KIT – Karlsruhe Institut of Technology

Praxis der Software-Entwicklung WS 2009/10

Gruppe 7

»Pflichtenheft« v 1.0

Analysesoftware für Soziale Netzwerke

20.11.2009

Phase	Phasenverantwortlich	e-mail
Pflichtenheft	Mark Engel	mark.engel@student.kit.edu
Entwurf	Barbara Sepic	barbara.sepic@student.kit.edu
Implementierung	Marc Typke	marc.typke@student.kit.edu
Testen	Felix Stahlberg	felix.stahlberg@student.kit.edu
Präsentation	Mark Engel	mark.engel@student.kit.edu

Inhaltverzeichnis

1 Einleitung 4
2 Zielbestimmung 5
2.1 Muss-Kritierien 5
2.2 Wunschkriterien 7
2.3 Abgrenzungskriterien 8
3 Produkt-Einsatz 9
3.1 Anwendungsbereich 9
3.2 Zielgruppe 9
3.3 Betriebsbedingungen 9
4 Produktumgebung 10
4.1 Software 10
4.2 Hardware 10
5 Produktfunktionen 11
5.1 Grundfunktionen 11
5.2 Erweiterte Funktionen 11
6 Produkt-Daten 13
6.1 System-Daten 13
6.2 Benutzer-Daten 13
7 Systemmodell 13
8 Produkt-Leistungen 15
9 Benutzeroberfläche 21
9.1 Einführung 21
9.2 Benutzerführung 22
9.2.1 Userinterface (normale User) 22
9.2.2 Userinterface (Administratorsicht) 24
10 Qualitätsbestimmungen 26
11 Testfälle und Test-Szenarien 27
11.1 Testfälle 27
11.1.1 Initialisierung 27
11.1.2 Benutzerinteraktion 27
11.1.3 interne Funktionen 27

11.2 Test-Szenarien 29
11.2.1 Testszenario 1 29
11.2.2 Testszenario 2 29
12 Entwicklungs-Umgebung 30
12.1 Software 30
12.2 Hardware 30
12.2.1 Server 30
12.2.2 User 30
13 Glossar 31

1 Einleitung

Ziel des Projektes ist es, eine Software zu entwickeln, die bereits vorhandenes Datenmaterial über soziale Netzwerke strukturiert und als Graph visualisiert.
Bei sozialen Netzwerken im informatorischen Sinne handelt es sich um Netzgemeinschaften oder Webdienste, welche Netzgemeinschaften beherbergen.
Es handelt sich um ein Phänomen, das Mitte der 1990er Jahre aufkam und seit ca.

2003 einen absoluten Boom erlebt.
Dies manifestiert sich sowohl in Nutzerzahlen, als auch in den Geldsummen, die Unternehmen zahlen um sich bei sozialen Netzwerken einzukaufen.
Die genaue wirtschaftliche sowie soziale Bedeutung sozialer Netzwerke ist nach wie vor nicht realistisch zu bemessen.

Zu den größten sozialen Netzwerken gehören Facebook und MySpace.
Beide zählen jeweils um die 300 Millionen Nutzer und haben geschätzte Gegenwerte

von mehreren 100 Millionen Dollar.
Das Interesse liegt hier klar bei den werberelevanten Zielgruppen, welche direkt über soziale Netzwerke ansprechbar sind.
Es gibt also definitiv ein ökonomisches Interesse an sozialen Netzwerken, aber auch für ein breites Spektrum in der Forschung (z.B. Spieltheorie, Sozialpsychologie oder Kommunikationswissenschaft) sind soziale Netzwerke interessant.
Ein Programm, das es ermöglicht, trockene Analysezahlen zu veranschaulichen, ihnen eine Struktur zu geben und sie miteinander in Verbindung zu setzen, ist definitiv ein Pflichtprodukt für Jeden, der ein soziales Netzwerk unterhält.

2 Zielbestimmung

2.1 Muss-Kritierien

2.1.1 Allgemeine Kriterien

Die Kernfunktionalität des Produktes besteht ganz allgemein darin, dass es lesend auf Datenbanken sozialer Netzwerke zugreift. Zum einen zeigt das Programm verschiedene Relationen zwischen den Benutzern (lokale Zentralitätsmaße) und zum Anderen jedem Nutzer absolut zuordbare Werte (globale Zentralitätsmaße) durch Graphen visualisiert an.
Es gibt also zwei Arten von Zentralitätsmaßen. Globale Zentralitätsmaße sind absolute Werte, die für jeden Knoten seperat ermittelt werden können, lokale Zentralitätsmaße sind Werte, die erst aus der Verknüpfung zweier Knoten entstehen. Das Produkt unterstützt beide Arten von Maßen.
Das Netzwerk wird mittels eines Graphen visualisiert. Die Knoten dieses Graphen stehen dabei jeweils für einen Spieler oder einen Benutzer, die Kanten repräsentieren die Beziehungen der Benutzer zueinander. Globale Zentralitätsmaße werden mittels Größe, Farbe oder Ähnlichem der Knoten visualisiert. In jedem Graphen wird also mindestens ein Zentralitätsmaß dargestellt.
Diese Beziehungen können gerichtet oder ungerichtet sowie gewichtet beziehungsweise ungewichtet sein.
Die Ausprägungen, Charakteristiken und Strukturen von Zentralitätsmaßen können in verschiedenen sozialen Netzwerken sehr unterschiedlich ausfallen, deshalb wird mittels einer Plugin-Struktur ermöglicht, nicht nur neue soziale Netzwerke hinzuzufügen (sofern sie den Vorraussetzungen entsprechen, siehe Kapitel Abgrenzungskriterien), sondern auch die jeweiligen Zentralitätsmaße eines sozialen Netzwerkes mittels XML-Dateien und Java-Plugins (genauere Spezifikation weiter unten) zu spezifizieren.
Desweiteren sind funktionierende Plugins für die Datenbanken der sozialen Netzwerke von Cosim und Postgame im Lieferumfang enthalten.

2.1.2 Benutzer Kriterien

Das Produkt kennt zwei Arten von Benutzern. Normale Benutzer und einen Administrator.
Normale Benutzer sehen nach dem Einloggen einen Graphen, in dem sich ihr Knoten zentral befindet und alle (über das aktuell ausgewählte Zentralitätsmaß) verknüpften Knoten um ihn herum.
Ein normaler Benutzer kann den Graphen nach den zur Verfügung stehenden Zentralitätsmaßen verändern, woraufhin der Graph anhand der neuen Kritierien neu berechnet wird. So kann er z.B. zwischen den Zentralitätsmaßen Aktivität seiner Freunden und deren Punktzahl wählen, worauf sich jeweils der Graph entsprechend aktualisiert. Außerdem kann der Benutzer (falls für dieses Zentralitätsmaß eine Historie existiert) den Graphen auf einen bestimmten Zeitraum beschränken. In diesem Fall wird der Durchschnitt dieses Zeitraums berechnet und angezeigt.
Die Credentials der Nutzer kommen aus den eingebundenen Datenbanken sozialer Netzwerke, so ist eine doppelte Speicherung der Daten nicht nötig.
Ein Administrator genießt den vollen Funktionsumfang normaler Nutzer. Da er aber keine eigene Knotenrepräsentation besitzt, werden bei ihm stattdessen die zentralsten Knoten in der Mitte angezeigt. Er kann sich außerdem globale Zentralitätsmaße anzeigen lassen (bei kleinen Datensätzen sogar für den ganzen Graphen), dabei liegen in der Nähe des Mittelpunktes die zentralsten Knoten.
Über den Funktionsumfang für normale Nutzer hinaus kann ein Adminstrator die Systemkonfiguration ändern, indem er neue Netzwerke (über die Eingabe der Verbindungsdaten zum jeweiligen Datenbankserver) in die Grapheneinsicht einbindet und den jeweiligen Nutzern des sozialen Netzwerkes über die Loginmaske zur Verfügung stellt.
Außerdem kann er neue Netzwerke einbinden, indem er die entsprechende Datenbankverbindung und das verwendete (vorher über die Plugin-Struktur definierte) Datenbankschema angibt. Mehr Informationen zu Plugins und die Erweiterungen für neue Netzwerke im Kapitel 6. Systemmodell.

2.2 Wunschkriterien

Die Wunschkriterien sind in der Reihenfolge ihrer Präferenz gelistet.

- Die Benutzeroberfläche ist mehrsprachig (Internationalisierung). Unterstützte Sprachen sind Deutsch, Englisch, Französisch und Slowenisch.
- Oft benutzte oder aufwendig zu ermittelnde Kantenanfragen werden in der applikationseigenen Datenbank gecached, um das Programm performant zu halten.
    - Der Einsatz von Caching hängt davon ab, ob das Programm ohne Caching performant (oder vielleicht sogar performanter) als mit Caching ist. Was genau gecacht wird davon abhängen an welchen Stellen Bottlenecks auftauchen werden
    - Mehr Informationen zum Cachingsystem im Kapitel 7.3.
- Ein Benutzer kann die Knoten des Graphen verschieben. Dies ist zum Beispiel sinnvoll, wenn sich viele Kanten überschneiden und durch Verschieben die Überschneidungen reduziert werden können. Hierbei muss berücksichtigt werden, dass womöglich ein Zentralitätsmaß durch Entfernung zum zentralen Knoten dargestellt werden und die Verschiebung dann nur auf der entsprechenden Kreislinie möglich ist. Die Verschiebungen werden in der applikationseigenen Datenbank abgelegt und stehen so für neue Graphengenerierungen bereit.
- Ein Benutzer kann Ansichten der Graphen speichern und laden. Das beinhaltet die darzustellende Zentralitätsmaße, deren Darstellungsart und deren Wertebereiche sowie die obere Grenze für die Anzahl der dargestellten Knoten, sofern dies vom Nutzer konfigurierbar ist (Wunschkriterium).
    - Die Ansicht wird auf dem Server gespeichert. Sodass der Nutzer auch von verschiedenen Workstations, sowie unter unterschiedlichen Browsern die gleichen Ansichten laden kann,
    - Der Benutzer kann mehrere Ansichten gleichzeitig speichern und jede gespeicherte Graphen-Ansicht laden.
- Der Benutzer kann zwischen unterschiedlichen optischen Darstellungen der Zentralitätsmaße wählen. Möglich wäre die Darstellung durch Entfernung, dicke der Kante, die Größe der Knoten oder auch der Farbe (Abbildung auf Rot- Grün- und Blauanteile).
- Es kann eine obere Grenze für die Anzahl dargestellter Knoten festgelegt werden. Dadurch wird es möglich, den Graphen bei einer Vielzahl von Knoten übersichtlich darzustellen, ohne die Relevanz der Daten zu verlieren (Unwichtigere Knoten, also Knoten mit schwachen Zentralitätsmaßen werden hier ausgeblendet). Eine obere Grenze wird durch den Adminstrator festgelegt, Nutzer können unterhalb dieser eigene obere Grenzen setzen.
- Die Knotenanzahl kann auch durch Angabe der maximalen Hops begrenzt werden.
- Mehrerer Zentralitätsmaße können gleichzeitig angezeigt werden. Dabei besitzen die unterschiedlichen Maße jeweils unterschiedliche der oben genannten Darstellungsarten.
- Zeitliche Entwicklung von Zentralitätsmaßen können als Film dargestellt werden.

2.3 Abgrenzungskriterien

Das System wird unter der Annahme entwickelt, dass die zu bearbeitende Anzahl von Knoten und Kanten eine Summe von 10.000 Datensätzen nicht überschreitet.
Für die Einbindung der sozialen Netzwerke wird prinzipiell jedes DBMS unterstützt, das von JDBC unterstützt wird und sich an den SQL-92 Standard hält. Garantiert wird jedoch nur die Unterstützung von Oracle 11 oder MySQL 5.
Zur Userauthentifizierung wird vorausgesetzt, dass es in jeder dieser Datenbank eine Tabelle mit einer eindeutigen ID, einem Namen und dem Passwort (entweder im Plain-Text oder mit einem Verschlüsselungsverfahren wie MD5, SHA oder Unix's crypt() Funktion) eines jeden Nutzers existiert. Dies ermöglicht, sich mit den gleichen Credentials, die sie auch im sozialen Netzwerk benutzen, ins System einzuloggen.
Es können nur Zentralitätsmaße unterstützt werden, die mit der gegebenen Plugin Struktur realisiert werden können. Mehr Informationen hierzu im Kapitel 7.2.

3 Produkt-Einsatz

3.1 Anwendungsbereich

Dieses Programm kann zur Analyse sowie zur Visualisierung sozialer Netzwerke benutzt.
Einerseits wird hiermit Administratoren, Wissenschaftlern und Marktforschern die Möglichkeit gegeben, die sozialen Verbindungen und Gemeinsamkeiten anschaulich visualisiert zu betrachten.
Andererseits kann dieses Programm auch auf Nutzerebene verwendet werden. Nutzer des jeweiligen Netzwerkes haben hiermit die Möglichkeit, sich über ihre sozialen Kontakte eine Übersicht zu bilden.

3.2 Zielgruppe

Die Zielgruppe sind Personen, die sich eine Übersicht über ein soziales Netzwerk verschaffen wollen.
Mögliche Zielgruppen sind wissenschaftliche Auswertung von Forschungsarbeiten Benutzung in der Marktforschung innerhalb sozialer Netzwerke, direkt von den Benutzern sozialer Netzwerke, um einen besseren Überblick über die sozialen Verknüpfungen zu bekommen
Diese Anwendung kann prinzipiell von jedem Nutzer eines sozialen Netzwerkes benutzt werden um soziale "Verknüfungen" zwischen den Benutzern zu visualisieren.

3.3 Betriebsbedingungen

Für die Benutzung des Webinterfaces wird ein aktueller Browser vorausgesetzt. Genauer ausgedrückt wird ein Browser benötigt, der durch das Google Web Toolkit und die in 4.1 genannten Bibliotheken unterstützt wird.
Garantiert wird Kompatibilität zu folgenden Browsern.

- Mozilla Firefox >= 2
- Opera >= 9.25
- Safari >= 3.0
- Chrome >= 1.0
- Interne Explorer >= 7

4 Produktumgebung

4.1 Software

Serverseitig

- Apache Tomcat 6.0 / optional Apache 2 Web Server mit mod_jk vorgeschaltet
- Datenbanken mit Nutzerdaten in vorher definierten Schemen (siehe 2.1 Musskriterien), Unterstützung der DBMSe MySQL und Oracle
- Java 1.6

Als Browser setzen wir einen aktuellen Browser (mindestens einen aus Kapitel 3.3) mit aktiviertem JavaScript vorraus.
Implementiert wird das Produkt mittels des Google Web Toolkits (GWT) 1.7, ist ein Framework von Google für Webanwendungen in Java. Des Weiteren wird GQuery, was eine an das GWT angepasste Version von jQuery ist. jQuery ist ein umfangreiches JavaScript Framework, das browserunabhängiges Schreiben von JavaScript sowie Manipulation des DOM-Trees ermöglicht.
Zusätzlich benötigen wir noch eine Bibliothek, mit der die Graphen letzendlich für den User im Browser optisch aufbereitet werden. Hier bieten sich momentan CanViz (ein Derivat von GraphViz) und GWT-Diagrams (einen Graphenimplemtation direkt für das GWT) an. Welche Graphenbibliothek schlussendlich verwendet wird hängt von Performancetests und Entwurfsentscheidungen ab.

4.2 Hardware

Serverseitig

- Beliebige Serversysteme, auf denen die unter 4.1 geforderten Anwendungen effizient lauffähig und installiert sind.

Clientseitig

- Hardwaremindestanforderungen entsprechen denen der unterstützten Browser (siehe 3.3 Betriebsbedingungen).

5 Produktfunktionen

5.1 Grundfunktionen

Nr	Beschreibung
/F010/	Das Programm muss in der Lage sein, die im Kapitel 7.2 umrissenen XML-Dateien zu parsen
/F020/	Mit Hilfe der Informationen aus den XML-Dateien und den über das Webinterface eingegebenen Credentials zu den Datenbanken ist es dem Produkt möglich, sich mit den Datenbanken zu verbinden
/F030/	Des Weiteren können die Datensätze aus der Datenbank in das Programm eingelesen werden und es können gemäß der zur Verfügung gestellten Zentralitätsmaße passende Datensätze aus der Datenbank geladen werden.
/F040/	Diese Daten werden in eine interne Struktur übersetzt. Genaueres in Kapitel 7.1.
/F050/	Aus diesem internen Format muss es möglich sein, die Daten in ein von CanViz oder GWT-Diagrams lesbares Format zu konvertieren um die Daten im Browser des Users anzeigen zu können.
/F060/	Der Zugriff auf die aufbereiten Graphen geschieht über einen Webbrowser
/F070/	Der Nutzer hat die Möglichkeit sich die Daten über einen bestimmten Zeitraum anzeigen zu lassen (falls dieser Zeitraum größer als 1 Zeiteinheit ist, wird der Durchschnitt der Werte gebildet)
/F080/	Verfügbarkeit globale Zentralitätsmaße. Globale Zentralitätsmaße werden einem einzigen Knoten zugeordnet.
/F090/	Verfügbarkeit lokaler Zentralitätsmaße. Lokale Zentralitätsmaße werden als Kanten zwischen 2 Knoten repräsentiert

5.2 Gewünschte Funktionen

Nr	Beschreibung
/W010/	Mehrsprachige Benutzeroberfläche
/W020/	Caching oft verwendeter und aufwendig zu ermittelnder Kantengewichte
/W030/	Benutzer können die Knoten des Graphen per Drag und Drop verschieben
/W040/	Die momentan angezeigte Ansicht des Graphen kann gespeichert und später wieder geladen werden *soll vermerkt werden, dass es auch mehree ansichten möglich sind oder ist das ein seperater punkt
/W050/	Einstellung beliebiger optischer Konfiguration der Kanten (beispielsweise Farbe, dicke der Kanten oder Abstand der Knoten)
/W060/	Einstellung beliebiger optischer Konfiguration der Knoten, bei Wahl eines lokalen Zentralitätsmaßes (Größe, Form, Farbe des Knoten)
/W070/	Einstellung der Anzahl der dargestellten Knoten über die Wahl eines globalen Maximums über den Administrator
/W080/	Einstellung der Anzahl der dargestellten Knoten über den Benutzer (zwischen 1 und globalen Maximum des Administrators)
/W090/	Begrenzung der Anzahl der dargestellten Knoten durch die Wahl der maximalen Hops (Beispiel: Zeige nur Kontakte an, die eine maximale Distanz von 2 Knoten zu mir haben)
/W100/	Visualisierung von mehreren lokalen Zentralitätsmaßen gleichzeitig
/W110/	Zeitliche Entwicklung des Graphen als Film

6 Produkt-Daten

6.1 System-Daten

Das Programm benötigt zum produktiven Einsatz Plugins jedes zu unterstützenden Netzwerks. Ein Plugin besteht aus einer XML-Datei und optional einigen Java-Dateien (kompiliert als bytecode vorliegend - .class). Genaueres zu Plugins in Kapitel 7.2.
Zusätzlich werden in einer internen Datenbank die Verbindungsinformationen zu den aktiven Datenbanken sozialer Netzwerke gespeichert.
Das Produkt wird mit einer Dokumentation der Software im Wiki der Redmine Projektverwaltung ausgeliefert.
Optional kommen noch Bilder sowie Logos hinzu.
Falls die Software, gemäß der Wunschkriterien internationalisiert wird, müssen die Übersetzungen gespeichert werden.
Eine minimale Konfigurationsdatei, die die Datenbankverbindung zur applikationseigenen Datenbank und einige Verzeichnisangaben beinhaltet, wird nötig sein.

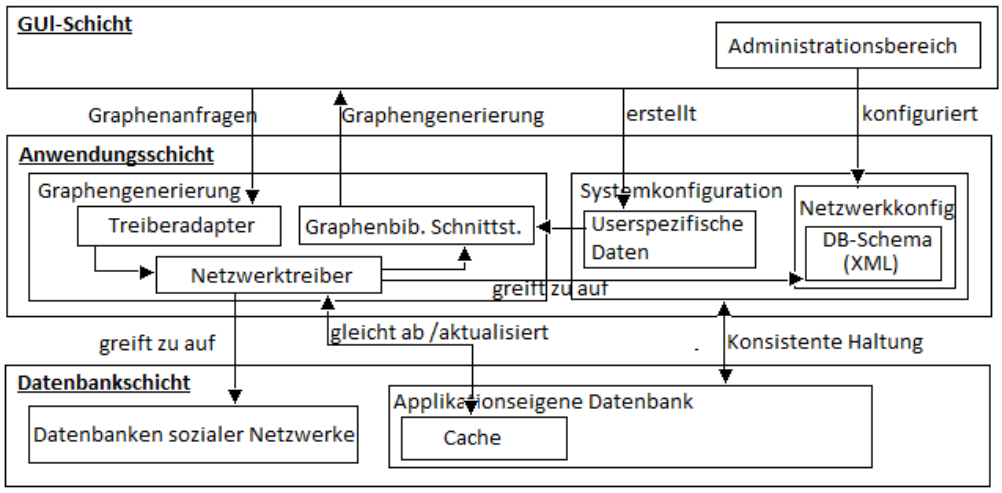
6.2 Benutzer-Daten

Als Benutzerdaten werden die Zugangsdaten des Superusers gespeichert. Zugangsdaten normaler User werden von den eingebundenen sozialen Netzwerken übernommen. Falls verschiebbare Knoten und das Speichern dieser Ansichten implementiert wird, werden entsprechende Daten in der applikationseigenen Datenbank abgelegt.

7 Systemmodell

7.1 Systemarchitektur

Die Software strukturiert sich durch eine intransparente 3-Schichten-Architektur (Benutzeroberfläche, Anwendungsschicht und Datenbankschicht), bei der die Anwendungsschicht intern partitioniert ist in Graphengenerierung und Systemkonfiguration.



Anmerkung zum Diagramm: Die Benutzrelation in einer Schichtenarchitektur ist zwischen den Schichten azyklisch, was dem vorliegendem Diagramm nach zu urteilen hier nicht der Fall ist. Das liegt daran, dass das Diagramm nicht nur Benutztrelationen zeigt, sondern auch Datenflüsse veranschaulicht. So benutzt der Netzwerktreiber zum Beispiel den Cache, gleicht Kantenanfragen mit ihm ab und aktualisiert ihn, jedoch benutzt der Cache nicht den Netzwerktreiber. Ähnlich ist bei allen anderen Pfeilen, die nach oben zeigen, zu argumentieren. Die Benutztrelation ist also auch im vorliegenden Fall linear und nicht zyklisch.
Die Hauptfunktion des Produktes ist die Generierung von Graphen. Diese wird in der GUI-Schicht vom User angestoßen und von einem Treiberadapter an den internen Netzwerktreiber weitergereicht. Der Netzwerktreiber holt sich Informationen zu dem darzustellenden Zentralitätsmaß aus dem Konfigurations XML und bearbeitet entweder die Graphenanfrage selbst oder reicht die Anfrage an ein PlugIn weiter (hier nicht dargestellt, da nicht teil des Systemkerns). Zudem arbeitet der Netzwerktreiber mit dem Cache, was in 7.3 näher erklärt ist. Der Netzwerktreiber reicht seine Ergebnisse zu der internen Implementierung der Schnittstelle der verwendeten Graphenbibliothek weiter, die unter Verwendung gespeicherter Userspezifischen Daten (wie Verschiebungen von Knoten) die Graphenbibliothek bedient, die so den Graph letzendlich dem User darstellen kann.

7.2 Plugin System

Plugins dienen dazu, neue soziale Netzwerke unterstützen zu können. Plugins müssen in einem bestimmten Verzeichnis unter einer bestimmten Struktur auf dem Server abgelegt werden. Dazu gibt es einen Plugin Ordner, in dem jedes Plugin in einem eigenen Unterordner liegt. In diesem Unterordner liegen beliebig viele Java .class Dateien und die Datei plugin.xml.
Anmerkung zu diesem Kapitel: Das hier Gezeigte ist eher als Entwurf zu verstehen. Im Laufe der Entwicklung ändern sich möglicherweise Benennungen oder Schnittstellen leicht, wenn zum Beispiel im Entwurf dafür Gründe gefunden werden.

7.2.1 plugin.xml

Die plugin.xml definiert das Plugin und gibt an, welche Zentralitätsmaße wie beziehungsweise wo implementiert werden. An einem Beispiel wird die grobe Struktur dieser Datei erklärt.
Wir wollen das folgende vereinfachte soziale Netzwerk mit folgendem Datenbankschema einbinden.

User
uid: Int
name: String
pass: String

Pictures
pid: Int
owner: Int
created: Date

Comments
cid: Int
picture: Int
author: Int
created: Date

Credits
uid: Int
capital: Float

User können Bilder hochladen, die von anderen Usern kommentiert werden können. Zusätzlich hat jeder User ein Konto, auf dem er Credits sammelt.
Jede plugin.xml besitzt mindestens die folgende Struktur, die den Namen, eine Beschreibung und Informationen zur Userverwaltung (entsprechende Tabelle und die zu verwendenen Felder in der Datenbank) definiert.

```
<common>
  <name>PictureVZ</name>
  <description>Einfaches Bilderportal</description>
  <user>
    <table>user</table>
    <login>name</login>
    <password>pass</password>
    <encryption>md5</encryption>
   </user>
</common>
```

Zusätzlich werden nun die implementierten Zentralitätsmaße definiert. Zur Implementierung können wir nun zwei Strategien verfolgen.

7.2.2 Implementierung mittels Java-Klassen

Die Implementierung eines Zentralitätsmaßen kann über Java-Klassen geschehen, die von bestimmten abstrakten Klassen erben und die Funktionalität implementieren. Diese Plugins können auf den Cache (siehe 7.3) und mit der korrespondierenden Datenbank mittels einer gestellten Schnittstelle kommunizieren. Es muss zwischen globalen und lokalen Zentralitätsmaßen unterschieden werden, da Implementierungen unterschiedliche Schnittstellen bereitstellen müssen. Die referenzierten Java-Klassen sollten wie oben beschrieben in dem vorgesehenden Ordner liegen. Ein Beispiel für ein globales Zentralitätsmaß sähe zum Beispiel so aus (der FQCN verweist hier auf die Java-Klasse im Plugin Ordner), mit lokalen Zentralitätsmaßen ist hier analog zu verfahren,

```
<measure-global>

  <name>Bilderanzahl</name>

  <description>Anzahl der Bilder, die ein User hochgeladen
hat</description>


<implementation>socialview.plugins.PictureVZ.PictureCount</implementat
ion>

</measure-global>
```

7.2.3 Implementierung mittels XML-Referenzen (Wunschkriterium)

Zudem besteht die Möglichkeit, bei der Implementierung auf Java-Klassen zu verzichten. Das hat verschiedene Vorteile.
Für einfach zu implementierende Zentralitätsmaße spart man das Schreiben von Code

- Das Caching (siehe 7.3) wird vom Netzwerktreiber übernommen, man bekommt gute Cachingheuristiken mitgeliefert
- Die Übersichtlichkeit ist bei vielen Java-Klassen, die jede Schnittstelle vielleicht schon mit einem Befehl implementieren können, nicht mehr gewährleistet
- Allerdings kann dieser Weg nicht bei allen Zentralitätsmaßen gegangen werden, da die Berechnung möglicherweise zu komplex ist um in die nun folgende XML Struktur übersetzt zu werden. Grob gesagt sollte aber die Mächtigkeit des hier vorgestellten Konzepts für fast alle Zentralitätsmaße reichen, die mit einem SELECT-Befehl mit normalen Joins über mehrere Tabellen berechnet werden können.

Man definiere zunächst die relevanten Tabellen und deren Schlüssel/Fremdschlüssel.

```
<table name="user">
  <keys>
    <key>uid</key>
  </keys>
</table>
<table name="pictures">
  <keys>
    <key>pid</key>
  </keys>
  <foreignKeys>
    <key for="user">owner</key>
  </foreignKeys>
  <timeField>created</timeField>
</table>
<table name="comments">
  <keys>
    <key>cid</key>
  </keys>
  <foreignKeys>
    <key for="pictures">picture</key>
    <key for="user">author</key>
  </foreignKeys>
  <timeField>created</timeField>
</table>
<table name="credits">
  <foreignKeys>
    <key for="user">uid</key>
  </foreignKeys>
</table>
```

Es folgen exemplarisch Definitionen zwei möglicher Zentralitätsmaße.

```
<measure-global>
  <name>Credits</name>
  <description>Aktueller Creditsstand eines Users</desciption>
  <implementation>
    <tableRef>credits</tableRef>
    <field>capital</field>
    <via>
      <tableRef>user</tableRef>
      <key>uid</key>
    </via>
  </implementation>
</measure-global>
<measure-global>
  <name>Kommentare erhalten</name>
  <description>Zählt, wie oft Bilder von diesem User insgesamt
kommentiert wurden</desciption>
  <implementation>
    <tableRef>comments</tableRef>
    <field>cid</field>
    <func>count</func>
    <via>
       <tableRef>pictures</table>
       <key>picture</key>
    </via>
    <via>
       <tableRef>user</tableRef>
       <key>owner</key>
    </via>
  </implementation>
</measure-global>
```

Die Definition lokaler Zentralitätsmaße ist ähnlich, hier muss zusätzlich noch die Ermittlung des "Ziel"users mit angegeben werden.

7.3 Caching (Wunschkriterium)

Das Produkt setzt auf ein zweistufiges Cachingsystem. Auf der ersten Stufe überprüft der Netzwerktreiber, ob genau diese Anfrage schonmal vorher bearbeitet wurde und im Cache liegt. Falls dem so ist, kann auf die Neuberechnung der Kanten verzichtet werden und die Daten aus dem Cache verwendet werden. Befinden sich die Daten nicht im Cache bearbeitet der Netzwerktreiber die Anfrage (siehe 7.2 Plugin System) und schreibt das Ergebnis nach bestimmten Heuristiken (Berechnung sehr lang gedauert / Anfrage wird sehr oft gestellt) in den Cache.
Dies erlaubt jedoch nur ein relativ eingeschränktes Caching, da sehr ähnliche Anfragen nicht davon profitieren. Als Beispiel seien zum Beispiel zwei Anfragen genannt, die sich nur darin unterscheiden, dass bei der ersten der Graph auf 100 Knoten, bei der zweiten auf 101 Knoten begrenzt ist (99 wäre kein Problem). Die Verwendung der Berechnung der ersten Anfrage würde wahrscheinlich einen erheblichen Speed-Up bedeuten, jedoch kann dies nicht im Netzwerktreiber, sondern muss im Plugin implementiert werden. Auf der zweiten Stufe des Cachingsystems können also auch Plugins über eine Schnittstelle den Cache verwenden, um Berechnungen performanter zu gestalten.

8 Produkt-Leistungen

Nr	Beschreibung
/L10/	Nachdem ein Benutzer sich eingeloggt hat, muss in weniger als einer Sekunde die Seite aufgebaut werden.
/L20/	Look and Feel: auf jede Benutzeraktion erhält der Benutzer sofort Rückmeldung von der Benutzeroberfläche zur Empfangsbestätigung. Dies geschieht meist durch Ausführen der entsprechenden Aktion. Falls dies zeitnah nicht möglich ist muss das dem Nutzer kenntlich gemacht werden.
/L30/	/L20/ gilt insbesondere bei der Graphengenerierung. Falls ein Graph nicht in weniger als einer Sekunde generiert werden kann, wird der Nutzer darauf mithilfe einer Wartebox oder Ähnlichem hingewiesen.
/L40/	Ein Graph mit 30 Knoten und einem Zentralitätsmaß muss in weniger als einer Sekunde aufgebaut werden können, sofern das Zentralitätsmaß nicht außergewöhnlich aufwendig zu berechnen ist.
/L50/	Die Anzeige eines Graphen mit bis zu 1000 Knoten und einem Zentralitätsmaß, das entweder einfach zu berechnen ist (zum Beispiel ein Feld in der Usertabelle) oder dessen Berechnung gecachet ist (Wunschkriterium, siehe 7.3) muss in weniger als fünf Sekunden gezeichnet werden.
/L60/	Änderungen der Anzeigeeinstellungen des Graphen wirken sich unter Berücksichtigung von /L40/ und /L50/ Just-in-Time auf den Graphen aus.

9 Benutzeroberfläche

9.1 Einführung

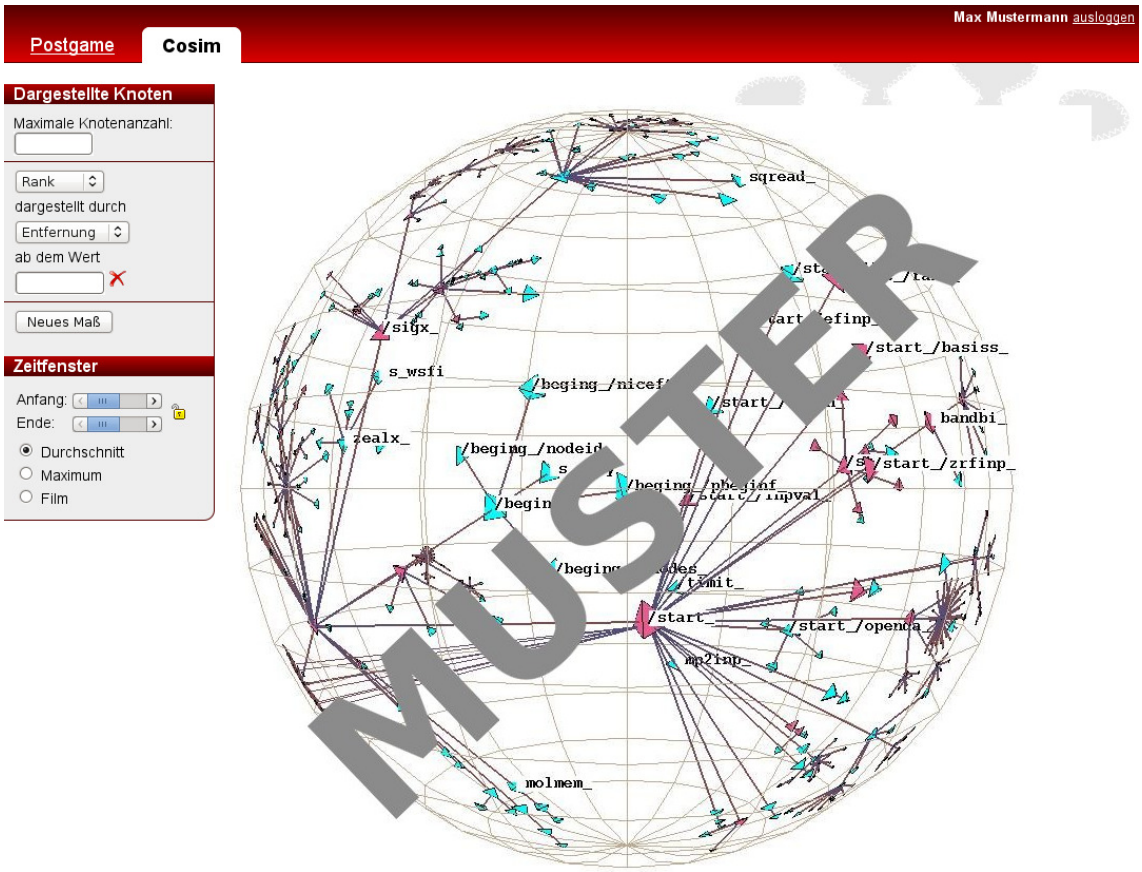
Die Herausforderung bei dem Entwurf des Userinterfaces liegt darin, dem User eine klar strukturierte und intuitive Benutzeroberfläche zu bieten, ohne jedoch die Möglichkeit zu versperren, komplexe und aufwendig zu berechnende Graphen darstellen zu können. Diese Gradwanderung wird vor Allem mit der Einhaltung der folgenden Designprinzipien realisiert:

Prinzip	**Beispiel im Userinterface**
Positionierung ähnlicher Elemente nah beieinander	Konfiguration der Graphendarstellung in einem Fenster, klar abgegrenzt von der Netzwerkauswahl
Verwendung bekannter Bedienelemente	Verwendung von Tabs bei der Netzwerkauswahl, Logout oben rechts, Links unterstrichen
Konsistenz	Die Darstellung jedes Zentralitätsmaßes wird auf gleiche Weise konfiguriert

9.2 Benutzerführung

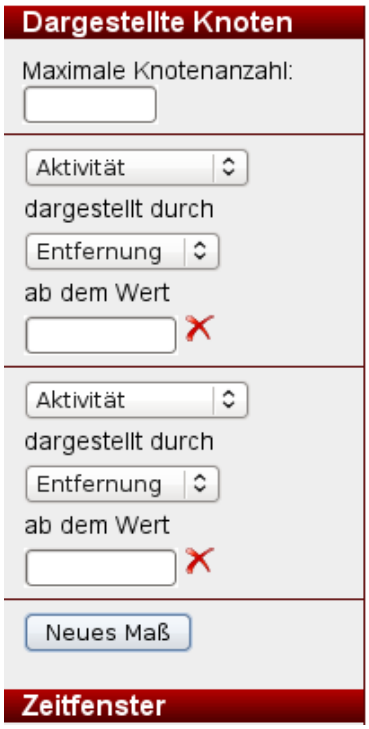
Den Benutzer erwartet zunächst eine Login-Maske. Nach dem Login stehen (abhängig davon, ob der Benutzer Administratorrechte besitzt oder nicht) weitere Funktionalität zur Verfügung.

9.2.1 Userinterface (normale User)



Nach dem Login erscheint die Hauptansicht des ersten konfigurierten Netzwerks. Die Tableiste oben, in der sich alle konfigurierten Netzwerke befinden, zeigt dem User, wo er sich in der Navigation momentan befindet. Zentrales Element ist der Graph, der durch das Konfigurationsfenster links angepasst werden kann (Wunschkriterium!). Dieses Fenster unterteilt sich in die Bereiche *Dargestellte Knoten* und *Zeitfenster,* die jeweils durch Klicks auf die Überschriften auf- und zugeklappt werden können.

9.2.1.1 Dargestellte Knoten (Wunschkriterium)



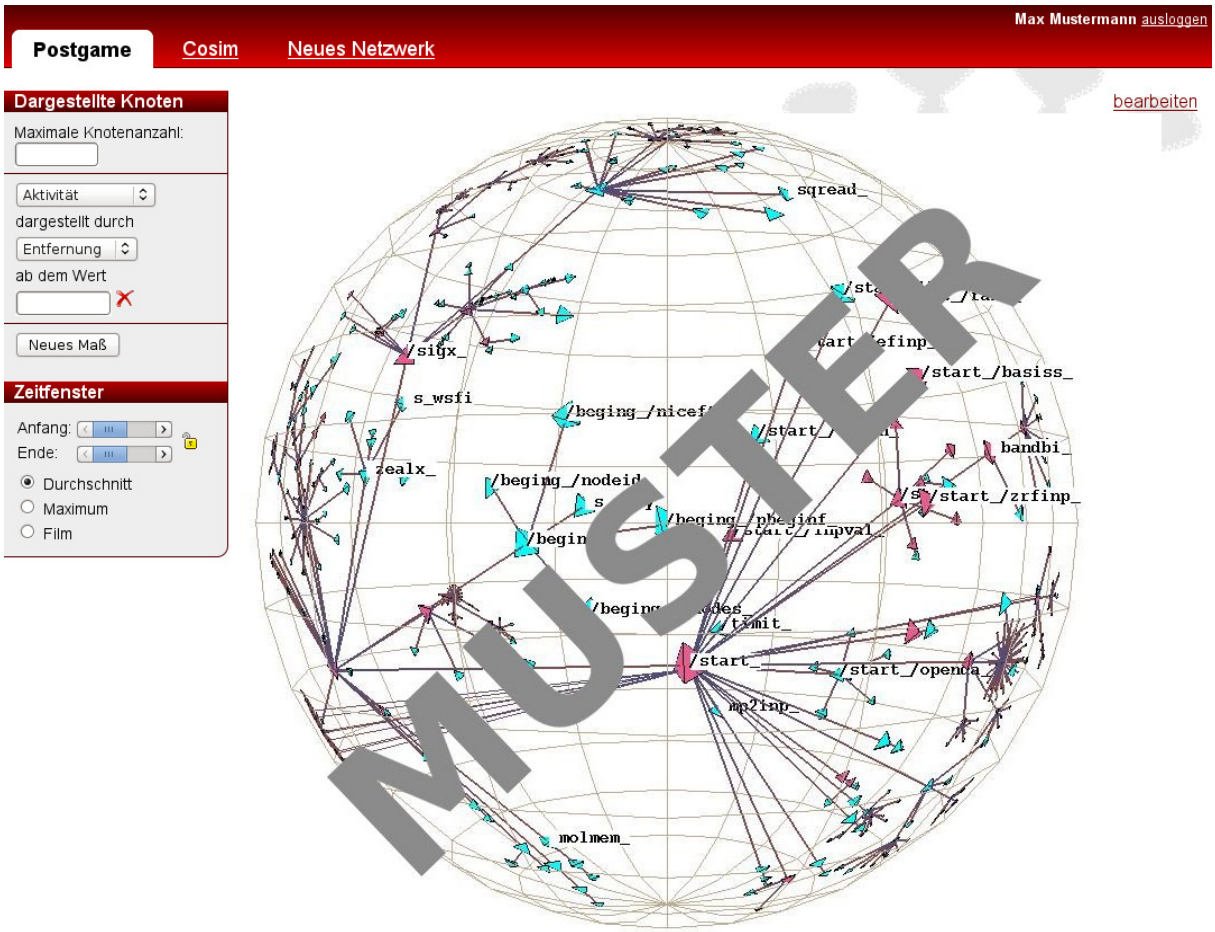
Hier werden die im Graph darzustellende Knoten und Kanten definiert. Hierfür kann der User das Zentralitätsmaß, die Darstellungsart der Kanten im Graph und einen Wertebereich angeben, um die Knotenanzahl einschränken zu können. Sollen mehrere Zentralitätsmaße angezeigt werden lässt sich diese Eingabemaske durch einen Klick auf *Neues Maß* entsprechend erweitern.

9.2.1.2 Zeitfenster (Wunschkriterium)

Hier kann (falls vorhanden) das Zeitfenster eingestellt werden. Wird hier ein Intervall gewählt, so kann zwischen folgenden Darstellungsarten gewählt werden.

- Durchschnitt: Kantengewichte zeigen den Durchschnitt des zu zeigenden Zentralitätsmaßes über den Zeitraum an.
- Maximum: Kantengewichte zeigen das Maximum der über den Zeitraum an erreichten Zentralitätsmaße an.
- Film: Zeitliche Entwicklung wird als Film dargestellt.
  Mit einem Klick auf das Schloss werden die Scrollleisten für den Anfangs- und Endwert des Zeitfensters gekoppelt, es wird also z.B. nur noch eine einzige Runde dargestellt.

9.2.2 Userinterface (Administratorsicht)



Zusätzlich zum normalen Userinterface stehen für Administratoren noch die Optionen *bearbeiten* und *Neues Netzwerk* zur Verfügung.
Des Weiteren muss ein Administrator in der Lage sein, bereits eingerichtet Netzwerke und Datenbankschemen aus der Anwendung zu entfernen.

9.2.2.1 Netzwerk bearbeiten

Die Konfiguration eines Netzwerks besteht aus seinem eindeutigen Namen, der Datenbankverbindung und der Angabe des Datenbankschemas. Für den letzten Punkt steht dem Administrator eine Auswahl schon vorhandener Datenbankschemen zur Auswahl. Wird ein neues Datenbankschema benötigt, kann mit einem Klick auf *Neues Schma hochladen* eine XML-Datei hochgeladen werden, die dieses Schema definiert.

9.2.2.2 Neues Netzwerk

Mit einem Klick auf *Neues Netzwerk* gelangt der Administrator zur der Eingabemaske, mit der neue Netzwerke angelegt werden können. Diese Maske ist konsistent mit 9.2.1.1.

10 Qualitätsbestimmungen

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Robustheit	x			
Zuverlässigkeit	x			
Korrektheit			x	
Benutzerfreundlichkeit	x			
Effizienz			x	
Portierbarkeit	x			
Kompatibilität	x			

Das Programm soll so robust wie möglich sein. Das bedeutet, dass selbst wenn ein Nutzer nicht erwartete Anfragen startet diese im schlimmsten Fall mit einer Fehlermeldung quittiert werden, wobei die Sessions anderer Nutzer davon unbeinflusst sind und die Server-Application weiterläuft.
Die angezeigten Graphen können, gerade im Hinblick auf Performance, auch approximiert werden, da gerade bei Webanwendungen, die größtenteils auf dem Server berechnet werden, eine schnelle Reaktionszeit wichtig ist. Gerade auch um die Anwendung performant zu halten.
Das Userinterface soll so einfach und intuitiv wie möglich gehalten sein. Die Design-Principles zum Userinterface können in Kapitel 9 genau nachgelesen werden.
Das Produkt soll gerade im Hinblick auf die Effizienz eine gute Performance garantieren, solange die unter 2.3 genannten Kriterien eingehalten wurden.
Da das Programm in Java mittels dem Google Web Toolkit implementiert wird und deshalb in jedem Standardbrowser (siehe 3.3) lauffähig ist, ist bereits durch die verwendete Architektur eine Portabilität gegeben.
Einhaltung von Webstandards (in Widgets) und die Kompatibilität mit einer breiten Masse von Browsern wird ebendso vom Google Web Toolkit induziert. An Stellen, an denen dies nicht geschieht (wie zum Beispiel css Angaben), wird durch uns jedoch nicht mit restriktiveren Einschränkungen zu rechnen sein.

11 Testfälle und Test-Szenarien

11.1 Testfälle

Testfälle werden durch Junit realisiert.
Falls Interaktionen mit der Benutzerschnittstelle benötigt sind, werden Browsereingaben mittels eines Frameworks wie HtmlUnit realisiert um ein automatisches Testen zu ermöglichen.

11.1.1 Initialisierung

- Einrichten des Superusers (beim ersten Aufrufen des unkonfigurierten Programms)
    - Danach ist ein Einloggen mit den registrierten Login-Daten möglich
- Einloggen mit den Daten des Superusers
    - Nach dem Einloggen muss im Browser die Hauptseite des Administrators angezeigt werden
- Einbinden einer Spieldatenbank
    - Implentiert eine neue Instanz einer Spieldatenbank

11.1.2 Benutzerinteraktion

- Versuch sich mit falschen Benutzerdaten einzuloggen
    - Zeigt Fehlermeldung an
- Einloggen mit Benutzerdaten
    - Zeigt Hauptseite der GUI an (aus User-Sicht)
- Änderung des Zentralitätsmaßes
    - Der Graph wird unter den geänderten Kriterien neu aufgebaut
- (Graph verschieben)
    - Ein Knoten befindet sich an einer anderen Position
- Speichern des modifizierten Graphen
    - Bekomme visuelle Bestätigung
- Laden des gespeicherten Graphen
    - Der Graph wird geladen (eventuell Überprüfung auf Koordinaten der Knoten)
- Ausloggen
    - Login-Seite wird angezeigt.

11.1.3 Interne Funktionen

- fehlerhafte XML Konfiguration parsen
    - Fehlermeldung und kein Programmabsturz
- XML Konfiguration parsen
    - Ein Objekt mit den geparsten Optionen
- mit geparster Konfiguration mit Datenbank verbinden
    - Resultat sollte ein funktionierendes Datenbank-Objekt sein.
- Rohdaten holen
- Rohdaten in interne Darstellung konvertieren
- interne Darstellung in Graphen-Format konvertieren

11.2 Test-Szenarien

11.2.1 Testszenario 1

Ein Administrator konfiguriert das frischinstallierte Programm.

1. Der Administrator öffnet die Webadresse des Programms in einem unterstützten Browser.
2. In das Setup Menü werden die Login-Daten des Super-Users eingegeben und das Formular abgeschickt
3. Der Administrator loggt sich in das Programm ein
4. Navigation zu Konfiguration -> Datenbanken
5. Auswahl des sozialen Netzwerkes und Eingabe der Verbindungsdaten zum Datenbankserver des Netzwerkes
6. Speichern
7. Kontrolle der Daten durch Ansicht des Graphen
8. Logout

11.2.2 Testszenario 2

Ein Nutzer des Spiels möchte sich eine Übersicht über seinen Account verschaffen

1. Einloggen mit den Userdaten
2. Ansicht des Graphen
3. Ändern der Anzeigekriterien des Graphen
4. Verschieben der Knoten
5. Abspeichern der Ansicht.
6. Logout

12 Entwicklungs-Umgebung

12.1 Software

- Java JDK 1.6
- Eclipse Galileo
  - Checkstyle Plugin - Kontrolle der JCC
  - Google Plugin - GWT Entwicklung
  - Subclipse - SVN Integration
  - Topcased - UML Editor
- Redmine Projekt-Managment Software
  - Verteilung der Tickets
  - Schreiben der Dokumentation mittels eines Wikis
  - Übersicht über die Arbeitszeit
  - Übersicht über die Versionskontrolle
- SVN Server
- Apache Tomcat 6.0
- aktueller Web-Browser (siehe 3.3)
- MySQL- und Oracle-Testdatenbanken mit Daten von soziale Netzwerke

12.2 Hardware

12.2.1 Server

In der Entwicklung wird die Software aus Eclipse in einer virtuellen Maschine gestartet und debuggt.

12.2.2 User

aktueller Browser (siehe 3.3)

13 Glossar

- **Zentralitätsmaß** - Maß für strukturelle Zentralität eines Systemes
  - **lokal** - im Verhältnis zu einem Benutzer (jede Kante hat einen Wert relativ zum Peer)
  - **global** - im Verhältnis zum ganzen System (jede Kante hat genau einen Wert)
- **Hops** - Kontakt von Kontakten
- **GWT** - Google Web Toolkit ist ein Framework zur Entwicklung von Webanwendungen. Seine Besonderheit ist ein Java-nach-Javascript-Compiler, so dass nahezu die gesamte Entwicklung von Client und Server auf Basis von Java realisiert werden kann.
- **jQuery** - jQuery ist ein freies, umfangreiches JavaScript-Framework, das komfortable Funktionen zur DOM-Manipulation und -Navigation zur Verfügung stellt.
- **canviz** - Canviz ist eine JavaScript Bibliothek zum zeichen von Graphviz Graphen in einem Browser.
- **Graphviz** - Graphviz ist ein von AT&T und den Bell-Labs entwickeltes plattformübergreifendes Open-Source-Programmpaket zur Visualisierung von Objekten und deren Beziehungen untereinander
- **Gwt Query** - Gwt Query ist eine jQuery-ähnliche API geschrieben für GWT.
- **Checkstyle** - Programm zur kontrollierten restriktiven Anwendung der JCC
- **JCC - Java Code Conventions** - Regeln für Syntaxformatierung unter Java
- **SVN - Subversion** - Software zur Versionskontrolle
- **FQCN - Fully-qualified class name** - Der volle und in der Applikation eindeutige Name einer Klasse in Java
- **HtmlUnit** - HtmlUnit ist ein GUI-loser Web Browser, mit dem Manipulationen von Webseiten auf einen hohen Level möglich sind.