Hochschule für Angewandte Wissenschaften Hamburg

Hamburg University of Applied Sciences

Bachelorarbeit

Daniel von Dombrowski

Methoden der Web-Analytik – Ein aktueller Überblick

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Daniel von Dombrowski

Methoden der Web-Analytik – Ein aktueller Überblick

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Wolfgang Gerken
Zweitgutachter: Prof. Dr. Olaf Zukunft

Abgegeben am 01.06.2011

Daniel von Dombrowski

Thema der Bachelorarbeit
Methoden der Web-Analytik – Ein aktueller Überblick

Stichworte
Internet, Statistik, Analyse, Geodaten, Logdateien, Data Warehouse, Datenbank

Kurzzusammenfassung
In dieser Arbeit werden die heute üblichen Methoden zur analytischen Untersuchung von Websites behandelt. Dazu gehören u. a. die Rohdatengewinnung, die Analyse dieser Rohdaten und die Generierung aussagekräftiger Statistiken. Ferner wird dargestellt, welche Schlüsse aus diesen Daten gezogen werden können und wie sie Betreiber von Websites bei ihrer Arbeit unterstützen können. Wie diese Methoden umgesetzt werden können, wird im Rahmen des prototypischen Entwurfs eines eigenen Web-Analytik-Tools erläutert.

Daniel von Dombrowski

Title of the paper
Methods of Web Analytics – A Topical Overview

Keywords
Internet, Statistics, Analysis, Geodata, Logfiles, Data Warehouse, Database

Abstract
This paper covers common methods for the analytic examination of websites. This includes the collection of raw data, analysis of such raw data, and generation of meaningful statistics. Moreover, data interpretation and how it may help website operators will be addressed as well. By designing a prototype web analytics tool, an example implementation of these methods will be explained.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Das Internet hat sich in den letzten Jahren immer weiter zum wichtigsten Kommunikationsmedium der Welt entwickelt. Neben klassischen Diensten, wie E-Mail und dem World Wide Web, gibt es noch viele weitere Online-Dienste, darunter Videostreaming, Dateitransfer und Kommunikationsdienste wie Instant Messaging und Voice-Over-IP. Die unzähligen Anwendungsmöglichkeiten des Internets haben zur rasanten Verbreitung dieses Mediums geführt. Sowohl im kommerziellen als auch im privaten Bereich ist das Internet heute nicht mehr wegzudenken.

Ein Medium dieser großen und weltweiten Reichweite bietet sich für kommerzielle Ziele geradezu an – sei es als Handelsplattform, als Werbefläche oder als Visitenkarte eines Unternehmens.

Ebenso wie bei klassischen Printmedien, ist es auch online von großer Bedeutung, über die Reichweite, Performance und Usability des eigenen Angebots genauestens informiert zu sein. Insbesondere im Bereich der Internet-Websites im World Wide Web kommen hierzu verschiedene Methoden zum Einsatz, die Betreiber bei wichtigen Optimierungsentscheidungen unterstützen können und auch Grundlage für die Werbeflächenvermarktung eines Internet-Angebots sind.

Das Ziel dieser Arbeit ist daher, aktuell eingesetzte Methoden im Hinblick auf ihre Funktionsweisen, sowie ihre Einsatz- und Verknüpfungsmöglichkeiten zu untersuchen, wobei sowohl marktübliche Lösungen beleuchtet werden als auch der Entwurf und die Implementierung eines eigenen Software-Prototypen durchgeführt werden. Das Resultat dieser

Untersuchungen kann Betreiber darin unterstützen, für sich individuell zusammengestellte Lösungen zu finden.

1.2 Gliederung der Arbeit

Die dieser Arbeit zugrundeliegenden Technologien und Begrifflichkeiten werden in Kapitel 2 behandelt.

In Kapitel 3 werden die Vor- und Nachteile der möglichen Quell- bzw. Rohdaten, mit denen Web-Analytik-Methoden arbeiten, untersucht.

Die Auswertung und Interpretation der gewonnenen Rohdaten werden dann zusammen mit gängigen Darstellungsverfahren im Anschluss in Kapitel 4 beschrieben.

Eine Marktanalyse vorhandener kommerzieller und nicht-kommerzieller Web-Analytik-Werkzeuge wird in Kapitel 5 durchgeführt.

Das darauf folgende Kapitel 6 befasst sich mit der Konzeption und der prototypischen Umsetzung einer exemplarischen Softwarelösung, in der bestimmte Verfahren nochmals verdeutlicht werden.

Eine Zusammenfassung der Erkenntnisse und ein Ausblick auf künftige Trends werden schließlich in Kapitel 7 beschrieben.

Kapitel 2

Grundlagen

2.1 Web-Analytik

Hinter dem Begriff Web-Analytik (engl. Web Analytics; auch Traffic-Analyse oder Web-Controlling) verbirgt sich eine große Anzahl von Methoden und Mechanismen, die das Erfassen und Analysieren von Benutzerdaten beschreibt, die im Internet oder auch in einem Intranet auf Websites anfallen. Aus diesen gesammelten Daten lassen sich vielerlei Informationen gewinnen, die den erfolgreichen und technisch reibungslosen Betrieb einer Website unterstützen können. [Ree08]

2.2 Webserver und Websites

Sowohl im lokalen Intranet als auch im global verfügbaren Internet werden Websites auf Webservern abgelegt. Websites bestehen wiederum aus einzelnen Webseiten, die entweder einzeln statisch erstellt werden oder aber dynamisch generiert werden können.

Damit Websites vom Anwender mit Hilfe einer Client-Applikation, dem Webbrowser, ansprechend dargestellt werden können, werden sie heutzutage vor allem mit Hilfe der Sprachen HTML und CSS erstellt. Bei diesen Sprachen handelt es sich nicht um Programmiersprachen, denn sie dienen ausschließlich der Formatierung, Strukturierung und Gestaltung von Websites.

Erst durch Skriptsprachen erhalten Websites zusätzlich eine gewisse Programmlogik, mit der auf Benutzerereignisse reagiert werden kann. Sowohl client- als auch serverseitige

Technologien kommen hier flächendeckend zum Einsatz. Zu den clientseitigen Sprachen zählen vor allem JavaScript und Flash-ActionScript. Diese Programmlogik läuft weitgehend autonom ab, sodass der Server in der Regel keine Informationen über den aktuellen Zustand des Programms hat.

Mittels serverseitiger Technologien, wie PHP, Perl, JSP oder ASP können Websites komplett dynamisch erzeugt werden. Bei dieser Methode werden Teilelemente je nach Situation oder Zustand generiert und zu einem Gesamtkonstrukt zusammengeführt. Gestalterisch bieten dynamisch generierte Webseiten den Vorteil, dass bestimmte Teile, wie Kopfzeilen, Fußzeilen oder Navigationsmenüs, nur ein einziges Mal erstellt werden müssen. Der eigentliche Inhalt der angeforderten Webseite kann in ein Grundgerüst an einer vordefinierten Stelle eingefügt werden. Programmiertechnisch bieten serverseitige Technologien vor allem den Vorteil, dass aktuellste Daten automatisch aus externen Quellen, wie z. B. Datenbanken, verwendet werden können, ohne ständige Änderungen an der eigentlichen Website durchzuführen. Da zudem der Quellcode ausschließlich auf dem Server ausgeführt wird, ist er im Gegensatz zur clientbasierten Programmlogik für den Client auch nicht einsehbar, wodurch wichtige Algorithmen vor unbefugten Zugriffen Dritter geschützt bleiben.

Weitere Anwendungsmöglichkeiten ergeben sich, wenn beide Technologien miteinander kombiniert werden. Beim so genannten AJAX-Ansatz (Asynchronous JavaScript and XML) kann beispielsweise über einen clientseitigen JavaScript-Code eine Verbindung zu einem serverseitigen PHP-Skript zum Austausch von Daten oder zum Nachladen von dynamischen Seiteninhalten aufgebaut werden.

2.2.1 Client-Server-Kommunikation

Webserver stellen Inhalte in der Regel per TCP (Transmission Control Protocol) über den Port 80 bereit und verwenden dafür das HTTP-Protokoll (Hypertext Transfer Protocol). Sowohl Server als auch Client müssen somit eine Reihe von Protokollen und Standards befolgen, um miteinander kommunizieren zu können. Gut veranschaulichen lassen sich diese im so genannten OSI-Schichtenmodell, einem Designmodell, das die für die Kommunikation wichtigen Protokolle und Verfahren sieben aufeinander folgenden Schichten zuweist und deren Eigenschaften beschreibt. [LL10]

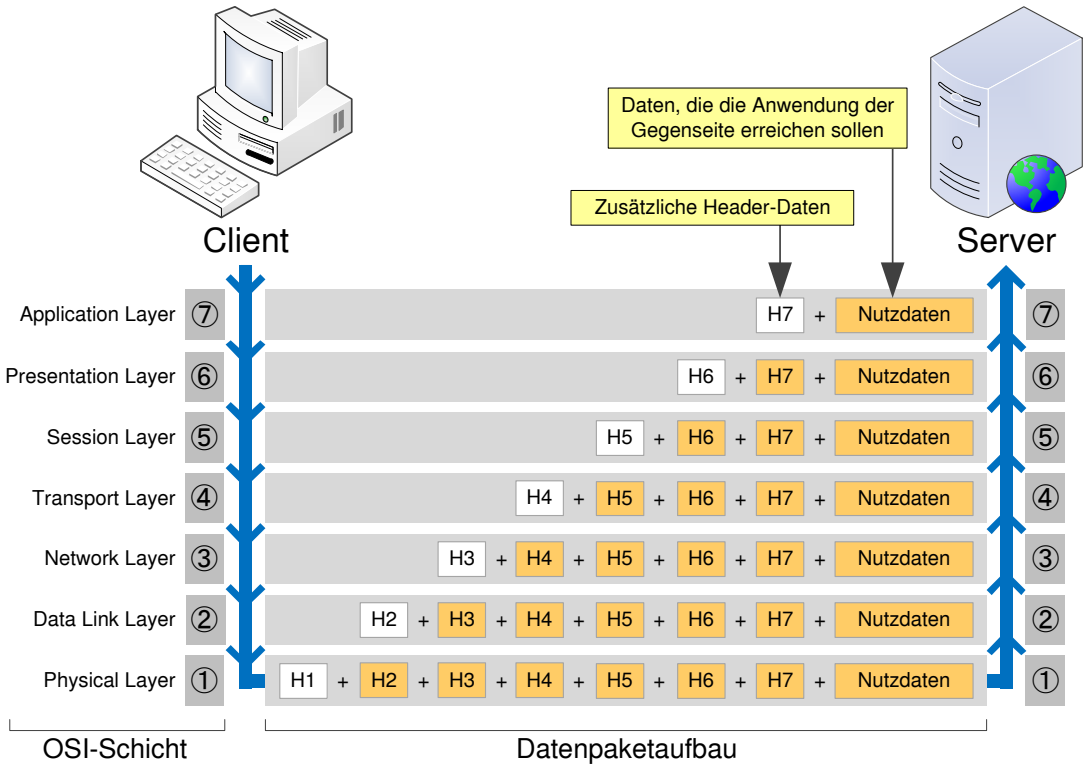


Abbildung 2.1: Vereinfachtes OSI-Schichtenmodell

Der Ablauf beginnt beim Client im Application Layer (Schicht 7), auf der die Anwendungs-Protokolle angesiedelt sind – u. a. solche, die bei einem Webbrowser zum Einsatz kommen. In dieser Schicht ist also auch das HTTP-Protokoll einzuordnen.

Im Presentation Layer (Schicht 6) werden Übertragungskonventionen und Zeichenkodierungen festgelegt.

Verbindungsaufbau und Probleme bei der Datenübertragung sind im Session Layer (Schicht 5) geregelt.

Schicht 4 (Transport Layer) stellt den größten Teil der für eine reibungslose Kommunikation benötigten Dienste bereit. Auch Protokolle wie TCP und UDP sind hier angesiedelt. Während das UDP-Protokoll verbindungslos ist, also jedes Datenpaket für sich alleine steht, wird beim TCP-Protokoll eine Verbindung aufgebaut und gehalten, solange die Datenübertragung stattfindet. UDP wird vor allem für Multimedia-Anwendungen, wie Videostreaming, genutzt, da es hier nicht darauf ankommt, dass wirklich jedes Datenpaket das Ziel erreicht.

Wenn eine Übertragungsgarantie und eine definierte Reihenfolge benötigt werden, wie es beim HTTP-Protokoll der Fall ist, so setzt man in der Regel auf TCP.

Dienste des Network Layers (Schicht 3) verwalten eintreffende und ausgehende Datenpakete im Hinblick auf Herkunft und Ziel bzw. Absender und Empfänger, die beispielsweise durch ihre IP-Adressen identifiziert werden. Zu den wichtigsten Protokollen in Schicht 3 zählen IPv4, IPv6 und ICMP.

Im Data Link Layer (Schicht 2) werden die Daten aufgeteilt und in kleineren Paketen zum Empfänger geschickt, der diese Daten wieder zusammenführt und anschließend auf Übertragungsfehler überprüft. Auf dieser Ebene werden zudem auch die physikalisch verbundenen Endgeräte mit ihrer Media-Access-Control-Adresse (MAC) adressiert, an die die aufgeteilten Datenpakete schließlich gesendet werden.

Schicht 1 (Physical Layer) beschreibt Übertragungsverfahren und Übertragungsmedien im Hinblick auf Spezifikationen wie Pinbelegungen, Spannungswerten und Drahtlosfrequenzen. Hier werden die einzelnen Bits schließlich über das Übertragungsmedium zum Empfänger transportiert.

Wie in Abbildung 2.1 dargestellt, werden den eigentlichen Nutzdaten auf der Absenderseite mit jeder Schicht weitere Zusatz-Header hinzugefügt, in denen sich z. B. Informationen wie Ziel-Adresse, -Port und Prüfsummen befinden. Aus Header und Nutzdaten wird anschließend ein Paket geschnürt und an die darunter liegende Schicht weitergegeben, welche dieses Paket wieder als Nutzdaten versteht. Auf der Gegenseite werden diese geschnürten Pakete durch die einzelnen Schichten nach und nach wieder geöffnet. In jeder Schicht werden die für sie bestimmten Header vom Stapel entfernt, bevor die Daten dann an die darüber liegende Schicht weitergegeben werden. Am Ende der Kette erreichen allein die tatsächlichen Nutzdaten ihr Ziel.

2.2.1.1 HTTP-Protokoll

Das HTTP-Protokoll hat einen für Netzwerkprotokolle typischen Aufbau, der neben den eigentlichen Nutzdaten auch über einen Header-Teil verfügt. Dieser Header-Teil findet sowohl bei Anfragen des Clients als auch bei Antworten des Servers Einsatz. Das Protokoll ist im OSI-Schichtenmodell im Application Layer angesiedelt.

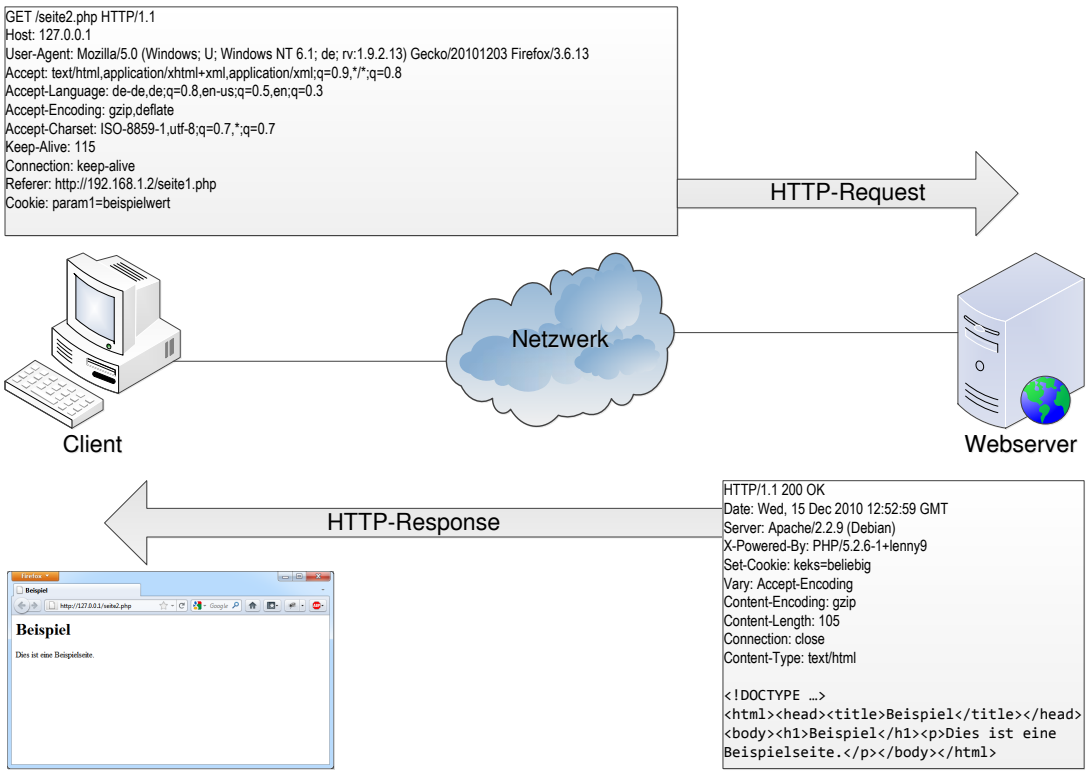


Abbildung 2.2: Typischer Ablauf einer HTTP-Anfrage

Wie in Abbildung 2.2 dargestellt, besteht die Anfrage (Request) des Clients für gewöhnlich ausschließlich aus dem Header-Teil, der im einfachsten Fall den Typ der Anfrage, den Namen des angeforderten Objekts, die Protokollversion und den Hostnamen des Servers beinhaltet. Ein Webbrowser nutzt den Header-Teil üblicherweise noch für zahlreiche weitere Attribute, die auch für die Web-Analytik von Bedeutung sind. Dazu gehören neben den o. g. Basis-Attributen auch `User-Agent`, `Referer` sowie `Cookie`. Bei einer Anfrage kommen Nutzdaten in der Regel nur dann zum Einsatz, wenn Formulare oder sonstige Daten, z. B. im Rahmen eines Datei-Uploads, an den Webserver übermittelt werden.

Die Antwort (Response) des Webservers ist sehr ähnlich aufgebaut, besitzt also ebenfalls einen Header. Bei Client-Anfragen des Typs `GET` und `POST` sind in der Antwort aber üblicherweise immer Nutzdaten enthalten. Diese Nutzdaten entsprechen dem Inhalt des angeforderten Objekts, d. h. entweder einer Datei des Dateisystems oder der dynamisch generierten Antwort einer serverseitigen Web-Applikation.

Webbrowser generieren Anfragen anhand von URIs, die u. a. vom Benutzer direkt in ein Adressfeld eingegeben werden können. URI steht für Uniform Resource Identifier und be-

steht aus den Segmenten `scheme` und `hier-part`. Letztgenanntes teilt sich noch in die beiden Teilsegmente `authority` und `path` auf. Optional kann dieses Grundgerüst noch um die beiden Segmente `query` und `fragment` ergänzt werden. [BLFM05]

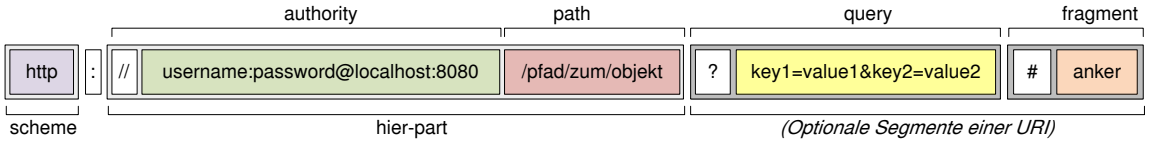


Abbildung 2.3: Aufbau einer URI

scheme beschreibt das zu verwendende Protokoll des Application Layers. In Webbrowsern ist `http` das gängigste, aber auch `https` (verschlüsselte HTTP-Verbindung), `mailto` (E-Mail-Adresse; kein Netzwerkprotokoll, aber dennoch als scheme zulässig) und `ftp` (File Transfer Protocol) werden von vielen Webbrowsern unterstützt. Dem scheme-Segment folgt stets ein Doppelpunkt.

authority Dieses Segment ist je nach verwendetem scheme verschieden. Zumeist befinden sich hier, wie im `http`-scheme, der Name des Zielservers und der entsprechende Port, auf dem ein Server-Dienst auf Anfragen wartet. Laut Spezifikation ist das authority-Segment ein optionales Feld. Während es bei den meisten schemes Pflicht ist, wird es z. B. beim `mailto`-scheme übersprungen. Kommt das authority-Segment zum Einsatz, so beginnt es stets mit der Zeichenfolge „`//`“.

path Das path-Segment beschreibt das angeforderte Objekt des angesprochenen Server-Dienstes. Dies kann sowohl ein bereitgestelltes Webserver-Objekt (im `http`- und `https`-scheme) als auch eine E-Mail-Adresse sein (im `mailto`-scheme). Das path-Segment ist ein Pflichtfeld, dessen Aufbau je nach scheme variiert.

query In diesem optionalen Segment, das durch ein Fragezeichen eingeleitet wird, befinden sich die an das angeforderte Objekt zu sendenden Daten. Diese Daten können beliebig aussehen. Üblicherweise werden hier unsortierte Key-Value-Paare übertragen, die durch das Zeichen `&` getrennt werden.

fragment Das fragment-Segment stellt die Referenz auf einen Teil des angeforderten Objektes dar. Dieses Segment wird nicht mit an den Server übertragen, sondern ausschließlich durch den Client verarbeitet. Webbrowser nutzen es für den Sprung zu einem definierten Ort auf der Webseite.

2.2.2 Aufbau von Webseiten

Wie in Abschnitt 2.2 bereits angeführt, werden Webseiten in einer Auszeichnungssprache, wie etwa HTML (Hypertext Markup Language), beschrieben. Während diese bei der reinen Strukturierung der darzustellenden Inhalte Verwendung findet, wird die äußere Gestaltung des Dokuments üblicherweise mit Hilfe einer Formatierungssprache – zumeist CSS (Cascading Style Sheets) – realisiert. Zusätzlich besteht die Möglichkeit, mit Hilfe von Skriptsprachen, wie z. B. JavaScript, programmähnliche Funktionalitäten auf der Webseite zur Verfügung zu stellen. Die Basis-Struktur eines gültigen HTML-Dokuments besteht mindestens aus drei Teilen DOCTYPE, HEAD und BODY. [MST]

DOCTYPE Der DOCTYPE-Teil enthält die Document Type Definition (DTD) des HTML-Dokuments. Mit dieser DTD wird die verwendete HTML-Version gekennzeichnet. Optional kann auch die Adresse zur DTD genannt werden, mit der ein Webbrowser die für diese HTML-Version gültigen Regeln abfragen kann.

HEAD Im HEAD-Teil befinden sich weitere Informationen, sowohl für den Webbrowser als auch für Suchmaschinen und sonstige Robots. Dazu gehört vor allem der Titel der Webseite, den der Browser in seiner Titelleiste anzeigt, das „Favicon", einer Symboldatei, die in der Adressleiste erscheint, sowie weitere Metadaten. Diese legen u. a. fest, in welchem zeitlichen Intervall der Webbrowser die Webseite neu laden bzw. auf eine dritte Webseite wechseln soll. Für Suchmaschinen befinden sich im HEAD-Teil u. a. Keywords und Beschreibungen der Seite. Außerdem lässt sich sowohl CSS- als auch JavaScript-Code im HEAD-Teil ablegen oder aus externen Quelldateien einbinden.

BODY In diesem Teil befinden sich schließlich die Elemente, die die Struktur einer HTML-Seite definieren. Neben simplen Text-Formatierungsanweisungen wie Fett- und Kursivschrift gehören auch komplexere Konstrukte, wie Tabellen, Auflistungen, Layer und Formulare dazu. Letztgenannte können dazu verwendet werden, Benutzereingaben an ein serverseitiges Skript zu übermitteln. Weitere wichtige Anweisungen umfassen Hyperlinks, also Verweise auf weitere Seiten, sowie das Einbinden von Grafiken und weiteren Multimedia-Objekten, die durch Plugins bereitgestellt werden.

HTML-Elemente besitzen einen öffnenden und einen schließenden Befehl (Tag), die den zu gestaltenden Inhalt umschließen. Ferner existieren auch Befehle, die für sich alleine stehen, also keinen schließenden Tag besitzen. In XHTML, einer neu definierten Fassung

von HTML in XML-Notation, wird bei diesen Befehlen ein Schrägstrich mit in den öffnenden Tag übernommen, der somit auch gleichzeitig zum schließenden Tag wird:

```
<img src=”bild1.png” alt=”Apfel” />
```

Mit CSS können Formatierungsanweisungen beschrieben werden, die weitaus umfangreichere Möglichkeiten bieten. CSS-Anweisungen können über das HTML-Elementattribut `style` definiert werden. Dieses Attribut steht für alle HTML-Elemente zur Verfügung, die eine visuelle Komponente beschreiben. Um mehrfache, identische Formatierungsanweisungen im HTML-Gesamtdokument zu vermeiden, bietet es sich an, eine Gruppe von CSS-Klassen zu definieren, die dann für mehrere HTML-Elemente verwendet werden können. Die CSS-Klassennamen, die stets mit einem Punkt (`.`) beginnen, können über das HTML-Elementattribut `class` verwendet werden. Klassennamen, die mit einer Raute (`#`) beginnen, werden auf Elemente angewendet, denen eine bestimmte, eindeutige `id` zugewiesen wurde. Allgemein gültige Klassen, die ohne eine explizite HTML-Anweisung zum Einsatz kommen, besitzen einen Klassennamen, der dem Namen eines HTML-Befehls entspricht. JavaScript-Code kann mit Hilfe von Event-Handlern ausgeführt werden, die über spezielle HTML-Attribute definiert werden, wie z. B. `onclick` (Klicken auf ein Element) oder `onmousemove` (Bewegen der Maus über einem Element).

Listing 2.1 HTML-, CSS- und JavaScript-Code einer Beispiel-Webseite

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
2  <html>
3    <head>
4      <title>Beispiel</title>
5      <style type="text/css">
6        body { font-family: Arial, sans-serif; }
7        table { background-color: lightgray; }
8        .fett { font-weight: bold; }
9        #absatz1 { font-size: 200%; }
10     </style>
11     <script type="text/javascript">
12       function meldung() {
13         alert("Hello World!");
14       }
15     </script>
16   </head>
17   <body>
18     <p id="absatz1" class="fett">Lorem ipsum dolor sit amet...</p>
19     <table style="border: 1px dotted black;"><tr>
20       <td style="color: red;" onclick="meldung();">Cell 1</td>
21       <td style="color: green; font-size: 150%;">Cell 2</td>
22       <td class="fett" style="color: blue;">Cell 3</td>
23     </tr></table>
24   </body>
25 </html>
```

Die CSS-Klassen sorgen in diesem Beispiel dafür, dass die Tabellen des HTML-Dokuments eine hellgraue Hintergrundfarbe erhalten. Zudem wird für alle visuellen Elemente eine serifenlose Schriftart, vorzugsweise „Arial", verwendet. Elemente, die die Klasse *fett* referenzieren, erhalten eine fettgedruckte Schrift und eine doppelte Schriftgröße. Durch den Klick auf den Inhalt der ersten Tabellenzelle („Cell 1") wird der JavaScript-Code der Funktion `meldung()` ausgeführt, mit dem der Text „Hello World!" in einem Meldungsfenster ausgegeben wird.

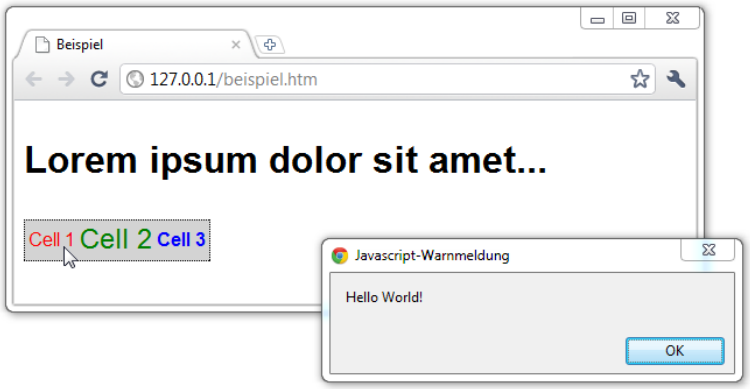


Abbildung 2.4: Darstellung des HTML-Dokuments aus Listing 2.1 in einem Webbrowser

2.3 eCommerce

Unter eCommerce versteht man heute vor allem Online-Handelsplattformen, auf denen interessierte Kunden die Möglichkeit haben, Produkte zu erwerben. Viele Händler bieten neben ihren lokalen Kaufhaus-Niederlassungen zunehmend auch Internet-Plattformen an, um sowohl ihren Kunden die Möglichkeit zu bieten, sich über das aktuelle Sortiment zu informieren als auch das Interesse neuer Kunden zu wecken. Während es bei einigen Anbietern beim einfachen Online-Warenkatalog bleibt, bieten andere als weiteren Vertriebsweg auch einen Online-Shop an, bei denen die Kunden eine Bestellung aufgeben können, um sich die Ware per Post zustellen zu lassen. Einer der bekanntesten Größen im Internet ist heute der Konzern Amazon, der sich ausschließlich auf den Internet-Handel spezialisiert hat und somit keine Filialen mit Verkaufsräumen besitzt.

Es haben sich neben dem klassischen eCommerce-Bereich aber auch noch viele weitere kommerzielle Dienste im Internet etabliert. Anbieter bzw. Betreiber von verschiedensten Online-Angeboten, wie Nachrichtenportalen, Communitys und Fachmagazinen verfolgen

zumeist nicht nur das Interesse, ihr Produkt zu vermarkten, sondern auch weitere Einnahmequellen zu erschließen. Hierzu stellen sie auf ihren Websites zum Teil große Werbeflächen bereit, die den Werbeanzeigen in Zeitschriften sehr ähneln. Im Gegensatz zur gedruckten Werbeanzeige bietet sich im Internet aber vor allem die Möglichkeit an, animierte und interaktive Werbung zu schalten.

Der Online-Auftritt eines Betreibers muss nicht nur zur Zielgruppe des werbenden Kunden passen, sondern diesen auch von der Reichweite seines Online-Auftritts überzeugen. Es ist daher wichtig, dass beiden Parteien genaue Kennzahlen über die Leserschaft zur Verfügung stehen.

Die Web-Analytik dient aber nicht nur einer bloßen Auswertung von Besucher-Kennzahlen, sondern kann auch Schwachstellen in der Benutzerführung und der Usability des Angebots aufdecken.

Kapitel 3

Rohdatengewinnung

3.1 Logdateien

Jede moderne Webserver-Software bietet die Möglichkeit, Zugriffe in einer Logdatei zu protokollieren. Diese kann nicht nur zur Fehlersuche verwendet werden, sondern ist auch essenziell beim Analysieren von Besucherdaten. Jeder Zugriff entspricht einer Protokollzeile in der Logdatei.

Obwohl sich die Implementierung der verschiedenen Webserver untereinander stark unterscheidet, bieten alle gängigen Webserver die Möglichkeit, das Protokollformat mittels Platzhaltervariablen frei zu konfigurieren. Ein Beispiel hierfür ist das so genannte „Common Logfile Format“, welches durch die folgende Struktur beschrieben wird (vgl. [W3C95, Apa, Lig11]):

```
remotehost rfc931 authuser [date] "request" status bytes
```

remotehost Der Remote Host ist entweder die IP-Adresse oder der daraus aufgelöste DNS-Hostname des Clients.

rfc931 RFC931 ist ein Standard der Internet Engineering Task Force (IETF), welcher die Identitätsbestimmung eines Clients beschreibt. Da diese Art der Identitätskontrolle im Internet aber so gut wie keine Rolle spielt, wird das Attribut zumeist ignoriert und in der Logdatei durch das Zeichen „-” ersetzt. [Joh85, Joh93]

authuser Dieses Attribut kommt zum Einsatz, wenn ein kennwortgeschützter Bereich einer Website aufgerufen wird. Hier steht in diesem Falle der Name des erfolgreich

angemeldeten und somit autorisierten Benutzers. Ansonsten wird auch hier üblicherweise das Zeichen „-" eingesetzt.

date An dieser Stelle befindet sich der sekundengenaue Zeitstempel des Zugriffs.

request Hier befindet sich neben der Protokollversion und dem Typ der Anfrage vor allem der Pfad des aufgerufenen Objekts. Dahinter kann sich sowohl eine Datei des Dateisystems auf dem Server, als auch ein virtueller Pfad befinden, der über eine Rewrite-Direktive eingerichtet wurde.

status Das Status-Attribut ist im IETF-Standard RFC2616 festgelegt und gibt mit Hilfe von dreistelligen Statuscodes vor allem Auskunft darüber, ob die Anfrage erfolgreich verlief. [FGM+99]

bytes In diesem Attribut befindet sich die Anzahl der Nutzdaten-Bytes. Diese Anzahl entspricht entweder der Größe einer Datei oder der Gesamtzeichenlänge eines dynamisch generierten Objekts.

Das „Common Logfile Format" wird von Webservern standardmäßig um zwei weitere, für die Web-Analytik unverzichtbare Attribute erweitert, die von allen gängigen Browsern im Request-Header übertragen werden. Diese Attribute werden jeweils von Anführungszeichen umschlossen und an das oben beschriebene Format angehängt. Das daraus resultierende Format trägt die Bezeichnung „Combined Logfile Format".

referer Dieses Attribut enthält eine URI, die kennzeichnet, auf welcher Webseite sich der jeweilige Benutzer gerade befindet, während er auf das angeforderte Objekt zugreifen will. Dieses Attribut kann somit Aufschluss darüber geben, über welche Suchmaschine und mit welchen Suchbegriffen ein Besucher die Website gefunden hat. Obwohl die korrekte Schreibweise des englischen Wortes eigentlich „Referrer" lautet, wird zumeist der Begriff „Referer" verwendet. [Ree08]

user-agent Heute versteht man unter dem User-Agent vor allem einen String, der die eingesetzte Software, also den Client oder Browser und ggf. das Betriebssystem, beschreibt. Dieses Feld existiert für statistische Zwecke, für das Untersuchen von Protokollverletzungen und für die automatisierte Client-Erkennung. Website-Betreiber können das Attribut verwenden, um z. B. auf Browser-Limitierungen zu reagieren. Ein weiterer Einsatzzweck kann eine spezielle Version einer Website sein, die dem GUI-Design bestimmter Geräte angepasst ist. Das `User-Agent`-Feld wird im HTTP-Request-Header übertragen. [FGM+99]

Eine gewöhnliche Zeile in der Logdatei kann somit wie folgt aussehen:

```
12.34.56.78 - - [12/Sep/2010:23:55:05 +0200]
"GET /script.php?p=12345 HTTP/1.1" 200 2248
"http://www.google.com/search?q=example"
"Opera/9.50 (Nintendo DSi; Opera/507; U; en-US)"
```

3.2 Zählpixel

Beim Besuchen einer Website mit einem Webbrowser wird standardmäßig jede auf der Webseite enthaltene Grafik heruntergeladen und dargestellt. Auch diese Eigenschaft lässt sich für die Gewinnung von Rohdaten nutzen.

Zum Einsatz kommt bei dieser Methode ein sogenannter Zählpixel. Dies ist eine in ein HTML-Dokument eingebundene, meist transparente und somit unsichtbare 1×1-Pixel-Grafik. Dahinter verbirgt sich keine gewöhnliche Grafikdatei, sondern ein serverseitiges Skript, das die Client-Informationen, die während des Zugriffs übertragen werden, verarbeitet und speichert. Diese Informationen können sowohl aus dem HTTP-Request-Header, als auch aus zusätzlich angegebenen Parametern bestehen, die in den `query`-Teil der URI gesetzt werden (vgl. Kapitel 2.2.2).

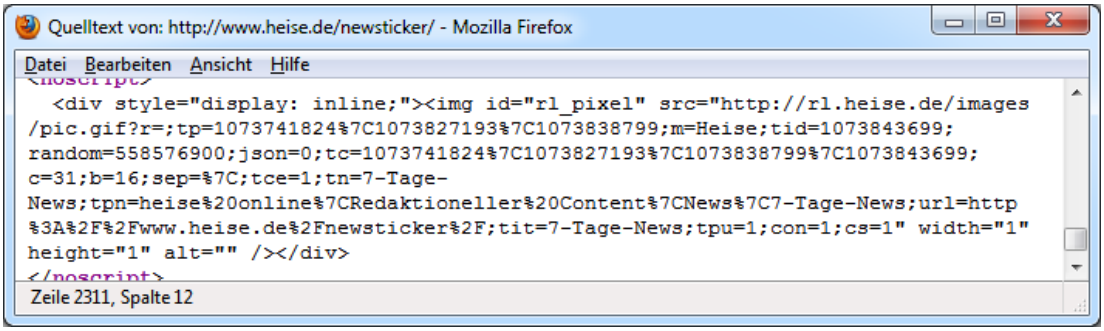


Abbildung 3.1: Typischer Zählpixel im HTML-Quellcode von heise online

Zählpixel sind heutzutage sehr verbreitet. Diese Art der Rohdatengewinnung wird bereits seit 1996 aktiv genutzt und besitzt somit auch bei Website-Betreibern einen großen Bekanntheitsgrad. [Ree08]

In Abbildung 3.1 ist ein typischer Zählpixel-Codeschnipsel im HTML-Quellcode der IT-News-Website „heise online" zu sehen. Gut zu erkennen ist, wie hier mit dem `<img>`-HTML-Tag ein Grafikobjekt mit dem Namen „pic.gif" geladen wird. Die Größenangabe `width="1" height="1"` ist dabei das typischste Merkmal eines Zählpixels. Aber auch die zusätzlichen Parameter, die beim Aufruf im `query`-Teil der URI übergeben werden, lassen auf die Nutzung eines Skripts zur Analyse von Besucherinformationen schließen. Zu erkennen sind in diesem Beispiel neben Zufallszahlen, Zeitstempeln und Informationen zur aufgerufenen Webseite auch kryptische Sitzungsinformationen, die vom Analyse-Skript intern weiterverarbeitet werden.

Dieser Codeschnipsel wird zumeist am Ende eines HTML-Dokuments, also im hinteren Bereich des HTML-BODY-Teils, platziert. Durch diese Art der Platzierung werden letztlich nur erfolgreich geladene Webseiten erfasst.

3.3 JavaScript

Neben Logdateien und Zählpixeln gibt es noch eine weitere, weit verbreitete Methode der Gewinnung von Rohdaten, welche auf die clientseitige Skriptsprache JavaScript setzt.

Die in Webbrowsern vordefinierten JavaScript-Objekte bieten den Zugriff auf weitere Daten (vgl. [MSS]), darunter:

window Hinter diesem Objekt verbergen sich Attribute und Funktionen, die das Browserfenster betreffen. Dazu gehören beispielsweise das Vorhandensein von Scrollleisten und die Abmessungen des Browserfensters.

document Dieses Objekt gewährt Zugriff auf Attribute und Funktionen der im Browser dargestellten Webseite. Hiermit können alle Eigenschaften von HTML-Elementen abgefragt und ggf. verändert werden. Zu den bereitgestellten Attributen zählen u. a. auch `referrer` und `cookie`.

screen Mit diesem Objekt können Bildschirmattribute wie Auflösung und Farbtiefe abgefragt werden.

navigator Dieses Objekt liefert Informationen zum eingesetzten Browser. Dazu zählt z. B. die Sprachversion des Browsers, aber auch die Cookie- und Java-Unterstützung lässt sich abfragen.

Auch bei dieser Methode muss auf den zu untersuchenden Webseiten ein Codeschnipsel eingefügt werden. Die Platzierung kann sowohl im HEAD- als auch im BODY-Teil des HTML-Dokuments erfolgen.

Um die gesammelten Daten zu verarbeiten und zu speichern, kann ein serverseitiges Skript eingesetzt werden, welches diese Daten entgegennimmt. Die Übertragung kann dabei beispielsweise asynchron über einen AJAX-Submit erfolgen. Eine gängigere Methode ist es, das JavaScript mit einem Zählpixel zu kombinieren und die zusätzlichen JavaScript-Daten über den `query`-Teil der URI übertragen (vgl. Kapitel 2, [BLFM05]). Hierbei kann der Codeschnipsel jedoch nur im BODY-Teil platziert werden, da Grafikelemente nicht im HEAD-Teil definiert werden können.

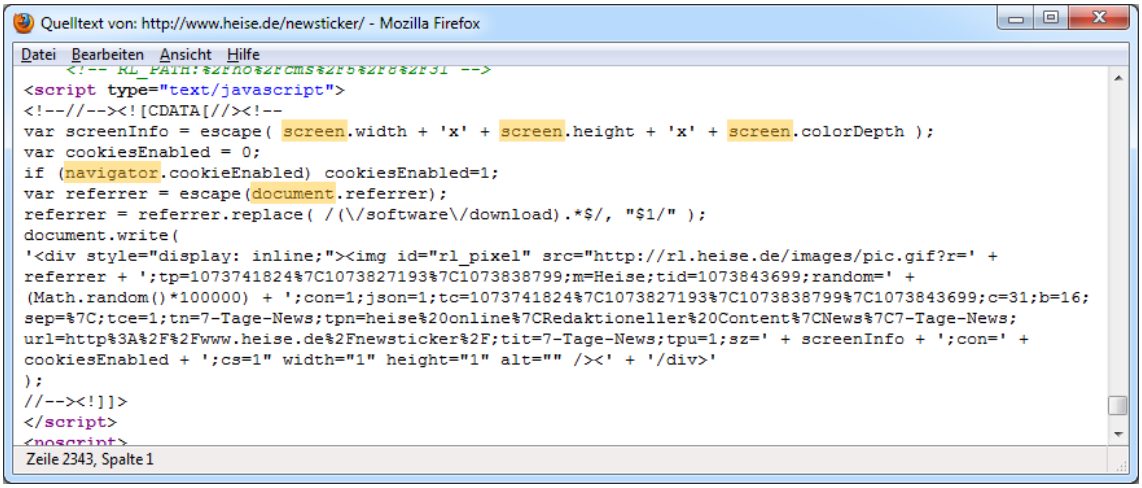


Abbildung 3.2: JavaScript-unterstützter Zählpixel im HTML-Quellcode von heise online

In Abbildung 3.2 ist der JavaScript-Teil des Tracking-Codeschnipsels von „heise online" dargestellt. Erkennbar ist, dass hier eine Kombination aus Zählpixel- und JavaScript-Logging zum Einsatz kommt, um zusätzliche Informationen zu erhalten, die nur durch die JavaScript-Objekte `document`, `screen` und `navigator` geliefert werden.

Neben dem klassischem Logging gibt es bei der JavaScript-Einbindung noch weitere Analyse-Möglichkeiten. Jedem Element einer Webseite kann Code hinterlegt werden, der beim Klicken oder auch beim bloßen Bewegen der Maus ausgeführt wird (vgl. Kapitel 2.2.3). Durch diese Technik können beispielsweise Daten für Usability-Analysen gesammelt werden.

3.4 Bewertung

Die Aufzeichnung einer Webserver-Logdatei ist die älteste Form der Rohdatengewinnung. Nach wie vor deckt dieses Verfahren einen Teil der heute eingesetzten Web-Analytik-Methoden ab, jedoch mit entscheidenden Schwächen. Auch irrelevanter Besucherverkehr, der beispielsweise durch Hotlinks oder Robots entsteht, wird hier in die Auswertung mit einbezogen. Da beim Hotlinking vor allem Bilddateien und herunterladbare Binärdateien („Downloads“) betroffen sind, wäre ein Filter nach Dateierweiterung zwar denkbar, jedoch insbesondere beim Einsatz von virtuellen Pfaden und serverseitigen Skriptsprachen nicht zuverlässig genug. Zudem ist die Vielfalt der erfassbaren Daten sehr begrenzt. Viele Kenngrößen der heutigen Web-Analytik lassen sich hieraus nicht ableiten, weshalb diese Methode bei professionell betriebenen Websites auch nicht mehr so verbreitet zum Einsatz kommt, wie die beiden anderen angeführten Methoden.

Die Logdatei-Methode bietet aber auch einige Vorteile. Sowohl bei der Zählpixel- als auch bei der JavaScript-Methode ist es erforderlich, alle betroffenen Webseiten mit einem Codeschnipsel auszustatten, damit sie in die Erfassung und Auswertung mit einbezogen werden können. Zudem wird bei diesen beiden Methoden in der Regel jeweils ein weiteres, serverseitiges Skript benötigt, um die gesammelten Daten schließlich abzuspeichern. Die Logdatei wird hingegen direkt vom Webserver geschrieben, der die angeforderten Objekte bereitstellt. Hieraus ergibt sich datenschutzrechtlich ein weiterer Vorteil, denn da die gesammelten Daten üblicherweise auf dem eigenen Server liegen, bleiben sie in der Hand des Betreibers. Dies ist bei der Zählpixel- und JavaScript-Methode nur dann der Fall, wenn eine separate Web-Analytik-Software auf dem eigenen Server betrieben wird. In der Regel greifen Betreiber jedoch bevorzugt auf Lösungen externer Dienstleister zurück.

Zählpixel-Skripte für sich allein bieten bereits eine deutlich höhere Datenvielfalt, als Webserver-Logdateien. Es ist möglich, Cookies zu setzen und somit wiederkehrende Besucher relativ problemlos wiederzuerkennen (vgl. [Ree08]). Die zumeist unerwünschten Hotlinks von Bilddateien, Downloads und sonstigen Objekten werden ferner nicht in Besucheranalysen mit einbezogen, da ein Browser nur bei einem echten Seitenaufruf auf den Zählpixel zugreift. Benötigt man jedoch Statistiken über Zugriffe, die Hotlinks umfassen, ist dennoch der Einsatz einer Logdatei als weitere Rohdatenquelle notwendig.

Die fortschrittlichste Methode, das JavaScript-Tracking, bietet den größten Umfang an erfassbaren Daten. Auf einen Großteil dieser Daten haben die beiden anderen angesprochenen Verfahren keinen Zugriff. Daten wie Bildschirmauflösung der Benutzer stellen sehr

wichtige Kennzahlen dar, weshalb heute vermehrt auf die JavaScript-Methode gesetzt wird, um Quelldaten für spätere Auswertungen zu sammeln.

Die Verbreitung von JavaScript und somit auch die Zuverlässigkeit dieser Methode der Rohdatengewinnung sind aber heute oftmals nicht ausreichend, um verlässliche Statistiken zu generieren. Dies liegt insbesondere an Webbrowsern, darunter auch solchen von mobilen Geräten, die über keine JavaScript-Funktionalität verfügen. Es gibt zudem Benutzer, die JavaScript aus verschiedensten Gründen deaktiviert haben. Damit dennoch eine verlässliche Statistik erstellt werden kann, geht man hier, wie am obigen Beispiel mit heise online erläutert, den Weg, die angesprochenen Arten der Rohdatengewinnung zu kombinieren.

	Logdatei	Zählpixel	JavaScript
Zuverlässigkeit der Rohdatengewinnung	Sehr gut	Gut	Schlecht
Erfordert Codeschnipsel-Einbau	Nein	Ja	Ja
Erfordert Zusatzskript für Datenspeicherung	Nein	Ja	Ja
Vielfalt der erfassbaren Daten	Schlecht	Gut	Sehr gut
Filtern von Hotlinks und Robots	Schwierig	Ja	Ja
Wiedererkennung von Besuchern	Schlecht	Sehr gut	Sehr gut

Tabelle 3.1: Vergleich der Methoden zur Rohdatengewinnung

Mit Tabelle 3.1 wird abschließend gezeigt, dass keines der drei Verfahren eine exakte Datenerfassung ermöglicht. Durch Kombination erreichen sie aber eine ausreichend hohe Genauigkeit, um die grundlegende Basis für anschließende Auswertungen zu bilden.

Kapitel 4

Auswertungs- und Interpretationsmöglichkeiten

Aus den gesammelten Rohdaten lassen sich vielfältige Schlüsse ziehen. Unterschieden wird bei der Auswertung grundsätzlich zwischen Statistiken, d. h. den periodisch erstellten Analysen, und Methoden zur Website-Optimierung, die Schwächen einer Website aufdecken können.

4.1 Statistiken

Wie bereits in Kapitel 2 angesprochen, sind Kennzahlen über die Reichweite und Effektivität einer Website sowohl für Betreiber als auch für Werbepartner von großer Bedeutung. Diese Kennzahlen können in Form von Statistiken auf vielerlei Möglichkeiten dargestellt werden.

Zu den wichtigsten Kennzahlen zählen die Gesamtzugriffszahlen, die auf verschiedene Arten ausgewertet werden können – darunter als Hits, Visits oder Page Views.

Zu den **Hits** zählt jeder einzelne Zugriff auf den Webserver, darunter auch Bilder und sonstige Downloads. Ein Hit entspricht dabei einer Zeile in der Logdatei, deren Aufbau in Kapitel 3 beschrieben wird. Parallel zu den Hits kann auch der Traffic gemessen werden, also die summierte Byte-Anzahl aller vom Webserver übertragenen Daten. Diese Kennzahlen eignen sich vor allem dazu, die Gesamtbelastung eines Webservers zu analysieren. Für Website-Vermarktungs- und -Optimierungszwecke sind sie heute jedoch eher bedeutungslos.

Weitaus aussagekräftiger sind die **Visits**, welche im Gegensatz zu Hits echte Besuchersitzungen beschreiben. Aufgrund des zustandslosen HTTP-Protokolls ist diese Information aber nicht auf direktem Wege aus gesammelten Daten extrahierbar. Jeder Zugriff auf ein Objekt des Webservers erfolgt technisch unabhängig vom vorherigen. Es müssen also mehrere Datensätze betrachtet werden und in Zusammenhang gebracht werden. Hierfür wird sich zumeist der IP-Adresse des Besuchers bedient. Innerhalb eines definierten Zeitraumes werden so alle Zugriffe von der gleichen IP-Adresse zu einer Besuchersitzung zusammengefasst. Ändert sich entweder die IP-Adresse oder wird das Zeitlimit überschritten, zählt ein neuer Zugriff gegebenenfalls als neue Besuchersitzung. Dieses Zeitlimit wird in der Praxis üblicherweise auf 30 Minuten festgesetzt. [Ree08]

Unique Visitors definieren sich als tatsächlich eindeutige Besucher und können ebenfalls separat ausgewertet werden. Der Betreiber erhält dabei Auskunft darüber, wie viele Menschen er tatsächlich mit seinem Angebot erreicht. Feststellen lässt sich mit dieser Kennzahl ebenso die Anzahl neuer Besucher, die zuvor noch nicht betrachtet wurden. Mittels Cookies wird jeder Besucher mit einer meist alphanumerischen ID versehen, die dessen Wiederkehr später erkennbar macht. Um Cookies hierfür nutzen zu können, muss, wie in Kapitel 3 beschrieben, ein Zählpixel- oder JavaScript-Codeschnipsel auf der Website hinterlegt werden. Sollen Visits ohne Cookies erfasst werden, kann man die IP-Adresse verbunden mit dem User-Agent dazu heranziehen. [Ree08] Diese Methode ist jedoch – aufgrund dynamisch vergebener IP-Adressen und mehrfach vorkommender User-Agent-Strings – so ungenau, dass sie heute unbedeutend für die Web-Analytik ist. Auf der Website `http://panopticlick.eff.org/` wird hierzu ein Ansatz vorgestellt, der mit Hilfe weiterer Rückmeldungen des Browsers eine beinahe eindeutige Identifizierung des Besuchers erzielen kann. Vor allem die installierten Schriftarten und Browser-Plugins sind unter den Besuchern selten identisch. Diese Informationen ergeben kombiniert ein Abbild, das kaum einem anderen gleicht. Noch findet dieser Ansatz jedoch keine oder kaum Verwendung.

Page Views beschreiben, wie viele echte Inhaltsseiten aufgerufen wurden. Auch für diese Kennzahl muss ein Codeschnipsel auf der Website hinterlegt werden, da die auf der Webseite eingebundenen Objekte, wie Bilder, Skripte oder Stylesheets nicht gezählt werden sollen. Ein Filter nach Dateiendung wäre für die Analyse ausschließlich mit Logdateien zwar ein denkbarer Ansatz – jedoch ist dieser nicht universell einsetzbar. Besonders bei der Verwendung von Skriptsprachen kann sich z. B. hinter der Endung `.php` jede Art von Objekt verbergen: eine echte Inhaltsseite, aber ebenso auch eine Grafik oder eine Binärdatei, die zum Download angeboten wird.

Die besonders für Werbepartner interessante Kennzahl **Ad Views** ist den Page Views sehr ähnlich. Sie beschreibt die Anzahl der angezeigten Werbebanner einer Kampagne auf einer Website. Im günstigsten Fall und bei nur einer geschalteten Werbekampagne entspricht diese Kennzahl genau den Page Views. Diese Zahl verringert sich deutlich durch den Einsatz von Werbeblockern bei den Besuchern. Wenn mehrere Werbekampagnen gleichzeitig gebucht wurden, teilt sich diese Kennzahl zudem unter den einzelnen Kampagnen auf. Weitere Einschränkungen kann es durch ein so genanntes Frequency Capping geben, welches kontrolliert dafür sorgt, dass eine einzelne Werbekampagne dem Besucher in einem definierten Zeitraum nur begrenzt oft angezeigt wird.

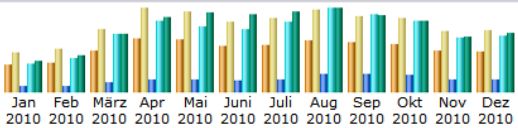
Um eine genauere Messung der insgesamt verfügbaren Ad Views zu erreichen, die also selbst Werbeblocker mitberücksichtigt, würde es sich anbieten, in den jeweiligen Zählpixel-JavaScript-Codeschnipsel eine gewisse Zeichenfolge als Skriptparameter zu integrieren, die von Werbeblockern gefiltert wird. Die Zeichenfolge `-728x90-` ist beispielsweise standardmäßig in der Filterliste gängiger Werbeblocker integriert, wodurch ein Aufruf ähnlich `skript.png?trick=-728x90-` den gewünschten Effekt erzielen würde, dass auch das Zählskript herausgefiltert wird.

Zusammenfassung

Zeitraum	Monat Juli 2010				
Erster Zugriff	01.07.2010 - 00:00				
Letzter Zugriff	31.07.2010 - 23:59				
	Unterschiedliche Besucher	Anzahl der Besuche	Seiten	Zugriffe	Bytes
gesehener Traffic *	**522044**	**817509** (1.56 Besuche/Besucher)	**22868604** (27.97 Seiten/Besuch)	**128927815** (157.7 Zugriffe/Besuch)	**999.16 GB** (1281.56 KB/Besuch)
nicht gesehener Traffic *			**1542144**	**2648188**	**22.67 GB**

* Nicht gesehener Traffic ist Traffic, welcher von Robots, Würmern oder Antworten mit speziellem HTTP-Statuscode

Monatliche Historie



Monat	Unterschiedliche Besucher	Anzahl der Besuche	Seiten	Zugriffe	Bytes
Jan 2010	299189	441023	10661610	52811949	387.12 GB
Feb 2010	320061	475069	11844157	62895668	451.75 GB
März 2010	454860	690229	17624725	107262440	718.99 GB
Apr 2010	588240	923368	22946820	131041460	921.12 GB
Mai 2010	575499	892404	22365452	121145532	977.50 GB
Juni 2010	503802	774696	21053339	115065788	959.37 GB
Juli 2010	522044	817509	22868604	128927815	999.16 GB
Aug 2010	572511	904445	33309663	153745857	1034.68 GB
Sep 2010	544362	840472	33892700	142137433	948.18 GB
Okt 2010	528457	816645	31489346	130178965	874.09 GB
Nov 2010	454623	676815	22798408	99572797	689.53 GB
Dez 2010	449017	683849	22942577	103579243	733.46 GB
Total	5812665	8936524	273797401	1348364947	9694.94 GB

Abbildung 4.1: Zugriffszahlen als Säulendiagramm und Tabelle (AWStats)

All diese Gesamtzugriffszahlen werden für gewöhnlich in Säulendiagrammen oder Tabellen in Bezug zur zeitlichen Komponente dargestellt (vgl. Abbildung 4.1). Sinnvoll sind tägliche, wöchentliche, monatliche und jährliche Auswertungen.

Auch weitere Besuchermerkmale spielen in Web-Analytik-Statistiken eine entscheidende Rolle. Die Informationen zur Client-Software eines Besuchers werden aus dem `User-Agent`-Attribut gewonnen. Enthalten sind hier generell mindestens Informationen zum eingesetzten Browser und Betriebssystem sowie den jeweiligen dazugehörigen Versionsnummern (vgl. Kapitel 3.1).

Informationen zur Besucher-Bildschirmauflösung und -Farbtiefe können bei Fragen der technischen und gestalterischen Realisierung unterstützen. Ähnlich wie bei der User-Agent-Auswertung, wäre es bei einer hohen Anzahl von niedrigen Bildschirmauflösungen eine Überlegung wert, eine mobile Version der Website zu erstellen, die besonders für kleinere Bildschirme geeignet ist.

Falls eine Website mehrsprachig angeboten werden soll, kann es von Interesse sein, die gesprochene Sprache der Besucher zu kennen. Dabei wird entweder die IP-Adresse verwendet, um die Herkunft festzustellen, oder aber das `Accept-Language`-Attribut im Request-Header genutzt, in dem gängige Browser eine oder mehrere Sprachkennungen mit Prioritätsangaben übertragen. Solch ein Attribut kann wie folgt aussehen [FGM+99]:

```
Accept-Language: de, en;q=0.8, ja;q=0.7
```

In diesem Beispiel ist Deutsch die am meisten bevorzugte Sprache. Englisch steht mit einer Priorität von 80 % an zweiter Stelle und Japanisch an dritter Stelle mit einer Priorität von 70 %. Es steht dem Betreiber einer Website aber völlig offen, dieses Attribut zu beachten und passende Inhalte bereitzustellen.

Jeder Besucher besitzt eine IP-Adresse, die ihm von seinem Internet Service Provider (ISP), also Internetdienstanbieter, zugeteilt wurde. In den meisten Fällen lässt sich problemlos bestimmen, welchem ISP eine IP-Adresse angehört und in welchem regionalen Bereich sie eingesetzt wird. Die IP-Adresse eines Besuchers lässt sich somit bis auf das Land oder sogar die Stadt auflösen, in der sich der Besucher aufhält. Zum Beispiel bietet der Dienstleister MaxMind hierfür mit seinen Produkten GeoIP und GeoLite unter `http://www.maxmind.com/app/ip-location` sowohl kostenpflichtige als auch etwas weniger genaue, kostenlose Lösungen an, die in Form von Binär- und CSV-Datenbanken heruntergeladen werden können. Besonders für Betreiber multilingualer Websites kann es von Interesse sein, aus welchen Ländern die jeweiligen Besucher stam-

men. Diese Informationen lassen sich traditionell in Tabellen und Diagrammen, jedoch auch grafisch auf einer Weltkarte darstellen.

Um festzustellen, aus welcher Quelle und auf welche Weise ein Besucher die Website gefunden hat, kann das `Referer`-Attribut ausgewertet werden, das sowohl mit der Logdatei- als auch mit der JavaScript-Analyse ermittelt werden kann. Es enthält die komplette URI der verweisenden Website, aus der ggf. auch abgeleitet werden kann, welche Suchmaschine verwendet wurde und mit welchen Suchbegriffen die Website gefunden wurde (vgl. Kapitel 3).

Top 10 of 48816 Total Referrers

#	Hits		Referrer
1	3884531	2.96%	- (Direct Request)
2	646866	0.49%	translat
3	70868	0.05%	www.fi
4	64698	0.05%	www.b
5	62080	0.05%	www.p
6	44743	0.03%	www.p
7	44422	0.03%	tieba.b
8	43981	0.03%	188.40
9	42001	0.03%	www.s
10	41329	0.03%	www.g

Abbildung 4.2: Tabellarische Referer-Auswertung (Webalizer)

Üblicherweise wird die statistische Darstellung tabellarisch und sortiert nach der Anzahl der Zugriffe realisiert (vgl. Abbildung 4.2).

Bewertend lässt sich sagen, dass mit den heute verbreiteten Technologien nie eine hundertprozentige Genauigkeit erreicht werden kann. Sowohl Caching-Methoden, die durch Proxyserver eingesetzt werden, als auch fehlende oder deaktivierte Browserfunktionalitäten üben eine verfälschende Wirkung auf die Auswertungen aus. Diese Ausreißer sind jedoch so gering, dass sie üblicherweise vernachlässigt werden können. Nur größere Ausreißer, wie solche, die bei den Ad Views durch Werbeblocker auftreten können, müssen kompensiert werden.

4.2 Website-Optimierung

Web-Analytik-Methoden bieten neben der klassischen Gewinnung von Besucherdaten zur Generierung von Statistiken auch Möglichkeiten zur Optimierung von Websites und zur Anpassung an aktuelle Trends.

4.2.1 Funnel-Analyse

Jeder Betreiber einer Website hat in der Regel Ziele, die er mit seinem Angebot erreichen möchte. Diese Ziele können sehr vielfältig und sehr individuell definiert sein. Im eCommerce-Bereich ist für einen Online-Shop eine erfolgreiche Warenbestellung sicherlich ein Ziel. Aber auch die Registrierung eines neuen Benutzers oder die erfolgreiche Teilnahme an einer Umfrage oder einem Gewinnspiel sind als solches denkbar. Wird ein Ziel erreicht, so spricht man von einer Conversion. Die Conversion Rate ist daher auch eine wichtige Kennzahl, die den Erfolg einer Website beschreibt.

In der Web-Analytik bezeichnet man den Pfad zu solch einem Ziel als Funnel. Ein Pfad ergibt sich dabei aus den voneinander abhängigen Einzelschritten, die ein Besucher gehen muss, um ein vordefiniertes Ziel zu erreichen. Einzelschritte werden auch als Events bezeichnet, die durch den Einbau eines Codeschnipsels aufgezeichnet werden können. [Dos09]

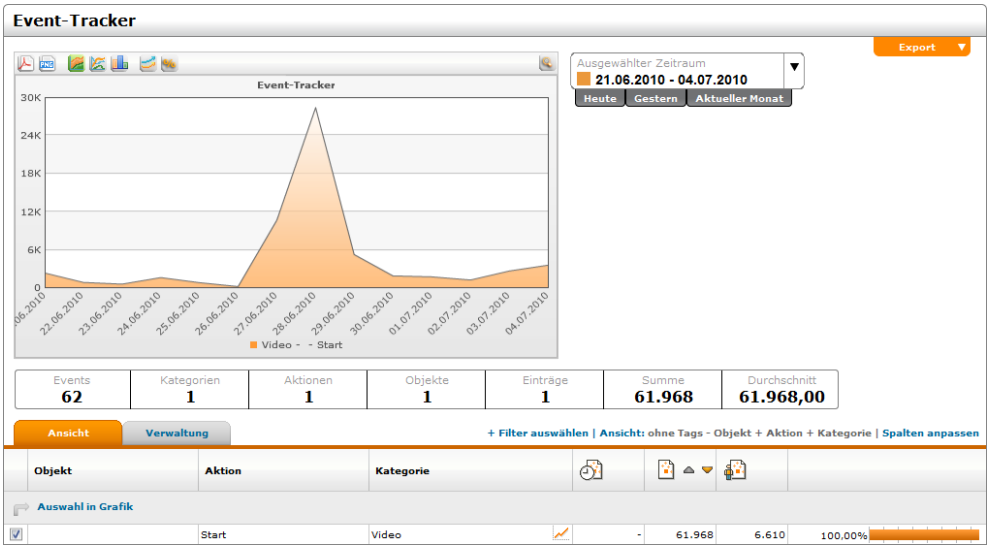


Abbildung 4.3: Starten eines auf der Website eingebundenen Videos als Event (etracker)

Ein Codeschnipsel wird so platziert, dass er bei erfolgreich abgeschlossenem Event ausgeführt wird. Hierdurch wird es dem dahinter stehenden Web-Analytik-System ermöglicht, die Events in Zusammenhang zu bringen und auszuwerten.

Wenn ein Besucher einen Pfad beschritten hat und vor Erreichung des Ziels diesen wieder verlässt, so wird dies als Absprung bezeichnet. So individuell, wie Ziele definiert werden, so verschieden können auch Einzelschritte sein, die Besucher dazu motivieren, den Pfad zu verlassen. Generell beeinflussen vor allem die Anzahl der nötigen Schritte oder ein schlechtes Seitendesign die Conversion Rate negativ. Da sich mit jedem Einzelschritt die Wahrscheinlichkeit von Absprüngen erhöht, beschreibt der Begriff Funnel (engl. für Trichter) dieses Verhalten sehr gut.

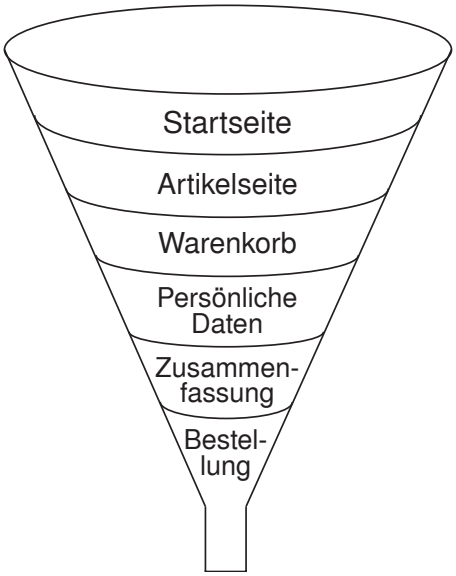


Abbildung 4.4: Darstellung der Funnel-Eigenschaft

Mit der Auswertung der aufgezeichneten Daten können kritische Punkte des Pfades im Rahmen einer Funnel-Analyse leicht identifiziert werden. Ist ein solcher kritischer Punkt bzw. Einzelschritt gefunden, bietet es sich an, mit Veränderung der Platzierung und Reihenfolge von Seitenelementen zu experimentieren und über einen Zeitraum Vorher-Nachher-Vergleichsstudien anzufertigen. Indem nach und nach die jeweils bessere Lösung dauerhaft implementiert wird, kann die Absprungrate schließlich deutlich verringert werden. [Dos09, Kin08]

Grundsätzlich hat sich herausgestellt, dass kritische Punkte vor allem so gestaltet werden müssen, dass ein Besucher hier möglichst wenig Ablenkung erfährt. Beim Online-Händler

Amazon.de sind z. B. alle Login- und Registrierungsseiten sehr minimalistisch gestaltet. Bei der Anlegung eines neuen Benutzerkontos besteht der Inhaltsteil ausschließlich aus einem einfachen Formular, das nur die grundlegendsten Daten abfragt: Name, E-Mail-Adresse und Passwort.

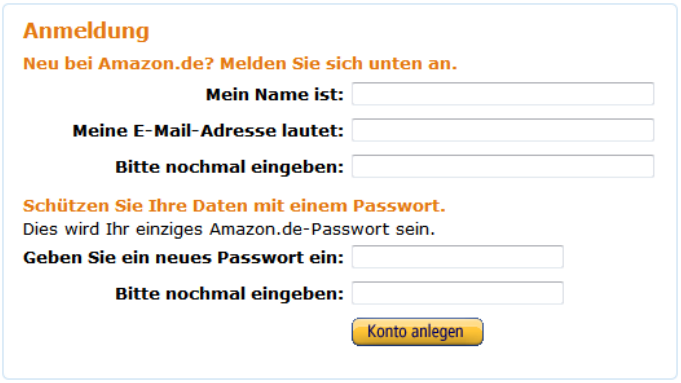


Abbildung 4.5: Optimierter Registrierungsvorgang bei Amazon.de

Erst im späteren Bestellvorgang wird der Benutzer gefragt, wohin er die Ware geliefert haben möchte und welche Zahlungsart er bevorzugt. Durch diese Vorgehensweise wird der Benutzer schnell und einfach durch die definierten Pfade geleitet.

4.2.2 Usability-Analyse

Unter Usability versteht man sowohl Benutzerfreundlichkeit als auch Gebrauchstauglichkeit. Diese Merkmale wiederum lassen sich durch die Eigenschaften Effektivität, Effizienz und Zufriedenstellung beschreiben. Ziele und deren Pfade sollten daher so gestaltet werden, dass sie unkompliziert und intuitiv vom Benutzer erreichbar sind, ohne dabei unnötig Fragen aufkommen zu lassen.

Während ein Entwickler normalerweise stets von seiner Aufgabengestaltung überzeugt ist, kommen aus Sicht eines Anwenders häufig Probleme und Fragen auf. Es ist für einen Entwickler nicht immer einfach, sich in die Lage des Anwenders zu versetzen, um Schwachstellen in der Usability identifizieren zu können. Entscheidende Unterstützung bei der Optimierung bieten daher die Usability-Analysen. Während es die klassische Usability-Analyse in einem Laborumfeld gibt, bei der Probanden das Produkt unter genauester technischer Beobachtung testen, existieren heute auch erste derartige Ansätze in der Web-Analytik.

Ähnlich wie bei Eyetrackern, die Augenbewegungen aufzeichnen, können in der Web-Analytik auch Heatmaps generiert werden, die jegliche Mausbewegungen oder -klicks eines bzw. aller Benutzer visualisieren. Die Rohdaten dafür können mittels JavaScript gesammelt werden. Gerade weil viele Menschen den Mauszeiger ihrem Blick folgen lassen, können aufgezeichnete Mausspuren dazu verwendet werden, aussagekräftige Heatmaps zu generieren, die eine Analyse des Benutzerverhaltens erlauben. Diese Methode kann Aufschluss über Mängel im Layout geben, denn wenn viele Besucher lange nach der gewünschten Information suchen müssen, dann ist dies auch ein deutliches Zeichen für schlechte Usability.

Abbildung 4.6: Mausspur-Visualisierung „Motion Map” (etracker)

Kapitel 5

Marktanalyse

Auf dem Markt existiert heute bereits eine Vielzahl an Web-Analytik-Tools für verschiedenste Anwendungsgebiete. Die ersten Tools dieser Art wurden bereits Mitte der 1990er Jahre eingeführt.

In diesem Kapitel wird eine Auswahl von bestehenden und etablierten Lösungen genauer beleuchtet, die aufgrund ihrer unterschiedlichen Anforderungen und Analysemöglichkeiten ausgewählt wurden.

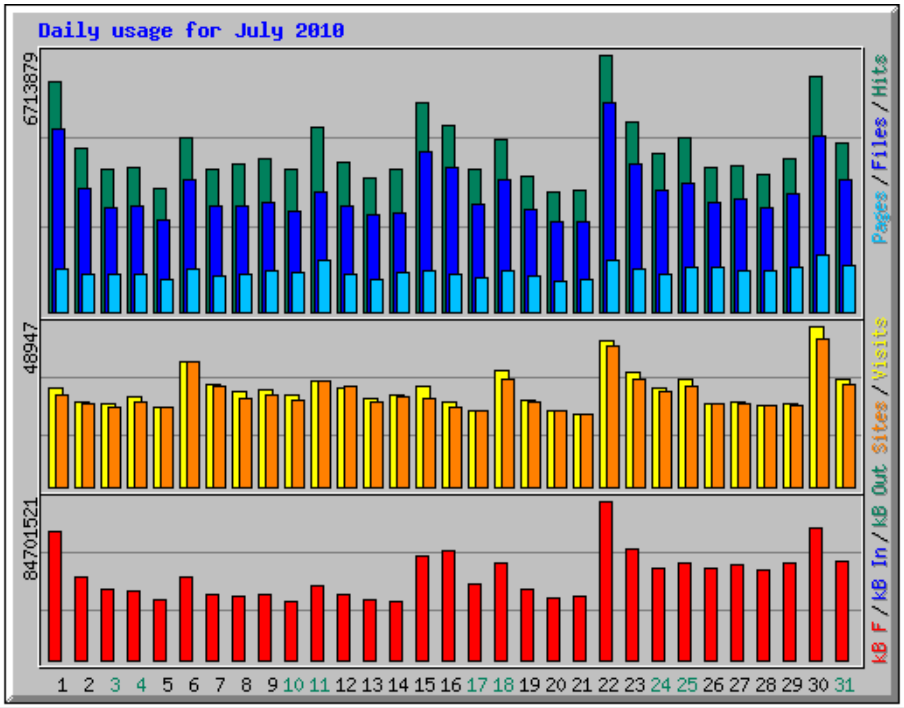
5.1 Webalizer

Webalizer ist eines der ältesten Web-Analytik-Werkzeuge. Es kann sowohl privat als auch kommerziell kostenlos eingesetzt werden. Der Autor Bradford L. Barrett begann mit der Entwicklung bereits im Jahre 1997 und führt das Open-Source-Projekt bis heute in unregelmäßigen Abständen fort. Das Tool setzt ausschließlich auf Logdateien und unterstützt sowohl das „Common Logfile Format" als auch das „Combined Logfile Format". Zusätzlich bietet es die Möglichkeit, Logdateien von FTP- und Proxy-Servern auszuwerten.

Das Tool wurde besonders im Hinblick auf statische Webseiten entwickelt und bot lange Zeit nur wenige Möglichkeiten, dynamisch generierte Webseiten korrekt zu erfassen. Da ferner nur Logdateien bei der Analyse erfasst werden, sind viele der heute wichtigsten Kennzahlen – wie z. B. die Page Views – mit Webalizer nicht erfassbar. Die Installation gestaltet sich sehr einfach, da Änderungen an den zu messenden Websites nicht erforderlich sind.

Da das Tool für jede zu erstellende oder zu aktualisierende Statistik erneut ausgeführt werden muss, bietet sich eine Automatisierung der Anwendungsstarts nach Zeitplan an. Die Resultate werden als statische HTML-Seiten und Diagramm-Grafikdateien ausgegeben und können mit einem Webbrowser betrachtet werden.

Webalizer wird auf der Website `http://www.webalizer.org/` angeboten.



Daily Statistics for July 2010

Day	Hits		Files		Pages		Visits		Sites		kB F		kB In		kB Out	
1	6002745	4.57%	4760334	4.84%	1098279	3.38%	29837	3.27%	28098	4.41%	68848922	4.87%	0	0.00%	0	0.00%
2	4278890	3.26%	3211366	3.27%	967914	2.98%	25752	2.82%	25218	3.95%	44316533	3.13%	0	0.00%	0	0.00%
3	3716764	2.83%	2731066	2.78%	978639	3.01%	25106	2.75%	24429	3.83%	37263758	2.64%	0	0.00%	0	0.00%

Abbildung 5.1: Typische Webalizer-Statistik (Ausschnitt)

5.2 AWStats

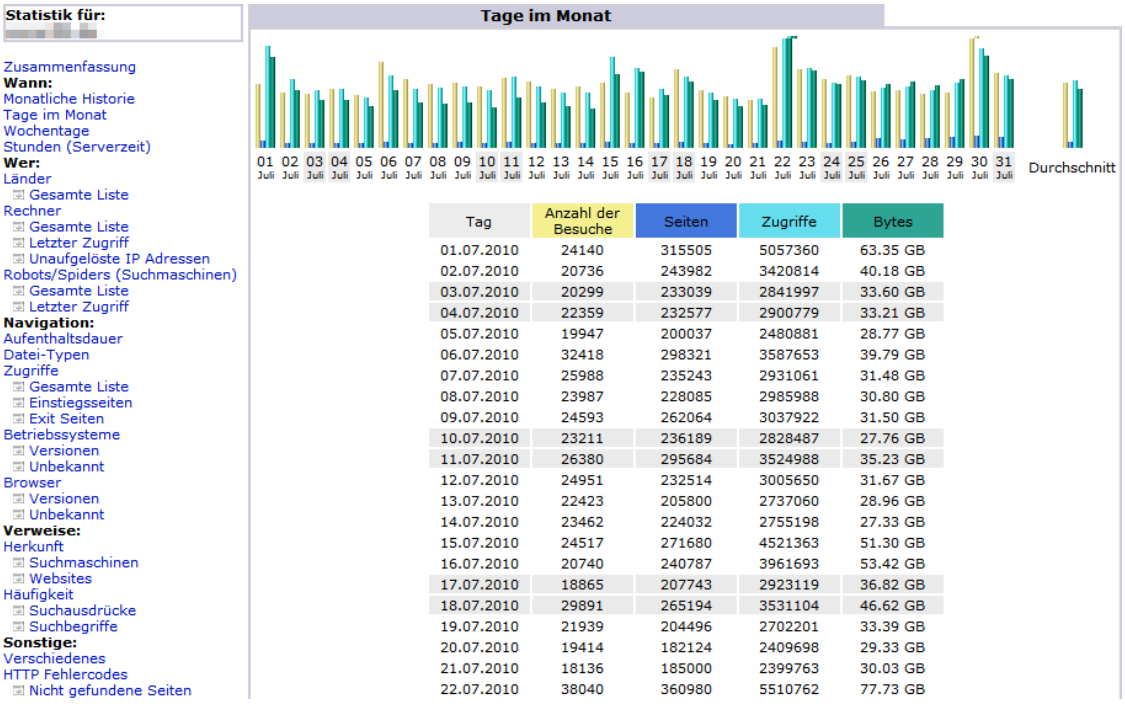
AWStats ist ein kostenloses Web-Analytik-Tool, das ebenfalls vor allem auf Logdatei-Analyse setzt. Mit der Entwicklung begann der Autor Laurent Destailleur in den frühen 2000er Jahren. Neue Versionen erscheinen durchschnittlich einmal im Jahr.

Das in der Programmiersprache Perl entwickelte Open-Source-Tool bietet sehr viele Konfigurationsmöglichkeiten und auch eine Erweiterungsschnittstelle für Plugins, die zusätzliche Daten bereitstellen können. Standardmäßig kann AWStats neben Webserver-Logdateien auch Mailserver- und FTP-Server-Logdateien verarbeiten.

Die Basis-Funktionalität erfordert während der Installation keinerlei Änderungen an den statistisch zu erfassenden Websites. Es ist jedoch optional möglich, einen JavaScript-Codeschnipsel in die Seiten einzubauen, mit dem weitere Daten gesammelt werden können, wie z. B. Versionsinformationen zu den beim Besucher installierten Webbrowser-Plugins (vgl. Kapitel 3).

Obwohl der Autor von AWStats das Tool als „real-time logfile analyzer" beschreibt, müssen Statistiken manuell oder nach Zeitplan generiert werden. Generierte Statistiken lassen sich in einer Monats- oder Jahresansicht im Webbrowser betrachten.

Die Website von AWStats ist unter `http://awstats.sourceforge.net/` zu erreichen.



Tag	Anzahl der Besuche	Seiten	Zugriffe	Bytes
01.07.2010	24140	315505	5057360	63.35 GB
02.07.2010	20736	243982	3420814	40.18 GB
03.07.2010	20299	233039	2841997	33.60 GB
04.07.2010	22359	232577	2900779	33.21 GB
05.07.2010	19947	200037	2480881	28.77 GB
06.07.2010	32418	298321	3587653	39.79 GB
07.07.2010	25988	235243	2931061	31.48 GB
08.07.2010	23987	228085	2985988	30.80 GB
09.07.2010	24593	262064	3037922	31.50 GB
10.07.2010	23211	236189	2828487	27.76 GB
11.07.2010	26380	295684	3524988	35.23 GB
12.07.2010	24951	232514	3005650	31.67 GB
13.07.2010	22423	205800	2737060	28.96 GB
14.07.2010	23462	224032	2755198	27.33 GB
15.07.2010	24517	271680	4521363	51.30 GB
16.07.2010	20740	240787	3961693	53.42 GB
17.07.2010	18865	207743	2923119	36.82 GB
18.07.2010	29891	265194	3531104	46.62 GB
19.07.2010	21939	204496	2702201	33.39 GB
20.07.2010	19414	182124	2409698	29.33 GB
21.07.2010	18136	185000	2399763	30.03 GB
22.07.2010	38040	360980	5510762	77.73 GB

Abbildung 5.2: Typische AWStats-Statistik (Ausschnitt)

5.3 Google Analytics

Das derzeit am häufigsten verwendete Web-Analytik-Tool ist Google Analytics. [W3T11]

Zu den Kunden zählen neben der Mindfactory AG und der CARLSEN Verlag GmbH zahlreiche weitere mittelständische Unternehmen.

Google Analytics wird kostenfrei vertrieben und besitzt einen sehr großen Funktionsumfang. Es können alle gängigen Besucher-Zugriffsdaten überblickt werden. Events, Funnel-Analysen und Kampagnen lassen sich ebenfalls detailliert definieren.

Eine Verbindung zu Google AdSense und Google AdWords ist möglich, sofern man diese Werbedienste in Anspruch nimmt. Google AdSense ermöglicht Betreibern Werbeanzeigen auf ihrer Website einzublenden und somit eine weitere Einnahmequelle zu sichern. Das Gegenstück dazu ist Google AdWords, das Betreibern ermöglicht, Werbung auf den Seiten von Google und den AdSense-Teilnehmern zu schalten.

Die Installation erfolgt über einen Codeschnipsel, der auf jeder Seite der Website eingebunden werden muss. Der Codeschnipsel von Google Analytics besteht ausschließlich aus einem JavaScript-Code ohne separatem Zählpixel. Eine Erfassung von Besuchern ohne aktivierter JavaScript-Unterstützung ist somit nicht möglich.

Alle gewonnenen Rohdaten werden auf den Servern von Google gespeichert. Statistiken werden automatisch täglich aktualisiert und lassen sich im Webbrowser betrachten. Eine „Radar"-Funktion verschickt automatisch Benachrichtigungen, falls zu einer beliebigen Kennzahl ein gewisser Schwellenwert unter- oder überschritten wird. Jede Detailseite bietet außerdem eine Exportfunktion in verschiedene Dateiformate, darunter auch CSV. Ferner wird eine Data Export API angeboten, mit der Kunden Zugriff auf die gesammelten Rohdaten erhalten können.

Google Analytics wird unter `http://www.google.com/analytics/` angeboten.

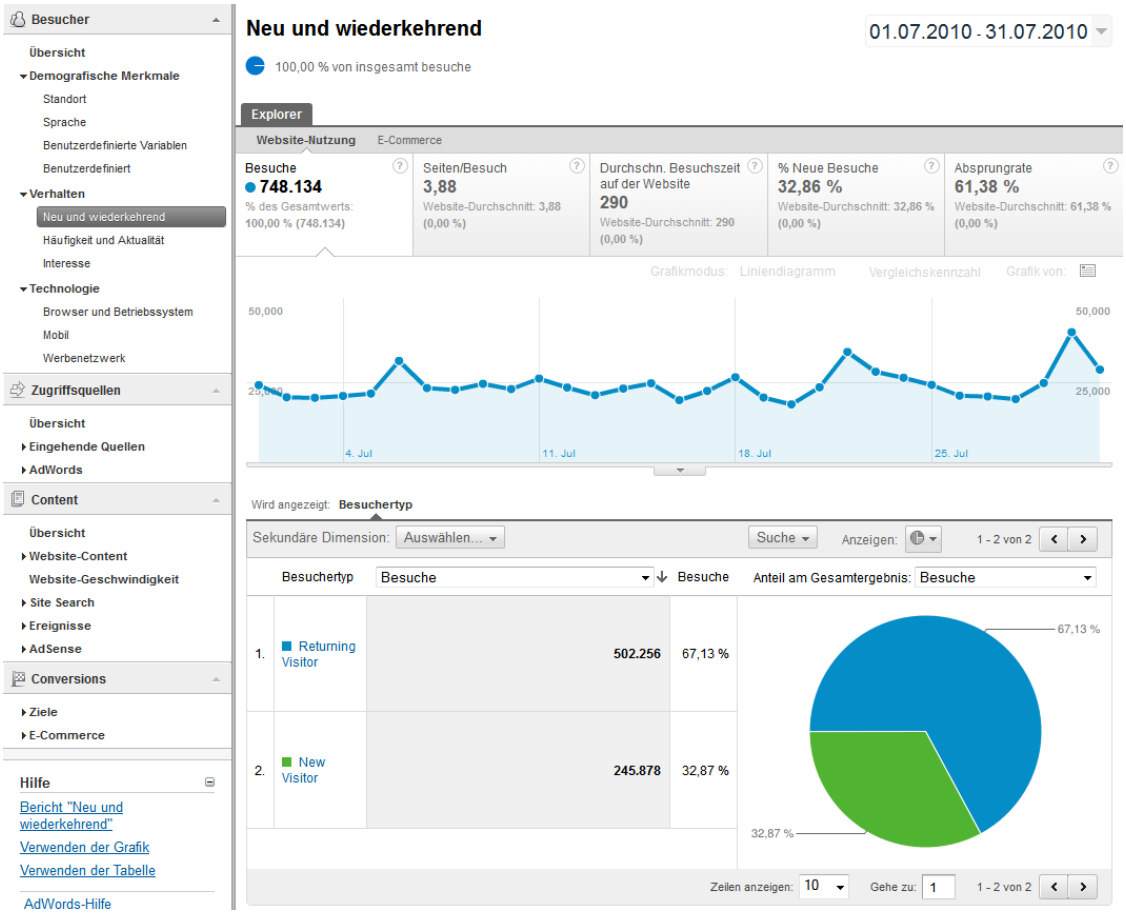


Abbildung 5.3: Typische Google-Analytics-Statistik (Ausschnitt)

5.4 piwik

piwik ist ein noch relativ junges Projekt eines größeren Teams von Entwicklern. Die erste Version 0.1 erschien am 6. März 2009. Laut offizieller Website ist es das Ziel von piwik, eine Open-Source-Alternative zu Google Analytics zu bieten.

Dieses Tool verwendet die Zählpixel- und JavaScript-Methode, um Rohdaten zu sammeln. Dadurch können zwar keine Hits gezählt und kein Traffic gemessen werden, aber es ergeben sich auch viele Vorteile, die die klassische Logdatei-Web-Analytik nicht bieten kann. piwik ermöglicht somit u. a. auch die Sammlung marketingrelevanter Daten, eine Kampagnen-Erfolgskontrolle und die Erkennung wiederkehrender Besucher.

Funnel-Analysen werden – bis einschließlich Version 1.4 – nicht standardmäßig unterstützt. Unter `http://dev.piwik.org/trac/ticket/220` (abgerufen am 02.05.2011) finden sich jedoch Informationen zu einem Plugin, das diese Funktionalität nachrüsten kann.

Die Installation von piwik ist, anders als bei den Web-Analytik-Tools, die auf Logdateien setzen, auch ohne administrativen Zugriff auf den Server möglich. Zudem finden Analyse und Statistikgenerierung in Echtzeit statt, sodass man zu jedem Zeitpunkt aktuellste Daten im Webbrowser betrachten kann. Dabei ist es auch möglich, den Besucherfluss live zu beobachten.

piwik erfordert für die Installation die Unterstützung der Skriptsprache PHP im Webserver, sowie eine SQL-Datenbank, in der die gesammelten Daten gespeichert werden. Während der Installation muss zudem zwingend ein Codeschnipsel auf die zu untersuchende Website platziert werden.

Die Projektseite von piwik ist unter `http://de.piwik.org/` zu erreichen.

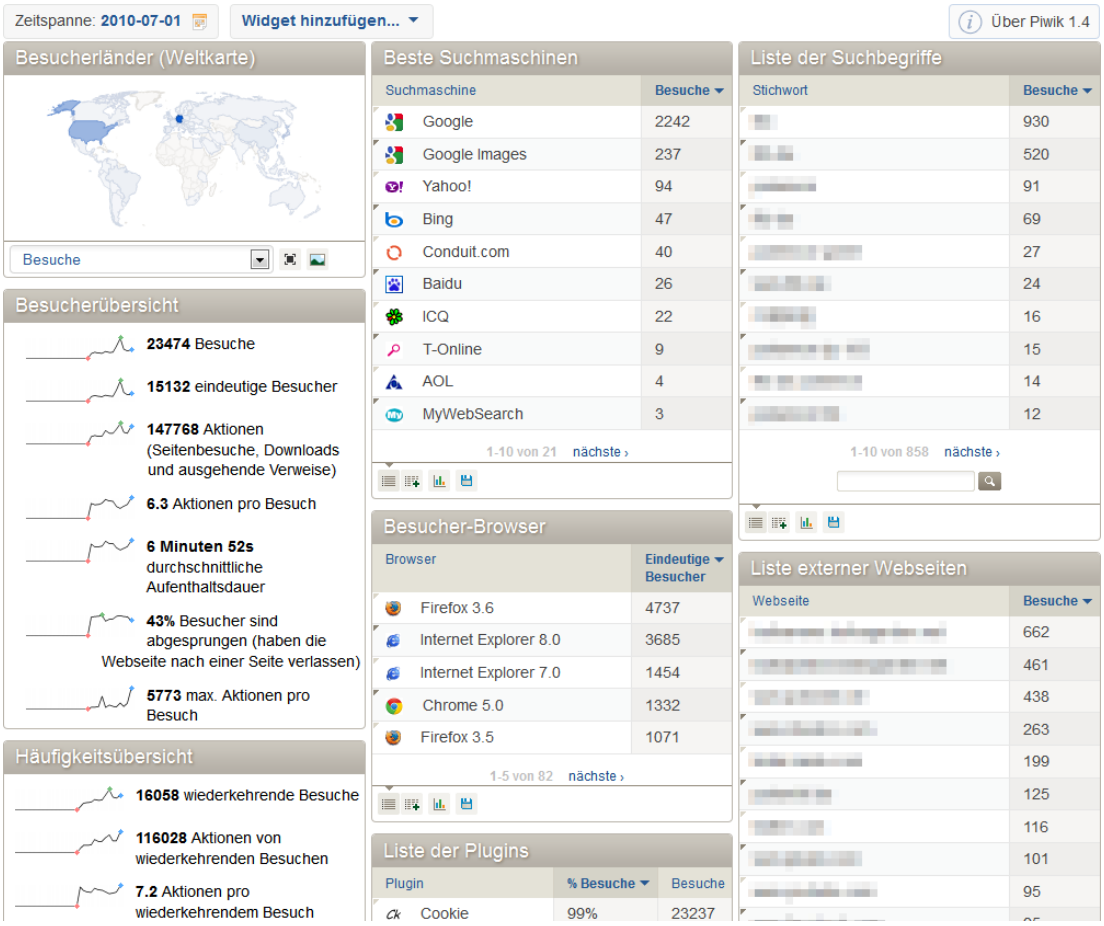


Abbildung 5.4: Typische piwik-Statistik (Ausschnitt)

5.5 etracker Web Analytics

Das kommerzielle Produkt „etracker Web Analytics” richtet sich vor allem an den Business-Bereich. Neben einer zeitlich befristeten Demolizenz werden nur Lizenzverträge angeboten, bei denen sich der Funktionsumfang und Preis nach der Anzahl der Besucher bzw. Page Views richtet. Zu den Kunden von etracker zählen größere Unternehmen, darunter Deutsche Bahn AG, Axel Springer AG, RTL2 Fernsehen GmbH & Co. KG und Toys”R”Us GmbH.

Bei etracker Web Analytics werden die Rohdaten über einen JavaScript- und Zählpixel-Codeschnipsel gesammelt, der in die zu messende Website integriert werden muss. Die

gesammelten Rohdaten werden nicht lokal auf dem eigenen Server, sondern entfernt auf den Servern der etracker GmbH gespeichert.

In den von etracker bereitgestellten Statistiken finden sich neben den generell üblichen Daten, wie Besucher-Zugriffszahlen, auch sehr umfangreiche Informationen zum Besucherverhalten. Dazu zählen u. a. die Auswertung von Klicks und Mausbewegungen. Zudem sind Funnel-Analysen, Warenkorb-Analysen (Data Mining), Event-Tracking, eine Kampagnen-Erfolgskontrolle, sowie Vorher-Nachher-Vergleiche im Rahmen von Website-Optimierungen möglich.

Statistiken können im Webbrowser betrachtet und dort einmalig oder regelmäßig automatisch als PDF-, CSV-, XML- oder Excel-Datei exportiert werden.

Die Website des Herstellers ist unter `http://www.etracker.com/` zu erreichen.

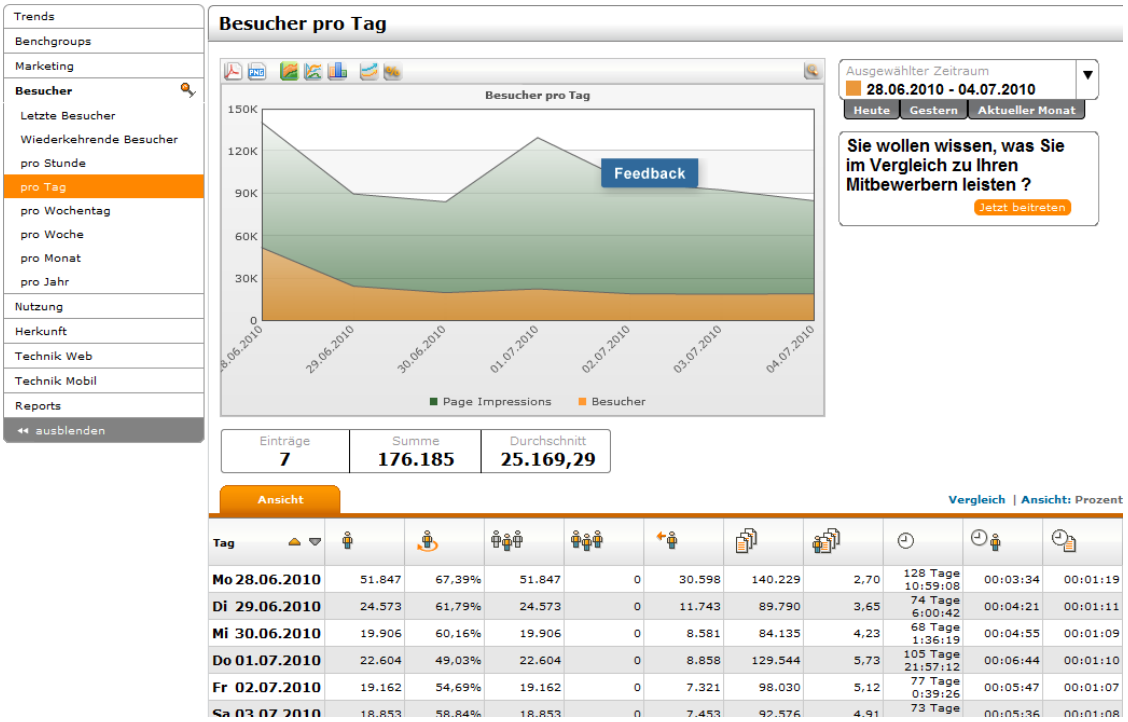


Abbildung 5.5: Typische etracker-Statistik (Ausschnitt)

5.6 Vergleich

	Webalizer	AWStats	Google	piwik	etracker
Mögliche Verfahren der Rohdatengewinnung	Logdatei	Logdatei, Zählpixel, JavaScript	JavaScript	Zählpixel, JavaScript	Zählpixel, JavaScript
Änderungen an der Website erforderlich	Nein	Optional	Ja	Ja	Ja
Standort gesammelter Daten	Eigener Server	Eigener Server	Dienstleister	Eigener Server	Dienstleister
JavaScript beim Besucher notwendig	Nein	Nein	Ja	Nein	Nein
Unterstützte Export-Dateiformate	TSV	PDF	CSV, TSV, XML, PDF	CSV, TSV, XML, PHP, JSON	CSV, TSV, XML, PDF
Auswertung der Besucher-Zugriffszahlen	Ja	Ja	Ja	Ja	Ja
Durchführung von Funnel-Analysen	Nein	Nein	Ja	Plugin	Ja
Unterstützung von Event-Tracking	Nein	Nein	Ja	Nein	Ja
Untersuchung von Kampagnen	Nein	Nein	Ja	Ja	Ja
Durchführung von Usability-Analysen	Nein	Nein	Nein	Nein	Ja
Kosten	Kostenlos	Kostenlos	Kostenlos	Kostenlos	Monatlich

Tabelle 5.1: Die vorgestellten Tools im Vergleich

Während der Untersuchung dieser Auswahl an marktüblichen Tools stellte sich heraus, dass jedes Tool durchaus seine eigene Existenzberechtigung hat.

Je nachdem, welche Ziele der Betreiber einer Website verfolgt, ergeben sich, wie in Tabelle 5.1 zu sehen, für jedes Tool individuelle Vor- und Nachteile, die er untersuchen sollte, bevor er sich für eine oder mehrere Lösungen entscheidet.

Kapitel 6

Prototypischer Entwurf

In diesem zentralen Teil der Arbeit wird ein eigenes Web-Analytik-Tool konzipiert und prototypisch in der Skriptsprache PHP implementiert.

Ziel der Implementierung ist es, möglichst viele der angesprochenen Verfahren und Methoden theoretisch und praktisch umzusetzen und so einen Einblick in die Funktionsweise der vorgestellten Tools zu erhalten.

Alle drei vorgestellten Verfahren der Rohdatenerfassung sollen im Prototypen Einsatz finden: Logdatei, Zählpixel und JavaScript. Die gesammelten Daten sollen sich dabei sinnvoll ergänzen, um eine starke Basis für die darauf folgende Analyse und Synthese zu bilden.

Zudem soll es möglich sein, aktuelle Statistiken zu generieren, die sowohl eine tägliche und monatliche als auch eine jährliche Betrachtung erlauben.

6.1 Datenbank-Design

Um im Rahmen einer Synthese später Statistiken generieren zu können, müssen die per Logdatei, Zählpixel und JavaScript gesammelten Rohdaten passend abgespeichert werden. Für dieses Problem sind verschiedene Ansätze denkbar, wie z. B. ein proprietäres Binärformat oder einzelne Textdateien. Am ehesten bietet sich für einen Datenbestand dieser Art jedoch die Speicherung in einem Datenbanksystem an. Datenbanksysteme können große Datenmengen effizient speichern und gleichzeitig einen schnellen und direkten Zugriff auf diese ermöglichen. Gestützt durch ein Transaktionssystem wird zudem selbst im Fehlerfall die Datenkonsistenz bewahrt.

Der zu implementierende Prototyp setzt daher auf eine relationale Datenbank. Als Datenbankmanagementsystem (DBMS) kommt der MySQL-Server zum Einsatz, da sich dieser einfach in eine PHP-Umgebung integrieren lässt und keine Lizenzgebühren fällig werden.

Eine Datenbank erfordert stets ein an das Aufgabengebiet angepasstes Design. Folgende Ziele sollen durch das Datenbankmodell erfüllt werden:

- Effiziente Datenspeicherung
- Redundanzfreiheit
- Einfache Wartung

Passend dazu wurden mehrere Ansätze evaluiert, die nachfolgend vorgestellt werden.

Im ersten Ansatz war es geplant, lediglich zwei Tabellen zu verwenden. Die erste war für die Visitors mit folgenden Attributen vorgesehen: IP-Adresse, User-Agent, Sprache, usw. Die zweite Tabelle hätte per Fremdschlüssel auf den Primärschlüssel der ersten Tabelle referenziert und pro Seitenaufruf eines Visitors einen Datensatz mit weiteren Informationen (Seitentitel, usw.) gespeichert. Dieser Ansatz hat jedoch entscheidende Nachteile. Der Speicherbedarf ist sehr hoch und auch das Generieren einer anschaulichen Statistik über einen größeren Zeitraum ist durch ein schlechtes Laufzeitverhalten für einen praktischen Einsatz ungeeignet.

In einem weiteren Ansatz wurde die Funktionsweise des Open-Source-Tools „piwik“ in der Version 1.0 untersucht. Hier wird für jeden Monat eine eigene Archiv-Tabelle angelegt, in der sich für jeden Tag eine Zusammenfassung befindet. In einer weiteren, statischen Tabelle werden alle Visits seit Beginn der Aufzeichnungen gespeichert. Diese Tabelle enthielt in der Evaluierungsphase innerhalb von wenigen Wochen mehrere Millionen Datensätze. Die allgemeine Performance und Funktionstüchtigkeit von piwik wurde durch diese Datenmenge jedoch stark beeinträchtigt, da der für die Statistikgenerierung zur Verfügung gestellte Arbeitsspeicher nicht mehr ausreichte.

Da sich diese Ansätze nicht als optimal herausstellten, wurde im nächsten Schritt der so genannte Data-Warehouse-Ansatz genauer beleuchtet.

6.1.1 Data Warehouse

Ein Data Warehouse ist eine Datenbank, die Informationen aus verschiedenen Quellen so speichert, dass sie späteren Analyse- und Syntheseprozessen geeignet zur Verfügung stehen. [Ger98]

Die in einem Data Warehouse gespeicherten Daten werden dabei in mehreren Dimensionen gespeichert. Jede Dimension lässt sich mittels so genannter Aggregationsebenen genauer beschreiben. Diese Aggregationsebenen können hierarchisch definiert werden, wodurch das Data Warehouse ganz besondere Eigenschaften gewinnt: Abstraktion und Spezialisierung. Der Vorgang der Abstraktion wird auch Roll-up genannt und der umgekehrte Vorgang, die Spezialisierung, trägt die Bezeichnung Drill-down.

Abbildung 6.1: Hierarchische Aggregationsebenen der Beispiel-Dimension Ort

In einem Data Warehouse werden vielleicht Ortsdaten mit Stadtteil-Genauigkeit gespeichert. Hier ermöglichen Roll-up und Drill-down, je nachdem welche Hierarchie-Ebene betrachtet wird, verschiedenartige Statistiken auf Stadtteil-, Stadt-, Bundesland- oder Landesebene (vgl. Abbildung 6.1).

Jede Aggregationsebene wird mit Dimensionswerten konkretisiert. Unter Dimensionswerten versteht man die eigentlichen Werte, wie z. B. die Stadt *Frankfurt*. Letztlich werden diesen Dimensionswerten nun noch Faktendaten zugeordnet. Diese Faktendaten können eine simple Gruppierung der Dimensionswerte darstellen, aber auch weitere Kennzahlen beinhalten.

6.1.1.1 Stern-Schema

Das so genannte Stern-Schema ist eines der am häufigsten genutzten Modelle für Data-Warehouse-Systeme (vgl. [Ger98]).

Aus jeder Dimension entsteht im Stern-Schema eine eigene Tabelle, die die jeweiligen Aggregationsebenen und Dimensionswerte beinhaltet. In der zentralen Faktentabelle werden diese Dimensionstabellen über die Spalten per 1:n-Fremdschlüsselreferenz verknüpft. Ferner bilden diese Fremdschlüssel gemeinsam den Primärschlüssel der Faktentabelle.

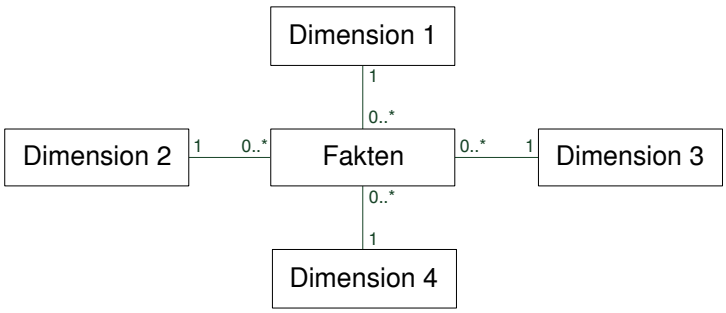
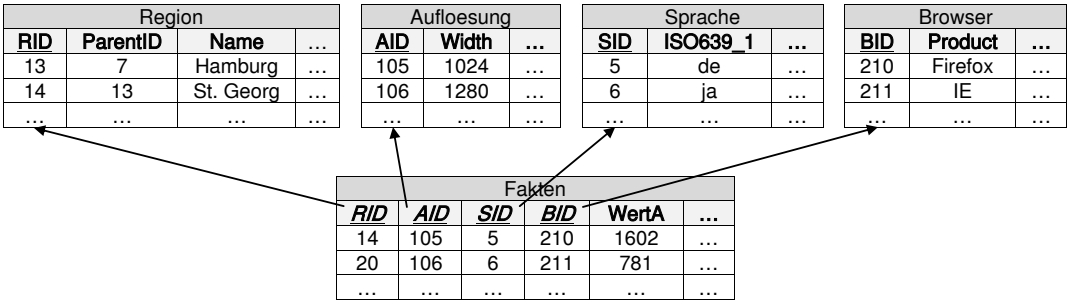


Abbildung 6.2: Stuktur des Stern-Schema-Modells

Ebenfalls finden sich in der Faktentabelle Kennzahlen, die in Zusammenhang mit den Referenzen auf die Dimensionswerte eine konkrete Aussage treffen.



Region

RID	ParentID	Name	…
13	7	Hamburg	…
14	13	St. Georg	…
…	…	…	…

Aufloesung

AID	Width	…
105	1024	…
106	1280	…
…	…	…

Sprache

SID	ISO639_1	…
5	de	…
6	ja	…
…	…	…

Browser

BID	Product	…
210	Firefox	…
211	IE	…
…	…	…

Fakten

RID	AID	SID	BID	WertA	…
14	105	5	210	1602	…
20	106	6	211	781	…
…	…	…	…	…	…

Abbildung 6.3: Beispiel-Tabellenstruktur

Ein Beispiel einer Stern-Schema-Tabellenstruktur ist in Abbildung 6.3 dargestellt. In der Tabelle zur Dimension *Region* existiert zwischen den Datensätzen zu Hamburg und St. Georg eine Hierarchie. Mit dem Attribut *ParentID* wird festgelegt, dass St. Georg ein Stadtteil von Hamburg ist, indem auf die dazugehörige *RID* verwiesen wird.

6.1.1.2 Generisches Modell

Das aus dem Stern-Schema entstandene Generische Modell setzt auf eine ganz bestimmte Tabellenstruktur, bei der es, entsprechend den weiter oben beschriebenen Eigenschaften, genau vier Tabellen gibt:

- Dimensionen
- Aggregationsebenen
- Dimensionswerte
- Faktenwerte

Die zentrale Idee ist es, die Faktentabelle aus dem in Abschnitt 6.1.1.1 beschriebenen Stern-Schema zu transponieren, wodurch sich der Vorteil einer variablen Anzahl an Attributen ergibt. Damit zusammenhängende Daten nicht auseinanderreißen, werden durchnummerierte Gruppen-Indizes verwendet.

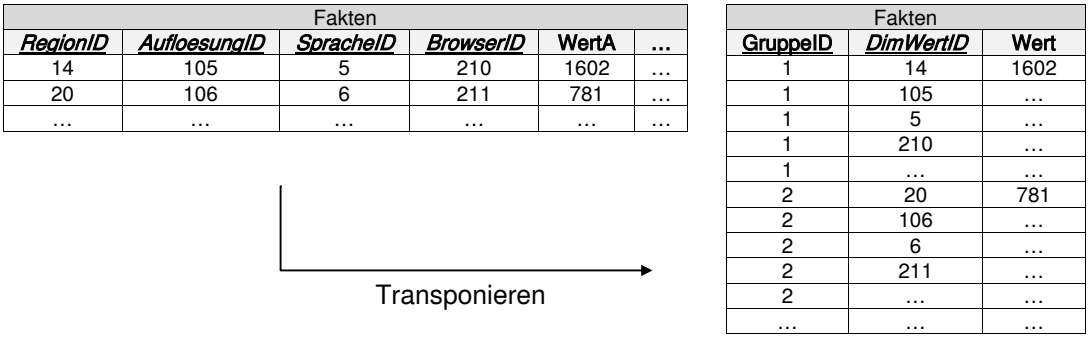


Abbildung 6.4: Transponierte Faktentabelle

Weitere Spalten in der Faktentabelle umfassen den Verweis auf die Dimensionswerte-Tabelle sowie den damit verbundenen Faktenwert, der jedoch optional ist und auch leer sein bzw. mit `NULL` gefüllt werden kann.

Auch die drei anderen Tabellen besitzen eine Struktur mit einem festen und unveränderlichen Spaltenschema, wie nachfolgend in Abbildung 6.5 dargestellt.

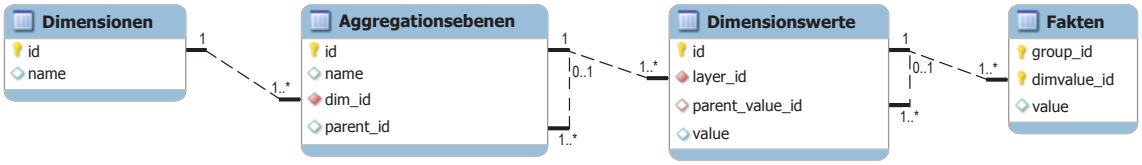


Abbildung 6.5: Stuktur des Generischen Modells

Die Hierarchie wird, ähnlich wie beim Stern-Schema, über die Parent-IDs in den Aggregationsebenen- und Dimensionswerte-Tabellen realisiert. Diese Parent-IDs stellen dabei Zeiger auf den jeweiligen Vorgängerwert dar.

6.1.1.3 Bewertung

In der Web-Analytik sind vor allem summierte Werte über einen definierten Zeitraum von Bedeutung. Die Struktur der Faktentabelle im Stern-Schema stellt hierfür jedoch ein großes Hindernis dar. Für jede neue Dimension ist zusätzlich auch ein Summen-Attribut notwendig. Im Hinblick auf nachträgliche Änderungen in den Dimensionen und Aggregationsebenen ist das Stern-Schema somit nicht ausreichend flexibel, denn für jede neue Dimension muss eine neue Tabelle angelegt werden. In der zentralen Faktentabelle wäre zudem je eine weitere Spalte für jede neue Dimension und jedes neue Fakten-Attribut (z. B. für Summen) notwendig. Eine solche Strukturänderung stellt bei vorhandenen großen Datenmengen eine sehr teure Operation dar.

Obwohl das Generische Modell auf den ersten Blick unstrukturierter wirken kann, erfüllt es alle definierten Ziele und Anforderungen. Summen für jeden gesammelten Dimensionswert können effizient in der Faktentabelle im *value*-Attribut gespeichert werden. Beim Einfügen neuer und neuartiger Daten sind keine Strukturänderungen, sondern ausschließlich Einfügeoperationen erforderlich. Da zudem keine ungewollten Redundanzen entstehen können, ist dieser Ansatz besonders speichereffizient. Es ist möglich, eine variable Anzahl von Fakten zu pflegen und nachträglich neue Dimensions-Hierarchien einzurichten, ohne das laufende System zu stören.

Für den prototypischen Entwurf wurde daher das Generische Modell gewählt.

6.2 Implementierung

Der konkrete Entwurf und die Implementierung des Prototypen, der nachfolgend den Namen *dvdstats* tragen wird, werden in diesem Abschnitt beschrieben.

6.2.1 Datenspeicherung

Obwohl alle vier Tabellen des Generischen Modells grundsätzlich jederzeit vollständig erweiterbar sind, wurden die Tabellen Dimensionen und Aggregationsebenen für dvdstats bereits in der Designphase festgelegt und werden während der Laufzeit auch nicht mehr verändert.

Die Tabelle Dimensionen beschreibt sehr abstrakt die Art der Daten. Im Rahmen der Implementierung wurden folgende Dimensionen eingerichtet:

- date (Datumsangaben)
- location (Besucherherkunft)
- useragent (Client-Informationen)
- display (Bildschirm-Informationen)
- language (Sprache)
- referer (Verweisende Webseite)
- site (Angefordertes Objekt)
- figures (Summierte Kennzahlen)

Diese Dimensionen stellen grobe Kategorien für die Aggregationsebenen dar, die zu einem großen Teil den wichtigsten statistischen Daten entsprechen. Neben diesen Daten werden zusätzlich auch die Kennzahlen Visits und Traffic mit aufgenommen (vgl. Kapitel 4.1).

In der nachfolgenden Abbildung 6.6 sind die Tabellen *dwh_dimensions* (Dimensionen) und *dwh_agglayers* (Aggregationsebenen) von dvdstats dargestellt.

dwh_dimensions	
id	**name**
1	date
2	location
3	useragent
4	display
5	language
6	referer
7	site
8	figures

dwh_agglayers			
id	**name**	***dim_id***	***parent_id***
1	year	1	*NULL*
2	month	1	1
3	day	1	2
4	country	2	*NULL*
5	city	2	4
6	latitude	2	*NULL*
7	longitude	2	*NULL*
8	browser	3	*NULL*
9	os	3	*NULL*
10	resolution	4	*NULL*
11	colordepth	4	*NULL*
12	laguage	5	*NULL*
13	referer	6	*NULL*
14	url	7	*NULL*
15	visits	8	*NULL*
16	traffic	8	*NULL*

Abbildung 6.6: Definierte Dimensionen und Aggregationsebenen

In die dritte Tabelle *dwh_dimvalues* (Dimensionswerte) werden dynamisch zur Laufzeit alle in den Rohdaten gefundenen und in den Aggregationsebenen festgelegten Eigenschaften eingetragen. Referenzen auf die Tabelle *dwh_agglayers* und Hierarchien werden dabei ebenfalls automatisch aufgebaut. Als Dimensionswerte gelten alle benutzerbezogenen Informationen, darunter Bildschirmauflösungen, Spracheinstellungen und Betriebssystem-Namen. Jeder dieser Dimensionswerte kommt dabei nur ein einziges Mal in der Tabelle vor, wodurch sie somit nur um neuartige, zuvor unbekannte Werte ergänzt wird.

Während der Implementierung stellte sich heraus, dass viele sequenziell ausgeführte `INSERT`- bzw. `UPDATE`-Einzeloperationen, wie sie in den Tabellen *Dimensionswerte* und *Faktenwerte* bei einem Analysedurchlauf zwangsläufig vorkommen, die Performance stark beeinträchtigen können, da ein Transaktionsabschluss (`COMMIT`) im Schnitt die meiste Zeit in Anspruch nimmt. Aus diesem Grunde wurde eine Warteschlange implementiert, die eine bestimmte Anzahl von `INSERT`- und `UPDATE`-Befehlen in eine Liste aufnimmt. Erst wenn die Warteschlange vollständig gefüllt ist bzw. keine weiteren Abfragen mehr folgen, wird eine einzelne Transaktion mit den gesammelten Befehlen im Datenbanksystem zur Ausführung gebracht.

dwh_dimvalues			
id	*layer_id*	*parent_value_id*	value
1	8	*NULL*	Firefox 3.6
2	9	*NULL*	Win7
3	12	*NULL*	ja
4	4	*NULL*	Japan
5	5	4	Tokyo
6	6	5	35.685
7	7	5	139.7514
8	10	*NULL*	1920x1200
9	11	*NULL*	32
10	13	*NULL*	http://www....
…	…	…	…
1183	3	NULL	2010-09-12
…	…	…	…
…	…	…	…
…	…	…	…

dwh_facts		
group_id	*dimvalue_id*	value
0	1	76
0	2	53
0	3	8
0	4	9
0	5	2
0	8	3
0	9	63
0	10	1
0	11	1
0	12	227
…	…	…
0	1183	*NULL*
1	1	101
1	2	68
…	…	…

Abbildung 6.7: Aus Beispieldaten gewonnene Dimensions- und Faktenwerte

Die vierte Tabelle *dwh_facts* (Faktenwerte) speichert im *value*-Attribut die Einzelsummen der gesammelten Dimensionswerte. Eine Referenz auf die Tabelle *dwh_dimvalues* befindet sich im Attribut *dimvalue_id*. Gruppiert werden die Informationen nach Datum über das *group_id*-Attribut. Der Datumswert selbst stellt ebenfalls einen Dimensionswert dar und ist in Abbildung 6.7 z. B. unter der *dimvalue_id* `1183` zu sehen.

Mit diesen vier Tabellen des Generischen Modells kann eine effiziente Endspeicherung der Daten erreicht werden. Dennoch müssen während eines jeden Besucherzugriffs die damit verbundenen Rohdaten aus Zählpixel und JavaScript auch in Echtzeit abgespeichert werden können. Für diese Aufgabe gibt es in der dvdstats-Implementierung noch eine fünfte Tabelle *access*. In diese Tabelle werden temporär alle Informationen geschrieben, die während eines Seitenaufrufs anfallen. Diese Informationen werden über ein serverseitiges Skript, das Daten des JavaScript-gestützten Zählpixels erhält, verarbeitet und in diese fünfte Datenbanktabelle gespeichert. Passend zur Data-Warehouse-Definition stammen die Rohdaten somit aus zwei unterschiedlichen Quellen: Webserver-Logdatei und *access*-Tabelle.

In der sechsten Tabelle *status* befinden sich schließlich Dateiname und weitere Informationen zur zuletzt genutzten Webserver-Logdatei. Unter anderem wird hier die Anzahl der zuvor bereits bearbeiteten Zeichen bzw. Bytes der Webserver-Logdatei gespeichert, wodurch dvdstats beim nächsten Durchlauf an derselben Stelle fortfahren kann, an der sich beim letzten Durchlauf noch das Ende der Logdatei befand.

status	
key	**value**
lastlog_filename	E:\acc.log
lastlog_filesize	10240
lastlog_id	75fa8…
lastlog_position	8192
lastrun	2011-0…

access						
id	**access**	**ip**	**useragent**	**referer**	**lang…**	**…**
1	2011-0…	80.134.119.x	Mozilla/4.0 (compatible; MSIE 7.…	http://www.f…	de	…
2	2011-0…	74.173.243.x	Mozilla/5.0 (Windows; U; Windows…		en	…
3	2011-0…	87.173.208.x	Mozilla/5.0 (Windows; U; Windows…	http://www.f…	de	…
4	2011-0…	92.73.194.x	Mozilla/5.0 (Windows; U; Windows…	http://www.f…	de	…
5	2011-0…	206.212.227.x	Mozilla/4.0 (compatible; MSIE 8.…	http://www.c…	en	…
6	2011-0…	87.3.83.x	Mozilla/5.0 (Windows; U; Windows…	http://www.f…	it	…

Abbildung 6.8: Zusätzliche Tabellen *status* und *access*

Mit jedem Analysedurchlauf werden die Rohdaten aus Logdatei und Datenbanktabelle *access* optimiert in die vier Tabellen des Generischen Modells übernommen. Zu den zeilenweise aus der Logdatei gelesenen Rohdaten wird jeweils geprüft, ob die Datenbanktabelle *access* weitere Informationen bereitstellen kann. Die Zuordnung der beiden Rohdatensätze erfolgt über die IP-Adressen im Rahmen einer Visits-Analyse (vgl. Kapitel 4.1). Im Anschluss daran werden die gesammelten, speicherintensiven Rohdaten aus der Tabelle *access* nicht mehr benötigt. Sie werden daher gelöscht, um Speicherplatz wieder freizugeben. Die Webserver-Logdatei bleibt hingegen stets unberührt und kann zudem, aufgrund der Informationen der *status*-Tabelle, jederzeit problemlos durch eine frische Logdatei ausgetauscht werden, beispielsweise durch das Linux-Tool *logrotate*.

6.2.2 Programmstruktur

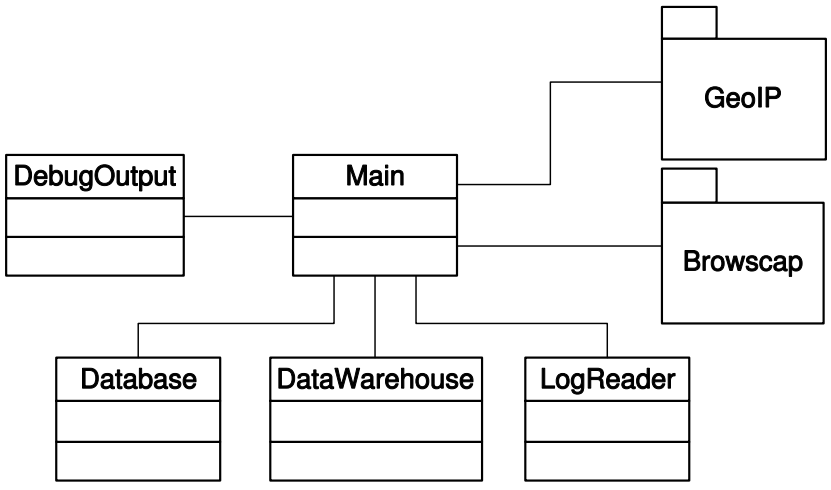


Abbildung 6.9: Klassenstruktur von dvdstats

Der PHP-Quellcode von dvdstats besteht aus den fünf Hauptklassen *Main*, *Database*, *DataWarehouse*, *LogReader* und *DebugOutput*. Das Zusatzpaket *GeoIP* stammt von der Firma MaxMind und ist unter `http://www.maxmind.com/app/geolitecity` kostenlos verfügbar. Das *Browscap*-Paket von Jonathan Stoppani wird auf der Seite `https://github.com/garetjax/phpbrowscap` angeboten. (Beide Websites wurden zuletzt am 19.05.2011 abgerufen.)

Database Diese Klasse regelt den Verbindungsaufbau zur MySQL-Datenbank und stellt Funktionen für den Zugriff auf diese bereit.

DataWarehouse Über diese Klasse wird das Generische Modell aufgebaut. Dazu gehört sowohl das Einfügen neuer Dimensions- und Faktenwerte, als auch das Pflegen und Aktualisieren vorhandener Daten.

LogReader Diese Klasse stellt Funktionen für das zeilenweise Lesen von Webserver-Logdateien im „Combined Logfile Format“ bereit.

DebugOutput Mit Hilfe dieser Klasse, deren Implementierung dem Singleton-Entwurfsmuster entspricht, können Fehlermeldungen, Warnungen und Testausgaben an die Konsole ausgegeben werden bzw. in eine Debug-Logdatei geschrieben werden.

GeoIP Dieses Zusatzpaket stellt die Auflösung von IPv4-Adressen in Geodaten, darunter Länder und Städte sowie deren Längen- und Breitengrade, zur Verfügung. Das Paket gibt ein Objekt mit den entsprechenden Attributen zurück.

Browscap Mit diesem Zusatzpaket können die einzelnen semantischen Teile aus User-Agent-Strings gelesen werden. Zu den daraus gewonnen Daten zählen vor allem Betriebssystem und Browser zzgl. der dazu gehörigen Versionsnummern. Eine Sammlung dieser Attribute wird in Form eines assoziativen Arrays zurückgegeben.

Main Die zentrale Hauptklasse instanziiert aus den oben genannten Klassen je ein Objekt und koordiniert den Analysevorgang mit Hilfe der bereitgestellten Funktionen.

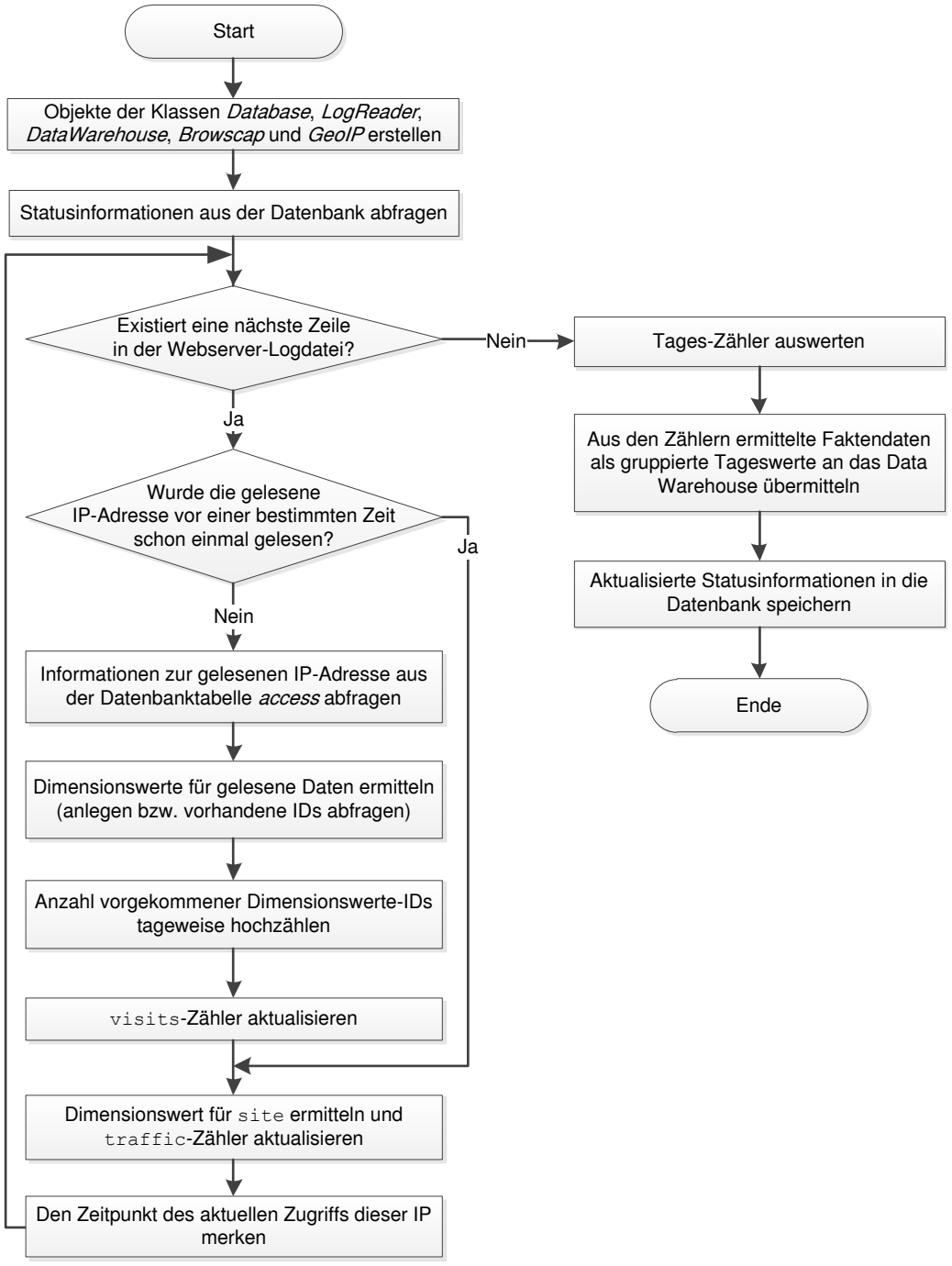


Abbildung 6.10: Programmablaufplan der *Main*-Klasse

Zusätzlich zu diesen Klassen existiert auch ein Programmstart-Skript, das manuell bzw. von der *Aufgabenplanung* (Windows) oder *crontab* (Unix) aufgerufen werden kann. Dieses Skript erstellt ein Objekt der Klasse *Main* und leitet den Analysevorgang ein. Zusätzlich misst es die Laufzeit und sorgt dafür, dass nur eine Instanz zur selben Zeit ausgeführt werden kann.

Sobald der Analysevorgang abgeschlossen ist, kann eine separat entwickelte Web-Applikation die Daten aus dem Data Warehouse auslesen, um daraus eine Statistik zu generieren.

Um beispielsweise eine Liste der 25 am häufigsten vorkommenden Referer zu generieren, wird zunächst die ID der dazu passenden Aggregationsebene abgefragt. Anschließend können mit Hilfe dieser ID alle dazugehörigen Dimensionswerte ermittelt werden, da die Daten der Dimensionswerte-Tabelle jeweils eine Referenz auf die Aggregationsebene führen. Zum Schluss wird die Faktentabelle mit Hilfe der gesammelten Dimensionswerte-IDs durchsucht, um die Anzahl der Vorkommnisse zu erhalten.

Während des Durchsuchens der Faktentabelle kann ein assoziatives Array aufgebaut werden. Als Schlüssel würde sich dabei der Dimensionswert selbst anbieten, also der Referer-String. Im dazugehörigen Array-Wert kann dessen Anzahl eingetragen werden. Es sollte sichergestellt werden, dass die ersten 25 gefundenen Einträge auch den am häufigsten vorkommenden entsprechen, damit der Vorgang an dieser Stelle aus Performance-Gründen kontrolliert abgebrochen werden kann. Um dies zu erreichen, muss die Faktentabelle in absteigender Reihenfolge der Werte vorsortiert werden, z. B. mit folgender SQL-Abfrage:

```
SELECT * FROM `dwh_facts` WHERE `group_id`='1183' ORDER BY `value` DESC;
```

6.3 Einrichtung

Der Prototyp dvdstats benötigt eine Webserver-Software mit PHP-Unterstützung und einer Logdatei im „Combined Logfile Format“, sowie eine MySQL-Datenbank. Für eine Testumgebung bietet sich das XAMPP-Paket der Website `http://www.apachefriends.org/` an, welches u. a. die Produkte Apache, MySQL, PHP und phpMyAdmin enthält und sowohl für Linux als auch für Windows, Mac OS X und Solaris zur Verfügung steht.

Die Datei *config.php* enthält die dvdstats-Konfigurationsvariablen, darunter Authentifizierungsdaten für die MySQL-Datenbank, Schwellenwerte für die Analyse sowie Ausgabeoptionen. Mit dem Installationsskript *install.php* werden automatisiert die Datenbank und die dazugehörigen Tabellen im System angelegt.

Damit neben der Webserver-Logdatei auch die zusätzlichen JavaScript-Attribute erfasst werden, muss in die zu messenden Webseiten ein Codeschnipsel eingebaut werden. Dieser besteht aus JavaScript-Code und einem Zählpixel (vgl. Kapitel 3).

Wie in Abschnitt 6.2 beschrieben, muss zum Starten des Analysevorgangs das Programmstart-Skript *dvdstats.php* ausgeführt werden. Wichtig ist, dass dabei der Dateipfad zur Webserver-Logdatei als Parameter übergeben wird.

Listing 6.1 Ausführung des Start-Skripts alle 6 Stunden mit crontab

```
1 # m h  dom mon dow   command
2 0 */6 * * * /pfad/zu/dvdstats/dvdstats.php /pfad/zur/logdatei/access.log
```

6.4 Test

Über einige Tage sammelte dvdstats Daten einer tatsächlich existierenden deutschen Website im Internet, die auf den Webserver lighttpd setzt. Ein JavaScript-Zählpixel-Codeschnipsel wurde für diesen Zeitraum auf der Website platziert, um den gesamten Funktionsumfang des Prototypen zu testen.

Die für diesen Test eingesetzte Logdatei war schließlich 7,55 GiB groß und enthielt rund 28 Millionen Einträge. Zusätzlich entstanden durch Zählpixel und JavaScript rund 685000 Datensätze, die gemeinsam einen Speicherplatz von etwa 260 MiB in der Datenbank einnahmen.

Aus diesen Quelldaten entstanden während einer Bearbeitungszeit von etwa 2 Stunden in den dvdstats-Tabellen *dwh_dimvalues* und *dwh_facts* schließlich optimierte Datensätze. Dies waren ca. 137000 Dimensions- und ungefähr 184000 Faktenwerte bei einem Speicherplatzverbrauch von insgesamt 63 MiB.

Im Anschluss daran konnte eine separat entwickelte Web-Applikation aus diesen Daten Statistiken generieren (vgl. Abschnitt 6.2) und diese in einem Webbrowser darstellen.

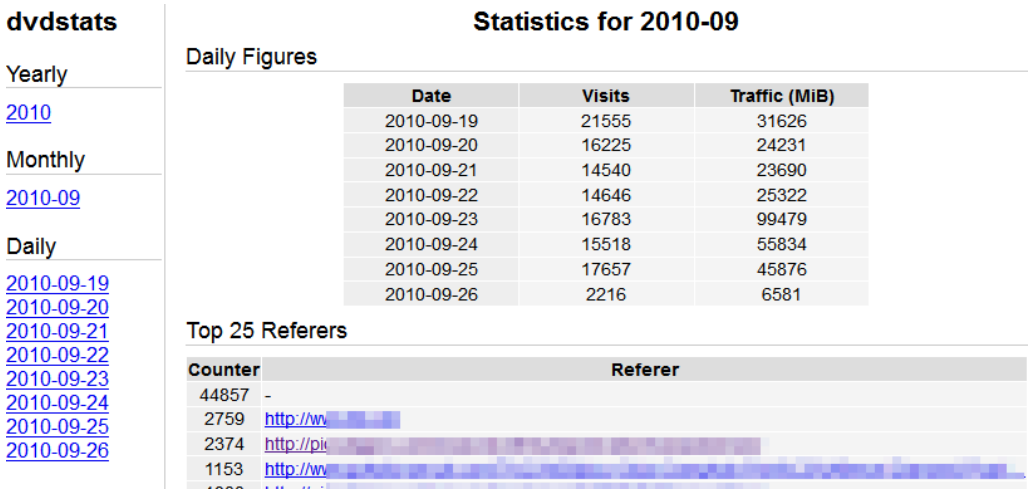


Abbildung 6.11: Beispiel-Statistiken, generiert aus dvdstats-Daten

Es stellte in diesem Test auch bei größeren Datenmengen kein Problem dar, tägliche und monatliche Statistiken zur Laufzeit aus der Datenbank zu generieren.

Aufgrund des Verzichts auf Redundanzen müssen die Daten jedes Tages stets einzeln verarbeitet werden, wodurch bei höherer Datenmenge die Generierung einer jährlichen Statistik einige Bearbeitungszeit in Anspruch nehmen kann. Es wäre aber denkbar, solche Statistiken zusammen mit der Auswertung über einen Zeitplan zu erstellen und per Cache bereitzuhalten.

In Abbildung 6.11 ist eine beispielhafte monatliche Statistik dargestellt, die Visits, Traffic und eine Liste der am häufigsten vorkommenden Referer umfasst. Diese Statistik kann mit wenig Aufwand auch um weitere Attribute, wie die am häufigsten aufgerufenen Objekte bzw. Dateien oder eingesetzte Client-Browser, erweitert werden.

Neben Listen und Tabellen lassen sich aus den Daten auch Diagramme generieren. Bei Kennzahlen, die die Client-Konfiguration beschreiben, bieten sich, wie in Abbildung 6.12 zu sehen, vor allem Kreisdiagramme an.

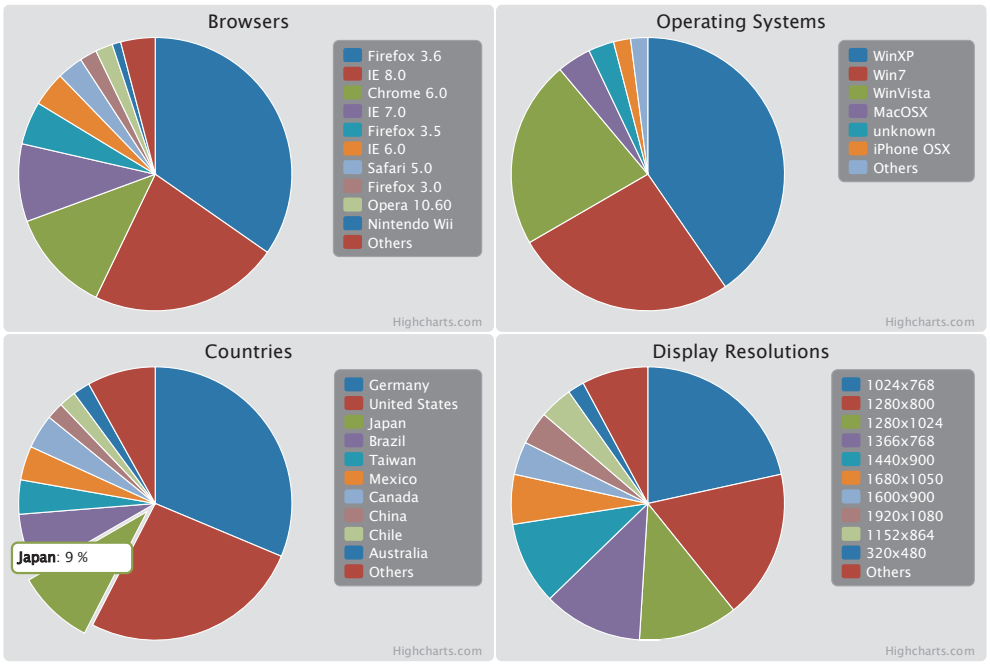


Abbildung 6.12: Kreisdiagramme, generiert aus dvdstats-Daten

In diesem Beispiel wurde auf die JavaScript-Bibliothek von Highcharts.com (`http://www.highcharts.com/`) zurückgegriffen, die aus bereitgestellten Daten im JSON-Format beliebige Diagramme generieren kann.

Die zuvor gesammelten Geodaten in Längen- und Breitengraden können schließlich mit Hilfe der Google Maps API (`http://code.google.com/intl/de/apis/maps/`) auf einer Karte dargestellt werden. Google Maps basiert auf JavaScript und kann so genannte Marker auf eine Weltkarte setzen, die ebenfalls im JSON-Format übergeben werden.

Eine sinnvolle Gruppierung dieser Marker kann durch die Erweiterung MarkerClusterer erreicht werden, die unter `http://code.google.com/p/gmaps-utility-library-dev/` (abgerufen am 28.05.2011) angeboten wird. Durch diese Gruppierung entsteht eine übersichtliche Ansicht, die sich dem aktuell sichtbaren Kartenausschnitt dynamisch anpasst.

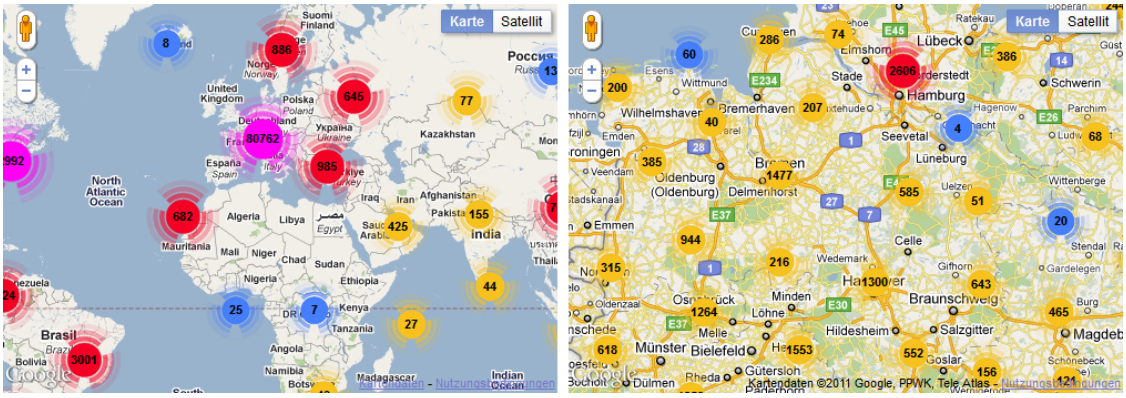


Abbildung 6.13: Google Maps mit gruppierten Markern

In Abbildung 6.13 ist zu sehen, wie z. B. für ganz Mitteleuropa eine große Gruppierung mit 80762 Einzelmarkern angezeigt wird. Zoomt man nun in einen Ausschnitt hinein, so werden kleinere Gruppierungen sichtbar, die zuvor Bestandteile der größeren Gruppierung waren.

6.5 Bewertung

Der entwickelte Prototyp erfüllt alle zuvor festgelegten Ziele. Die zu analysierenden Rohdaten wurden aus einer Webserver-Logdatei gewonnen, die um zusätzliche Rohdaten aus einem Zählpixel-gestützten JavaScript-Codeschnipsel ergänzt wurden. Die Datenspeicherung ist effizient möglich und bietet eine hierarchische Struktur. Es entstehen zudem keine Redundanzen und auch die Wartung gestaltet sich einfach, da bei Ergänzungen die Datenbank-Struktur nicht modifiziert werden muss.

Während des Tests stellte sich heraus, dass der Prototyp performancetechnisch nicht mit marktüblichen Tools mithalten kann. Während dvdstats für 1 Million Einträge ungefähr 4 Minuten benötigt, dauert ein ähnlicher Vorgang mit dem etablierten Tool Webalizer laut Angaben des Autors nur etwa 15 Sekunden. Natürlich muss Webalizer seine Rohdaten nicht mit zusätzlichen Daten aus einer Datenbank abgleichen; aber auch die Wahl der Programmiersprache lässt Rückschlüsse zu. PHP-Programme arbeiten deutlich langsamer als identische Programme, die in der hardwarenäheren Programmiersprache C geschrieben wurden. Dies lässt sich z. B. im „Computer Language Benchmarks Game“ nachvollziehen, welches unter `http://shootout.alioth.debian.org/u32/benchmark.php?test=all&lang=php&lang2=gcc` (abgerufen am 19.05.2011) erreichbar ist.

Kapitel 7

Fazit

7.1 Zusammenfassung

Die in dieser Arbeit aufgezeigten Methoden der Web-Analytik stellen eine wichtige Grundlage dar, um das eigene Internet-Angebot durch Fakten gestützt zu bewerten. Sie decken zudem Schwächen in der Usability auf und können dabei helfen, diese Schwächen aktiv zu bekämpfen. Je weniger Schwächen in der Gestaltung und Benutzerführung existieren, desto höher ist die Chance, angestrebte Ziele zu erreichen.

Obwohl einige der in Kapitel 3 und 4 aufgezeigten Methoden schon seit vielen Jahren eingesetzt werden, bilden sie auch heute noch das Fundament der Web-Analytik. Um erfolgreich einen Online-Shop zu betreiben, die eigenen Produkte zu vermarkten und Werbeflächen anzubieten (vgl. Kapitel 2.3), ist es nach wie vor von großer Wichtigkeit, über die eigenen Visits und Page Views bzw. Ad Views informiert zu sein. Klassische Kennzahlen, wie Hits ohne Besuchersitzungserkennung, spielen heute hingegen kaum noch eine Rolle, da diese eher Auskunft über die Gesamtbelastung des Webservers geben, als über den tatsächlichen Besucherverkehr.

Wie in Kapitel 4 beschrieben, bieten modernere Verfahren, die sowohl auf eine server- als auch auf eine clientseitige Programmlogik setzen, eine große Vielfalt an zusätzlichen Möglichkeiten. Nicht nur die Wiedererkennung früherer Besucher und die Analyse erweiterter Browser-Attribute werden dadurch ermöglicht; Klickpfade und das Benutzerverhalten im Rahmen einer Usability-Analyse können ebenfalls untersucht werden.

Die Marktanalyse in Kapitel 5 hat gezeigt, dass viele Open-Source-Produkte mit den kommerziell angebotenen Lösungen noch nicht gleichziehen können. Google bietet mit Google

Analytics ein kostenloses Tool, das nahezu auf dem Stand der Forschung und Technik ist. Das Unternehmen etracker bietet mit seiner kostenpflichtigen Web-Analytik-Lösung zudem einige noch sehr neue Methoden, wie die der Usability-Analyse per Mausspur-Tracking.

Mit dem Entwurf des eigenen Prototypen dvdstats in Kapitel 6 konnte ein guter Einblick in die eingesetzten Methoden gewonnen werden. Dieser Einblick trug auch zur objektiven Bewertung der etablierten Lösungen bei. Es stellte sich heraus, dass die grundlegenden Kennzahlen direkt aus vorhandenen Logdatei-Rohdaten extrahiert werden können. Da die moderneren Methoden, die zusätzlichen clientbasierten Programmcode (JavaScript) benötigen, sehr vielfältig sind, wurden im Rahmen des prototypischen Entwurfs nur die Rohdaten untersucht, die aus den vordefinierten JavaScript-Objekten gewonnen werden können. Eine besondere Herausforderung stellte auch die optimale Speicherung der gewonnenen Daten dar (vgl. Kapitel 6.1.1.3). Schließlich wurde das Generische Modell des Data-Warehouse-Ansatzes gewählt, welches sich perfekt für derart dynamische Rohdaten, die aus mehreren Quellen stammen, eignet. Trotz geringer Performance-Schwäche, die z. B. durch die Portierung in eine andere Programmiersprache abgewendet werden kann, ist mit dvdstats ein Web-Analytik-Tool entstanden, das bereits als Prototyp brauchbare Ergebnisse liefern kann.

7.2 Ausblick

Die Entwickler marktüblicher Web-Analytik-Tools werden künftig auch noch an der Erschließung neuer Methoden arbeiten. Das relativ neue Verfahren des Mausspur-Trackings ist mit großer Wahrscheinlichkeit nur der Anfang einer Reihe künftiger Möglichkeiten zur Website-Optimierung.

Während sich der Trend immer weiter in Richtung umfangreicherer Usability-Analysen bewegt, gibt es auch noch Forschungsbedarf bei der Verbesserung vorhandener Methoden. So existiert z. B. bereits ein vielversprechender Ansatz zur Identifizierung von wiederkehrenden Besuchern, die Cookies ablehnen (vgl. Kapitel 4.1).

Der implementierte Prototyp dvdstats kann im derzeitigen Entwicklungsstand keine Event-, Funnel- und Usability-Analysen durchführen. Eine Erweiterung um diese Funktionen wäre mit dem eingesetzten Datenbankmodell, dem Generischen Modell des Data-Warehouse-Ansatzes, aber denkbar. Mit einer Erweiterung um diese und weitere Funktionen könnte sich aus dvdstats durchaus eine ernstzunehmende Alternative zu heute etablierten Tools entwickeln.

Glossar

API API steht für Application Programming Interface (Programmierschnittstelle) und bietet Zugriff auf vom Anbieter zur Verfügung gestellte Daten und Dienste.

Caching Dieser Begriff beschreibt den Vorgang des Zwischenspeicherns von zuvor abgefragten Daten zugunsten des Performance-Gewinns bei einem erneuten Zugriff.

Cookie Unter Cookies versteht man kleine Dateneinheiten, die zusammen mit einem Verfallsdatum und dem Gültigkeitsbereich innerhalb einer Domain genau ein Key-Value-Paar speichern können.

CSV CSV steht für Comma Separated Values, also durch Komma getrennte Werte im Klartext-Format.

Hotlink Man spricht von Hotlinking, wenn Objekte (Bilder, Downloads usw.) – meist ohne Erlaubnis – auf Websites Dritter eingebunden werden.

Intranet Unter einem Intranet versteht man ein nicht-öffentliches Netzwerk in einem begrenzten Teilnehmerkreis. Es existieren Parallelen zum globalen Internet, da auch in Intranets oftmals Web-Dienste, wie z. B. Webserver oder E-Mail, eingesetzt werden.

JSON JSON steht für JavaScript Object Notation und ist ein Format, das sich gut zum Austausch von Daten zwischen verschiedenen Anwendungen eignet.

MAC-Adresse Eine Media-Access-Control-Adresse (MAC) stellt eine feste, weltweit eindeutige Hardware-Adresse dar, die jedem Netzwerkgerät während der Herstellung fest zugewiesen wird.

Rewrite-Direktive In Rewrite-Direktiven werden reguläre Ausdrücke definiert, mit denen virtuelle Pfade angelegt werden können, unter denen Objekte bereitgestellt werden.

Robot Robots (auch Crawler genannt) sind Programme, die automatisiert auf Websites zugreifen und dabei Informationen sammeln. Zum Beispiel besitzt jede Suchmaschine eigene Robots, um ihre Datenbanken aufzubauen.

Virtueller Pfad Bei einem virtuellen Pfad wird ein Objekt unter einer abweichenden, selbst definierten Adresse bereitgestellt.

Werbeblocker Werbeblocker können zumeist als Webbrowser-Plugin oder Proxyserver installiert werden. Mittels einer Filterliste, die aus regulären Ausdrücken besteht, werden Werbe-Webserver erkannt. Jegliche Werbeinhalte können dadurch nahezu vollständig aus einer Webseite herausgefiltert werden.

Literaturverzeichnis

[Apa] *Apache HTTP Server – Log Files*. http://httpd.apache.org/docs/current/logs.html. – abgerufen am 28.03.2011

[BLFM05] BERNERS-LEE, T. ; FIELDING, R. ; MASINTER, L.: *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986 (Standard). (Januar 2005). http://www.ietf.org/rfc/rfc3986.txt

[Dos09] DOSHI, Suhail: Introduction to Analytics: Funnel Analysis. (2009). http://blog.mixpanel.com/introduction-to-analytics-funnel-analysis. – abgerufen am 17.12.2010

[FGM+99] FIELDING, R. ; GETTYS, J. ; MOGUL, J. ; FRYSTYK, H. ; MASINTER, L. ; LEACH, P. ; BERNERS-LEE, T.: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard). (Juni 1999). http://www.ietf.org/rfc/rfc2616.txt

[Ger98] GERKEN, Wolfgang: Ein generisches Data Warehouse im Kontext einer Anwendungsarchitektur. (1998)

[Joh85] JOHNS, M. S.: *Authentication server*. RFC 931. (Januar 1985). http://www.ietf.org/rfc/rfc931.txt

[Joh93] JOHNS, M. S.: *Identification Protocol*. RFC 1413 (Proposed Standard). (Februar 1993). http://www.ietf.org/rfc/rfc1413.txt

[Kin08] KING, Andrew B.: *Website Optimization: Speed, Search Engine & Conversion Rate Secrets*. 1. Auflage. O'Reilly Media, 2008 http://www.websiteoptimization.com/secrets/metrics/. – ISBN: 978-0596515089

[Lig11] *Lighttpd Docs: Accesslog*. (2011). http://redmine.lighttpd.net/wiki/1/Docs:ModAccessLog. – abgerufen am 28.03.2011

[LL10] LIENEMANN, Gerhard ; LARISCH, Dirk: *TCP/IP - Grundlagen und Praxis: Protokolle, Routing, Dienste, Sicherheit*. 1. Auflage. Heise, 2010. – ISBN: 978-3936931693

[MSS] MÜNZ, Stefan ; SCHÄFER, Mathias ; STRÜBIG, Joachim: *SELFHTML: JavaScript-Objektreferenz*. http://de.selfhtml.org/javascript/objekte/. – abgerufen am 28.03.2011

[MST] MÜNZ, Stefan ; SKOP, Roland ; TURSKI, Ingo: *SELFHTML: HTML/XHTML*. http://de.selfhtml.org/html/allgemein/grundgeruest.htm. – abgerufen am 17.05.2011

[Ree08] REESE, Frank: *Web Analytics – Damit aus Traffic Umsatz wird – Die besten Tools und Strategien*. BusinessVillage, 2008. – ISBN: 978-3-938358-71-9

[W3C95] *Logging Control In W3C httpd*. (1995). http://www.w3.org/Daemon/User/Config/Logging.html. – abgerufen am 28.03.2011

[W3T11] *Usage of traffic analysis tools for websites*. (2011). http://w3techs.com/technologies/overview/traffic_analysis/all. – abgerufen am 04.05.2011

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Inhalt der CD

A.1 Quellcode des Prototypen dvdstats

Enthalten ist der komplette Quellcode des Analyseprogramms und der separaten Web-Applikation.

A.2 Diese Arbeit

Auf der CD befindet sich eine digitale Kopie dieser Arbeit im PDF-Format.

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw.§24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Glinde, 01.06.2011 Daniel von Dombrowski