

Repräsentation von Kurseinheiten der FernUniversität als Hyperaudio- Dokumente in Moodle: Design und Implementierung

Michael Lämmermann

Matrikelnummer 9611711

E-Mail: michael.laemmermann@studium.fernuni-hagen.de

B.Sc. Wirtschaftsinformatik

Studium in: Vollzeit

Drei Monate

Betreuer: Niels Seidel

1. Problemstellung

1.1. Motivation

Die Motivation zur Ausschreibung dieses Themas besteht darin, dass ca. 80% der Studierenden in Teilzeit studieren und 80% gleichfalls neben dem Studium arbeiten. Unter diesen Umständen beschäftigen sich viele Studierende erst kurz vor der Prüfung mit den Lerninhalten. An der Fakultät Mathematik/Informatik und auch an der Fakultät für Wirtschaftswissenschaften bestehen diese Lerninhalte zu einem guten Teil aus textlastigen Kurseinheiten, die Abbildungen und Formeln enthalten. Die Idee besteht nun darin, den Lernenden erstens eine alternative Repräsentation (Modalität) der Lerninhalte anzubieten und ihnen zweitens das Lernen während ungenutzter Alltagssituationen zu ermöglichen (z.B. lange Autofahrten, Pendeln in Bus und Bahn, beim Joggen, etc.). Auf diese Weise könnten die Lernenden die Inhalte häufiger rezipieren und einüben. Ergänzt um gute E-Assessments (Selbsttests) hätten Sie in Summe eine Chance, sich frühzeitig und kontinuierlich auf die Prüfung vorzubereiten und vielleicht bessere Lernerfolge zu erzielen.

1.2. Aufgabenstellung

Es soll ein Moodle-Plugin entwickelt werden, mit dem es möglich ist, Audiodateien abzuspielen, bei welchen Studierende eines Kurses zeitgenau persönliche und für Kommilitonen sichtbare

Exposé zur Abschlussarbeit am LG Kooperative Systeme der FernUniversität in Hagen
Kommentare hinterlegen können, welche in Echtzeit dargestellt werden. Zusätzlich soll es möglich sein, Inhalte (z.B. Abbildungen, Formeln, Verweisziele von Hyperlinks) aus den Kurseinheiten an den Audiodokumenten zeitlich zu verankern, damit diese beim Anhören zum entsprechenden Zeitpunkt dargestellt werden. Zusätzlich sollen auch akustische Signale auf das Vorhandensein visueller Inhalte hinweisen. Das Moodle-Plugin übernimmt somit auch Aufgaben eines Group Awareness-Tools.

1.3. Intendierte Ergebnisse

Konzeption und Entwicklung eines lauffähigen Prototyps des Moodle-Plugins. Neben einem ansprechend gestalteten und responsiven Design soll das Plugin einfach zu bedienen sein.

2. Aktueller Stand der Technik

Moodle besitzt standardmäßig das Plugin „VideoJS Player“ [1] (GitHub-Projekt, Apache-Lizenz 2.0) [2]. Hierbei handelt es sich um eine JavaScript Umgebung für die Wiedergabe von Audio- und Video-Dateien, welche durch den nativen Audio-/Videoplayer des Browsers wiedergegeben werden. Zusätzlich besitzt es einen optionalen Fallback auf den FlashPlayer. Außerdem gibt es das „HTML5 Audio“-Plugin, welches ebenfalls den nativen Audioplayer des Webbrowsers nutzt.

Abseits der Moodle Standard-Plugins existiert das „H5P“-Plugin [3] basierend auf der gleichnamigen Software zum Erstellen von interaktiven Lerninhalten (GitHub-Projekt, MIT Lizenz) [4]. Dabei handelt es sich um eine Sammlung von interaktiven Komponenten, darunter Course Presentation, Timeline, Interactive Video und Audio. Course Presentation bietet die Möglichkeit interaktive Präsentationen zu gestalten. Timeline kann genutzt werden um Inhalte anhand eines Zeitstrahls darzustellen. Interactive Video ermöglicht, ähnlich wie Course Presentation, die Interaktion während des Abspielens eines Videos. Audio stellt analog zu „HTML5 Audio“ einen Audioplayer dar.

Des Weiteren stellen auch Moodle-fremde Frameworks einen Lösungsansatz dar. Hier ist besonders das Framework wavesurfer.js [5] (GitHub-Projekt, BSD-3-Clause) [6] hervorzuheben. Es existiert bereits ein Beispiel für ein Annotation-Tool [7]. Dieses ermöglicht es zu einem definierbaren Zeitfenster innerhalb eines Audiodokuments Textinformationen zu erfassen und darstellen zu lassen. Zu diesem Zweck wurden basierend auf diesem Framework noch einige weitere Projekte entwickelt, wie zum Beispiel BAT (GitHub-Projekt, GNU General Public License 3) [8] und audio-annotator (GitHub-Projekt, BSD-2-Clause) [9].

Das Framework timesheets.js (ehemaliges GitHub-Projekt, MIT Lizenz) [10] bietet ebenfalls die Möglichkeit Informationen in definierten Zeitfenstern eines Audiodokuments darstellen zu lassen.

Die Mozilla Corporation bietet mit Popcorn.js (GitHub-Projekt, MIT Lizenz) eine Bibliothek an, welche eine standardisierte Steuerung von Medieninhalten aus verschiedenen Quellen ermöglicht. Zusätzlich wird ein Plugin-System unterstützt, womit das Video zeitgenau um externe Inhalte (z.B. Quellenangaben oder beliebigen JavaScript-Code) zu erweitern. Die Wartung für die Bibliothek wurde seitens Mozilla zwar mittlerweile eingestellt, das Projekt steht aber weiterhin auf GitHub zur Verfügung.

Einzelnen betrachtet bietet keine der angesprochenen Ansätze eine Lösung der Problemstellung. Jene Komponenten, welche Audiodateien abspielen können, bieten keinerlei Möglichkeit zur Interaktion und Verknüpfung mit weiteren Inhalten. Andersherum fehlt bei Komponenten, die über entsprechende Interaktionsmöglichkeiten verfügen, die Möglichkeit zum Abspielen und Verknüpfen von Audioinhalten.

3. Lösungsidee

Als Ausgangspunkt dienen die Bedürfnisse bzw. Anforderungen der Studierenden und Lehrenden (siehe Tabelle 1 bis 4). Diese werden definiert und die Zusammenhänge der verschiedenen Inhalte analysiert. Anhand dieser Informationen werden Nutzungsszenarien aufgestellt und ein Konzept für die Wiedergabe von Hyperaudio-Dokumenten entworfen.

Tabelle 1 – Anforderungen an die Schnittstelle

Anforderung	Priorität	Betrifft
1. Definition einer Schnittstellen-Datei		
1.1. Definition der Zuordnung zu einem Kurs, deren Kurseinheit und den Kapiteln	hoch	Lehrende
1.2. Definition der abzuspielenden Audiodatei	hoch	Lehrende
1.3. Definition der zeitabhängigen Annotation von Bildern, Tabellen, Formeln etc.	hoch	Lehrende
1.4. Definition von Metadaten (Autor, Quellen etc.)	mittel	Lehrende
2. Entwickeln eines Imports für die Schnittstellen-Datei	hoch	Lehrende

Tabelle 2 - Anforderungen an die Integration in die Moodle-Oberfläche

Anforderung	Priorität	Betrifft
1. Einbindung in die Moodle Startseite analog zu „Helpdesk Wiki“	mittel	Lehrende und Studierende
2. Einbindung der Hyperaudio-Dokumente in die Kursseiten	hoch	Lehrende und Studierende
3. Einbindung der Administration in die Kursseite	hoch	Lehrende
4. Einbindung als eigenständige Übersicht in der alle Hyperaudio-Angebote gesammelt werden analog „Lernumgebungen“ gegliedert nach Kurs, Kurseinheiten und Kapitel/Inhaltsverzeichnis	mittel	Lehrende und Studierende
5. Hyperaudiodokumentenseite		
5.1. Einbindung des Players	hoch	Lehrende und Studierende
5.2. Einbindung der Kommentarsektion	hoch	Lehrende und Studierende
5.3. Einbindung der Favoritenfunktion	niedrig	Lehrende und Studierende
5.4. Einbindung einer Galerie der zeitabhängig annotierten Bilder, Tabellen, Formeln etc.	mittel	Lehrende und Studierende
6. Übersicht über zuletzt abgespielte Hyperaudiodokumente	niedrig	Studierende
7. Favoritenfunktion analog zu „Meine Lernumgebungen“		

Exposé zur Abschlussarbeit am LG Kooperative Systeme der FernUniversität in Hagen

7.1. Möglichkeit Hyperaudiodokumente als Favorit zu speichern	niedrig	Studierende
7.2. Eine Übersicht über alle als Favorit gespeicherten Hyperaudiodokumente	niedrig	Studierende

Tabelle 3 - Anforderungen an den Player für Hyperaudio-Dokumente

Anforderung	Priorität	Betrifft
1. Wiedergabe der Audiodatei	hoch	Lehrende und Studierende
2. Darstellung der zeitabhängig annotierten Bilder, Tabellen, Formeln etc.	hoch	Lehrende und Studierende
3. Wiedergabe der Audio Cues	mittel	Lehrende und Studierende
4. Einbettung der Kommentare und persönlichen Notizen/Markierungen		
4.1. Visualisierung der zeitabhängig annotierten Kommentare und persönlichen Notizen/Markierungen	hoch	Lehrende und Studierende
4.2. Entwicklung von Interaktionsmöglichkeiten		
4.2.1. Weiterleitung zu vorhandenem Kommentaren und persönlichen Notizen/Markierungen in der Kommentarsektion	niedrig	Lehrende und Studierende
4.2.2. Erstellung zeitabhängiger annotierter Kommentare und persönlicher Notizen/Markierungen	hoch	Lehrende und Studierende
5. Funktion zum Fortsetzen von unterbrochenen Wiedergaben nach erneuter Anmeldung im Moodle	niedrig	Lehrende und Studierende

Tabelle 4 - Anforderungen an die Kommentarsektion

Anforderung	Priorität	Betrifft
1. Chronologische Darstellung der Kommentare	hoch	Lehrende und Studierende
2. Möglichkeit direkt auf Kommentare zu antworten	mittel	Lehrende und Studierende
3. Möglichkeit persönliche Notizen/Markierungen zu bearbeiten/löschen	mittel	Studierende
4. Suchfunktion in den Kommentaren und persönlichen Notizen	niedrig	Lehrende und Studierende

Das Konzept bildet die Basis für die Implementierung. Wie im vorherigen Kapitel beschrieben gibt es bereits Plugins, welche die jeweiligen gewünschten Funktionen im Einzelnen abdecken. Vor der Implementierung soll erörtert werden, ob es sinnvoll ist, eines der Plugins um die Funktionalitäten der anderen Plugins zu erweitern, oder ein neues Plugin basierend auf den Features der anderen Plugins und Frameworks zu entwickeln. Nach dieser Entscheidung soll das Plugin entsprechend in einem iterativen Vorgehen implementiert werden. Nach jeder Iteration soll die Erfüllung der Anforderungen anhand der Nutzungsszenarien durch manuelle Tests am Prototypen überprüft werden. Testvorgehen und -resultate werden dokumentiert. Anhand des lauffähigen Prototyps kann

Exposé zur Abschlussarbeit am LG Kooperative Systeme der FernUniversität in Hagen
dann in einer anschließenden Arbeit evaluiert werden, inwiefern die gewünschte Problemlösung erarbeitet wurde oder wie das Plugin verbessert werden kann.
Am Ende der Arbeit wird ein Ausblick mit eventuell offenen Fragen gegeben. Zum Beispiel kann die Fragestellung thematisiert werden, wie Hyperaudio-Dokumente mitsamt zugehöriger Kommentare weitergegeben werden können.

Vorläufige Gliederung

1. Einführung
 - 1.1. Motivation
 - 1.2. Problemstellung
 - 1.3. Zielsetzung
2. Analyse
 - 2.1. Bedürfnisse der Studierenden und Lehrenden
 - 2.2. Lernen mit Hyperaudio-Dokumenten
 - 2.3. Zusammenhänge der medialen Komponenten von Hyperaudio-Dokument und Annotationen
 - 2.4. Evaluation bestehender Komponenten
3. Konzeption zur Repräsentation, Erstellung und Pflege von Kurseinheiten als Hyperaudio-Dokumente
 - 3.1. Nutzungsszenarien für das Moodle-Plugin
 - 3.2. Architektur des Moodle-Plugins
 - 3.3. Gestaltung der Benutzeroberfläche
 - 3.4. Definition des Schnittstellenformats für Hyperaudio-Dokumente
4. Implementierung
 - 4.1. Moodle-Integration
 - 4.2. Player
 - 4.3. Kommentarsektion
 - 4.4. Evaluation
5. Zusammenfassung und Ausblick

Vorläufiger Zeitplan

Geplant ist, dass der Bearbeitungszeitraum ca. 13 Wochen umfasst (gewünschte Anmeldung am 21.05.2018):

- Einführung (Woche 1)
- Analyse (Woche 2 bis Woche 3)
- Konzeption zur Repräsentation von Kurseinheiten als Hyperaudio-Dokumente (Woche 4 bis Woche 6)
- Implementierung (Woche 7 bis Woche 10)
- Zusammenfassung und Ausblick (Woche 11)
- Korrekturen (Woche 12 bis Woche 13)

Exposé zur Abschlussarbeit am LG Kooperative Systeme der FernUniversität in Hagen
Ausgangsliteratur

- [1] „Video.js: The Player Framework,“ [Online]. Verfügbar: <http://videojs.com/>. [Zugriff am 4 Februar 2018].
- [2] „Video.js · GitHub,“ [Online]. Verfügbar: <https://github.com/videojs>. [Zugriff am 4 Februar 2018].
- [3] „H5P – Create and Share Rich HTML5 Content and Applications,“ [Online]. Verfügbar: <https://h5p.org/>. [Zugriff am 4 Februar 2018].
- [4] „H5P,“ [Online]. Verfügbar: <https://github.com/h5p>. [Zugriff am 4 Februar 2018].
- [5] „wavesurfer.js,“ [Online]. Verfügbar: <https://wavesurfer-js.org/>. [Zugriff am 4 Februar 2018].
- [6] „GitHub - katspaugh/wavesurfer.js: Navigable waveform built on Web Audio and Canvas,“ [Online]. Verfügbar: <https://github.com/katspaugh/wavesurfer.js>. [Zugriff am 4 Februar 2018].
- [7] „wavesurfer.js | Annotation tool,“ [Online]. Verfügbar: <http://wavesurfer-js.org/example/annotation/>. [Zugriff am 4 Februar 2018].
- [8] „GitHub - BlaiMelendezCatalan/BAT,“ [Online]. Verfügbar: <https://github.com/BlaiMelendezCatalan/BAT/>. [Zugriff am 4 Februar 2018].
- [9] „GitHub - CrowdCurio/audio-annotator: A JavaScript interface for annotating and labeling audio files,“ [Online]. Verfügbar: <https://github.com/CrowdCurio/audio-annotator>. [Zugriff am 4 Februar 2018].
- [10] „timesheets.js,“ [Online]. Verfügbar: <http://wam.inrialpes.fr/timesheets/>. [Zugriff am 4 Februar 2018].
- [11] J. Zumbach und C. Kroeber, „Learning with Hyperaudio: Cognitive Load and Knowledge Acquisition in Non-Linear Auditory Instruction,“ in *Avoiding simplicity confronting complexity. Advances in studying and designing (computer-based) powerful learning environments*, 2006, pp. 359-170.
- [12] J. Zumbach und N. Schwartz, „Hyperaudio learning for non-linear auditory knowledge acquisition,“ in *Computers in Human Behavior* 41, 2014, pp. 365-373.
- [13] D. Petrelli und E. Not, „User-centred design of flexible hypermedia for a mobile guide: Reflections on the HyperAudio experience,“ in *User Modeling and User-Adapted Interaction* 15.3-4, 2005, pp. 303-338.
- [14] A. E. Barron, „Auditory instruction,“ in *Handbook of research on educational communications and technology* 2, 2004, pp. 949-978.
- [15] „MoodleDocs,“ [Online]. Verfügbar: https://docs.moodle.org/dev/Main_Page. [Zugriff am 4 Februar 2018].
- [16] W3C, „SMIL 3.0 Timing and Synchronization,“ [Online]. Verfügbar: <https://www.w3.org/TR/SMIL3/smil-timing.html>. [Zugriff am 4 Februar 2018].
- [17] W3C, „SMIL Timesheets 1.0,“ [Online]. Verfügbar: <https://www.w3.org/TR/timesheets/>. [Zugriff am 4 Februar 2018].