



Dipl.-Wirtsch.-Inf. Stefan Breitenfeld

Barrierefreiheit und Webstandards - Kugelsicheres Webdesign der Zukunft

**Beitrag zum Kolloquium
"Die Zukunft der Software - die Software der Zukunft"**

Inhaltsverzeichnis

1	Einleitung.....	1
2	Webstandards.....	2
3	Barrierefreiheit.....	16
4	Ausblick auf zukünftige Entwicklungen.....	21
	Literaturverzeichnis.....	23

Verzeichnis der Abkürzungen und Akronyme

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ATOM	Atom Syndication Format
BITV	Barrierefreie Informationstechnik - Verordnung
CI	Corporate Identity
CSS	Cascading Stylesheets
DOM	Document Object Model
HTML	Hypertext Markup Language
JAXP	Java API For XML Processing
JSP	Java Server Pages
MSAA	Microsoft Active Accessibility - Interface
RDF	Resource Description Framework
RSS	Really Simple Syndication
SGML	Standard Generalized Markup Language
SPARQL	SPARQL Protocol And RDF Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
WWW	World Wide Web
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSL-FO	Extensible Stylesheet Language - Formatting Objects
XSLT	Extensible Stylesheet Language Transformations

Abbildungsverzeichnis

Abb. 2.1: Schematische Darstellung einer Transformation	4
Abb. 2.2: Schematische Darstellung des CSS-Boxmodells.....	8
Abb. 2.3: Die grundsätzliche Funktionsweise von AJAX	9
Abb. 2.4: CSS-formatierte Box mit abgerundeten Ecken.....	14
Abb. 2.5: Box mit manuell veränderter Schriftgröße	14

1 Einleitung

Kaum ein Bereich des öffentlichen Lebens hat sich in den vergangenen zehn Jahren so sehr gewandelt wie das Internet. Eine ständige Zunahme an vernetzten Haushalten und Organisationen, höhere Bandbreiten, in allen Belangen leistungsstärkere Rechner sowie neue Technologien und Standards haben dazu geführt, dass Web-Applikationen, Web-Services und Oberflächendesigns inzwischen wie selbstverständlich Nutzungspotenziale umsetzen und Möglichkeiten zum Datenaustausch eröffnen, die noch vor zehn bis zwölf Jahren nur kühne Visionäre prophezeit haben dürften. Eine aktuelle, diesen Sachverhalt illustrierende Strömung dürften wohl die so genannten Web 2.0 - Anwendungen darstellen.¹

Es mag überraschend klingen, aber einen wesentlichen Grundstein für diese Entwicklung legte sicherlich der Niedergang der Firma Netscape am Ende des 20. Jahrhunderts, bedeutete er doch das Ende des „Browserkrieges“.² Damit und mit dem zunehmenden Einfluss von globalen Standardisierungsgremien wie dem World Wide Web Consortium (W3C) wurde der Weg für dynamischere Webauftritte und die Durchsetzung von Webstandards geebnet. Auf Basis dieser Standards und eines neuen Bewusstseins für die Belange der Endnutzer wurde in den letzten Jahren der Begriff der *Accessibility* geprägt, um verschiedenen Nutzergruppen gleichermaßen uneingeschränkten Zugriff auf Webapplikationen zu ermöglichen. Mit diesen Konzepten der Webstandards und der Barrierefreiheit sollen sich die folgenden Kapitel kurz und prägnant auseinandersetzen.

¹ Der etwas unscharfe Begriff Web 2.0 ist eigentlich nichts weiter als ein zu Marketing-Zwecken eingeführtes Schlagwort, oder besser noch: ein Hype. Er steht stellvertretend für eine Generation von breitbandigen Anwendungen und Diensten, die verschiedenen Technologien (wie etwa Asynchronous JavaScript And XML, kurz AJAX, oder Really Simple Syndication, kurz RSS) integrieren, um dynamische Plattformen (z.B. Wikis, Weblogs, Foto-, Audio- und Videoportale) zu schaffen, welche zwar zentral von einem Provider zur Verfügung gestellt, aber dezentral von den Plattformnutzern gepflegt und mit Inhalten gefüllt werden. Ein Marketing-Ziel ist es sicher, Kunden dazu zu bringen, bestimmte Webseiten oder Portale von sich aus zu besuchen, indem man ihnen in einem abgesteckten Rahmen aktive Gestaltungsmöglichkeiten einräumt.

² Der „Browserkrieg“ fand in den 90er Jahren zwischen Netscape auf der einen und Microsoft auf der anderen Seite statt. Im Kampf um Marktanteile reicherten beide Hersteller ihre jeweiligen Browser immer mehr mit zusätzlichen, einzigartigen Features an, die das jeweilige Konkurrenzprodukt nicht bot. Diese Inkompatibilitäten sorgten dafür, dass es gleichsam eine Kunst wurde, Webseiten zu erstellen, die einerseits eine ansprechende Funktionalität und eine überzeugende Oberfläche boten, andererseits aber auf beiden Plattformen fehlerfrei verfügbar waren und (annähernd) identisch aussahen. Letztendlich konnte Netscape mit dem Budget von Microsoft nicht mehr Schritt halten und musste die Segel streichen.

2 Webstandards

Im modernen Webdesign wird immer häufiger unter Verwendung von Webstandards gearbeitet. Ein Grund dafür liegt sicher in den Restriktionen der Browser. Mit dem Ende des „Browserkrieges“ zwischen Netscape und Microsoft und dem steigenden Einfluss des W3-Konsortiums gestalten etablierte Browserhersteller wie Microsoft, Mozilla oder auch Opera ihre Produkte zunehmend standardkonform. Die Arbeit des Webdesigners wird damit durchaus vereinfacht, da er - so er sich beim Development an diese Standards hält - in immer höherem Maße darauf vertrauen kann, dass sein Endprodukt auf den verschiedenen Browsern tatsächlich ident dargestellt wird. Auf der anderen Seite verzeiht ein Webbrowser durch nachlässige Programmierung hervorgerufene Syntaxfehler weniger, als noch vor einigen Jahren, insbesondere bei Verwendung von gültigen XML-Dokumenten^{3,4}. Darüber hinaus bietet die Verwendung von Webstandards aber auch eine Reihe eindeutiger Vorteile, von denen die wesentlichen hier kurz aufgeführt werden sollen.

Webstandards

- gewährleisten unter Verwendung aktueller Browser ein identisches Layout auf verschiedenen Plattformen⁵,
- können bei der Trennung von Struktur und Inhalt in Webseiten unterstützen,
- vereinfachen die Wartung und Erweiterung bestehender Webseiten,
- können die Arbeit des Entwicklers erheblich erleichtern und Entwicklungszeiten verringern
- können den Komfort und die Zugangsmöglichkeiten für Benutzer erhöhen und
- unterstützen ein breites Spektrum von Anwendungen.

Auf dieser Basis erstellte Dokumente haben darüber hinaus den Vorteil, auch für Menschen klar und verständlich lesbar zu sein.

³ XML - Extensible Markup Language, eine Teilmenge von SGML (Standard Generalized Markup Language)

⁴ Ein XML-Dokument wird als *wohlgeformt* bezeichnet, wenn seine Syntax den Anforderungen von XML genügt. Lässt sich das Dokument darüber hinaus noch erfolgreich gegen eine Document Type Definition (DTD) oder ein XML-Schema validieren, so wird es als *gültig* bezeichnet.

⁵ Zumindest sollte es so sein - der Internet Explorer von Microsoft erfordert auch in der aktuellen Version 7 hier und da noch eine Sonderbehandlung.

Ein Vorreiter bei der Standardisierung des World Wide Web (WWW) ist zweifelsohne das World Wide Web Consortium (W3C)⁶. Das 1994 von *Tim Berners-Lee* gegründete, aus internationalen Firmen und Organisationen⁷ bestehende W3C ist zwar keine zwischenstaatlich anerkannte Organisation, seine - deshalb *Empfehlungen* (engl.: *Recommendations*) genannten - Standards haben jedoch trotzdem ein hohes Gewicht. Im Folgenden soll auf ausgewählte, bedeutende Standards kurz näher eingegangen werden.

XML / XSL(T)

XML ist eine Auszeichnungssprache⁸, die gezielt vom W3C spezifiziert wurde⁹, um Informationen über das WWW bereitzustellen. „Das Ziel ist es zu ermöglichen, generisches SGML in der Weise über das Web auszuliefern, zu empfangen und zu verarbeiten, wie es jetzt mit HTML möglich ist. XML wurde entworfen, um eine einfache Implementierung und Zusammenarbeit sowohl mit SGML als auch mit HTML zu gewährleisten.“¹⁰ Um dieses Ziel zu erreichen, wurde XML unter der Berücksichtigung von 10 Design Goals entworfen¹¹, die darauf abzielen, möglichst schnell eine klar lesbare, einfach handhabbare und leicht einsetzbare Sprache zur Verfügung zu stellen. Grundsätzlich werden mit XML eigene Elemente definiert, die verwendet werden können, um Dokumenttypen zu definieren, die mit den Möglichkeiten der vordefinierten Elemente in HTML nicht ausreichend beschrieben werden können, z.B.

- Dokumente, die nicht aus typischen Komponenten bestehen (z.B. Absätze, Listen, Tabellen),
- Datenbank-ähnliche Dokumente oder
- Dokumente, die in einer hierarchischen Baumstruktur organisiert werden sollen.¹²

⁶ Vgl. <http://www.w3.org/>.

⁷ Eine Mitgliederübersicht ist unter <http://www.w3.org/Consortium/Member/List.php3> abrufbar.

⁸ Diese Aussage ist etwas umstritten; es finden sich auch Lehrmeinungen, die besagen, XML wäre lediglich eine Metasprache, mit deren Hilfe beliebige andere Auszeichnungssprachen definiert werden können. Für beide Sichtweisen gibt es plausible Begründungen. Zum einen steckt in XML explizit die Bezeichnung „Markup Language“, zum anderen muss die eigentliche Sprache für eine XML-Applikation (z.B. XHTML oder Docbook) tatsächlich erst mit den Ausdrucksmitteln von XML festgelegt werden.

⁹ Vgl. <http://www.w3.org/TR/REC-xml/>.

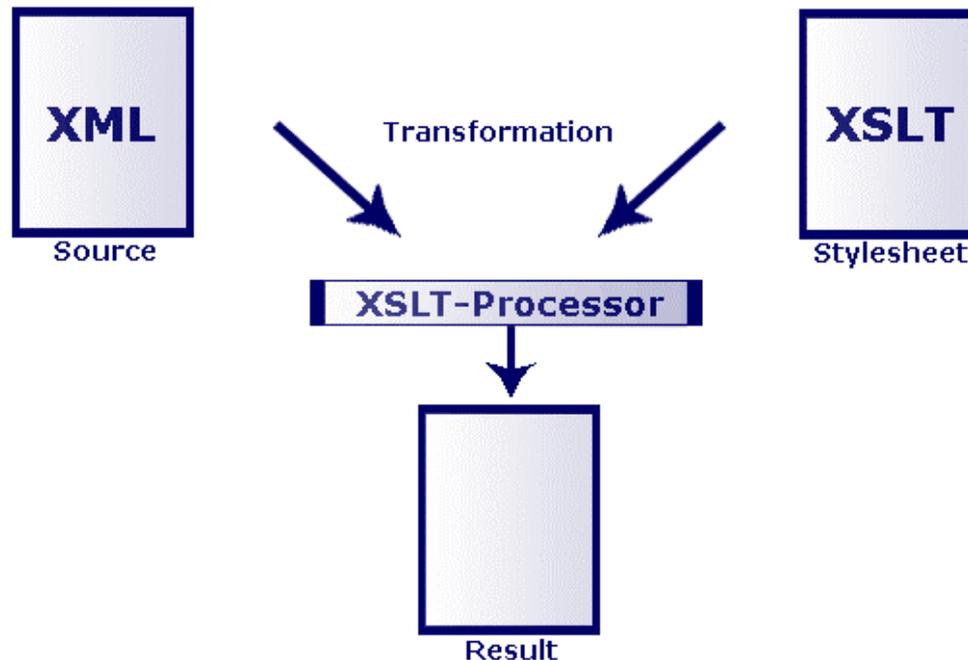
¹⁰ Young, M. J. (2001), S. 25.

¹¹ Vgl. <http://www.w3.org/TR/REC-xml/>.

¹² Vgl. Young, M. J. (2001), S. 29.

Das Vokabular der verwendeten XML-Sprache kann gegen Document Type Definitions (DTDs) oder XML-Schemas validiert werden, wobei letztere wiederum selbst in der XML-Syntax verfasst werden. Dies schließt u.a. das Erzwingen bestimmter Element-Abfolgen, die Verwendung von Entities und Namespaces oder die Definition eigener, komplexer Datentypen ein.

Zu Präsentationszwecken eignen sich XML-Dokumente in ihrer Reinform jedoch nur bedingt, da sie die enthaltenen Informationen lediglich in ihrer Struktur beschreiben. Um eine optisch und inhaltlich ansprechende Ausgabe zu erzielen, ist es daher zumeist erforderlich, XML-Dokumente geeignet zu sortieren bzw. zu filtern, umzuformen und zu transformieren. Hierzu können beispielsweise die Extensible Stylesheet Language Transformations (XSLT) verwendet werden. Dabei führt ein XSLT-Prozessor (z.B. Apache Cocoon^{13,14}) eine Transformation durch, an der zumindest drei Dokumente beteiligt sind: ein oder mehrere XML-Quelldokumente, ein XSLT-Stylesheet und das Ergebnisdokument, welches sich aus den Anweisungen des Stylesheets und dem (ggf. modifizierten, s.o.) Inhalt des Quelldokumentes ergibt. Das Stylesheet kann beispielsweise HTML-Elemente, aber auch noch zusätzlichen (statischen) Inhalt enthalten¹⁵. Abb. 2.1 stellt die Transformation schematisch dar.



Quelle: Vgl. Bongers, F. (2005), S. 28.

Abb. 2.1: Schematische Darstellung einer Transformation

¹³ siehe <http://cocoon.apache.org/>.

¹⁴ Weiterführende Literatur: Niedermeier, S. (2006).

¹⁵ Vgl. Bongers, F. (2005), S. 27 ff.

Über XSL Formatting Objects (XSL-FO) sind auch andere Ausgabeformate, wie z.B. PDF zur Bereitstellung im Netz oder als Druckvorlage denkbar.

XHTML

Die Extensible Hypertext Markup Language (XHTML) ist eine „Erweiterung der Seitenbeschreibungssprache HTML, so dass sie den XML-Regeln entspricht. Damit wird die Trennung von Inhalt und Layout bei gleichzeitiger voller Kontrolle über die Gestaltung möglich.“¹⁶ XHTML 1.0¹⁷ Transitional ist dabei zur letzten HTML-Version 4.01 kompatibel. Für XHTML 1.0 Strict und die noch in der Entwicklung befindliche Version 2.0 ist dies nicht mehr gegeben. Die Notation ist aber grundsätzlich restriktiver geworden. Dies ergibt sich aus den strengeren syntaktischen Anforderungen von XML im Vergleich zur SGML, aus der HTML abgeleitet wurde. (X)HTML ist nach wie vor das Grundgerüst zur Erstellung von Oberflächen in Webapplikationen. Allerdings ist zu berücksichtigen, dass z.B. Java Server Pages (JSP) nicht mehr in reinem XHTML geschrieben werden können, da sie das existierende Markup um eigene Elemente bereichern und somit keine Validierung mehr gegen die XHTML-DTD vorgenommen werden kann.

Zur Trennung von Layout und Struktur enthält ein XHTML-Dokument vorzugsweise nur noch die Elemente und ihre Inhalte (und ggf. JavaScript-, PHP-Code, o.Ä. zur Dynamisierung von Inhalten). Die Formatierung der Ausgabe wird im Allgemeinen über Cascading Stylesheets (CSS, s.u.) vorgenommen und nicht mehr über `<p>` - Elemente oder komplex verschachtelte Tabellenlayouts.

CSS

Cascading Stylesheets (CSS) sind ein erstmals 1994 vorgeschlagenes Konzept zur Trennung von Inhalt und Präsentation auf Webseiten¹⁸. Schon 1998 lag die Recommendation zu CSS 2 vom W3C vor, es dauerte jedoch noch einige Jahre, bis alle namhaften Browserhersteller die entsprechenden Spezifikationen in ihren Browsern implementiert hatten.¹⁹ Die Notwendigkeit zur flächendeckenden Verbreitung von CSS ergab sich seit etwa der Jahrtausendwende vor allem aus der Tatsache, dass präzise

¹⁶ Jacobsen, J. (2005), S. 412.

¹⁷ XHTML liegt ebenfalls als W3C-Recommendation vor. Vgl. <http://www.w3.org/TR/xhtml1/>.

¹⁸ Wiederum liegt dieses Konzept als W3-Recommendation vor. Vgl. <http://www.w3.org/TR/REC-CSS2/>.

¹⁹ Zumindest theoretisch; der Internet Explorer von Microsoft interpretiert einzelne Bestandteile auch in der aktuellen Version 7 immer noch anders...

Layouts in HTML ursprünglich nicht vorgesehen und damit nur schwer und auf Umwegen realisierbar waren. Das Ergebnis war ein mit steigenden Anforderungen an Inhalt und Aussehen von Webseiten immer schwieriger wartbarer Cluster aus verschachtelten Tabellen, die zur Schaffung bestimmter Layouts missbraucht wurden. Dabei sind Tabellen als Gestaltungsmittel aus mehreren Gründen denkbar ungeeignet:

- Sie führen zu unnötig langen Ladezeiten.
- Sie verführen zum Einsatz von uneffizienten Blind- bzw. Leergrafiken zur Gestaltung, die den Seitenaufbau verzögern, da Webbrowser Tabellen als ganze Einheit vom Server laden und die Inhalte erst darstellen, wenn die gesamte Tabelle auf dem Client verfügbar ist.
- Die Wartung und Erweiterung von Seiten mit Tabellenlayouts ist hochgradig problematisch, da schon geringe Änderungen im Quellcode das gesamte Seitenlayout zerstören können.²⁰
- Eine einheitliche Darstellung auf verschiedenen Plattformen kann nicht gewährleistet werden.
- Tabellenlayouts führen zu einer drastischen Beschränkung der Accessibility einer Seite (s.u.).

All diese und andere Probleme lassen sich durch den Einsatz von CSS umgehen. Dabei haben Cascading Stylesheets im direkten Vergleich mit XSLT-Stylesheets (s.o.) noch weitere Vorteile: Sie sind leichter zu handhaben, was hauptsächlich am geringeren Sprachumfang und damit einer geringeren Komplexität liegt. Darüber hinaus unterstützen aktuelle Browser CSS bis einschließlich Version 2, es wird also im Gegensatz zu XSLT auch kein separater Transformationsprozessor benötigt.

Zwar lassen sich CSS-Formatierungsanweisungen direkt in ein XHTML-Quelldokument integrieren, dies widerspricht aber dem Designziel einer Trennung von Inhalt und Layout. Im Normalfall sollten die Anweisungen deshalb in einem externen Stylesheet enthalten sein, welches im Quelldokument referenziert wird. Die Formatierungsanweisungen im Stylesheet werden dann mit dem Quelldokument verknüpft, indem sie an einzelne Elemente gebunden werden. Diese Bindung kann über IDs erfolgen, mit denen die XHTML-Elemente eindeutig identifizierbar sind, oder über Klassenbezeichner, falls bestimmte Formatierungen auf mehrere Elemente gleichermaßen angewandt werden sollen. Da die Klassifikation in diesem Fall

²⁰ Vgl. Shafer, D. / Yank, K. (2004), S. 5 ff.

ausschließlich zu Layoutzwecken genutzt wird, können Elemente in einer Klasse zusammengefasst werden, die in ihrer Art und ihrem Anwendungsgebiet eigentlich grundverschieden sind. Mittels der Kaskadierung ist es innerhalb eines oder mehrerer Stylesheets möglich, Formatierungsanweisungen zu überschreiben. Auf ein Element oder eine Klasse wird immer genau die Formatierung angewandt, die als letzte spezifiziert wurde. So ist es beispielsweise möglich, in einem Stylesheet eine grundlegende Formatierung im Hinblick auf Schriftart, -größe und -farbe vorzugeben, die in einem weiteren Stylesheet dann für einzelne Elemente bei Bedarf detailliert individualisiert werden kann.

Zur Positionierung von Elementen wird in CSS ein enorm flexibles Boxmodell verwendet. „Das Boxmodell ist einer der Grundpfeiler von CSS und schreibt vor, wie Elemente dargestellt werden und - bis zu einem gewissen Grad - wie sie sich gegenseitig beeinflussen. Jedes Seitenelement wird als rechteckiger Container (oder ‚Box‘) betrachtet, der die folgenden Bestandteile umfasst: Elementinhalt (Text, Grafik), Innenabstand (`padding`), Rahmen (`border`) und Rand bzw. Außenabstand (`margin`).“²¹ Abb. 2.2 stellt das Boxmodell schematisch dar. Innerhalb dieses Boxmodells können nun Blockelemente mit verschiedenen Verfahren flexibel positioniert werden, um mehrspaltige oder wie auch immer geartete Layouts zu konstruieren. Unterschieden wird dabei zwischen

- Relativer Positionierung (innerhalb des normalen Dokumentflusses; das Element nimmt immer seinen ursprünglichen Raum ein, auch wenn es verschoben wird - es kann daher zu Überlappungen mit anderen Boxen kommen),
- Absoluter Positionierung (das Element wird aus dem normalen Dokumentfluss herausgelöst und ist für relativ positionierte Elemente nicht mehr existent) und
- Schwimmender Positionierung (sog. *Floating*; das Element befindet sich ebenfalls nicht im normalen Dokumentfluss, wird aber automatisch an anderen Float-Boxen oder am Elternelement ausgerichtet und behält diese Ausrichtung auch bei Layoutänderungen bei).^{22,23}

Grundsätzlich lässt sich daher mit den gegebenen Möglichkeiten jedes in reinem (X)HTML erstellte Seitendesign in ein auf CSS basierendes Layout umwandeln, ohne dass dem Betrachter mit bloßem Auge ein Unterschied auffällt.²⁴ Das CSS-Layout

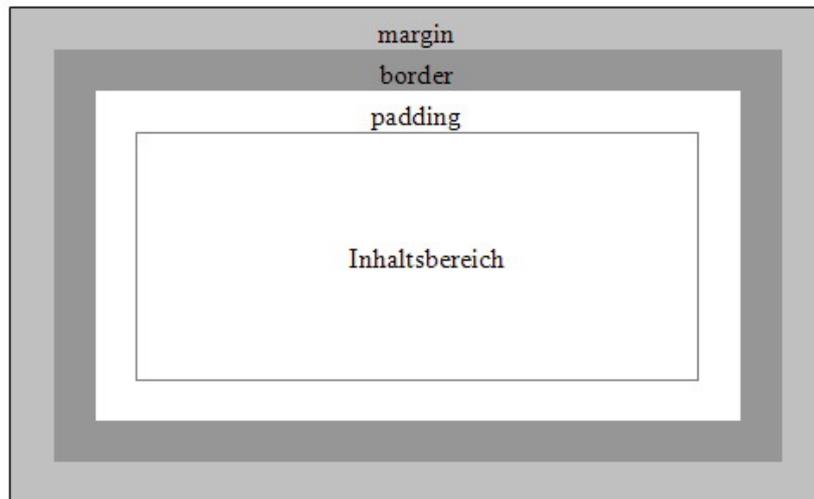
²¹ Budd, A. / Moll, C. / Collison, S. (2007), S. 32

²² Der Internet Explorer bis einschließlich Version 6 enthält diesbezüglich mehrere bekannte Bugs und unterstützt die Positionierungsmodelle in manchen Fällen nur sehr unzureichend.

²³ Vgl. Budd, A. / Moll, C. / Collison, S. (2007), S. 39 ff.

²⁴ Vgl. hierzu u.a. Meyer, E. (2005), S. 2 ff. und S. 225 ff.

reduziert dafür in nahezu allen Fällen den Umfang des Quellcodes und damit die Ladezeiten, ist schneller zu implementieren und zu warten und vereinfacht Updates und Änderungen an bestehenden Webseiten. Schlussendlich lassen sich komplette Neugestaltungen von Layouts inklusive einer geänderten Abfolge der Elementinhalte in der Darstellung durchführen, ohne das Quelldokument überhaupt verwenden zu müssen.²⁵



Quelle: Vgl. Budd, A. / Moll, C. / Collison, S. (2007), S. 32

Abb. 2.2: Schematische Darstellung des CSS-Boxmodells

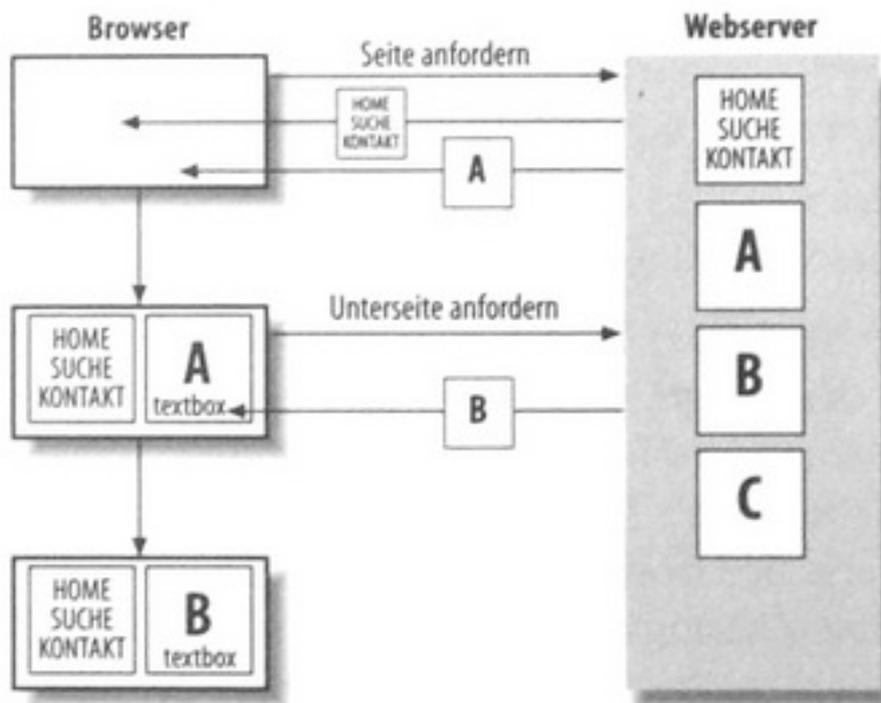
AJAX

Hinter dem erstmals 2005 in einem Aufsatz von Jesse James Garrett verwendeten Begriff AJAX (Asynchronous JavaScript And XML) verbirgt sich ein Konzept, welches wesentlichen Anteil an der Durchsetzung so genannter Web 2.0 - Anwendungen hat. Die Grundidee ist es, zwischen Client und Server einen bidirektionalen Datenaustausch vorzunehmen, ohne dass eine auf XHTML/CSS basierende Website hierzu einem kompletten Reload unterzogen werden muss. Die versendeten oder nachgeladenen Daten können vielmehr punktuell bestimmten Seitenbereichen zugewiesen werden, ohne dass die restliche Seite in Form eines vollständigen Antwortdokumentes erneut vom Server angefordert werden muss. Der klare Vorteil zur bisherigen Kommunikation liegt in der Reduzierung von Latenzzeiten, da die Kommunikation mit dem Webserver auf das Nötigste beschränkt werden kann. Webanwendungen sollen in ihrem Verhalten

²⁵ Ein sehr gelungenes Beispiel hierfür stellt der 2003 von Dave Shea ins Leben gerufene CSS-Zen-Garden dar: Webdesigner aus aller Welt schaffen CSS-basierte Layouts für eine vorgegebene HTML-Datei, deren Inhalt nicht modifiziert werden kann. Derzeit liegen auf <http://www.csszengarden.com/> über 1200 Designs vor. Vgl. hierzu auch Shea, D. / Holzschlag, M. E. (2005), S. 54 ff.

auf diesem Wege näher an Desktopapplikationen heranrücken, indem ihre Geschwindigkeitsnachteile reduziert werden. Die grundsätzliche Funktionsweise von Ajax wird in Abb. 2.2 dargestellt.

Unabdingbar für den Einsatz von AJAX in einer Webanwendung sind gewisse technische Voraussetzungen, die in dieser Form erst seit etwa zwei bis drei Jahren gegeben sind. So müssen die verwendeten Browser die XMLHttpRequest-Programmierschnittstelle (Application Programming Interface, kurz API) unterstützen²⁶, um eine asynchrone Nachforderung von Daten möglich zu machen. Dieser Ansatz ist grundsätzlich nicht neu, sondern wurde von Microsoft bereits seit 1998 in verschiedenen Produkten (z.B. Exchange Server) integriert. Es handelte sich dabei allerdings immer um „[...] isolierte Anwendungen, die sich aber nicht flächendeckend durchsetzen konnten.“²⁷ Erst in den letzten Jahren haben auch andere Browserhersteller entschieden, diese API in ihren Produkten zu implementieren.



Quelle: Carl, D. (2006), S. 4.

Abb. 2.3: Die grundsätzliche Funktionsweise von AJAX

²⁶ Das XMLHttpRequest-Objekt stellt diese Programmierschnittstelle innerhalb von JavaScript in Form eines Objektes zur Verfügung. Im Internet Explorer der Firma Microsoft wird die API bis einschließlich Version 6 unter dem Namen XMLHttpRequest als ActiveX-Komponente implementiert, was trotz diverser Updates ein nicht unerhebliches Sicherheitsrisiko darstellt. Erst mit der 2006 veröffentlichten Version 7 hat auch Microsoft die ActiveX-Komponente durch das nativ implementierte XMLHttpRequest ersetzt.

²⁷ Steyer, R. (2006), S. 33.

Grundsätzlich kann das XMLHttpRequest-Objekt auf reinen Textdaten operieren. Liegen die zu übertragenden Daten jedoch im XML-Format vor, so bietet das Objekt Methoden und Eigenschaften an, die verwendet werden können, um auf komplex strukturierte XML-Dokumente komfortabel zugreifen zu können. Dazu wird das Document Object Model²⁸ (DOM) verwendet, welches alle Elemente eines XML-Dokumentes in einer Baumstruktur repräsentiert, die wiederum die Hierarchie des Markups darstellt.²⁹ Mit den Methoden des XMLHttpRequest-Objektes ist es nun möglich, über Referenzen auf beliebige Elemente lesend oder schreibend zuzugreifen und entsprechend die Elementinhalte zu modifizieren, ohne dass die umgebende Website in diesen Prozess mit eingebunden werden muss.³⁰ Typische AJAX-Vertreter sind z.B. die Suchmaschine Google Suggest³¹ oder Widget-Anwendungen, wie das oftmals auch als Online-Desktop bezeichnete Netvibes-Dashboard³².

Grundsätzlich ist AJAX ein sehr mächtiges Konzept mit vielfältigen Anwendungsmöglichkeiten. Dessen ungeachtet gibt es jedoch auch eine Reihe von Nachteilen und Problemen bei der Verwendung von AJAX, die vor der Implementierung (mit Blick auf die Zielgruppe der Anwendung) bedacht werden sollten.³³

- Hat ein Benutzer in seinem Browser JavaScript deaktiviert (z.B. aus Sicherheitsgründen), so steht die gesamte Anwendung nicht mehr zur Verfügung. Neben dem damit einhergehenden technischen Problem kann dies zusätzlich noch zu einem für den Benutzer völlig inakzeptablen Seitenlayout führen, wenn bestimmte Seiteninhalte gar nicht oder unvollständig geladen werden. Einer solchen Situation kann durch das Angebot eines Fallbacks³⁴ begegnet werden, diese Lösung ist allerdings sehr aufwändig.
- Nutzer des Internet Explorers bis einschließlich Version 6 (die derzeit noch extrem verbreitet sein dürfte), haben auf Grund berechtigter

²⁸ Auch das DOM ist eine Empfehlung des W3C.

²⁹ Allerdings ist das DOM-API des W3C nicht unumstritten. Da die Vorgaben des W3C sehr allgemein und programmiersprachenunabhängig gehalten sind, ist ein wesentlicher Kritikpunkt, dass das DOM-API weder als besonders performant, noch als anwenderfreundlich bezeichnet werden kann. Insbesondere für die Entwicklung von Webapplikationen auf Basis von Java und XML gibt es daher neben der Java API for XML Programming (JAXP) noch eine Reihe weiterer, von der W3C-Recommendation losgelöster DOM-Implementierungen. Vgl. hierzu auch Niedermeier, S. / Scholz, M. (2006), S. 161 ff. und S. 603 ff. sowie Ullenboom, C. (2006), S. 805 ff.

³⁰ Vgl. hierzu Carl, D. (2006), S. 25 ff.

³¹ <http://www.google.com/webhp?complete=1&hl=en>

³² <http://www.netvibes.com/>

³³ Vgl. Carl, D. (2006), S. 35 ff.

³⁴ Ein Fallback bezeichnet ein Alternativangebot, welches zur Verfügung gestellt wird, wenn AJAX nicht ausgeführt werden kann.

Sicherheitsbedenken oftmals die ActiveX-Engine deaktiviert. Auch unter dieser Voraussetzung können AJAX-Applikationen nicht realisiert werden.

- Die Funktion des browserspezifischen „Back“-Buttons wird durch per AJAX geladene und dynamisch veränderte Inhalte ausgehebelt. Gleiches gilt für Bookmarks, da sich ein URL nicht ändert, wenn innerhalb der Seite Elementinhalte ersetzt werden.
- Inhalte, die über JavaScript dynamisch in eine Website geladen wurden, werden von allen gängigen Suchmaschinen ignoriert.³⁵

Im Grunde genommen ist das oftmals nur als Schlagwort verwendete AJAX kein Standard im klassischen Sinne, nicht zuletzt, weil keine Spezifikation oder Empfehlung eines anerkannten Gremiums vorliegt. Das Konzept hat sich allerdings auch so enorm schnell etabliert, ist aus aktuellen Webanwendungen und Web Services nicht mehr wegzudenken und hat deren Funktionalität in kurzer Zeit stark erweitert. Deshalb ließe sich AJAX wohl am ehesten als De-Facto-Standard beschreiben. Vor dem Hintergrund der bereits angesprochenen Nachteile muss das AJAX-Konzept jedoch auch im Kontext der Barrierefreiheit nochmals genau betrachtet werden.

RSS / ATOM

Really Simple Syndication (RSS)³⁶ und das Atom Syndication Format (ATOM)³⁷ sind XML-Standards. Sie beschreiben elektronische Nachrichtenformate zum plattformunabhängigen Nachrichtenaustausch. Dabei ist insbesondere RSS kein unproblematisches Format. Verschiedene Versionen von RSS wurden von verschiedenen, unabhängigen Entwicklergruppen (Version 1.0 auch mit Unterstützung durch das W3C) entwickelt; teilweise existieren inoffizielle Weiterentwicklungen zu einzelnen Versionen; die erste Version (0.90) basierte auf dem Resource Description Framework (RDF), 0.91 und 0.92 nicht mehr, 0.93 und 0.94 kamen über das Entwurfsstadium nicht hinaus, Version 1.0 basierte wieder auf dem RDF, Version 2.0 wiederum nicht - und Version 2.0 ist - entgegen den Aussagen der UserLand-Entwicklergruppe - mit RSS 0.90 nicht mehr vollständig kompatibel. Sogar die Abkürzung RSS hatte bereits mehrere Bedeutungen.^{38,39} Insgesamt hat RSS zwar eine

³⁵ Vgl. Fischer, M. (2006), S. 82 f. sowie S. 210 f.

³⁶ Aktuell standardisiert von UserLand; vgl. <http://www.rssboard.org/rss-specification>.

³⁷ Standard der ATOM Enabled Alliance; vgl. <http://www.atompub.org/>.

³⁸ RSS steht bzw. stand für Rich Site Summary (Version 0.9x), RDF Site Summary (Version 1.0) und Really Simple Syndication (Version 2.0)

³⁹ Vgl. <http://de.wikipedia.org/wiki/Rss>.

hohe Verbreitung gefunden, die Entwicklung lief bislang jedoch ausgesprochen unkoordiniert ab.

Über RSS-Feeds bereitgestellte Inhalte werden automatisch auf den Rechner des jeweiligen Abonnenten übertragen. Sie können zusätzlich auch in Webapplikationen (z.B. Mail-Clients) bereitgestellt werden. Neben Text sind auch Audio- und Videodaten als Inhalte möglich. An diesem Punkt setzt ATOM an. Als wesentlicher Unterschied zu RSS muss in inhaltstragenden Elementen bei Verwendung von ATOM über ein Attribut explizit in maschinenlesbarer Form angegeben werden, um welche Inhaltsform es sich jeweils handelt. Dennoch hat sich ATOM bislang gegenüber RSS noch nicht flächendeckend durchsetzen können.

RDF

Das Resource Description Framework (RDF) ist eine vom W3C entwickelte und standardisierte formale Sprache⁴⁰, die dazu verwendet wird, Webinhalte um maschinenlesbare Metadaten zu ergänzen. Das RDF ist damit einer der Eckpfeiler der Semantic Web - Initiative.⁴¹ Es arbeitet mit sog. *Statements*, mit denen sich Aussagen über beliebige Ressourcen machen lassen. Ein Statement ist immer ein Tripel, bestehend aus

- einem Subjekt,
- einem Prädikat, das eine Aussage zu einem Subjekt macht und
- einem Objekt, dies ist der Wert des Prädikates und kann seinerseits sowohl ein einfaches Literal, als auch wiederum eine weitere Resource sein, die über einen Uniform Resource Identifier (URI) eindeutig identifiziert wird.

Prädikate und Objekte können ihrerseits wieder Subjekte in weiteren Statements sein. Auch Reifikationen sind möglich, d.h. ein komplettes Statement bildet das Subjekt eines anderen Statements. Ein RDF-Dokument ist somit eine Sammlung aus miteinander verlinkten Statements.

⁴⁰ Vgl. <http://www.w3.org/RDF/>.

⁴¹ Das Semantic Web ist ein von Tim Berners-Lee erdachtes Konzept, um Webinhalte mit maschinenlesbaren Metadaten anzureichern, welche die Semantik der Inhalte näher beschreiben sollen. Die Zielstellung ist es, Anfragen auf Grund der Bedeutung ihres Inhaltes und nicht - wie bislang - auf Grund ihrer Schreibweise bearbeiten zu können. Erste Beispiele hierfür sind u.a. das Friend Of A Friend (FOAF) - und das Description Of A Career (DOAC) - Projekt. Siehe hierzu <http://www.foaf-project.org/> bzw. <http://ramonantonio.net/doac/>. Auch die Europäische Union hat unter dem Namen SemanticGov ein Projekt initiiert, um die EU-weite Zusammenarbeit zwischen Behörden zu vereinfachen. Siehe hierzu <http://www.semantic-gov.org/>.

Die Abbildung von RDF-Statements kann über einen beschrifteten und gerichteten so genannten RDF-Graphen erfolgen, für den mit der vom W3C entwickelten SPARQL Protocol And RDF Query Language⁴² (SPARQL) auch eine Anfragesprache vorliegt. Darüber hinaus existieren aber auch zwei Implementierungen für RDF: zum einen die Notation 3⁴³ (N3) von Tim Berners-Lee und zum anderen eine (gebräuchlichere) Syntax auf Basis von XML⁴⁴, in der RDF-Graphen über Elemente mit Inhalten und Attribute mit Attributwerten serialisiert werden.

„Kugelsichere“ Designs

Allein die Tatsache, dass Webstandards existieren und eine Reihe von Vorteilen bieten, führt noch nicht zwangsläufig zu besseren Webseiten. Statistiken zufolge werden auf über 95% (in manchen Quellen über 99%) aller existierenden Webseiten Standards wie die oben beschriebenen nicht oder nur unzulänglich bzw. fehlerhaft eingesetzt.⁴⁵ Aber auch der konsequente und spezifikationsgemäße Einsatz von Webstandards muss nicht unbedingt zu einer Verbesserung der Qualität führen. Zwar kann auf der Entwicklerseite von den angesprochenen Vorteilen profitiert werden, auf der Benutzerseite reicht die reine Standardkonformität jedoch nicht immer aus. Hier sind zusätzliche Überlegungen zur *Accessibility* einer Website anzustellen, um zu verhindern, dass Benutzer beim Versuch, eine Weboberfläche an die eigenen Bedürfnisse anzupassen oder bei anderweitigen unvorhergesehenen Aktionen das Layout über seine Flexibilitätsgrenze hinaus belasten.⁴⁶ Dies soll kurz an einem sehr einfachen Beispiel mit CSS verdeutlicht werden.

Ein sehr beliebtes Gestaltungsmittel bei Verwendung von CSS ist es, die eigentlich rechteckigen Boxen aus dem Boxmodell mit abgerundeten Ecken zu versehen. Dies wird über Hintergrundgrafiken ermöglicht, die in vielen Fällen zur visuellen Auflockerung auch noch einen dezenten Farbverlauf erhalten. Der optische Effekt ist sehr ansprechend (Abb. 2.4) und die benötigten CSS-Anweisungen bleiben überschaubar.⁴⁷

⁴² Vgl. <http://www.w3.org/TR/rdf-sparql-query/>.

⁴³ Vgl. <http://www.w3.org/DesignIssues/Notation3.html>.

⁴⁴ Vgl. <http://www.w3.org/TR/rdf-syntax-grammar/>.

⁴⁵ Vgl. Zeldman, J. (2006), S. 23 ff.

⁴⁶ *Dan Cederholm* hat für solcherart strapazierfähige Layouts den Begriff des „Bulletproof Webdesign“ eingeführt - daher auch die Überschrift.

⁴⁷ Das Beispiel ist auf die Anpassung des Tags `<h2>` für den Titel der Box fixiert. Prinzipiell gelten die Ausführungen natürlich auch für die Unterkante einer solchen Box, die ggf. separat behandelt werden muss.

Die Hintergrundgrafik ist Bestandteil des Layouts und wird somit nicht im Quelldokument, sondern im Stylesheet eingebunden. Die Anweisung umfasst lediglich eine Zeile (auf weitere Formatierungen wird aus Gründen der Übersichtlichkeit verzichtet):

```
h2
{
    background:url('h2_back.gif') no-repeat top left;
}
```

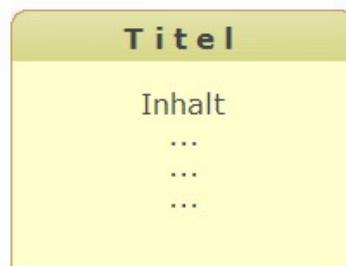


Abb. 2.4: CSS-formatierte Box mit abgerundeten Ecken

Im Normalfall wird es mit dieser - in allen Belangen standardkonformen - Lösung keine Probleme geben. Wenn sich ein Benutzer jedoch (z.B. auf Grund einer Sehschwäche) dazu entschließt, die Schriftgröße im Browser manuell zu verändern, kann die Hintergrundgrafik sich ab einem bestimmten Grad nicht mehr ausreichend anpassen. Das Layout wird je nach Anwendungsfall gravierend beeinträchtigt (Abb. 2.5).



Abb. 2.5: Box mit manuell veränderter Schriftgröße

Innerhalb gewisser Grenzen lässt sich einem solchen Problem begegnen, indem die Grafik entsprechend größer gewählt und der jeweils nicht benötigte Teil hinter der Inhaltsfläche versteckt wird. Größere Grafiken benötigen jedoch auch mehr Webspace und längere Ladezeiten. Darüber hinaus stößt diese Lösung spätestens dann an Grenzen, wenn die Überschrift durch inhaltsbedingte Änderungen erweitert oder gar auf mehrere Zeilen ausgedehnt werden muss. Abgesehen vom Imageschaden, der durch einen solchen Fehler gegenüber dem Benutzer entsteht, geht damit auch ein wesentlicher

Vorteil der Verwendung des CSS-Standards verloren: Die schnelle bzw. sogar automatische Anpassung des Layouts an geänderte Inhalte ist nicht mehr gegeben.

Mit einer „kugelsicheren“ Lösung nach *Dan Cederholm* lässt sich dieses Problem beheben.⁴⁸ Zunächst wird die Box mit CSS rechteckig gestaltet, mit zu den ursprünglichen Grafiken passenden Füllfarben ausgestaltet und mit einem entsprechenden Rahmen versehen. Dazu werden nur noch zwei winzige Grafiken benötigt. Diese enthalten ausschließlich die beiden Rundungen und werden links bzw. rechts oben im `<h2>`-Teil der Box fixiert, um die Ecken des CSS-Modells zu überlagern. Gegebenenfalls kann noch eine weitere kleine rechteckige Grafik verwendet werden, die den Farbverlauf erhält und mit der CSS-Anweisung `repeat-x` über die Breite der Box beliebig oft wiederholt werden kann. In gleicher Weise kann natürlich auch mit der Unterkante der Box verfahren werden. Das Ergebnis ist eine Box, die mit deutlich weniger Speicherplatz und somit kürzeren Ladezeiten auskommt, dabei aber beliebige inhaltliche Erweiterungen einerseits und Anpassungen durch den Benutzer andererseits verwalten kann, ohne beschädigt zu werden.

Dieses einfache Beispiel steht stellvertretend für eine ganze Reihe von Weboberflächen, die - trotz konsequent standardkonformer Gestaltung - unter bestimmten Bedingungen nicht wie erwartet funktionieren (müssen). Neben ausführlichen Tests auf z.T. immer noch nicht standardkonformen Browsern ist zur Vermeidung solcher Situationen bei der Entwicklung insbesondere auf die Accessibility zu achten: Ist die Webseite bzw. die Webanwendung weitgehend frei von Barrieren für verschiedene Benutzergruppen? Wie verhält sie sich, wenn (behinderte) Benutzer Anpassungen vornehmen oder andere Ausgabegeräte als den Bildschirm verwenden? Mit diesen Fragestellungen beschäftigt sich das folgende Kapitel zur Barrierefreiheit.

⁴⁸ Vgl. Cederholm, D. (2006), S. 126 ff.

3 Barrierefreiheit

„Für die meisten Menschen ist das Internet mittlerweile zur Selbstverständlichkeit geworden. Informationen aus aller Welt stehen zum Abruf bereit. [...] Von dieser Entwicklung profitieren jedoch nicht alle. Gerade Menschen, denen durch irgendeine körperliche oder geistige Behinderung die volle Teilnahme am gesellschaftlichen Leben erschwert ist, könnten von der Kommunikationstechnik enorm profitieren. Doch sie scheitern oft an Barrieren, die ihnen den Zugang zu Informationen oder die Nutzung von Angeboten erschweren oder ganz unmöglich machen. [...] Barrierefreies Webdesign zielt dementsprechend darauf ab, Inhalte und Interaktionen im Netz für (möglichst) alle Nutzergruppen und Endgeräte zugänglich zu machen.“⁴⁹

Das Streben nach einer möglichst hohen Accessibility ist nicht neu. Aber erst in den vergangenen Jahren ist es - insbesondere durch das Aufkommen und die zunehmende Verbreitung der Webstandards - möglich geworden, diesen Anforderungen Rechnung zu tragen und Barrierefreiheit technisch zufriedenstellend umzusetzen, was wiederum dazu geführt hat, dass diese Problematik mehr und mehr in den Vordergrund rückt. Die Umsetzung der Barrierefreiheit ist bei der Entwicklung bzw. Anpassung von Webseiten zwar im Normalfall mit einem höheren Aufwand verbunden, die Notwendigkeit hierfür liegt jedoch auf der Hand: Statistiken zufolge haben alleine in Deutschland etwa 8 % der Internetbenutzer eine (oder mehrere) physische Einschränkung(en), die ihnen die Nutzung von Webservices oder anderen Online-Angeboten erschweren. Diese Einschränkungen können vielfältiger Natur sein. Sie reichen von leichten Sehbehinderungen bis zur Blindheit, beinhalten aber auch kognitive Behinderungen, Gehörschäden, welche die Aufnahme akustischer Inhalte erschweren oder unmöglich machen, Krankheiten wie etwa Epilepsie oder Probleme mit Eingabegeräten. So ist z.B. nicht jeder Nutzer in der Lage, eine Maus oder eine Tastatur zu bedienen, einem anderen kann es wiederum Schwierigkeiten bereiten, grelle Farben zu erfassen. Zusätzlich können weitere Hürden den Zugriff erschweren, die sich grob in folgende Kategorien unterteilen lassen:

- ortsgebundene Hürden - beispielsweise langsame Netzanbindungen, ungünstige Lichtverhältnisse oder ein verordneter Verzicht auf Ton,
- Verständnisprobleme - hierzu gehören fehlende Mehrsprachigkeit (inkl. eines Zusatzangebotes in Gebärdensprache) und mangelnde Alternativbeschreibungen

⁴⁹ Radtke, A. / Charlier, M. (2006), S. 1.

in Text und/oder Ton, aber auch undurchsichtige oder ungewohnte Navigationen und nicht ausreichend verständlich aufbereitete Textinhalte⁵⁰, sowie

- technische Hürden - diese sind bedingt durch unterschiedlich stark ausgeprägte Standardkonformität bei Webbrowsern, die Verwendung nicht oder nur schwer erhältlicher zusätzlicher Plugins und die fehlende Anpassung von Inhalten an verschiedene Endgeräte, wie z.B. Mobiltelefone, PDAs und assistive Techniken (Screenreader bzw. Braillezeile).⁵¹

Der Begriff der Barrierefreiheit umfasst also die Beseitigung einer Vielzahl möglicher Beeinträchtigungen beim Zugriff auf Inhalt und Layout einer Weboberfläche. Bei allen Bemühungen zur Verbesserung der Accessibility kann jedoch davon ausgegangen werden, dass eine 100%ige Umsetzung nicht gelingen wird, da einige Anforderungen, etwa zur farblichen Gestaltung, auch miteinander in Widerspruch stehen können.⁵² Grundsätzlich gilt daher, dass jeder Schritt zur Vermeidung von Barrieren wertvoll ist, auch wenn in Summe nicht alle Vorgaben umgesetzt werden können. Ausnahmen bilden jene Organisationen, in denen Barrierefreiheit gesetzlich vorgeschrieben ist. Dies betrifft hauptsächlich Einrichtungen und Körperschaften des öffentlichen Rechts (gleichermaßen in Bund und Ländern).⁵³ In Deutschland sind die hierzu erforderlichen Maßnahmen - zumindest auf Bundesebene - in der „Barrierefreien Informationstechnik - Verordnung“ (BITV) geregelt.⁵⁴ Für die Länder gelten eigene Gleichstellungsregelungen, die sich jedoch in weiten Teilen an der BITV orientieren. Ursprung für die BITV und ähnliche Verordnungen sind die 1999 vom W3C standardisierten Web Content Accessibility Guidelines (WCAG) 1.0⁵⁵ der Web Accessibility Initiative (WAI)⁵⁶.

Eine Grundvoraussetzung für die Erstellung möglichst barrierearmer Seiten ist einmal mehr das Einhalten von Webstandards, hier natürlich insbesondere XHTML und CSS. Entgegen landläufig verbreiteter Meinungen muss damit keine Einschränkung im Seitendesign einhergehen. Die konkreten Maßnahmen sind äußerst vielfältig und wesentlich von den zu verbreitenden Inhalten und der Zielgruppe abhängig. An dieser Stelle sollen daher nur einige ausgewählte allgemeine Richtlinien angeführt werden.

⁵⁰ Insbesondere die Online-Angebote von Behörden können erheblich darunter leiden.

⁵¹ Vgl. Radtke, A. / Charlier, M. (2006), S. 2.

⁵² Beispielsweise können Menschen mit Fehlsichtigkeiten die Elemente einer Weboberfläche nur durch starke Kontraste voneinander trennen - bei anderen Personen kann dies wiederum eine Überreizung hervorrufen und den Inhalt schwer lesbar machen.

⁵³ Es gibt Bestrebungen, in naher Zukunft auch verbindliche Regelungen für Unternehmen und ggf. private Einrichtungen durchzusetzen.

⁵⁴ Vgl. <http://www.gesetze-im-internet.de/bitv/index.html>.

⁵⁵ Vgl. <http://www.w3.org/TR/WAI-WEBCONTENT/>.

⁵⁶ Vgl. <http://www.w3.org/WAI/>.

- **Strukturierung:** Ein erster, bedeutender Schritt ist es, die in XHTML verfügbaren Elemente ausschließlich auf Grund ihrer semantischen Bedeutung einzusetzen. Beispielsweise sollten Überschriften immer hierarchisch durch die Tags `<h1>` bis `<h6>` abgebildet werden. So können assistive Techniken, wie z.B. Screenreader, erkennen, welche Bedeutung der jeweilige Inhalt im Gesamtkontext hat und den Benutzer entsprechend informieren. Tabellen sollten nicht zur Positionierung dienen, sondern ausschließlich dann verwendet werden, wenn der darzustellende Inhalt von seinem Charakter her ein Tabellenlayout bedingt (z.B. ein Finanzbericht o.ä.). Zusätzlich ist auf die Anordnungsreihenfolge der Seiteninhalte im Quelldokument zu achten. Unabhängig von der Positionierung von Elementen auf einer Webseite geben Screenreader und auch Braillezeilen den Inhalt immer in jener Abfolge wieder, in der er auch im Quelldokument spezifiziert ist. Für den Benutzer ist es entsprechend mit Zeitaufwand und Frustration verbunden, wenn ihm auf jeder Webseite zunächst die kompletten Menüs, Statuszeilen, Grafiktexe u.a. dargeboten werden, bevor der eigentliche Inhalt erfasst werden kann.
- **Besondere Auszeichnungen:** Akronyme, Abkürzungen und Sprachwechsel müssen im Quelldokument durch `<acronym>`, `xml:lang` und ähnliche Markierungen kenntlich gemacht werden. So kann gewährleistet werden, dass beim Zugriff mittels assistiver Techniken Inhalte korrekt vorgelesen bzw. interpretiert werden können. Auch Bilder müssen - so es sich nicht um grafische Elemente handelt, die einzig dem Seitenlayout dienen (wie etwa Grafiken für Menüleisten) - über entsprechende `<alt>`-Texte oder sogar aus externen Dokumenten eingebundene `<longdesc>`-Beschreibungen gekennzeichnet und erläutert werden. Enthält eine Abbildung keine inhaltsrelevante Information, so kann sie ggf. als Hintergrundgrafik im Stylesheet deklariert werden und ist für Hilfsgeräte damit nicht mehr sichtbar.
- **Interaktion:** Während insbesondere Webformulare klar bezüglich ihrer Strukturierung eingeteilt und abgegrenzt werden sollten, empfiehlt es sich auch, interaktive Schaltflächen bevorzugt mittels CSS zu gestalten und auf JavaScript- oder Flash-Anwendungen weitestgehend zu verzichten.
- **Skalierung:** Schriftarten sollten generell vom Benutzer skalierbar sein und deshalb nur in relativen, nicht in absoluten Werten angegeben werden.⁵⁷

⁵⁷ Vgl. Radtke, A. / Charlier, M. (2006), S. 55 ff. und http://de.wikipedia.org/wiki/Barrierefreies_Internet.

Zur Überprüfung, ob alle Maßnahmen zur Gestaltung einer barrierefreien bzw. zumindest barrierearmen Weboberfläche ergriffen und korrekt umgesetzt wurden, ist eine manuelle Kontrolle unumgänglich. In einigen Bereichen können aber ergänzend so genannte *Accessibility-Tester* zum Einsatz kommen, welche die Einhaltung bestimmter Richtlinien selbstständig überprüfen.⁵⁸ Stellvertretend seien hier Cynthia Says⁵⁹ und die Mozilla Accessibility Toolbar⁶⁰ erwähnt, die als Browser-Plugin beispielsweise ermöglicht, CSS und Scripting zu deaktivieren, bestimmte Elemente hervorzuheben, oder High Contrast Stylesheets zu Testzwecken einzubinden.

Auch bei Berücksichtigung aller Maßnahmen zur Gestaltung eines möglichst zugänglichen Internetauftrittes bleiben einige Problemformate, deren Einsatz gesondert betrachtet werden muss. Besonders zu erwähnen sind in diesem Zusammenhang PDF-Dokumente und Flash-Animationen.

Für den Einsatz von PDF-Dokumenten spricht, dass sie auf allen Plattformen exakt aussehen, wie das Print-Äquivalent. Zum Betrachten wird allerdings ein externes Plugin benötigt, und für Menschen mit Behinderungen sind diese Dokumente oftmals nur schwer zugänglich. PDFs sollten deshalb nur eingesetzt werden, wenn Funktionalitäten benötigt werden, die XHTML in dieser Form nicht zur Verfügung stellen kann, z.B.:

- „Dokumente mit Schutz gegen Veränderungen des Inhalts (Rechnungen, Terminalsachen),
- Wissenschaftliche Texte, bei denen die Seitenaufteilung oder die Zuordnung von Fußnoten große Bedeutung hat oder die besondere Grafikelemente (z.B. mathematische Formeln) enthalten,
- Werbeschriften oder Produktinformationen mit hohem Corporate Identity - (CI-) Faktor, die weniger als Webdokumente fungieren, sondern das Internet primär als Verteilermedium nutzen,
- Interaktive Formulare mit höherem Funktionsumfang als derzeit mit HTML realisierbar“.⁶¹

Neben zunehmender Funktionalität von Screenreadern existieren inzwischen (wenige) Programme, mit denen es möglich ist, die semantische Struktur von PDF-Dokumenten

⁵⁸ Beispielsweise kann ein solches Tool verifizieren, ob alle Bilder einer Website mit zusätzlichen Alternativtexten versehen sind oder alle verwendeten Links tatsächliche valide sind. Der Umfang variiert natürlich je nach verwendeter Testsoftware.

⁵⁹ Vgl. <http://contentquality.com/>.

⁶⁰ Vgl. <http://firefox.cita.uiuc.edu/index.php>.

⁶¹ Radtke, A. / Charlier, M. (2006), S.205.

durch Tags hervorzuheben. Die Erstellung solcher *Tagged PDFs* ist allerdings derzeit noch mit hohem Aufwand verbunden, außerdem ist das benötigte Know-How noch relativ dünn gesät.

Die Zugänglichkeit von Flash-Animationen⁶² hat sich hingegen in den vergangenen Jahren kaum verbessert. Zwar hat Adobe⁶³ diesbezüglich einige Anstrengungen unternommen, allerdings auf Basis der Microsoft Active Accessibility - Schnittstelle (MSAA), welche ab Vista nicht länger Bestandteil der Windows-Betriebssysteme ist. Zusätzlich zur oftmals schwierigen Beschaffung des lediglich abwärtskompatiblen Plugins kommt, dass die Bedienung eines Flash-Objektes innerhalb einer Webseite nur über die Tastatur nicht möglich ist. Zwar gibt es durchaus einige Möglichkeiten, Flash zugänglicher zu gestalten, von den verfügbaren assistiven Technologien wird dies jedoch nur sehr unvollständig unterstützt. Im Sinne einer möglichst hohen Accessibility bietet es sich daher eher an, auf den Einsatz von Flash-Animationen nach Möglichkeit zu verzichten.⁶⁴

Auch der Einsatz des inzwischen sehr verbreiteten AJAX-Konzeptes ist der Barrierefreiheit nicht zuträglich. Negativ hervorzuheben sind die bereits in Kapitel 2 angesprochenen Nachteile (Verwendung von JavaScript bzw. ActiveX, keine Back-Funktionalität im Browser, keine Bookmarks für mittels AJAX geladene Inhalte). Dies spricht nicht grundsätzlich gegen den Einsatz von AJAX, es ist lediglich zu berücksichtigen, dass Alternativangebote eingerichtet werden, um Menschen mit Behinderungen die Nutzung nicht zusätzlich zu erschweren oder gar unmöglich zu machen.

Zu guter Letzt sei noch erwähnt, dass auch die Verwendung von Content Management Systemen (CMS) problematisch ist. Verbreitete CMS stellen die Inhalte oftmals noch auf Basis aufgeblähter Tabellenstrukturen dar und sind in dieser Form ganz klar nicht barrierefrei. Auch eine sinnvoll strukturierte Ausgabe der Inhalte gemäß ihrer semantischen Bedeutung ist zumeist nur möglich, wenn gravierende und aufwändige Anpassungen des Kernsystems vorgenommen werden. In den Entwicklergruppen der Open-Source-Systeme Joomla!⁶⁵ und Typo3⁶⁶ werden mittlerweile aber verstärkt Anstrengungen unternommen, um die Systeme von Haus aus zugänglicher zu gestalten.

⁶² Die große Stärke von Flash, die zweifelsohne zur raschen Verbreitung beigetragen hat, liegt in der Erstellung interaktiver Oberflächen unter Einbindung von Audio- und Videoelementen. Kritisch ist anzumerken, dass die Funktionalität in vielen Fällen nur zur Erstellung datenintensiver, aber inhaltlich bedeutungsloser Intros auf Webseiten bzw. in Webanwendungen verwendet wird.

⁶³ Nach dem Kauf von Macromedia im Jahre 2005 gehört Adobe das proprietäre Format Flash.

⁶⁴ Vgl. Radtke, A. / Charlier, M. (2006), S. 203 ff.

⁶⁵ Vgl. <http://joomla.org/>.

⁶⁶ Vgl. <http://typo3.org/>.

4 Ausblick auf zukünftige Entwicklungen

Mit einer Zunahme von Breitbandanschlüssen und der Verbreitung von Webstandards haben Webanwendungen und -oberflächen deutlich an (Gestaltungs-)Möglichkeiten und Nutzungspotenzialen gewonnen. Dokumentiert wird dies nicht zuletzt durch eine immer höhere Präsenz von Web 2.0 - Applikationen und Services. Dennoch ist nach wie vor eine Reihe von Problemen auszumachen, zu deren Beseitigung zukünftige Anstrengungen zu erwarten sind. Es ist naturgemäß relativ schwierig, Prognosen für einen Bereich abzugeben, in dem die technologische Entwicklung vergleichsweise schnell voranschreitet, dennoch sollen an dieser Stelle einige ausgewählte Aspekte kurz angerissen werden.

Die Weiterentwicklung von Webstandards gestaltet sich oftmals langwierig. Bedingt durch die Zusammensetzung des W3C gibt es immer wieder Firmen, die versuchen, eigene Interessen zu fördern, was die Formulierung allgemein anerkannter Recommendations erschwert. So ist die Spezifikation zu CSS 3 bereits seit Ende des vergangenen Jahrtausends in Arbeit, ohne dass eine fertige Empfehlung in unmittelbarer Aussicht steht. Zwischenzeitlich wurde CSS 3 sogar in einzelne Module unterteilt, die getrennt voneinander veröffentlicht werden, um Entwicklungsergebnisse präsentieren zu können, die dann auch noch von den Browserherstellern implementiert werden müssen. Ähnlich zeitaufwändig gestaltet sich die Spezifikation von XHTML 2. Dazu kommt, dass auch aktuelle Browser bereits bestehende Standards noch immer nicht konsequent umsetzen. Insbesondere der Internet Explorer von Microsoft muss bei bestimmten Anforderungen an das Layout nach wie vor mit Hacks, die z.T. vor anderen Browsern versteckt werden müssen, „überredet“ werden, Inhalte wie gewünscht darzustellen, was ein standardkonformes Design erschwert. Hier ist eindeutig noch Nachholbedarf vorhanden. Bei der Barrierefreiheit ist zu hoffen, dass diese Thematik in naher Zukunft noch stärker in den Fokus der Applikationsentwickler und ihrer Auftraggeber rückt, insbesondere was die Einbindung derzeit problematischer Formate und Konzepte betrifft.

Auch auf anderer Ebene müssen technologische Voraussetzungen weiter ausgebaut werden. Durch die zunehmende Verbreitung von Audio/Video-Portalen wie YouTube, MySpace oder Flickr und den damit verbundenen Datentransfer wird die verfügbare Bandbreite zunehmend ausgereizt. Experten warnen bereits vor einem (partiellen) Netzwerk-Kollaps durch eine Überlastung der verfügbaren Datenleitungen. Betroffen sind auch drahtlose Netzwerke, die kurzfristig weiter an Bedeutung gewinnen und langfristig kabelgebundene Endanschlüsse vermutlich ersetzen werden.

Der Funktionsumfang von Webbrowsern wird sich in Zukunft weiter erhöhen. Wie sich die „Out Of The Box“ - Einbindung von Plugins, etwa für PDF-Dokumente oder Flash-Animationen entwickelt, wird nicht zuletzt von der Bereitschaft von Adobe abhängen, diese Plugins in Lizenz herauszugeben. Zumindest die Schnittstellen zum Processing von PDF-Dateien liegen inzwischen offen. Außerdem kann davon ausgegangen werden, dass derzeit nur als externe Software verfügbare XSLT-Prozessoren mit zunehmender Verbreitung von XML-Applikationen mittelfristig als fixe Bestandteile in Webbrowsern Einzug halten werden. Zu guter Letzt sei noch erwähnt, dass die Marktmacht von Suchmaschinen-Anbietern (insbesondere Google) zukünftig Gegenstand genauerer Untersuchungen sein wird. So muss herausgearbeitet werden, inwiefern diese Anbieter durch ihre (technische) Manipulierbarkeit und gestaffelte Zahlungsverpflichtungen zur Reihung in den Suchergebnissen, aber auch durch eigene Manipulationen von Rankings, mangelnde Auskünfte zum Datenschutz (oder Verstöße dagegen) und regionale Zensur von Inhalten ihre enorme Marktmacht unrechtmäßig ausnutzen. Diese Untersuchungen werden durch ein nach wie vor nicht existentes internationales Recht allerdings zusätzlich erschwert.

Literaturverzeichnis

Bücher

Bongers, F. (2005): XSLT 2.0 - Das umfassende Handbuch. 1., korrigierter Nachdruck. Galileo Press, Bonn.

Budd, A. / Moll, C. / Collison, S. (2007): CSS Mastery. Addison-Wesley, München.

Carl, D. (2006): Praxiswissen Ajax. O'Reilly, Köln.

Cederholm, D. (2006): Bulletproof Webdesign. Addison-Wesley, München.

Fischer, M (2006): Website Boosting. Suchmaschinen-Optimierung, Usability, Webseiten-Marketing. MITP, Heidelberg.

Jacobsen, J. (2005): Website-Konzeption. Erfolgreich Web- und Multimedia-Anwendungen entwickeln. 3., erweiterte Auflage. Addison-Wesley, München.

Meyer, E. (2005): Eric Meyer's CSS. Addison-Wesley, München.

Nidermeyer, S. (2006): Cocoon 2 und Tomcat. XML-Publishing mit dem Open-Source-Framework. Galileo Press, Bonn.

Niedermeier, S. / Scholz, M. (2006): Java und XML. Grundlagen, Einsatz, Referenz. Galileo Press, Bonn.

Radtke, A. / Charlier, M. (2006): Barrierefreies Webdesign. Attraktive Websites zugänglich gestalten. Addison-Wesley, München.

Shafer, D. / Yank, K. (2004): Cascading Stylesheets. dpunkt.verlag, Heidelberg.

Shea, D. / Holzschlag, M. E. (2005): Zen und die Kunst des CSS-Designs. Addison-Wesley, München.

Steyer, R. (2006): AJAX mit PHP. Beschleunigte Webapplikationen für das Web 2. Addison-Wesley, München.

Ullenboom, C. (2006): Java ist auch eine Insel. Das umfassende Handbuch. 5., aktualisierte und erweiterte Auflage. Galileo Press, Bonn.

Young, M. J. (2001): XML - Schritt für Schritt. 2. Auflage. Microsoft Press Deutschland, Unterschleißheim.

Zeldman, J. (2006): Webdesign mit Webstandards. Addison-Wesley, München.

Web-Adressen

Beckett, D. / McBride, B. (2004): RDF/XML Syntax Specification (Revised). <http://www.w3.org/TR/rdf-syntax-grammar/>. 26.04.2007.

Berners-Lee, T. (2006): Notation3 (N3) - A Readable RDF Syntax. <http://www.w3.org/DesignIssues/Notation3.html>. 26.04.2007.

Bos, B. et al. (1998): Cascading Style Sheets, Level 2. <http://www.w3.org/TR/REC-CSS2/>. 25.04.2007.

Bray, T. et al. (2006): Extensible Markup Language (XML) 1.0 (Fourth Edition). <http://www.w3.org/TR/REC-xml/>. 25.04.2007.

Chisholm, W. / Vanderheiden, G. / Jacobs, I. (1999): Web Content Accessibility Guidelines 1.0. <http://www.w3.org/TR/WAI-WEBCONTENT/>. 27.04.2007.

Parada, R. A. (2006): DOAC - Description Of A Career. <http://ramonantonio.net/doac/>. 26.04.2007.

Prud'hommeaux, E. / Seaborne, A. (2007): SPARQL Query Language For RDF. <http://www.w3.org/TR/rdf-sparql-query/>. 26.04.2007.

Shea, D. (2007): css Zen Garden: The Beauty In CSS Design. <http://www.csszengarden.com/>. 26.04.2007.

o.V. (o.J.): Atom. <http://www.atompub.org/>. 26.04.2007.

o.V. (2007): Barrierefreies Internet - Wikipedia. http://de.wikipedia.org/wiki/Barrierefreies_Internet. 29.04.2007.

o.V. (o.J.): BITV - nichtamtliches Inhaltsverzeichnis. <http://www.gesetze-im-internet.de/bitv/index.html>. 27.04.2007.

o.V. (2007): Google Suggest. <http://www.google.com/webhp?complete=1&hl=en>. 25.04.2007.

o.V. (2006): Illinois Center For Information Technology Accessibility: Firefox Accessibility Extension. <http://firefox.cita.uiuc.edu/index.php>. 29.04.2007.

- o.V. (2007): Joomla! <http://joomla.org/>. 29.04.2007.
- o.V. (2007): Netvibes. <http://www.netvibes.com/>. 25.04.2007.
- o.V. (2004): Resource Description Framework (RDF) / W3C Semantic Web Activity. <http://www.w3.org/RDF/>. 26.04.2007.
- o.V. (2006): RSS 2.0 Specification (Version 2.0.8). <http://www.rssboard.org/rss-specification>. 26.04.2007.
- o.V. (2007): RSS - Wikipedia. <http://de.wikipedia.org/wiki/Rss>. 26.04.2007.
- o.V. (2006): SemanticGov :: Services For Public Administration. <http://www.semantic-gov.org/>. 26.04.2007.
- o.V. (2007): The Apache Cocoon Project. <http://cocoon.apache.org/>. 25.04.2007.
- o.V. (o.J.): The Friend Of A Friend (FOAF) Project. <http://www.foaf-project.org/>. 26.04.2007.
- o.V. (2007): Typo3.org: Typo3 Content Management System - Developer Resource. <http://typo3.org/>. 29.04.2007.
- o.V. (2006): Web Accessibility Initiative (WAI). <http://www.w3.org/WAI/>. 27.04.2007.
- o.V. (2003): Welcome To The HiSoftware Cynthia Says Portal. <http://contentquality.com/>. 29.04.2007.
- o.V. (2007): World Wide Web Consortium. <http://www.w3.org/>. 25.04.2007.
- o.V. (2007): W3C Members. <http://www.w3.org/Consortium/Member/List.php3>. 25.04.2007.
- o.V. (2002): XHTML 1.0: The Extensible Hypertext Markup Language (Second Edition). <http://www.w3.org/TR/xhtml1/>. 25.04.2007.