

Online Präsentationstool

Sascha Kühndel, Torsten Büchner, Thomas Glase

21. Dezember 2005

Inhaltsverzeichnis

1 Einführung	1
1.1 Aufgabenstellung	1
1.2 Motivation	1
1.3 Funktionalität	2
2 Konzept	3
2.1 Folienstruktur	3
2.2 Technisches Konzept	4
3 Formate	5
3.1 XML-Format	5
3.2 Importfunktion	5
3.3 Das Oasis Format	6
3.4 Konvertierung	6

1 Einführung

1.1 Aufgabenstellung

Ziel dieses Projektes ist die Erstellung eines Online Präsentationstools welches im tele-TASK System [8] integriert werden soll. Die Anforderung einer Umsetzung ausschliesslich für den Browser ist dadurch von vornherein gegeben. Dabei sollen die grundsätzlichen Funktionalitäten sowie das Look-And-Feel der üblichsten Präsentationswerkzeuge PowerPoint und OpenOffice imitiert werden, so dass der Nutzer möglichst wenig davon merkt, in einem Browser zu arbeiten. Dafür ist es notwendig, neben den Grundfunktionalitäten der Folienpräsentation, die am häufigsten gebrauchten Features, wie das Blättern der Folien per Tasten- und Mausclick, animierte Folienwechsel sowie das Einbinden unterschiedlicher Medientypen in den Browser zu portieren.

1.2 Motivation

Die bisherige Umsetzung der Vorlesungspräsentation im tele-TASK System basiert auf Videostreams. So werden auch die Folien abgefilmt und nur als Stream abgelegt. Dies hat zwar den Vorteil, dass die Animationen und der Ablauf der Präsentation zusammen mit dem Video des Vortragenden leicht abgespeichert und wiederabrufbar sind, jedoch bringt diese Lösung einige Nachteile mit sich.

Präsentationsfolien bestehen üblicherweise zum grössten Teil aus Text und einigen Grafiken. Audio und Videoelemente sind möglich, werden aber eher weniger häufig genutzt.

Deshalb ist zur Speicherung von Präsentationsfolien ein Videostream ein grundsätzlich ungeeignetes Format, gegen das vor allem die mangelnde Flexibilität spricht. Einmal aufgenommen sind Inhalt und Form der Präsentation nicht mehr zu ändern, Suchfunktionen sind nur mit grossem Aufwand möglich und die visuelle Qualität leidet unter der Komprimierung der Videodaten, welche trotzdem mehr Speicherplatz in Anspruch nehmen als die ursprünglichen Folien.

1.3 Funktionalität

Auf Grund der Beschränkungen, die sich den Projektteilnehmern durch die angestrebte Platzierung des Tools im Browser entgegenstellen, können nicht sämtliche Features der genannten Präsentationstools portiert werden. Es werden daher zunächst folgende Funktionalitäten umgesetzt:

- Etablierte Präsentationsformate sollen importiert werden können, so dass der Nutzer zur Erstellung seiner Folien nicht gezwungen wird, ein anderes als sein gewohntes Tool zu verwenden.
- Neben reinem Text sollen verschieden Medientypen, wie zum Beispiel Grafiken und Videos eingebunden werden können. Die grundlegenden und häufig verwendeten Animationen aus PowerPoint und OpenOffice zum Folienwechsel sollen unterstützt werden.
- Der Ablauf einer Präsentation soll zwecks einer möglichen, späteren automatischen Wiedergabe in einem geeigneten Format abgespeichert werden. Ein Logging der Aktionen des Vortragenden ist dazu notwendig.
- Eine höchstmögliche Kompatibilität für die üblichsten Browser, mindestens Internet Explorer 6 und Firefox soll gewährleistet werden.
- Die Auflösung der Präsentation soll variabel sein, Schrift- und Bildgrößen müssen also bei einer Grössenänderung des Anzeigefensters dynamisch skaliert werden.

Optional ist geplant, zusätzlich zur dynamischen Version des Tools, eine statische Version den Nutzern bereitzustellen, welche kein Javascript zulassen wollen. Zur Unterstützung der genannten dynamischer Aktivitäten ist dieses nämlich unabdingbar.

Darüber hinaus wäre eine White-Board Funktion denkbar. Das heisst, dass der Nutzer mit der Maus Notizen, Skizzen u.ä. an den Folien machen kann. Diese Notizen sollten, analog zum Ablauf mit abgespeichert und wiederabrufbar sein. Schliesslich wäre die Möglichkeit der Aufnahme und Wiedergabe einer Tonspur, insbesondere dem gesprochenen Wort des Vortragenden synchron zum Präsentationsablauf, wünschenswert und würde das Tool als Online Präsentationswerkzeug sinnvoll komplettieren.

2 Konzept

2.1 Folienstruktur

Die Folien werden in zwei Teile aufgegliedert, dies lehnt sich an die schon bestehenden Präsentationstools an (wie PowerPoint und OpenOffice). Einmal in den Inhalt der Folie, und einmal in das Design der Folie. Das Design wird in dem so genannten „Folienmaster“ (oder auch „Master-Folie“) abgelegt. Der Inhalt wird in der „Inhaltsfolie“ abgelegt. Diese Aufteilung ist sinnvoll, weil das Design fast aller Folien gleich ist. Meist hat nur das Titelblatt ein anderes Design als der Rest der Präsentation. Manche haben vielleicht noch ein oder zwei zusätzliche Folien-Designs, aber mehr ist äußerst selten. Dem gegenüber steht eine große Anzahl an Inhaltsfolien. Damit stellt die Aufgliederung eine Verringerung des Speicherplatzverbrauches, eine Verbesserung des Traffics und eine Vereinfachung der Eingabe dar (da die Designs dadurch nur einmal erstellt werden müssen). Der Master definiert die Struktur (Abbildung 1) einer Folie. Er enthält:

- ein Hintergrundbild
- Bereiche, welche mit dem Inhalt gefüllt werden
- die Formatierung der Bereiche
 - Schrifttype
 - Schriftgröße (relativ zur Standard-Schriftgröße)
 - Farbe
 - Dekoration (unterstrichen, kursiv)

Eine Präsentation kann auch mehrere Folienmaster haben. Die Inhalt-Folien ent-

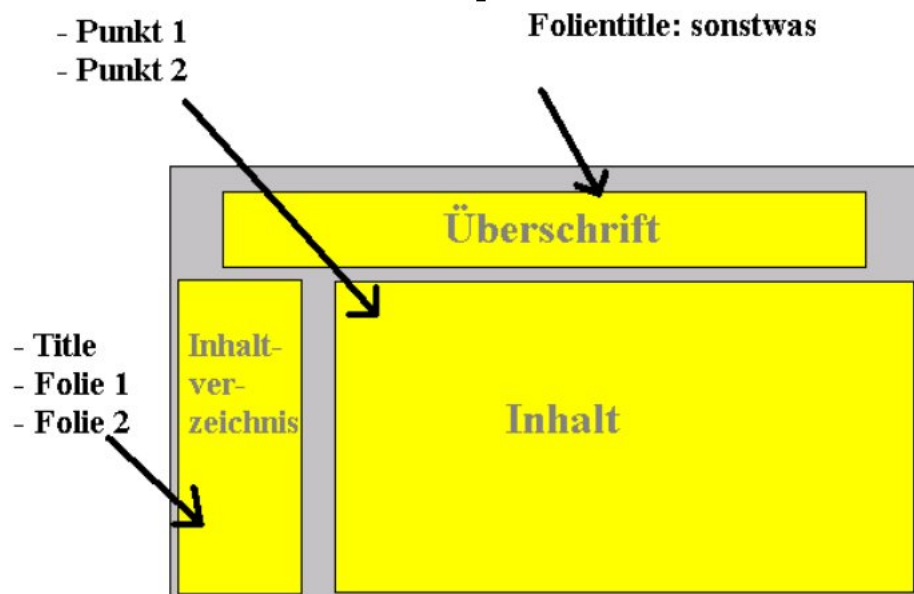


Abbildung 1: Master

halten nur die Inhalte der Folie. Das können Texte, Grafiken oder auch andere Medienstreams sein. Es können natürlich nicht alle möglichen Medien eingebunden werden. Die natürliche Beschränkung ist Kompatibilität mit den Browsern. Des Weiteren ist es möglich, inhaltspezifische Designspezifikationen auch hier vorzunehmen. Dies ist zwar eine Aufweichung einer strikten Trennung von Inhalt und Design, macht aber Sinn, da es öfter mal notwendig ist ein Wort oder einen Satz hervorzuheben. Eine Auslagerung dieser Funktionalität in die Master würde einen erheblichen Aufwand für den Endnutzer bedeuten.

2.2 Technisches Konzept

Aufgrund der Aufgabenstellung wird das Softwaresystem in einer Server-Client-Struktur umgesetzt. In dieser Struktur gibt es zwei Möglichkeiten. Einmal einen zustandsbehafteten Server und einmal ein zustandslosen. In unserem Modell handelt es sich um einen zustandslosen Server. Der Server hat damit „nur“ die Aufgabe die Folien bereitzustellen. Der Client muss die Folien gezielt anfordern. Dies hat den Vorteil, dass der Server keine Daten über die einzelnen Verbindungen hinweg speichern muss, allerdings muss der Client dem Server immer ganz genau mitteilen, was er benötigt.

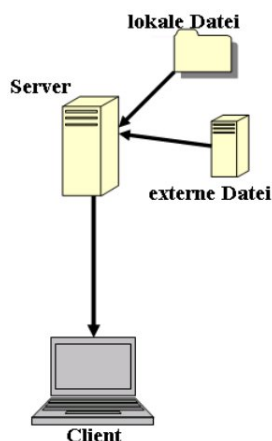


Abbildung 2: Kommunikation

Ablauf einer Präsentation:

- 1. Der Client fordert die Startseite an:** Der Client muss übermitteln welche Präsentation er anzeigen möchte, und dass es sich um die Startseite handelt. Dies wird als Argument (GET-Methode) mit der URL übermittelt. Es reicht also aus, einen normalen Link zu erstellen (`presentation.php?mode=master&file=vortragX.xml`). Darauf sendet ihm der Server eine HTML-Seite zurück. Diese muss alles enthalten, was zur Datenübertragung dient und was wiederkehrend benötigt wird. Dazu gehört die JavaScript-Datei mit den Methoden zu Datenübertragung, zum Einfügen der Inhaltsfolien in die Master, und den Methoden um die Folienaktivitäten durchzuführen. Dazu werden alle Masterfolien mit übertragen.
- 2. Der Client fordert wiederholt die einzelnen Folien an:** Die einzelnen

Folien werden als ein XML-Dokument übertragen. Sie enthält die ID des Masters, in dem die Inhaltfolie angezeigt werden soll.

3. Der Client beendet die Präsentation: Der Client beendet einfach die Präsentation. Der Server muss nicht extra informiert werden.

Dies ermöglicht das Verändern des Folieninhaltes, ohne dass die ganze Seite neugeladen werden muss. Es entspricht der Konzeption von AJAX. Allerdings haben wir uns gegen AJAX entschieden. Dafür gibt es zwei Gründe:

- AJAX liegt bis jetzt noch nicht in einem fertigen (stable) Paket vor. Es gibt zwar viele AJAX-Development-Projekte, aber alle stecken noch in der Beta-Phase.
- AJAX ist ein sehr mächtiges Tool, und für unsere Bedürfnisse überdimensioniert.

Wie kommt der Server an die Präsentationsdatei? Dafür gibt es zwei Möglichkeiten. Einmal: die Datei mit der Präsentation liegt auf dem Server selbst, dann kann einfach über das lokale Dateisystem darauf zugegriffen werden (Achtung: der Zugriff wird mit dem User des Webservers durchgeführt!!!). Die zweite Möglichkeit ist, dass die Präsentationsdatei auf einem anderen Webserver im Netzwerk/ Internet liegt. Dann greift der Präsentationsserver über das http-Protokoll auf die Datei zu.

3 Formate

3.1 XML-Format

Zur Datenhaltung benötigen wir ein eigenes Datenformat (Abbildung 3). Hier haben wir uns zu einem XML konformen Format entschieden, da ein solches viele Vorteile mit sich bringt. Zum einen ist XML ein fester und weit verbreiteter Standard des W3-Konsortium, für dessen Verarbeitung viele Tools existieren. Desweiteren ist ein im XML-Standard erstelltes Dokument eine von Maschinen wie von Menschen lesbare Baumstruktur, was die Weiterentwicklung und Erweiterung des Präsentationstools durch Dritte ermöglicht und vereinfacht.

Die Datenhaltung inclusive der Bild- und Multimediadaten soll möglichst mit einer einzigen XML-Datei realisiert werden, um den flexiblen Umgang mit den Daten zu gewährleisten. In der Dokumentstruktur sind die Daten der Master von den Folien getrennt gespeichert.

3.2 Importfunktion

Um den Import einer schon vorhandenen Präsentation zu ermöglichen, muss eine Möglichkeit gefunden werden, verbreitete Formate wie Microsoft Powerpoint (*.ppt) oder OpenOffice bzw. StarOffice(*.odt) in das eigene XML-Dateiformat umzuwandeln. Das derzeit wohl am häufigsten benutzte Format von Microsofts Powerpoint ist allerdings eher schlecht geeignet. Die Daten werden hier in von Menschen unlesbarer Form abgespeichert. Auch die Exportfunktionen der Software lassen sich nur sehr schwierig verwenden. Im Gegensatz dazu setzen die Programme OpenOffice und Staroffice auf ein Datenformat, das nicht nur standardisiert und sogar von der EU im Rahmen ihres IDA-Programms (Interchange

```

- <presentation>
  <title>Testpresentation</title>
  - <masters>
    + <master id="1"></master>
    - <master id="2">
      <background type="intern">background2</background>
      + <body></body>
    </master>
  </masters>
- <folien start="1">
  + <folie id="1" master="1" title="Deckblatt"></folie>
  - <folie id="2" master="2" title="Folie 1">
    + <fill id="menu"></fill>
    + <fill id="body"></fill>
    - <do>
      + <step></step>
      + <step></step>
      + <step></step>
    </do>
  </folie>
  + <folie id="3" master="2" title="Folie 2"></folie>
  + <folie id="4" master="1" title="Päsentation"></folie>
</folien>
- <images>
  + <image id="background1" type="jpeg"></image>
  + <image id="background2" type="jpeg"></image>
  + <image id="animaco" type="png"></image>
</images>
</presentation>

```

Abbildung 3: die XML Datei

of Data between Administrations) empfohlen wird [6], sondern auch XML konform ist: das OASIS OpenDocument Dateiformat. Damit ist unserer Meinung nach dieses Format besser geeignet, auch weil die Software OpenOffice unter der LGPL steht und zudem relativ zuverlässig PowerPoint Folien in das OASIS Format umwandeln kann. Zudem hat auch Microsoft angekündigt in Zukunft auf ein freies XML Format zu setzen.

3.3 Das Oasis Format

Das Oasis Format ist ein ZIP-Archiv. Dieses Archiv enthält einen Verzeichnisbaum mit hauptsächlich 4 XML Dateien, die getrennt Inhalt, Programmeinstellungen (z.b. Standarddrucker), Metainformationen (Autor, Titel) sowie die Syles des Dokuments enthalten. In OpenDocument werden Grafiken in einem SVG[7] ähnlichen Code gespeichert. Weiterhin wird von folgenden bereits bestehenden Standards Gebrauch gemacht: HTML[2], XSL[12], SMIL[5], XLink[11], XForms[10], MathML[3] und Dublin Core[1]. [4]

3.4 Konvertierung

Die Konvertierung des OASIS Open Document Formates soll mit Hilfe eines XSLT/XPath Stylsheets in das eigene XML konforme Format umgewandelt werden. XSLT (Extensible Stylesheet Language Transformation) ist Teil der XSL Sprachfamilie die neben XSLT noch XSL-FO für Formatierungen [9]. XSLT ermöglicht die Umwandlung einer XML-Datei in ein beliebiges Format wie html, eine binäre Datei, pdf oder eben in eine weitere XML-Datei. XPath wird dazu benutzt einzelne Elemente eines XML-Dokumentes zu adressieren. Die Quell-XML Datei wird dabei als logischer Baum betrachtet dessen Elemente mit Hilfe

von Templates und XPath basierter Pattern in den Ziel-Baum umgewandelt. Der Vorteil von XSLT liegt neben der leichten Erweiterbarkeit in der Tatsache, daß es in PHP5 eine XSLT API gibt, die bereits mehrere Bibliotheken unterstützt. Da wir in unserem Präsentationstool nicht alle Funktionen von OpenOffice darstellen können wird während der Konvertierung in das eigene Präsentationsformat eine Filterung der unterstützten Funktionen stattfinden. Dies ermöglicht einen einfacheren Aufbau dessen und damit die eventuelle einfache Nachbearbeitung bzw. Erstellung von Hand.

Literatur

- [1] . Dublin Core - <http://dublincore.org/>.
- [2] . HTML - <http://www.w3.org/TR/html/>.
- [3] . MathML - <http://www.w3.org/TR/mathml/>.
- [4] . oasis - <http://xml.coverpages.org/ni2005-01-04-a.html>.
- [5] . SMIL - <http://www.w3.org/TR/smil/>.
- [6] . sun - <http://de.sun.com/products/software/star/staroffice/index.jsp>.
- [7] . SVG - <http://www.w3.org/TR/SVG/>.
- [8] . tele-Task - <http://www.tele-task.de>.
- [9] . wikipedia - <http://www.wikipedia.de>.
- [10] . XForms - <http://www.w3.org/TR/xforms/>.
- [11] . XLink - <http://www.w3.org/TR/xlink/>.
- [12] . XSL - <http://www.w3.org/TR/xsl/>.