MEDIENKONZEPTION UND PRODUKTION

FACHHOCHSCHULE KAISERSLAUTERN, STANDORT ZWEIBRÜCKEN

PROJEKT: GAGVIS

ENTWICKLERDOKUMENTATION

GAGVIS.SOURCEFORGE.NET

DANIEL SEILER

FELIX KIEFER

INHALT

1. EINFÜHRUNG

1.1 VORWORT

Dieses Dokument beschreibt den Umgang und die Verwendung der Anwendung GAGVis und soll vor allem Programmierern als Grundlage zur Weiterentwicklung dienen. Es enthält sowohl allgemeine Informationen über die Anwendung, als auch detaillierte Informationen über die Softwarearchitektur, die einzelnen Packages und Klassen.

1.2 EINLEITUNG

‚Deutschland AG', so nennt man das Netzwerk von Verbindungen von deutschen Unternehmen untereinander. Häufig sind Manager nicht nur im Vorstand oder Aufsichtsrat einer Firma, sondern leiten gleich mehrere. Dadruch entstehen Beziehungen unter den einzelnen Konzernen, die von aussen oft gar nicht zu erkennen sind. Betrachtet man nun auch noch Abgeordnete des Deutschen Bundestages, die nicht selten neben ihrem Politikerdasein auch noch in den Chefetagen von Konzernen sitzen, so entstehtein schier undurchschaubares Geflecht von Verbindungen.

Dieses Geflecht zu entwirren und klar aufzuzeigen ist die Aufgabe von ‚GAGVis'.

1.3 GAGVIS

Das Projekt GAGVis enstand im Rahmen der Veranstaltung Medienkonzpetion und Produktion der Fachhoschule Kaiserslautern, Standort Zweibrücken. Es hat das Ziel die Zusammenhänge zwischen den einzelnen deutschen Unternehmen untereinander zu visualisieren und aufzuzeigen welcher Politiker in welcher Firma mitwirkt. Basierend auf der Hochschuldatenbank der Hoppenstedt AG und den Offenlegungen der Nebeneinkünfte der Abgeordneten, ist es möglich das Netzwerk der ‚Deutschland AG' zu visualisieren.

Der Name GAGVis setzt sich aus der Abkürzungen ‚GAG' = Germany AG und ‚Vis' = Visualisierung zusammen. Es ist eine webbasierte Anwendung die auf eine HSQL-Datenbank (www.hsqldb.org) zugreift und mit der Programmiersprache Java erstellt wurde. Zur Visualisierung dient das Opensource-Framework ‚Prefuse' (prefuse.org).

GAGVis selbst unterliegt auch einer Opensourcelizens und wird auf dem Entwicklerportal Sourceforge.net (www.sourceforge.net) gehostet.

GAGVis selbst unterliegt auch einer Opensourcelizens und wird auf dem Entwicklerportal Sourceforge.net gehostet.

1.4 Lizenz und Haftungsauschluss

GAGVis wurde unter der GNU Library or Lesser General Public License (LGPL) entwickelt. Die Anwendung behält sich vor weder vollkommen korrekt noch auf dem aktuellsten Stand zu sein. Bei der Entwicklung wurde als Datengrundlage auf die Daten der Firma Hoppenstedt aus dem Jahre 2007, sowie die auf Spiegel Online veröffentlichten Politikergehälter zurückgegriffen (http://www.spiegel.de/politik/deutschland/0,1518,492613,00.html).

1.5 Weiterentwicklung

Das Entwicklerteam von GAGVis bietet jedem Interessenten die Möglichkeit den Quellcode nach seinen eigenen Vorstellungen anzupassen und weiterzuentwickeln. Durch die sauber aufgebaute Struktur der Software im Gesamten und eine ausführliche Kommentierung des Quellcodes wird der Einstieg in die Weiterentwicklung erleichtert. Des weiteren steht in der Dokumentation die JavaDoc zur verfügung. Bei Detailfragen sind die beiden Entwickler Daniel Seiler und Felix Kiefer per Mail jederzeit erreichbar.

2. Anforderungen, Installation und Verwendung

2.1 Systemvorraussetzung für die Entwicklung

Zur Weiterentwicklung von GagVis benötigen Sie folgende Anforderungen:

- Java Development Kit ab Version 5.0 der Java Standard Edition J2SE
- Visualisierungstoolkit Prefuse
- JFreeChart (Werkzeug zur Darstellungen der Statistiken)
- Jakarta http Client
- HSQLDB Datenbank
- Java Entwicklungsumgebung (hier verwendet Eclipse SDK 3.3.0)

Bezugsquellen:

- Java Entwicklungsumgebung: Eclipse 3.2 www.eclipse.org
- Java JDK: www.sun.com

2.2 Installation der Anwendung für Entwickler

Zum einen gibt es die Möglichkeit, auf www.sourceforge.net das komplette Projekt über das CVS Repository auszuchecken oder auf der Website des Projektes, gagvis.sourceforge.net das komplette ZIP-Archiv herunterzuladen.

Wichtig ist die für das Projekt benötigten Bibliotheken im Java Build Path einzubinden. Diese befinden sich in der Zip, wie auch nach einem Checkout unter dem Ordner src/libs.

WICHTIG: Zur Zeit funktioniert das Abfragen von Datenbankinhalten nicht online. Sie müssen sich den Inhalt der HSQLDB-Datenbank von der GAGVis-Homepage herunterladen und in ihr temporäres Verzeichnis des Betriebssystems entpacken.

2.3 Systemvoraussetzungen für die Nutzung

Für die Ausführung der Anwendung müssen folgende Softwareanforderungen erfüllt sein:

- Java Runtime Environment ab Version 5.0.
- Internet Browser (JavaScript aktiviert)

2.4 Installation für die Nutzung

Um das Applet starten zu können, müssen Sie Applets im Allgemeinen für Ihren Web-Browser zulassen (Onlinehilfe zum Aktivieren von Java für den Web-Browser). Als Laufumgebung müssen Sie die Java Runtime Environment JRE1.6 auf Ihrem Computer installieren (JRE1.6 herunterladen, folgen Sie einfach den Installationsanweisungen der JRE1.6).

Öffnen Sie nun in Ihrem Browser die Adresse: gagvis.sourceforge.net/applet.html.

Nachdem Sie die URL geöffnet haben, werden Sie aufgefordert eine Datei auf ihrem PC zu speichern. In ihr sind sämtliche Daten enthalten, auf die die Applikation zugreifen muss. Entpacken Sie das ZIP-Archiv in Ihr Temporäres Verzeichnis (Bsp. für Windows: C:\Dokumente und Einstellungen\‘Benutzername‘\Lokale Einstellungen\Temp). Diese Maßnahme ist nötig, da die Applikation zur Abgabe nicht fertig entwickelt wurde, und der Zugriff auf eine, sich im Internet befindende Datenbank nicht möglich ist.

2.5 Bedienung

Nachdem das Applet geladen wurde sehen wir zunächst ein leeres Display. Wie der Information zu entnehmen ist, kann mit einem Doppelklick auf das Display eine neue Map erstellt werden. Im anschliessenden Screen kann wahlweise nach Politikern, Aufsichtsräten, Vorständen oder Firmen gesucht werden. Wurde eine Auswahl getroffen, wird diese im Display angezeigt. Alle mit einem + versehenen Nodes können noch weiter aufgeklickt werden. Klickt man mit der rechten Maustaste einen Node an, bekommt man ua. Die Möglichkeit sich Statistiken oder Details anzeigen zu lassen. Durch doppelklick auf den Hintergrund kann jederzeit eine neue Map erzeugt werden.

Für weitere Informationen zur Benutzung der Anwendung, siehe Benutzerdokumentation (gagvis.sourceforge.net/doku.html).

2.6 Programmiersprachen

Die gesamte Anwendung wurde mit jdk1.6.0-02 entwickelt. Auf der Datenbankseite wurde eine HSQLDB-Datenbank verwendet.

2.7 Hardwareanforderungen

Für das Ausführen der Anwendung werden folgende Hardwareanforderungen vorausgesetzt:

- Mindestens Pentium 4 mit 1,2 GHz.
- Mindestens 512 MB RAM, empfohlen 1024 MB oder höher.
- 100 MB freier Festplattenspeicher.
- Eine Bildschirmauflösung von mindestens 1024 x 768 Pixel.

Für das Ausführen des Crawler, bzw. die Entwicklung werden deutlich höhere Systemressourcen empfohlen.

- Mindestens Pentium 4 mit 2 GHz.
- Mindestens 1024 MB RAM oder höher
- Bis zu 200 MB freier Festplattenspeicher.

3. DATENBANK

Als Datenbank wurde eine Standalone HSQLDB-Datenbank (www.hsqldb.org) eingesetzt.

3.1 DATENBANKSTRUKTUR

Die Datenbankstruktur ist folgendermaßen aufgebaut: Es gibt die zentrale Tabelle **‚Firma'**, in der die grundlegenden Daten zu den Firmen, wie Name, Ort, Kontakt, URL usw. gespeichert werden. Da spezifische Daten wie Aufsichtsräte, Vorstände etc. mehrfach zu jeder Firma vorliegen, wurden für diese Daten 7 weitere Tabellen angelegt, in welchen diese Daten abgelegt und über die jeweilige Firmen-ID referenziert werden. Folgende Tabellen wurden mit den gecrawlten Daten der Hoppenstett Datenbank gefüllt:

- Aufsichtsrat
- Firma
- Vorstand
- Beschaeftigte
- Umsatz
- Konzernumsatz
- Beteiligungen
- Banken

Die Politikerdaten von Spiegel-online wurden in weiteren Tabellen abgelegt, auch hier existiert eine zentrale Tabelle **‚Politiker'**, und eine weitere Tabelle Unternehmensfunktion, da jedem Politiker mehrere Unternehmensfunktionen zugeordnet werden konnten. Somit ergeben sich die zwei Tabellen:

- Politiker
- Unternehmensfunktion

Eine weitere Tabelle namens **‚Firmtranslate'** ist dafür zuständig die Firmendaten der Spiegel-online Seite eindeutig einer Firma der Hoppenstedt Daten zuzuordnen. Dies war leider notwendig, da die Firmennamen nicht immer übereinstimmen, obwohl die gleiche Firma gemeint ist (Bsp.: Alte Leipziger-Hallesche Lebensversicherung A.G. <--> ALTE LEIPZIGER Holding Aktiengesellschaft). Dies ist auch die einzige Tabelle, welche händisch gepflegt werden muss. Zu guter letzt wurde noch eine View namens fidzuordnung angelegt um die Performance beim Crawlen etwas zu verbessern.

Damit ergeben sich die beiden letzten Tabellen / View

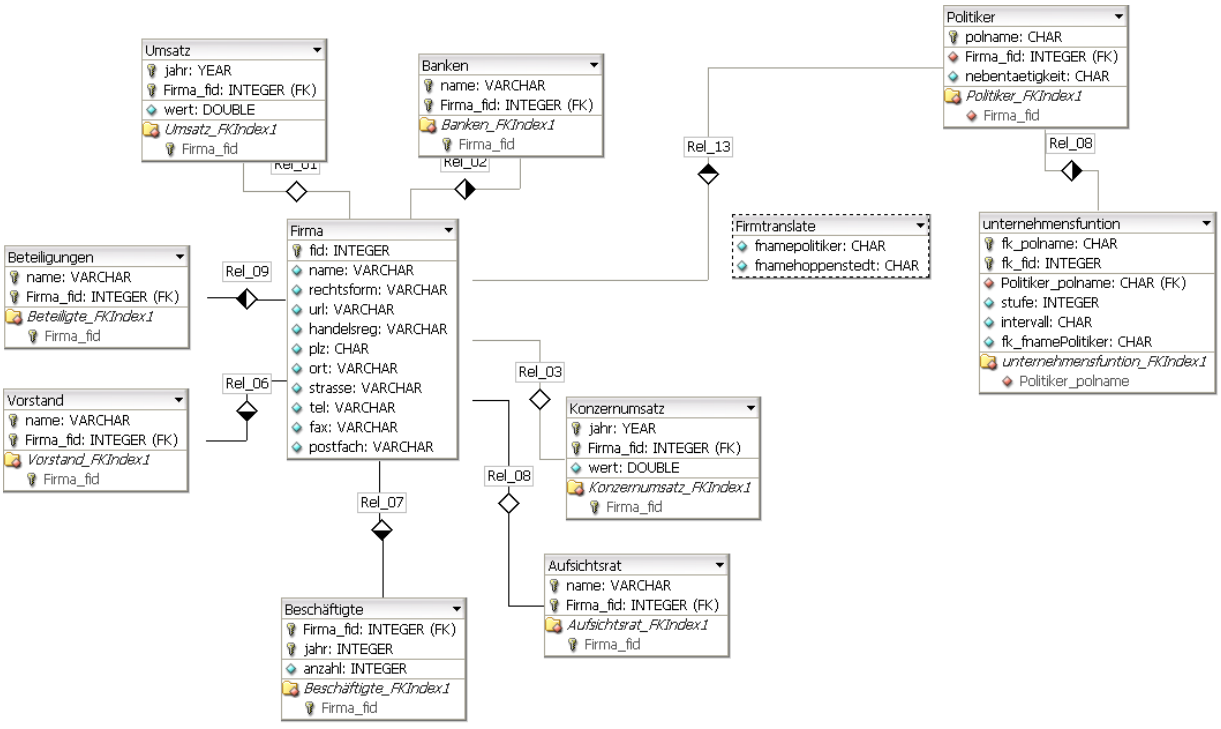
- Firmtranslate
- Fidzuordnung

Für die Erstellung der Tabellen, der View und der Inhalte von Firmtranslate haben wir die drei SQL Scripte

- Creates.sql
- Firmtranslate.sql
- firmtanslate.sql

Diese sollten auch in der Reihenfolge ausgeführt werden, um Probleme zu vermeiden.

3.2 DB-Design



3.3 Datenbank-Creates

3.3.1 Tabelle Firma

```
CREATE TABLE Firma(
 fid INT NOT NULL,
 name VARCHAR(80) NOT NULL,
 rechtsform VARCHAR(20),
 url VARCHAR(100),
 handelsreg VARCHAR(80),
 plz CHAR(5),
 ort VARCHAR(60),
 strasse VARCHAR(80),
 tel VARCHAR(25),
 fax VARCHAR(25),
 postfach INT,
 PRIMARY KEY (fid)
);
```

3.3.2 Tabelle Banken

```
CREATE TABLE Banken (
 banname VARCHAR(60) NOT NULL,
 fk_fid INT NOT NULL,
 PRIMARY KEY (banname, fk_fid),
 FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
 ON UPDATE CASCADE
 ON DELETE CASCADE
);
```

3.3.3 Tabelle Aufsichtrat

```
CREATE TABLE Aufsichtsrat (
 aufname VARCHAR(80)  NOT NULL,
 fk_fid INT NOT NULL,
 PRIMARY KEY (aufname, fk_fid),
 FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
 ON UPDATE CASCADE
 ON DELETE CASCADE
);
```

3.3.4 Tabelle Vorstand

```
CREATE TABLE Vorstand (
  vorname VARCHAR(80) NOT NULL,
  fk_fid INT NOT NULL,
  PRIMARY KEY (vorname, fk_fid),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

3.3.5 Tabelle Beteiligungen

```
CREATE TABLE Beteiligungen (
  betname VARCHAR(200) NOT NULL,
  sitz VARCHAR(80) ,
  anteil FLOAT ,
  fk_fid INT NOT NULL,
  PRIMARY KEY (betname, fk_fid, sitz),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

3.3.6 Tabelle Umsatz

```
CREATE TABLE Umsatz (
  jahr int NOT NULL,
  wert DOUBLE NOT NULL,
  fk_fid INT NOT NULL,
  PRIMARY KEY (jahr, fk_fid),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

3.3.7 Tabelle Konzernumsatz

```
CREATE TABLE Konzernumsatz (
  jahr int NOT NULL,
  wert DOUBLE NOT NULL,
  fk_fid INT NOT NULL,
  PRIMARY KEY (jahr, fk_fid),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
```

```
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

3.3.8 Tabelle Beschaeftigte

```
CREATE TABLE Beschaeftigte (
  jahr INT NOT NULL,
  anzahl INT NOT NULL,
  fk_fid INT NOT NULL,
  PRIMARY KEY (jahr, fk_fid),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

3.3.9 Tabelle Politiker

```
CREATE TABLE Politiker (
  polname VARCHAR(80) NOT NULL,
  nebentaetigkeit VARCHAR(3000),
  PRIMARY KEY (polname)
  );
```

3.3.10 Tabelle Firmtranslate

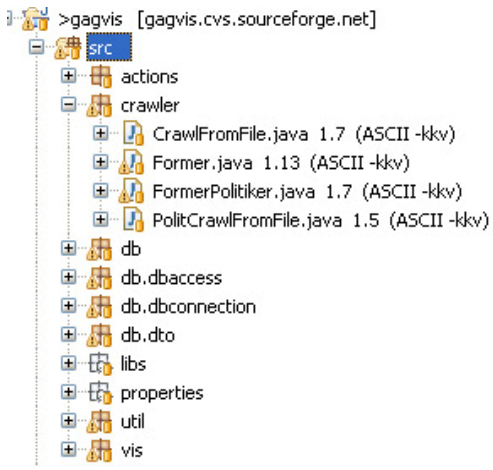
```
CREATE TABLE firmtranslate(
 fNamePolitiker VARCHAR(150),
 fNameHoppenstedt VARCHAR(150),
 PRIMARY KEY (fNamePolitiker)
);
```

3.3.11 Tabelle Unternehmensfunktion

```
CREATE TABLE Unternehmensfunktion (
  fk_polname VARCHAR(80) NOT NULL,
  fk_fid INT NOT NULL,
  stufe INT,
  intervall VARCHAR(50),
  fk_fNamePolitiker VARCHAR(150),
  PRIMARY KEY (fk_fid, fk_polname,fk_fNamePolitiker),
  FOREIGN KEY (fk_polname) REFERENCES Politiker (polname),
  FOREIGN KEY (fk_fid) REFERENCES Firma (fid)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

4. PACKAGES – KLASSENBESCHREIBUNG

4.1 CRAWLER



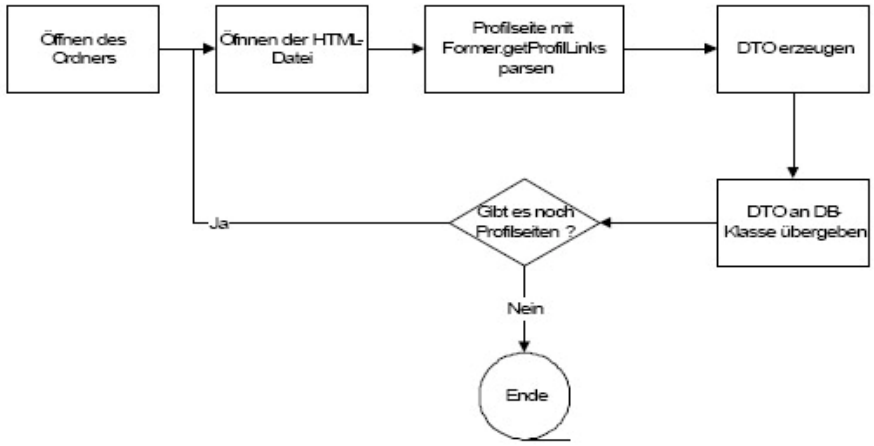
Das Package Crawler enthält folgende Klassen:

- CrawlFromFile.java
- Former.java
- FormerPolitiker.java
- PolitCrawlFromFile.java

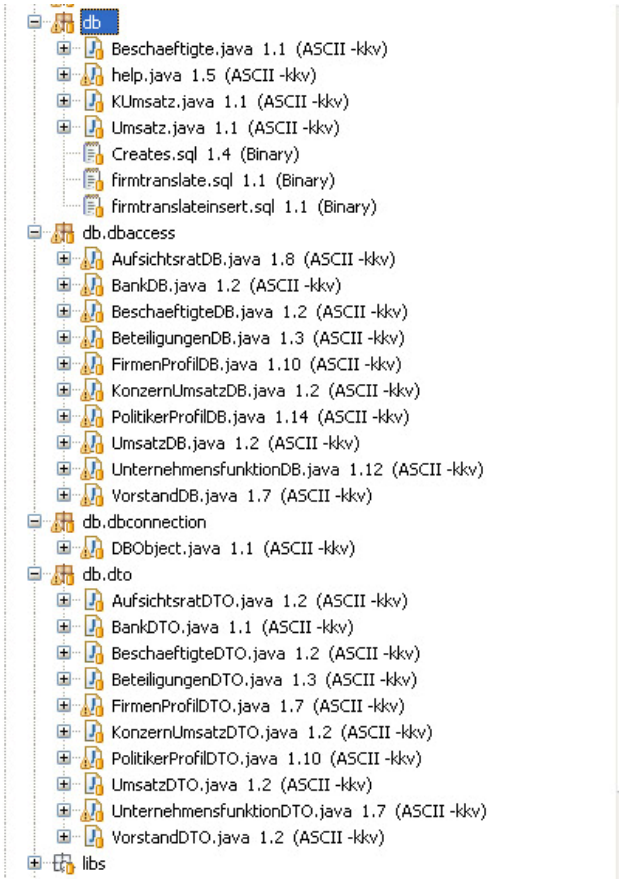
Die Klasse CrawlFromFile.java erlaubt es alle, unter Data vorhandenen Hoppenstedt Firmenprofile, zu Crawlen und zu Parsen. Die Klasse parst den Inhalt und erzeugt DTO-Objekte, die dann der DB-Klasse übergeben werden. Die Dateinamen und -endungen sind dabei egal. Es werden alle im Ordner befindlichen Dateien der Reihe nach abgearbeitet. Dabei enthält die Former.java alle Parser-Methoden. Gleiches gilt für die Klasse PolitCrawlFromFile.java und FormerPolitiker.java allerdings werden hierzu die Profile der Politiker herangezogen. Diese befinden sich unter data/politiker. Die Politikerdaten mussten auch lokal abgespeichert werden, da wir nicht sicher gehen konnten, wie lange diese online auf Spiegel-online zur verfügung gestellt werden. Details zur

Programmierung entnehmen Sie der JavaDoc, bzw. Den Kommentaren im Quellcode.

Grundlegner Ablauf:



4.2 DB



Das Package db unterteilt sich nochmals in db.dbaccess, db.dbconnection und db.dto

4.2.1 DB

Unter db ist lediglich die Klassen help.java von bedeutung. Diese enthält zwei Methoden um die Datenbanken zu leeren, wenn ein neuer Crawl gestartet wird.

4.2.2 DB.DBACCESS

Zu jeder DTO-Klasse gibt es auch eine DB-Klasse. Diese DB-Klasse stellt Methoden zur Verfügung, die die für die Anwendung wichtigen Datenbank-Aktionen auf die dazugehörende Tabelle erledigen (z.B. finde alle, finde nach ID, finde nach Name, füge ein…) Die Abfrage-Methoden liefern je nach Logik entweder ein DTO zurück oder eine Collection (java.util.LinkedList) mit mehreren DTOs zurück. Die Methode zum einfügen hat ein DTO als Parameter. Zusätzlich verfügen die DB-Klassen über Abfrage-Methoden, die ein Table-Objekt zurückliefert, welches direkt von Prefuse verarbeitet werden kann.

4.2.3 DB. DBCONNECTION

Dieses Package enthält nur die Klasse DBObject.java, die über eine von ihr erzeugten Instanz eine Datenbankverbindung aufbaut, und diese über die Objektmethode getConnection() zurückgibt. Sie enthält 2 zentrale Methoden: getConnection() -> liefert SQL-Connection (java.sql.Connection) zurück getDataSource()-> liefert Datasource auf die DB, mit welcher Prefuse direkt auf der DB arbeiten kann.

4.2.4 DB.DTO

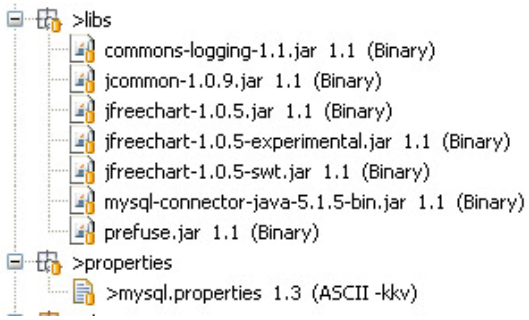
In diesem Package befinden sich alle Database-Transfer-Objects. Diese Klassen repräsentieren jeweils eine Tabelle der Datenbankstruktur, d.h. jede Klasse hat die Attribute, der zugehörigen Tabelle. Dazu enthält sie noch 3 Konstruktoren:

- Standard-Konstruktor (erzeugt leeres Objekt)
- Konstruktor mit ResultSet als Parameter um aus einer selektierten Tabellenzeile ein Objekt zu erzeugen).
- Konstruktor mit Einzelparameter zum „manuellen“ erzeugen und initialisieren eines Objektes.

Ansonsten verfügen diese Klassen noch über die nach den Java-Bean-Konventionen üblichen getter- und setter-Methoden.

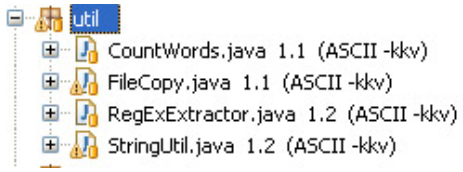
Auch hier gilt, details zur Programmierung entnehmen Sie der JavaDoc, bzw. den Kommentaren im Quellcode.

4.3 Libs und Properties



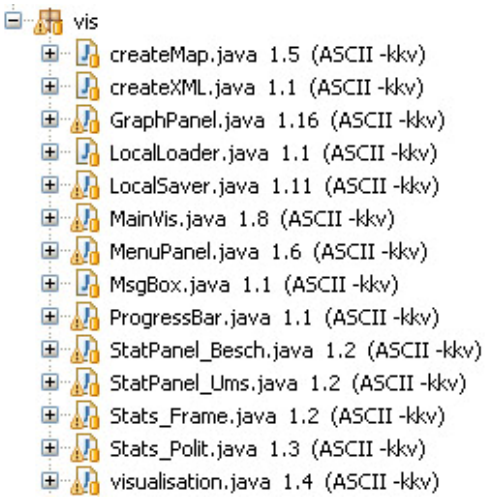
Zwar nicht weiter Erwähnenswert aber der Vollständigkeitshalber hier aufgeführt sind die beiden Packages libs und properties. Unter Libs finden Sie alle benötigten jar Dateien, die zur Entwicklung unter dem Java Build Path eingebunden werden müssen. In properties findet man die mysql.properties Datei, in der die Datenbank-verbindungsdaten hinterlegt sind.

4.4 Util



Auch im Util Package befinden sich lediglich 4 Hilfsklassen, CountWords.java erlaubt es in einem sehr langen String die Häufigkeit bestimmter Wörter zu zählen. (Wird für die Statistiken der Politiker benötigt) FileCopy.java enthält wie der Name schon sagt, eine Methode um eine Datei an einen bestimmten Platz zu kopieren. Die letzten beiden Klassen RegExExtractor.java und StringUtil.java werden benötigt um in den Regulären ausdrücken, welche vor allem in den Formerklassen beim crawlen vorkommen, diverse modifikationen vorzunehmen. Z.B. Das ersetzen von html Tags usw.

4.5 Vis



Beim Package Vis um das Herzstück der ganzen Anwendung. Dieses Package enthält alle Klassen des User Interface, der Statistiken, sowie alle Komponenten zur Erzeugung der xml Dateien, die von Prefuse benötigt werden um den Graphen zu erzeugt werden.

Im folgenden sollen nur die wichtigsten Klassen vorgestellt werden.

4.5.1 MainVis.java

Hierbei handelt es sich praktisch um die Hauptklasse des Programms. MainVis.java ist ein Jprefuse Applet, welches in einm Browser geladen werden kann. Beim starten der Klasse passieren 3 Grundlegende Dinge.

Zum einen wird im temporären Verzeichnis des Benutzers eine leere XML Datei angelegt, in die später dann die Grapheninformationen für Prefuse abgelegt werden. Zum anderen werden die benötigten Icons ebenfalls ins Temp-Verzeichnis des Benutzers geladen. Dies ist notwendig, da in der XML File die entsprechenden Icons für Vorstand, Aufsichtsrat, Firma und Politiker gleich mit angegeben werden müssen und beim aufklicken der Map jedesmal die XML Datei neu geschrieben wird.

Zu guter letzt wird die Infozeile sowie das Display in das Applet eingebunden. Das Display selbst wird durch eine Instanz der Klasse GraphPanel erzeugt (siehe hierzu Graphpanel.java)

4.5.2 GraphPanel.java

Dies ist die wohl größte Klasse des ganzen Projekts. Hier findet nicht nur die Visualisierung statt, sondern auch die komplette Steuerung ist in dieser Klasse implementiert. Ursprünglich war es geplant die Steuerung über ein extra Panel in der MainVis einzubinden. Nach ersten Versuchen hat sich allerdings recht schnell herausgestellt, das ein weiteres Panel mit Steuerelementen zuviel Platz wegnimmt. Um die Übersichtlichkeit weiterhin zu gewährleisten haben wir uns dazu entschieden die komplette Steuerung über Kontextmenüs zu implementieren. Was auch der grund ist wieso in der GraphPanel.java 2 ActionListener enthalten sind. Eigentlich sollten diese in extra Klassen stehen, allerdings kam es dann zu diversen Problemen, weshalb wir sie nun direkt in der GraphPanel.java eingebaut haben. Details sind wie immer dem Quellcode sowie der JavaDoc zu entnehmen

4.5.3 LocalSaver.java

Wie weiter oben schon erwähnt wird mit jedem klick auf einen bestimmten Node im Graph oder beim starten einer neuen Suche die xml Datei neu erstellt. Das erstellen dieser Datei wird von der Klasse LocalSaver.java durchgeführt. Am wichtigsten ist hier die Funktion public void saveMap(String name,int id, String type). Diese bekommt als Übergabeparameter den Namen, die Firmenid und den Typ des gesuchten bzw. angeklickten Objektes. Darauf hin werden alle mit diesem Objekt verbundenen Nodes in der XML Datei ergänzt.

4.5.4 Stats_.java und StatPanel_.java

Bei den vier Klassen handelt es sich um die Statistiken und die Details die zu Firmen bzw. Politikern im Graph angezeigt werden können. Diese zusatzinformationen werden jeweils als InternalFrames dargestellt. InternalFrames sind praktisch Frames, die innerhalb des Frames bleimen, indem Sie geladen wurden.

5. Syleguide

5.1 Schriftart

Als Schriftart wurde im Programm ausschliesslich “Tahoma” verwendet. Schriftgrösse in den TitledBoarders ist 10. In der Detailansich der einzelnen Nodes 12.

5.2 Icons

Es wurde zwei Iconsätze für die Anwendung entworfen.

5.2.1 Iconsatz 1

Aufsichtsrat

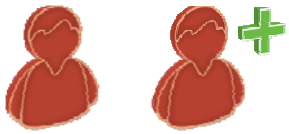
Politiker

Vorstand

Firma

5.2.3 ICONSATZ 2

Aufsichtsrat



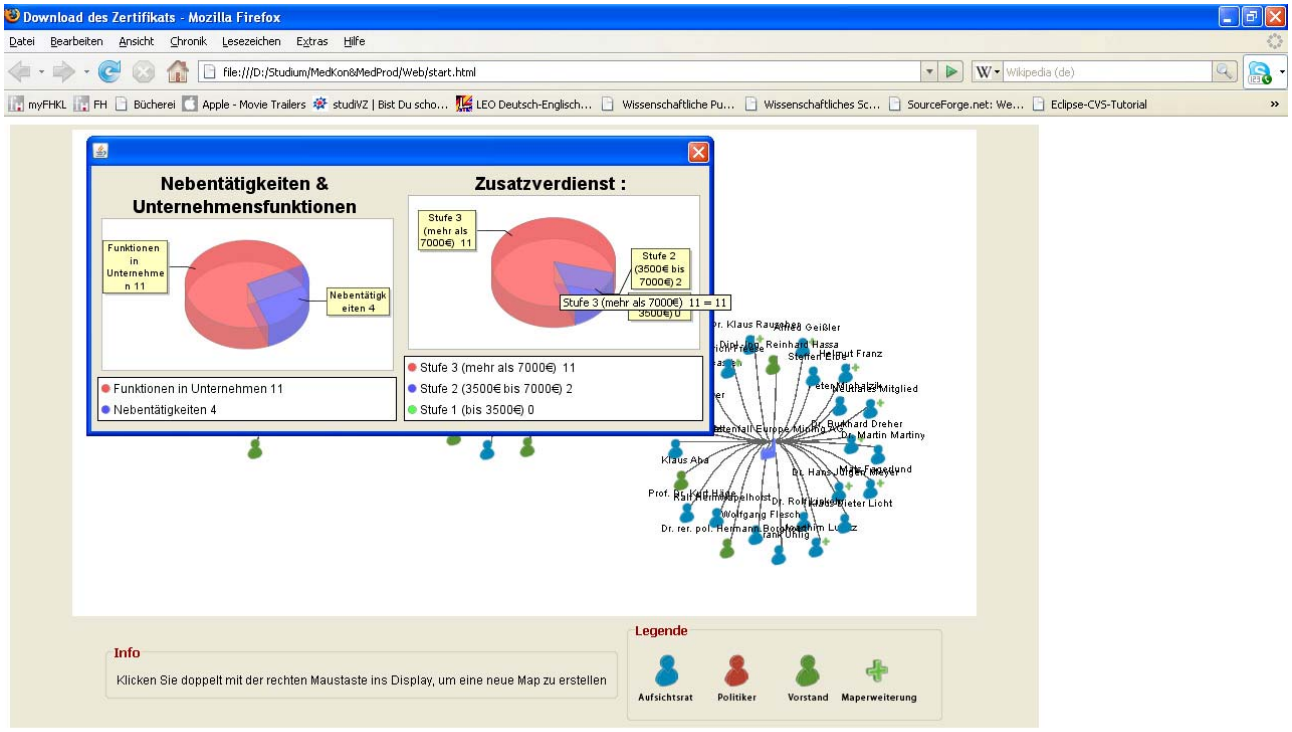
Politiker

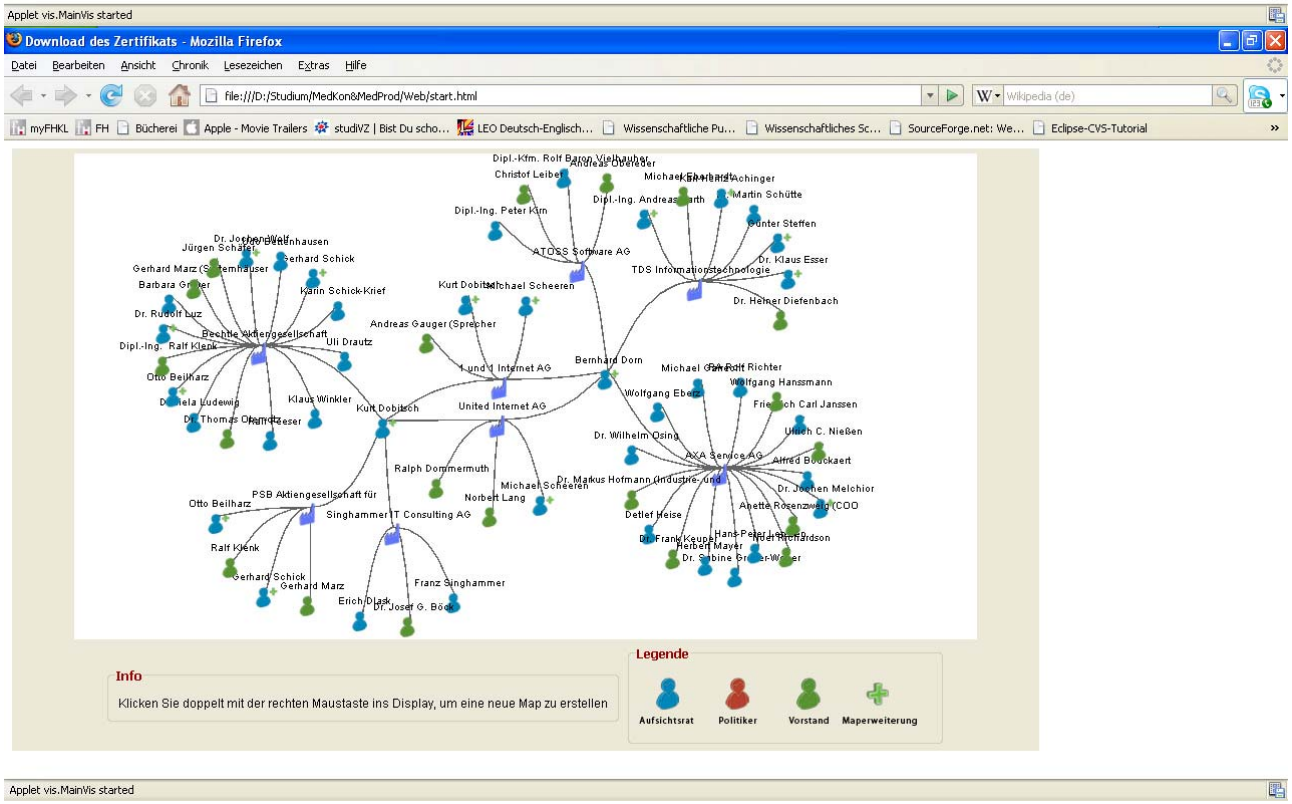
Vorstand

Firma

Letztendlich haben wir uns für den Iconsatz 2 entschieden. Besteht ein Graph aus vielen Nodes lassen sich diese durch die einfacheren Icons doch deutlich besser unterscheiden. Da diese sich nicht durch Form sondern Farbe unterscheiden.

5.3 SCREENSHOTS





6. KONTAKT

6.1 DIE ENTWICKLER

Die Anwendung GagVis wurde Entworfen und Entwickelt von:

- Felix Kiefer, felix_kiefer@users.sourceforge.net
- Daniel Seiler, daniel-seiler@users.sourceforge.net

6.2 BETREUUNG

Prof. Hendrik Speck, www.hendrikspeck.com

Medienkonzeption und Produktion im Wintersemester 2007/2008

Fachhochschule Kaiserslautern Standort Zweibrücken