

# Distributed Discussions in Online Social Networks

Masterarbeit

Florian Müller

Betreuer: Prof. Dr. Max Mühlhäuser

Verantwortlicher Mitarbeiter: Dipl.-Inform. Kai Höver

Darmstadt, 6. August 2013



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Informatik  
Telekooperation  
Prof. Dr. Max Mühlhäuser



---

# Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 6. August 2013

---

(Florian Müller)



---

# Zusammenfassung

Inhalt...



---

## Inhaltsverzeichnis

1	Einführung	3
2	Verwandte Arbeiten	5
2.1	Tweetdeck ähnliche Ansätze . . . . .	5
2.2	Reclaim . . . . .	5
2.3	SIOC - Semantic Interlinked Online Communities . . . . .	5
3	Anforderungsanalyse	7
4	Design	9
4.1	Wahl des Zwischenformats . . . . .	9
4.1.1	RSS . . . . .	9
4.1.2	Dublin Core . . . . .	9
4.1.3	SIOC und FOAF . . . . .	9
4.2	Social Online Community Connectors . . . . .	9
4.2.1	Aufbau . . . . .	9
4.2.2	SIOC Service Auth Ontology . . . . .	9
4.2.3	SOCC ConnectorCFG Ontology . . . . .	9
4.2.4	W3 Web ACL Ontology . . . . .	9
4.3	Routen der Einträge . . . . .	9
5	Implementierung	11
5.1	Verwalten SIOC Daten . . . . .	11
5.2	Zugriff auf Online Netzwerke und Abbildung in SIOC . . . . .	11
5.2.1	Moodle . . . . .	11
5.2.2	Canvas LMS . . . . .	11
5.2.3	Facebook . . . . .	11
5.2.4	Google Plus . . . . .	11
5.2.5	Youtupe . . . . .	11
5.3	Implementieren der Connectoren . . . . .	11
6	Fazit	13
A	Hi, Appendix	17





---

# ToDo

	<b>P</b>
1. Hier fehlt ein Anfang . . . . .	7
2. Quelle suchen . . . . .	8



---

# 1 Einführung

- Einleitung (Definition des Problems, Einbettung in das Forschungsfeld)
- Beschreibung der Problemstellung (Welches Problem soll erarbeitet werden?)
- Lernen in sozialen Netzwerken
- Nicht jeder ist in jeden Netzwerk vertreten
- Backup/ Synchronisierung zwischen Netzwerken



---

## 2 Verwandte Arbeiten

- Verwendete Ansätze, Methoden und/oder Modelle (Sprachen, Entwurfsmethoden, Datenmodelle, Analysemethoden, Formalismen)
- Kritische Würdigung (Vergleich mit verwandten Arbeiten, Diskussion offener Punkte und insbesondere auch Ihre persönliche Bewertung)

---

2.1 Tweetdeck ähnliche Ansätze

---

---

2.2 Reclaim

---

---

2.3 SIOC - Semantic Interlinked Online Communities

---



---

## 3 Anforderungsanalyse

### 1. Hier fehlt ein Anfang

Soziale Netzwerk A speichert die Daten der Nachricht in sein eigens Format A. Um diese Beiträge in das soziale Netzwerk B transferieren zu können, müssen zuerst die Daten über eine API von den Servern des sozialen Netzwerks A herunter laden. Da in der Regel nicht automatisch bekannt ist, wann eine neue Nachricht vorhanden ist, müssen die Server in zeitlichen Abständen abgefragt werden und die zurückgelieferten Daten nach neuen Beiträge durchsucht werden. Wurden neue Beiträge gefunden können diese nicht direkt an das soziale Netzwerk B geschickt werden, da sich diese in der Regel im verwendeten Format unterscheiden. Die einfachste Möglichkeit wäre nun die Daten von Format A nach Format B zu konvertieren. Bei zwei Netzen ist dies noch sehr einfach. Es müsste lediglich nur ein Konverter von A nach B und einer von B nach A implementiert werden. Kämme nun ein drittes Netzwerk C hinzu, wären sechs Konverter nötig ( $A \Rightarrow B$ ,  $A \Rightarrow C$ ,  $B \Rightarrow A$ ,  $B \Rightarrow C$ ,  $C \Rightarrow A$  und  $C \Rightarrow B$ ). Nimmt man an  $n_{sn}$  sei eine beliebige Anzahl sozialer Netzwerke, entspricht die Anzahl der Konverter  $n_{k1} = n_{sn} * (n_{sn} - 1)$ , da für jedes Netzwerk ein Konverter in alle anderen Netzwerke nötig wird. Aber auch die Konverter selber sind eine Herausforderung für sich. Nicht alle Formate sind gleich aufgebaut. Im einfachsten Fall unterscheiden sich die einzelnen Daten, aber ist es auch möglich das ein Netzwerk die Features eines Anderen nicht oder in einer abgewandelten Form anbietet. Als Beispiel wäre die Bewertung einzelner Beträge zu nennen. Ein Netzwerk nutzt hier einen, oft anzutreffenden, null bis fünf Sterne Bereiche, ein Anderes setzt auf positive und negative Bewertungen.

Eine elegantere Methode, welche sowohl das Problem mit der starken Zunahme an Konvertern als auch der ähnlichen beziehungsweise unterschiedlichen Features lösen könnte, wäre die Einführung eines Zwischenformates. Alle anfallenden Daten würden erst einmal in dieses Zwischenformat und später in das entsprechende Zielformat konvertiert. Dadurch müssten für jedes neue soziale Netzwerk zwei Konverter, für das Schreiben und Lesen dieses Zwischenformates, implementiert werden. Die Anzahl an Konvertern bezieht sich dem zufolge auf  $n_{k2} = n_{sb} * 2$ . Nachteile hätte dieser Ansatz nur für  $n_{sn} = 2$  und  $n_{sn} = 3$ , da in diesen Fällen mehr beziehungsweise gleich viele Konverter gegenüber der ersten Methode nötig wären. Aber schön für geringfügig höhere Werte sinkt die Anzahl der Konverter signifikant. Für  $n_{sn} = 4$  wären es  $n_{k2} = 8$  statt  $n_{k1} = 12$  und für  $n_{sn} = 5$  ergibt sich  $n_{k2} = 10$  statt  $n_{k1} = 20$  Konvertern. Demzufolge verringert sich auch der Implementierungsaufwand wenn mehrere Netzwerke unterstützt werden sollen.

Liegen nun alle Daten in diesem Zwischenformat können diese an einen zentralen Ort gespeichert werden. So ist es möglich diese Daten für die verschiedensten Zwecke weiter zu verarbeiten. Die Suche nach ähnlichen oder inhaltlich zusammenhängenden Beiträgen aus verschiedenen Quellen. Durch das gemeinsame Format ist dies einfacher möglich als sich erst mit den unterschiedlichen Formaten der einzelnen Quellen auseinandersetzen zu müssen.

Da mit diesem Projekt auch eine automatische Synchronisation neuer Beträge zwischen unterschiedlichen sozialen Netzwerken ermöglicht werden soll, muss das Eintreffen eines neuen Betrags an die entsprechenden Stellen weitergereicht werden. Da dieser Zeitpunkt nicht im Vorneherein bekannt ist würde sich hier ein Event-basierter Ansatz anbieten. So entsteht eine

---

zeitliche Entkopplung des Lesens und Schreibens von Beiträgen. Der Einsatz eines „Publish-Subscribe“-Mechanismus wäre hier ebenfalls von großem Vorteil, da so die Beiträge eines sozialen Netzwerkes mit mehreren Synchronisiert werden kann.

Wurde nun ein Event, dass ein neuer Beitrag gelesen wurde, abgeschickt wird dieser über den besagten „Publish-Subscribe“-Mechanismus an alle dadurch verbundenen Endstellen weitergeleitet. Dieses konvertieren nun den Beitrag in das entsprechende Format des sozialen Netzwerkes und holen sich alle nötigen Informationen um diesen dort hin zuschicken. Eine wünschenswerte Erweiterung hierbei wäre **2. Quelle suchen**, wenn der so geschriebene Beitrag dann so erscheinen würde, als hätte der Autor des ursprünglichen Beitrags ihn selber geschrieben. Hierzu müssten die einzelnen Benutzer das System, zum Beispiel durch das Hinterlegen von Accesstoken<sup>1</sup> oder anderen Authentifizierungsparameter, autorisieren in ihren Namen Nachrichten zu schreiben. Diese könnten dann anhand der Benutzerkennung herausgesucht und benutzt werden.

---

<sup>1</sup> Siehe <http://oauth.net/>



---

## 4 Design

- Die sollen die Anforderungen aus Kapitel 3 letztendlich in ein funktionstüchtiges Programm umgesetzt werden.

---

### 4.1 Wahl des Zwischenformats

---

- Warum kein eigenes Format
- Warum nicht RSS
- Warum nicht Dublincore + X
- Warum SIOC + FOAF + X

---

#### 4.1.1 RSS

---

---

#### 4.1.2 Dublin Core

---

---

#### 4.1.3 SIOC und FOAF

---

---

### 4.2 Social Online Community Connectors

---

Einleitung über den Sinn und Zweck eines Connectors in SOCC

---

#### 4.2.1 Aufbau

---

---

#### 4.2.2 SIOC Service Auth Ontology

---

---

#### 4.2.3 SOCC ConnectorCFG Ontology

---

---

#### 4.2.4 W3 Web ACL Ontology

---

---

### 4.3 Routen der Einträge

---

- Eigener Ansatz mit JMS
- Apache Camel



---

# 5 Implementierung

- Probleme und deren Lösung während der Umsetzung
- Beispiel: Zugriff auf Moodle über Webservice

---

## 5.1 Verwalten SIOC Daten

---

- Sesame, Jena -> RDF2Go
- In Memory, XML/RDF File, Tripplestore

---

## 5.2 Zugriff auf Online Netzwerke und Abbildung in SIOC

---

---

### 5.2.1 Moodle

---

---

### 5.2.2 Canvas LMS

---

---

### 5.2.3 Facebook

---

---

### 5.2.4 Google Plus

---

---

### 5.2.5 Youtupe

---

---

## 5.3 Implementieren der Connectoren

---

-





---

## 6 Fazit



---

# Literaturverzeichnis





---

# A Hi, Appendix

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then...