

Kapitel 5

Browser

So wie eine Uhr die Zeit anzeigt, ein Telefon es bei anderen Telefonen klingeln lässt und Ihre Anlage alle CDs abspielt, sollte ein Browser alle Webseiten korrekt darstellen.

[5]

Die unrühmlichen Browserkriege haben ihren Tribut gefordert. Denn als ob es nicht genügen würde, dass der Anwender zwischen mehreren Produkten auswählen kann, steht es ihm auch noch frei, eine ältere Version einer modernen Software einzusetzen. Erschwerend kommt der Umstand hinzu, dass kein einziger Browser auf dem Markt als fehlerfrei bezeichnet werden kann. Kein Wunder dass sich die Webdesigner ständig die gleichen Fragen stellen müssen: Muss ich mehrere Versionen meiner Seite entwerfen? Wie kann ich mir diesen Mehraufwand leisten? Bin ich dazu verpflichtet, meine zahllosen wunderbaren Ideen zugunsten des kleinsten gemeinsamen Nenners aufzugeben? Dieses Kapitel legt dar, warum die einzelnen Browser so grundverschieden sind, wie Webdesigner Strategien entwickeln können, um diese Klippen zu umschiffen, und welche Basistechnologien man nutzen kann, um die mangelnde Unterstützung von Standardtechnologien auszugleichen. Zum Abschluss begeben wir uns noch auf die Suche nach Lösungsansätzen und erfahren darüber hinaus, was die Browserhersteller über dieses Thema denken.

Ist es nicht verblüffend, dass das Internet funktioniert? Wir sitzen vor unseren Computern, browsen durch das World Wide Web, verschicken E-Mails, steuern FTP-Server an und machen uns keinerlei Gedanken darüber, wie das funktioniert. Kaum jemand interessiert sich für das Zusammenspiel der

Protokolle, die das Rückgrat des weltweiten Netzwerks darstellen.

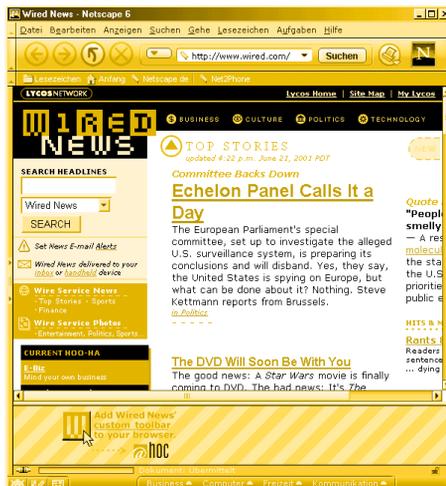
Andererseits können wir aber froh sein, dass wir diese komplexen Zusammenhänge nur am Rande streifen und uns stattdessen nur mit einfach gestrickten Benutzeroberflächen, Fenstern und Assistenten abgeben müssen. Diese Vereinfachung führt dazu, dass kaum ein Anwender darüber nachdenkt, dass das Internet im Grunde genommen nichts weiter als eine Ansammlung von Standards ist, auf die man sich einst geeinigt hat. Und ohne diese Standards gäbe es kein Internet.

Da stellt sich die Frage, warum wir trotz dieser scheinbar unverrückbaren Regeln Probleme mit dem Internet haben. Eine sehr gute Frage, die übrigens seit der Einführung des ersten Browsers gestellt wird und immer noch unbeantwortet ist.

In letzter Zeit wurde vermehrt Wert auf die zugrunde liegenden Standards – oder besser gesagt, auf den Mangel an Unterstützung vorgegebener Standards – gelegt, die zum Erfolg des Internets beigetragen haben. Entwickler haben sich schon immer über die Inkompatibilitäten verschiedener Browser (und sogar unterschiedlicher Versionen eines Herstellers) aufgeregt: »Das kann doch nicht so schwer sein!« und »Warum muss ich zusätzliche Zeit investieren, um eine Seite an alle Browser anzupassen?« gehören noch zu den harmloseren Einwänden.



Browser im Wandel: Die Wired-Homepage im Netscape Navigator 2.01 aus dem Jahre 1996 ...



...und der aktuellen Version Netscape 6. Die Unterschiede sind nicht zu übersehen.

Diese auf Produktpräsentationen und Entwicklerkonferenzen immer wieder auftauchenden Fragen wurden scheinbar nicht ernst genommen. Die beiden großen Browser-Hersteller Netscape und Microsoft entfachten einen derartigen Hype um ihre Produkte, das einem unbedarften Beobachter Angst und Bange wurde: »Neue Features! Neue Technologien! Neue Erweiterungen des W3C-Standards!«, tönte es aus den PR- und Marketing-Abteilungen der Softwareproduzenten. Die Antworten der Webdesigner ließen nicht lange auf sich warten, immer in dem Tenor: »Steckt voller Fehler! Ungenügende Implementierung der Standards! Zu viele proprietäre Erweiterungen!«

Lassen Sie uns einen Blick in die Vergangenheit werfen, um diesem Problem auf die Spur zu kommen und ganz nebenbei seine Auswirkungen auf die täglichen Arbeiten eines Webdesigners aufzuzeigen.

Innovationen und Hemmschuhe

Interoperabilität ist untrennbar mit der Geschichte des Internets verbunden. Seitdem Computer miteinander kommunizieren können, streiten sich Ingenieure und Entwickler darüber, wie diese Kommunikation durchgeführt werden soll.

Dieser Disput lässt sich ganz einfach veranschaulichen: Es könnte ein Problem aufkommen (»Wir müssen den Standard so ausrichten, dass E-Mails auf allen Systemen gelesen werden können«), jemand macht einen Vorschlag zu dessen Lösung (»Der Header einer E-Mail muss so und so formatiert werden«) und eine dritte Person kümmert sich dann um die notwendige Programmierung. Die in diesen Vorgang eingebundenen Entwickler überprüfen den Quellcode, diskutieren über die Spezifikationen und überarbeiten den Lösungsansatz so lange, bis sie mit dem Ergebnis zufrieden sind. Anschließend muss eine unterbezahlte Kraft das Ganze in Worte fassen und der staunenden Öffentlichkeit vorstellen. Letzteres wird dadurch erreicht, dass das Dokument als RFC (Request for Comments) auf einen öffentlich zugänglichen Server gestellt wird, um allen Interessierten den Zugriff zu ermöglichen.

Doch dann kam das Web und alles war auf einmal anders. Denn die explosionsartige Zunahme der Webanwender erschwerte nicht nur die Standardisierung, sondern stellte alle bisher gültigen Regeln auf den Kopf. Unternehmen, die Web-Software produzierten, mussten ihre Produkte so schnell wie möglich auf den Markt bringen, um die wachsende Zielgrup-

pe zu versorgen und ganz nebenbei den Aktienkurs zu steigern. An eventuell bestehende Standards oder gar den Sinn proprietärer Erweiterungen wurde kein Gedanke verschwendet.

Die Auswirkungen dieser Politik sind kaum zu übersehen. Schlecht implementierte »Erweiterungen« bereits bestehender Standards bringen Webdesigner auch heute noch um den Schlaf. Beispiel Netscape: Um den Ruf der Anwender nach mehr gestalterischen Möglichkeiten zu befriedigen, führte das Unternehmen das schon mehrfach erwähnte ``-Tag ein. Und obwohl bereits zu diesem Zeitpunkt über die Leistungsfähigkeit von Cascading StyleSheets (CSS) diskutiert wurde, piff Netscape auf bestehende Standards und entschied sich für die einfachere Lösung. Natürlich profitierten auch die Entwickler davon, urplötzlich Texte in sieben verschiedenen Größen darzustellen, doch der ursprüngliche Sinn von Überschriften, Aufzählungen und Zwischentiteln blieb auf der Strecke.

Weitere Beispiele gefällig? Denken Sie an die unzähligen Möglichkeiten, mit denen sich multimediale Elemente in Webseiten integrieren lassen. Neben dem inzwischen zum Standard ausgerufenen `<OBJECT>`-Tag, gibt es noch die Befehle `<EMBED>`, `<INSERT>`, `<APP>`, `<APPLET>` und ``, die alle dasselbe machen.

Diese immer kürzer werdenden Innovationszyklen sind aber beileibe nichts Neues. Schon in der Anfangszeit des Internets überboten sich die Programmierer mit immer neuen Ideen, die gedankenlos in die Produkte implementiert wurden. Beispielsweise war der Versionsprung zum Netscape Navigator 2.0 gewaltig, da die Entwickler neue Features wie das Font-Tag und Frames einbauten und damit neue Standards schufen. Und als dann auch noch Microsoft mit dem Internet Explorer im Geschäft mitmischte, war das Rennen eröffnet. Einziges Ziel dieses inzwischen schon legendären Konkurrenzkampfs: Jede neue Version eines Browsers sollte das Produkt des Mitbewerbers überbieten. Und so zeichnete sich jede Version durch eine Vielzahl von Innovationen aus, die aber alles andere als standardisiert waren. Den beiden Unternehmen ging es nur darum, der Welt zu beweisen, dass der eigene Browser das Nonplusultra darstellt.

Wo liegt das Problem? Da Webdesigner, Anwender und Marketing Manager grundverschiedene Vorstellungen vom Nonplusultra haben, deckten sich die Erwartungen nicht immer. Erschwerend kam an dieser Stelle der Umstand hinzu,

dass die Softwarehersteller auch anderweitige Wünsche und Bedürfnisse befriedigen mussten.

Die Browserhersteller erzählten immer wieder, dass ihre Produkte auch auf Content-Provider und Softwareentwickler zugeschnitten sind. Das beweise, so wollten sie uns Glauben machen, dass es sich um ein hochwertiges Produkt handle, welches die ambitionierteste Arbeit der Entwickler unterstütze. Im Klartext bedeutet diese Aussage: »Nachdem wir in einer früheren Version einige unausgegrenzte Sachen integriert haben, müssen wir sie auch zukünftig mit uns herumschleppen.«

Noch schlimmer: Anstatt sich um die Fehlerbehebung zu kümmern, statteten die Hersteller ihre Produkte mit immer neuen Features und Funktionen aus. Handelte es sich zu Beginn noch um vernünftige Erweiterungen wie beispielsweise E-Mail- und News-Client, packten die Entwickler alsbald HTML-Editor und Personal Information Manager dazu. Das Ergebnis sind die heute üblichen, mehrere Megabyte großen Browser-Monster, die an den – ohnehin stets zu knappen – Ressourcen des Rechners nagen.

Und nachdem alle Webentwickler unter der etwas eigenwilligen Definition von »Unterstützung« seitens der Browserhersteller zu leiden hatten, stellen die Designer die alles entscheidende Frage: »Was sollen wir eigentlich unterstützen?«

Vielleicht sollten wir aufgeben

Tatsache ist, dass ein Seitenlayout von jedem Besucher anders gesehen wird. Kein unbedingt erbaulicher Gedanke, dass ein mit viel Kreativität umgesetzter Webauftritt auf den Bildschirmen der Betrachter anders aussieht als ursprünglich geplant. Doch was können wir dagegen unternehmen?

Zunächst einmal muss ich erwähnen, dass Sie nicht allein sind. So veröffentlichte Jupiter Communications (www.jup.com) vor kurzem eine Studie, die sich mit den Eigenschaften der 100 größten kommerziellen Sites beschäftigte. Dabei kamen ein paar unerfreuliche Dinge heraus: Nahezu zwei Drittel dieser Sites gibt es in zwei verschiedenen Versionen (für die beiden Browser Netscape Navigator und Microsoft Internet Explorer). Für diese doppelte Arbeit setzen die befragten Unternehmen knapp 40% der Entwicklungsressourcen ein. Und nur die wenigsten Firmen planen in nächster Zeit den Einsatz »fortgeschrittener Browsertechnologien«.

Von einer anderen Warte aus betrachtet, sieht die Sache noch unerfreulicher aus. Wie viel Zeit und Geld muss ein Unternehmen aufwenden, um die von den Browser-Herstellern nachlässig durchgeführte Implementierung klar definierter Standards durch programmiertechnische Tricks auszugleichen? Wie sehr wird der Fortschritt dadurch verlangsamt, dass Unternehmen auf den Einsatz neuer Technologien verzichten und sich stattdessen auf den kleinsten gemeinsamen Nenner beschränken?

Darüber zu klagen, ist ganz einfach – und das werden wir in diesem Kapitel des Öfteren tun, glauben Sie mir. Als weitaus schwieriger entpuppt es sich, diesen Missstand zu beheben. Werfen wir also einen Blick darauf, wie wir mit den Schwächen der Browser, der ungenügenden Unterstützung bestehender Standards und den Tücken der verschiedenen Surf-Hilfen umgehen können.

Browserschwächen verstehen

Will man sich mit der verwirrenden Welt der Browser auseinander setzen, muss man die Probleme, die hinter jeder Ecke lauern, kennen. Zu Beginn betrachten wir die komplette Branche, um uns dann Schritt für Schritt an Ihre Webseite heranzutasten. Sie müssen sich durch eine Menge von Informationen wühlen, um in puncto Inhalt und Zielgruppe die richtigen Entscheidungen treffen zu können; daher gehen wir von allgemeinen Trends zu speziellen Aufgaben über.

Doch zuvor müssen noch folgende Fragen beantwortet werden:

- Welche Browser gibt es?
- Welcher Browser unterstützt welche Verfahren und Tags?
- Welcher Anteil meiner Zielgruppe nutzt welchen Browser?

Sobald Sie im Besitz dieser Informationen sind, können Sie sich an die Entwicklung einer Strategie wagen. Im Folgenden weise ich Sie auf Möglichkeiten hin, die Ihnen zwei Aufgaben erleichtern sollen: Welche Features sollen in die Webseite eingebaut werden, und wie passe ich die jeweiligen Funktionen an die diversen Browsertypen an?

Warum gibt es so viele Browser?

Wer bereits mit dem Entwurf einer komplexen Webseite in Berührung gekommen ist, wünscht sich nichts sehnlicher als ein Browsermonopol. »Wie schön wäre es doch, wenn ich meine Arbeit nur an einem Produkt ausrichten müsste.«

Vorsicht! Dieser Gedanke ist falsch und wäre der zukünftigen Entwicklung des Webs nicht eben förderlich.

Die große Beliebtheit des Webs rührt auch aus seiner Vielfalt. Ein Grundprinzip des Webs ist, dass man unabhängig von eingesetztem Computer, Betriebssystem und Browser Teil des Ganzen sein kann. Um dieses Ziel zu erreichen, setzte das ursprüngliche Internet auf die folgenden Komponenten: eine Methode, um alle Internet-Server und deren Inhalte eindeutig zu identifizieren, einen Übertragungsstandard und einen Weg, um die Inhalte auf dem Bildschirm des Betrachters anzuzeigen.

Den ersten Stützpfiler stellt der so genannte Uniform Resource Locator, kurz URL, dar. Ohne diese Angaben müssten sich Websurfer nicht nur Domain-Namen und IP-Adressen merken, sondern müssten auch die auf den Servern abgelegten Inhalte kennen. Heutzutage machen die Browser den schnellen und einfachen Zugriff auf alle Inhalte möglich und lassen die komplizierte – früher aber unumgängliche – befehlzeilenorientierte Kommunikation per Telnet vergessen.

Mit der zweiten grundlegenden Technologie wollen wir uns nicht lange aufhalten, da die Vorteile des Hypertext Transfer Protocols, kurz http, auf der Hand liegen. Der, abgesehen von der einfachen Navigation, mit Abstand größte Gewinn besteht darin, dass alle Browser und Webserver die »gleiche Sprache sprechen«.

Sorgen bereitet uns lediglich die dritte Grundvoraussetzung: eine standardisierte Methode zur Darstellung von Webseiten.

Bei der Definition der Hypertext Markup Language (HTML) standen die drei Faktoren Einfachheit, Erweiterbarkeit und Unabhängigkeit von Computern und Betriebssystemen im Vordergrund. Und Letzteres umfasste nicht nur PCs und Macs, sondern wirklich alle auf einem Mikrochip aufbauenden Geräte wie beispielsweise Mainframe-Terminals, hochmoderne Multimedia-Maschinen und sogar Mobiltelefone.

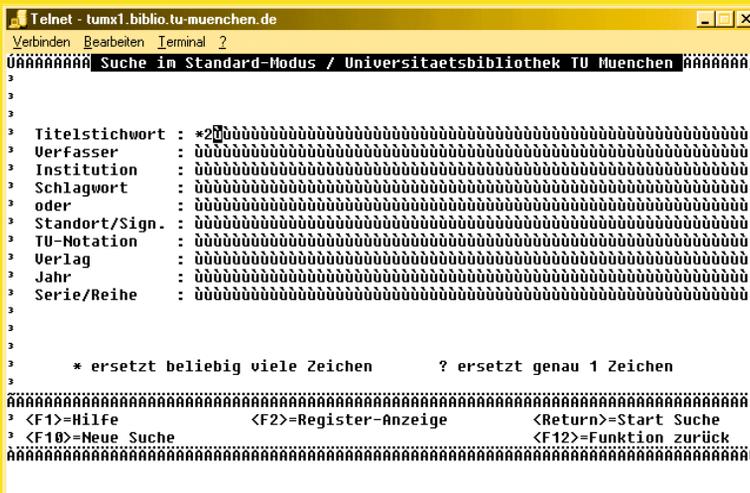
Telnet oder die Zeit vor den Browsern

Für uns ist es inzwischen zur Normalität geworden, alle Informationen per Mausklick zu erhalten und mittels Hyperlinks von einer Webseite auf eine andere geleitet zu werden. Die wenigsten Websurfer führen sich vor Augen, dass das World Wide Web noch gar nicht so alt ist. In jenen Zeiten aber, als Suchmaschinen ebenso wenig bekannt waren wie Bookmarks und die permanenten Klick-mich-Aufforderungen, war das Internet ein echtes Freak-Medium.

Zum einfachen Austausch von Dateien musste man die IP-Adresse der Gegenstelle in einen FTP-Client eintippen; wer auf der Suche nach Textinformationen war, kam nicht um den Einsatz einer Telnet-Applikation herum: Zuerst wurde die IP-Adresse des anderen Rechners eingetippt. Nach dem Login erschien die Unix-Oberfläche des fremden Rechners auf Ihrem Bildschirm, und Sie konnten mit der Recherche beginnen. Am einfachsten können Sie sich dieses Prinzip als eine

Art Fernsteuerung vorstellen, wie sie heute noch von Programmen wie beispielsweise *LapLink* oder *PCAnywhere* angeboten werden. Und auch wenn heute nahezu alle Telnet-Dienste der bunten WWW-Oberfläche zum Opfer gefallen sind, hält ein aufrechtes Häuflein von Unix-Anwendern die Telnet-Fahne hoch. Der einstige Status zeigt sich allerdings auch noch daran, dass in einem Betriebssystem wie Windows 98 nach wie vor ein DOS-basierender Telnet-Client integriert ist. Um diesen zu aktivieren, öffnen Sie eine DOS-Box und geben am Prompt einfach *telnet* ein.

Ein gutes Beispiel für die ursprüngliche Arbeitsweise des Internets stellt der Bibliotheksserver der Technischen Universität München dar (tumx1.biblio.tu-muenchen.de). Obwohl die Benutzerführung nach heutigen Maßstäben alles andere als einfach ist, gibt es an der Qualität der Suchfunktion nichts auszusetzen.



So sah das Internet vor der Entwicklung des World Wide Webs aus.

Was aber in der Theorie ein hehres Ziel ist, wird in der Praxis zu einem nicht zu unterschätzenden Problem. Web-Browser gewannen so schnell an Popularität, dass die Hersteller unzählige Neuerungen einbauen mussten, um die Wünsche der Anwender zu erfüllen. Dies führte dazu, dass der Markt binnen kürzester Zeit mit Hunderten von Browsern überschwemmt wurde. Und obwohl diese Vielfalt eigentlich dem ursprünglichen HTML-Grundprinzip der Unabhängigkeit entspricht, führte sie auch dazu, dass ein und dieselbe Webseite von fast jedem Browser anders dargestellt wurde. Und genau diese Inkonsistenz will ein Webdesigner um jeden Preis vermeiden. Somit müssen sich Designer mit einer Vielzahl von Varianten auseinandersetzen und sich bereits im Vorfeld überlegen, auf welcher Grundlage eine Webseite entstehen soll. Viele wählen deshalb den Weg des geringsten Widerstands und verzichten auf innovative Elemente, um ja keinen Browser zu vernachlässigen. Es liegt auf der Hand, dass das nicht im Sinne des Erfinders ist. Doch auch hier gibt es Lösungen.

Erkennen des Problems

Wie bereits mehrfach erwähnt, gibt es eine schier unüberschaubare Auswahl an Browsern, Betriebssystemen und anderen Variablen, die oft dazu führen, dass eine im Grunde perfekt gestaltete Seite falsch dargestellt wird. Zählt man noch den Umstand hinzu, dass kein Browser frei von Bugs ist, kann es durchaus passieren, dass das Projekt eines Webdesigners unausgegoren wirkt. Wie kann man dem entgegenwirken?

Der eine Weg besteht darin, seinen Kopf in den Sand zu stecken und Seiten zu entwerfen, die grundsätzlich nur die gängigsten Tags verwenden. Wer sich also mit `<H1>` und `<P>` zufrieden gibt, ist fein raus und erspart sich Ärger und Überstunden. Die Seite erinnert vom Design her zwar eine Studentenarbeit der frühen Neunziger des vergangenen Jahrhunderts, dafür gibt es keinerlei Darstellungsprobleme und auch die Ladezeiten sind perfekt.

Ich bin der Auffassung, dass ein Webdesigner seine Vorstellungen verwirklichen können muss, wenn er eine optisch und technisch hochwertige Seite mit einem ganz eigenen Charakter erschaffen will. Dieses Ziel ist aber nur dann zu erreichen, wenn man gewillt ist, bestimmte Risiken auf sich zu nehmen und sich mit der Vielzahl von Technologien und Tags auseinanderzusetzen. Leider bedeutet das ein

nicht unerhebliches Maß an zusätzlicher Arbeit, da Sie aus Kompatibilitätsgründen mehrere Versionen einer Seite gestalten müssen.

Nur gut, das dieses Ziel weitaus einfacher zu erreichen ist, als es auf den ersten Blick erscheint. Schließlich haben sich schon viele Leute mit diesem Thema befasst und eine Reihe von Werkzeugen entwickelt, die diese Aufgabe wesentlich vereinfachen. Doch genug der Vorrede, wenden wir uns endlich dem professionellen Seitendesign zu.

Unser Interesse gilt zunächst der Branche. Zunächst werfen wir einen Blick auf die Geschichte, um zu erfahren, wie sich diese auf die heutige Browserstatistik auswirkt. Dadurch sollten wir in der Lage sein abzuschätzen, wie viele Personen mit welcher Browserversion im Netz unterwegs sind. Noch wichtiger sind natürlich fundierte Aussagen darüber, mit welchem Browser Ihre Seite angesteuert wird. Um diese Information zu erhalten, führe ich Sie in die Welt der Server-Logs ein. Schlussendlich befassen wir uns mit dem Design multipler Benutzeroberflächen, die exakt auf die Fähigkeiten der wichtigsten Browser zugeschnitten sind. In diesem Zusammenhang treffen wir auch auf eine Reihe wichtiger Technologien, die zum Grundwissen aller ambitionierten Webdesigner gehören sollten.

Was will die Branche?

Während der Arbeit an diesem Buch fanden sich unter der Yahoo-Kategorie »Computers & Internet > Software > Internet > World Wide Web > Browsers« unglaubliche 248 Einträge. Und obwohl es sich dabei nicht nur um Links auf Software handelt, spiegelt dies die enorme Vielfalt wider, mit der sich ein Webdesigner auseinander setzen muss. Dazu kommen die einzelnen Versionen, die insbesondere das Netscape-Angebot nahezu unüberschaubar machen. So wurde der Netscape Navigator sowohl als 16bit- als auch 32bit-Applikation angeboten; auf Seiten der Betriebssysteme unterstützt der Browser Windows, Mac OS und diverse Unix-Derivate. Jede dieser Versionen wurde im Lauf der Jahre überarbeitet und unter einer neuen Versionsnummer bzw. einer alternativen Produktbezeichnung (angefangen bei den Betas bis hin zu Standard- und Komplettlösungen wie beispielsweise die Gold-Variante) auf den Markt gebracht. Diese Auswahl beschränkt sich aber nicht nur auf Produkte für die »traditionellen« Computer. So gibt es inzwischen jede

Menge Browser für die sehr beliebten PDAs wie Palm oder Pision, spezielle Varianten für blinde Internetnutzer sowie abgespeckte Derivate, die beispielsweise an speziellen Terminals zum Einsatz kommen. Nicht zu vergessen diejenigen Produkte, die den heimischen Fernseher internetfähig machen wollen. Sie sehen, je intensiver man die Suche nach Browservarianten betreibt, desto mehr Produkte tauchen auf.

Glücklicherweise können wir die Anzahl der tatsächlich wichtigen Produkte auf eine Hand voll begrenzen. Unsere Basisstrategie, auf die ich an späterer Stelle detailliert eingehen werde, lehnt sich an das bekannte Motto »teile und herrsche« an. Zunächst stellen wir fest, wie viele unterschiedliche Versionen einer Seite programmiert werden müssen, um diese in einem nächsten Schritt an die jeweiligen Browsergruppen anzupassen. Die einfachste Form dieser Strategie besteht darin, jeweils eine High- und Lowend-Variante zu gestalten und anschließend die gängigen Browser in diese beiden Gruppen einzuteilen. Dabei muss das Seitendesign natürlich exakt auf diese beiden Optionen abgestimmt werden.

Wenden wir uns nun wieder der ursprünglichen Frage zu, welche Browser am häufigsten verwendet werden. Hier dürfte es keine große Überraschung darstellen, dass ein Großteil der Anwender auf einige wenige Produkte zurückgreift. Die Browsernutzung folgt der Zipfschen Kurve, mit deren Hilfe man Zusammenhänge zwischen zwei breit gefächerten Komponenten (auf der einen Seite die vielen Browser, auf der anderen Millionen von Anwendern) darstellen kann.

Doch was bedeutet das für Ihre Webseite? Sehr viel, da die gängigen Browserstatistiken auch exotische Plattformen wie den seligen Amiga sowie steinalte Beta-Versionen berücksichtigen, obwohl solche Produkte einen verschwindend geringen Anteil besitzen. Bedeutet das, dass man sie guten Gewissens vernachlässigen kann? Nein, da professionelle Seiten von *allen* Browsern dargestellt werden müssen. Dazu später mehr.

The screenshot shows the 'Stats Station' page of BrowserWatch. It features a navigation menu with links like 'Monthly News', 'News Archives', 'Articles', 'Browser Profiles', 'Chat', 'Plugins', and 'Stats Station'. A sign-up form for a newsletter is also present. The main content area is titled 'BrowserWatch Stats Station' and includes a notice about computer platform data. Below this is a table titled 'Browser Types Visiting BrowserWatch' with columns for Browser, Value, and Percent.

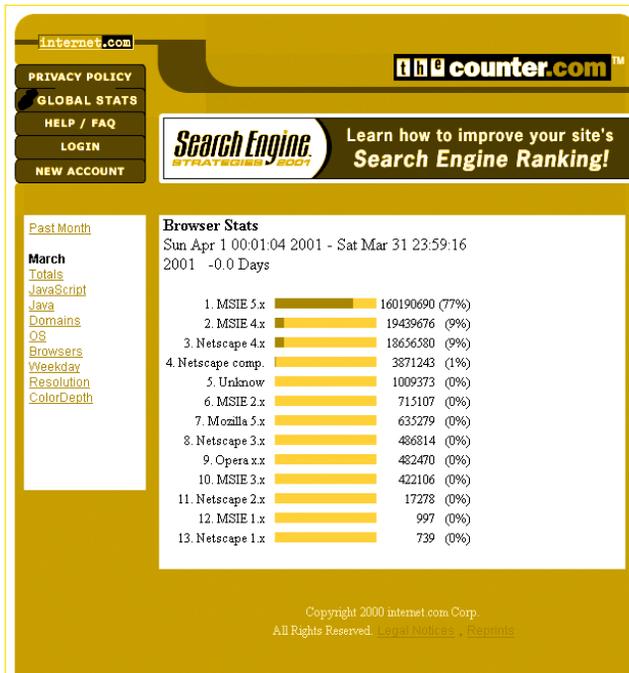
Browser	Value	Percent
Microsoft Internet Explorer	65622	63.1%
Netscape Navigator	25606	24.6%
Opera	4383	4.21%
AvantGo 3.2 (compatible, AvantGo 3.2)	1066	1.02%
Powermarks-3.5 (compatible, Powermarks)	721	0.69%
Konqueror-2.1.1 (compatible, Konqueror)	720	0.69%
Konqueror-2.1.2 (compatible, Konqueror)	681	0.65%
Lynx	448	0.43%
IBrowse	406	0.39%

Dave Garaffas Webseite BrowserWatch ist die beste Anlaufstelle, um sich über die Verbreitung aktueller Browser zu informieren.

Welche Browser werden denn nun am häufigsten verwendet? Falls Sie auf Netscape Navigator und Microsoft Internet Explorer tippen, liegen Sie richtig. Um einen Blick auf die Beliebtheitsskala vom Juni 2001 zu ergattern, besuchen wir die ausgezeichnete Seite BrowserWatch (www.browserwatch.com). Diesen Angaben zufolge liegt der Internet Explorer mit 63,1 % unangefochten an der Spitze, an zweiter Stelle steht der Netscape Navigator mit 24,6 %. Somit nutzen knapp 88 % aller BrowserWatch-Besucher einen der beiden kommerziellen Browser, die übrigen 12 % surfen mit anderen Produkten durch das Web. Allerdings dürfen Sie nicht vergessen – und der Betreiber der Seite weist noch einmal explizit darauf hin –, dass diese Site hauptsächlich von Leuten aus dem IT-Bereich besucht wird. Und diese, auch als Early Technology Adaptors bezeichneten Surfer sind dafür bekannt, stets die aktuellste Hard- und natürlich auch Software einzusetzen. Insofern unterscheiden sie sich vom typischen Heimanwender.

Um noch detailliertere Informationen zum Thema Nutzungshäufigkeit der Browser zu erhalten, müssen Sie nur The Counter (www.thecounter.com) besuchen. Diese Seite bietet kostenlose Zähler für private Homepages an. Solche Elemente sind bei Hobby-Designern überaus beliebt, da sie eine Seite aufwerten und allen Besuchern anzeigen, wie viele Personen sich bereits auf die Homepage verirrt haben. Doch erst eine Zusatzleistung macht The Counter zur beliebten Anlaufstelle für professionelle Designer: Denn jede der mit einem kostenlosen Zähler ausgestatteten Seite liefert Daten über die von den Besuchern genutzten Browsertypen. Und da The Counter mehrere Millionen privater Homepages versorgt, sind diese Daten sehr aussagekräftig.

Nichtsdestoweniger beweisen auch diese Zahlen die Vorherrschaft des Internet Explorers. Summiert man die Häufigkeit der einzelnen Versionen auf, ergibt sich folgendes Bild: 86 % (und das sind fast 25 % mehr Nutzer als bei BrowserWatch) surfen mit einem Microsoft-Produkt durch die Weiten des World Wide Web. Erfahrungswerte belegen, dass diese Angaben ein realistischeres Bild der tatsächlichen Verteilungshäufigkeit zeichnen. Ich weise Sie deswegen auf diese Diskrepanz hin, damit Sie folgende Regel verinnerlichen: Ihre Seite kann sich von den üblichen Websites unterscheiden – und auch Ihre Klientel.



The Counter informiert Sie sogar über die Versionsnummer der eingesetzten Browser.

Nun sind wir endlich in der Lage, fundierte Aussagen über die Beliebtheit der Browser zu machen. Allerdings stellen diese Zahlen nur die halbe Wahrheit dar, da wir auch an künftige Versionen und die Häufigkeit der Aktualisierung denken müssen. Genau diese ungewissen Faktoren sind es, die einem Webdesigner zu schaffen machen. Es ist frustrierend, neue Technologien nicht gleich einsetzen zu können. Warum? Ganz einfach. Es macht ist schließlich wenig sinnvoll, eine perfekt gestylte und mit allen technischen Wassern gewaschene Seite auf die Beine zu stellen, um dann feststellen zu müssen, dass die Seite nur von einem geringen Prozentsatz der Besucher dargestellt werden kann. Und wieder einmal stellt sich die Frage: »Was kann man dagegen unternehmen?«

Alte Browser sterben nicht aus

In den Anfangszeiten des World Wide Web (also im Herbst 1994) griffen die wenigen Online-Nutzer bevorzugt zum Browser NSCA Mosaic, einer vom National Center for Super

Computing Applications (Universität von Illinois, Champaign-Urbana) entwickelten Software. Mit dem Erscheinen des Netscape Navigators änderten sich die Nutzervorlieben drastisch und der neue Browser wurde auf immer mehr Rechnern installiert. Es hatte den Anschein, dass fast alle Online-Nutzer auf die Qualitäten des Newcomers setzten.

Doch bereits in den Kindertagen des Webs mussten die Anwender in regelmäßigen Abständen große Softwarepakete aus dem Internet herunterladen, da Netscape immer wieder überarbeitete Versionen veröffentlichte. Knapp ein Jahr später etablierte sich Microsoft mit dem Internet Explorer im Kreis der Browserhersteller. Einstieg und folgender Konkurrenzkampf mündeten im so genannten »Browserkrieg«, mit dem beide Unternehmen die Vorherrschaft auf dem Markt erringen wollten.

Um sich ein Bild über die Akzeptanz der einzelnen Browser zu machen, müssen wir einen Blick auf die letzten fünf Jahre werfen. Das nachfolgende Diagramm zeigt den Verlauf innerhalb von fünf Jahren (1995 bis 2000) und betrifft ausschließlich die Browserversionen. Dabei ist zu beachten, dass sich die angegebenen Versionsnummern auf die unterstützten Funktionen beziehen. So entsprechen beispielsweise die Features des zweiten Navigators denen des ersten Internet Explorers; der Funktionsumfang von Netscape Navigator 4 kann hingegen mit dem des Internet Explorers 4 verglichen werden. Und auch die Tatsache, dass Microsoft niemals einen Internet Explorer 1.0 auf den Markt brachte, spiegelt sich in dieser Aufstellung wider.

Wie das Diagramm beweist, stieg die Beliebtheit des ersten Netscape-Browsers fast über Nacht auf knapp 80% an. Die nachfolgende Version hatte es weitaus schwerer und konnte den Grad der Verbreitung nur mühsam steigern. Als Faustregel lässt sich feststellen, dass eine neue Version 16 Monate benötigte, um den Vorgänger vom Markt zu verdrängen. Doch leider hat auch diese Regel ihre Ausnahmen.

Nicht jeder Anwender kann sich für jeden beliebigen Browser entscheiden. Denn während die meisten Leser dieses Buchs wahrscheinlich im Wochentakt neue Programme installieren, gibt es viele Online-Nutzer, die nicht über so große Freiheiten verfügen. An erster Stelle stehen hier natürlich diejenigen Surfer, die vom Arbeitsplatz aus auf das Internet zugreifen. Mit welchen Programmen das erfolgt, hängt in den meisten Fällen von den Vorgaben des Systemadministrators ab. Denn in großen, vernetzten Unternehmen

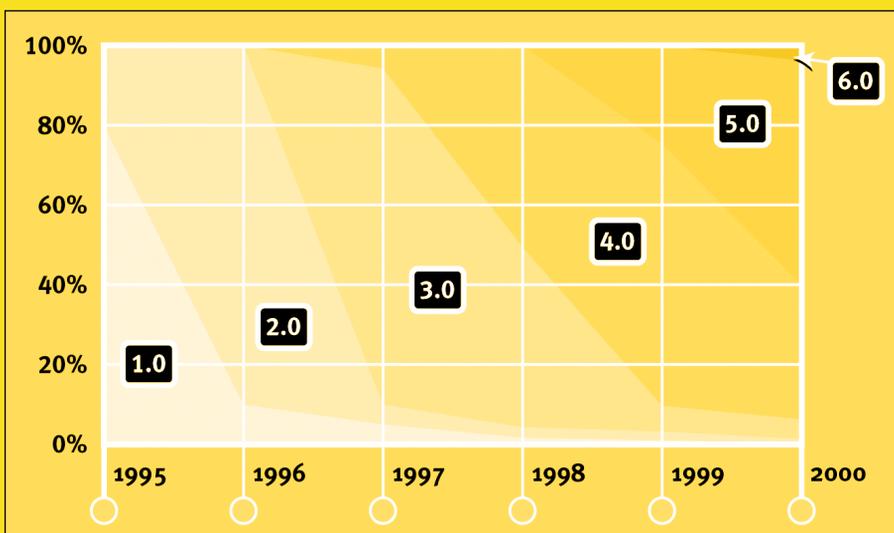
haben nur diese Leute das Recht, neue Software zu installieren. Und wenn diese Hüter der Netzressourcen der Meinung sind, dass die jeweils aktuellste Browsergeneration noch nicht ausgereift sei (Stichwort Sicherheitslücken), bewegen sich Tausende von Angestellten eben mit einer älteren Browserversion durch das Netz.

Viele Fachleute führen dieses vorsichtige Vorgehen auf das Release der vierten Browsergeneration zurück, als Netscape Navigator und Microsoft Internet Explorer um die Vorherrschaft kämpften. Die Hersteller blähten ihre Produkte auf und die Browser mutierten von einfachen Surfhilfen zu stark in die Ressourcen des Betriebssystems eingreifenden Applikationen. Somit wurde das Upgrade eines Browsers zu einer waghalsigen Angelegenheit. Angesichts dessen darf es nicht weiter verwun-

Browserupgrades im Lauf der Jahre

Sobald eine neue Browsergeneration angeboten wird, rüsten die Anwender binnen kürzester Zeit auf. Wie dieses Diagramm aber zeigt, gilt dies nicht zwangsläufig für alle Internet-User. Die vertikale Achse stellt den Prozentsatz der Anwender dar, die einen Browser nutzen; die horizontale Achse markiert den Zeitraum von 1995 bis 2000, und

die im Diagramm untergebrachten Zahlen stehen stellvertretend für die jeweilige Browsergeneration. Wie sich unschwer erkennen lässt, greifen nicht alle Anwender stets zum aktuellsten Produkt. Daraus ergibt sich ein Versionswirrwarr, das die Arbeit eines Webdesigners unnötig erschwert.



dern, dass sich die um Systemintegrität bemühten Unternehmen weitere Upgrades verkniffen.

Aber auch die Privatanwender, also alle Leute, die täglich nur wenige Minuten vor dem Rechner sitzen, hielten sich zurück. Denn gerade bei diesen ungeübten Nutzern gilt nach wie vor die Regel: »Never change a running system.« Einzig der Kauf eines neuen Systems (mit vorinstallierter Software) machte diesen Personenkreis mit neuen Browserversionen bekannt. Die dritte Gruppe derjenigen, die nach wie vor einen älteren Browser benutzen, setzt sich aus Besitzern älterer Rechner zusammen. Denn wer beispielsweise immer noch mit einem 386er mit 8 Megabyte RAM arbeitet, kann gar keinen modernen Browser verwenden. Die Folge: Alte Browsergenerationen sterben nicht aus.

Das lässt sich ganz klar aus dem Diagramm erkennen. Denn obwohl jede neue Version von Power-Usern und Technologiefreaks sofort eingesetzt wird, bleibt ein riesiger Brocken scheinbar veralteter Varianten bestehen. Und je schneller neue Browserversionen auf den Markt geworfen werden, desto komplizierter wird das Ganze, da sich aktuelle Varianten mit älteren Versionen vermengen.

Trotzdem ist ein wenig Optimismus angesagt. Schließlich hat uns die Statistik von The Counter (www.thecounter.com) darüber aufgeklärt, dass die Mehrheit der Anwender mindestens einen Browser der vierten Generation einsetzt. Und diese hohe Durchdringung führt wiederum dazu, dass diese Versionsnummern als Fixpunkte verwenden können.

Wie sehen die Zahlen bei Ihnen aus?

Sehr gut. Nun sind wir in der glücklichen Lage, einen ungefähren Eindruck über die globale Verteilung der einzelnen Browserversionen zu haben. Doch wie sieht es eigentlich mit Ihrer Seite aus? Wissen Sie darüber Bescheid, welche Browser die Besucher einsetzen? Falls nicht, sollten Sie sich diese Daten umgehend besorgen. Es ist überhaupt nicht kompliziert.

Der Softwaremarkt bietet eine schier unübersehbare Auswahl an Programmen, die Ihnen bei dieser Aufgabe hilfreich zur Seite stehen. Dabei setzen alle Tools auf das gleiche Prinzip: Sobald ein Besucher Ihre Seite ansteuert, fordert der Browser Daten (Text, Bilder und verlinkte Objekte) an. Der Server erledigt diese Datenübertragung und führt

gleichzeitig Buch über jeden einzelnen Vorgang. Dieses als Server-Log bezeichnete Protokoll liefert ausgezeichnete Hinweise auf die Aktivitäten einer Webseite. In Kombination mit den bereits erwähnten anderen Werkzeugen stellen diese Logs so etwas wie ein Informations-Füllhorn dar, das dem Betreiber einer Webseite exakte Benutzerstatistiken liefert.

Da wir uns in diesem Kapitel aber nur für die eingesetzten Browserversionen interessieren, konzentrieren wir uns auch ausschließlich darauf. In jeder Zeile des Server-Logs finden sich Informationen wie die Angabe, welche Datei angefordert wurde, Zeit und Datum des Requests und natürlich die exakte Kennung des eingesetzten Browsers (Fachausdruck »User Agent String«).

»User Agent« steht dabei stellvertretend für die eingesetzte Software, die Informationen von einem Server abrufen; »String« bezeichnet die charakteristischen Angaben, die währenddessen im Server-Log hinterlassen werden. Und da die meisten Surfer mir einem der beiden großen Browser unterwegs sind, finden sich entsprechend viele Einträge im Server-Protokoll. Die wenigen Ausnahmen stammen von Suchmaschinenrobotern, Proxy Servern und Programmen, die ganze Webseiten herunterladen. Ein typischer User Agent String sieht folgendermaßen aus:

```
Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
```

Die Bedeutung dieser auf den ersten Blick kryptisch anmutenden Zeile ist überaus einfach: Der Ausdruck »Mozilla« ist ein Fantasienamen, der auf den internen Codenamen des ersten Netscape-Browsers zurückgeht. Und da Webmaster in jenen Tagen ganz begeistert von den Fähigkeiten dieses neuen Browsers waren, richteten sie ihre Seiten auf diese Software aus. Mit dem Auftauchen alternativer Browser (an erster Stelle dem Microsoft Internet Explorer), die die von Netscape vorgegebenen Funktionen ebenfalls unterstützten, hatten die Webmaster allerdings nicht gerechnet. Um die Seiten aber trotz der Ausrichtung auf »Mozilla« korrekt anzuzeigen, segelten die Alternativ-Browser fortan unter falscher Flagge und gaben sich am Server ebenfalls als »Mozilla« aus. Über die Jahre hinweg wurde diese Beflagung zu einem Quasi-Standard, dem sich kein neuer Browser entziehen konnte. Der große Nachteil: Wenn sich jeder Browser als »Mozilla« ausgibt, kann der Webmaster keinerlei

Aussagen über das tatsächlich eingesetzte Produkt machen. Um dieses Problem zu lösen, einigte man sich darauf, den Browser durch eine zusätzliche Angabe eindeutig zu identifizieren. Im obigen Beispiel handelt es sich um einen Microsoft Internet Explorer (MSIE) 4.01, der auf einem Rechner mit dem Betriebssystem Windows 98 läuft.

Wie bereits erwähnt, schlummern diese Informationen in Ihrem Server-Log und warten nur darauf, von Ihnen ausgewertet zu werden. Die meisten Analysewerkzeuge geben Ihnen eine Vielzahl von Statistiken in die Hand, die nahezu beliebig konfiguriert werden können. So ist es beispielsweise auch möglich, nur die Browserversionen auszulesen.

Zahlenspiele

Bei der Gewinnung dieser Informationen stehen Ihnen – abhängig davon, ob Sie Zugriff auf die Server-Logs haben – mehrere Wege offen. Sind Sie Webdesigner oder mit der Verwaltung des Servers betraut, stellt es kein großes Problem dar, an die Serverprotokolle heranzukommen. Haben Sie hingegen keinen Zugriff auf die Serverstruktur, sei es weil Sie dazu nicht befugt sind oder Ihre Seite von einem externen Dienstleister gehostet wird, sieht die Sache ein wenig komplizierter aus.

Zunächst einmal geht es um diejenigen, die in der glücklichen Lage sind, über uneingeschränkte Zugriffsrechte zu verfügen. Diese Gruppe kommt in den Genuss leistungsfähiger Analysen, die nahezu alle Aspekte der Besucherstatistik umfassen. Die erste Entscheidung besteht dann darin, festzulegen, ob ein Analysewerkzeug direkt auf dem Server ausgeführt wird oder ob die Server-Logs erst auf den Desktop-Computer übertragen und anschließend ausgewertet werden. Produkte wie Hit List von Accrue (www.accrue.com) oder die zahlreichen Werkzeuge von WebTrends (www.webtrends.com) arbeiten serverseitig, werden also direkt auf dem Webserver installiert und ausgeführt. Mit diesen Programmen ist es möglich, automatische Protokolle zu generieren, um eine Aufstellung aller verwendeten Browser zu erhalten. Diese lassen sich anschließend automatisieren und so konfigurieren, dass Ihnen beispielsweise jeden Montag die aktuelle Statistik per E-Mail zugesandt wird.

Server-Logs mit Analog auslesen

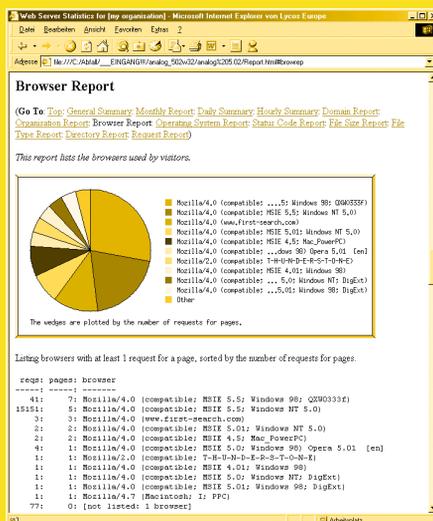
Auch wenn das Programm ein wenig veraltet wirkt, mit dem kostenlosen Analog lassen sich Server-Logs bequem auslesen.

Auf der Betreiber-Homepage (www.analog.cx, alternativ unter www.statslab.cam.ac.uk/~sret1/analog) laden Sie sich die aktuelle Version 5.02 herunter. Nach der Installation müssen Sie noch einige Modifikationen durchführen, um das Programm an Ihre Bedürfnisse anzupassen. Zunächst einmal müssen Sie die Konfigurationsdatei `analog.cfg` in einem Texteditor wie dem Windows-Notepad öffnen.

Die Menge der angebotenen Möglichkeiten ist überwältigend, da wir uns aber primär für Angaben über die verwendeten Browser interessieren,

beschränken wir uns auf diese Statistik. Dazu müssen Sie dem Programm mitteilen, wo die Server-Logs gespeichert sind (Zeile `LOGFILE`) und unter welchem Dateinamen das Ergebnis (Zeile `OUTFILE`) abgelegt werden soll. Dabei können sowohl lokale Pfade als auch Server-Ressourcen angegeben werden. Im zweiten Schritt geht es um die Browserberichte, die durch den manuell einzufügenden Parameter `BROWSER` bzw. `FULLBROWSER ON` aktiviert werden. Anschließend starten Sie das Programm und genießen die durch Grafiken ergänzten Statistiken.

Sehr lobenswert: Der Bericht wird als HTML-Datei abgelegt, die Navigativon stellt somit kein großes Problem dar. Ein Klick auf den »Hyperlink Browser Report« führt direkt zum entsprechenden Kapitel, im dem Sie exakte Angaben über alle verwendeten Browser finden.



Alles im Blick: Detaillierte Informationen über jeden bei einem Besuch Ihrer Seite eingesetzten Browser.

The screenshot shows the configuration file `analog.cfg` in a text editor. The file contains various configuration options for the web server statistics program. The options are organized into sections, with lines starting with `#` representing comments. The configuration includes settings for the log file, output file, browser types, and search engines.

```
# Configuration file for analog 5.02
# See http://www.analog.cx/

# Here are a couple of configuration commands to get you started. Add any more
# you like from the README.

# Lines starting with # are comments.
# There is a much more extensive configuration file in examples/big.cfg
# If you need a LOGFORMAT command it must go here, above the LOGFILE commands.
LOGFILE logfile.log
# LOGFILE /old/logs/access_log.*
OUTFILE Report.html
HOSTNAME "my organisation"
# REFLINKINCLUDE pages
# REFLINKINCLUDE pages
# REFLINKINCLUDE *
# REFLINKINCLUDE *
# REFLINKINCLUDE *
# UNCOMPRESS =.gz.*.Z "gzip -cd"
BROWOUTPUTALIAS Mozilla Netscape
BROWOUTPUTALIAS Mozilla (compatible) "Netscape (compatible)"
BROWOUTPUTALIAS IEM6 AOL
SUBBROW /*
SUBTYPE =.gz.*.Z
# Add whichever of these types of pages you have on your server, or others.
# PAGEINCLUDE *.html
# PAGEINCLUDE *.asp
# PAGEINCLUDE *.jsp
# PAGEINCLUDE *.cfm
# PAGEINCLUDE *.pl
# PAGEINCLUDE *.php
SEARCHENGINE http://altavista.*/* q
SEARCHENGINE http://yahoo.*/* p
SEARCHENGINE http://google.*/* q
SEARCHENGINE http://lycos.*/* query
SEARCHENGINE http://aol.*/* query
SEARCHENGINE http://excite.*/* search
SEARCHENGINE http://go2net.*/* general
```

Die Konfiguration des Analysewerkzeugs muss in Handarbeit durchgeführt werden.

Wer keinen Zugriff auf seinen Webserver hat, kann auf ein Tool wie Analog (www.analog.cx) oder The Webalizer (www.mrunix.net/webalizer) zurückgreifen. Dabei handelt es sich um einfache Software, die Sie auf Ihrem lokalen Rechner installieren. Anschließend laden Sie die Server-Logs herunter und die Analyse kann beginnen.

Wer keinerlei Zugriffe auf seine Server-Logs hat, muss nicht gleich aufgeben. Denn es gibt noch einen weiteren Weg, um an die begehrten Statistiken heranzukommen – auch wenn Sie dabei auf weiterführende Funktionen wie automatische Listenerzeugung und Grafiken verzichten müssen. Diese Methode führt über die zahlreichen kostenlosen Webdienste die sich darauf spezialisiert haben, beliebige Internetseiten zu beobachten. Im Gegenzug verpflichten Sie sich dazu eine Art Zugriffszähler oder ein Banner auf Ihrer Seite zu platzieren. Nach diesem Eingriff werden bei jedem Seitenaufruf Datenpakete an den Statistikdienst übertragen, der diese Informationen für Sie verwaltet. Anschließend besuchen Sie die Betreiberseite und werfen einen Blick auf das Profil Ihrer Besucher.

Die Nachteile dieser Methode liegen auf der Hand. So müssen Sie zunächst einmal das Logo des Dienstleisters in Ihr Seitenlayout integrieren. Dieses aufgezwungene Element stört nicht nur das Layout, es verlangsamt auch den Seitenaufbau (da ja bei jedem Aufruf auch eine Verbindung zum Server des Betreibers aufgebaut wird). Und vergessen Sie auch nicht den Datenschutz, schließlich kennt das Unternehmen die Server-Logs ebenso gut wie Sie. Angesichts der Tatsache, dass dieser Service kostenlos zur Verfügung gestellt wird und für viele Webdesigner die einzige Möglichkeit zur Auswertung der Log-Dateien darstellt, sind diese negativen Aspekte aber zu verschmerzen. Wir haben ja bereits einen Blick auf das Angebot TheCounter.com geworfen. Dieses Unternehmen bezieht seine Zahlen genau daher. Einen ähnlichen Service bietet auch Hitbox (www.hitbox.com) an. Falls Sie sich selbst auf die Suche nach solchen Angeboten begeben wollen, müssen Sie in der Suchmaschine Ihres Vertrauens einfach nur »Web counter« eingeben.

Nutzen Sie die Zahlen

Sobald Sie über die Zahlen der eingesetzten Browser informiert sind, sollten Sie diese auch gewinnbringend einsetzen. Bevor Sie aber loslegen, sollten Sie sich vor Augen halten, was

solche Statistiken eigentlich ausdrücken. Möglichweise werden Sie darüber erstaunt sein, dass sich Ihre ganz persönliche Browserstatistik grundlegend von den im Web üblichen Angaben unterscheidet. Ich habe diese Frage bereits an dem Beispiel BrowserWatch abgehandelt. Diese Seite zielt auf fortgeschrittene Anwender, Webdesigner und IT-Spezialisten, deren Nutzerprofil und damit auch die eingesetzte Software sich stark vom Durchschnittsanwender unterscheiden. Ihre Seite zieht möglicherweise eine ähnlich homogene Nutzergruppe an, wie ein Blick auf die Statistik belegen könnte.

Das führt zu einer weiteren wichtigen Erkenntnis: Ihr Seitendesign fungiert als eine Art Filter. Es mag selbstverständlich erscheinen, dass der Einsatz fortschrittlicher Technologien einen gewissen Benutzerkreis ausschließt, nur vergisst man das leicht. Setzen Sie beispielsweise auf Funktionen, die nur von aktuelleren Browserversionen unterstützt werden (denken Sie an die dynamische Seitenpositionierung mit Hilfe von JavaScript und Cascading StyleSheets), ohne sich Gedanken um eine Alternativversion der Seite zu machen, wird Ihre persönliche Browserstatistik in Richtung Highend-User tendieren. Aus diesem Grund ist es unumgänglich, zwei unterschiedliche Server-Log-Statistiken zu generieren: Eine Auswertung deckt die Anwender pro Browser ab, die andere umfasst die Page Views im Verhältnis zur eingesetzten Software. Die Unterschiede können gravierend ausfallen. Eine auf Highend-Browser ausgerichtete Seite könnte bis zu zehn Mal mehr Power-User anziehen, obwohl die Anzahl der Besucher gleich bleibt.

Diese Informationen ergeben in Verbindung mit den zuvor beschriebenen Statistiken ein recht gutes Bild von der Besucherstruktur einer Seite. Und nun stellt sich die große Frage: Welche browserspezifischen Tags und Funktionen wollen Sie unterstützen? Um die richtige Entscheidung zu treffen, sollten Sie selbstverständlich genau darüber Bescheid wissen, welche Funktionen von einer bestimmten Browserversion unterstützt werden. Diese müssen dann auf die Besucherstatistiken umgerechnet werden, um eine maximale Schnittmenge zu erhalten. Erst wenn Sie diese Überlegung angestellt haben, können Sie damit beginnen, spezielle Strategien für die anvisierten Zielgruppen zu entwerfen.

Obwohl die Seitenbeschreibungssprache HTML eigentlich einen Standard darstellt, wurde sie im Zuge der zunehmenden Verbreitung des Web immer wieder verändert und erweitert. Einige Browser unterstützen alle Befehle der vom W3C

überwachten Spezifikationen, andere setzen hingegen auf die Kraft proprietärer Erweiterungen. Als direkte Auswirkung des Browserrennens setzen die beiden populären Vertreter auf unterschiedliche Strategien. Diese Politik schlägt sich in einem undurchschaubaren Versions-Dickicht nieder. So muss ein Browser mit der Versionsnummer 3.0 nicht zwangsläufig mehr Features unterstützen als ein 2.0-Produkt.

Gut, dass es hier Hilfestellungen in Form informativer Webangebote gibt. Eine der beliebtesten Anlaufstellen stellt Index Dot HTML (www.blooberry.com/indexdot/) dar. Abgesehen von der ein wenig gewöhnungsbedürftigen Oberfläche ist dies die ideale Informationsquelle, um sich mit den Feinheiten der gängigen Browser vertraut zu machen. Dabei gehen die Autoren auf alle Tags und Attribute ein und stellen die Veränderungen der letzten Jahre dar. Ebenso

Alternative Browser

Es ist allgemein bekannt, dass der einst so vielfältige Browsermarkt mittlerweile von zwei Produkten beherrscht wird: Entweder man setzt auf die Kraft des stets ein wenig fortschrittlicheren Netscape oder man nutzt den Marktführer Internet Explorer. Dazwischen gibt es nichts. Stimmt das überhaupt? Nein, denn nach wie vor gibt es jede Menge Alternativen. Das mit Abstand bekannteste Produkt kommt aus Norwegen und nennt sich Opera (www.opera.com). Das schlanke Programm wird seit einigen Monaten kostenlos vertrieben, was der Verbreitung sicherlich sehr zuträglich ist. Noch häufiger wird aber der zum Lieferumfang von AOL gehörende Browser eingesetzt. Und obwohl es sich dabei im Grunde genommen um eine spezielle Version des Internet Explorers handelt, werden nicht alle Tags unterstützt. Und auch die wachsende Gruppe der TV-Surfer darf nicht unterschätzt werden.

Gut, dass all diese speziellen Browser mit einer detaillierten Dokumentation ausgeliefert werden, die alle Einschränkungen auflistet. So bietet zum Beispiel AOL eine speziell auf die Bedürfnisse von Webdesignern zugeschnittene Infoseite (webmaster.info.aol.com) an.

AOL Browser Features and Functionality Table

Browser Version	MSIE 2.1	MSIE 4.01	MSIE 5.0	MSIE 5.01	MSIE 5.02	MSIE 6.0	MSIE 6.01	MSIE 5.0	MSIE 5.01	MSIE 5.5
AOL Versions Possible	3	4, 5	3, 4, 5, 6	3, 4	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6
OS Platform Req. or Usage	Mac	Mac	Win32	Win16	Win32	Win32	Win32	Win32	Win32	Win32
HTML	0.1	2.0	0.8	0.1	0.2	0.1	6.6	63.5	8.1	8.6
HTML specification	3.2	4	3.2	3.2	4	4	4	4	4	4
Supported by all:	Tables, Frames, Client-side ImageMaps, Borderless Frames, Background color in table cell									
Background image in table cell	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Floating frames	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Cascading style sheets*	N	Y	N	N	Y	Y	Y	Y	Y	Y
VHTML	N	Y	N	N	Y	Y	Y	Y	Y	Y
DHTML	Supported in MSIE 4.0+ . Exact capabilities documented at: http://msdn.microsoft.com/workshop/frames/html?workshop/author/Default.asp									
XML	Supported in MSIE 4.0+ . Exact capabilities documented at: http://msdn.microsoft.com/xml/Default.asp									
Graphics	Supported by all: GIF, Animated GIF, Transparent GIF, JPG, Progressive JPG, BMP, ART, J-G 3.0									
J-C 4.0 format (Slideshow)	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Sound										

Welche AOL-Version unterstützt die von Ihnen bevorzugten Technologien? Dieser spezielle Service verrät es.

werden Fragen wie »Wann wurde ein Tag vom W3C verabschiedet?« und »Welcher Browser unterstützte als Erster diese oder jene Funktion?« abgehandelt. Keine Wünsche bleiben offen. So werden beispielsweise nicht nur alle Browser aufgelistet, die Frames unterstützen; es ist auch möglich, nach einzelnen Attributen wie »Frameborder« oder »Bordercolor« zu filtern. Ideal, um sich bereits im Vorfeld über Stärken und Schwächen der diversen Browservarianten zu informieren.

Da aber Index Dot HTML nur Browser, Tags, Attribute und CSS-Definitionen abdeckt, sollten Sie eine zweite Informationsquelle hinzuziehen: Webmonkey's BrowserKit (www.webmonkey.com/browserkit/). Hier erwarten Sie auch Infos dazu, wie populäre Browser mit JavaScript, Cascading StyleSheets und XML umgehen. Die Informationen beschränken sich nicht nur auf Betriebssysteme wie Mac OS, Windows

Die Fülle der Informationen umfasst alles Wissenswerte rund um die Surf-Hilfe des Online-Dienstes. Darüber hinaus erwarten Sie zahlreiche Grundlagenartikel, die Sie darüber aufklären, wie AOL-Grafiken komprimiert werden und wie das Cache-System funktioniert. Und selbst wenn Sie nicht auf AOL-Nutzer abzielen, lohnt sich ein Besuch immer.

Falls Sie sich Gedanken darüber machen, ob Ihre Seite auch von TV-Surfern betrachtet werden kann, müssen Sie die WebTV-Homepage (developer.webtv.net) ansteuern. Das ist vor allem vorausschauenden Entwicklern zu empfehlen, da die momentan noch verschwindend geringe Zahl von WebTV-Surfern in den nächsten Jahren sprunghaft ansteigen wird. Wie schon das AOL-Angebot begeistert auch dieser Service durch die Vielfalt der Informationen.

Das absolute Highlight steht Ihnen in Form eines Emulators (WebTV

Viewer) zur Verfügung. Nach der Installation dieses Tools können Sie beliebige Webseiten auf ihre WebTV-Kompatibilität überprüfen.

The screenshot shows the website developer.webtv.net with the following structure:

- Header: **webtv** logo, [designing for television](#), [authoring for webtv](#), [interactive television](#), [developer forums](#)
- Navigation: [designing for television](#), [authoring for webtv](#), [interactive television](#), [developer forums](#)
- Featured Content:
 - Secure Site Handling and Gaming Update** (with a star icon)
 - designing for television**: Learn how to make your Web sites look great on television.
 - [WebTV Viewer](#)
 - [Ads for WebTV](#)
 - [More on design...](#)
 - authoring for webtv**: How the WebTV® browser supports different Internet technologies.
 - [Better Flash Support](#)
 - [Does WebTV Support...?](#)
 - [More on authoring...](#)
 - interactive television**: Discover how to create your own interactive programming.
 - [Interactive TV FAQ](#)
 - [Displaying TV in Web pages](#)
 - [More on interactive TV...](#)
 - developer forums**: Share your questions and thoughts with other developers.
 - [Interactive TV](#)
 - [Web Design](#)
 - [JavaScript](#)

Auf der Supportseite von WebTV erwarten Sie unzählige Hinweise zum richtigen Umgang mit der alternativen Plattform.

und Linux, sondern decken auch so exotische Einsatzgebiete wie NextStep, OS/2 und das Fernsehen (Stichwort WebTV) ab.

Anderen Technologien sind sogar komplette Webangebote gewidmet. So finden sich allein zum Thema Cascading StyleSheets mehrere Informationsseiten, auf denen erfahrene Designer über Stärken und Schwächen dieser dynamischen Komponente berichten. Eine der bekanntesten CSS-Quellen stellt WebReview (style.webreview.com) dar. Im Gegensatz zu vielen anderen Infoseiten beschränkt sich WebReview nicht nur auf die Vorstellung der Browserkompatibilität, sondern geht auch detailliert auf Probleme und bekannte Bugs ein. Und das Wissen um die Schwächen bestimmter Browservarianten ist für einen Designer ebenso wichtig wie eine Liste der unterstützten Funktionen.

Entscheiden Sie sich für eine Strategie

Inzwischen sind Sie mit den unterschiedlichen Analysewerkzeugen vertraut, wissen über die Verteilung der einzelnen Browserversionen Bescheid und sind sich dessen bewusst, dass eine Statistik unterschiedlich interpretiert werden kann. Also ist es an der Zeit, eine Strategie zu entwickeln, mit der sich eine Webseite exakt an Zielgruppen und Technologien anpassen lässt.

Und wie schon so oft, gibt es auch hier mehrere Ansätze, die sich folgendermaßen einteilen lassen:

- Was ich nicht weiß, macht mich nicht heiß
- Browserspezifischer Ansatz
- Dynamisches Prinzip

Bedenken Sie, dass sich diese drei Strategien überlappen.

Was ich nicht weiß, macht mich nicht heiß

Die einfachste Strategie ist im Grunde genommen gar keine. Nach dieser Methode können Sie eine sehr einfache Seite entwerfen, die in allen möglichen Browsern korrekt angezeigt wird. Die Betonung liegt dabei auf dem Wörtchen *können*.

So stellt es keine große Herausforderung dar, sich beim Seitendesign auf diejenigen Tags zu beschränken, die von allen Browservarianten und -generationen unterstützt werden.

Dazu zählt man alle grundlegenden HTML-Befehle, mit denen Tabellen, Frames, Fonts und das eine oder andere Applet definiert werden. Als Anhaltspunkt gelten die HTML-3.2-Spezifikationen. Solche Websites werden vor allem von den Designern bevorzugt, die weder über das Know-how noch über die nötigen Ressourcen verfügen, um ein technisch hochwertiges Angebot auf die Beine zu stellen. Dieser Kreis hat eigene Regeln definiert und gibt sich damit zufrieden, dass eine Seite von allen Besuchern betrachtet werden kann. Möglicherweise wird Sie die Feststellung verblüffen, dass ein Großteil der Webdesigner diesem Prinzip folgt.

Viele der zuvor vorgestellten Recherche-Tools helfen einem Designer bei der Anpassung dieser Strategie: Entwickler wollen sich nicht mit einem neuen – in den meisten Fällen unausgereiften – Code auseinander setzen oder sie möchten sich nicht mit der komplexen Materie der scriptgesteuerten Anpassung dynamischer Seiten befassen. Diese Gruppe bevorzugt den einfachen Weg und setzt nur diejenigen Tags und Technologien ein, die beispielsweise von Index Dot HTML als zuverlässig eingestuft werden. Nach diesem Schema aufgebaute Seiten werden mit Sicherheit niemals eine Auszeichnung bekommen, dafür korrespondieren sie aber mit nahezu allen möglichen (und auch unmöglichen) Browsern. Und diese uneingeschränkte Kompatibilität ist doch das alleinige Ziel, oder?

HTML wurde einst mit dem Hintergedanken entwickelt, abwärtskompatibel und konsistent zu sein. Im Klartext bedeutet dies, dass die Regeln einer neuen HTML-Version niemals bereits bestehende Grundsätze außer Kraft setzen könnten. Doch dieser im Grunde genommen rühmliche Ansatz sorgt in der Praxis dafür, dass ein Browser, der einen HTML-2.0-Befehl nicht umsetzen kann, auch mit allen späteren Versionen dieses Tags Probleme hat. Zwar können Bilder durch den Befehl `` in Seiten integriert werden, das eigentlich ebenso wichtige Attribut `>ALT=<` (Text, der den Inhalt des Bildes beschreibt) wird hingegen regelmäßig vergessen. Die HTML-Spezifikationen lassen also genügend Raum, um die hier angesprochene Strategie an die eigenen Wünsche und Fähigkeiten sowie an die verschiedenen Browserversionen anzupassen.

Werfen wir einen Blick auf eine beispielhafte Zeile Quelltext, die für die Einbindung mehrerer Objekte zuständig ist:

```
<object src="katze.mov">
  <img "katze.jpg" alt="Meine Hauskatze Fritz">
</object>
```

Die erste Zeile definiert ein Quick-Time-Filmchen, das eine Katze zeigt. Um aber auch Besitzer älterer Browser multimedial zu unterhalten, wurde zusätzlich ein JPG-Bild integriert. Unterstützt die Surfhilfe des Besuchers das Tag *<OBJECT>* nicht, kommt der Surfer zumindest in den Genuss des Bildes. Und selbst für den Fall, dass der Browser keine Bilder darstellen kann, ist vorgesorgt, da zumindest die Beschreibung »Meine Hauskatze Fritz« auf dem Bildschirm angezeigt wird. Dieser alternative Text stellt also die Abwärtskompatibilität sicher. Ganz zu schweigen von dem Umstand, dass der Text von Suchmaschinen indiziert und von speziell für Blinde entwickelten Browsern vorgelesen werden kann.

Browserspezifischer Ansatz

Auf der anderen Seite der Skala befinden sich diejenigen Designer, die gerne experimentieren. Diese auch als Early Technology Adopters bezeichneten Entwickler lieben es, die Browserfähigkeiten bis zum Maximum auszureizen und ständig neue Ansätze in das Webseitendesign einfließen zu lassen. Dieser Gruppe ist es egal, welchen Browser ein Besucher einsetzt – solange es sich um die jeweils aktuellste Version handelt. Die Seiten dieser Designer zeichnen sich vor allem dadurch aus, dass sie keinerlei Kompromisse eingehen und einen nicht unerheblichen Teil der Surfer ignorieren. Achten Sie einfach auf die Angabe »Optimiert auf« oder »Best when viewed with...« oder entsprechende grafische Elemente.

Über Vor- und Nachteile eines hochwertigen Seitendesigns müssen wir erst gar nicht sprechen, da solche Seiten als Vorreiter fungieren und Tausenden unentschlossenen Designern die Richtung vorgeben. Allerdings sollten Sie stets daran denken, dass die Entscheidung, einen Großteil der Surfer zu ignorieren, nicht nur vom Wunsch nach einem spektakulären Seitendesign abhängt, sondern auch mit der Zielgruppe verknüpft sein sollte. Denn während beispielsweise die Homepage einer Werbeagentur als Aushängeschild der

kreativen Köpfe erhalten muss, steht bei einem Online-Shop der Nutzwert im Vordergrund.

Dynamisches Prinzip

Wer sich weder mit der einen noch mit der anderen Strategie anfreunden kann, muss auf das dynamische Prinzip setzen.

Ganz gleich ob private Homepage oder professionelle Website – ein erfahrener Entwickler kann es sich nicht leisten, bestimmte Benutzergruppen auszuschließen. Denn wie wir wissen, ist es vielen Surfern gar nicht möglich, auf die jeweils aktuellste Browserversion aufzurüsten. Diesem Wissen steht der verständliche Wunsch gegenüber, seine gestalterischen Fähigkeiten in Form eines perfekten, technologisch ausgereiften Webdesigns zur Schau zu stellen. Der Ausweg aus diesem Dilemma: Sie müssen mehrere Versionen einer Seite entwerfen, die sich an die Funktionen der jeweiligen Browser anpassen. Diese Dynamik stammt entweder von einem auf dem Server untergebrachten Quellcode oder von den in den Seiten eingebundenen Scripts, die Betriebssystem und Browserversion des Besuchers in Erfahrung bringen und diesem eine exakt auf seine Bedürfnisse abgestimmte Version der Seite präsentieren.

Nachdem Webdesigner diesem Prinzip bereits seit Jahren folgen, ist es ein offenes Geheimnis, dass niemals alle theoretisch denkbaren Kombinationen aus Browser und Betriebssystem aufgefangen werden können. Wie schon an anderer Stelle erwähnt, produzieren die Designer deswegen jeweils eine High- und Lowend-Version und teilen die Browser anschließend in diese beiden Bereiche ein. Und auch wenn dieses Vorgehen nach wie vor aktuell ist, haben sich die Definitionen von High- und Lowend extrem verändert. So gestaltete ich vor Jahren beispielsweise Alternativseiten für diejenigen Browser, die keine Tabellen unterstützten. Würde ich diesem Gedanken auch heute folgen, würde die Browserverteilung so aussehen: 99,99 % Highend-Browser, der Rest Lowend. Allerdings dürfen Sie eine aktuelle Browserversion nicht unbesehen in die Highend-Rubrik einordnen. Beispielsweise sorgte die erschreckend schlechte CSS-Implementierung des Netscape Navigators 4.0 dafür, dass ich diesen Browser kurzerhand als Lowend-Surf-Hilfe einstuft und den Benutzern eine deutlich reduzierte Seite zuwies. Damit Sie diese Einteilung bereits im Vorfeld vornehmen können, habe ich Ihnen auf den vorangegangenen Seiten eine Auswahl von

Softwarehilfen und Webreferenzen vorgestellt, mit deren Hilfe sich Browserhäufigkeit und Umsetzung der HTML-Vorgaben in Erfahrung bringen lassen.

Selbstverständlich führt diese Mehrarbeit zu einem gewissen Maß an Frustration. Denn während sich das Web mit scheinbar unendlich hoher Geschwindigkeit weiterentwickelt, müssen Designer jede neue Technologie analysieren und darauf warten, dass der Großteil der Benutzer damit ausgestattet ist. Dazu gesellt sich die Uneinsichtigkeit der Browserhersteller, die auch in Zukunft ihr eigenes Süppchen kochen werden.

Da stellt sich die berechnete Frage, wie man das dynamische Prinzip am einfachsten umsetzt? Wie bereits kurz angerissen, gibt es zwei Ansatzpunkte: clientseitig (also am Browser) und serverseitig.

Browsercheck mit Macromedia Dreamweaver

Beim Einsatz eines HTML-Editors wie beispielsweise Macromedia Dreamweaver 4.0 reduziert sich der Arbeitsaufwand zur Umsetzung bestimmter Funktionen um einiges. Wollen Sie zum Beispiel den Quellcode um eine rudimentäre Browserweiche erweitern, erreichen Sie dies mit wenigen Mausklicks. Im Zusatzfenster »HTML-Stile« aktivieren Sie die Registerkarte »Verhalten«. Ein Klick auf das Plus-Symbol öffnet ein Pop-up-Menü, in dem Sie die Option »Browser überprüfen« wählen.

Im nächsten Dialogfeld können Sie dann festlegen, welchen Versionen des Netscape Navigators bzw. des Internet Explorers bestimmte Seiten zugewiesen werden. Andere HTML-Editoren wie Adobe GoLive 5 verfügen über ähnliche Funktionen.



Mit modernen HTML-Editoren wie zum Beispiel Macromedia Dreamweaver 4 lassen sich Browserweichen im Handumdrehen gestalten.

Clientseitiger Ansatzpunkt

Ganz gleich für welche Vorgehensweise Sie sich entscheiden, besteht der erste Schritt darin, den Quellcode zu modifizieren. Dieser muss den User Agent String auslesen, die Information weitergeben und zudem veranlassen, dass die exakt auf diesen Parameter abgestimmte Alternativseite übertragen wird. Darüber hinaus muss festgelegt werden, ob mögliche Scripts auf dem Server ausgeführt oder direkt im Browser gestartet werden. Beide Methoden haben ihre Vor- und Nachteile.

Die Übertragung von in HTML-Quellcode eingebundenem JavaScript, der als Browserweiche dienen soll, ist recht einfach. Das folgende Listing zeigt einen mit Macromedia Dreamweaver generierten Beispielcode, der festlegt, welche Codeabschnitte an welche Browser übertragen werden sollen.

```
<SCRIPT> <!--
// Windows IE 4 oder hoeher...
  if ((document.images) &&
      (navigator.appName != "Netscape") &&
      (navigator.appVersion.indexOf('Mac') == -1))
  {
    document.writeln("<link rel=\"stylesheet\"
      type=\"text/css\" href=\"ie.css\">")
  };

// Navigator 4...
  if ((navigator.appName == "Netscape") &&
      (parseInt(navigator.appVersion) == 4))
  {
    document.writeln("<link rel=\"stylesheet\"
      type=\"text/css\" href=\"nav.css\">")
  };
-->
</SCRIPT>
```

Der Code bewirkt nur, dass der vom Besucher verwendete Browser (`>>appName<<`) bzw. die Version (`>>appVersion<<`) abgefragt werden. Anschließend wird automatisch ein `<LINK>`-Tag hinzugefügt, dass auf das entsprechende StyleSheet verweist.

Allerdings bringt diese Abfrage nicht immer was. So deaktivieren beispielsweise viele User aus Sicherheitsgründen

die Scriptfähigkeiten des Browsers. Die Folge: Das gerade sorgsam ausgearbeitete JavaScript wird gar nicht oder nur unvollständig ausgeführt; im Browserfenster des Anwenders werden Fehlermeldungen angezeigt. Erschwerend kommt die Tatsache hinzu, dass diese Fehleranfälligkeit mit zunehmender Komplexität des Seitendesigns ansteigt.

Gehen wir noch einmal kurz auf die Grundregeln des clientseitigen Ansatzpunktes ein: Stellen Sie den Besuchern zwei Versionen Ihrer Seite zur Verfügung, und setzen Sie dabei auf eine JavaScript-Browserweiche, kann es vorkommen, dass die Browser einiger Anwender automatisch beide Versionen anfordern, obwohl nur eine angezeigt wird (nämlich diejenige, die in der Browserweiche definiert ist). Benutzt ein Besucher beispielsweise eine ältere Browserversion, wird die Software zunächst einmal die Startseite anfordern, um dann den Code auszuführen. Anschließend wird dieser Besucher auf die speziell auf seine Surf-Hilfe zugeschnittene Seite umgeleitet. Wie sich unschwer erkennen lässt, kann diese Taktik dazu führen, dass unnötig viele Daten ausgetauscht werden und sich der Seitenaufbau somit verzögert. Und auch das ist alles andere als benutzerfreundlich.

Aus diesem Grund setzen professionelle Webdesigner auf den serverseitigen Ansatz.

Serverseitiger Ansatz

Anstatt das Risiko einer langsamen Datenübertragung und falschen Seitendarstellung einzugehen, sollten wir uns mit der aktuellen Servertechnologie auseinandersetzen. Der Idealfall sähe folgendermaßen aus: Der Server erkennt den Browser und die betreffende Version und versorgt den Surfer automatisch mit der Seite, die auf seine Ausstattung optimiert ist. Dabei spielen mögliche Sicherheitsvorkehrungen wie beispielsweise eine deaktivierte Scriptunterstützung keine Rolle.

Derzeit gibt es eine ganze Reihe solcher serverbasierender Scriptsprachen. Die bekanntesten sind Microsoft Active Server Pages (.asp), Allaire's Cold Fusion, PHP und Embedded Perl. Allen Dialekten ist gemeinsam, dass sie Code auf dem Server ausführen, sobald ein Besucher eine Seite anfordert. Weitere Details lassen wir an dieser Stelle außen vor; mehr erwartet Sie in Kapitel 8.

Im Grunde genommen unterscheidet sich der Denkansatz der serverseitigen Programmierung kaum von seinem clientseitigen Pendant. In den meisten Fällen reichen ein paar

Zeilen aus, um an die Variable User Agent String zu kommen und entsprechende Aktionen zu definieren. Anschließend können Sie die einzelnen Seitenversionen mit Hilfe logischer Verknüpfungen so anordnen, dass jedem Browser seine ganz spezielle Seite zugewiesen wird.

Abgesehen davon, dass auf dem Server abgelegte Scripts problemlos ausgeführt werden, stehen Ihnen nun auch mehr Möglichkeiten offen, um den HTML-Quellcode zu variieren. Einige Beispiele für eine serverbasierende Browserweiche:

- **Mehrere Versionen einer Seite:** Sie können ein kleines Script generieren, das den Besucher automatisch auf die seiner Ausstattung entsprechende Seite umleitet. Die Programmierung dieses Scripts ist sehr einfach, Sie müssen aber mehrere Seitendesigns zur Verfügung stellen.
- **Eine Version, mehrere Weichen:** Eine sehr beliebte Methode, um eine Seite in mehreren Browserversionen darzustellen. Sie programmieren eine Seitenversion und erweitern den Quelltext um logische Abfragen (if...then), die alle kritischen Seitenelemente abfangen.
- **Gabelprinzip:** Die logische, weitaus komplexere Fortsetzung der zuvor erwähnten Methode, bei der die einzelnen Seitenelemente als isolierte Komponenten betrachtet werden. Dabei wird jedes Element in browserspezifischem Code programmiert, assembliert und anschließend an den Browser gesendet. Auf diese Art und Weise werden Inhalte von Präsentationselementen getrennt – die Skalierbarkeit ist weitaus höher. Der Nachteil dieser Methode: Die im Vorfeld zu erledigenden Planungen sind sehr zeitaufwendig.

Ist Besserung zu erwarten?

Falls Sie nun der Meinung sind, dass das alles nach enorm viel Arbeit riecht, muss ich Ihnen zustimmen. Die Entwickler und Designer setzen sich stundenlang mit der Überarbeitung und Verbesserung einer Seite auseinander und müssen im Extremfall das komplette Design auf den Kopf stellen und sich mit dem kleinsten gemeinsamen Nenner zufrieden geben. Ganz zu schweigen vom Verzicht auf neue Technologien. Gibt es denn keinen Ausweg? Kümmert sich kein Mensch um dieses Problem?

Doch, das World Wide Web Consortium (kurz W3C).

Werfen wir noch einmal einen Blick auf die Geschichte, die von dem Ideal geprägt ist, die Darstellung elektronischer Inhalte von der dazu benötigten Semantik abzunabeln. Auf den vorhergehenden Seiten habe ich Sie schon auf die Bedeutung dieses Gedankens hingewiesen.

Was hat das mit den altbekannten Standardisierungsproblemen zu tun? Vereinfacht gesagt gehen viele Webdesigner und -Entwickler von der These aus, dass ihre Inhalte konsistent dargestellt werden. Dem steht die Tatsache gegenüber, dass die Seitenbeschreibungssprache HTML unter der Prämisse entwickelt wurde, »jederzeit und überall« zu gleichen Ergebnissen zu führen – eine wichtige Einschränkung, die sich jeder ambitionierte Designer stets vor Augen halten muss. Denn bei der Entwicklung von HTML wurde Wert darauf gelegt, dass sich eine Seite an die jeweilige Ausstattung des Betrachters anpasst. Das führt dazu, dass beispielsweise eine mit `<H1>` definierte Überschrift von einem PC-gestützten Browser ganz anders dargestellt wird als von einem Palm.

So weit die Theorie. In der Realität sieht es jedoch so aus, dass die Mehrzahl der Surfer mit ähnlichen Computersystemen und Browserversionen im Web unterwegs sind. Die Folge ist ein De-facto-Standard, der die optische Interpretation der HTML-Befehle festlegt. Sobald wir einer Textstelle ein Tag zuweisen, erwarten wir, dass dieser Text stets gleich angezeigt wird, obwohl dies dem HTML-Grundgedanken widerspricht. Wenn ich eine Seite gestalte, möchte ich, dass der Zwischenraum zwischen den einzelnen Absätzen gleich bleibt – unabhängig vom eingesetzten Browser. Ist das nicht der Fall, sieht die Seite nicht gut aus. Und wer will das schon?

Als weitaus größeres Problem erweist sich aber der Umstand, dass dieser Quasi-Standard nicht skalierbar ist, da HTML ständig um neue Befehle und Attribute erweitert wird: Ein `<p>`-Tag wird innerhalb einer mit `` definierten Liste anders dargestellt; eine Tabellenzelle wird größer, wenn das abschließende Tag `</TD>` in einer anderen Zeile als der Inhalt steht. Somit ist es nahezu unmöglich, sich auf alle Eventualitäten einzustellen.

Die Browserlösung

Gesetzt den Fall, dass das W3C den perfekten Standard entwickelt, muss dieser auch von den Browserherstellern akzeptiert und in die Software implementiert werden. Und wenn man die heutigen, alles andere als fehlerfreien Browser als Maßstab nimmt, wird man sich bewusst, wie gering die Aussichten sind, einen solchen Standard tatsächlich umzusetzen.

Netscape und Microsoft (und natürlich alle anderen Mitbewerber) haben also viel Arbeit vor sich, um Ihre Browser so auszurichten, dass alle Standards unterstützt werden.

In seinem letzten Buch »The Invisible Computer« setzte sich Donald Norman mit den grundlegenden Problemen der Computerindustrie auseinander. Dabei beginnt er mit einer interessanten Frage: Wer hat sich eine Uhr nur deswegen gekauft, weil dieses Modell die Zeit besser anzeigt? Die Antwort lautet selbstverständlich niemand, da fast alle Uhren die Zeit bis auf ein, zwei Sekunden exakt anzeigen. Aus diesem Grund legen Uhrenkäufer Wert auf Aussehen, Prestige und Preis.

Der Erfinder des Webs, Tim Berners-Lee, drückt es in der Technology Review, vom Juli 1996, so aus:

Jeder, der bei einem Surfausflug über die Angabe »diese Seite wurde für Browser X optimiert« stolpert, macht mit den schlechten alten Zeiten Bekanntschaft. Schließlich standen damals die Chancen sehr gering, Dokumente zu lesen, die auf anderen Betriebssystemen, Netzwerkprotokollen und Textverarbeitungen beruhten.

Und da sich diese Aussage auch auf die einzelnen Browser anwenden lässt, müssen sich die Hersteller folgendes Ziel setzen: Jeder Browser muss die vom W3C definierten Tags unterstützen, Scripts exakt ausführen und StyleSheets einheitlich interpretieren, anstatt den Besucher mit Hinweisen wie »Diese Seite ist für Browser X optimiert« zu nerven. Eben so wie eine Uhr die Zeit anzeigt, ein Telefon mit einem anderen Anschluss Kontakt aufnimmt und eine Stereoanlage alle Audio-CDs abspielt. Zukünftige Browserversionen sollen sich vor allem durch Geschwindigkeit, Benutzeroberfläche und Desktop-Integration auszeichnen.

Bis es aber so weit ist, müssen wir uns mit den Unzulänglichkeiten des Webs auseinander setzen.