

# Arbeitsplan – EMM16

Softwaretechnikpraktikum 2016  
SS 2016  
Gruppe EMM16

Betreuer: Prof. Gräbe  
Tutor: Christoph Jobst  
Project Owner: Kamilla Skuldeny

Datum: 21.02.2016

## Inhaltsverzeichnis

1. Projektvision .....	2
2. Voraussetzungen .....	2
3. Designübersicht und Funktionalität .....	3
Nutzerszenarien .....	3
User .....	3
Webadministrator .....	4
Server .....	5
4. Arbeitspakete .....	6
5. Vorprojekt .....	6
6. Glossar .....	7

## 1. Projektvision

An sächsischen Hochschulen sind verschiedene E-Learning-Portale im Einsatz. Eine Vernetzung dieser Portale auf der Ebene von E-Learning-Ressourcen ist schwierig. Ein sinnvoller Austausch ist nur auf der Ebene von Kursexporten möglich. Hierfür gibt es mit Learning Objects Metadata (LOM) einen offenen Standard zum Austausch von Metadaten (IEEE). Es soll eine Applikation entstehen, welche E-Learning-Kurse sächsischer Hochschulen darstellt. Zielgruppe für die Plattform sind Studieninteressierte. Die Applikation soll eine Nutzeroberfläche bereitstellen, die in vorhandene Websites eingebunden wird.

Im Vorfeld eines solchen Austauschs ist es interessant, einen Überblick über die in den verschiedenen Portalen angebotenen Kurse in standardisiertem Format zu erhalten. Ein solcher Export von Metainformationen kann auf datenschutz- und urheberrechtlich unbedenkliche Weise organisiert werden, da weder personenbezogene Daten exportiert werden müssen noch der Zugang zu den Metadaten die Zugänglichkeit der Ressource selbst impliziert. Kursmetadaten können also prinzipiell als Linked Open Data zur Verfügung gestellt werden. Die Integrationslösung soll RDF-basierte semantische Technologien auf der Ebene des Datenmodells und zur Datenintegration verwenden. Zur Suche dient ein SPARQL-Endpunkt auf dem entsprechenden RDF-Store. Besondere Rolle spielt dabei die Volltextsuche über die Tripel.

Die angesprochene Oberfläche soll ansehnlich darstellen, wo welche Kurse angeboten werden. Der Nutzer soll die Möglichkeit auf den semantischen Daten zu suchen. Ergebnisse sollen übersichtlich und strukturiert nach Hochschule und/oder Studiengang dargestellt werden. Der Nutzer soll die Möglichkeit haben zu einem Kurs Detailinformationen zu.

## 2. Voraussetzungen

Damit die entstehende Applikation den Anforderungen der Produktvision entsprechen kann, werden folgende Voraussetzungen definiert:

- **Systemanforderungen an Webserver:**

Auf dem Webserver läuft eine normale Homepage, in die das Frontend eingebunden wird. Damit die Applikation im Hintergrund Aufgaben bewältigen kann müssen ein RDF-Store mit SPARQL-Endpunkt (min. 1.1 zur Datenmanipulation) PHP Server, Java Server und ein Apache Webserver im Hintergrund in aktuellen Versionen laufen. Weiterhin wird Speicherplatz zum Cachen der Daten und Rechenleistung zum abrufen und verarbeiten der Daten benötigt.

- **Nutzeranfragen**

Der Anfragende formuliert seine Anfragen mittels Wörtern. Möglich sind auch mehrere Wörter, die wie in aktuellen Suchmaschinen üblich UND-verknüpft werden. Hat der Nutzer einen Kurs gewählt können ihm auf Basis dessen ähnliche Kurse vorgeschlagen werden.

- **Speicherung und Suche auf den Daten**

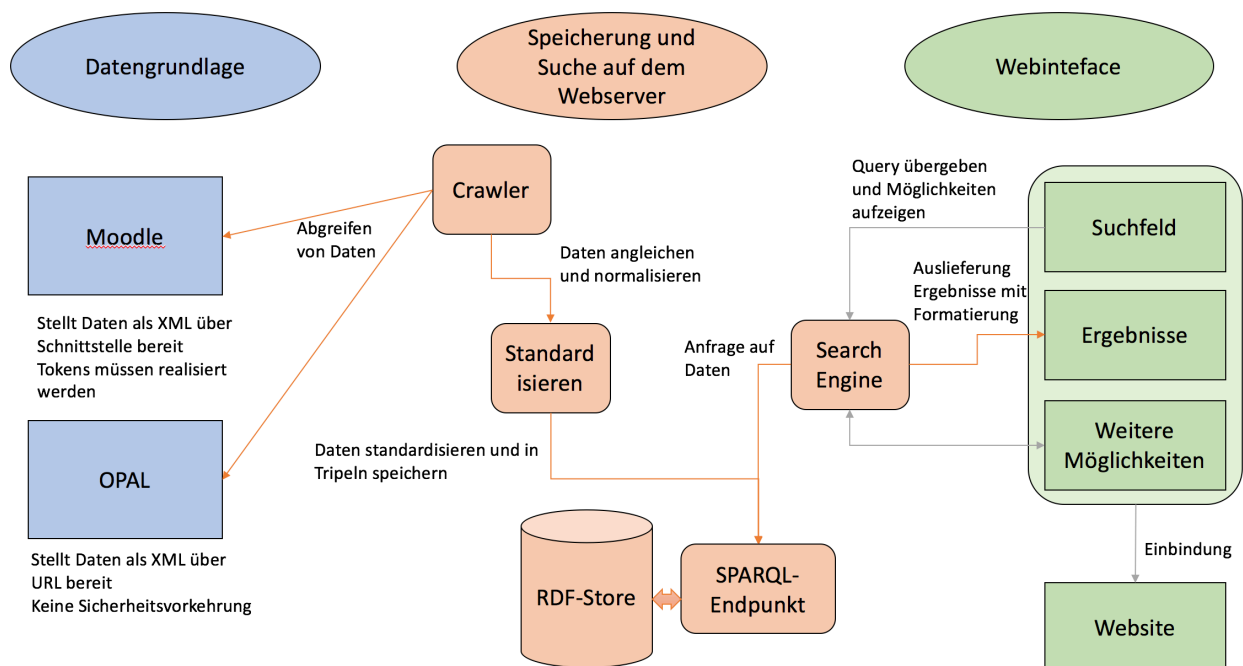
Die Daten werden aus den beiden Portalen gewonnen. Bei der Speicherung müssen die vorliegenden Daten soweit als möglich einem Standard angepasst werden, sodass diese weitläufig verwendbar werden. Entsprechenden Standards des Dublin Core und der LOM sind zu folgen.

### 3. Designübersicht und Funktionalität

Die einzige für den Nutzer sichtbare Oberfläche ist das eingebundene Webinterface. In diesem kann er eine Suche in einem Suchfeld tätigen. Dort wird ihm auch erklärt, welche Möglichkeiten er zur Suche hat.

Ein Crawler sammelt die Daten der beiden Portale ein und übergibt sie in den Standardisierungsprozess. In diesem werden die Daten in Tripel umgewandelt, um dann in einem RDF-Store gespeichert zu werden. Dieser RDF-Store hat einen zugehörigen SPARQL-Endpunkt, welcher sowohl eine Navigation in den Tripeln ermöglicht als auch eine Volltextsuche auf allen Knoten.

Als RDF-Framework bietet sich Apache Jena an, da sowohl volle SPARQL Unterstützung für die Modelle in RDF besteht als auch eine einfache Implementierung einer Volltextsuche möglich ist.



#### Nutzerszenarien

##### User

/Use Case 10: Verschiedene Endgeräte und Usability

**Beschreibung:** Zugriff auf das Interface von verschiedenen Endgeräten.

**Funktionale Anforderungen:** Das Interface soll selbstständig erkennen, wo es angezeigt wird und einerseits demnach skalieren, als auch die Nutzersteuerung optimieren. Egal durch welches Endgerät betrachtet wird, der Nutzer soll alle Funktionen nutzen können.

**Ziel:** Der Nutzer hat keine Probleme bei der Nutzung des Interfaces.

**Beteiligte Rollen:** Nutzer

/Use Case 20: Geschäftsprozess - Aufruf der Website mit dem Webinterface

**Beschreibung:** Der User greift erstmalig auf die Website mit dem Webinterface zu.

**Funktionale Anforderungen:** Schnelle Darstellung des Suchfeldes und aller sonstiger Informationen eingebunden in die Website. Der User soll intuitiv erkennen können, was er mit dem Webinterface machen kann. Die Suche selbst soll intuitiv nutzbar sein, zusätzliche Möglichkeiten können angezeigt werden (z.B. Beispielqueries).

**Ziel:** Der Nutzer fühlt sich von der Suche angesprochen und nutzt sie.

**Beteiligte Rollen:** Nutzer

/Use Case 30: Geschäftsprozess - Suche des Nutzers

**Beschreibung:** Der User sucht über das Suchfeld nach Informationen.

**Funktionale Anforderungen:** Durch klicken des Suchbuttons oder der ENTER-Taste führt der Nutzer die Suche aus, sodass die Daten an den Search-Server übergeben werden. Der Query wird dabei in einen SPARQL-Query umgeformt und verarbeitet. Dies erfolgt dynamisch ohne Neu laden der Seite. Der Nutzer soll merken, dass seine Anfrage bearbeitet wird.

**Ziel:** Der Nutzer übermittelt seine Anfrage.

**Beteiligte Rollen:** Nutzer

/Use Case 40: Darstellung der Ergebnisse

**Beschreibung:** Der User hat gesucht und bekommt seine Ergebnisse angezeigt.

**Funktionale Anforderungen:** Die Inhalte werden unter dem Suchfeld dargestellt. Es ist zu erkennen, um welchen Kurs von welcher Hochschule es sich handelt. Die Ergebnisse werden nach ihrer Relevanz sortiert. Eine einfache Navigation ist zu gewährleisten. Für den Nutzer muss es ersichtlich sein, wie er zu mehr Informationen zu einem Kurs gelangt.

**Ziel:** Der Nutzer betrachtet die Ergebnisse seiner Suche.

**Beteiligte Rollen:** Nutzer

/Use Case 50: Geschäftsprozess - Detailansicht eines Kurses

**Beschreibung:** Der User fordert mehr Informationen zu einem Kurs an.

**Funktionale Anforderungen:** Im selben Browserfenster werden ohne Neu laden alle verfügbaren und öffentlichen Informationen zum Kurs dargestellt. Die Informationen werden durch Navigation im RDF-Store gewonnen. Weiterhin kann der Nutzer, wenn ähnliche Daten bestehen, durch SPARQL durch Kurse navigieren.

**Ziel:** Der Nutzer kann sich zu einem Kurs näher informieren.

**Beteiligte Rollen:** Nutzer

Webadministrator

/Use Case 60: Geschäftsprozess - Erstinstallation der Applikation

**Beschreibung:** Der Administrator bezweckt die Applikation auf seiner Website zu nutzen.

**Funktionale Anforderungen:** Es soll eine ausführliche Dokumentation bereitstehen, wie die Applikation installiert wird und was dabei zu beachten ist. Nach dem hochladen der Daten wird der Admin durch ein Setup geführt, der die Systemanforderungen testet und notwendige Eingaben abfragt. Dabei sollen viele Beispiele und Hilfestellungen vorhanden sein. Nach der Einrichtung soll der Setup nicht mehr zugänglich sein. Nur durch bestimmte Techniken, die Dokumentiert sind lassen sich dann noch Daten ändern.

**Ziel:** Der Admin richtet die Applikation erstmalig ein.

**Beteiligte Rollen:** Admin

/Use Case 70: Geschäftsprozess - Einbindung in Website

**Beschreibung:** Der Administrator will das Webinterface in seine Website einbinden.

**Funktionale Anforderungen:** Für den Admin soll ausführlich dokumentiert sein, wie er das Interface in die Seite einbindet. Ihm sollten dabei mehrere aktuelle Techniken zur Verfügung stehen, sodass eine Kompatibilität zu diversen Webseiten erreicht wird. In gewissem Maße sollen ihm Designentscheidungen überlassen werden z.B. Farbe, um eine optische Einheit mit der Website zu erreichen.

**Ziel:** Integration in Webauftritt

**Beteiligte Rollen:** Admin

/Use Case 80: Normalbetrieb

**Beschreibung:** Die Applikation läuft auf der Website.

**Funktionale Anforderungen:** Der Admin soll bei Fehlern informiert werden. Sollten keine Daten abgeholt werden können oder sonstige Fehler auftreten, so ist eine Fehlermeldung per Mail zu senden. Die Dokumentation stellt Hilfen zu bekannten Problemen bereit. Treten keine Fehler auf, so erzeugt die Applikation keine Nachrichten.

**Ziel:** Fehlermeldung und Erkennung

**Beteiligte Rollen:** Admin

Server

/Use Case 90: Abgreifen der Daten

**Beschreibung:** Der Crawler sammelt die bereitgestellten Daten ein.

**Funktionale Anforderungen:** Der Crawler sammelt selbstständig einmal täglich nachts die Daten ein und übergibt diese zur Standardisierung. Dafür ist ein Cronjob oder Ähnliches notwendig.

**Ziel:** Daten werden aktualisiert

/Use Case 100: Standardisierung der Daten

**Beschreibung:** Die Daten werden gelesen, verarbeitet und gespeichert.

**Funktionale Anforderungen:** Das Modul zu Standardisierung bekommt die Daten der Portale übergeben, dabei muss erkannt werden, welches Portal die Daten bereitstellt. Diese werden dann in eine eindeutige und einheitliche Form gebracht, sodass die Datenstruktur für jeden Kurs gleich ist. Anschließend werden alle standardisierten Daten dem RDF Store über den SPARQL-Endpunkt als Tripel übergeben.

**Ziel:** Verarbeitung und Speicherung der Daten

/Use Case 110: Suche auf den Daten

**Beschreibung:** Die Suche bekommt einen Query übergeben.

**Funktionale Anforderungen:** Zunächst wird der Query einem QueryParser übergeben, der daraus einen SPARQL-Query baut. Dieser neue Query wird nun dem SPARQL-Endpunkt zur

Suche auf allen Kursknoten übergeben. Zurückgegeben wird eine Liste mit den passenden Kursknoten aus denen sich dann die nötigen Informationen beziehen lassen.

**Ziel:** Beantwortung der Anfragen durch die Suche auf dem RDF-Store

/Use Case 120: Beantworten der erweiterten Anfrage

**Beschreibung:** Der User fordert weitere Informationen zu einem Kurs an.

**Funktionale Anforderungen:** Der User hat einen Kurs angeklickt, sodass er Detailinformationen bekommen will. Mittels des Zeigers auf den aktuellen Kursknoten werden alle zugehörigen Informationen bezogen. Durch Navigation in den RDF-Daten können ähnliche Kurse vorgeschlagen werden.

**Ziel:** Übergabe Details

## 4. Arbeitspakete

Jedes Arbeitspaket geht mit seiner Testung und Optimierung einher.

Muss-Ziele:

- 35% Vorprojekt – Installation einer Moodleinstanz. Auswertung welche Daten aus XML Dokumenten der Plattform extrahiert werden können. Diese Daten werden anschließend in ein Standardisiertes Modell überführt, sodass eine eindeutige Datentransformation und Datenzusammenführung (Standardisierung) festgelegt wird.
- 5% Erstellung Crawler – Der Crawler wird entwickelt und mit Funktionen für Moodle und OPAL ausgestattet. Die vom Crawler übergebenen Dokumente der Standardisierung übergeben.
- 20% Suche auf dem RDF-Store (SPARQL) – Das RDF-Framework Apache Jena wird so eingerichtet, dass eine Volltextsuche auf den Kursdaten möglich wird. Dafür werden Zusatzmodule installiert. Weiterhin wird ein QueryParser gebaut, der die normalen Suchquers in SPARQL-Quers wandelt und diese dem Endpunkt übergibt.
- 35% Entwicklung Webinterface – Die Oberfläche für den Nutzer wird entworfen und umgesetzt. Dabei spielen sowohl Einbindung, Usability sowie Responsives Design eine Rolle. Wichtig ist die dynamische Aktualisierung der Inhalte, durch den Webserver. Es müssen das Suchfeld, die Ergebnisanzeige und die Detailanzeige umgesetzt werden. Es ist darauf zu achten, dass das Design ansprechend und intuitiv ist.
- 5% Auslieferung des Endprodukts – Eine Ausführliche Dokumentation zur Installation wird erstellt. Alle Komponenten werden verknüpft und auf ihre Integration getestet.

Kann-Ziele:

- 5% *Zusatz* Keyword Wolke – Dem Nutzer werden bei Aufruf der Suche oft genutzte Suchbegriffe in Form einer Tag-Wolke angezeigt.
- 5% *Zusatz* Suchbewertung – Der Nutzer hat die Möglichkeit die Suchergebnisse zu bewerten.
- 20% *Zusatz* Systematisierung der Suche – Die Suche kann einerseits über Suchbegriffe genutzt werden, andererseits hat der Nutzer die Möglichkeit über vorgegebene Facetten zu navigieren.

## 5. Vorprojekt

Am Ende des Vorprojektes soll der fertige RDF-Store stehen. Zunächst müssen alle Daten zusammengetragen und ausgewertet werden. Aus der Sichtung verschiedener Datensätze und wenn vorhanden den zugehörigen Dokumentationen werden Modelle erstellt, die die schlussendliche Datenstruktur festlegen. Diese Modellierung bildet Grundlage für alle weiteren Schritte und ist beispielsweise in UML festzuhalten.

Im nächsten Schritt werden Methoden entwickelt die die gegebenen XML Dateien in die Modelle transformieren können. Logisch ist hier die Modellierung der Objekte der Modellierung als Klassen. Diese Klassen werden dann durch die Standardisierung mit den Daten aus den beiden Portalen gleichsam befüllt, sodass beide Datenquellen gleichsam verwendet werden können. Hierbei ist auf alle möglichen Ausprägungen der XML Rücksicht zu nehmen, genauso wie die Wahl der richtigen Datentypen und deren Größe.

Schlussendliche muss eine Möglichkeit gefunden werden die Objekte der zugehörigen Klassen in einer passenden RDF-Struktur zu speichern. Dabei muss sowohl die Speicherung vom Modell in den RDF-Store also auch vom RDF-Store in ein Objekt möglich sein. Besonders wichtig ist hierbei die richtige Wahl der Datentypen und das Fehlermanagements bei der Überführung.

Es entsteht also eine standardisierte Datengrundlage auf der die weitere Arbeit beruht. Dies ist notwendig, da sonst keine weiteren Modellierungsentscheidungen getroffen werden können. Es muss feststehen, welche Daten wie vorliegen bevor z.B. die Oberfläche geplant wird.

## 6. Glossar

- **Nutzer/Zielgruppe**

Die Applikation wird von Studienberatungsseiten in ihre Websites implementiert. Darauf greifen Studieninteressierte zu, um sich über Module und Kurse an Hochschulen zu informieren.

- **Kurs und Metadaten**

Kurse sind zusätzliche Angebote die Online zusätzliche Informationen bereitstellen. In unserem Fall sind Kurse zu Modulen begleitende Angebote zu Modulen, die eine Hochschule anbietet. Meta-Daten sind, die zu einem Objekt (Kurs) existieren und diesen beschreiben Attribute. Oft folgen sie einem bestimmten Muster, sodass Daten geordnet und vergleichbar sind.

- **Crawler**

Sind Programme, die wiederkehrend Daten von Websites einsammeln und die Daten an den zugehörigen Speicher übergeben. Im vorliegenden Fall sammelt der Crawler die öffentlichen Kursdaten der E-Learning Portale ein.

- **Suchmaschine**

Eine Suchmaschine ist hergestellt, um die in einem Computer oder einem Computernetzwerk wie z. B. dem World Wide Web gespeicherte Dokumenten zu recherchieren. Das bedeutet, dass die Suchmaschine einen Schlüsselwort-Index für die Dokumentbasis erstellen, um Suchanfragen über Schlüsselwörter mit einer nach Relevanz geordneten Trefferliste zu beantworten.

- **Moodle**

Moodle ist ein freies objektorientiertes Kursmanagementsystem und eine Lernplattform. Die Software bietet die Möglichkeiten zur Unterstützung kooperativer Lehr- und Lernmethoden.

- **OPAL**

OPAL ist eine ganzheitliche und hochschulübergreifende IT-Struktur für E-Learning.

- **RDF-Store**

Ist ein Tripelstore (Datenbank) zum speichern und retrieval von Tripeln. Ein Tripel ist dabei eine Konkatenation von Subjekt, Prädikat und Objekt. Tripel können hierbei durch Ressource Description Frameworks importiert und oder exportiert werden.

- **SPARQL-Endpunkt**

Ist eine Schnittstelle zu einem RDF-Store. Mit der zugrundeliegenden Anfragesprache kann in den Tripeln navigiert werden und Daten ausgelesen. Seit der Version 1.1 ist auch die Datenmanipulation über SPARQL möglich.