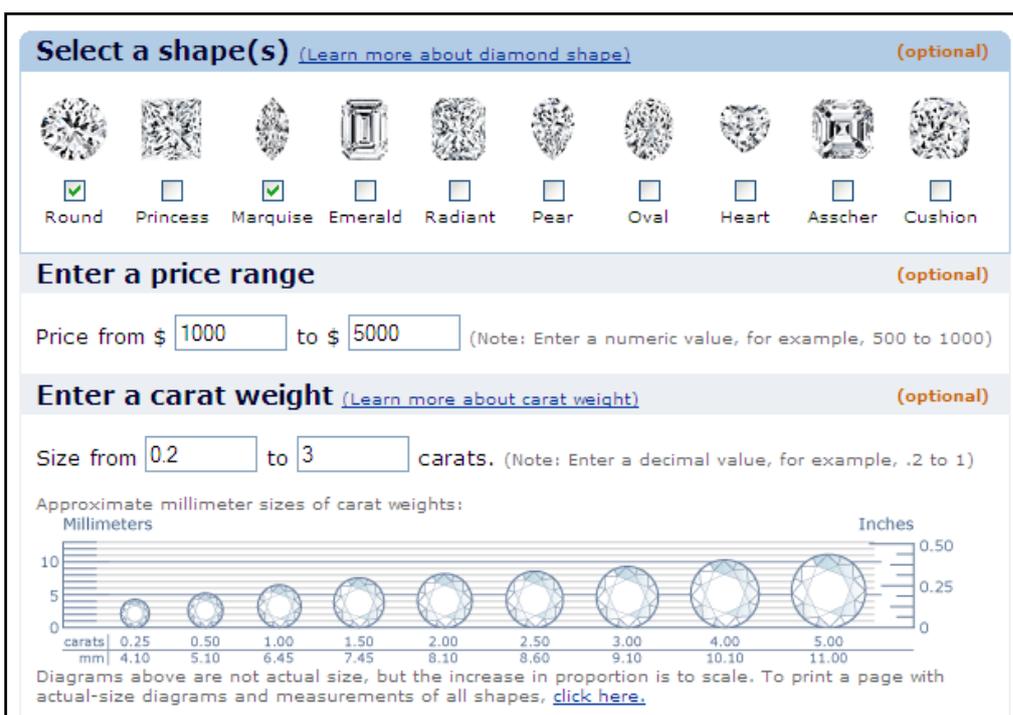


Abb. 22: Amazon.com Diamond Search bei deaktiviertem JavaScript

Eine Webseite wirklich barrierefrei zu gestalten ist nur mit erheblich größerem Aufwand möglich, denn Barrierefreiheit bedeutet mehr als nur das Umgehen von JavaScript. Es beinhaltet genauso das Verwenden von lesbaren Schriften, abgestimmten Farben, logisch sinnvoller Navigation und einfacher Benutzbarkeit (Usability). Dies alles zu berücksichtigen ist nicht einfach und erfordert auch einiges an Know-how im Bereich der barrierefreien Webprogrammierung.

Vor diesem Problem stehen aber nicht nur Entwickler von AJAX-Anwendungen, auch Entwickler von traditionellen Webseiten haben mit den Problemen der Barrierefreiheit zu kämpfen. So müsste zum Beispiel auch komplett auf den Einsatz von Flash-Technologie verzichtet werden, da sehbehinderte Personen, welche den Inhalt der Webseite über eine Braillezeile wahrnehmen oder einen Screenreader verwenden, nichts mit Flash-Objekten anfangen können.

Barrierefreiheit ist ein aktuelles Thema auch aufgrund der österreichischen Gesetzeslage und des Drucks der Europäischen Union, welche die Gleichberechtigung von behinderten Personen immer stärker vorantreibt. Webentwickler, welche Webanwendungen programmieren, die barrierefrei sein müssen, sollten daher auf den Einsatz von AJAX verzichten oder eine alternative Webseite mit AJAX-Funktionalität anbieten.

Für weiterführende Informationen sei auf die Webseite der „Aktion Mensch-Initiative für barrierefreies Web“ verwiesen (<http://www.einfach-fuer-alle.de>).

4.2.9. Suchmaschinenindizierung

Ein weiteres Problem von AJAX-Webseiten ist die mangelnde Verarbeitung der Seite durch Suchmaschinen bzw. ihrer Webcrawler. Da bei AJAX Inhalte dynamisch erzeugt werden, kommt es zu Problemen, wenn ein Webcrawler eine Webseite indiziert. Der Webcrawler ist nicht in der Lage, den Inhalt dynamisch nachzuladen, da er keinen JavaScript Code ausführen kann. Da JavaScript eine clientseitige Skriptsprache ist, welche im Webbrowser ausgeführt wird, müsste der Webcrawler ebenso JavaScript unterstützen, wenn er denselben Inhalt sehen wollte wie der User. So wie es derzeit aussieht, werden aber auch in naher Zukunft Webcrawler keinen JavaScript Code ausführen können. Auch Google weist in seinem Webmaster Help Center auf dieses Problem hin:

When Googlebot indexes a page containing Javascript, it will index that page but it cannot follow or index any links hidden in the Javascript itself. (Google, 2007)

Ein weiteres Problem ist, dass sich bei der Verwendung von AJAX die URL nicht ändert, wenn Inhalte nachgeladen werden (siehe Kapitel 4.2.4 - Eingeschränkte Bookmark-Funktion). Der Webcrawler, welcher auf Links und somit neue URLs angewiesen ist, steht hier vor einem Problem, da er nicht an den kompletten Inhalt der Webseite herankommt. Im schlimmsten Fall indiziert der Webcrawler nur die Startseite, jeglicher Inhalt, der dynamisch nachgeladen wird, bleibt aber für den Webcrawler unentdeckt.

Um dies praktisch zu verdeutlichen, wird das Beispiel aus Kapitel 3.6 herangezogen und in dem textbasierten Webbrowser Lynx geöffnet. Der Webbrowser, welcher ähnlich arbeitet wie Webcrawler, zeigt zwar den Button aus dem Beispiel an, jedoch lässt sich dieser nicht anklicken. Somit ist es auch für einen Webcrawler nicht möglich, den Inhalt, welcher generiert wird, nachdem der User auf den Button geklickt hat, weiter zu verfolgen. Der Grund dafür ist auch hier die fehlende JavaScript-Unterstützung, denn klickt der User auf den Button, wird dies mit dem Event-Handler `onclick` abgefangen und die Funktion `getZitat()` (siehe Listing 3) aufgerufen. Dies ist für einen Webcrawler bis jetzt ein unlösbares Problem.

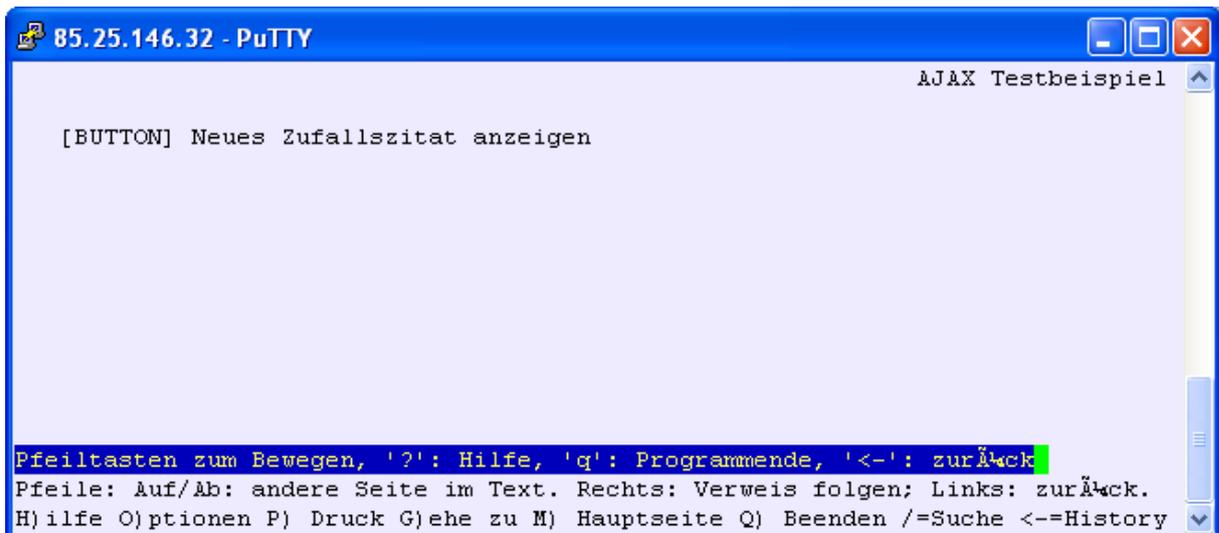


Abb. 23: AJAX-Beispiel aus Kapitel 3.6 geöffnet im Lynx-Webbrowser

Eine Möglichkeit, wie Suchmaschinen doch zu dem kompletten Inhalt der Webseite kommen, wäre das sogenannte Cloaking, dabei wird dem Webcrawler eine andere Webseite präsentiert als dem User. Diese Methode, die früher gerne eingesetzt wurde, ist aber mittlerweile bei den Suchmaschinenbetreibern sehr verpönt, da sie in der Vergangenheit leider zu oft missbräuchlich verwendet wurde. So täuschten Spammer den Suchmaschinen einen anderen Inhalt vor, als die Webseiten dann tatsächlich enthalten haben. Google äußert sich dazu folgendermaßen:

If your site contains elements that aren't crawlable by search engines (such as Flash, Javascript, or images), you shouldn't provide cloaked content to search engines. (Google, 2007)

Eine Lösung dieses Problems ist nicht ganz einfach. Ist eine Indizierung der Webseite durch eine Suchmaschine unbedingt erforderlich, so sollte gut überlegt werden, ob der Einsatz von AJAX überhaupt sinnvoll ist oder nicht (siehe dazu auch Kapitel 5.1 - AJAX sinnvoll oder nicht?). Am besten wäre es, dem User (und somit auch der Suchmaschine) eine alternative Webseite anzubieten, wo er dieselbe Funktionalität hat, jedoch ohne Verwendung von AJAX. Wie dies zu lösen ist, zeigt Amazon mit seiner Diamantring-Suche aus dem vorigen Kapitel. Amazon (2008) weist hier seine Kunden mit der Meldung „*If you encounter problems with this page, try our basic diamond search*“ darauf hin, dass sie die Webseite auch nutzen können, wenn es Probleme mit der Darstellung gibt (zum Beispiel wegen deaktiviertem JavaScript). Klickt der User auf „basic diamond search“, kann er die alternative Suchfunktion verwenden (vergleich Abb. 21 mit Abb. 22). Suchmaschinen können somit ebenfalls dem Link folgen und die alternative Webseite vollständig indizieren.

Wird AJAX im Intranet oder in passwortgeschützten Bereichen verwendet, so kann das Problem der Suchmaschinenindizierung getrost ignoriert werden, da hier ohnehin keine Indizierung durch Suchmaschinen gewünscht ist.

4.3. Web 2.0 Akzeptanz-Umfrage

Im Rahmen des Unterrichtsfaches „IT-Markt“ an der Fachhochschule Technikum Wien wurde vom 27.11.2007 bis 17.12.2007 eine Online-Umfrage bei österreichischen Unternehmen durchgeführt. Die Umfrage hatte den Titel „IT in österreichischen Produktions- und Dienstleistungs-Unternehmen“ und richtete sich an die IT-Verantwortlichen der jeweiligen Firmen. Innerhalb von drei Wochen füllten 144 IT-Verantwortliche vollständig den Fragebogen mit insgesamt 38 Fragen aus. Zwei dieser Fragen beinhalten das Web 2.0-Thema und werden daher genauer betrachtet.

4.3.1. Frage 1: Haben Sie einen öffentlichen Webauftritt?

Die erste Frage rund um das Thema Web 2.0 lautete: „Haben Sie einen öffentlichen Webauftritt?“ Als Antwortmöglichkeit konnten die Teilnehmer unter anderem ankreuzen, dass sie in ihrem Webauftritt bereits neueste Web 2.0-Technologie einsetzen. Mit dieser Frage sollte herausgefunden werden, ob österreichische Unternehmen dem aktuellen Trend nachgehen und bereits auf die Web 2.0-Technologie setzen.

Haben Sie einen öffentlichen Webauftritt?	Ergebnis	Prozent
Ja, jedoch nur statische Webseiten (reines HTML)	43	30%
Ja, mit dynamischen Webseiten (z.B. PHP, .NET, Java-Applets)	91	63%
Ja, wir verwenden neueste Web 2.0 Technologie (z.B. AJAX)	4	3%
Nein	6	4%
Gesamtergebnis	144	100%

Tabelle 3: Ergebnis der Umfrage: Haben Sie einen öffentlichen Webauftritt?

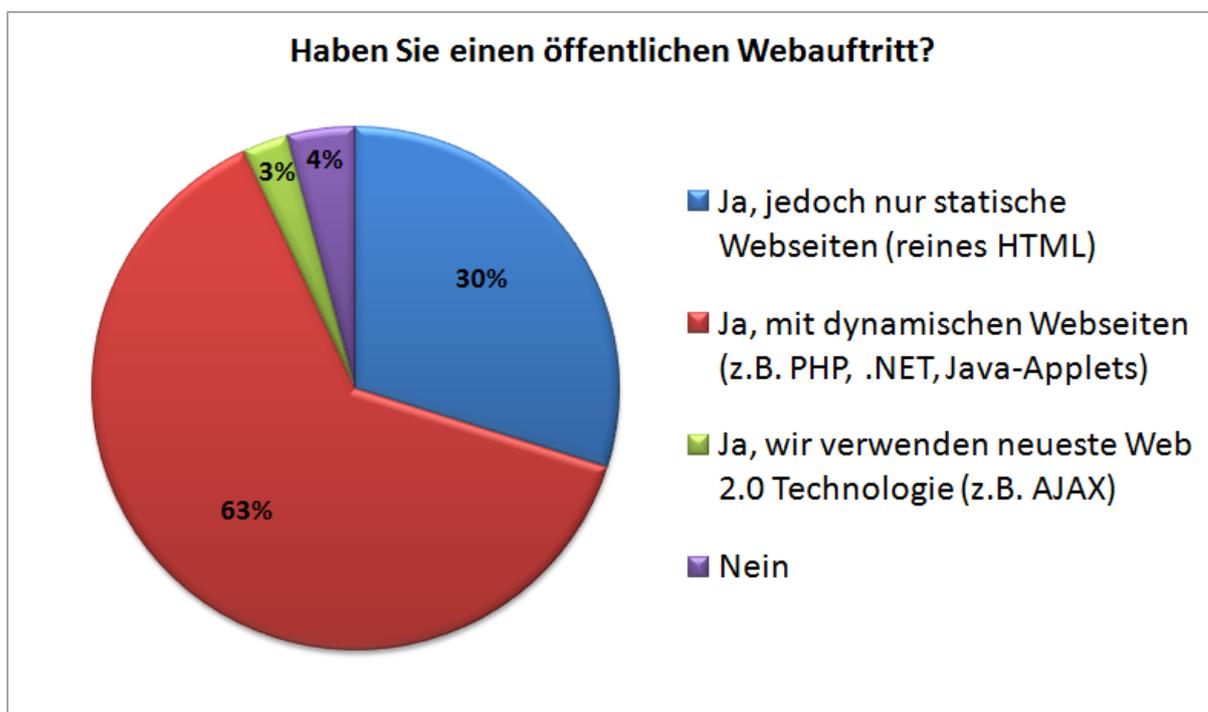


Abb. 24: Umfrage Diagramm: Haben Sie einen öffentlichen Webauftritt?

Die Auswertung dieser Frage war ernüchternd. Von den 144 teilnehmenden Unternehmen verfügen bereits 96% über einen öffentlichen Webauftritt, jedoch nur 3% der Unternehmen verwenden bereits neuartige Technologien wie zum Beispiel AJAX.

Dieser Prozentsatz ist für die Tatsache, dass es AJAX bereits seit dem Jahr 2005 gibt, ziemlich gering, bietet es doch interessante Vorteile für das Unternehmen und den User. Möglicherweise liegt die geringe Akzeptanz an dem Standort Österreich, da viele österreichische Unternehmen erst einmal abwarten, wie sich dieser Hype entwickelt und nicht gleich alles umsetzen, was aus dem Land der unbegrenzten Möglichkeiten nach Europa dringt.

Es sollte aber bedacht werden, dass Web 2.0 mittlerweile anerkannt ist und es kein Hype oder eine Modeerscheinung ist (so auch Jakubetz, 2006).

4.3.2. Frage 2: Wie sieht die künftige Entwicklung aus?

In der zweiten Frage ging es um die zukünftige Entwicklung der IT-Abteilung, dabei wurde auch die Frage gestellt, ob ein verstärkter Einsatz von Webtechnologien wie zum Beispiel AJAX geplant ist. Die Antwortmöglichkeiten waren dabei auf „Ja“ oder „Nein“ beschränkt.

Verstärkter Einsatz von Webtechnologien (Web 2.0, AJAX)?	Ergebnis	Prozent
Ja	63	44%
Nein	81	56%
Gesamtergebnis	144	100%

Tabelle 4: Ergebnis der Umfrage: Wie sieht die künftige Entwicklung aus?

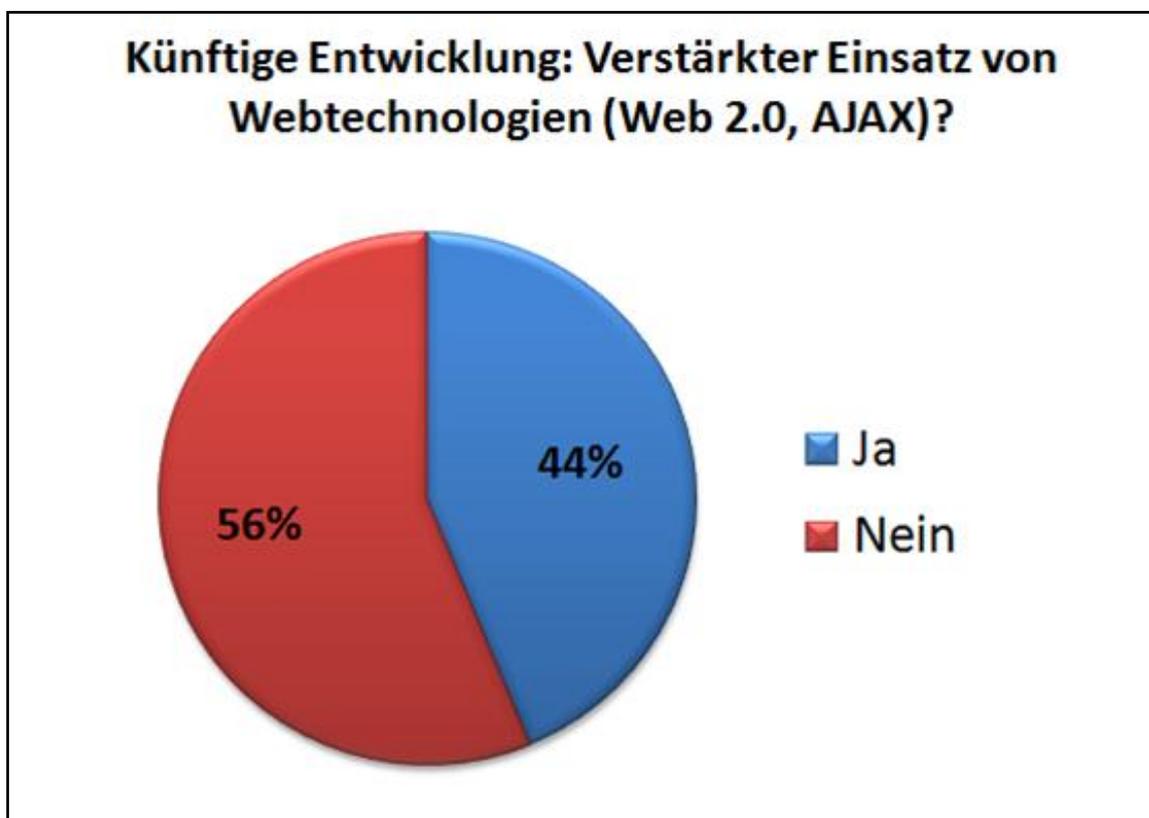


Abb. 25: Umfrage Diagramm: Wie sieht die künftige Entwicklung aus?

Die Auswertung der Frage ergab, dass die überwiegende Mehrheit der Teilnehmer in Zukunft keine Webtechnologien wie AJAX einsetzen will. Jedoch sind immerhin 44% der Unternehmen bereit, in Zukunft einen Versuch zu wagen. Dies ist ein erfreulicher Prozentsatz und zeigt, dass sich Unternehmen mit neuer Technologie beschäftigen und diese auch einsetzen wollen.

Betrachtet man die Umfrage im Kontext, ist unter anderem interessant festzustellen, dass vor allem die älteren IT-Verantwortlichen bei dieser Frage mit „Nein“ geantwortet haben. Unter den Befragten gab es zwei Personen, welche über 60 Jahre alt sind. Beide haben einen verstärkten Einsatz von Web 2.0 verneint. Auch 65,8% der Personen im Alter von 46-60 Jahren kamen zu demselben Entschluss. Bei den jüngeren IT-Verantwortlichen im Alter von 20-25 Jahren beantworteten dagegen 75% die Frage mit „Ja“.

Aufgrund dieses Ergebnisses lässt sich erkennen, dass vor allem ältere Personen den Einsatz von neuen Technologien meiden, hingegen junge Personen kein Problem mit neuer Technologie haben und auch deren Einsatz planen.

5. Diskussion

In den vorigen Kapiteln wurden die Vor- und Nachteile der AJAX-Technologie ausführlich erläutert. Werden nun die Vorteile gegenüber den Nachteilen abgewogen, so kann grundsätzlich gesagt werden, dass die Vorteile hier überwiegen und einem Einsatz der AJAX-Technologie in Webanwendungen nichts im Wege steht. Das Argument, dass AJAX bei abgeschaltetem JavaScript und bei älteren Webbrowsern nicht mehr funktioniert, kann hier nicht geltend gemacht werden, da es sich um einen minimalen Prozentsatz handelt, der in der Praxis vernachlässigbar ist, wenn ein paar Regeln beachtet werden. So sollte der User, welcher JavaScript deaktiviert hat, darauf hingewiesen werden, dass er manche Funktionen der Webanwendung nicht nutzen kann.

Nach dem heutigen Stand (Februar 2008) können etwa 97 Prozent aller Internetnutzer AJAX verwenden. In den nächsten Jahren wird dieser Prozentsatz auf fast 99 Prozent ansteigen, da Webbrowser, welche kein XMLHttpRequest-Objekt erzeugen können und somit AJAX nicht unterstützen, in nächster Zeit aussterben und die neuen Webbrowser die alte Generation vertreiben werden. Auch JavaScript wird immer weniger deaktiviert, da die User mittlerweile mitbekommen haben, dass von JavaScript kaum Gefahr ausgeht. Unter Windows wird es in einer Sandbox ausgeführt und erlaubt dadurch keinen Zugriff auf das Dateisystem.

Die Umfrage aus Kapitel 4.3 hat gezeigt, dass sehr wohl Interesse an neuen Webtechnologien (insbesondere AJAX) vorhanden und ein Einsatz in den nächsten Jahren auch geplant ist. Einen produktiven Einsatz dieser Technologien gibt es jedoch erst bei 3% der befragten Unternehmen. Dieser Prozentsatz wird in den nächsten Jahren aufgrund der Umfrageergebnisse sicherlich um einiges zunehmen.

5.1. AJAX sinnvoll oder nicht?

Bevor AJAX jedoch in einer Webanwendung eingesetzt wird, sollte gut überlegt werden, ob der Einsatz überhaupt sinnvoll ist. So kann es sein, dass der Einsatz kaum Vorteile bringt und die Funktionalität der Anwendung den Benutzer nur verwirrt (vor allem, wenn keine Aktivitätsindikatoren verwendet werden). Im Folgenden findet sich eine kurze Liste von Anwendungen, wo der Einsatz von AJAX sinnvoll ist und wo nicht.

Einsatz von AJAX sinnvoll:

- Bei Anwendungen, wo eine schnelle Response-Zeit gefordert ist (zum Beispiel beim Nachladen von Information wie bei Google Maps oder Google Finance)
- Zwecks Plausibilitätsüberprüfung bei der Eingabe von Daten in Webformularen (zum Beispiel kann hier direkt überprüft werden, ob eine eingegebene E-Mail-Adresse gültig oder ein Passwort sicher genug ist)
- Wenn schneller Datenaustausch von Nutzdaten gefordert ist (zum Beispiel bei einem Chat oder Online-Messenger wie Meebo.com)
- Bei Votings (nach einem Voting kann gleich das Ergebnis angezeigt werden, ohne die Webseite neu zu laden)
- Beim Tagging in Form einer Autovervollständigung von Tags
- Bei Ratings (der abgegebene Wert kann auf Knopfdruck gespeichert werden)
- In Bildergalerien zum dynamischen Nachladen von Bildern
- Bei einem Dateiuupload

Einsatz von AJAX nicht sinnvoll:

- Bei Webseiten, die barrierefrei sein müssen (zum Beispiel Webseiten von Regierungen, Behörden und Ämtern)
- Bei Webseiten, deren Zielgruppe ältere Personen sind (die Interaktivität und Neuartigkeit von AJAX kann ältere Personen überfordern)
- Wenn große Textblöcke nachgeladen werden (zum Beispiel beim Paging)
- Bei der Verwendung von einfachen Formularen
- Bei der Anzeige von Suchergebnissen
- In Navigationselementen

Diese Liste erhebt nicht den Anspruch auf Vollständigkeit. Sollte eine Anwendung hier nicht aufgelistet sein, so muss anhand der Vor- und Nachteile aus Kapitel 4 individuell überlegt werden, ob der Einsatz für den spezifischen Fall sinnvoll ist oder nicht.

5.2. Verwendung von Frameworks

Man soll das Rad nicht neu erfinden, heißt eine bekannte Redewendung, welche auch auf die Erstellung von AJAX-Applikationen zutrifft. Die AJAX-Engine, welche das XMLHttpRequest-Objekt erstellt und verwaltet, braucht in der heutigen Zeit nicht mehr selbst geschrieben werden. Denn seit Beginn des AJAX-Booms schießen AJAX-Frameworks unterschiedlichster Couleur wie Pilze aus dem Boden (Mintert & Leisegang, 2007, S. 110) und bieten dem Webentwickler eine großartige Hilfestellung.

Frameworks oder Bibliotheken (Libraries), wie sie auch noch genannt werden, bieten dem Webentwickler eine Sammlung von Funktionen und Methoden für die verschiedensten Aufgaben. Darunter fällt zum Beispiel das Handling von Objekten, Event-Verwaltung oder das Erstellen von visuellen Effekten.

Vorteile bieten Frameworks zuhauf, so steckt in den Frameworks oft monate- oder jahrelange Entwicklungsarbeit von mehreren Programmierern, die auch die Funktionalität des Frameworks in den gängigsten Webbrowsern testen und Fehler beseitigen. So funktionieren die Frameworks meist besser und fehlerfreier als der selbst geschriebene Code.

Aufgrund der Komplexität mancher Frameworks eignen sie sich jedoch nicht für jede Webanwendung. Bei kleineren Webanwendungen mit wenig AJAX-Technologie lohnt sich der Einsatz meist nicht, da die Einarbeitungszeit in das jeweilige Framework zu viel Zeit in Anspruch nehmen würde. Werden jedoch viele oder große Webanwendungen entwickelt, so kann mit der Verwendung eines Frameworks schneller und effizienter gearbeitet werden. So sollte bei der Entwicklung einer Webanwendung vorher überlegt werden, ob der Einsatz eines Frameworks vorteilhaft ist oder nicht.

Im Folgenden finden sich die fünf bekanntesten und meist verwendeten Frameworks für Entwicklung von AJAX-Anwendungen:

- Prototype (<http://www.prototypejs.org>)
- Script.aculo.us (<http://script.aculo.us>)
- Dojo Toolkit (<http://dojotoolkit.org>)
- Plex Toolkit (<http://www.plextk.or>)
- MooTools (<http://www.mootools.net>)

Eine umfangreiche Liste von weiteren Frameworks findet sich auf <http://ajaxpatterns.org>.

5.3. Ausblick auf das Web 3.0

Wie wird es weitergehen, wenn sich das Web 2.0 weitgehend etabliert hat und wie wird das Web 3.0 aussehen? Fragen, die nicht einfach zu beantworten sind, da niemand hundertprozentig wissen kann, was die Zukunft bringt. Trotzdem sollen hier einige Gedanken und Ideen zum Thema Web 3.0 eingebracht werden.

„*Web 1.0 was dial-up, 50K average bandwidth, Web 2.0 is an average 1 megabit of bandwidth and Web 3.0 will be 10 megabits of bandwidth all the time, which will be the full video Web, and that will feel like Web 3.0*“ (Farber, 2006) sagt Reed Hastings, Gründer und CEO von Netflix, dem weltgrößten Online-DVD-Vermieter, auf der TechNet Summit im November 2006 und prophezeit damit, dass ein beschleunigtes Internet ausschlaggebend für das Web 3.0 ist. Dass die Internet-Bandbreite entscheidend für Entwicklung neuer Technologien und das Nutzungsverhalten der User ist, steht außer Frage. So haben, wie Hastings richtig erkannt hat, höhere Bandbreiten zur Entwicklung des Web 2.0 positiv beigetragen (siehe dazu auch Kapitel 3.3.1 - Höhere Downloadgeschwindigkeiten).

Der alleinige Geschwindigkeitszuwachs kann aber nicht das komplette Web 3.0 auszeichnen, so bietet zum Beispiel UPC, ein österreichischer führender Anbieter von Internet-Services, bereits Internetanschlüsse mit bis zu 25 Mbit/s (UPC, 2008). Von Web 3.0 spricht aber deswegen derzeit noch niemand. Höhere Bandbreiten fördern allerdings die Entwicklung hin zu einem beschleunigten und interaktiven Web und damit zu neuen Möglichkeiten. Video-on-Demand, Video-Streaming bzw. komplettes Fernsehen über das Internet ist teilweise schon Realität und wird in Zukunft sicher vermehrt genutzt werden. So wird es möglich sein, sich komplette Filme legal und in guter Qualität auf Knopfdruck aus dem Internet herunterzuladen und anzusehen. Finanzieren werden sich solche Angebote durch Abodienste oder Werbeeinblendungen innerhalb der Filme.

Aber nicht nur die Bandbreite wird zunehmen, auch die Nutzungshäufigkeit wird ansteigen. Mobiles Internet, surfen über das Handy und billigere Tarife tragen dazu bei, dass der User jederzeit und überall online sein kann. Laut einer Umfrage der Firma Result (Result, 2007) verwenden bereits 43,5% der befragten Personen das Internet, um sich über aktuelle Ereignisse zu informieren. Fernsehen und Tageszeitungen nehmen nur mehr Platz zwei und drei ein.

Mit den Worten „*Der gedruckte Brockhaus ist tot*“ beginnt die Financial Times Deutschland (Knappmann & Lachmann, 2008) ihren Artikel über das angekündigte Ende der Brockhaus Enzyklopädie und bescheinigt dem Brockhaus Verlag damit keine rosige Zukunft. Auch der Verlagschef Ulrich Granseyer und Verlagssprecher Klaus Holoch bekennen mit den Worten „*Wir mussten einsehen, dass die Leute Informationen im Internet suchen*“ und „*Die Zeit, in der man sich eine hervorragende Enzyklopädie von anderthalb Meter Umfang ins Regal stellt, um sich dort herauszusuchen, was man wissen will, scheint vorbei zu sein*“ das Ende von Print-Lexikas. Somit hat es Wikipedia innerhalb von nur sieben Jahren geschafft, dass eine fast 200 Jahre lange Ära der Brockhaus Enzyklopädie dem Ende zugeht und es eine Neuauflage des Lexikons, in gedruckter Form, mit großer Wahrscheinlichkeit nicht mehr geben wird.

An diesem aktuellen Beispiel wird deutlich, dass immer mehr Menschen das Internet als Informationsquelle nutzen. So werden vor allem Online-Lexikas wie Wikipedia auch im Web 3.0 einen wichtigen Platz einnehmen. Durch verfeinerte Suchalgorithmen und semantische

Anreicherung der Artikel könnte es sogar möglich sein, dass eine Anfrage wie zum Beispiel „Wann starb Falco?“ auf Anhieb das richtige Ergebnis liefert und das Todesdatum anzeigt.

Eine weitere Entwicklung, welche das Web 3.0 auszeichnen wird, ist die schon in Kapitel 3.3.3 erwähnte Verschmelzung von Desktopanwendungen und Webanwendungen. Durch AJAX sind Webanwendungen dynamischer und interaktiver geworden und ähneln in ihrer Handhabung und Funktionalität den Desktopanwendungen. In Zukunft werden somit immer mehr Desktopanwendungen in das Web ausgelagert werden.

Möglich wird es auch sein, ein komplettes Betriebssystem samt Office Paket online zur Verfügung zu stellen, für dessen Nutzung nur mehr ein Webbrowser benötigt wird. Die Daten werden online via Betriebssystem erstellt und gespeichert, ein Zugriff ist jederzeit und von überall auf der Welt möglich. Was heute vielleicht noch unrealistisch klingt, wird in einigen Jahren bereits Realität sein. Webanwendungen wie OOS, eyeOS oder PHP Explorer arbeiten in diese Richtung und bieten schon heute ansehnliche Online-Betriebssysteme (siehe Abb. 26).

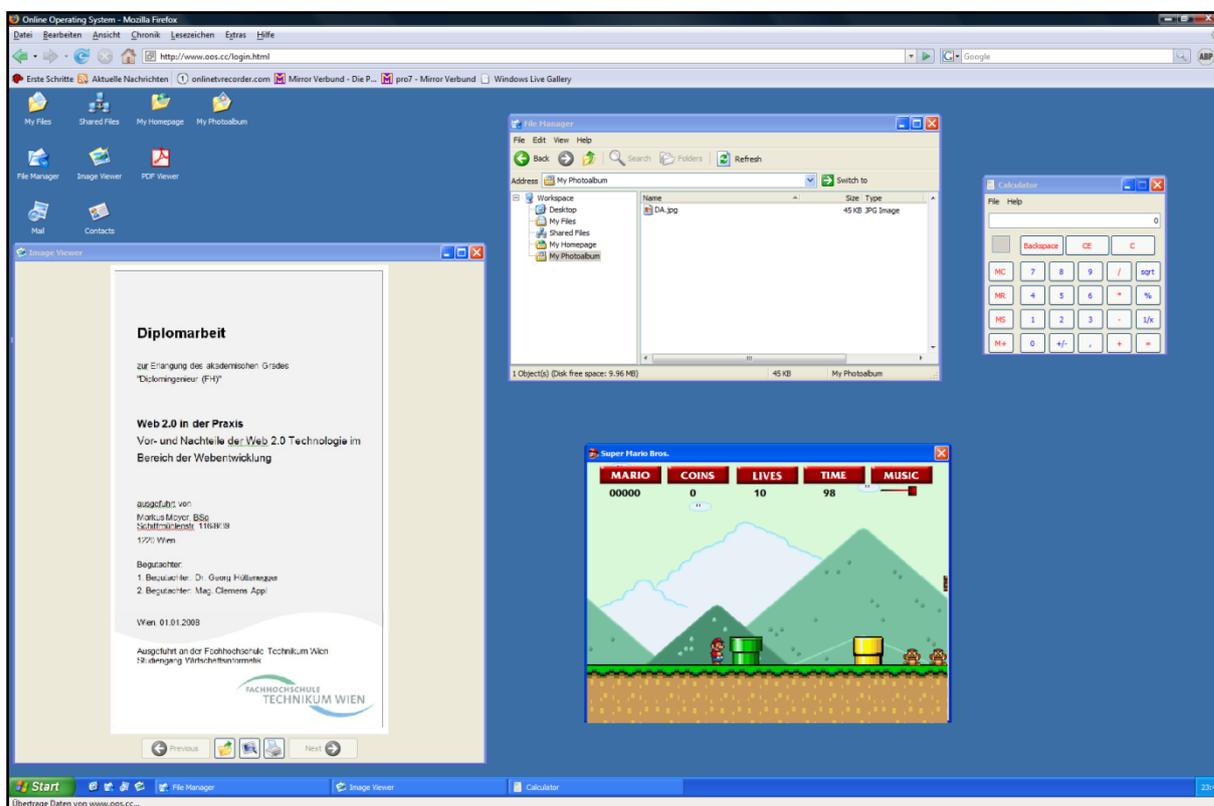


Abb. 26: Screenshot des Online-Betriebssystems OOS.cc

Internet immer und überall mit noch höheren Bandbreiten als bisher, so wird die Zukunft aussehen. Bücher, Zeitschriften und Fernsehen werden sich in das Web der Zukunft verlagern und der Computerarbeitsplatz wird bei vielen Menschen zu einem Thin Client werden. In schätzungsweise 5 bis 10 Jahren wird es soweit sein.

5.4. Fazit

Abschließend lässt sich sagen, dass AJAX eine faszinierende Technologie mit hervorragenden Möglichkeiten ist, deren Potenzial derzeit noch nicht vollkommen ausgeschöpft ist. Komplexe AJAX-Anwendungen werden derzeit hauptsächlich von größeren Firmen entwickelt. Durch die Verwendung von Frameworks soll aber auch der einzelne Webentwickler interessante AJAX-Anwendungen erstellen können.

So kann gespannt in die Zukunft geblickt und beobachtet werden, wie Webanwendungen Desktopanwendungen immer ähnlicher werden und vielleicht eines Tages sogar miteinander verschmelzen.

Abbildungsverzeichnis

Abb. 1: Verbindungstechniken der Unternehmen für den Internetzugang 2001-2007 (Djahangiri & Edelhofer, 2007)	7
Abb. 2: Prozent der weltweiten Internet-User welche diese Webseiten besuchten.....	9
Abb. 3: Vandalismus beim Falco-Artikel auf Wikipedia (2008)	12
Abb. 4: RSS-Logo.....	13
Abb. 5: Veröffentlichungszyklus Flickr Versus Microsoft (Musser & O'Reilly, 2006)	14
Abb. 6: Die Long Tail Kurve (Anderson, 2006).....	15
Abb. 7: Tagging am Beispiel des Musikers Falco auf der Webseite last.fm.....	16
Abb. 8: Statistik der meist genutzten APIs auf programmableweb.com.....	17
Abb. 9: Beispiel für ein Mashup – Verwendung von Google Maps in einer Webseite.....	18
Abb. 10: Beispiel DOM-Struktur ausgelesen mit der IE Developer Toolbar	22
Abb. 11: Beispiel von Google Suggest (Google, 2008)	24
Abb. 12: Traditionelles Modell von Webanwendungen im Vergleich zum AJAX-Model (Garrett, 2005)	25
Abb. 13: Screenshot des AJAX-Beispiels.....	29
Abb. 14: Zitate Datenbank (http://www.evangeliums.net).....	31
Abb. 15: Verwendung von JavaScript (webhits, 2008)	34
Abb. 16: Warnhinweis im Onlineshop der Firma Zimek.....	35
Abb. 17: Der HTML-Anker nach dem #-Zeichen in URLs lässt sich als Status-ID missbrauchen, sodass sich deren Zustand in einer Bookmark erfassen lässt (Hauser, 2007).	36
Abb. 18: Warnhinweis beim Erzeugen des XMLHttpRequest-Objektes (getestet im Opera 7.52).....	37
Abb. 19: Eingblendeter Permalink bei Google Maps	38
Abb. 20: Beispiel für grafische Aktivitätsindikatoren	38
Abb. 21: Amazon.com Diamond Search bei aktiviertem JavaScript	42
Abb. 22: Amazon.com Diamond Search bei deaktiviertem JavaScript	42
Abb. 23: AJAX-Beispiel aus Kapitel 3.6 geöffnet im Lynx-Webbrowser.....	44
Abb. 24: Umfrage Diagramm: Haben Sie einen öffentlichen Webauftritt?	45
Abb. 25: Umfrage Diagramm: Wie sieht die künftige Entwicklung aus?	46
Abb. 26: Screenshot des Online-Betriebssystems OOS.cc	51
Abb. 27: Umfrage Diagramm: Können Sie sich unter dem Begriff Web 2.0 etwas vorstellen?	65
Abb. 28: Umfrage Diagramm: Durchschnittsalter der Nutzer von Web 2.0-Anwendungen ...	65
Abb. 29: Umfrage Diagramm: Frauen und Männer als Nutzer von Web 2.0-Anwendungen .	66
Abb. 30: Umfrage Diagramm: Nutzung von Web 2.0-Anwendungen.....	66
Abb. 31: Umfrage: Mediennutzungsverhalten	67

Abkürzungsverzeichnis

ADSL	Asymmetrical Digital Subscriber Line
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ASP	Active Server Page
BGStG	Bundes-Behindertengleichstellungsgesetz
BITV	Barrierefreie Informationstechnik-Verordnung
BKA	Bundeskanzleramt
CEO	Chief Executive Officer
CMS	Content Management System
CSS	Cascading Style Sheet
DOM	Document Object Model
DSL	Digital Subscriber Line
DVD	Digital Versatile Disk
E-GovG	E-Government-Gesetz
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
ISDN	Integrated Services Digital Network
JSP	Java Server Pages
MS	Microsoft
PHP	PHP Hypertext Preprocessor
RSS	Really Simple Syndication
UGC	User Generated Content
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language for Transformation

Tabellenverzeichnis

Tabelle 1: Vergleich - Desktopanwendungen versus Webanwendungen	8
Tabelle 2: AJAX Browserunterstützung.....	26
Tabelle 3: Ergebnis der Umfrage: Haben Sie einen öffentlichen Webauftritt?	45
Tabelle 4: Ergebnis der Umfrage: Wie sieht die künftige Entwicklung aus?	46

Listingverzeichnis

Listing 1: Beispiel einer RSS-Datei von zdnet.de	13
Listing 2: HTML-Code (Dateiname test.html)	21
Listing 3: Style Sheet (Dateiname style.css)	21
Listing 4: Referenz auf das Style Sheet	21
Listing 5: Beispiel für eine XML-Datei	22
Listing 6: Beispiel für ein JavaScript.....	23
Listing 7: AJAX-Beispiel – Darstellung der HTML-Grundstruktur.....	27
Listing 8: AJAX-Beispiel – Erzeugen des XMLHttpRequest-Objekts	27
Listing 9: AJAX-Beispiel – Herstellen der asynchronen Verbindung.....	28
Listing 10: AJAX-Beispiel – Quellcode der PHP-Datei	28
Listing 11: AJAX-Beispiel – Verarbeiten der Antwort des Servers.....	29
Listing 12: Aufruf der open-Methode des XMLHttpRequest-Objekts	32
Listing 13: Beispiel für die Initialisierung des XMLHttpRequest-Objekts	37
Listing 14: Beispiel für das Einblenden eines Aktivitätsindikators	39
Listing 15: Quellcode der HTML-Datei aus dem AJAX-Beispiel (Kapitel 3.6)	63
Listing 16: Quellcode des PHP-Skripts aus dem AJAX-Beispiel (Kapitel 3.6).....	64
Listing 17: Quellcode des PHP-Skripts aus dem AJAX-Beispiel (Kapitel 3.6) mit einer Datenbankabfrage.....	64

Glossar

ActiveX: Von Microsoft entwickelte Antwort auf Java. Damit ist es möglich, interaktive Elemente in Webseiten zu integrieren.

ADSL: Technologie zur schnellen Datenübertragung über Kupferkabel, wird auch als Breitbandinternetanschluss bezeichnet.

AJAX: Technologie zur asynchronen Datenübertragung zwischen Webbrowser und Webserver.

AJAX-Engine: JavaScript Code, welcher das XMLHttpRequest-Objekt verwendet, um eine Verbindung mit dem Webserver herzustellen.

API: Schnittstelle, um mit Hardware oder Software kommunizieren zu können.

Black-Box-Test: Testverfahren, wobei keine genau Information über den inneren Aufbau des Systems (zum Beispiel Quellcode) vorhanden ist.

Blog: Abkürzung für Weblog, wird auch als Internettagebuch bezeichnet. Personen veröffentlichen dabei ihre Meinungen auf Webseiten.

Cross Browser Compatibility: Bezeichnet die Fähigkeit, dass unterschiedliche Webbrowser die selbe Webseite gleich (ohne Darstellungsfehler) wiedergeben.

CSS: Stilvorlagen für die Darstellung von Webseiten. Dabei wird versucht, Stil und Inhalt voneinander zu trennen.

Event-Handler: Lösen nach bestimmten Ereignissen (zum Beispiel Tastendruck, Mausklick, Laden einer Webseite udgl.) bestimmte Funktionen aus.

Flickr: Riesiges Fotoportal im Internet. Benutzer können kostenlos ihre Fotos uploaden, verwalten und mit Kommentaren und Tags versehen. URL: <http://www.flickr.com>

Framework: Sammlung von Klassenbibliotheken zur einfachen Verwendung von wiederkehrenden Programmteilen.

HTML: Ist eine Auszeichnungssprache für das Erstellen von Webseiten. Damit werden Texte, Bilder, Links udgl. strukturiert.

iTunes Store: Onlineshop von Apple zum Kauf von Musikstücken und Filmen. Benötigt das kostenlose Programm iTunes.

Last.fm: Populäres Internetradio mit sozialer Vernetzung. Dem User werden dabei aufgrund seiner Hörgewohnheit und der anderer User neue Musikstücke vorgeschlagen. URL: <http://www.last.fm>

Lynx: Textbasierter Webbrowser auf Linux- und Unix-Systemen, welcher keine grafischen Elemente anzeigen kann. Mit ihm kann das Verhalten von Screenreadern und Webcrawlern überprüft werden.

Mashups: Anwendungen (Webseiten), welche frei verfügbaren Inhalt anderer Webseiten oder Datenquellen in ihre eigene Webseite integrieren, um damit ein eigenes Angebot zu

erstellen.

Pageing: Bezeichnet das Seitenweise weiterblättern in Webseiten, wenn zu viele Datensätze ausgegeben werden und diese auf mehreren Seiten angezeigt werden.

Permalink: Eindeutiger, langlebiger Verweis (Link) auf eine Webseite inklusive deren Zustand beim Generieren des Verweises.

PHP: Sehr populäre, serverseitige Skriptsprache für die Erstellung von dynamischen Webseiten und Webanwendungen.

Picasa: Kostenfreie Bildverwaltungssoftware von Google. Teilweise können die Bilder auch bearbeitet werden. URL: <http://picasa.google.com>

Plugin: Softwareprogramm, welches sich in eine andere Software einklinkt und dadurch Zusatzfunktionen bietet.

Sandbox: Abgeschirmter Bereich innerhalb des Betriebssystems, in welchem bösartige Software keinen Schaden anrichten kann.

Silverlight: Von Microsoft entwickelte Alternative zu Adobe Flash für die interaktive Gestaltung von Webseiten.

Tagging: Verschlagworten von Objekten (zum Beispiel Bildern, Videos, Webseiten), wobei dem Objekt frei wählbare Schlüsselwörter (Tags) zugeordnet werden, um dieses zu kategorisieren.

Template: Designvorlage für Webseiten, welche schnell ausgetauscht oder angepasst werden kann. So ist es möglich, innerhalb kurzer Zeit, das komplette Design einer Webseite zu ändern.

Thin Client: Computer, dessen Funktionalität sich nur auf die Ein- und Ausgabe beschränkt.

W3C: Abkürzung des World Wide Web Konsortiums, welches für die Standardisierung und Entwicklung des Internets zuständig ist. URL: <http://www.w3c.org>

Webcrawler: Auch Spider oder Robot genannt, wird von Suchmaschinen eingesetzt, um Webseiten zu indizieren.

Webmail: Online E-Mail-Programm zur Verwaltung der E-Mails über einen Webbrowser

Wikipedia: Ist eine frei zugängliche Online-Enzyklopädie, welche von hunderttausenden Autoren gepflegt wird. Jeder Benutzer kann Artikel erstellen und bearbeiten. URL: <http://de.wikipedia.org>

Wikis: Sind Nachschlagewerke, welche von den Benutzern über das Internet bzw. Intranet bearbeitet werden können.

XHTML: Von der W3C als Nachfolger von HTML empfohlen. Basiert auf XML und soll eine strengere Strukturierung der Webseite schaffen.

XSLT: Ist eine Programmiersprache zur Konvertierung von XML Daten in andere Dateiformate.

YouTube: Kostenloses, riesiges Videoportal, wo Benutzer ihre eigenen Videoclips uploaden können. URL: <http://www.youtube.com>

Literaturverzeichnis

ADOBE, 2007. Flash content reaches over 98% of Internet viewers [online]. Verfügbar bei http://www.adobe.com/products/player_census/flashplayer/ [Zugang am 14. Januar 2008]

ALBY, T., 2007. Web 2.0 – Konzepte, Anwendungen, Technologien. München: Carl Hanser Verlag.

AMAZON, 2008. Amazon.com Loose Diamonds [online]. Verfügbar bei <http://www.amazon.com/Loose-Diamonds-Diamond-Engagement-Rings/loosediamonds> [Zugang am 28. Januar 2008]

AMRUTH, D., 2006. AJAX Grundlagen und Verwendung in .NET – Stichwort „ATLAS“. München: GRIN Verlag.

ANDERSON, C., 2006. The Long Tail, in a nutshell [online]. Verfügbar bei <http://www.longtail.com/about.html> [Zugang am 20. Januar 2008]

BOßMANN, C., 2003. Urheberrechtsverletzungen im Online-Bereich und strafrechtliche Verantwortlichkeit der Internet-Provider. Frankfurt am Main: Peter Lang GmbH.

BRIEGLEB, V., 2007. Universal Music verklagt weiteres Videoportal [online]. Verfügbar bei <http://www.heise.de/newsticker/meldung/95617> [Zugang am 28. Januar 2008]

CHISHOLM, W., 2000. HTML Techniques for Web Content Accessibility Guidelines 1.0 nutshell [online]. Verfügbar bei <http://www.w3.org/TR/WCAG10-HTML-TECHS/#scripts> [Zugang am 22. Januar 2008]

DJAHANGIRI, N. & EDELHOFER, E., 2007. IKT-EINSATZ – Einsatz von Informations- und Kommunikationstechnologien in Unternehmen und in Haushalten 2007. Wien: Statistik Austria.

ERESULT, 2007a. Web 2.0: Nutzung, Nichtnutzung und Erfolgsfaktoren [online]. Verfügbar bei http://www.eresult.de/Texte/Web_20_Nutzung_Nichtnutzung_und_Erfolgsfaktoren.pdf [Zugang am 18. Januar 2008]

ERESULT, 2007b. Wording-Studie 3.0 – Web 2.0- und E-Commerce-Begriffe, Funktionen und Elemente [online]. Verfügbar bei http://www.eresult.de/Texte/eResult_Wordingstudie_Kurzversion.pdf [Zugang am 18. Januar 2008]

FARBER, D., 2006. TechNet Summit: The new era of innovation [online]. Verfügbar bei <http://blogs.zdnet.com/BTL/?p=3959> [Zugang am 6. Februar 2008]

FRIEDMANN, V., 2007. Praxisbuch Web 2.0. Bonn: Galileo Press.

GAMPERL, J., 2007. AJAX – Grundlagen, Frameworks, APIs. 2. Auflage. Bonn: Galileo Press.

GARRETT, J. J., 2005. Ajax: A New Approach to Web Applications [online]. Verfügbar bei <http://www.adaptivepath.com/ideas/essays/archives/000385.php> [Zugang am 30. Dezember 2007]

GARTNER, 2006. Emerging Technologies Hype Cycle Highlights Key Technology [online].

- Verfügbar bei <http://www.gartner.com/it/page.jsp?id=495475> [Zugang am 22. Januar 2008]
- GEHTLAND, J. & GALBRAITH, B., 2006.** Ajax – Eine pragmatische Einführung in Web 2.0. München: Carl Hanser Verlag.
- GOOGLE, 2007.** Cloaking, sneaky Javascript redirects, and doorway pages [online]. Verfügbar bei <http://www.google.com/support/webmasters/bin/answer.py?answer=66355> [Zugang am 29. Januar 2008]
- GOOGLE, 2008.** Google Suggest [online]. Verfügbar bei <http://www.google.de/webhp?complete=1&hl=en> [Zugang am 5. Januar 2008]
- GÖHRING, M. & HAPP, S. & MÜLLER, T., 2006.** Web 2.0 im Kundenmanagement, HMD – Praxis der Wirtschaftsinformatik, Heft 252, 55-65
- GRATZER, M.A., 2006.** Web 2.0. Diplomarbeit. Technische Universität Wien
- HAUSER, T., 2007.** Rettungsanker – Bookmarks und der Zurück-Button in Ajax-Anwendungen [online]. Verfügbar bei http://www.linux-magazin.de/heft_abo/ausgaben/2007/02/rettungsanker [Zugang am 6. Januar 2008]
- HEIN, A., 2007.** Web 2.0 – Das müssen Sie wissen. Planegg: Rudolf Haufe Verlag GmbH & Co. KG.
- HENRY, S.L., 2005.** Introduction to Web Accessibility [online]. Verfügbar bei <http://www.w3.org/WAI/intro/accessibility.php> [Zugang am 21. Januar 2008]
- HOLZ, P., 2006.** Was ist Web 2.0? [online]. Verfügbar bei http://www.distinguish.de/?page_id=45 [Zugang am 22. Dezember 2007]
- HOLZNER, S., 2006.** Ajax für Dummies. Weinheim: WILEY-VCH Verlag GmbH & Co. KgaA.
- IBM, 2006.** DeveloperWorks Interviews: Tim Berners-Lee [online]. Verfügbar bei <http://www-128.ibm.com/developerworks/podcast/dwi/cm-int082206.txt> [Zugang am 22. Dezember 2007]
- JAKUBETZ, C., 2006.** Citizen Media, Transit2 – Medien und Designsymposium [Vortrag]. FH Augsburg. 29.06.2006
- KIENITZ, G., 2007.** Web 2.0 – Der ultimative Guide für die neue Generation Internet. Kempen: moses. Verlag GmbH.
- KNAPPMANN, L. & LACHMANN, J., 2008.** Brockhaus-Wissen war Macht [online]. Verfügbar bei http://www.financial-times.de/technik/medien_internet/Agenda%20Brockhaus%20Wissen%20Macht/316389.html [Zugang am 14. Februar 2008]
- KOLLMANN, T. & HÄNSEL, M., 2007.** Web 2.0 – Trends und Techniken im Kontext der Net Economy. Wiesbaden: Deutscher Universitäts-Verlag.
- KRAUSE, J., 2003.** PHP 4 – Grundlagen und Profiwissen. München: Carl Hanser Verlag.
- LE HÉGARET, P., 2005.** Document Object Model (DOM) [online]. Verfügbar bei <http://www.w3.org/DOM/> [Zugang am 20. Januar 2008]

- MAYER, M., 2008.** Referenz-Webseiten [online]. Verfügbar bei <http://www.mm-webconsulting.com/referenzen.php> [Zugang am 20. Januar 2008]
- MCLAUGHLIN, B., 2006.** Ajax von Kopf bis Fuß. Köln: O'Reilly Verlags GmbH & Co. KG.
- MICROSOFT, 2008.** Internet Explorer Developer Toolbar [online]. Verfügbar bei <http://www.microsoft.com/downloads/details.aspx?FamilyID=e59c3964-672d-4511-bb3e-2d5e1db91038&DisplayLang=en> [Zugang am 20. Januar 2008]
- MINTERT S. & LEISEGANG C., 2007.** Ajax – Grundlagen, Frameworks und Praxislösungen. Heidelberg: dpunkt.verlag GmbH
- MUSSER, J. & O'REILLY, T., 2006.** Web 2.0 Principles and Best Practices. Köln: O'Reilly Verlags GmbH & Co. KG.
- NACK, J., 2007.** "Photoshop Express" RIA sneak-peeked today [online]. Verfügbar bei http://blogs.adobe.com/jnack/2007/09/photoshop_expre.html [Zugang am 14. Januar 2008]
- O'REILLY, T., 2005.** What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software [online]. Verfügbar bei <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> [Zugang am 20. Dezember 2007]
- O'REILLY, T., 2006.** Web 2.0 Compact Definition: Trying Again [online]. Verfügbar bei http://radar.oreilly.com/archives/2006/12/web_20_compact.html [Zugang am 20. Dezember 2007]
- PFEIFFER, H., 2006.** AJAX – Modernes Webscripting. Düsseldorf: Data Becker
- RESULT, 2007.** Web 2.0 – Begriffsdefinition und eine Analyse der Auswirkungen auf das allgemeine Mediennutzungsverhalten [online]. Verfügbar bei http://www.result.de/assets/public/doc/web-2.0-studie_result_sw_r_februar_2007.pdf [Zugang am 6. Februar 2008]
- SCHIEMANN, D., 2007.** Why Comet Daily? [online]. Verfügbar bei <http://cometdaily.com/2007/10/18/why-comet-daily/> [Zugang am 29. Januar 2008]
- STEYER, R., 2006.** AJAX mit PHP – Beschleunigte Webapplikationen für das Web 2. München: Addison-Wesley Verlag.
- STEYER, R. & FUCHS, J. 2006.** AJAX mit ASP.NET und Atlas. München: Addison-Wesley Verlag.
- ULBRICHT, C., 2007.** Rechtsprechung zur Haftung für „User Generated Content“ bestätigt [online]. Verfügbar bei <http://www.rechtzweinnull.de/index.php?/archives/25-Rechtsprechung-zur-Haftung-fuer-User-Generated-Content-bestaetigt.html> [Zugang am 6. Februar 2008]
- UPC, 2008.** Chello plus [online]. Verfügbar bei <http://www.upc.at/chello/plus/> [Zugang am 6. Februar 2008]
- VANDER, T., 2005.** Folksonomy Definition and Wikipedia [online]. Verfügbar bei <http://www.vanderwal.net/random/entrysel.php?blog=1750> [Zugang am 18. Januar 2008]
- W3C. 1999.** Scripts in HTML documents [online]. Verfügbar bei <http://www.w3.org/TR/html401/interact/scripts.html#h-18.3.1> [Zugang am 20. Januar 2008]

WALTER, T., 2008. Kompendium der Web-Programmierung – Dynamische Web-Sites. Berlin: Springer-Verlag.

WEBHITS, 2008. Web-Barometer [online]. Verfügbar bei <http://www.webhits.de/deutsch/index.shtml?webstats.html> [Zugang am 10. Januar 2008]

WELT ONLINE, 2008. Brockhaus: „Wir werden der Wissensnavigator im Netz sein“ [online]. Verfügbar bei http://www.welt.de/welt_print/article1666922/Brockhaus_Wir_werden_der_Wissensnavigator_im_Netz_sein.html [Zugang am 14. Februar 2008]

WENZ, C., 2006. JavaScript und AJAX. Bonn: Galileo Press.

WILKENS, A., 2007. Viacom verlangt 1 Milliarde US-Dollar Schadensersatz von Google [online]. Verfügbar bei <http://www.heise.de/newsticker/meldung/86655> [Zugang am 28. Januar 2008]

WIKIPEDIA, 2008. Falco - Unterschied zwischen Versionen [online]. Verfügbar bei <http://de.wikipedia.org/w/index.php?title=Falco&diff=42045279&oldid=42045267> [Zugang am 16. Februar 2008]

ZACHTE, E., 2007. Wikipedia Statistik – Neue Artikel pro Tag [online]. Verfügbar bei <http://stats.wikimedia.org/DE/TablesArticlesNewPerDay.htm> [Zugang am 16. Januar 2008]

ZANGER, G., 2000. MP3 macht Musik frei? Urheberrechtsverletzungen im Internet. In: B. FELDNER & N. FORGÓ, Hrsg. Chaos Control. Das Internet als dunkle Seite des Rechts? Wien: Manz, 51-55.

ZETTEL, C., 2007. Wikipedia lässt Brockhaus alt aussehen - Online-Enzyklopädie in punkto Aktualität nicht zu schlagen [online]. Verfügbar bei <http://www.pte.at/pte.mc?pte=071205025> [Zugang am 17. Januar 2008]

ZURAWSKI, K., 2007. Bieten an der Wissensbörse. bild der wissenschaft, 11/2007, 26-27

Anhang

```
01. <html>
02. <head>
03. <title>AJAX Testbeispiel</title>
04. <script language="javascript" type="text/javascript">
05.
06. var xhrObjekt = null;
07. try {
08.   xhrObjekt = new XMLHttpRequest();
09. }
10. catch(error){
11.   try {
12.     xhrObjekt = new ActiveXObject("Msxml2.XMLHTTP");
13.   }
14.   catch(error){
15.     try {
16.       xhrObjekt = new ActiveXObject("Microsoft.XMLHTTP");
17.     }
18.     catch(error) {
19.       alert("Fehler beim Erzeugen des XMLHttpRequest-Objekts");
20.     }
21.   }
22. }
23.
24. function getZitat()
25. {
26.   var url = "zitat.php";
27.   url = url + "?dummy=" + new Date().getTime();
28.   xhrObjekt.open("GET", url, true);
29.   xhrObjekt.onreadystatechange = updateSeite;
30.   xhrObjekt.send(null);
31. }
32.
33. function updateSeite()
34. {
35.   if (xhrObjekt.readyState == 4 && xhrObjekt.status == 200)
36.   {
37.     document.getElementById("zitat").innerHTML =
38.       xhrObjekt.responseText;
39.   }
40. }
41. </script>
42. </head>
43. <body>
44. <button onClick="getZitat();">Neues Zufallszitat
45. anzeigen</button>
46. <br />
47. <span id="zitat"></span>
48. </body>
49. </html>
```

Listing 15: Quellcode der HTML-Datei aus dem AJAX-Beispiel (Kapitel 3.6)

```
01. <?
02. srand(time());
03. $zufallszahl = rand(1,7);
04. switch($zufallszahl)
05. {
06.     case 1: echo "Zitat Nr. 1"; break;
07.     case 2: echo "Zitat Nr. 2"; break;
08.     case 3: echo "Zitat Nr. 3"; break;
09.     case 4: echo "Zitat Nr. 4"; break;
10.     case 5: echo "Zitat Nr. 5"; break;
11. }
12. ?>
```

Listing 16: Quellcode des PHP-Skripts aus dem AJAX-Beispiel (Kapitel 3.6)

```
01. <?
02. // Datenbankinformationen:
03. $hostname = "localhost";
04. $username = "web1";
05. $password = "my4pass";
06. $database = "zitateDB";
07.
08. // Verbindung zur Datenbank herstellen:
09. mysql_connect($hostname, $username, $password) OR
    die(mysql_error());
10. mysql_select_db($database) OR die(mysql_error());
11.
12. $ergebnis = mysql_query("SELECT * FROM zitate_autor AS autor,
    zitate AS zitate WHERE zitate.autor_id = autor.autor_id ORDER BY
    RAND() LIMIT 1");
13.
14. // Ausgabe der Datensätze:
15. while($row = mysql_fetch_object($ergebnis))
16. {
17.     echo $row->zitat . "-" . $row->autor;
18. }
19. ?>
```

Listing 17: Quellcode des PHP-Skripts aus dem AJAX-Beispiel (Kapitel 3.6) mit einer Datenbankabfrage

Studienauswertungen

Auswertung der Studie „Web 2.0- und E-Commerce-Begriffe, Funktionen und Elemente“ durchgeführt von der Firma eResult GmbH im Dezember 2007 (eResult, 2007b).

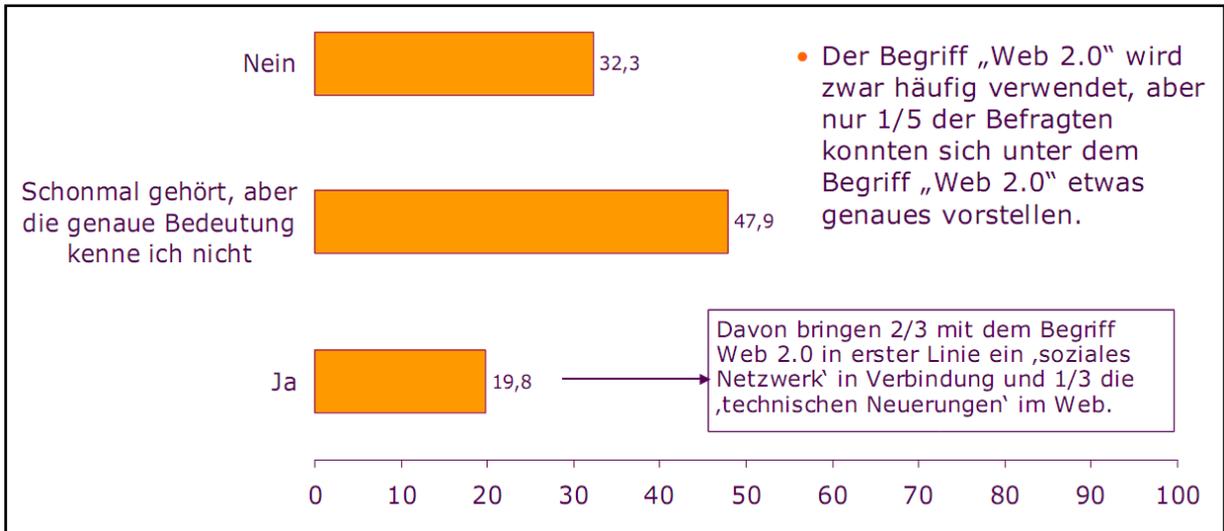


Abb. 27: Umfrage Diagramm: Können Sie sich unter dem Begriff Web 2.0 etwas vorstellen?

Auswertung der Studie „Nutzung von Web 2.0 Anwendungen“ durchgeführt von der Firma eResult GmbH im Auftrag der Boogie Medien GmbH vom Februar bis März 2007 (eResult, 2007a).

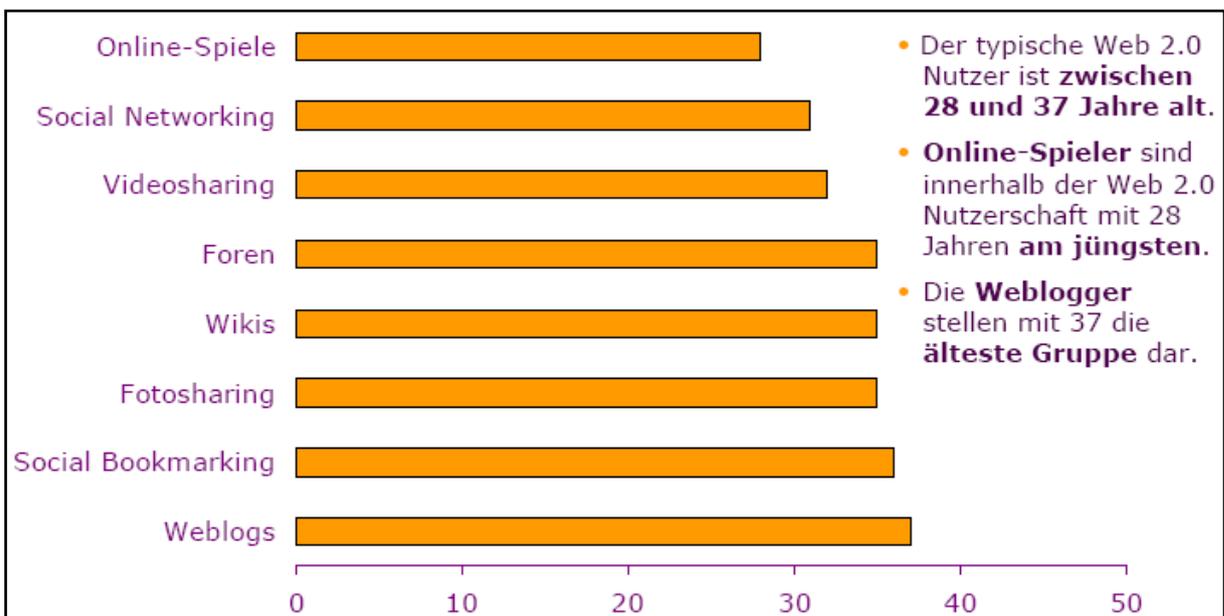


Abb. 28: Umfrage Diagramm: Durchschnittsalter der Nutzer von Web 2.0-Anwendungen

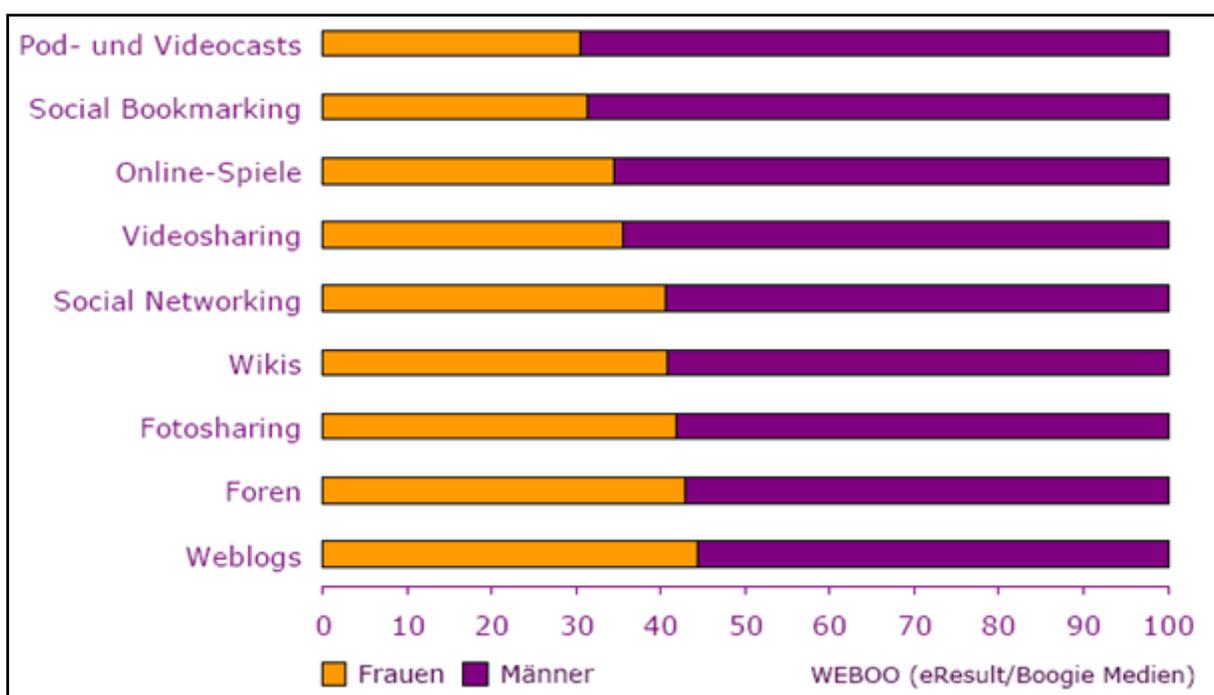


Abb. 29: Umfrage Diagramm: Frauen und Männer als Nutzer von Web 2.0-Anwendungen

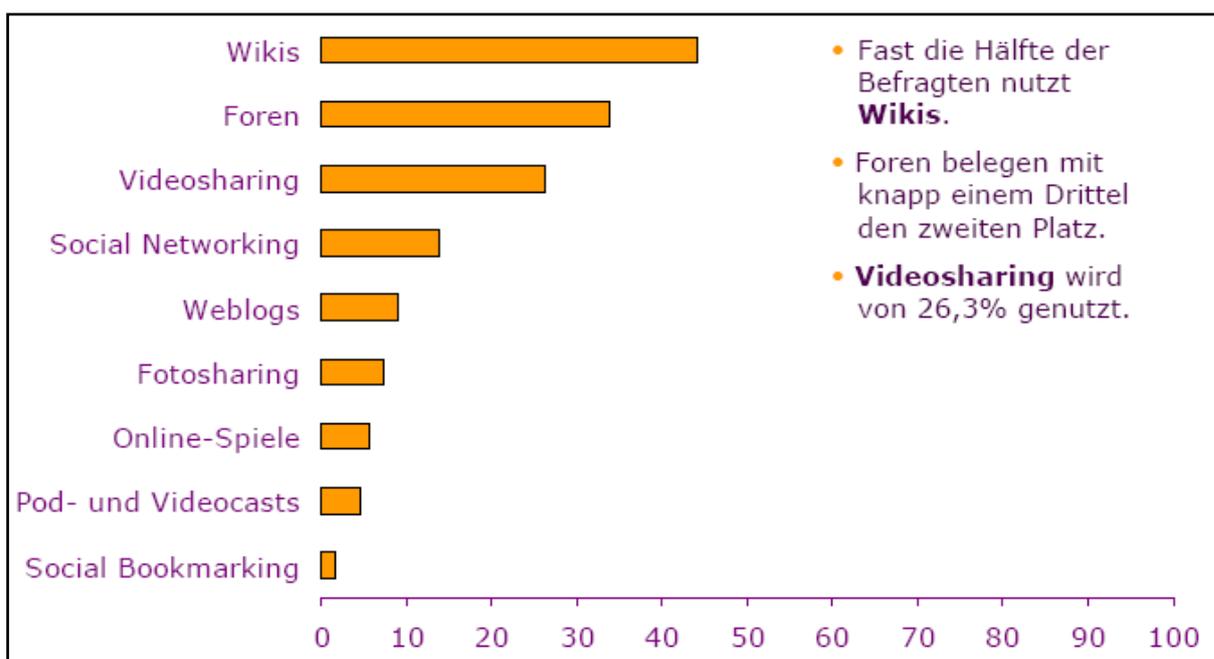


Abb. 30: Umfrage Diagramm: Nutzung von Web 2.0-Anwendungen

Auswertung der Studie „Web 2.0 - Begriffsdefinition und eine Analyse der Auswirkungen auf das allgemeine Mediennutzungsverhalten“ durchgeführt von der Firma result gmbh im Januar 2007 (Result, 2007).

Nutzungsmotiv: Information			
„Jedes Medium kann ja unterschiedliche Bedürfnisse befriedigen. Bitte geben Sie an, welches Medium Sie am ehesten nutzen würden, wenn Sie Folgendes suchen: WENN ICH MICH ÜBER AKTUELLE EREIGNISSE INFORMIEREN MÖCHTE“			
„am ehesten genutzt“	Gesamt	Aktiv partizipierende Nutzer	Passiv partizipierende Nutzer
Basis (abs.)*	501	287	214
Fernsehen	28,1%	24,0%	33,6%
Radio	6,4%	7,3%	5,1%
Internet	43,5%	46,7%	39,3%
Tageszeitung	20,0%	19,5%	20,6%
Zeitschriften	1,2%	1,4%	0,9%
DVD/VHS	0,4%	0,7%	0,0%
Weiß nicht	0,4%	0,3%	0,5%

Grafik: result *Basis: 501 Nutzer von „Web 2.0“-Angeboten

Abb. 31: Umfrage: Mediennutzungsverhalten