

Karlsruher Institut für Technologie  
Fakultät für Wirtschaftswissenschaften

## Seminararbeit

am Institut für Angewandte Informatik und Formale  
Beschreibungsverfahren

### Zusammenfassung

Text der Zusammenfassung

**Thema:** Linked Open Data basierte Web 3.0 Anwendungen

**eingereicht von:** Xinji Du, <jacobdu@hotmail.com>  
Christoph Gielisch, <christoph.gielisch@web.de>  
Andreas Gutzan, <andigutzan@gmx.de>  
Clemens Stolle, <clemens.stolle@gmail.com>

**eingereicht am:** 09. Juli 2011

**Betreuer:** Herr Prof. Dr. Rudi Studer  
Herr Dipl.-Wirt.-Ing. Daniel Herzig  
Herr Dipl.-Inf. Benedikt Kämpgen  
Herr Dipl.-Inf. Günter Ladwig

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemdefinition . . . . .	1
1.2	Herangehensweise und Ziele . . . . .	1
<b>2</b>	<b>Beschreibung der Idee</b>	<b>2</b>
2.1	Anwendungsszenario . . . . .	2
2.2	Spielablauf . . . . .	2
2.3	Alleinstellungsmerkmal und Abgrenzung . . . . .	3
2.4	Semantic Gaming . . . . .	3
<b>3</b>	<b>Projektplanung</b>	<b>4</b>
3.1	Arbeitspakete und Zuständigkeiten . . . . .	4
3.2	Evaluierung . . . . .	4
<b>4</b>	<b>Umsetzung, Softwarepakete und Programm</b>	<b>6</b>
4.1	Schematischer Aufbau, Konzeptskizze . . . . .	6
4.2	Verwendete LOD-Datensätze . . . . .	6
4.3	Verwendete Technologien und Frameworks . . . . .	7
4.4	Dokumentation der Methoden und Funktionen . . . . .	7
4.5	Stabilität und Verfügbarkeit . . . . .	8
4.6	Generierung strukturierter Datensätze . . . . .	8
4.7	Installation und Betrieb . . . . .	9
<b>5</b>	<b>Lessons learned</b>	<b>10</b>
5.1	Reflektion der Projektplanung . . . . .	10
5.2	Erkenntnisse . . . . .	10
5.3	Vor- und Nachteile von Linked Open Data . . . . .	10
<b>6</b>	<b>Fazit</b>	<b>11</b>
6.1	Zusammenfassung . . . . .	11
6.2	Stärken und Schwächen . . . . .	11
6.3	Ausblick . . . . .	11

## Abbildungsverzeichnis

1	Aufteilung der Arbeitspakete . . . . .	4
2	Konzeptskizze // VERALTET . . . . .	6

## 1 Einleitung

### 1.1 Problemdefinition

Neben einer Vielzahl von unstrukturierten Daten entsteht im World Wide Web eine große Wolke mit frei verfügbaren Daten, die per URI (Uniform Resource Identifier) kodiert und verlinkt sind. Diese Linked Open Data (LOD) sind Teil des Semantic Web und besitzt gegenüber konventioneller Datenrepräsentation viele Vorteile.

Das folgende Projekt entsteht in Gruppenarbeit als Teil des Seminarpraktikums „Web 3.0 – Linked Open Data Applications“.

Es hat das Ziel einen lauffähigen Prototyp einer LOD Anwendung zu erstellen, der mindestens zwei LOD Datensätze verwendet und dessen Verwendung einen direkten Nutzen aus diesen Datensätzen zieht. Das Ergebnis soll so flexibel gehalten werden, dass weitere Datensätze potentiell integriert werden können.

### 1.2 Herangehensweise und Ziele

Im Folgenden wird sowohl die Beschreibung der Projektidee als auch die technische Umsetzung geschildert. Ebenso wird kurz auf die Projektplanung eingegangen. Der Fokus liegt speziell auf der Evaluierung der Vorteilhaftigkeit bei der Verwendung von Linked Open Data.

Die Projektidee muss innovativ und gleichzeitig im technischen sowie zeitlichen Rahmen realisierbar sein. Das Ziel soll primär auf der Verwendung von mehreren LOD Datensätzen liegen, deren Benutzung die Vorteile von LOD Datensätze ermöglicht. Des Weiteren wird Wert auf die flexible Erweiterungsmöglichkeit gelegt.

Dem Projektteam ist bewusst, dass das Ergebnis kein marktfähiges Produkt sondern lediglich ein Prototyp darstellen kann. Einbußen bei der Bedienerfreundlichkeit, Geschwindigkeit sowie Fehlerfreiheit werden notwendigerweise in Kauf genommen. Schwächen und Schwierigkeiten bei der Verwendung von LOD werden explizit den Stärken gegenübergestellt.

Dieser Arbeit liegt eine CD bei, die den Quellcode des kompletten Projektes beinhaltet.

## 2 Beschreibung der Idee

### 2.1 Anwendungsszenario

Für die Anwendbarkeit des Programmes gilt es zunächst zwei verschiedene Grundüberlegungen zu separieren. Es stellt sich die Frage, welchen Nutzen das Produkt zum einen für den potentiellen Anwender, also den Spieler, und zum anderen für den Entwickler bzw. den Vertreibenden bietet.

Für den Anwender ist das Programm ein Quiz- oder Lernspiel. Abgefragt werden hauptsächlich geografische Kenntnisse. Dabei sorgt eine Punktevergabe für einen kompetitiven Faktor. Das Spiel positioniert sich somit sowohl als Edutainment-Software<sup>1</sup> als auch als Unterhaltungssoftware für die kurzweilige Ablenkung, z.B. als Facebook-Spiel.

Als Anbieter der Software ist neben der Schaffung einer Einnahmemöglichkeit über Werbeeinblendungen oder Verkauf der Software vor Allem die Generierung von strukturierten Daten interessant.

### 2.2 Spielablauf

Der Spielablauf gliedert sich in zehn Fragerunden. In jeder Fragerunde sucht das Programm drei Fotografien zu einer europäischen Stadt, die gewissen Ansprüchen genügen muss<sup>2</sup>, heraus. Der Spieler muss nun versuchen diese Stadt mit Hilfe der Fotografien zu erkennen und sie auf einer Europakarte mit einem Mausklick möglichst genau zu lokalisieren. Er bekommt dabei weniger Punkte je weiter sein Tipp vom richtigen Zielort entfernt liegt, wobei ab einer gewissen Entfernung pauschal null Punkte vergeben werden.

Weiterhin kann er sich nach Bedarf in jeder Fragerunde im Tausch gegen Punkte drei neue Fotografien oder auch einen 200 Zeichen langen Kurzhinweis anzeigen lassen. Wenn der Spieler mit seinem Tipp eine gewisse Maximalentfernung nicht überschreitet bekommt er darüber hinaus eine Bonusfrage zur aktuellen Stadt oder deren Land gestellt. Diese vermittelt zusätzliche Hintergrundinformationen und gibt dem Spieler die Möglichkeit ein paar Bonuspunkte zu erspielen.

Die so in einer Fragerunde erspielten Punkte werden kumuliert. Ziel des Spiels ist es nach zehn Fragenrunden einen möglichst hohen Punktestand zu erreichen.

---

<sup>1</sup>Hier noch ne schöne Quelle hin

<sup>2</sup>Verweis auf Filterbeschreibung in Kap.4

### 2.3 Alleinstellungsmerkmal und Abgrenzung

Die grundlegende Spielidee wurde so schon in Facebook- sowie iPhone-Apps implementiert<sup>3</sup>. Allerdings setzen diese zur Erzeugung der Fragen auf statische Datenbanken und sind somit im Fragenumfang limitiert. Durch den Einsatz von Linked-Open-Data konnte hier eine dynamische, sich selbstständig aktualisierende Variante geschaffen werden. Desweiteren sorgt der Aufbau des Programmes, gerade in Bezug auf die Bonusfragen, für eine modularisierte Erweiterbarkeit des bestehenden Programmes mit anderen Datenquellen im Semantic Web.

Dazu kommt, dass das Programm keinerlei Anforderungen an den Nutzer stellt, wie die Wahl eines geeigneten Betriebssystems oder einen eingerichteten Account auf der Website. Lediglich ein aktueller Browser wird benötigt.

### 2.4 Semantic Gaming

Dieser Abschnitt gibt einen kurzen Überblick über den Begriff des Semantic Gaming und beschreibt, welche Teile davon für uns von Interesse sind. Die genaue technische Umsetzung findet sich in Kapitel 4.6.

Ein grundlegendes Problem des Semantic Webs ist die, verglichen mit anderen Web 2.0 Anwendungen, relativ geringe Nutzerbeteiligung. Es benötigt diese allerdings, da das Strukturieren der Daten ein Vorgang ist, der zumeist nicht einfach automatisiert werden kann, sondern durch Menschen geschehen muss. Derjenige, der die Daten strukturiert, zieht aber häufig erstmal keinen direkten Nutzen daraus. Zu hoffen, dass das Strukturieren durch reinen Altruismus oder die Bezahlung in irgendeiner Art und Weise befriedigend erfolgt, erscheint wenig vielversprechend. An dieser Stelle setzt das System der Games with a Purpose for the Semantic Web an. Es setzt darauf dem Nutzer bzw. in diesem Fall dem Spieler für das Strukturieren der Daten einen Gegenwert in Form von Spielspaß und einer Art intellektuellen Wettbewerb zu geben.

Diese Strukturierung der Daten passiert dabei unbewusst und im Hintergrund. Die bei uns dabei hauptsächlich vom Nutzer verrichtete Arbeit liegt nach Uschold/King beim hinzufügen von

---

<sup>3</sup>Hier noch ein paar Namen bzw. Beispiele nennen

## 3 Projektplanung

### 3.1 Arbeitspakete und Zuständigkeiten

Innerhalb des Teams wurde zunächst das Ziel formuliert sowie analysiert, welche Möglichkeiten LOD Datensätze bieten. Als Kommunikationsplattform hat sich das Projektteam bewusst auf ein regelmäßiges, wöchentliches Treffen verständigt. Dies diente der Projektplanung, zeitlicher und inhaltlicher Kontrolle sowie der Abstimmung und Zusammenführung individuell erarbeiteter Teilmodule.

Zur Arbeitsteilung wurden die folgenden Arbeitspakete formuliert:

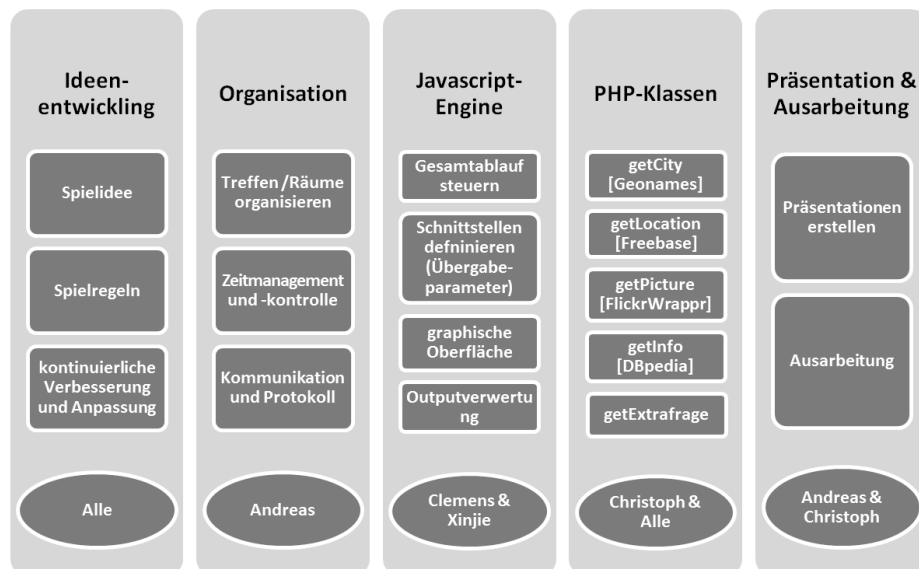


Abbildung 1: Aufteilung der Arbeitspakete

Innerhalb des Projektteams wurde für jede Aufgabe Milestones mit Fristen definiert und einer Person direkt zugeordnet.<sup>4</sup>

### 3.2 Evaluierung

Als vorteilhaft hat sich herausgestellt keine absolute Frist für die Projektidee zu setzen. Aufgrund von Schwierigkeiten mit LOD Datensätzen musste die Idee inkrementell angepasst werden. Die Trennung in Klassen ermöglichte die parallele Ausarbeitung, die mit dem Versionsverwaltungstool Github effizient ermöglicht wurde. Da trotzdem viel Abstimmung nötig war, stellte

<sup>4</sup>Ein Auszug der Liste mit Arbeitspaket, Milestones, Frist und Zuständigkeit ist in der Zwischenpräsentation enthalten.

das wöchentliche Treffen das Kernelement der Projektdurchführung dar. Die Aufnahme des Arbeitspakets „Organisation“ war maßgeblich für die effiziente und erfolgreiche Projektkontrolle verantwortlich. Für unseren Einsatzzweck hat sich das Treffen als effektiveres Mittel herausgestellt als eine zu zeitlich determinierte Gesamtplanung innerhalb eines Projektplanungstools.

Die Unausgeglichenheit der Projektaufgaben aufgrund unterschiedlicher Ausgangspositionen stellte zunächst eine Schwierigkeit dar. Durch klare Verhaltensregeln, spezifische Einarbeitung sowie die Anpassung der Aufgaben an das vorhandene Know-how ist diese Situation nachhaltig verbessert worden.



## 4 Umsetzung, Softwarepakete und Programm

#### 4.1 Schematischer Aufbau, Konzeptskizze

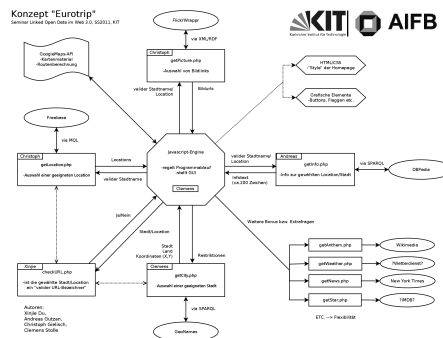


Abbildung 2: Konzeptskizze // VERALTET

Konzeptionell baut das Programm auf einer zentrierten Struktur auf.

## 4.2 Verwendete LOD-Datensätze

Die verwendeten LOD-Datensätze sind:

- **Geonames** wird eingesetzt, um eine Auswahl an Städten zu erzeugen, die der Spieler später dann lokalisieren muss. Diese Datenquelle wurde gegenüber der DBpedia bevorzugt, da bei ihr die Städte klarer strukturiert abgespeichert sind. So fällt die Abfrage der Städte und die Länder in denen diese liegen leichter.
- Die **DBpedia** wird gleich an mehreren Stellen im Programm verwendet. Grundsätzlich gilt, dass wenn Namen von Städten, Ländern oder Sehenswürdigkeiten zwischen verschiedenen Programmteilen getauscht werden, dass diese dann mit ihrem DBpedia URI übergeben werden müssen, um valide zu sein. Somit hat man eine gemeinsame Basis, auf der man kommuniziert. Weiterhin werden sowohl der Kurzhinweis als auch die existierende Bonusfrage mit Hilfe der DBpedia generiert. Auch wird dort mit dem dafür zuständigen Tag geprüft, ob eine Sehenswürdigkeit oder Stadt wirklich eine FlickrWrappy-FotoCollection besitzt.
- **Freebase** dient vor Allem der Optimierung der Foto-Suchergebnisse. Da sich rein mit einer Stadt getaggte Bilder als teilweise ungenügend herausgestellt haben, wird die Freebase zunächst nach bekannten touristischen Sehenswürdigkeiten der einzelnen Städte abgefragt, da diese i.d.R. bessere Suchergebnisse liefern.

- Der **Flickrwrapp** basiert auf einem PHP-Skript der FU Berlin. Man übergibt diesem Skript als Übergabeparameter einen Stadt- oder Sehenswürdigkeitsnamen, welcher dem URI der DBpedia entsprechen sollte. Der Flickrwrapp wiederum antwortet mit Links zu bei Flickr geogetaggten Fotografien als strukturierte Daten in Form einer XML/RDF, die wir wiederum einlesen und verarbeiten.
- Im Zuge der Bonusfrage können außerdem noch **weitere Datensätze** in das bestehende Programm integriert werden. Solange diese die gleiche Schnittstelle wie die existierende Bonusfrage benutzen, können sie ohne weitere Probleme in das Programm integriert werden.

### 4.3 Verwendete Technologien und Frameworks

Neben dem Linked-Open-Data Ansatzes des Semantic Webs benutzt das Programm selbst noch weitere Technologien und Frameworks.

Die verwendeten Technologien sind dabei hauptsächlich Javascript für die eigentliche Engine und PHP für die Abfrageklassen. Zur Kommunikation unterhalb dieser Programmteile wird auf AJAX sowie JSON gesetzt. Serverseitig kommt bei uns ein Apache-Server zum Einsatz.<sup>5</sup> Desweiteren wird die gesamte Webseite natürlich in HTML eingebettet und in einem aktuellen Browser über das Internet ausgeführt.

Als Framework wird vor Allem die "Google Maps API"<sup>6</sup> eingesetzt. Sie liefert uns das Kartenmaterial, dessen Anzeige sowie die Abwicklung der Interaktion des Spielers mit ebenjener. Auch die Routenberechnung läuft über dieses Framework.<sup>7</sup>

### 4.4 Dokumentation der Methoden und Funktionen

GETCITY.PHP		
INPUT	OUTPUT	BESCHREIBUNG
-	Array mit: - Stadtname - DBpedia-URI - Land pro Eintrag	Durchsucht Geonames mit gewissen Filterkriterien nach Städten. So wird nach europäischen Städten mit mehr als 400000 Einwohnern gesucht <sup>8</sup> , die nicht in Russland oder der Ukraine liegen.
GETPICTURES.PHP		

<sup>5</sup>Einfach mal geraten - NACHPRÜFEN!

<sup>6</sup>Verlinkung/Quelle

<sup>7</sup>Haben wir noch weitere Frameworks im Einsatz?

INPUT	OUTPUT	BESCHREIBUNG
-	Array mit: - Stadtname - DBpedia-URI - Land pro Eintrag	Durchsucht Geonames mit gewissen Filterkriterien nach Städten. So wird nach europäischen Städten mit mehr als 400000 Einwohnern gesucht <sup>9</sup> , die nicht in Russland oder der Ukraine liegen.

#### 4.5 Stabilität und Verfügbarkeit

Das Programm an sich läuft trotz seines Prototypen-Status recht stabil. Allerdings ist die Erreichbarkeit von drei Datenquellen für den grundlegenden Ablauf der Fragerunden zwingend erforderlich. Diese sind Geonames, die DBpedia sowie der Flickrwrapp. Ein Ausfall der Freebase würde lediglich die Qualität der Bilder senken, da dann das Heraussuchen von Touristischen Attraktionen wegfallen würde. Sollte einer der Datenbanken der Bonusfragen nicht antworten, wäre ein Fallback auf eine andere Bonusfrage oder das komplette Deaktivieren denkbar. Bei beiden Möglichkeiten ist weiterhin ein stabiler, wenn auch eingeschränkter Betrieb des Programms möglich.

// Hier noch etwas zur Verfügbarkeit der einzelnen Datenquellen

#### 4.6 Generierung strukturierter Datensätze

Ein weiterer technischer Aspekt ist die Generierung eigener strukturierter Daten. Dies ist einer der Grundgedanken sowohl des Semantic Webs als auch des Semantic Gamings. Unser Programm ist dabei in der Lage schon rudimentär neu von uns verknüpfte Daten in Kombination mit Daten, die wir durch die Eingabe der Nutzer gewonnen haben, als strukturierte Daten abzuspeichern. Dabei wird für jede Stadt jeweils ein XML-Datensatz angelegt, der dann befüllt wird mit den aktuellen Sehenswürdigkeiten, Bildlinks sowie der vom Nutzer bei genau dieser Kombination erreichten Entfernung zum Zielort.

Dadurch erschaffen wir eine Datenbank aus der man durch geschicktes Abfragen z.B. mit XPath eine ganz neue Qualität an Information ziehen kann. So sind wir in der Lage bei ausreichend großen Nutzerdateninput durch gemittelte Entfernungen Aussagen über den Bekanntheitsgrad von Städten im Allgemeinen, aber auch deren Sehenswürdigkeiten zu treffen. Denkbar wäre z.B. auch das Blacklisting einzelner Bildlinks als schlechte Fotografien, die eine signifikante Abweichung zu anderen Bildlinks der gleichen Sehenswürdigkeit etc. aufweisen.

Wir sind damit in der Lage das bei LOD grundlegend heikle Thema der

Bewertung von Elementen innerhalb einer Cloud zu verbessern, da man nun mit der Entfernung zumindestens ein Maß für die Bekanntheit dieser Städte und Sehenswürdigkeiten besitzt.

Zusammenfassend ist zu sagen, dass die Datensätze so wie wir sie im Moment abspeichern exakt dem entsprechen, was wir zu Beginn der Arbeiten eigentlich als Datenbank gebraucht hätten. Dies dürfte jedoch dem Grundgedanken des Semantic Webs entsprechen, da dieses seine Mächtigkeit ja gerade durch die nahezu unendliche Möglichkeit der Kombination und Verknüpfung von bestehenden und das Einpflegen von neuen strukturierten Daten bezieht.

#### **4.7 Installation und Betrieb**

Das Spiel selbst benötigt keine clientseitige Installation. Einzig ein funktionierender, aktueller Browser, der in der Lage ist Javascript auszuführen, ist von Nöten. Der Start erfolgt durch Ansteuerung der Webadresse.

Für den Betrieb des Spiels ist der Anbieter des Programms allerdings gezwungen einen Webserver zu betreiben. Dieser muss in der Lage sein PHP-Code zu interpretieren und so die nötigen Abfragen auf die verschiedenen Datenquellen auszuführen.

## **5 Lessons learned**

### **5.1 Reflektion der Projektplanung**

### **5.2 Erkenntnisse**

### **5.3 Vor- und Nachteile von Linked Open Data**

## **6 Fazit**

### **6.1 Zusammenfassung**

### **6.2 Stärken und Schwächen**

### **6.3 Ausblick**