CSS:
Angepasste Inhalte für mobile Endgeräte

Dirk Schürjohann

E-Mail: selfl@decaf.de

Homepage-URL: http://decaf.de

Bei Fragen zu diesem Beitrag bitte den Autor des Beitrags kontaktieren!

Einleitung

Die Verbreitung mobiler Endgeräte (PDA, Handhelds, Smartphones) nimmt stetig zu und übertrifft die Anzahl von Desktop-Computern längst um ein Vielfaches. Viele von ihnen sind inzwischen in der Lage, auch "normale" Webseiten statt speziell dafür aufbereitete Inhalte auszugeben. Allerdings funktioniert oft weder die Ansprache noch die Umsetzung der Webseiten zuverlässig.

Dieser Artikel beschreibt Möglichkeiten, wie mobile Endgeräte recht zuverlässig angesprochen und daraufhin mit alternativen Inhalten und CSS-Stylesheets bedient werden können, wenn es sich um ein Webangebot mit nur einem Inhalt für die Verwendung in unterschiedlichen Ausgabemedien handelt.

Probleme von `media="handheld"` und `display:none`

In HTML können über das `media`-Attribut separate Stylesheets für unterschiedliche Ausgabemedien aufgeführt werden, jedoch liegt die Entscheidung darüber, welche Angaben verwendet werden, allein beim Browser/Ausgabegerät. Aktuell ignorieren leider noch viele mobile Ausgabegeräte die Angabe `media="handheld"` und greifen stattdessen auf das Standard-Stylesheet der Seite zu, das über `media="all"` oder `media="screen"` eingebunden wird.

Hinzu kommt die Interpretation von `display:none`, das in Handheld-Stylesheets gerne dafür verwendet wird, Bilder oder Inhaltsbereiche auszublenden, um die Menge der zu übertragenden Daten zu verringern. Die Inhalte einer Seite könnten dadurch aufs Wesentliche reduziert werden und eine schnelle Übertragung zum mobilen Gerät begünstigen. Theoretisch eine schöne Methode, in der Praxis jedoch wieder problematisch: viele Handhelds verstehen zwar die Angabe, Inhalte auszublenden, fordern sie jedoch trotzdem vollständig vom Server an. Es werden also unnötige Daten übertragen, die die Anzeige der Seite massiv ausbremsen können und gegebenenfalls das Trafficvolumen des Nutzers vergeuden.

Fazit: beide Methoden – die Verwendung von alternativen Stylesheets für mobile Ausgabegeräte und das Ausblenden von Bildern oder Inhaltsbereichen – werden von vielen Geräten noch nicht zufriedenstellend unterstützt.

Mögliche Maßnahmen: eine zuverlässigere Ansprache von mobilen Endgeräten und die daraus resultierende Möglichkeit, nicht zu übertragende Inhalte zu verstecken.

Maßnahmen zur Ansprache mobiler Endgeräte

Eine erste Einschätzung über die Art des Ausgabegeräts kann über die Browserkennung (User Agent) erfolgen. Das ist keine sichere Methode, weil die Browserkennung vom Benutzer verändert oder unterdrückt werden kann, und weil die Liste aller Kennungen nur schwerlich aktuell gehalten werden kann, jedoch hilft dieses Vorabprüfung durchaus, eine große Gruppe von mobilen Geräten, die das `media`-Attribut verpasst haben, wieder einzufangen:

Beispielfunktion zur Prüfung der Browserkennung (PHP)

Die Funktion wird ausgelagert und später in jede Seite des Webangebots per `include/require` eingebunden, um die Liste der Browserkennungen zentral in nur einer Datei pflegen zu müssen, falls sich Änderungen ergeben.

check_mobile.php:

```
<?php

function check_mobile() {
  $agents = array(
    'Windows CE', 'Pocket', 'Mobile',
    'Portable', 'Smartphone', 'SDA',
    'PDA', 'Handheld', 'Symbian',
    'WAP', 'Palm', 'Avantgo',
    'cHTML', 'BlackBerry', 'Opera Mini',
    'Nokia'
  );

  // Prüfen der Browserkennung
  for ($i=0; $i<count($agents); $i++) {
    if(isset($_SERVER["HTTP_USER_AGENT"]) && strpos($_SERVER["HTTP_USER_AGENT"], $agents[$i]) !== false)
      return true;
  }

  return false;
}

?>
```

Erklärung der Funktion: die (höchstwahrscheinlich unvollständige) Liste von Zeichenketten aus Browserkennungen, hinter denen mobile Endgeräte vermutet werden, wird in einem Array erfasst und daraufhin mit der vom Server hoffentlich ausgegebenen Kennung in $_SERVER["HTTP_USER_AGENT"] verglichen. Wird ein Schlagwort erkannt, liefert die Funktion ein true.

Einbinden der Funktion

An den Anfang jeder Seite des Webangebots stellen wir den Aufruf der externen Funktion:

```
<?php

require('check_mobile.php');

$style = '';

if(check_mobile()) $style = 'handheld';

?>
```

Erklärung: gibt die per `require` eingebundene Funktion ein true zurück, wird der Variable $style (die zuvor zwecks eines sauberen Programmierstils durch `$style = ''` initialisiert wurde) der Inhalt 'handheld' zugewiesen. Im weiteren Verlauf der Seite kann diese Variable nun jederzeit abgefragt werden, um bei Bedarf einen Inhaltsbereich für die Verwendung auf mobilen Ausgabegeräten anzupassen.

Weitere Maßnahmen?

An dieser Stelle könnten nun weitere Maßnahmen angeführt werden, die z.B. auf die technischen Möglichkeiten des Ausgabegeräts eingehen und anhand fehlender Unterstützung bestimmter Methoden und Techniken versuchen, mobile Geräte einzugrenzen. Aber auch diese Maßnahmen wären kaum vollkommen zuverlässig und mit einem Aufwand verbunden, der die Absicht kaum rechtfertigt. Es steht allerdings noch eine Ressource zur Verfügung, die wir recht sicher abfragen können: **der Benutzer**. Anhand eines schlichten Links kann er entscheiden, welche Art von Stylesheet verwendet werden soll:

Manuelle Auswahl durch den Benutzer

Definition der Stylesheets im Seitenkopf (`<head>`):

```
<link rel="stylesheet" type="text/css" href="screen.css"   media="screen"   />
<link rel="stylesheet" type="text/css" href="handheld.css" media="handheld" />
```

Links zur manuellen Auswahl des Stylesheets im Body (`<body>`):

```
<div id="header">
  <?php
  if(empty($style) || $style == 'default') {
    echo '<span id="nopocket">Style ändern: <a href="?handheld">Pocket-Version</a>
      für PDA und Handhelds. </span>';
  }
  ?>
  <span id="pocket">Style ändern: <a href="?default">Desktop-Version</a></span>
</div>
```

Angaben in screen.css:

```
#pocket   { display:none; }
```

Angaben in handheld.css:

```
#pocket   { display:inline; }
#nopocket { display:none; }
```

Erklärung: Zur Auswahl des Stylesheets durch den Benutzer werden zwei Links eingesetzt, von denen im besten Fall jeweils nur einer angezeigt wird: der Link zur Pocket-Version in der Desktop-Ansicht, und anders herum der Link zur Desktop-Version in der Pocket-Ansicht. Das geschieht zum einen durch die Zuweisung der IDs `pocket` und `nopocket` und dessen nachfolgende Angaben in den Stylesheets, zum anderen aber auch durch ein PHP-Konstrukt, das den ersten Link sicher ausblendet, falls `$style` nicht definiert oder als `'default'` gesetzt wurde.

Doch warum dieses Konstrukt nur für einen Link, obwohl man durch eine `if-else`-Variante beide Links ansprechen könnte? Das geschieht deshalb, weil es unter Umständen mobile Ausgabegeräte geben kann, dessen Prüfung der Browserkennung versagt. Kommt dann hinzu, dass weder ein Cookie vorhanden noch eine manuelle Auswahl durch den Benutzer stattgefunden hat, kann das zu verwendende Stylesheet nicht bestimmt werden und die Ausgabe der Inhalte hängt allein von der Interpretation des `media="handheld"` ab. Eine `if-else`-Routine würde dann entweder den "falschen" oder aber gar keinen Link ausgeben, was in keinem Fall gewünscht ist.

Parameter innerhalb der Links: Die beiden Links zur Auswahl des Stylesheets enthalten in diesem Artikel lediglich die einfachen Parameter `?default` und `?handheld`, die wie im nachfolgenden Absatz beschrieben als Query-String ausgelesen und weiterverarbeitet werden. Für den Einsatz in der Praxis ist diese Methode allerdings häufig nicht sinnvoll und mindestens die kompatiblere Variante mit Parametern in der Form `?style=default` nötig, um Konflikte mit anderen Parametern zu vermeiden.

Query-Strings auslesen und Cookie setzen

Die folgenden Angaben werden an den Anfang jeder Seite innerhalb des Webprojekts unmittelbar **hinter** die eingebundene Funktion zur Prüfung der Browserkennung (`check_mobile()`) gesetzt:

```
<?php

$query = $_SERVER['QUERY_STRING'];

if($query != 'handheld' && $query != 'default') unset($query);
else {
  $style = $query;
  setcookie('ausgabemedium', $style, time()+60*60*24*360, '/');
}

?>
```

Erklärung: der Query-String wird über die Server-Variable `$_SERVER['QUERY_STRING']` ausgelesen. Handelt es sich weder um die Angaben `'handheld'` oder `'default'`, wird der übergebene Wert gelöscht. Andernfalls wird wie schon zuvor bei der Browserkennung die Variable `$style` mit dem zu verwendenden Stylesheet beschrieben. Und weil diesmal die Entscheidung aus sehr sicherer Quelle stammt, nämlich vom Benutzer selbst, halten wir den Wert in einem Cookie fest, um nicht bei jedem Aufruf der Seite aufs Neue entscheiden zu müssen.

Die Abfrage des Cookies (Bezeichnung bei Bedarf anpassen!) sieht etwa so aus:

Cookie auslesen

```
<?php

if(isset($_COOKIE["ausgabemedium"])) $style = $_COOKIE["ausgabemedium"];
```

```
?>
```

Die Abfrage des Cookies setzen wir nach der Maßnahme der Browserkennung und vorm Auslesen des Query-Strings ein, um eine sinnvolle Abfolge zu erhalten:

Richtige Reihenfolge der Maßnahmen

1. **Vorabprüfung der Browserkennung**
   Diese Methode alleine ist nicht zuverlässig, weil die Browserkennung vom Benutzer verändert oder unterdrückt werden kann, und weil die Liste aller Kennungen nur schwerlich aktuell gehalten werden kann. Die Vorabprüfung ist jedoch hilfreich, um eine große Gruppe von mobilen Geräten, die das `media`-Attribut verpasst haben, wieder einzufangen.
2. **Auslesen des Cookies**
   Ist ein Cookie vorhanden, können wir davon ausgehen, dass seine enthaltenen Informationen zum verwendeten Stylesheet recht zutreffend sind, denn der Cookie wird nur dann gesetzt, wenn eine Auswahl durch den Benutzer erfolgte. Ein vorhandener Cookie kann deshalb für die Einschätzung des Ausgabegeräts als zuverlässiger angesehen werden als die Prüfung der Browserkennung, so dass sein Inhalt den Wert der bereits definierten `$style`-Variable gegebenenfalls bewusst überschreibt.
3. **manuelle Auswahl durch den Benutzer**
   Die einzig relevante Maßnahme zur Bestimmung des Ausgabegeräts ist der Benutzer selbst. Ihm unterstellen wir, dass er das verwendete Stylesheet bewusst wählt und danach beibehalten möchte, egal was die beiden Prüfungen zuvor ergeben haben. Seine Auswahl wird in einem Cookie gespeichert bzw. der bereits vorhandene Cookie aktualisiert.

Erklärung: Die drei beschriebenen Maßnahmen werden in einer Reihenfolge eingesetzt, die dem Benutzer die finale Bestimmung des Ausgabegeräts überlässt. Sollte er keine Auswahl treffen, wird der Inhalt des Cookies relevant, enthält er doch das zuletzt benutze Stylesheet, das zuvor wiederum ausschließlich durch den Benutzer bestimmt wurde. Ist kein Cookie vorhanden, richtet sich die Auswahl nach der Browserkennung. Konnte auch diese nicht ermittelt werden, weil sie vom Browser nicht übertragen wird oder in der vorhandenen Liste der Kennungen in `check_mobile.php` nicht aufgeführt ist, bleibt nur die Hoffnung auf eine richtige Interpretation der Angabe `media="handheld"`.

Bilder oder Inhaltsbereiche vor mobilen Ausgabegeräten verstecken

Nachdem nun recht sicher bestimmt werden konnte, welche Art von Ausgabegerät es zu bedienen gilt, können Bilder und Inhaltsbereiche zuverlässig versteckt werden, ohne dass die Daten trotz `display:none` übertragen werden und dadurch den Benutzer und die Benutzbarkeit stören. Dazu wird die vorher definierte Variable `$style` ausgelesen, so wie bei diesem Beispiel:

```
<?php

if ($style != 'handheld')
  echo ' ..[Bild oder Inhaltsbereich, der für Handhelds ausgeblendet wird].. ';

?>
```

Weitere Maßnahmen nach Belieben hinzugeben..

Fazit und Anmerkung

Fazit: Die Unzulänglichkeit, dass noch immer viele mobile Ausgabegeräte nicht auf die Angabe von separaten Stylesheets durch `media="handheld"` reagieren, kann mit etwas Aufwand recht zuverlässig umgangen werden. Dennoch geht es nicht sicher ohne die entscheidende Angabe durch den Benutzer. Wurde durch diese Maßnahmen das richtige Stylesheet bestimmt, können weitere Schritte unternommen werden, um eine Website für mobile Geräte benutzbarer zu machen. Dazu gehört vor allem das Ausblenden von Bildern oder Inhaltsbereichen, die aufgrund ihrer Struktur oder großen Datenmenge nicht für mobile Geräte geeignet sind. Die Interpretation vieler Geräte, diese Daten trotz Angabe von `display:none` dennoch zu übertragen, kann dadurch in vielen Fällen ausgeglichen werden.

Anmerkung zu mobilen Webinhalten: dieser Artikel behandelt lediglich die Ansprache von mobilen Ausgabegeräten und die nachfolgende Möglichkeit, Bilder oder Inhaltsbereiche zu verstecken. Er geht nicht darauf ein, wie mobile Inhalte zu strukturieren und zu gestalten sind, und welche Maßnahmen generell nötig sind, um ein Webangebot für den mobilen Einsatz vollkommen tauglich zu machen. Denn neben dem Ansatz von nur einem Inhalt für die Verwendung in unterschiedlichen Medien durch alternative Stylesheets gibt es durchaus die Möglichkeit, separate Inhalte ausschließlich für mobile Endgeräte zu produzieren. Nähere Informationen dazu behandelt Cameron Molls 4-teilige Serie "Mobile Web Design - The Series" und einige der nun folgenden weiterführenden Links:

Weiterführende Links

Die folgenden Stellen werden empfohlen, um den Artikel besser zu verstehen und weitere Möglichkeiten und Details zu erfahren.

- A List Apart, Pocket-Sized Design: Taking Your Website to the Small Screen
- An Overview of Mobile Versions of XHTML

- Creating Web Content for Mobile Phone Browsers, Part 1
- Creating Web Content for Mobile Phone Browsers, Part 2
- XHTML-MP Style Guide
- XHTML Mobile Profile / XHTML MP Tutorial

- WURFL, the Wireless Universal Resource File
- List of User Agent Strings
- Mobile Browser ID (User-Agent) Strings

SELFHTML aktuell Artikel CSS