

# Distributed Discussions in Online Social Networks

## Masterarbeit

Florian Müller

Betreuer: Prof. Dr. Max Mühlhäuser

Verantwortlicher Mitarbeiter: Dipl.-Inform. Kai Höver

Darmstadt, 10. 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 10. August 2013

---

(Florian Müller)



---

# Zusammenfassung

Inhalt...



---

## Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>3</b>
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>5</b>
2.1	Tweetdeck ähnliche Ansätze . . . . .	5
2.2	Reclaim . . . . .	5
2.3	SIOC - Semantic Interlinked Online Communities . . . . .	5
<b>3</b>	<b>Anforderungsanalyse</b>	<b>7</b>
3.1	Neuen Beitrag verfassen . . . . .	7
3.2	Beiträge von sozialen Netzwerk A lesen . . . . .	7
3.3	Beitrag an soziales Netzwerk B weiterleiten . . . . .	9
<b>4</b>	<b>Design</b>	<b>11</b>
4.1	Wahl des Zwischenformats . . . . .	11
4.1.1	RSS . . . . .	11
4.1.2	Dublin Core . . . . .	11
4.1.3	SIOC und FOAF . . . . .	11
4.2	Social Online Community Connectors . . . . .	11
4.2.1	Aufbau . . . . .	11
4.2.2	SIOC Service Auth Ontology . . . . .	11
4.2.3	SOCC ConnectorCFG Ontology . . . . .	11
4.2.4	W3 Web ACL Ontology . . . . .	11
4.3	Routen der Einträge . . . . .	11
<b>5</b>	<b>Implementierung</b>	<b>13</b>
5.1	Verwalten SIOC Daten . . . . .	13
5.2	Zugriff auf Online Netzwerke und Abbildung in SIOC . . . . .	13
5.2.1	Moodle . . . . .	13
5.2.2	Canvas LMS . . . . .	13
5.2.3	Facebook . . . . .	13
5.2.4	Google Plus . . . . .	13
5.2.5	Youtupe . . . . .	13
5.3	Implementieren der Connectoren . . . . .	13
<b>6</b>	<b>Fazit</b>	<b>15</b>
<b>A</b>	<b>Anhang</b>	<b>19</b>
A.1	Anforderungsanalyse Ablaufdiagramm . . . . .	19





---

# ToDo

1. Quelle suchen . . . . .	<b>P</b>
	<b>9</b>



---

# 1 Motivation

- 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

Um sich eine Vorstellung davon zu machen, wie ein System auszusehen hat und welche Komponenten dazu nötig sind um zwei oder mehrere sozialen Netzwerke zu verbinden, soll hierzu ein kleines Beispiel<sup>1</sup> konstruiert werden.

### 3.1 Neuen Beitrag verfassen

Alles beginnt damit, dass ein zum Beispiel ein Student im sozialen Netzwerk A im Forum zur Veranstaltung Telekooperation 1 eine Frage zur aktuellen Übung stellen will. Er geht zuerst in den passenden Thread und beginnt einen neuen Beitrag zu schreiben. Sobald er fertig ist, klickt er auf „Absenden“ und sein Beitrag wird in der Datenbank des sozialen Netzwerkes A gespeichert und als neuer Eintrag im Thread angezeigt (siehe Abbildung 3.1).

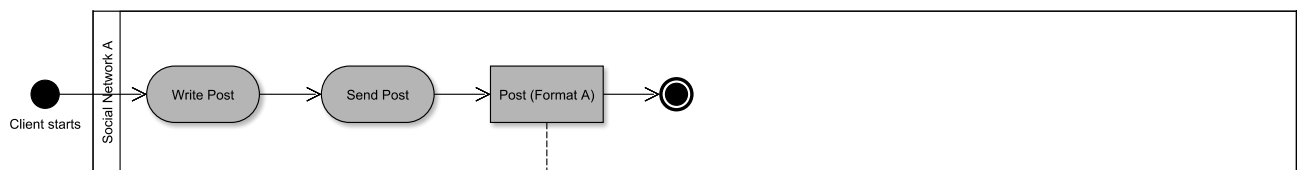


Abbildung 3.1.: Benutzer erstellt einen Beitrag im sozialen Netzwerk A.

### 3.2 Beiträge von sozialen Netzwerk A lesen

Um diese Beitrag in das soziale Netzwerk B transferieren zu können, müssen zuerst die Daten über eine öffentliche Schnittstelle von den Servern des sozialen Netzwerkes A heruntergeladen werden. Da in der Regel nicht automatisch bekannt ist, wann ein neuer Beitrag vorhanden ist, müssen die Server in zeitlichen Abständen abgefragt (polling genannt) und die zurückgelieferten Daten nach neuen Beiträgen durchsucht werden. Sind ein oder mehrere neue Beiträge gefunden worden, können diese nicht direkt an das soziale Netzwerk B geschickt werden, da sich diese in der Regel im verwendeten Datenformat unterscheiden. Diese müssen zuvor konvertiert werden.

Die einfachste Möglichkeit wäre nun die Daten von Format A nach Format B zu konvertieren. Bei zwei Formaten ist dies noch sehr einfach. Es müsste lediglich ein Konverter von Format A nach Format B und einer in die umgekehrte Richtung implementiert werden. Für den Fall, dass nun ein weiteres Netzwerk C unterstützt werden soll, würde ich die Anzahl an nötigen Konvertern auf Sechs erhöhen, wie Tabelle 3.1 zeigt.

Nimmt man an  $n_{sn}$  sei eine beliebige Anzahl Netzwerke, entspricht die Anzahl der notwendiger Konverter  $n_{k1} = n_{sn} * (n_{sn} - 1)$ , da für jedes Netzwerk ein Konverter in alle anderen Netzwerke

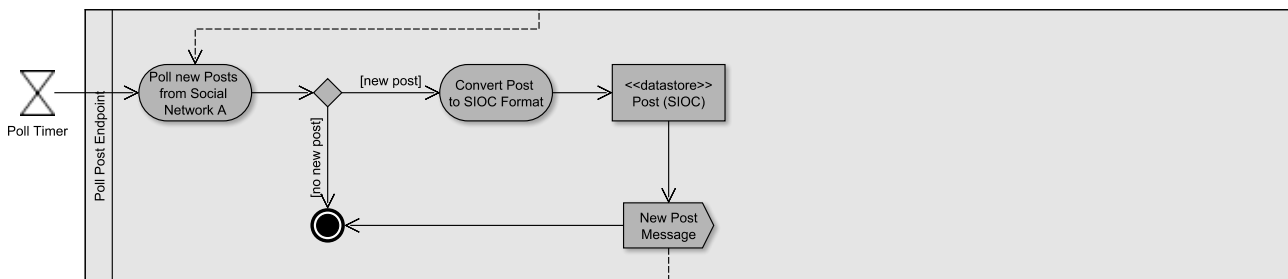
<sup>1</sup> Das vollständige Ablaufdiagramm befindet sich im Anhang A.1

$A \Rightarrow B$	$A \Rightarrow C$
$B \Rightarrow A$	$B \Rightarrow C$
$C \Rightarrow A$	$C \Rightarrow B$

**Tabelle 3.1.:** Konverter bei Drei sozialen Netzwerken

erzeugt werden muss. Sollen nur Zwei oder Drei Netzwerke unterstützt werden ist der Aufwand noch sehr überschaubar, bei mehr kann dies aber sehr Aufwendig werden.

Eine elegantere Methode, welche die Anzahl zu implementierender Konverter in Grenzen halten kann, wäre die Einführung eines Zwischenformates. Geht man davon aus, dass die Daten aller Netzwerke nur in dieses Zwischenformat geschrieben und aus diesem gelesen werden müssen, würde sich der Aufwand auf maximal Zwei Konverter pro neuem Netzwerk reduzieren. Für eine beliebige Anzahl Netzwerke wären also  $n_{k2} = n_{sn} * 2$  Konverter nötig. 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 erforderlich wären. Erhöht man die Anzahl Netzwerke jedoch nur geringfügig, sinkt die Menge an Konvertern sichtbar. 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. Gleichzeitig können so syntaktische Unterschiede in den einzelnen Formaten angeglichen werden, was sie leichter handhabbar macht.



**Abbildung 3.2.:** Lesen des erstellten Beitrags und konvertieren in das Zwischenformat.

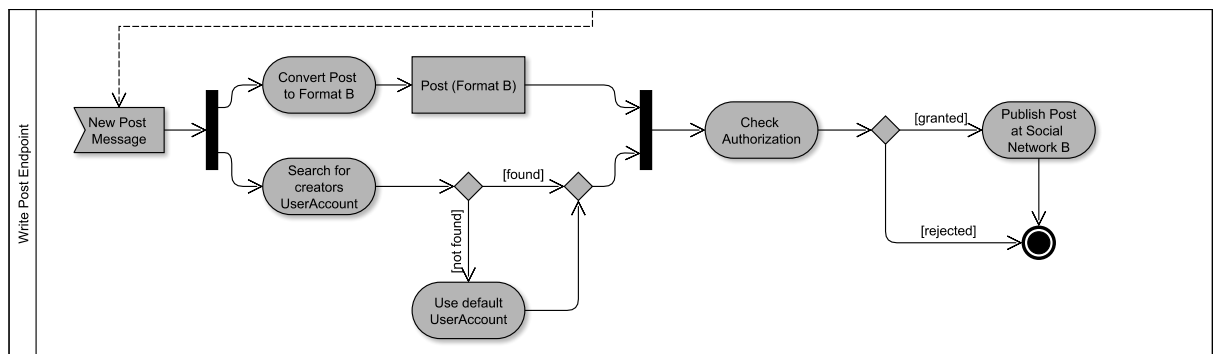
Liegen nun alle Daten in diesem Zwischenformat können diese an einen zentralen Ort gespeichert werden. So ist es denkbar diese Daten zum Beispiel für die Suche nach ähnlichen oder inhaltlich zusammenhängenden Beiträgen weiter zu verwenden. Die Möglichkeiten sind hier vielfältig.

Da mit diesem Projekt auch eine automatische Synchronisation neuer Beiträge zwischen unterschiedlichen sozialen Netzwerken ermöglicht werden soll, muss das Eintreffen eines neuen Beitrags an die entsprechenden Stellen weitergereicht werden. Da dieser Zeitpunkt nicht im Vorherigen 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 großen Vorteil, da so die Beiträge eines sozialen Netzwerkes mit mehreren Synchronisiert werden kann. Der komplette Ablauf des Lesens, Konvertierens und bekanntmachen neuer Beiträge wird in Abbildung 3.2 noch einmal graphisch dargestellt.



### 3.3 Beitrag an soziales Netzwerk B weiterleiten

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 soziale Netzwerks und holen sich alle nötigen Information um diesen dort hin zuschicken. Eine wünschenswerte Erweiterung hierbei wäre **1. 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 Access Token<sup>2</sup> oder anderen Authentifizierungsparameter, autorisieren in ihren Namen Nachrichten zu schreiben. Diese könnten dann anhand der Benutzerkennung herausgesucht und benutzt werden.



**Abbildung 3.3.:** Konvertierten des Beitrags in das Format B und schreiben in das soziale Netzwerk B

<sup>2</sup> <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

## Abbildungsverzeichnis

3.1	Benutzer erstellt einen Beitrag im sozialen Netzwerk A. . . . .	7
3.2	Lesen des erstellten Beitrags und konvertieren in das Zwischenformat. . . . .	8
3.3	Konvertierten des Beitrags in das Format B und schreiben in das soziale Netzwerk B	9

## Tabellenverzeichnis

3.1	Konverter bei Drei sozialen Netzwerken . . . . .	8
-----	--	---



# A Anhang

## A.1 Anforderungsanalyse Ablaufdiagramm

