

Semesterarbeit Web 2.0 und Ajax

Autoren: *Gregor Wagner*(s744174; *Ajax*), *Florian Oeser*(s744400; *Web 2.0*)

Kontakt: florian.oeser@web.de, gregor.wagner@gmail.com

Letzte Änderung: **01.07.08**

Inhaltsverzeichnis

- 1 Web 2.0.....3**
- 1.1 Begriffsdefinition: Web 2.0.....4
- 1.2 Erörterungen und Hintergründe.....5
- 1.3 Wie ist der Begriff „Web 2.0“ entstanden?.....7
- 1.4 Techniken/Features hinter Web 2.0 und deren Anforderungen.....8
 - 1.4.1 Weblogs.....8
 - 1.4.1.1 Bestandteile eines Weblogs.....8
 - 1.4.1.1.1 Einträge.....9
 - 1.4.1.1.2 Kommentare.....9
 - 1.4.1.1.3 Trackback.....9
 - 1.4.1.1.4 Blogroll.....9
 - 1.4.2 RSS.....9
 - 1.4.3 Das Atom.....11
 - 1.4.4 Webservices.....12
- 2 Ajax.....14**
- 2.1 Was ist Ajax.....14
 - 2.2 Grundprinzip und Techniken.....16
 - 2.3 XMLHttpRequest.....17
 - 2.4 Vorteile von Ajax.....17
 - 2.5 Nachteile von Ajax.....17
 - 2.5.1 Polling-Problem.....18
 - 2.5.2 Optische Trägheit.....18
 - 2.6 Bekannte Java-Script-Frameworks für Ajax.....18
 - 2.6.1 Prototype (prototype.js).....18
 - 2.6.2 script.aculo.us.....19
- 3 Beispiele für Web 2.0 und Ajax.....20**
- 4 Kritik an Web 2.0 und Zukunftsbetrachtung.....21**
- 4.1 Kritik.....21
- 4.2 Zukunft21
- 5 Glossar.....22**
- 6 Quellen.....25**

1.1 Begriffsdefinition: Web 2.0

Der Begriff „Web 2.0“ bezieht sich primär auf eine veränderte Nutzung und Wahrnehmung des Internets als auf die Verwendung neuer spezifischer Technologien oder Innovationen. Sehr wichtig dabei ist, dass der Benutzer in hohem Masse selber die Inhalte erstellt und bearbeitet. Dies gilt qualitativ als auch quantitativ. Somit werden Inhalte nicht mehr nur zentral von Medienunternehmen erstellt und verbreitet, sondern vielmehr von einzelnen Persönlichkeiten und Individuen. Ein Aspekt ist dabei auch die soziale Vernetzung untereinander. Typische Beispiele hierfür wären *Wikis*, *Blogs*, Videoportale wie „YouTube“ oder auch „MySpace“.

Somit gehen auch die Zeiten der lokalen Datenhaltung zu Ende. Wenn man will hat man heute schon per Handy vollen Zugriff auf das Web. Es ist nicht mehr wichtig wo man ist, man braucht lediglich einen Zugang zum Internet egal ob per Kabel oder Funk, die Daten und Inhalte sind immer und überall verfügbar.

Der eigentliche Begriff „Web 2.0“ wurde eher als Marketing-Schlagwort eingeführt und stieß auf relativ großen Widerhall in den Medien. Denn wie erwähnt bezeichnet „Web 2.0“ keine spezielle Technik, sondern das Zusammenwirken verschiedener Methoden und Werkzeuge sowie eine soziale und wirtschaftliche Entwicklung. Da somit technische Mittel oft nicht ins Gewicht fallen wird oft kritisiert das der Begriff zu unscharf sei und somit beliebig verwendet werden kann um zum Beispiel Trends zu prägen und zu vermarkten.

Rein technisch betrachtet bezeichnet „Web 2.0“ oft nur eine Kombination von Methoden die bereits Ende der 90'er Jahre entwickelt wurden, die aber erst durch die hohe Anzahl an breitbandigen Internetzugängen weltweit und allgemein verfügbar wurden.

1.2 Erörterungen und Hintergründe

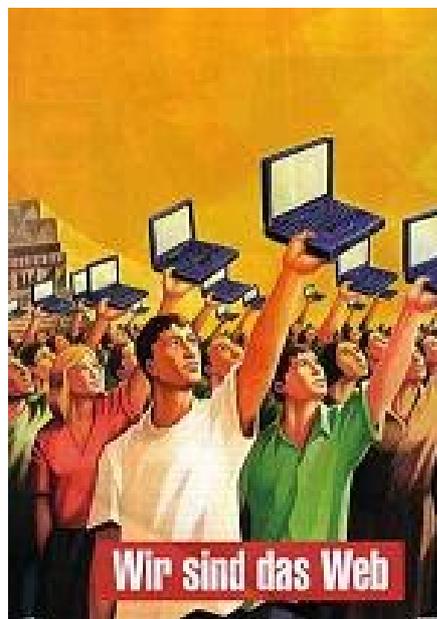
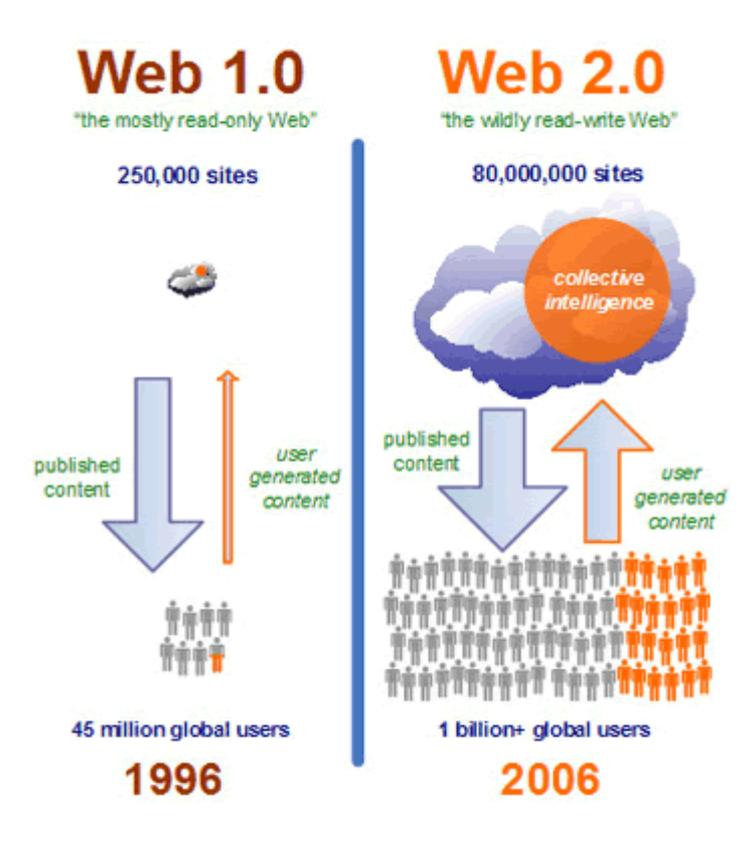
Das Internet und deren Nutzung war für die meisten Menschen ein nicht ganz einfach zu nutzendes technisches System. Es gab verhältnismäßig wenig Leute und Firmen die Inhalte für das Web erstellten bzw. Informationen bereitstellten. Im Gegensatz dazu gab es aber viele Benutzer die dieses Angebot nutzten.

Das Web bestand viele Jahre überwiegend aus statischen HTML-Seiten, die nur spärlich überarbeitet, aktualisiert oder ausgetauscht wurden. Um den immer größer werdenden Anfragen gerecht zu werden und diese zu verwalten, wurden *ContentManagementSysteme* und aus Datenbanken gespeiste Systeme entwickelt. Dies half die Inhalte dynamisch zur Laufzeit auszutauschen oder zu ersetzen.

Es gab viele Entwicklungen die mit der Einführung von „Web 2.0“ zur veränderten Nutzung des Internets beigetragen haben. So zum Beispiel braucht ein Anwender keine überdurchschnittlichen technischen Kenntnisse mehr um Datenspeicher, wie Tauschbörsen oder Services um Bilder hochzuladen, zu benutzen. Es kommt auch immer mehr zu einer Trennung zwischen lokalen und netzbasierten Anwendungen. So aktualisieren sich Programme autark über das Internet und immer mehr Anwendung benutzen einen Internet-Browser als Benutzerschnittstelle. Ein weiterer wichtiger Punkt bei diesen Entwicklungen ist das die Grenze zwischen Konsumenten und den Anbietern von Informationen immer weiter schmilzt. So wird und wurde es immer üblicher das Menschen, auch mit wenig EDV-Erfahrung, zum Beispiel eigene Weblogs pflegen oder überhaupt private Daten wie Fotoalben ins öffentliche Netz verlagern. Auch durch Neuerungen an der Programmierung von Web-Applikationen ist es Nutzern mit wenig Erfahrung in dem Bereich möglich an der Informations- und Meinungsverteilung teilzunehmen.

Damals, also vor der Einführung von „Web 2.0“, war es allgemein üblich das die Benutzer lediglich durch navigieren über die Webseite an ihre Informationen gelangt sind. Das lag vor allen daran das noch kein Nutzen daraus gezogen wurde das Benutzer die Seite individuell gestalten und anpassen könnten. Wie erwähnt waren die entsprechenden Techniken wie *PHP*, *XML* oder *RSS* schon entwickelt, wurden aber nicht in dem Umfang und für diese speziellen Zwecke eingesetzt. Dies änderte sich aber mit „Web 2.0“ bzw. resultierte „Web 2.0“ aus diesen Änderungen und man begann die Benutzer an der Entwicklung des WWW teilzuhaben und die Benutzung zu vereinfachen. Dadurch entstand ein neues Nutzerspektrum was nicht fast nur aus Spezialisten wie vorher bestand, sondern eine breite Masse die das Internet zu Berufs- und Freizeitgestaltung nutzten.

Zwei Grafiken die diese Entwicklung bzw. das Umdenken darstellen sind auf der nächsten Seite zu finden.



1.4 Techniken/Features hinter Web 2.0 und deren Anforderungen

Die Techniken die sich hinter dem Begriff „Web 2.0“ verbergen entwickeln sich wie fast überall schnell und ständig weiter und sind ebenfalls sehr vielfältig. Typische Anwendung sind Abonnementdienste wie *RRS* oder das *Atom*, *Weblogs*, Anwendungen für soziale Netzwerke oder allgemein Webservices. Eine ebenfalls typische Technik, die es ermöglicht, Webseiten wie bekannte Desktopanwendungen zu bedienen und aussehen zu lassen, ist *Ajax*.

Es gibt typische Anforderungen an Anwendungen, die dem Begriff „Web 2.0“ zugeordnet sind:

- ◆ Seiten werden dynamisch erzeugt und nicht statisch geladen
- ◆ Veränderungen je nach Benutzereingabe
- ◆ Benutzer haben die Möglichkeit selbst Inhalte zu erstellen
- ◆ fein ausgeklügelte Rechte- und Rollensysteme, um Benutzer zum Beispiel in Gruppen einzuteilen, kommen zum Einsatz
- ◆ persönliche Profile um sich mit Anderen zu identifizieren
- ◆ Benutzer haben die Möglichkeit die Seite nach ihren Wünschen anzupassen; das betrifft unter anderem die Auswahl von Themen, Farbe oder auch Hintergrundbildern.
- ◆ jeder Beitrag kann von anderen Benutzern kommentiert werden
- ◆ die Seiten sind intuitiv bedienbar um größtmöglichen Komfort zu bieten; das könnte beispielsweise über *Ajax*-Funktionalitäten umgesetzt werden

1.4.1 Weblogs

Ein *Weblog* oder auch *Blog* ist ein auf einer Webseite geführtes und damit öffentliches einsehbares Tagebuch bzw. Journal. Es handelt sich dabei um eine Webseite bei der die Einträge zu meist als eine lange, abwärts chronologisch sortierte Liste dargestellt wird. Ein *Blog* ist für den Verfasser (der „Blogger“) und dessen Leser ein einfach zu handhabendes Medium welches zum Darstellen von Aspekten des eigenen Lebens oder Meinungen zu spezifischen Themen dient. Der *Blog* dient somit dem Austausch von Informationen, Gedanken und Erfahrungen. Er ist insofern einem Internetforum ähnlich.

Ein *Weblog-Publishing-System* beruht prinzipiell auf einem *ContentManagementSystem* und ermöglicht dem Nutzer neue Inhalte zu erstellen oder zu verändern, auch wenn dieser über keine oder nur sehr geringe Kenntnisse im Bereich des Webdesign verfügt. Dies folgt dem Prinzip des „Web 2.0“. Den gestalterischen Aspekt dabei übernehmen dabei so genannte Templates oder Themes, welche sich sehr leicht ersetzen lassen. Die meisten Systeme dieser Art, wie „Wordpress“ oder „Serendipity“ basieren auf *MySQL* und *PHP* Lösungen. Die in den jeweiligen Programmen geschriebenen Einträge können zum größten Teil auch als *RSS-Newsfeed* lesbar gemacht werden.

1.4.1.1 Bestandteile eines Weblogs

Die folgenden zum Teile technischen Elemente eines *Weblogs* sind mehr oder weniger

typisch, beziehungsweise machen einen *Weblog* aus. Da es viele solcher Bestandteile gibt sind hier nur die Wichtigsten beschreiben.

1.4.1.1.1 Einträge

Ein Eintrag oder auch Post genannt ist ein Hauptbestandteil eines *Weblogs*. In diesem verfasst man die oben aufgeführten Sachen wie Informationen zu einem gewissen Thema. Die Einträge werden umgekehrt chronologisch sortiert angezeigt, das heißt die neusten Beiträge stehen ganz oben. Ältere Beiträge werden zum Teil auf weiteren Seiten oder in Archiven aufgelistet.

1.4.1.1.2 Kommentare

Bei vielen *Weblogs* ist es gemäß der „Web 2.0“-Definition möglich Kommentare zu einen Eintrag zu schreiben. Ein solcher Kommentar wird dann unter dem Eintrag auf der gleichen Seite angezeigt. Es ist jedoch möglich und auch wichtig für den Besitzer des Bloges diese Kommentare moderieren zu können. So kann er einstellen ob ein Kommentar sofort oder erst nach Freischaltung angezeigt wird. Dies wird häufig angewandt um Vandalismus und Spam in den *Blogs* zu verhindern, da es immer mehr automatisierte Programme (so genannte *Bots*) gibt, die Blogs mit viel Werbung etc. überhäufen.

1.4.1.1.3 Trackback

Der *Trackback* oder auch *Pingback* ist ein sehr nützliches Tool. Dies ermöglicht das wenn man in einem Eintrag auf seinem *Blog* einen Eintrag eines anderen „Bloggers“ erwähnt (über die URL), das bei diesem dann ähnlich wie ein Kommentar ein Hinweis darauf gesetzt wird. Somit kann man ein dichtes Informationsnetz erzeugen. Jedoch unterstützt nicht jede *Weblog-Software* dieses System.

1.4.1.1.4 Blogroll

Eine *Blogroll* ist eine für den Leser öffentliche Linksammlung zu verschiedenen Webseiten oder anderen interessanten *Blogs*. Diese Linkliste ist meist auf dem *Weblog* sichtbar platziert. Der Zweck dabei ist, Webseiten und *Blogs* die für ein persönlich interessant sind oder zum eigenen Blogthema passen, Anderen mitzuteilen um dort weitere Informationen zu beziehen.

1.4.2 RSS

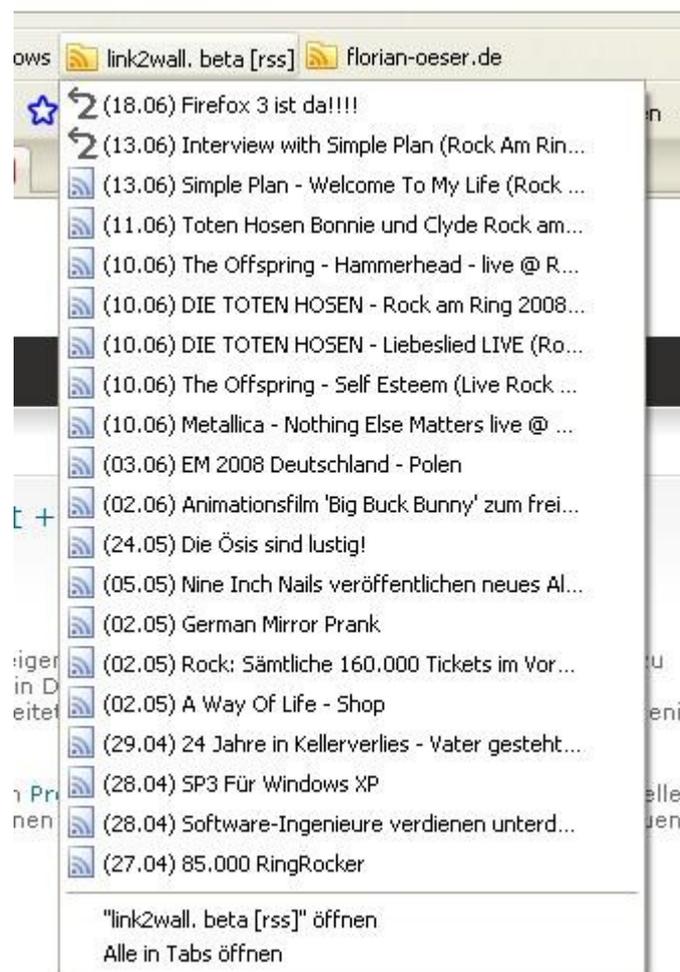


Die *Really Simple Syndication* zu deutsch etwa „wirklich einfache Verbreitung“, ist ebenfalls ein wichtiger Bestandteil von „Web 2.0“. Dieses System wird zum Teil auch *RSS-Feed* bezeichnet und wurde nach der Entstehung der *Blogs* zu einem beliebten Weg, Informationen schnell und leicht zu verbreiten. Diese *Feeds* verbreiten kontinuierlich ihre

Informationen, so dass Neuerungen, wie zum Beispiel ein neuer Eintrag auf einem *Blog*, vom Empfänger in regelmäßigen Abständen erkannt und angezeigt werden. Dies führt dazu, dass der Abonnent immer auf den neusten Stand gebracht wird, ohne dass der Verbreiter der Informationen dafür Sorge tragen muss. Es ist möglich, diese *RSS-Feeds* in eine Webseite einzubinden und diese individuell zu gestalten. Das passiert bei *Weblog-Software* beispielsweise automatisch.

Technisch gesehen ist *RSS* eine Familie von *XML*-basierten Formaten, welche optimiert und genormt Daten für verschiedene Endgeräte darstellt. Die Daten, welche man in einem standardisierten Format speichert, können dann mit *RSS-Parsern* äußerst einfach auf eine Webseite eingebunden werden.

Ein Beispiel, wie *RSS* für einen *Blog* in „Mozilla Firefox“ integriert wird:



Sobald man auf das Lesezeichen- bzw. den Button klickt, wird die *RSS-URL* automatisch aufgerufen und es werden alle neuen Einträge eingefügt. Interessant ist auch die Möglichkeit, bereits angeschauten Einträgen mit einem Pfeil zu markieren, wie es bei den ersten beiden Einträgen dargestellt ist.

Ein Beispiel für eine *RSS 2.0* Datei von „Wikipedia“ folgt auf der nächsten Seite:

```
<?xml version="1.0" encoding="utf-8"?>

<rss version="2.0">

  <channel>
    <title>Titel des Feeds</title>
    <link>URL der Webpräsenz</link>
    <description>Kurze Beschreibung des Feeds</description>
    <language>Sprache des Feeds (z. B. "de-de")</language>
    <copyright>Autor des Feeds</copyright>
    <pubDate>Erstellungsdatum</pubDate>
    <image>
      <url>URL einer einzubindenden Grafik</url>
      <title>Bildtitel</title>
      <link>URL, mit der das Bild verknüpft ist</link>
    </image>

    <item>
      <title>Titel des Eintrags</title>
      <description>Kurze Zusammenfassung des Eintrags</description>
      <link>Link zum vollständigen Eintrag</link>
      <author>Autor des Artikels, E-Mail-Adresse</author>
      <guid>Eindeutige Identifikation des Eintrages</guid>
    </item>

    <item>
      ...
    </item>

  </channel>

</rss>
```

1.4.3 Das Atom



Das *Atom Syndication Format* oder kurz *Atom* wird ähnlich wie *RSS* zum dynamischen Austausch von Informationen über die *Feeds* genutzt. Das *Atom* wurde entwickelt um die Vorteile der verschiedenen *RSS-Formate* zu vereinen und damit abzulösen und steht somit in direkter Konkurrenz zu *RSS*. Das *Atom* ist seit 2005 ein öffentlicher Internetstandard. Auch dieses Verfahren beruht auf *XML-Standards* und wird auch also solche Datei (*.xml)

veröffentlicht.

Der größte Unterschied des Atoms im Vergleich zu RSS ist die Möglichkeit, dass bei inhaltstragenden Elementen ausdrücklich angegeben werden kann, um was für Inhalte es sich handelt. So kann man das Atom um ein „type-Attribut“ ergänzen, indem dann neben normalen Text und maskiertem HTML auch direkt *XHTML-Markup* mit Namensraum-Angaben eingebettet werden können.

Ein Beispiel für eine Atom Datei von „Wikipedia“:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <author>
    <name>Autor des Weblogs</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Titel des Weblog-Eintrags</title>
    <link href="http://example.org/2003/12/13/atom-beispiel"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Zusammenfassung des Weblog-Eintrags</summary>
    <content>Volltext des Weblog-Eintrags</content>
  </entry>
</feed>
```

Ein Beispiel für das „type-Attribut“ (HTML):

```
<summary type="html">Beispielzusammenfassung mit
  &lt;strong&gt;wichtigem Text&lt;/strong&gt;</summary>
```

Das „type-Attribut“ für Text also „text“ kann bei Gebrauch auch weggelassen werden, da dies das Standardattribut von *type* ist.

Ein Beispiel für das „type-Attribut“ (XHTML):

```
<content type="xhtml" xml:base="http://example.org/">
  <div xmlns="http://www.w3.org/1999/xhtml">
    <p>Beispielabsatz mit <strong>wichtigem Text</strong> und einem
    <a href="beispiel">relativen Hyperlink</a>.</p>
  </div>
</content>
```

1.4.4 Webservices

Ein *Webservice* bzw. Webdienst ist eine Software-Anwendung deren Schnittstellen als *XML*-Artefakte definiert, beschrieben und gefunden werden können. Dieser *Webservice* unterstützt die direkte Interaktion mit anderer Software unter Verwendung *XML*-basierter Nachrichten durch den Austausch über internetbasierte Protokolle.

Webservices sind Softwaresysteme die automatisiert Daten austauschen und/oder Funktionen auf entfernten Rechnern aufrufen und sind stückweise vergleichbar mit dem was Webseiten für den Menschen sind. *Webservices* vereinen daher verteilte und

objektorientierte Programmierstandards und orientieren sich somit an der *Serviceorientierten Architektur (SOA)*. Erreichbar sind *Webservices* über eine eindeutige URL.

Google betreibt seit 2002 einen *Webservice* der die gleichen Möglichkeiten bietet wie die Webseite selbst. So können andere Programme nun mit dem Ansprechen der Schnittstelle direkt nach Informationen im Internet suchen und erhalten ebenfalls über diese Schnittstelle die Ergebnisdaten und können diese nun auswerten.

Ein anderes abstrakteres Beispiel ist die Interaktion zwischen Fluggesellschaft und Reisebüros. Die Fluggesellschaften stellen die Möglichkeit zum Nachschlagen bzw. Buchen von Flügen über einen *Webservice* bereit. Die Reisebüros bieten auf ihrer Webpräsenz Flüge verschiedener Fluggesellschaften an, von denen die Reisebüros zur Laufzeit erfahren. Der Kunde kann auf der Webseite des Reisebüros nun zentral Preise und Termine verschiedener Flüge vergleichen und direkt buchen.

2 Ajax

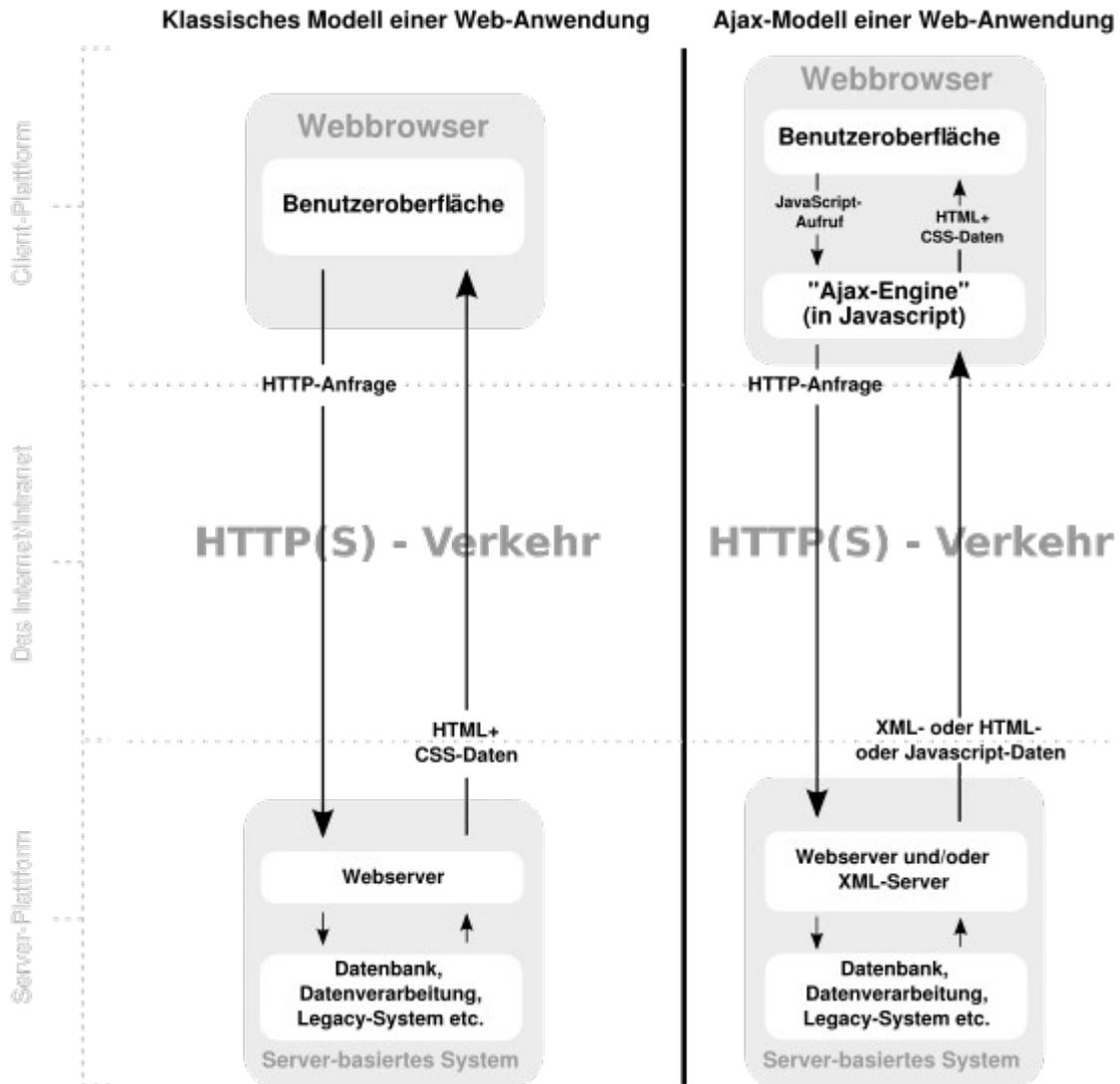
2.1 Was ist Ajax

Ajax bedeutet ausgesprochen *Asynchronuus JavaScript and XML*. Ähnlich ist wie Web 2.0 ist es auch nur eine neue Bezeichnung für bereits vorhandene Dinge. Erstmals geprägt wurde der Begriff *Ajax* von Jesse James Garret in dem Artikel: „Ajax: A new Approach to Web Applications“ am 15. Februar 2005. *Ajax* beschreibt eine Methode zur asynchronen Kommunikation von Client-seitigen Webanwendungen mit Webservern, welche schon vorher als *XMLHttpRequest* bekannt war.

Bis zu diesem Zeitpunkt interagierten Webanwendungen immer synchron mit einem Webserver, das heißt der Benutzer hatte nur die Möglichkeit eine Aktion auszuführen. Das Betätigen eines Links. Dadurch erfolge eine Verarbeitung auf dem Server und es wurde die ganze Seite mit neuem Inhalt dargestellt. Die Möglichkeit asynchrone Anfragen zu stellen war bis dato nur mit Hilfe von *InlineFrames* gegeben.

Im Gegensatz dazu sind *Ajax*-Anwendungen in der Lage Anfragen an den Server zu senden, bei denen nur die Daten angefordert werden, die tatsächlich benötigt werden. Dies geschieht durch den Aufruf eines Web-Services. Der Aufruf erfolgt als asynchrone Kommunikation, was bedeutet dass der Benutzer weiter mit der Oberfläche interagieren kann während die Daten geladen werden. Diese werden, sobald sie fertig geladen sind, durch eine vorher benannte Java-Script-Funktion in die Webseite eingebunden.

„Jede Benutzeraktion die für gewöhnlich eine HTTP-Anfrage erzeugen würde, erzeugt nun einen Java-Script-Aufruf, der an die Ajax-Engine delegiert wird [...]“, so beschreibt es Garret in seinem Essay. *„Jede Antwort auf eine Aktion den Benutzers, die keine Verbindung zum Server erfordert, wie beispielsweise das validieren von Daten, das verändern von Daten welche sich im Speicher befinden, und sogar das navigieren zwischen einzelnen Elementen der Webseite – dies kann alles von der Ajax-Engine bewältigt werden. Benötigt die Ajax-Engine Daten vom Server, um eine bestimmte Aktion erfolgreich durchführen zu können (beispielsweise das Übertragen von Daten, die verarbeitet werden müssen, um das Nachladen einzelner Bausteine der Benutzeroberfläche oder um das Laden neuer Daten) – führt diese eine asynchrone Anfrage, für gewöhnlich in form eines XML_Dokumentes an den Server durch. Dabei wird jedoch die Interaktion des Benutzers mit der Webeite wie dies bei gewöhnlichen Webanwendungen der Fall wäre, nicht unterbrochen [...]“* (Jesse James Garret).



2.2 Grundprinzip und Techniken

Um *Ajax*-Anwendungen zu verwirklichen benötigt man innerhalb des Webbrowsers und auf dem Webserver eine Komponente welche *Ajax*-Anwendungen ermöglicht.

Die Client-seitige Umsetzung (Webbrowser) erfolgt in den meisten Fällen auf Basis von *JavaScript* und dem *XMLHttpRequest-Object*. Für die Implementierung gibt es zwei Varianten:

Direkte *Ajax*-Implementierung:

Bei der direkten Implementierung werden so genannte Einstiegspunkte der Webseite definiert, worüber neue Daten nachgeladen werden können. Das heißt Teilfragmente der Seite können eigenständig ihren Inhalt nachladen. Dies schont die serverseitige Auslastung, erfordert aber relativ hohen Programmieraufwand.

Indirekte *Ajax*-Implementierung:

Hierbei werden neue HTML-Fragmente vom Server an den Client gesendet, um die vorhandene Seite zu ergänzen. Dazu wird in der Regel die anzuzeigende Seite auf dem Server komplett neu aufgebaut, aber nur die relevanten Unterschiede werden zum Client übertragen. Bei diesem Verfahren ist die Implementierung im Gegensatz zur direkten Implementierung sehr stark vereinfacht.

Die Umsetzung auf dem Server erfolgt mit Hilfe von *.Net* – bzw. einer *Script*-Komponente oder aber auch als *EJB's*. Die *Ajax*-Anwendung selbst braucht keine bestimmte Technik um auf dem Server umgesetzt zu werden. Sie muss nur das *XMLHttpRequest-Object* zur Verfügung stellen. Die restlichen Komponenten werden durch den Browser verwirklicht.

Ajax-Anwendungen an sich arbeiten mit schon länger bekannte Mitteln. Diese waren früher jedoch nur als *XMLHttpRequest* bekannt. Jesse James Garret hat durch seinen Artikel lediglich die zur Verfügung stehenden Mittel benannt und somit die Marke *Ajax* erschaffen.

Der Aufbau einer Anwendung mit *Ajax* sieht wie folgt aus:

Es wird ein Dokument als Träger, in den meisten Fällen HTML aber auch *XHTML* benötigt. Für die Client-Server-Interaktion wird *JavaScript* mit *XMLHttpRequest* verwendet. Die Daten werden dann über ein *XML*-standardisiertes Format validiert. Der eigentliche Hauptbestandteil ist der Kommunikationsfluss zwischen Client und Server. Die Seitengestaltung wird wie gewohnt in HTML geschrieben und Desktop ähnliche Interaktionen mit Hilfe von *JavaScript* gesteuert, welche allerdings auf die neuen *Frameworks* der *Ajax*-Schnittstelle zugreifen und damit wieder zur *Ajax*-Programmierung gehören.

2.3 XMLHttpRequest

Das *XMLHttpRequest-Object* ist eine *API* welche zum Senden verschiedener Daten über das HTTP Protokoll genutzt wird. Dazu werden verschiedene Abfragetypen genutzt. Die größte Rolle spielen dabei : GET, POST, HEAD und PUT. Anfragen welche mittels *XML*-Struktur an den Server gesendet werden, können über das *XMLHttpRequest* als *Plain-Text* oder als *DOM*-Struktur zurückgesendet werden. Meist wird hierbei die *DOM*-Struktur bevorzugt. Mittels des *XMLHttpRequest* kann eine Webseite dynamisch Daten nachladen ohne einen Seitenrefresh zu verursachen.

Beispiel für einen *XMLHttpRequest*:

```
var xmlHttpRequest = null;
if (typeof XMLHttpRequest != undefined) {
    xmlHttpRequest = new XMLHttpRequest ();
}
if (xmlHttpRequest) {
    xmlHttpRequest.open('GET', 'example.xml', true);
    xmlHttpRequest.onreadystatechange = function () {
        if (xmlHttpRequest.readyState == 4) {
            alert (xmlHttpRequest.responseText);
        }
    };
    xmlHttpRequest.send (null);
}
```

2.4 Vorteile von Ajax

Zweifelsohne ist der größte Vorteil von *Ajax* die Möglichkeit serverseitige Daten ohne das ein Browserrefresh durchgeführt werden muss. Damit wird das fortlaufende Arbeiten mit einer Webseite ohne Unterbrechungen gewährleistet, und es vermeidet ständigen Datenaustausch zwischen Client und Server.

Zudem wird kein Browserplugin benötigt, was bedeutet das *Ajax* auf allen *Java-Script*-unterstützenden Browsern lauffähig ist. Dies ist ein grosser Vorteil gegenüber ähnlich dynamischen Formaten wie „Adobe Shockwave“ oder „Flash“ welche zunächst ein Browserplugin installieren müssen um ihre Anwendungen lauffähig zu machen.

2.5 Nachteile von Ajax

Einer der häufigsten Kritikpunkte an *Ajax*-basierten Websites ist, dass die Funktion der Zurückschaltfläche die jeder Browser anbietet durch das dynamische Laden von Inhalten ausgehebelt wird. Die Meisten Browser speichern nur statische Seiten in ihrer Historie, und somit ist es oftmals nicht möglich eine Aktion über die Browserfunktion rückgängig zu machen. Dazu kommt, dass Lesezeichen eventuell den genauen Aufbau einer Seite nicht genau nachempfinden können, da in der Adresszeile der Anhang fehlt, welcher das Laden von Informationen auf der Webseite regelt.

2.5.1 Polling-Problem

Der Webserver ist nicht in der Lage asynchrone Events an den *Ajax*-Client zu liefern. Dies hat zur Folge, dass der Client permanent nachfrage muss ob eine serverseitige Veränderung auftrat, oder ein Event gefeuert wurde. Dies konfrontiert den Webserver mit einem Problem welches bei normalen Webanwendungen bisher nicht auftrat.

Die Lösung war die Entwicklung einer Technik, die Anfragen solange blockiert, bis tatsächlich ein Ereignis oder ein Timeout auftritt.

Diese Technik wirft jedoch eine neue Problematik auf. Bei herkömmlichen Webanwendungen wurde pro Anfrage an den Server ein Thread erstellt, dessen Ressourcen nach seiner Bearbeitung wieder freigegeben wurden. Allerdings wären diese Ressourcen bei der besagten *Polling-Technik* weiterhin belegt. Aus diesem Grund muss sich der Programmierer Gedanken über die Skalierbarkeit seiner *Ajax*-Anwendung machen.

2.5.2 Optische Trägheit

Es sollte darauf geachtet werden, den Benutzer über den Status der Ladezeit zu informieren, da beim Laden von Informationen ansonsten der Eindruck entstehen könnte, dass die getätigte Aktion keinerlei Auswirkungen hatte.

2.6 Bekannte Java-Script-Frameworks für Ajax

2.6.1 Prototype (prototype.js)

Prototype enthält vorgefertigte Klassen, die mit dem *XMLHttpRequest-Object* arbeiten, und so den Einstieg in *Ajax* erleichtern.

[Http://www.prototypejs.org/api/](http://www.prototypejs.org/api/)

Die Ajax-Klassen von Prototype:

Ajax.PeriodicalUpdater (container, url [, options])

Bewirkt einen periodischen *XMLHttpRequest* und aktualisiert den Inhalt des Containers basierend auf der *XML*-Antwort des Servers. Die Periodendauer kann als Parameter übergeben werden.

Ajax.Request (url [, options])

Initiiert einen *XMLHttpRequest* und führt diesen aus.

Ajax.Responders.register (responder)

Ajax.Responders.unregister (responder)

Dies ist ein Listener der *Ajax-Requests* erkennt. Er unterscheidet in erlaubte und unerlaubte Anfragen.

Ajax.Updater (container, url [, options])

Führt ein *Ajax-Request* aus und aktualisiert den Inhalt des Containers basierend auf der Antwort des Servers.

2.6.2 script.aculo.us

Script.aculo.us ist ein *Framework* für größtenteils grafische Effekte die auf *Ajax*-Basis gelöst werden. Als Basis wird das Prototype Framework benötigt.

Möglichkeiten mit script.aculo.us:

Effects

Eine Effektsammlung in der Ein- und Ausblendungen zusammengefasst sind

Draggable / Droppable

Hiermit ist es möglich bestimmte Container innerhalb eines Browserfensters per Drag'n'Drop zu verschieben.

Sortables / Slider

Bietet die Möglichkeit bestimmte Container zu vertauschen und zu verschieben, so dass der Benutzer sich Inhalte aus einer Tabelle selbst sortieren kann.

Autocompleting

Vervollständigt Texteingabefelder selbstständig nach bestimmten Suchkriterien.



In Place Editing

InPlace Editing bietet die Möglichkeit Kommentare direkt an bestimmte Bildpunkte zu schreiben um beispielsweise Bilder zu kommentieren.

Builder

Mit diesen Klassen steht einem die Möglichkeit zur Verfügung selbst *dynamische DOM-Objekte* hinzuzufügen.

3 Beispiele für Web 2.0 und Ajax

- ◆ <http://yedda.com/>
 - ◆ Community, Forum, persönliche Profile, Interessengruppen, Chat
- ◆ <http://www.ning.com/>
 - ◆ Plattform zum Erstellen eigenen Communitys
- ◆ <http://www.meebo.com/index-de.html>
 - ◆ Online-Plattform für alle gängigen InstantMessenger
- ◆ <http://www.vsocial.com/>
 - ◆ Videoupload-Portal mit Communityfunktionen
- ◆ <http://www.congoo.com/>
 - ◆ News und Informationen in Echtzeit; Suche auch über andere Portale
- ◆ <http://www.pando.com/>
 - ◆ kostenloser Dienst um Daten jeglicher Art zu tauschen; erweitert auch den Speicherplatz von E-Mailadressen
- ◆ <http://www.flickr.com/>
 - ◆ Dienst zum hochladen, sortieren, anschauen und drucken von Bildern im Internet
- ◆ <http://maps.google.de/>
 - ◆ Tool zum berechnen und betrachten von Routen; Ansicht über Satellitenbilder möglich
- ◆ <http://www.lastfm.de/>
 - ◆ großes deutsches Musiknetzwerk- bzw. Community
- ◆ <http://de.youtube.com/>
 - ◆ Videoportal mit Communityfunktionen

4 Kritik an Web 2.0 und Zukunftsbetrachtung

4.1 Kritik

„I think Web 2.0 is of course a piece of jargon, nobody even knows what it means „

Zu deutsch: „Ich finde, das Web 2.0 natürlich ein Jargonausdruck ist, niemand weiß wirklich was er bedeutet“. Dies sagte Tim Bremers-Lee, der Begründer des WWW und vertritt damit die Ansicht, dass das angeblich „neue Netzverständnis“ des Web 2.0 in Wahrheit nichts anderes als das ursprüngliche Netzverständnis ist, das bereits dem Web 1.0 zugrunde lag. Den Bremers-Lee konzipierte das Web von Anfang an im gleichen Maße zum Publizieren wie zum Konsumieren der Inhalte.

Viele Kritiker reden bei dem Begriff „Web 2.0“ von einer Marketingblase, welche vermeidet, Neuerungen genau zu beschreiben, indem viele Neuerungen ohne genaue Unterscheidung dem Web 2.0 zugeschlagen werden. Außerdem lege der Begriff „Web 2.0“ vereinfacht nahe, das Internet sei interaktiver geworden, obwohl es schon seit den Anfängen des Internets rege Usenet-Gemeinschaften gegeben habe. Weiterhin seien Techniken schon lange bevor der Begriff verwendet wurde, vorhanden gewesen. So kann man sagen das Dienste wie „Flickr“ oder *Blogs* mit diesen Mitteln entwickelt wurden und nicht aufgrund irgendwelcher theoretischen Überlegungen zu „Web 2.0“ entstanden sind.

4.2 Zukunft

Es wurde erst kürzlich behauptet das das Jahr 2008 das Ende für „Web 2.0“ sei. Berücksichtigt man, das der Begriff „Web 2.0“ ein Schlagwort für einen Trend im Internet darstellt, so darf man durchaus sagen, dass dieses Schlagwort die ersten Abnutzungserscheinungen zeigt. Einige Unternehmen werden mittlerweile auch die Erfahrung gemacht haben, dass der Erfolg im Internet nicht zwingend mit der Einbindung der Benutzer verbunden ist. Es könnte dieses Jahr also eine Rückbesinnung auf das Wesentliche erfolgen, aber sicher nicht das Ende des „Web 2.0“. Das heißt Einbindung des Benutzers da wo es Sinn macht, aber nicht um jeden Preis.

Neue Ideen unter dem Namen „Web 3.0, die dynamische Vernetzung“ sind bereits entstanden. Danach sollte es unter Anderem möglich sein, hoch frequentierte Blogs von ganzen Autorentams und Gastautoren zu erstellen und zu warten. Es gibt auch Ansätze die Inhalte von Zusatzinformationen zu entkoppeln. Einfach gesagt würde man das *Ipv6-Protokoll* benutzen um jeden Artikel, jedes Profil des jeweiligen Autors usw. mit einer eindeutigen IP zu verbinden. Erst bei der Datenbereitstellung werden dann die entsprechenden Inhalte zusammengeführt. Diese Überlegungen werden jetzt schon heftig im Internet diskutiert und vor allem kritisiert.

5 Glossar

Content Management System

Kurz CMS werden mittlerweile häufig im Bereich von Communities und grossen Firmenseiten an denen mehrere Redakteure arbeiten eingesetzt. CMS basiert auf Datenstrukturen hinterlegt in XML-Dateien oder Datenbanken wie MySQL. Damit wird das Einlesen von Informationen dynamisch und das Anpassen vom Layout der Webseite getrennt von den Inhalten. Das CMS dient als Denktopanwendung um dem Endbenutzer auf möglichst intuitive Weise das Erstellen von neuen Artikeln, Bildern, usw. zu erleichtern.

XML

Extensible Markup Language ist eine Auszeichnungssprache zur hierarchischen, strukturierten Darstellung in Textdateien.

MySQL

Dies ist ein relationales Datenbankverwaltungssystem zum Speichern von webbasierten Informationen, welche mit verschiedenen Programmiersprachen wie PHP, ASP etc. über die Datenbankschnittstelle SQL angesprochen werden können.

JavaScript

JavaScript ist eine Skriptsprache mit Elementen aus den funktionalen Programmiersprachen, die hauptsächlich für Dynamic HTML bei Web-Browsern eingesetzt wird. Es ist eine dynamisch typisierte, objektbasierte Sprache, ähnlich den objektorientierten, jedoch ohne Polymorphie und Vererbung.

API (Application Programming Interface)

Auf deutsch „Schnittstelle zur Anwendungsprogrammierung“. Dies definiert eine Programmierschnittstelle welche durch ein Softwaresystem zur Anbindung an andere Programme zur Verfügung gestellt wird.

DOM (Document Object Model)

Dies ist eine API zum Zugriff auf HTML oder XML-Dokumente. Die Standardisierung beruht auf dem World Wide Web Consortium (W3C) welches eine einheitliche Struktur für solche Dokumente beschloss.

W3C World Wide Web Consortium

Das Gremium des W3C sorgt für die Standardisierung der fürs Web bereitstehenden Techniken. Die Organisation wurde 1994 von Tim Berners-Lee gegründet, welcher auch für die Erfindung des World Wide Web bekannt ist.

.NET (dot net)

.NET ist eine Softwareplattform, die von Microsoft entwickelt wurde. Sie beinhaltet eine Laufzeitumgebung, eine API, sowie verschiedene angeschlossene Dienstprogramme (Services).

EJB

Enterprise Java Beans sind standardisierte Komponenten für Webserver, welche die Java Enterprise Edition unterstützen.

Tagclouds

zu deutsch: Schlagwortwolke; Eine Schlagwortwolke ist eine Methode zur Informationsvisualisierung, bei der eine Liste aus Schlagworten alphabetisch sortiert flächig angezeigt wird, wobei einzelne unterschiedlich gewichtete Wörter größer oder auf andere Weise hervorgehoben dargestellt werden.

Wikis

Ein Wiki, seltener auch WikiWiki oder WikiWeb genannt, ist eine Software und Sammlung von Webseiten, die von den Benutzern nicht nur gelesen, sondern meist auch direkt online geändert werden können. Wikis ermöglichen es verschiedenen Autoren, gemeinschaftlich an Texten zu arbeiten. Ziel eines Wiki ist es im Allgemeinen, die Erfahrung und den Wissensschatz der Autoren kollaborativ in Texten auszudrücken.

„What is Web 2.0“

dieser Artikel ist auf folgender Seite zu finden:

<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

Weblog-Publishing-System

Ein Weblog Publishing System (Weblog-Publikationssystem) ist im Grunde ein Web-Content-Management-System, dessen Schwerpunkt allerdings bei der Erstellung von Weblogs liegt. Charakteristische Leistungsmerkmale sind unter anderem das einfache Einpflegen von Beiträgen ohne HTML-Kenntnisse, die Ausgabe von RSS-Feeds sowie eine Kommentarfunktion, mit der an einer aktiven Diskussion zum Thema des Blog-Beitrages teilnehmen kann.

Serviceorientierten Architektur (SOA)

Serviceorientierte Architektur (SOA), engl. „service oriented architecture“, ist ein Ansatz der Informationstechnik, Dienste von Mitarbeitern und Organisationen zu strukturieren und zu nutzen. Diese Strukturierung und Nutzung kann auf unterschiedlichen Ebenen erfolgen. „Vergib einen Kredit“ ist beispielsweise auf einer hohen Ebene angesiedelt, dahinter versteckt sich bei einer größeren Bank ein Geschäftsprozess mit einigen beteiligten Personen und informationstechnischen Systemen (eröffnen der Geschäftsbeziehung, eröffnen

eines/mehrerer Konten, Kreditvertrag, ...), während „Trag den Kunden ins Kundenverzeichnis ein“ ein Dienst auf einer niederen Ebene ist.

Parser

Ein Parser (engl. to parse „analysieren“ bzw. von lateinisch pars „Teil“; im Deutschen gelegentlich auch Zerteiler) ist ein Computerprogramm, das in der Computertechnik für die Zerlegung und Umwandlung einer beliebigen Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist. Häufig werden Parser eingesetzt, um im Anschluss an den Analysevorgang die Semantik der Eingabe zu erschließen und daraufhin Aktionen durchzuführen.

XHTML-Markup

Der W3C-Standard Extensible HyperText Markup Language (erweiterbares HTML; Abkürzung: XHTML) ist eine textbasierte Auszeichnungssprache zur Darstellung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten. Im Gegensatz zu seinem Vorgänger HTML, welcher mittels SGML definiert wurde, verwendet XHTML die strengere und einfacher zu parsende SGML-Teilmenge XML als Sprachgrundlage. XHTML-Dokumente genügen also den Syntaxregeln von XML.

Ipv6-Protokoll

Das Internet Protokoll Version 6 (IPv6) ist der Nachfolger des gegenwärtig im Internet noch überwiegend verwendeten Internet Protokoll in der Version 4. Beide Protokolle sind Standards für die Vermittlungsschicht (Schicht 3) des OSI-Modells und regeln die Adressierung und das Routing von Datenpaketen durch ein Netz.

6 Quellen

Webseiten

- ◆ http://de.wikipedia.org/wiki/Web_2.0
- ◆ <http://www.drweb.de/weblog/weblog/?p=623>
- ◆ <http://de.wikipedia.org/wiki/RSS>
- ◆ [http://de.wikipedia.org/wiki/Atom_\(XML-Format\)](http://de.wikipedia.org/wiki/Atom_(XML-Format))
- ◆ <http://de.wikipedia.org/wiki/Weblog>
- ◆ <http://web-zweinull.de/index.php/was-ist-web-2/>
- ◆ <http://web-zweinull.de/index.php/>
- ◆ <http://de.wikipedia.org/wiki/Webservice>
- ◆ [http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung))
- ◆ <http://www.drweb.de/weblog/weblog/?p=454>

Magazine

- ◆ iX (09/2007 „Ajax-Server-Frameworks“)