Datenbank-basierte Webserver

Design-Optimierung eines Datenbank-basierten Webservers

Karl Riedling
Institut für Sensor- und Aktuatorsysteme

Design-Optimierung

- Grundsätzliche Überlegungen zur Gestaltung von Webseiten
- Spezielle Aspekte bei Formularen
- Browser-unabhängige Webseiten
- Test von dynamischen Webseiten

Design-Optimierung

- **Grundsätzliche Überlegungen zur Gestaltung von Webseiten**
- Spezielle Aspekte bei Formularen
- Browser-unabhängige Webseiten
- Test von dynamischen Webseiten

Grundsätzliche Überlegungen

- Klare, übersichtliche Seitengestaltung
- Hinreichende Navigationshilfen für die User vorsehen (Navigations-Bereich im Seitenkopf oder links)
- Guter Kontrast zwischen Schrift- und Hintergrundfarbe
- Vernünftige Schriftgrößen
- Browser können ohne zusätzliche Maßnahmen nur Fonts anzeigen, die am Client-Rechner installiert sind – keine ausgefallenen Fonts verwenden!
- Optische Effekte (Animationen usw.) *sparsamst* verwenden

Grundsätzliche Überlegungen

- Bilder und andere *Plug-In*s in vernünftiger Auflösung (d.h. Dateigröße) anbieten – Ladezeiten so kurz wie möglich halten
- Bilder mit jener Auflösung am Server bereit stellen, mit der sie angezeigt werden sollen – keine Skalierung mit dem `<img>`-*Tag*!
- Linienzeichnungen nie in einem Daten-Format anbieten, das verlustbehaftete Komprimierung verwendet (JPEG!); besser: GIF, PNG

Grundsätzliche Überlegungen

- Browser können Tabellen, die Objekte mit fixer, aber nicht definierter Größe enthalten (z.B. Bilder) vielfach erst dann anzeigen, wenn das letzte Bild geladen ist ➔ in `<img>`-*Tag*s immer „`height=`“ und „`width=`“-Attribute *korrekt* setzen!

Grundsätzliche Überlegungen

- Seitencode grundsätzlich mit so vielen Browsern wie möglich testen (manche Browser tolerieren Fehler, die andere total verwirren!)
- Zweckmäßig: **Webseiten-Validierungs-Services verwenden (z.B. `http://validator.w3.org/`)!**
- Alte *Opera*-Browser (bis Version 12) haben eine integrierte Validierungs-Funktion, die `http://validator.w3.org/` aufruft
- Alternativ: Validierungs-Funktionen des Browsers (soweit vorhanden) verwenden

Design-Optimierung

- Grundsätzliche Überlegungen zur Gestaltung von Webseiten
- **Spezielle Aspekte bei Formularen**
- Browser-unabhängige Webseiten
- Test von dynamischen Webseiten

Spezielle Aspekte bei Formularen

- Formulare möglichst übersichtlich gestalten
- Gesamter Formularinhalt sollte möglichst auf einer Bildschirmseite sichtbar sein
- Umfangreiche Formulare aber nicht aufteilen (Navigation ist umständlich oder sogar unmöglich!)
- Ähnliches Design für verwandte Formularseiten verwenden (z.B. immer gleiche Anordnung diverser Schaltflächen)
- HTML-Tabellen eignen sich hervorragend für ein gutes Formular-Layout!

Spezielle Aspekte bei Formularen

- Browser können unterschiedliche Fonts für die Anzeige von Formularinhalten (Textfelder, *Text Areas*) und unterschiedliche Designs von Schaltflächen verwenden – beim Design nicht davon ausgehen, dass z.B. Auswahllisten, Textfelder, *Text Areas* und Schaltflächen auf allen Browsern in gleicher Breite erscheinen!
- Für eine saubere Anzeige *Styles* statt (oder zusätzlich zu) den `size=`- oder `col=`-Attributen von Formular-elementen verwenden!

Spezielle Aspekte bei Formularen

- Die Breite von Klapp- und Auswahllisten und das Design von Schaltflächen kann nur mit *Styles* eingestellt werden!
- Schriftgröße und -type von Formularelementen nur mit „`style`“-Attributen oder *Style Sheets* einstellen (keine `<font>`-*Tags*!)

Spezielle Aspekte bei Formularen

- JavaScript-Funktionen als Eingabehilfen:
  - Eingabedaten möglichst immer am Client-Rechner validieren (mit JavaScript) – spart den Usern längere Ladezeiten
  - Lokal generierte JavaScript-Fehlermeldungen (mit `alert()`) oder Benutzer-Entscheidungen (mit `confirm()`) sind schneller und auffälliger und erlauben auch eine einfache Rückkehr zu den gerade bearbeiteten Formulardaten

Spezielle Aspekte bei Formularen

- JavaScript-Funktionen als Eingabehilfen:
  - „Intelligente“ Einstellung von Optionen in Abhängigkeit von anderen Formular-Einstellungen (mit `onChange`- oder `onClick`-*Event Handlern*)
  - Eingabe-Fokus auf das wichtigste Formularelement – Methode `focus()`, z.B.:
    `document.MyForm.Text1.focus();`

Design-Optimierung

- Grundsätzliche Überlegungen zur Gestaltung von Webseiten
- Spezielle Aspekte bei Formularen
- **Browser-unabhängige Webseiten**
- Test von dynamischen Webseiten

Browser-unabhängige Webseiten

- Grundregel: Webseiten so einfach wie möglich gestalten!
- Vorsicht bei der Auswahl von *Web Authoring*-Software:
  - *Microsoft Frontpage* bzw. *Expression Web* erzeugt sehr *Internet Explorer*-spezifische Seiten
  - *Adobe Dreamweaver* generiert portableren HTML-Code

Browser-unabhängige Webseiten

- HTML ist von seinem Konzept her ein Instrument zur Anzeige von Informationen und keine Text- oder Grafik-Formatierungs-Umgebung – versuchen Sie nicht, es zu einer solchen zu machen!
- Wenn es auf präzise Formatierung ankommt, PDF verwenden (PDF kann übrigens auch mit PHP generiert werden!)

Browser-unabhängige Webseiten

- Unterschiede zwischen Browsern zu erwarten bei:
  - Schriftgrößen, auch bei Verwendung von `style`-Attributen
  - Breite und Design von Schaltflächen
  - Zeilenabständen – Vorsicht bei Fenstern mit fixer Größe!
  - Tabellen, auch mit `width`=-Attributen in den `<td>`-*Tag*s, speziell wenn die Breite des Tabellenfeldinhalts die spezifizierte Spaltenbreite übersteigt

Browser-unabhängige Webseiten

- Alle (einigermaßen) modernen Browser ergeben aber bei Verwendung von „`style`“-Attributen oder *Style Sheets* eine sehr ähnliche Seiten-Darstellung

Browser-unabhängige Webseiten

- Layout von Webseiten mit Hilfe von Tabellen:
  - Immer ***eine*** Tabelle pro Seite (bei Bedarf mit eingeschlossenen Untertabellen) – Formulare „fallen“ sonst gelegentlich „auseinander“!
  - Für Tabellen, die lückenlos zusammenhängen sollen, `<table **cellpadding="0" cellspacing="0"**>` verwenden

Design-Optimierung

- Grundsätzliche Überlegungen zur Gestaltung von Webseiten
- Spezielle Aspekte bei Formularen
- Browser-unabhängige Webseiten
- **Test von dynamischen Webseiten**

Test von dynamischen Webseiten

- Test des PHP- und MySQL-Codes:
  - Zweckmäßig mit provisorischen Ausgaben (➔ Dynamische Erstellung von Webseiten mit PHP)
  - Zur Kontrolle von MySQL-Operationen Status-Abfrage (Ergebnis der Methode `query()`) und die Property `error` verwenden (➔ Datenbank-Operationen unter Verwendung von PHP und MySQL)

Test von dynamischen Webseiten

- Test des JavaScript-Codes:
  - Die meisten Browser (praktisch alle modernen) haben eine JavaScript-Konsole oder Fehleranzeige-Funktion:
    - Aufrufbar aus dem „*Extras*…“- oder „*Tools*…“-Menü (moderne Browser)
    - Aufrufbar durch Eingabe von „`javascript:`“ in die URL-Zeile des Browsers (alte Versionen von *Firefox*, *Mozilla* und *Netscape*)

Test von dynamischen Webseiten

- Test des JavaScript-Codes:
  - *Internet Explorer* hat einen *Script-Debugger* (defaultmäßig deaktiviert!)
  - Vorsicht: *Script-Debugger* des *Internet Explorer* nicht permanent aktivieren! Viele Applikationen verwenden implizit den *Internet Explorer* mit oft fehlerhaftem Script-Code.
  - JavaScript mit möglichst vielen Browsern auf korrekte Funktion testen (oft Browser-Versions-spezifische Bugs)!

Test von dynamischen Webseiten

- Test des HTML-Codes:
  - Seiten mit möglichst vielen Browsern anzeigen und die Seitenwiedergabe vergleichen
  - Eklatante Abweichungen im Erscheinungsbild der selben Seite auf mehreren Browsern sind meist auf HTML-Fehler zurückzuführen (z.B. vergessene oder überschüssige „`<td>`“ oder „`</td>`“-*Tags*)
  - Speziell unter HTML (im Gegensatz zu XHTML) stellen solche Fehler nicht notwendiger Weise einen Verstoß gegen die HTML-Syntax dar!

Test von dynamischen Webseiten

- Test des HTML-Codes:
  - In der Regel können mehrere Browser(versionen) gleichzeitig aktiv sein; Ausnahmen:
    - *Internet Explorer*: Ohne zusätzliche Maßnahmen immer nur eine Version pro Rechner verfügbar
    - Für Tests mit unterschiedlichen Versionen von *Internet Explorer* virtuelle Maschine(n) installieren

Test von dynamischen Webseiten

- Empfohlene Testbrowser unter Windows:
  - *Internet Explorer* (in der verfügbaren Version)
  - *Firefox* (in einer aktuellen Version)
  - Google Chrome (in einer aktuellen Version)
  - *Opera* (in einer aktuellen Version)
  - Safari (in einer aktuellen Version)
- „Historische“ Browser (alte Versionen der oben genannten Browser) können Design-Schwächen aufdecken!

Test von dynamischen Webseiten

- Unterschiedliche Versionen der gleichen Browser-Type sind vorteilhaft (Erkennen Versions-spezifischer Browser-Bugs – speziell bei alten Versionen von *Opera*)
- Zusätzlich: **Webseiten-Validierungs-Funktionen oder Services verwenden!**