

Technische Fachhochschule Berlin  
Fachbereich VI Informatik und Medien  
Medieninformatik  
Softwareprojekt I und II  
(WS 2005/2006 - SS 2006)

Multimedia-Konvertierungsserver

## **Abschlußbewertung**

Joachim Gärtner

Michael Mrotzek

Benjamin Schwarze

Nils Mitoussis

## Inhaltsverzeichnis

<b>1</b>	<b>Kurzbeschreibung</b>	<b>2</b>
1.1	Team . . . . .	2
1.2	Zuständigkeiten . . . . .	2
<b>2</b>	<b>Realisierung</b>	<b>2</b>
<b>3</b>	<b>Technologien</b>	<b>3</b>
3.1	Webframework: Stripes . . . . .	3
3.2	Datenbank-Mapping: Hibernate . . . . .	3
3.3	Videokonvertierung: FFMpeg . . . . .	4
<b>4</b>	<b>Projekterfahrungen</b>	<b>4</b>
4.1	Joachim Gärtner . . . . .	4
4.2	Michael Mrotzek . . . . .	5
4.3	Nils Mitoussis . . . . .	5
4.4	Benjamin Schwarze . . . . .	5

## 1 Kurzbeschreibung

Die Konvertierung von Videodaten ist im Normalfall ein extrem ressourcen- und zeitraubender Vorgang. Je nach Format, Auflösung und Bildrate kann z.B. die Formatwandlung eines im DivX-Format vorliegenden Videos von 2 Stunden Dauer in das SVCD-Format (MPEG-2) selbst auf einem aktuellen Rechner noch mehrere Stunden in Anspruch nehmen - auf älteren Systemen manchmal sogar Tage. Hinzu kommt auch noch die nicht triviale Welt der Videoformate und Codecs.

Da man einen Rechner im Normalfall besser zum daran arbeiten nutzt, anstatt ihn mit einer alle Ressourcen auslastenden Konvertierung zu blockieren, liegt es nahe, diesen Vorgang auf eine andere, dafür optimierte Maschine auszulagern. Desweiteren ist die Benutzung der für Konvertierung nötigen Software selten einfach und selbsterklärend; oft wird auch mehr als nur ein Programm benötigt, um das gewünschte Ziel zu erreichen.

Ziel dieses Projektes ist es, Benutzern zu ermöglichen, per Web-Interface Videodateien auf einen Server hochzuladen, der anschließend die Konvertierung in das oder die gewünschten Formate durchführt. Die notwendigen Einstellungen werden ebenfalls per Web-Interface vorgenommen. Die einzelnen Konvertierungsaufträge werden in einer Warteschlange gespeichert und verwaltet. Diese wird sequentiell oder nach definierbaren Priorisierungskriterien abgearbeitet. Die konvertierten Dateien werden im Dateisystem gespeichert, und können schließlich vom Benutzer wieder heruntergeladen werden.

### 1.1 Team

- Joachim Gärtner
- Nils Mitoussis
- Michael Mrotzek
- Benjamin Schwarze

### 1.2 Zuständigkeiten

Name	Zuständigkeit
Joachim Gärtner	ffmpeg, Fachlogik
Nils Mitoussis	Hibernate, Fachlogik
Michael Mrotzek	Stripes, JSP
Benjamin Schwarze	Ajax, CSS, Fachlogik

## 2 Realisierung

Zur Umsetzung der Projektidee nutzten wir das klassische Vorgehensmodell für Softwareentwicklung. Nach einer ausgiebigen *Objektorientierten Analyse (OOA)* folgte der Systementwurf

im *Objektorientierten Design (OOD)*.

Die Koordination der Aufgaben im Team erfolgte mittels dem Projektmanagement System Trac<sup>1</sup>. Dieses webbasierte System unterstützt zudem Subversion als Versionsverwaltung, das wir für unser Projekt nutzten. Weiterhin verwendeten wir Eclipse als Entwicklungsumgebung und Ant als Buildsystem.

## 3 Technologien

### 3.1 Webframework: Stripes

Wir haben uns für Stripes<sup>2</sup> als Präsentationsframework entschieden, weil es im Gegensatz zu Struts oder JSF<sup>3</sup> weniger Konfigurationen mittels XML Dateien benötigt. Zwar kannten wir keines der drei Frameworks, aber die Entscheidung fiel auf Stripes. Spätere Erfahrungen mit Struts und JSF, im Rahmen des Wahlpflichtfachs *User Interface Design*, zeigten, dass es eine gute Wahl war.

Stripes beeindruckt durch ein Minimum an Konfiguration innerhalb der `web.xml` Datei. Die benötigten Einstellungen werden einfach über Annotations direkt im Quellcode beschrieben. Außerdem ist der Start mit Stripes einfach, da die sogenannten `ActionBeans` noch gar nicht geschrieben sein müssen, und trotzdem die JSP Seite ohne Ausnahmen dargestellt wird. Das erleichtert die Entwicklung einer Webanwendung erheblich.

Reicht die bereitgestellte Funktionalität nicht aus oder soll etwas anders ablaufen, so ist es sehr einfach das Stripes Framework zu erweitern. Von großem Vorteil ist auch die sehr gut gepflegte Dokumentation, sowie der Support über die Mailingliste.

### 3.2 Datenbank-Mapping: Hibernate

Nach den Vorlesungen über relationale Datenbanken und ersten Einblicken in Objektdatenbanken interessierten wir uns dafür, das objektrelationale Mapping anhand eines gängigen Tools einzusetzen. Hibernate<sup>4</sup> in Version 3.1 hat sich als relativ ausgereifte und umfangreiche Lösung erwiesen. Komplexe Funktionen wie selbständig kaskadierende Operationen im Objektbaum haben wir beispielsweise mit eigenen Ereignissen ergänzen können. Mit Reflection agiert Hibernate weitgehend im Hintergrund, aber ein grundlegendes Verständnis für die internen Abläufe von Hibernate ist dennoch nötig.

Neben der Einarbeitung in die Konzeption und Funktionsweise von Hibernate war die Konfiguration des Mappings eine der Hauptschwierigkeiten. Die offizielle Dokumentation ist knapp gehalten und der Aufbau der Mapping-Beschreibung in XML ist in komplexeren Situationen nicht mehr einfach zu überschauen. Die optionale Möglichkeit, Java Annotations zu

---

<sup>1</sup><http://www.edgewall.com/trac/>

<sup>2</sup><http://stripes.mc4j.org/confluence/display/stripes/Home>

<sup>3</sup>Java Server Faces

<sup>4</sup><http://www.hibernate.org>

verwenden, so dass die Mapping-Informationen direkt in den Klassen stehen können, ist noch nicht über Beta- und Release-Candidate-Status herausgekommen.

Optimistisches Locking mit Versionsnummern wird einfach und sinnvoll unterstützt. Die Abfragesprachen konnten gut objektorientiert eingesetzt werden und sind vergleichbar mit denen von Objektdatenbanken. Diese können allerdings auf das Mapping weitgehend verzichten.

### 3.3 Videokonvertierung: FFMpeg

Die in Java direkt mögliche Videoverarbeitung lässt mehr als zu wünschen übrig. Sowohl die über das Java-Mediaframework verfügbaren nativen Codec-Implementierungen, Containerformate als auch allem voran die Geschwindigkeit der Verarbeitung entsprachen bei weitem nicht unseren Erwartungen. Durch vorhandene Erfahrungen mit Videobearbeitung und Videokonvertierung fiel die Wahl für die zur Konvertierung zu nutzenden Software auf FFMpeg<sup>5</sup>. Bestehend aus Bibliotheken für nahezu alle erdenklichen Codecs und Formate bietet das in C geschriebene Framework die umfassendste Implementierung von sowohl frei verfügbaren als auch teilweise von patentgeschützten Komprimierungs-Algorithmen, die uns bekannt ist. Durch eine sehr große Verbreitung von FFMpeg als de-facto Standard als Basis für andere Video-Software wie z.B. VLC oder MPlayer, ist eine ausreichende Qualität der Codecs und Formate sichergestellt.

Zunächst war geplant, die komplette Anbindung an Java über JNI direkt an die API von FFMpeg vorzunehmen. Allerdings stellte sich nach und nach heraus, dass die FFMpeg-Dokumentation mehr als dürftig ist, und auch der Quelltext sehr schlecht dokumentiert ist. Da hierdurch der JNI-Ansatz wenig Aussicht auf Erfolg hatte, entschieden wir uns für den direkten Aufruf von FFMpeg über `Runtime.exec()` aus Java heraus. Für die Auswertung der Ausgaben des Prozesses werden diese zeilenweise eingelesen und mit regulären Ausdrücken nach Schlüsselsequenzen durchsucht, die dann entsprechend weiterverarbeitet werden.

## 4 Projekterfahrungen

### 4.1 Joachim Gärtner

Das Erstellen des MultiMediaKonvertierungsServers im Rahmen des Softwarewareprojekts über zwei Semester war das erste Mal, daß ein größeres und vor allem auch zusammenhängendes Projekt in einem Team erstellt wurde. Die hierbei gewonnenen Erfahrungen und Einsichten sind von der Tiefe und Umfänglichkeit her bisher absolut unerreicht, und reichen vermutlich an die Rahmenparameter 'echter' Projekte heran. Schwierigkeiten gab es aus meiner Sicht vor allem bei der Koordination der Arbeiten sowie der Einschätzung des aktuellen Fortschritts des Projekts als Ganzes.

---

<sup>5</sup><http://ffmpeg.mplayerhq.hu/>

Abschliessend betrachtet bleibt zu sagen, daß ich sehr viel Neues in verschiedensten Bereichen gelernt habe, sowie Einblicke in Bereiche und Technologien gewonnen habe, welche für mein weiteres Berufsleben sicher sehr nützlich sein werden.

#### **4.2 Michael Mrotzek**

Durch die Bearbeitung eines Projekts über zwei Semester, wird die eigene Arbeitsorganisation auf die Probe gestellt. Bei kleineren Projektarbeiten in vorherigen Semestern war dieser Faktor noch nicht so wichtig für den Projekterfolg. Doch bei einem Projekt, das über knapp ein Jahr realisiert werden soll, ist die Planung und Einhaltung von Aufgaben enorm wichtig. Weiterhin benötigt die Umsetzung eine sehr gute und ausführliche Kommunikation zwischen den Teammitgliedern. Ansonsten verlaufen Arbeitsschritte aneinander vorbei. Dies haben wir auch erst im Verlauf des Projekts gelernt. Abschließend kann ich sagen, trotz großem Arbeitsaufwand, hat die Durchführung Spaß gemacht. Nebenbei habe ich sehr viel über J2EE und Projektorganisation gelernt.

#### **4.3 Nils Mitoussis**

Die Bearbeitung eines Teilaspekts innerhalb eines größeren Projekts ist innerhalb des Studiums selten und daher besonders wichtig. Die intensiven Absprachen und Diskussionen mit den anderen Teammitgliedern waren oft lehrreich und haben zu neuen Ideen und Betrachtungen geführt. Neben der Teamarbeit war es sehr interessant, objektrelationales Mapping und Transaktionen vertieft zu betrachten, auch als Grundlage um sich später einmal mit Objekt-datenbanken zu befassen. Der Arbeitsaufwand für das Projekt war relativ groß und nur mit guter Projektorganisation zu bewältigen. Sich in geeignete Projektmanagement-Instrumente einzuarbeiten war ebenfalls eine sehr nützliche Erfahrung.

#### **4.4 Benjamin Schwarze**

Die Entwicklung und Programmierung des Multimedia-Konvertierungsservers war ein Projekt, durch das ich an Erfahrung gewonnen habe. Die Abstimmung der einzelnen Schichten (Datenbank, Logik, User Interface) untereinander und das Kommunizieren mit den Teammitgliedern schulte die Teamfähigkeit. Probleme mussten verständlich an die anderen vermittelt werden, sowie Vor- und Nachteile größerer Strukturänderungen mit allen abgestimmt werden. Obwohl die meisten Technologien (JSP, Stripes, AJAX) für mich neu waren, machte es mir Spaß mich in diese einzuarbeiten. Dadurch konnte ich mich zum ersten Mal etwas mehr mit webbasierter Softwareentwicklung auseinandersetzen.