

Lars-Projekt IT Service Workshop

# **Projektseminar IT Workshop**

## **SS 2009**

### **Semesterbericht**

Hochschule der Medien Stuttgart

**Herausgegeben von  
Wolf-Fritz Riekert und  
Belmondo Kovac**

Stuttgart, Juli 2009



## Vorwort – unbedingt lesen!

Dies ist das Musterdokument zur Erstellung der Dokumentation für ein Teilprojekt im Rahmen des Projektseminars IT Workshop. **Das zweite Kapitel soll als Schema verwendet werden. Bitte ersetzen Sie den dort stehenden Text durch Ihren eigenen Text.**

Das dritte Kapitel enthält Hinweise zur Formatierung Ihres Texts. Diese sind weitgehend aus der „Dokumentvorlage für Diplomarbeiten und andere wissenschaftliche Arbeiten“<sup>1</sup> entnommen, die von vielen bereits genutzt wird. In diesem Projektbericht stehen dieselben Formatvorlagen zur Verfügung, nur haben sie ein etwas anderes Aussehen, da der Bericht zweiseitig formatiert ist und als Sammelband mit mehreren Autoren konzipiert ist.

Bitte beachten Sie insbesondere folgende Regeln:

- **Verwenden Sie nur die durch die Formatvorlagen vorgesehenen Formatierungsmöglichkeiten! Insbesondere ändern Sie nicht den Schrifttyp oder die Schriftgröße oder die Seitenränder!**
- Bitte trennen sie Absätze nur durch einmaliges Drücken der Entertaste!
- Bitte formatieren Sie Abbildungen genauso wie in Kapitel 3 angegeben.

Wenn Sie mit der Dokumentation Ihrer Arbeiten fertig sind, soll Kapitel 2 Ihren Text enthalten, Kapitel 3 können Sie löschen, wenn Sie möchten. **Die ersten Seiten (Titel, Vorwort, Inhaltsverzeichnis, Kapitel 1) lassen Sie bitte bestehen. Bitte laden Sie dann Ihren Text auf das dafür vorgesehene Verzeichnis in Live-link hoch (Unterverzeichnis „Bericht“ unserer Lehrveranstaltung). Der Name soll Ihrem Teilprojekt entsprechen, z.B. „slipstream-unattended.doc“.**

Die einzelnen Teile werden dann von mir oder einer von mir beauftragten Person zu einem Gesamtdokument zusammengefügt.

*Wolf-Fritz Riekert*

---

<sup>1</sup> siehe: <http://v.hdm-stuttgart.de/~riekert/theses/#dokumentvorlage>



# Inhaltsverzeichnis

<b>Vorwort – unbedingt lesen!</b> .....	<b>III</b>
<b>Inhaltsverzeichnis</b> .....	<b>V</b>
<b>1 Überblick</b> .....	<b>1</b>
<i>Wolf-Fritz Riekert und Belmondo Kovac</i>	
<b>2 Integration des Flowplayers in WordPress</b> .....	<b>3</b>
<i>Sebastian Hauptelshofer, Patrick Geiger, Martin Wörz</i>	
2.1 Aufgabenstellung.....	3
2.2 Überlegungen und Systemanalyse .....	4
2.3 Implementierung des Flowplayer Plugins .....	13
2.4 Probleme .....	18
2.5 Ergebnis .....	19
Quellen: .....	20

## Abbildungsverzeichnis

Abbildung 1: WP Medienverwaltung - Neue Datei hochladen .....	6
Abbildung 2: Ausschnitt der Datenbanktabelle "wp_posts".....	7
Abbildung 3: .....	12
Abbildung 4: WordPress Flowplayer Demo .....	12
Abbildung 5: Write a Plugin .....	14
Abbildung 6: Erweiterung des SaiWeb Flowplayer- Beschreibung.....	15
gelangt man zum Captions Editor. Dieses Menü wurde wie oben beschrieben über Hooks zum WordPress System hinzugefügt. Hier kann der Anwender intuitiv Untertitel zu den aus der Mediathek ausgewählten Medien hinzufügen. Es ist ihm dabei möglich zu <b>jedem beliebigen Zeitpunkt</b> <b>innerhalb einer Videodatei Text einzublenden</b> .....	18

# 1 Überblick

*Wolf-Fritz Riekert<sup>1</sup> und Belmondo Kovac<sup>2</sup>*

Hier kommt noch ein Übersichtskapitel rein.

---

<sup>1</sup> Dr. Wolf-Fritz Riekert <riekert@hdm-stuttgart.de> ist Professor an der Hochschule der Medien Stuttgart. Er ist zugleich wissenschaftlicher Leiter des Informationstechnischen Service (IT-Service) der Fakultät Information und Kommunikation der Hochschule. Er unterrichtet die Gebiete Computernetze, Datenbanken und Webapplikationen in den Studiengängen Informationswirtschaft (Diplom und Master) und Information Systems (Bachelor).

<sup>2</sup> Belmondo Kovac ist als technischer Mitarbeiter im IT-Service der Fakultät Information und Kommunikation für das Systems Management der Arbeitsplatzcomputer in der Fakultät verantwortlich.



## 2 Integration des Flowplayers in WordPress

*Sebastian Hauptelshofer, Patrick Geiger, Martin Wörz<sup>1</sup>*

In diesem Projekt wird eine vollständige Integration von Flowplayer, einem auf Flash basierenden Abspielprogramm für Internet-Video-Formate, in das quell-offene Blogging-System WordPress durchgeführt. Dem Nutzer wird dadurch die Möglichkeit gegeben, auf einfache Weise den Flowplayer in eigene Artikel einfügen zu können, ohne dass er große Programmierkenntnisse benötigt. Dadurch hat der Nutzer die Möglichkeit, neben Bildern und Texten auch Videos einbinden zu können. Dies ist so bisher mit Wordpress 2.7.1 bisher nicht möglich. Der Name des Plugins „pash-m“ steht für **P**atrick Geiger, **S**ebastian Hauptelshofer und **M**artin Wörz.

### 2.1 Aufgabenstellung

Ausgangspunkt sind zwei unabhängig voneinander funktionierende Systeme. Das Open Source Blogging System WordPress auf der einen Seite, welches durch Module erweitert werden kann und auf der anderen Seite der Flowplayer<sup>2</sup>, der auch der GPL unterliegt. Der Flowplayer lässt sich eingebettet in jeder Webseite nutzen.

Eine besondere Eigenschaft liegt im Flowplayer, da dieser durch Plugins erweitert werden kann.

Das WordPress System verfolgt hauptsächlich das Ziel dem Nutzer eine leicht zu bedienende und moderne Webblogging-Software („Content-Management-System“) bereit zu stellen.<sup>3</sup>

---

<sup>1</sup> Sebastian Hauptelshofer <[sh112@hdm-stuttgart.de](mailto:sh112@hdm-stuttgart.de)>, Patrick Geiger <[pg018@hdm-stuttgart.de](mailto:pg018@hdm-stuttgart.de)> und Martin Wörz <[mw088@hdm-stuttgart.de](mailto:mw088@hdm-stuttgart.de)> studieren im 4. Semester Information Systems / E-Services an der Hochschule der Medien Stuttgart.

<sup>2</sup> Piirainen, T. (10. 06 2009)

<sup>3</sup> wordpress.org. (25. 06 2009)

Im Gegensatz zu gewöhnlichen Content-Management-Systemen, die den Fokus auf die Dokumentenverwaltung legen, setzt WordPress System auf Usability und Simplizität.

Ziel der Integration des Flowplayers in das WordPress ist das Erstellen eines Plugins („Flowplayer Plugin“) für das Blogging System WordPress mit folgenden Anforderungen:

1. die Möglichkeit innerhalb eines Blog-Eintrags ein Video aus der zentralen WordPress-Mediathek einzubinden.
2. die Integration eines Flowplayer-Formulars zum Ansprechen und Auswählen der Medien innerhalb der Blogging Umgebung, genauer gesagt im WYSIWG<sup>4</sup>- Editor von WordPress.
3. das Einblenden von Captions (Untertiteln) zu einem Video aus der Mediathek<sup>5</sup>.
4. eine Oberfläche zum Erstellen und Bearbeiten der Untertitel.

## 2.2 Überlegungen und Systemanalyse

Um festzustellen wie das WordPress-System mit einem abzuspielenden Medium umgeht und wie Medien verwaltet werden, müssen die Standardprozesse des Systems durchlaufen und analysiert werden. Gleichzeitig müssen die technischen Anforderungen an dem Flowplayer und dem sich daraus ergebenden Flowplayer-Plugin für WordPress überprüft werden.

Die Einstiegsfrage lautet daher, wie Medien im WordPress-System abgelegt werden können. Daraus ergibt sich der erste zu analysierende Schritt: Der Ablauf des Hinzufügens neuer Medien in die WordPress-Mediathek.

---

<sup>4</sup> WYISYG: **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et. Ein Dokument wird während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe über ein anderes Gerät, z. B. einem Drucker, aussieht.

<sup>5</sup> Mediathek: Benutzeroberfläche für die Wordpress-Mediendatenbank

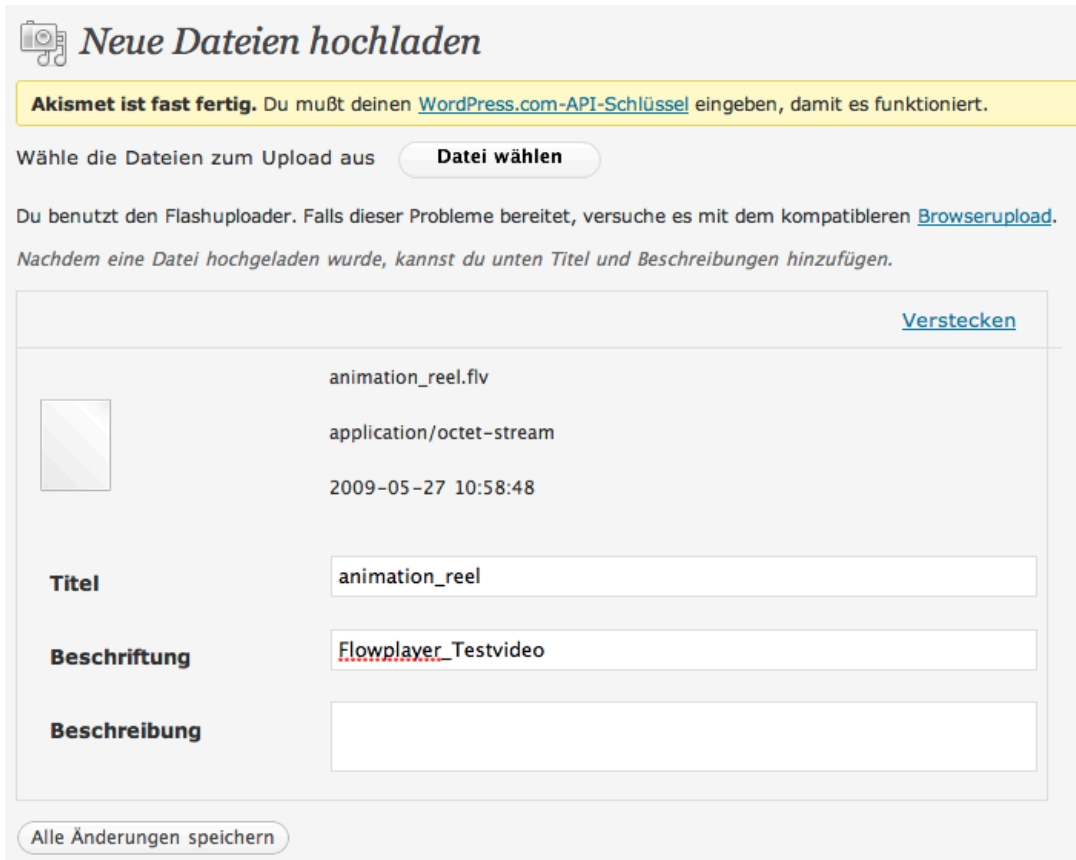
Erstellung des WordPress Flowplayer Plugins „pash-m“

### 2.2.1 Analyse des Standardprozesses „Medienverwaltung in WordPress“

Generell können Medien in WordPress über die Administrationsoberfläche innerhalb der Mediathek hochgeladen werden. Somit wurde der Vorgang „Neue Dateien hochladen“ durchlaufen und analysiert. Als Beispieldaten wurde einem Video, das wir zum Testen zu Verfügung gestellt bekommen haben<sup>6</sup> der Titel “animation\_reel” zugewiesen und “Flowplayer\_Testvideo” als Beschriftung verwendet. Als Beschreibung wurde „Dies ist ein Beispielvideo, das zuvor mit dem Flowplayer getestet wurde.“ verwendet. Nachdem nun ein Beispielvideo hochgeladen wurde, konnte im Bearbeitungsmodus “Medien bearbeiten” bei genauerer Untersuchung der URLs des Webbrowsers festgestellt werden, dass diese Datei eine eindeutige Identifikationsnummer „id“ („?attachment\_id=306“) zugewiesen bekam (siehe **Abbildung 1: WP Medienverwaltung - Neue Datei hochladen**).

---

<sup>6</sup> Quelle: Steckel, M. (Produzent), Steckel, M. (Autor), & Steckel, M. (Regisseur). (2008). *Animation Showreel*.



**Neue Dateien hochladen**

**Akismet ist fast fertig.** Du mußt deinen [WordPress.com-API-Schlüssel](#) eingeben, damit es funktioniert.

Wähle die Dateien zum Upload aus

Du benutzt den Flashuploader. Falls dieser Probleme bereitet, versuche es mit dem kompatibleren [Browserupload](#).

*Nachdem eine Datei hochgeladen wurde, kannst du unten Titel und Beschreibungen hinzufügen.*

[Verstecken](#)


	animation_reel.flv application/octet-stream 2009-05-27 10:58:48
<b>Titel</b>	<input type="text" value="animation_reel"/>
<b>Beschriftung</b>	<input type="text" value="Flowplayer_Testvideo"/>
<b>Beschreibung</b>	<input type="text"/>

Abbildung 1: WP Medienverwaltung - Neue Datei hochladen

Daraus konnte der Schluss gezogen werden, dass dieser Eintrag unter selbiger "id" in der von WordPress verwendeten MySQL-Datenbank hinterlegt sein muss. Bei der Analyse der Datenbank wurde festgestellt, dass der oben genannte Eintrag in der „Posts“-Tabelle des WP-Systems abgelegt wird. Erkennbar war dies durch Vergleich des Datums des so genannten Posts in der Tabelle „wp\_posts“. Da nun offensichtlich diese Tabelle für das vorhaben interessant ist, musste die Struktur dieser Tabelle genau untersucht werden. Die Struktur der Tabelle „wp\_posts“ sieht folgendermaßen aus:

## Erstellung des WordPress Flowplayer Plugins „pash-m“

	Feld	Typ
<input type="checkbox"/>	<b>ID</b>	bigint(20)
<input type="checkbox"/>	<b>post_author</b>	bigint(20)
<input type="checkbox"/>	<b>post_date</b>	datetime
<input type="checkbox"/>	<b>post_date_gmt</b>	datetime
<input type="checkbox"/>	<b>post_content</b>	longtext
<input type="checkbox"/>	<b>post_title</b>	text
<input type="checkbox"/>	<b>post_category</b>	int(4)
<input type="checkbox"/>	<b>post_excerpt</b>	text
<input type="checkbox"/>	<b>post_status</b>	varchar(20)
<input type="checkbox"/>	<b>comment_status</b>	varchar(20)
<input type="checkbox"/>	<b>ping_status</b>	varchar(20)
<input type="checkbox"/>	<b>post_password</b>	varchar(20)
<input type="checkbox"/>	<b>post_name</b>	varchar(200)
<input type="checkbox"/>	<b>to_ping</b>	text
<input type="checkbox"/>	<b>pinged</b>	text
<input type="checkbox"/>	<b>post_modified</b>	datetime
<input type="checkbox"/>	<b>post_modified_gmt</b>	datetime
<input type="checkbox"/>	<b>post_content_filtered</b>	text
<input type="checkbox"/>	<b>post_parent</b>	bigint(20)
<input type="checkbox"/>	<b>guid</b>	varchar(255)
<input type="checkbox"/>	<b>menu_order</b>	int(11)
<input type="checkbox"/>	<b>post_type</b>	varchar(20)
<input type="checkbox"/>	<b>post_mime_type</b>	varchar(100)
<input type="checkbox"/>	<b>comment_count</b>	bigint(20)

Abbildung 2: Ausschnitt der Datenbanktabelle "wp\_posts"

Da nicht alle Datenbankfelder benötigt werden, stellt sich die Frage, welche für das Plugin relevant sind und welche nicht. Der Flowplayer lässt nur das Abspielen bestimmter Medien-Formate zu. Um den Rahmen dieser Arbeit nicht zu sprengen wurde demonstrativ das meistverwendete (laut Angaben der Seite [www.flowplayer.org](http://www.flowplayer.org)<sup>7</sup> zu 97,7%) Internet-Format „FLV“, also Flash verwendet. Neben Flash unterstützt dieser Player auch noch H.264-Formate im Mpeg2- und Mp4-Containerformat<sup>8</sup>.

Für dieses Projekt ist wesentlich, wie Medien aus der Datenbank zurückgegeben werden, die im Flash-Format dort hinterlegt sind.

---

<sup>7</sup> Quelle: Piirainen, T. (10. 06 2009)

<sup>8</sup> Quelle: Piirainen, A. (23. 06 2009)

Die Endung von Flash-Files lautet „flv“. Des Weiteren sollen die zugehörigen Titel und Pfade abgefragt werden, die eine Auskunft darüber geben, wo das Medium abgespeichert ist.

Somit ergaben sich folgende Felder als relevant:

- „ID“ - die eindeutige Identifikationsnummer des Posts bzw. in dem Fall Mediums in der Datenbank
- „post\_title“ - der Titel des Posts, der auch über die WordPress Oberfläche verändert werden kann
- „post\_excerpt“ - die Beschreibung zu einem Medium
- „guid“ - der externe Link außerhalb der Datenbank und innerhalb der Ordnerstruktur des Servers zu der Mediendatei (wird automatisch von WordPress generiert, sofern es sich um eine Mediendatei handelt)
- „post\_mime\_type“ - hier wird der Mimetype gespeichert. Der Mimetype ist für die Kategorisierung von Dateien zuständig. In diesem Fall handelt es sich um ein Streaming-Medium (*application/octet-stream*).

Durch den Mimetype wird der interpretierenden Software mitteilt, um welchen Datentyp es sich handelt. Sofern ein Mimetype nicht bekannt ist, wird ein Standard-Mimetype verwendet. Somit kann WordPress auch keine zugehörige Standardsoftware hinterlegen.

Aus dieser Erkenntnis konnte anschließend eine SQL-Anweisung erhoben werden, die später für das Plugin notwendig ist. Die erste Anforderung ist die Ausgabe einer Liste mit allen Flash-Formaten. Um eine Liste mit allen flv-Videodateien aus der WordPress-Mediathek zu erhalten, muss die in Abbildung 3 dargestellte SQL-Anweisung ausgeführt werden.

Um zu gewährleisten, dass die Groß- und Kleinschreibung beim Durchsuchen der Mediendateien ignoriert wird, musste die SQL Anweisung „LIKE“ verwendet werden. Mit **LIKE** '%flv' werden alle ‚Flv‘, FLV, flv, -Endungen und weitere angenommen). Da später durch eine Auswahl eine ganz bestimmte Mediendatei an Flowplayer übergeben werden sollte, muss die eindeutige Nummer des Medium (ID) in der SQL-Anweisung berücksichtigt werden. Als Beispiel wurde das zuvor hochgeladene Medium mit der ID „306“ ausgewählt. Anbei

Erstellung des WordPress Flowplayer Plugins „pash-m“

muss noch vermerkt werden, dass dieses Videofile separat mit Flowplayer erfolgreich getestet wurde.

Somit muss die zugehörige SQL- Anweisung für das Medium “306“ dann wie folgt lauten:

```
SELECT `ID`, `post_title`, `post_excerpt`, `guid`, `post_type`  
FROM `wp_posts`  
WHERE `ID` = 306
```

Auf dieser Grundlage wurde ein verlässliches Fundament gewonnen, durch welches Anforderungen an das Plugin gestellt werden konnten. Es ist möglich:

- relevante, für den Flowplayer abspielbare Medien, in der Mediathek zu hinterlegen.
- diese Medien in einer Liste aus der WordPress-Datenbank wieder auszugeben.
- ein bestimmtes Medium durch die Angabe der „ID“ auszuwählen um den Speicherort herauszufinden.

## 2.2.2 Funktionsanalyse Flowplayer

Der Flowplayer gibt Videos wieder, die auf der lokalen Festplatte oder auf dem eigenen Webserver gespeichert sind bzw. ermöglicht das Einbinden von Clips fremder Webservern auf einer Website.

Er bietet einige Einstellmöglichkeiten. Zum Beispiel ist es möglich, mehrere Videos über einen Stream laufen zu lassen. Des Weiteren kann dem Zuschauer Untertitel zeitgleich zum Video eingeblendet werden. Die Captions (Untertitel) können über eine XML-Datei eingebunden werden.

Ein weiterer Vorteil besteht in der Unterstützung des "Pseudostreamings". Das bedeutet, dass nicht das komplette Videofile in seiner gesamten Länge her-

untergeladen werden muss wenn eine bestimmte Stelle des Videos ausgewählt wird. Beim Pseudostreaming muss kein Streaming-Server installiert sein.

Hinzu kommt, dass der Flowplayer quelloffen und durch Plugins erweiterbar ist. Das für die Nutzer sichtbare Wiedergabemodul enthält die klassenüblichen Bedienelemente wie Lautstärke, Vollbild, Play und Pause.

### 2.2.3 Umfeldanalyse

Da nun die wichtigsten Funktionen für die Grundlage dieses Projektes überprüft worden sind, konnte mit der Umfeldanalyse um das System WordPress begonnen werden. Um in diesem Projekt bei der Idee des Open Source-Gedankens zu bleiben, wurden auf die Webseite von [wordpress.org](http://wordpress.org) nach offiziellen Plugins gesucht, die einen ähnlichen Gedanken verfolgen. Gesucht wurde nach Begriff Flowplayer auf der Plugin Unterseite von [wordpress.org](http://wordpress.org)<sup>9</sup>.

Diese Suche ergab 5 Treffer:

1. Flowplayer for WordPress
2. WordPress Flowplayer
3. wordTube Flowplayer
4. Hana FLV Player
5. ILC FLVBox

Eine genauere Untersuchung ergab, dass keines der Plugins die WordPress integrierte Mediathek nutzt.<sup>10</sup> Somit würden diese Plugins am Ziel der späteren Kompatibilität vorbeilaufen.

Das „wordTube Flowplayer“-Plugin ist das einzige Plugin, welches auf eine Mediathek anstatt einer direkten URL zugreift. Jedoch ist diese Mediathek eine unabhängige Entwicklung vom WordPress-System.

Da nun keines der Plugins der Anforderung gerecht wird, bleibt nur die Modifizierung des Plugins, das am ehesten der Zielsetzung gerecht wird.

---

<sup>9</sup> Quelle: [wordpress.org](http://wordpress.org). (25. 06 2009)

<sup>10</sup> Quelle: Mullenweg, M. (26. 06 2009)

## Erstellung des WordPress Flowplayer Plugins „pash-m“

Um herauszufinden, welche Plugins am geeignetsten sind, wurden alle Plugins, die im WordPress System installiert sind, aktiviert. Danach wurde getestet, ob dieses Plugin die Funktionalität auch erfüllt. Das Ergebnis spricht für sich: Nicht alle Plugins haben einwandfrei funktioniert. Ein Grund dafür ist, dass sich Plugins in einem vorstadiums-ähnlichen Entwicklungsstand befinden. Zum Beispiel befindet sich das „wordTube FlowPlayer“-Plugin im „Beta“-Status. Bei den Tests funktionierte das „SaiWeb“ Plugin am problemlosesten. Außerdem beschäftigte sich David Busby, der Autor mit der Implementierung des Plugins schon seit über 6 Monaten.

Bei dem Plugin „SaiWeb Flowplayer for WordPress“ handelt es sich um ein Plugin, das den Flowplayer in das WordPress System integriert.

Hier folgt die Analyse über gemeinsame Schnittmengen in der Funktionalität dessen was für dieses Projekt verwendet wird.

### **2.2.4 Funktionsanalyse „SaiWeb Flowplayer for WordPress“**

Um das Plugin nutzen zu können, muss herausgefunden werden, wie dieses in WordPress verwendet wird.

Hier gibt es eine Applikationsoberfläche, mit der unter Anderem die Standardgröße eines Mediums oder die Farbe des Players konfiguriert werden kann.

Ein Video kann im Saiweb-Plugin über folgende Befehlszeile eingefügt werden:

```
[Flowplayer=URL, 640, 480]
```

Das Plugin wertet diese Befehlszeile aus (Parsing) und wandelt sie in flowplayer-tauglichen HTML-Code um, sprich er übersetzt sie in die entsprechende Struktur.

Das Plugin wurde in NetBeans nach Funktionen, Methoden und Klassen analysiert. Außerdem wurde analysiert, wie das Plugin in WordPress angesprochen wird. Es legt in der Navigationsstruktur einen Extrapunkt an, damit man das Plugin aufrufen kann. Dadurch weiß der Nutzer, wo er das Plugin in WordPress

benutzen

kann.

WP Hooks

```
add_action('wp_head', 'flowplayer_head');  
add_filter('the_content', 'flowplayer_content');  
add_action('admin_menu', 'flowplayer_admin');
```

Abbildung 3: WP Hooks: add\_action und add\_filter

WP Hooks sind Befehle in WordPress, in der man an eine gewünschte Position des Blogs springen kann, um dort entweder etwas einzufügen oder etwas verändern zu können. „add\_action“- Befehle dienen dazu, an dieser Stelle eine Funktion aufzurufen, die üblicherweise ein Formular darstellt. „add\_filter“- Befehle hingegen verändern den an der Stelle angegebenen Inhalt.

In diesem Fall wurde ein mit `add_action('admin_menu', 'flowplayer_admin')` ein Menüpunkt bzw. Link für das Flowplayer-Plugin in der Navigationsleiste der Administrationsoberfläche abgelegt.

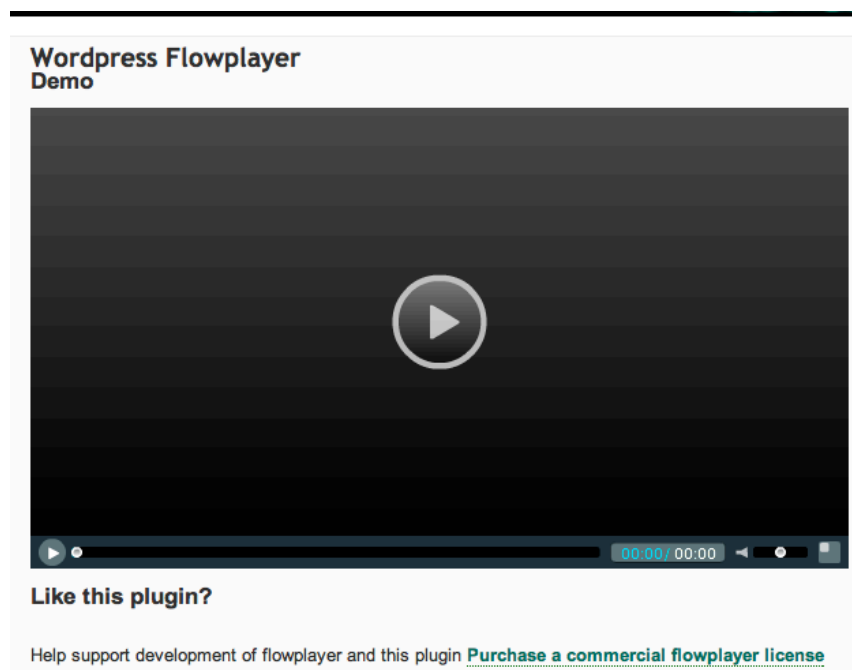


Abbildung 4: WordPress Flowplayer Demo

Erstellung des WordPress Flowplayer Plugins „pash-m“

### **2.2.5 Funktionsanalyse des Captions Skripts von Benjamin Bestmann**

Benjamin Bestmann, ein Student der Hochschule der Medien, hat für dieses Projekt ein PHP-Skript zur Verfügung gestellt. In wie weit dieses Skript als Grundlage für die Entwicklung des „pash-m“ Plugins verwendbar ist, wurde hier geklärt.

Das Skript besteht aus drei Dateien und einer „Captions“-spezifischen Datenbank. Die Datenbank wird nicht verwendet, weil wir für unser Plugin mit XML-Dateien arbeiten. Es wurde lediglich das auf AJAX basierende Formular genutzt, um die Captions hinzuzufügen und die entsprechende XML-Datei auszugeben. Der Vorteil von AJAX liegt in der dynamischen Erweiterbarkeit des Formulars ohne weiteren Datenaustausch zwischen dem Webbrowser und Server, was mit normalem XHTML nicht möglich ist.

### **2.2.6 Bedarfs- und Funktionsanalyse der WordPress API**

Hier wurde geklärt welche Funktionen und Methoden aus der WordPress API für diese Projekt verwendet werden können.

1. Datenbankaufrufe durch die globale Variable \$wpdb
2. Das Einbinden von Steuerelementen über so genannte Hooks
3. Befehlsaufrufe zum Hinzufügen von Formularen über WordPress spezifische Labels und Fieldsets

## **2.3 Implementierung des Flowplayer Plugins**

In diesem Abschnitt folgt die eigentliche Entwicklung des Plugins. Auf der Seite<sup>11</sup> ist dokumentiert, welche Strukturen ein Plugin hat und wie diese im System angesprochen werden.

Zu Beginn sollte das Plugin im WordPress System im Verzeichnis %WP\_ROOT%/wp-content/plugins/ angelegt sein. Das Erkennungsmuster für ein Plugin ist „Plugin Name: Name“. Sobald dieses zu Beginn in einer Datei

---

<sup>11</sup> Quelle: Writing a Plugin « WordPress Codex. (14. 06 2009)

steht, lädt WordPress diese Datei und zeigt diese in der Plugin-Kategorie auf der Administrationsoberfläche an.

```
<?php
/*
Plugin Name: Name Of The Plugin
Plugin URI: http://URI\_Of\_Page\_Describing\_Plugin\_and\_Updates
Description: A brief description of the Plugin.
Version: The Plugin's Version Number, e.g.: 1.0
Author: Name Of The Plugin Author
```

Abbildung 5: Write a Plugin<sup>12</sup>

Dort legen wir folgende php-Datei an.: „wordpress-flowplayer\_pash-m.php“.

---

<sup>12</sup> Quelle: Writing a Plugin « WordPress Codex. (14. 06 2009)

## Erstellung des WordPress Flowplayer Plugins „pash-m“

Von der Seite lernen wir, wie die Grundstrukturen für die Bezeichnung eines

```
/*
Plugin Name: Flowplayer for WordPress Pash-M
Plugin URI: http://ieservices.de/wordpress/plugins/wordpress-flowplayer
Description: This plugin unions the Flowplayer with the WordPress System.
Comparing to those other plugins, this plugin uses the WordPress media
database as source for the videos. Within this option you are able to use
the common upload function. The project is developed by students at the
Media University Stuttgart. To use the ideas of previous people we used
as origin the plugin from David Busby (URL:
http://saiweb.co.uk/wordpress-flowplayer - http://saiweb.co.uk)
Version: 0.0.1
Author: Patrick Geiger, Sebastian Hauptelshofer and Martin Andreas Woerz
Author URI: http://ieservices.de
*/
/**
 * FlowPlayer for WordPress
 * ©2009 (Patrick Geiger, Sebastian Hauptelshofer and Martin Andreas
Woerz) origin (David Busby)
 * @see http://creativecommons.org/licenses/by-nc-sa/2.0/uk
 */
```

Plugins lauten sollen:

### Abbildung 6: Erweiterung des SaiWeb Flowplayer- Beschreibung

Dort wurde oben stehende php-Datei angelegt: „wordpress-flowplayer\_pash-m.php“

Aufgrund der Tatsache, dass wir das Saiweb Flowplayer Plugin als Grundlage verwendet haben, wurde neben unseren Namen zusätzlich der Name des ursprünglichen Autors, David Bubsy, in die Plugin-Beschreibung hinterlegt. David Bubsy hat sein Plugin unter der Creative Commons Lizenz geschrieben,

welche eine Weiterverwendung des Programmcodes erlaubt, sofern der Originalautor erwähnt bleibt.

Da das System WordPress weit verbreitet ist, stellt sich folgende Frage ob es bereits vorhandene Plugins gibt und wenn ja welche Synergien genutzt werden können. Das Saiweb Plugin unterstützt lediglich die Angabe von direkten URLs (Unified Resource Links) zu den direkten Medien außerhalb der WP-Mediathek. Somit muss das Plugin um die Verknüpfung mit Mediendateien aus der WP-Datenbank erweitert werden.

### 2.3.1 Implementierung der Mediathekenfunktion im Plugin

Bisher konnten mit dem Plugin „SaiWeb Flowplayer for WordPress“ nur Medien mit einer direkten URL an den Flowplayer weitergegeben werden. Um die vorhandenen Medien direkt aus der WordPress Mediathek verwenden zu können, muss das „SaiWeb Flowplayer for WordPress“ Plugin um diese Funktionalität erweitert werden. Bisher wurden die Videos direkt über [Flowplayer=http://domain.com/media.flv, 600, 450] angesprochen.<sup>13</sup>

Ziel war es, einzelne Medien aus der Mediathek von WordPress anzusprechen. Das kann über die ID des Mediums erfolgen.

Das von SaiWeb verwendete Muster zur Einbindung von Videos ist nicht für Erweiterungen nicht flexibel genug und wurde durch ein neues Konzept ersetzt.

Üblicherweise werden div-Container genutzt um Objekte auf einer Webseite zu positionieren. Dieses Konzept wurde hier übertragen, nicht zuletzt weil auch viele Nutzer mit der Struktur des div-Containers vertraut sind.

Der Beispiel für „pash-m“ könnte lauten:

```
<div class="pashm" width="640" height="480" videoID="306">  
Das ist ein Beispielvideo</div>
```

---

<sup>13</sup> Busby, D. (05.06.2009)

Erstellung des WordPress Flowplayer Plugins „pash-m“

Dieser div-Container wurde geparkt (analysiert) und der Wert des Elementes „videoID“ an die zentrale Funktion „getURLfromID“ übergeben. Die Funktion unterstützt String und Integer-Werte.

### 2.3.2 Implementierung eines Flowplayer Einstellungsmenüs

Hier folgt die Entwicklung eines WordPress Formulars, das sämtliche Einstellungsmöglichkeiten des „SaiWeb Flowplayer for WordPress“ ermöglichen soll.

```
public function build_min_player($settings) {
```

Um die Entwicklung zu erleichtern wurde für sämtliche Einstellungsmöglichkeiten ein assoziatives Array namens \$settings angelegt.

Dieses enthält:

- Auflösung (Formularfeld)

```
$options["variables"]["url"];
```

```
$options["variables"]["height"]; //Höhe des Videos
```

```
$options["variables"]["width"]; //Breite des Videos
```

- Schriftart (Auswahlliste)

In diesem Feld sollen alle möglichen Schriftarten hinterlegt sein.

```
$options["variables"]["font_type"] // array ("Verdana","Arial","Helvetica");
```

In diesem Feld soll die Auswahl für die Oberfläche möglich sein.

```
$options["variables"]["font_type_selection"]; //Integer (1-5)
```

- Schriftgröße (Auswahlliste)

```
$options["variables"]["font_size"]; // Integer (4-30)
```

- Schriftfarbe (Formularfeld)

```
$options["variables"]["font_color"]; // String (#123456)
```

- Standardlautstärke (Formularfeld 10er- Schritte)

```
$options["variables"]["volume"]; // Integer (0-100)
```

- Die Auswahl einer Captions-XML aus der Mediathek kann nun über `$options["variables"]["captionId"]`; erfolgen.

### 2.3.3 Implementierung des Captions Skripts

Über ein eigenes „pash-m“-Menü und Plugin, gelangt man zum Captions Editor. Dieses Menü wurde wie oben beschrieben über Hooks zum WordPress System hinzugefügt. Hier kann der Anwender intuitiv Untertitel zu den aus der Mediathek ausgewählten Medien hinzufügen. Es ist ihm dabei möglich zu **jedem beliebigen Zeitpunkt innerhalb einer Videodatei Text einzublenden**.

Unter Verwendung CSS-Technik würde es dem Anwender ermöglicht unterschiedliche Schriftarten und -formatierungen innerhalb einer Captions-Datei zu verwenden.

## 2.4 Probleme

- Die meisten Probleme traten beim Implementieren der Parsing-Funktion auf. Diese Funktion liest den Inhalt und Variablen von div-Containern aus. Dazu mussten bestimmte Muster definiert werden (Regular Expressions). Die Verwendung von Regular Expressions zur Erkennung von Mustern konnte nur zum Teil verwendet werden, da diese bei der Parameterübergabe zwischen 2 Funktionen unterschiedlich reagierte.
- Zur automatischen erstellen Captions-XML Dateien durch das PHP-Skript mussten auf dem Webserver spezielle Schreibrechte aktiviert werden.
- Zur Anpassung des von Hr. Bestmann zur Verfügung gestellten Skriptes wurde mehr Zeit benötigt als geplant, da es nur teilweise verwendet werden konnte.

Erstellung des WordPress Flowplayer Plugins „pash-m“

## **2.5 Ergebnis**

Folgende Ziele wurden erreicht:

- Der Flowplayer wurde erfolgreich in WordPress eingebunden.
- Mit Hilfe eines div-Containers ist es möglich, Flowplayer in einen Artikel im WordPress System einzubinden.
- Es wurde die Möglichkeit geschaffen, über eine Funktion die in der Mediathek gespeicherten Videos auszuwählen und mit dem Flowplayer abspielen zu lassen.
- Durch ein Auswahlmenü wird der Flowplayer-Div-tag automatisch generiert.
- Caption-Skript zur Bearbeitung von Untertiteln wurde erfolgreich und anwenderfreundlich integriert.

## Literaturverzeichnis

Busby, D. (05. 06 2009). *WordPress Flowplayer | Saiweb*. Abgerufen am 05. 06 2009 von Saiweb: <http://www.saiweb.co.uk/wordpress-flowplayer>

Commons, C. (11. 06 2009). *Creative Commons Attribution-Noncommercial-Share Alike 2.0 UK: England & Wales*. Abgerufen am 11. 06 2009 von Creative Commons: <http://creativecommons.org/licenses/by-nc-sa/2.0/uk/>

Mullenweg, M. (26. 06 2009). *WordPress › WordPress Plugins*. Abgerufen am 26. 06 2009 von WordPress:

<http://wordpress.org/extend/plugins/search.php?q=flowplayer>

Piirainen, A. (23. 06 2009). *Flowplayer - Flash Video Player for the Web*. Abgerufen am 23. 06 2009 von Flowplayer: *Flowplayer - Flash Video Player for the Web*

Piirainen, T. (10. 06 2009). *Flowplayer Ltd*. Abgerufen am 10. 06 2009 von *Flowplayer - Flash Video Player for the Web*: [www.flowplayer.org](http://www.flowplayer.org)

Steckel, M. (Produzent), Steckel, M. (Autor), & Steckel, M. (Regisseur). (2008). *Animation Showreel* [Kinofilm].

[wordpress.org](http://wordpress.org). (25. 06 2009). *WordPress › Blog Tool and Publishing Platform*. Abgerufen am 25. 06 2009 von <http://wordpress.org/>: <http://wordpress.org/>

*Writing a Plugin « WordPress Codex*. (14. 06 2009). Abgerufen am 14. 06 2009 von *Main Page « WordPress Codex*:

[http://codex.wordpress.org/Writing\\_a\\_Plugin](http://codex.wordpress.org/Writing_a_Plugin)