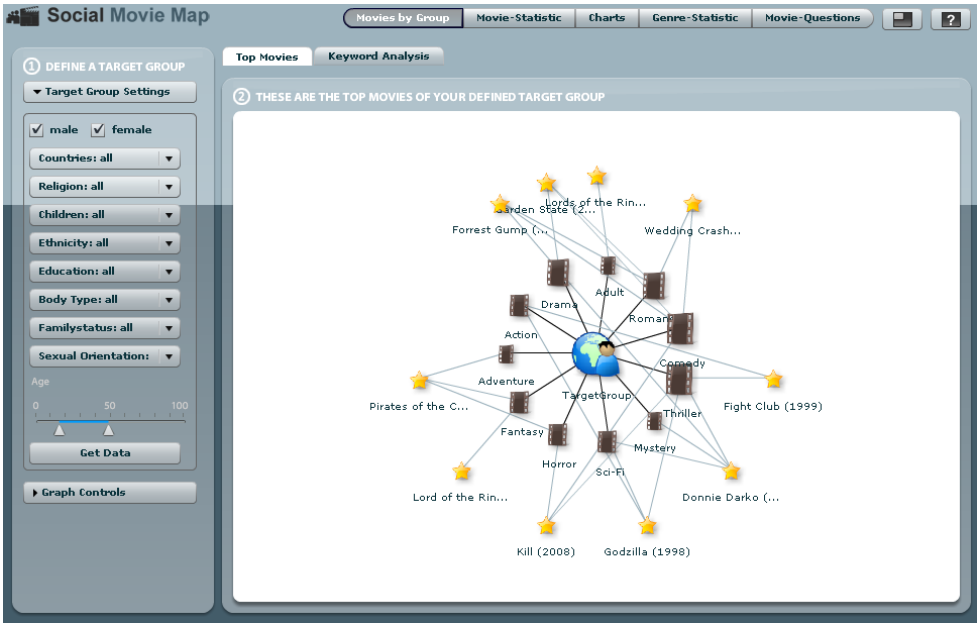
Social Movie Map

- Technische Dokumentation -



an Open Source web application

by Philipp Bender & Florian Wilhelm

Inhaltsverzeichnis

Einleitung

Filme gehören zu den wichtigsten Medien in unserer heutigen Zeit und sind eine der beliebtesten, interkulturellen Quellen für Unterhaltung. Die visuellen Elemente eines Kino-Filmes haben eine einzigartige Art der Kommunikation.

Ein Film ist zweifellos ein starkes Medium und wird von allen Menschen auf der Welt gleich betrachtet. Warum also nicht dieses leistungsstarke Medium nutzen, um eine Analyse über die Lieblingsfilme durchzuführen.

Es ist kein Problem mit anderen Menschen über Filme zu sprechen, sondern eher das Problem zu Lieblingsfilme zu finden. Unsere Anwendung kann Ihnen dabei helfen, Filme zu finden und analysieren.

Auf der anderen Seite können Sie sehen, welche Filme von anderen Personen bevorzugt werden. Haben Sie Interesse was kanadischen weibliche Jugendliche als Lieblingsfilm haben? Oder was sind besonders beeindruckende Filme für Menschen zwischen 40 und 50 Jahren in Großbritannien? Entdecken Sie eine neue Art der Analyse von Lieblingsfilmen. Vielleicht können Sie sich damit identifizieren.

Zusammenfassung

Ziel des Projektes “Social Movie Map” ist die Visualisierung von gesammelten Lieblingsfilmen von MySpace Benutzern. Die Darstellung kann durch Auswahlkriterien wie Alter, Geschlecht, Herkunft oder Film- Genre entsprechend angepasst werden. Alle Daten basieren auf den Angaben der öffentlichen MySpace Profilen. Für die Darstellung und Visualisierung ist eine Webbasierende Anwendung geplant, auf der ein Benutzer seine benutzerspezifischen Analysen durchführen kann. Des Weiteren kann nach einem gewünschten Filmtitel gesucht werden, für den die zugehörigen Statistiken angezeigt werden.

Funktionsprinzip

Ein Benutzer kann zwischen fünf Arten von Visualisierungen, bei Social Movie Map, auswählen: Movie by Group, Movie-Statistic, Genre-Statistic und Movie Questions. Jede Kategorie repräsentiert andere Statistiken und hat eigene Diagramme und Visualisierungen. Für die Darstellung der Statistiken werden XML Dateien verwendet. Diese Daten werden aus einer Datenbank gelesen und in entsprechender XML Struktur an die Social Movie Map Anwendung übergeben. Zusätzlich werden die Ergebnisse über eine Cache Funktion zwischengespeichert. Dadurch wird das Datenbanksystem entlastet und der Endnutzer hat eine schnellere Anwendung.

Datenbeschaffung

Um die benötigten Daten zu sammeln wird ein MySpace Crawler eingesetzt. Insgesamt wurden ca. 6 Millionen öffentliche MySpace Profile erfasst.

Technologie und Architektur

Crawler

Der Crawler ist eine Java Server / Client Anwendung. Um den erhaltenen HTML Inhalt (MySpace Öffentliche Profile) zu parsen wird der Jericho HTML Parser, eine Java Bibliothek zum Analysieren und Manipulieren von HTML Dokumenten, in der Version 2.6 verwendet. Entwickelt wurde der Crawler unter der Entwicklungsumgebung Eclipse 3.4. Als Datenbanksystem wird MySQL 5.0.67 eingesetzt. Für die Abfragen wird Standard SQL: 2003 eingesetzt.

Database Synchronization

Um die gesammelten Daten aufzubereiten wurde eine Java Applikation mit Java 1.5 entwickelt. Um die, von den MySpace Personen, angegebene Lieblingsfilme an die Film Datenbank anzupassen wurden die Filme mit dem Levenshtein Algorithmus verglichen. Als Datenbank wird ebenfalls MySQL 5.0.67 verwendet. Für die Abfragen wird Standard SQL: 2003 eingesetzt.

Durch die Entwicklung mit Java ist der Datenabgleich an kein Betriebssystem gebunden.

Social Movie Map

Social Movie Map ist eine mit Adobe Flex 3.0.2 entwickelte Flash Anwendung. Die Anwendung läuft auf einer Ubuntu Linux Distribution. Als Webserver wird ein Apache Server Version 2.2.11 mit PHP 5.2.8 verwendet. Für die Datenhaltung wird das frei erhältliche Datenbankverwaltungssystem MySQL 5.0.67 verwendet.

Als Schnittstelle zwischen PHP und Mysql wird MySQLi mit Standard SQL: 2003 eingesetzt. An Application created with Flex can run in the browser using Adobe Flash Player software.

Adobe Flex

Flex is a free, open source framework for building highly interactive, expressive web applications that deploy consistently on all major browsers, desktops, and operating systems. It provides a modern, standards-based language and programming model that supports common design patterns. MXML, a declarative XML-based language, is used to describe UI layout and behaviors, and ActionScript 3, a powerful object-oriented programming language, is used to create client logic. Flex also includes a rich component library with more than 100 proven, extensible UI components for creating rich Internet applications (RIAs), as well as an interactive Flex application debugger.

RIAs created with Flex can run in the browser using Adobe Flash Player software or on the desktop on Adobe AIR, the cross-operating system runtime. This enables Flex applications to run consistently across all major browsers and on the desktop. And using AIR, Flex applications can now access local data and system resources on the desktop.

MySpace

MySpace ist eine mehrsprachige Plattform, die sich über Werbung finanziert und den Nutzern ermöglicht, kostenlose Benutzerprofile mit Fotos, Musik, Videos, Blogs, Gruppen usw. einzurichten. MySpace wird als der bekannteste Vertreter eines als Website realisierten Sozialen Netzwerks (Web 2.0) angesehen.

Datenbankstruktur

In diesem Kapitel findet die Analyse über die Inhalte von den MySpace Profilen statt sowie die Abbildung der Daten in einem relationalen Datenbankmodell.

Analyse der MySpace Profile

Als erstes werden alle relevanten Angaben, die eine Person auf einem MySpace Profil angeben kann, betrachtet. Des Weiteren gibt es zu den MySpace Profilen noch MySpace Bands. Diese Unterscheiden sich von den möglichen Angaben, haben jedoch Gemeinsamkeiten wie id, profilename, Freunde, Kommentare etc.

Dabei gibt es Attribute die durch ein Selections-Feld vorgegeben sind. Diese sind deutlich besser einzulesen und zu behandeln, als freie Eingabefelder wie AboutMe oder Lieblingsfilme. Die Lieblings-Bücher, -Music, -Movies, -Television und –Helden werden bei jedem, ; / \ sowie HTML-Zeilenumbrüche wie
 und
 getrennt. Dies ist notwendig da die Angaben als ein Textstring vorhanden sind und jeder Benutzer eigene Trennzeichen verwendet. Andere Trennzeichen wie & + - oder „and“ konnten nicht verwendet werden, da diese z.T. Teil eines Filmtitels / Buchttitels waren.

Daten wie Vorname, Nachname, E-Mail oder Passwort sind geschützte Inhalte und werden auf den Profilen nicht angezeigt und konnten somit nicht eingelesen werden.

Alle relevanten Attribute finden Sie inklusive des Field Type auf der nächsten Seite.

Category	Attribute	Field Type
About Me	About Me	Textarea
	Who I'd Like to Meet	Textarea
Interests	General	Textarea
	Music	Textarea
	Movies	Textarea
	Television	Textarea
	Books	Textarea
	Heroes	Textarea
Basic Info	Gender	Male / female
	Display Name	Input
	Headline	Input
	City	Input
	Zip-Code	Input
	Country	Selection
	Region	Selection
	Photo	File-Upload
Details	I am her for	Checkbox
	Hometown	Input
	Height	Input
	Occupation	Input
	Sexual Orientation	Selection
	Bodytype	Selection
	Ehtnicity	Selection
	Religion	Selection
	Smoker	Selection
	Drinker	Selection
	Children	Selection
	Education	Selection
	Family Status	Selection
	Income	Selection
Schools	Country	Selection
	State	Selection
	City	Selection
	Your School	Selection
MySpace Bands	Page Views	Text
	Member since	Text
	Label	Text
	Kind Label	Text
	Website	Text
	Band Member	Text
	Influence	Text
	Sound like	Text
	Friends	Text
	Comments	Text

Tabelle 1: MySpace Profile-Attribute

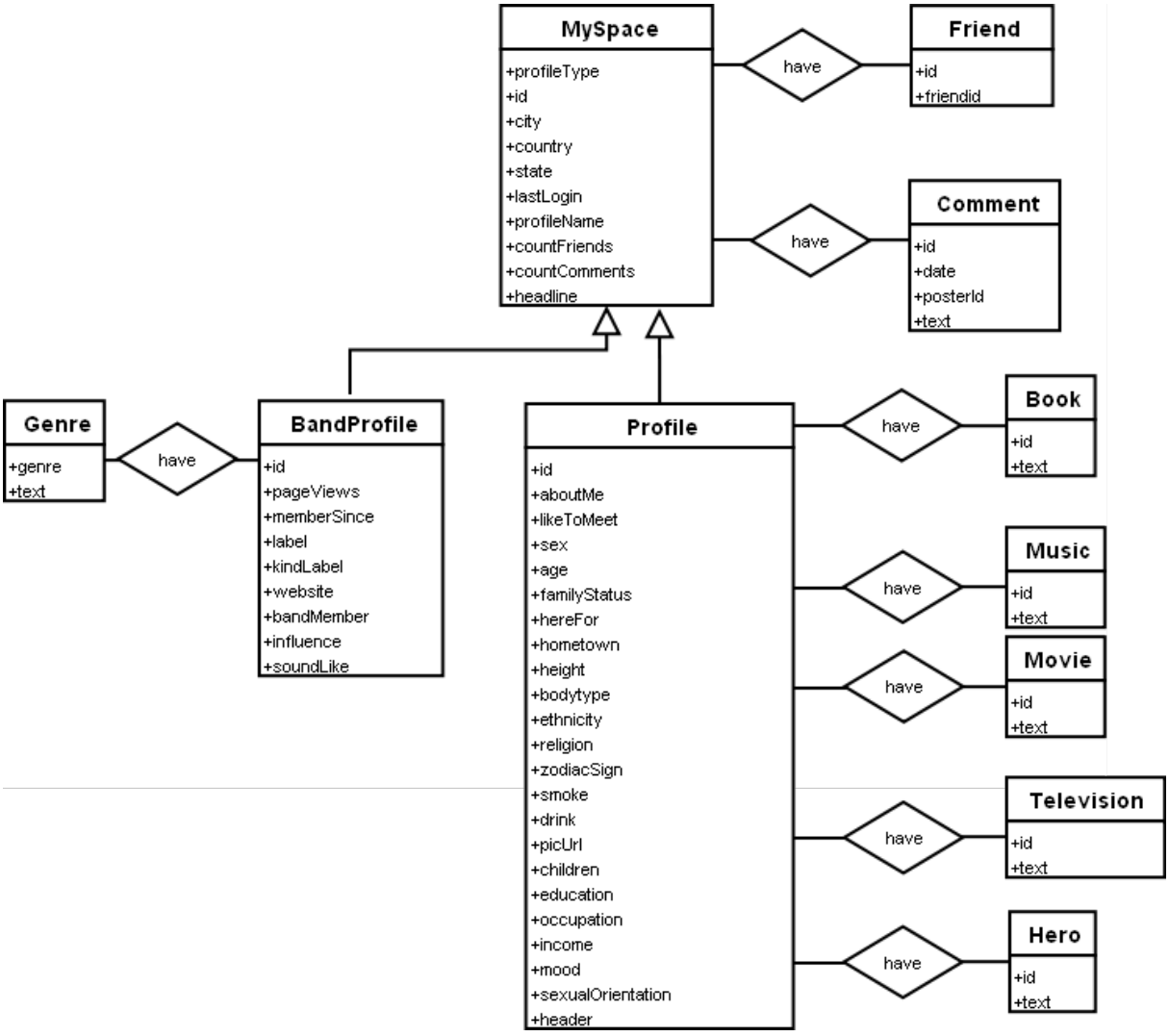


Abbildung 1: Entity Relation Modell - Crawler

In der Tabelle MYSPACE werden alle von den Bands und Profilen gemeinsamen Attributen gespeichert. Hier zu zählen die Attribute profileType (profile / band), id, city, country, state, lastLogin, profileName, countFriends, countComments und headline. Jedes MySpace Profil hat N Anzahl an Kommentaren und Freunden. Dies entspricht ebenfalls für Bands sowie den Profilen. Die Attribute von einem Profil werden in der PROFILE Tabelle abgespeichert. Eine Person kann zusätzlich noch mehrer Lieblings- Bücher, -Music, -Movies, -Television oder -Helden haben, deshalb werden diese in extra Tabellen verwaltet.

Die Attribute für eine Band werden in die BandProfile hinterlegt. Jede Band wird bestimmte Musik Genre zugewiesen. Diese Genre werden in GENRE Tabelle gespeichert.

Der Primärschlüssel MySpace.id entspricht der MySpace Profil Nr.

Durch diese Datenstruktur ist das Soziale Netz inklusiven alle öffentlichen Angaben, einer Person, komplett in der Datenbank hinterlegt und kann nachgebildet werden.

Analyse der IMDB DB

IMDb Lizenz

Limited non-commercial use of IMDb data is allowed, provided the following conditions are met:

You agree to all the terms of our copyright/conditions of use statement, located at http://www.imdb.com/conditions.
Please also note that IMDb reserves the right to withdraw permission to use the data at any time in its discretion.

The data must be taken only from the plain text data made available from our FTP sites (see http://www.imdb.com/interfaces for more details and for links to our FTP servers). You may not use data mining, robots, screen scraping, or similar online data gathering and extraction tools on our website. If the information/data you want is not present in the data files available from our FTP sites, it means it's not available for non-commercial usage.
If you do want to use IMDb data for commercial purposes, you must contact our Content Licensing Department at http://www.imdb.com/Licensing/.

The data can only be used for **personal and non-commercial** use and must not be altered/republished/resold/repurposed to create any kind of online/offline database of movie information (except for **individual personal** use). Please refer to the copyright/license information enclosed in each file for further instructions and limitations on allowed usage.

You must acknowledge the source of the data by including the following statement: Information courtesy of
The Internet Movie Database (http://www.imdb.com).
Used with permission.

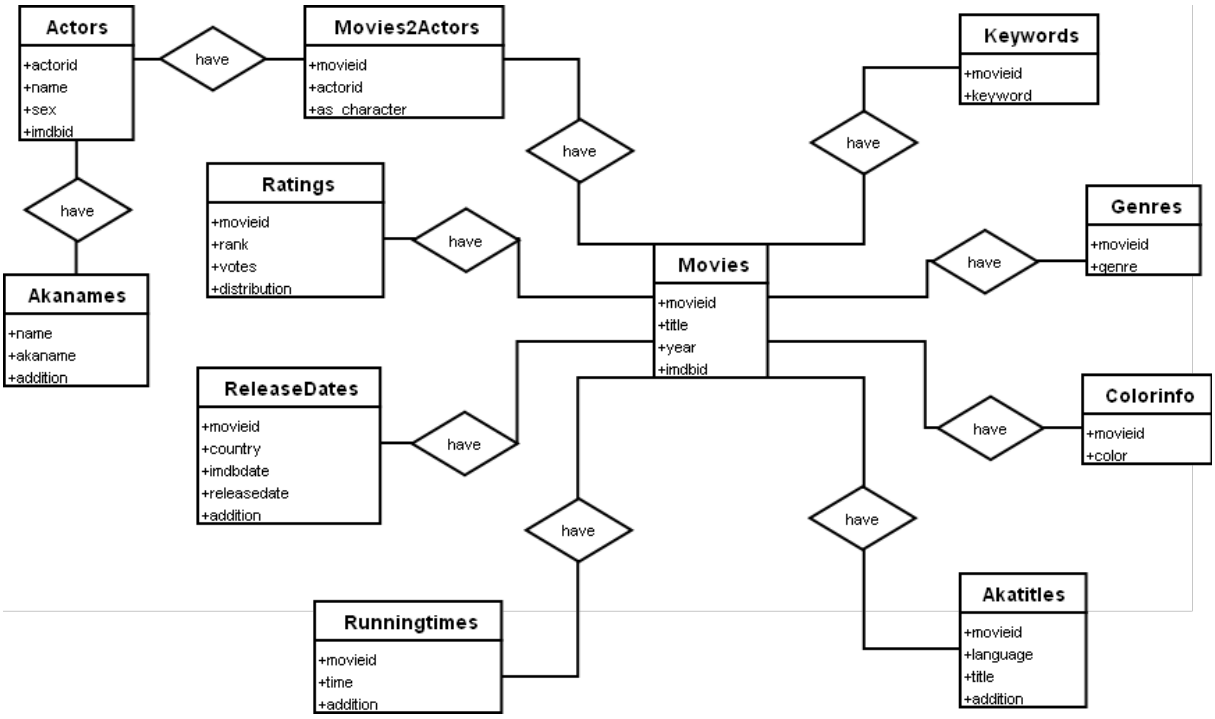


Abbildung 2: alle benötigten IMDb Tabellen für Social Movie Map

Die Filme und TV-Serien werden in der MOVIES Tabelle gespeichert. Der Primärschlüssel der MOVIES ist die movieid. Des Weiteren wird der Original- Filmtitel und das Erscheinungsjahr mit abgespeichert. Die anders sprachigen Filmtitel werden in der AKATITLES hinterlegt.

Jeder Film kann zu bestimmten Keywords zugewiesen werden, sowie Genre. Des Weiteren hat ein Film weitere Informationen: Colorinfo, Runningtimes, Release Dates und Ratings von den IMDB users. Diese Angaben werden in getrennten Tabellen verwaltet. Jedem Film werden über die Movies2Actors die Schauspieler, die im Film mitgewirkt haben, zugewiesen. Der Primärschlüssel MOVIES.movieid ist bei KEYWORDS, GENRES, COLORINFO, AKATITLES, RUNNINGTIMES, RELEASEDATES, RATINGS, MOVIE2ACTORS als Fremdschlüssel deklariert.

Datenaufbereitung

Hier findet die Analyse zu der Datenaufbereitung statt. Es werden alle Tabellen betrachtet, die für den Einsatz in der Social Movie Map Anwendung benötigt werden.

Zur Optimierung der Abfragen wurden die Tabellen möglichst zusammengefügt. Die MOVIES Tabelle wurde mit den Informationen wie Rating, Release Dates, Runningtimes und Colorinfo erweitert. Ebenso wurde die Tabelle MySpace und Profile miteinander verbunden. Alle unnützen Tabellen wurden aus Speichergründen nicht in der Social Movie Map Datenbank aufgenommen (z.B. Bandprofile).

Somit können alle relevanten Abfragen mit wenigen JOINs realisiert werden. Die Datenstruktur konnte dadurch beibehalten werden ohne dabei großartige Redundanzen zu erzeugen.

Da in der IMDB auch viele TV- Serien auftraten und somit den Datenabgleich verfälschten, wurden alle Serien gelöscht. Dies hatte die Folge, dass von den ehemals 1280222 Filmen nur noch 771299 übrig blieben.

Für das Live System werden nur die Profile in der Datenbank gehalten, welche auch Lieblingsfilme haben. Dies hat natürlich Vorteile in der Geschwindigkeit sowie Speicherbedarf.

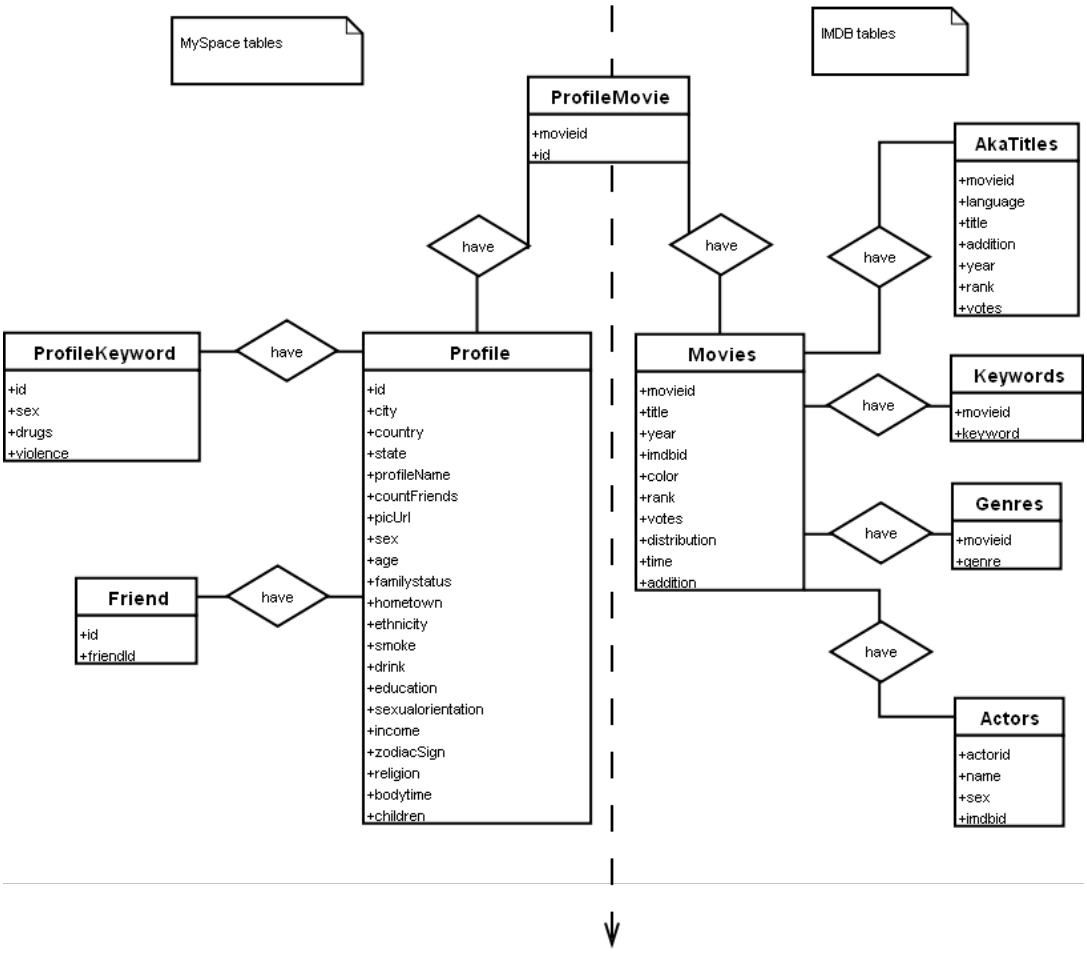


Abbildung 3: Entity Relationship Model

Datenabgleich

Der Filmabgleich wurde mit dem Levenshtein Algorithmus und dem Minimum Prinzip realisiert. Dabei wird die Levenshtein Distanz von einem Lieblingsfilm und Film berechnet. Der Film der die kleinste Distanz vorzeigt, wird in der Datenbank mit der entsprechenden movieid eingetragen.

Wenn eine Distanz jedoch größer als 1 / 3 des kompletten Filmtitels ist, wird dieser nicht eingetragen.

Levenshtein distance

In information theory and computer science, the Levenshtein distance is a metric for measuring the amount of difference between two sequences (i.e., the so called edit distance). The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

Favorite Movie (MySpace Profile)	Film Title (IMDB)	Levenshtein distance
Lord of the Rings	Lord of the Rings	0
The Lord of the Rings	Lord of the Rings	4
The Fellowship of the Ring	Lord of the Rings: The Fellowship of the Ring	19
Pulp Fiction	Pulp Fiction	0
Plup Fiction	Pulp Fiction	2
Plup Fiction	Plump Fiction	1

Tabelle 2: Levenshtein Distanz Beispiele

Durch den Abgleich über die Levenshtein Distanz können unter Umständen falsche Ergebnisse auftauchen. Ein Großteil der Filme kann nicht zugewiesen werden, da die MySpace Benutzer selten den kompletten Filmtiteln angeben. Bei falsch betitelten Lieblingsfilmen kann je nach Distanz ein anderer Film als Ergebnis erscheinen (Bsp.: Tabelle: Pulp Fiction).

Des Weiteren gibt es für manche originalen Filme mehrere Neuverfilmungen und da die MySpace Benutzer keine Jahreszahl angeben, wird von der neusten Verfilmung ausgegangen. Dadurch wird ebenfalls das Ergebnis manipuliert, da viele Benutzer evtl. den Original Titel zitierten.

Viele Benutzer haben durch die freie Texteingabe volle Texte angeben: „I like all Action Films“. Diese konnten natürlich nicht synchronisiert werden. Es wurden nur Lieblingsfilme verglichen bei denen die Textlänge maximal 50 Zeichen betrug, da die Wahrscheinlichkeit bei einer größeren Textlänge immer kleiner wird um das entsprechend Ergebnis zu erreichen.

MySpace Crawler

Der MySpace Crawler arbeitet mit dem OpenSource Jericho Java HTML Parser um die HTML Inhalte auszulesen. Die Benutzer bei MySpace haben eine fortlaufende Nummer (http://profile.....viewprofile&friendid=123456789). Durch einfaches inkrementieren der friendid, konnte der Crawler die verschiedenen Profile aufrufen. Durch dieses sequenzielle Vorgehen mussten man folgende Fälle abfangen:

- Ungültige FriendID
- Privates Profil
- Band Profil
- Öffentliches Profil

Manche friendid's sind nicht bei MySpace vergeben und geben daher die Rückmeldung „Ungültige FriendID". Diese Seiten wurden ohne weitere Bearbeitung übersprungen. Jeder Benutzer kann persönlich angeben ob sein Profil öffentlich zugänglich oder privat ist. Die privaten Profile wurden ebenfalls übersprungen, da bei diesen Profilen nur der Profilname entnommen werden konnte. Ausgelesen wurden nur Band Profile und Öffentliche Profile.

Der Crawler ist eine Server- Client Java Anwendung.

Der Server verwaltet die friendid Bereiche und füttert die Clients (Datensammler) mit einer Liste von freien ID's. Zusätzlich kann ein Server eine unbegrenzte Anzahl an Clients verwalten. Der Server hat zwei Einstellungsmöglichkeiten bevor der Server gestartet wird (Start-ID, Paket-Größe). Diese werden vor dem Start initialisiert und können während des laufenden Betriebes nicht geändert werden.

Start-ID	Ist die FriendID bei der, der Crawler das sammeln der Daten beginnt.
Paket- Größe	Anzahl der FriendID's die ein Client zugewiesen bekommt. Erst wenn der Client diese Liste abgearbeitet hat, bekommt dieser neue FriendID's.

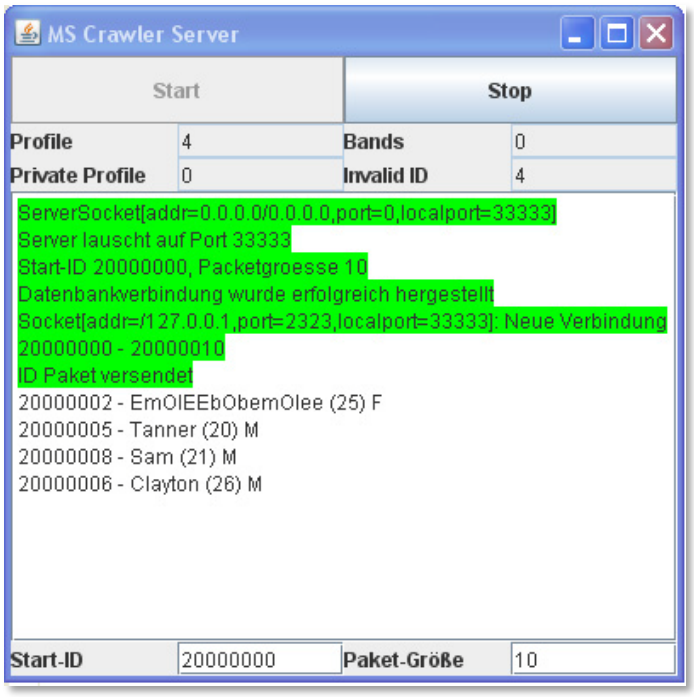


Abbildung 4: MS Crawler Server

Der Client (Datensammler) baut aus der erhaltenen friendid Liste die MySpace Profil URL zusammen und lädt den Quellcode in ein Source Objekt. Dieses Source Objekt kann nun geparst werden und alle entsprechende Attribute können ausgelesen werden. Für die Identifizierung des Profiles (Ungültige FriendID, Privates Profile, Band Profil, Öffentliches Profil) ist die Klasse PageIdentifier.java zuständig. Diese gibt je nach Identifizierung eine Zahl zurück.

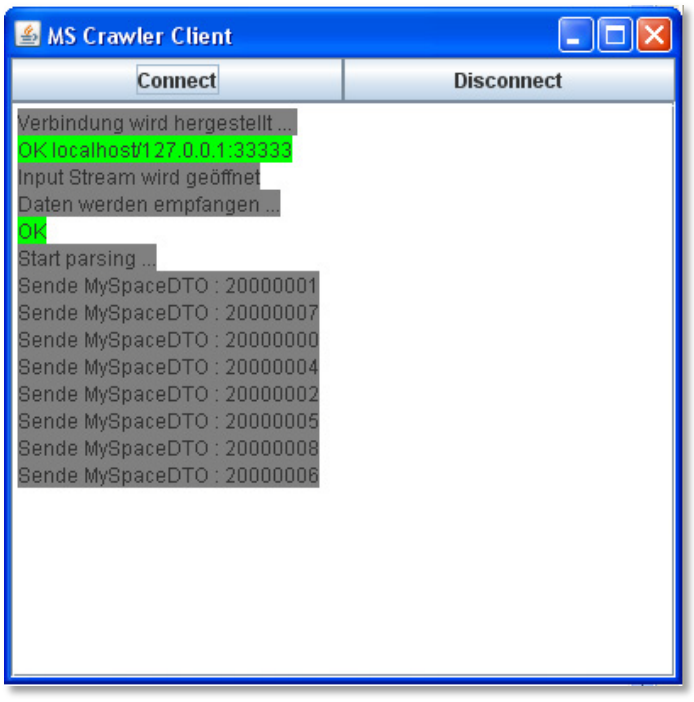


Abbildung 5: MS Crawler Client

Je nach Identifizierung wird der entsprechende Parser aufgerufen (ParseProfile.java oder ParseBand.java).

Die Syntaxregeln für die Regulären Ausdrücke werden in einer externen Property Datei gespeichert. Dadurch kann schnell auf Änderungen von MySpace reagiert und agiert werden.

Die Freundesliste wird bei MySpace über eine andere URL aufgerufen (http://friends.myspace.com/....viewfriends&friendID=123456789) und beinhaltet 40 Freunde pro Seite. Da MySpace jedoch aus Performance Gründen nur eine begrenzte Anzahl an Freunden anzeigt, ist es nicht möglich den kompletten Freundeskreis einer beliebten Person auszulesen. Diese Grenze liegt bei 40.000 Freunden und wird meist nur von Superstars oder dem MySpace Gründer „Tom (6221)“ erreicht.

Für die Kommentare, auf einem Profil, ist die ParseComments.java verantwortlich. Da für das Auslesen aller Kommentare ein Login bei MySpace notwendig ist wurden nur die neusten 50 Kommentare direkt auf der Profilseite ausgelesen.

Als Ergebnis einer erfolgreich analysierten Profile Seite wird eine DTO Datei (Data Transfer Objects) an den Server zurück gegeben. Eine DTO bündelt mehrere Daten in einem Objekt, sodass durch einen einzigen Programmaufruf diese übertragen werden können. Der Server speichert nun über die Datenbankschnittstelle diesen erhaltenen Datensatz (DTO) in die Datenbank.

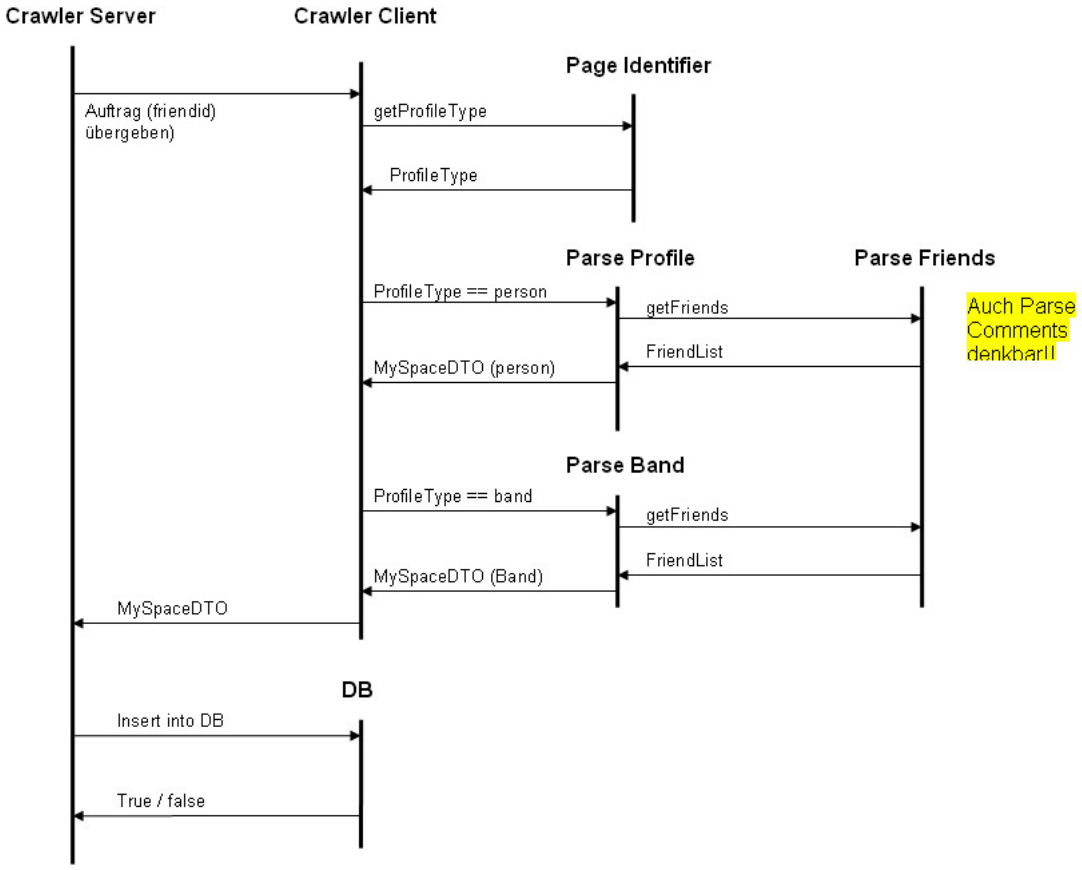


Abbildung 6: Sequenzdiagramm Crawler

Problems

Bei der sequenziellen Inkrementierung der friendid ist eine maximale Ausbeutung von 60%- 70% vorhanden, da viele „Ungültige FriendID“ oder „Private Profile“ berücksichtigt werden müssen. Zudem wird das Soziale Netzwerk nicht komplett erfasst, da nicht alle Freundes– Freunde in der Datenbank gespeichert werden. Durch rekursives Auslesen der MySpace Profile wäre die Ausbeutung bis auf die Privaten Profile optimiert. Zudem würden alle erreichbaren Freundes- Freunde in der Datenbank abgebildet. Durch dieses Vorgehen müsste man mehrere Personen aus verschiedenen Herkunftsländern auswählen, um möglichst viele Profile mit unterschiedlichen Kulturen und Herkunftsländer zu bekommen.

Keine Einstellmöglichkeiten für die IP Adresse auf die der Client verbinden soll. Diese Adresse steht statisch im Quellcode und muss entsprechend der Server IP angepasst werden.

Paketstruktur vom MySpace Crawler

Paketname	Kurzbeschreibung
client	Clientspezifische Client Klassen
db	Crawler spezifische Datenbankobjekte
dto	Crawler spezifische Data Transfer Objects
init	Crawler Initial Klassen
server	Serverspezifische Server Klassen
util	Crawler spezifische Hilfsklassen

Tabelle 3: Paketstruktur

In dem Paket „client“ sind alle für den Client benötigten Klassen. Hierzu zählt der eigentliche Client und Client GUI sowie die Klassen zum Analysieren der Band Profile, Öffentliche Profile, Kommentare und Freunden.

In dem „db“ Paket sind alle Datenbankklassen die der Crawler benötigt. Gleichermaßen befinden sich alle benötigten Data Transfer Objekte in dem „dto“ Paket.

In dem Paket Initial befinden sich die Klassen zum initialisieren von MySpace FriendID’s.

Innerhalb des Paket „server“ befinden sich alle Klassen die für den Betrieb des Server zuständig sind. Hierzu zählt die Server Klasse sowie die Server GUI.

Die Hilfsklassen für den Crawler sind im „util“ Ordner abgelegt.

Datenbank Synchronisation

Die Datensynchronisation ist dazu zuständig um die gesammelten MySpace Daten logisch mit der IMDb abzugleichen.

Filmabgleich

```
public void run()
{
    Collection<ProfilMovieDTO> pmdto = new ArrayList<ProfilMovieDTO>();
    pmdto = CrawlerDB.findAllMoviesByFirstChar(cn_crawler, ch);

    for(ProfilMovieDTO dto : pmdto) {
        int min = 1000;
        MovieDTO mov = null;

        for(MovieDTO movie : movies) {
            int distanz = SyncUtil.levenstein(dto.getTitle(), movie.getTitle());

            if(distanz <= 2) {
                mov = movie;
                break;
            }

            if(distanz < min) {
                min = distanz;
                mov = movie;
            }
        }
        if(mov != null) {

            if(min < (int)(mov.getTitle().length() / 3)) {
                ProfileMovieDB.insert(cn, dto.getId(), mov.getMovieid());
                CrawlerDB.update(cn_crawler, dto.getDieid(), "1");
            }
            else {
                CrawlerDB.update(cn_crawler, dto.getDieid(), "2");
            }

            try
            {
                cn.commit();
            }
            catch (SQLException e)
            {
                e.printStackTrace();
            }
        }

        min = 1000;
        mov = null;
    } // end for
}
```

Abbildung 7: Quellcode Datenabgleich mit Levenshtein

Für den Abgleich wurden die Filme anhand des Anfangsbuchstaben in kleinere Mengen unterteilt. Durch diese Unterteilung wurde der benötigte Aufwand für das berechnen des richtigen Filmes verkleinert. Jeder von den MySpace Benutzer angegebene Lieblingsfilm wurde mit den Filmen aus der IMDb mit dem Levenshtein Algorithmus verglichen. Sobald eine errechnete Levenshtein Distanz kleiner gleich 2 beträgt, wird die Schleife unterbrochen und der gefundene Film wird eingetragen. Durch diesen gezielten Abbruch wird der Abgleich um ein vielfaches schneller, führt aber in manchen Fällen zu schlechten Ergebnissen. Jedoch kann davon ausgegangen werden das es sich bei einer Levenshtein Distanz von 2 um einen Rechtschreibfehler handelt.

Wird kein Film mit einer Levenshtein Distanz von kleiner gleich 2 gefunden, so wird du Suche bis zum Ende durchgeführt. Hat ein Film eine Levenshtein Distanz die kleiner ist als die bereits durchsuchten Filme, so wird dieser Film als optimale Lösung markiert. Aus Toleranzgründen wird nur ein Film eingetragen wenn die kleinste gefundene Levenshtein Distanz kleiner als 1/3 der Textlänge des entsprechenden Filmes beträgt.

Wird ein Film eingetragen so wird dieser als „bearbeitet“ markiert um eine wiederholende Bearbeitung zu verhindern. Filme bei denen die Levenshtein Distanz nicht den Toleranz Abfragen entspricht werden extra markiert.

Stichwort Analyse

Der Inhalt der 6 Millionen gesammelten MySpace Profile wurden mit einem repräsentativen Stichwort Liste über Sex, Drogen und Gewalt analysiert und abgeglichen. Durch diese Analyse konnten die einzelnen Profile auf eins dieser Kriterien charakterisiert werden. Die Durchführung wurde mit dem Bayesian Algorithmus bearbeitet. Als Grundlage für den Algorithmus gibt es drei Listen mit insgesamt rund 2000 Stichworten.

Sex	Drogen	Gewalt
african sex free	amphetamines cocaine	conviction
african sex movie	amphetamines drugs	cop beating
african sex pics	binge drinking	cost of capital punishment
african sex video	brain heroin	court shooting
allgangbang	caffeine addiction	crime
amateur	cannabis drugs	crime murder
amateur adult cams	cocain	criminal arrested
amateur adult web cam	cocain abuse	date rape
amateur adult webcam	cocain addiction	daughter killing
amateur adult webcams	cocain addictive	david killing
weitere …	weitere …	weitere …

Tabelle 4: Beispiel Stichworte Sex, Drogen und Gewalt

Für die Implementierung wurde die Classifier4J (http://classifier4j.sourceforge.net/), eine Java Open Source Library für die Klassifizierung von Texten, verwendet. Classifier4J beinhaltet den Bayesian Algorithmus sowie andere wichtige Funktionen für die Gliederung von Texten.

Der Bayesian Algorithmus gibt bei einer erfolgreichen Findung eines Stichwortes eine 0.99 zurück. Ein neutrales Profil wird mit 0.5 gekennzeichnet. Jedes Profil wird mit den drei verschiedenen Listen analysiert. Die Analyse beinhaltet folgende Profil Attribute: ProfileName, Headline, Header, About, LikeToMeet, HereFor, Mood, SexualOrientation und Kommentare. Die Klassifizierung wird anschließend in der Tabelle PROFILEKEYWORD abgespeichert.

Probleme

Wie schon bereits unter Datenabgleich beschrieben kann es zu Fehlentscheidungen kommen. Durch die Fehlertoleranz von einer Levenshtein Distanz von 2 können falsche Filme zugewiesen werden.

Trotz Toleranz Abfragen ist der Abgleich sehr Zeitaufwendig, da jeder angegeben MySpace Lieblingsfilm mit tausenden Filmen abgeglichen werden muss. Der Aufwand bei der sequenziellen Durchführung beträgt O(N).

Paketstruktur von der Datenbank Synchronisation

Paketname	Kurzbeschreibung
(default)	Datenabgleich spezifische Klassen
data	Daten für den Abgleich
db	Datenabgleich spezifische Datenbankobjekte
dto	Datenabgleich spezifische Data Transfer Objects
util	Datenabgleich spezifische Hilfsklassen

Tabelle 5: Paketstruktur

In dem Paket „(default)“ befinden sich alle ausführenden Klassen, die für den Datenabgleich notwendig sind. Hierzu zählen die Klassen für den Filmabgleich mit dem Levenshtein Algorithmus sowie die Keyword Analyse mit dem Bayesian Algorithmus.

Die Stichwortlisten zu Sex, Drogen und Gewalt befinden sich im „data“ Paket.

In dem „db“ Paket sind alle Datenbankklassen die für den Datenabgleich benötigt werden. Gleichermaßen befinden sich alle benötigten Data Transfer Objekte in dem „dto“ Paket.

Hilfsklassen sind in dem Paket „util“ hinterlegt.

Social Movie Map

Social Movie Map ist eine mit Flex erstellen Webanwendung die alle gesammelten Informationen Visuell darstellt. Für die Darstellung von komplexeren Visualisierungen wurde die BirdEye RaVis von Google Code verwendet.

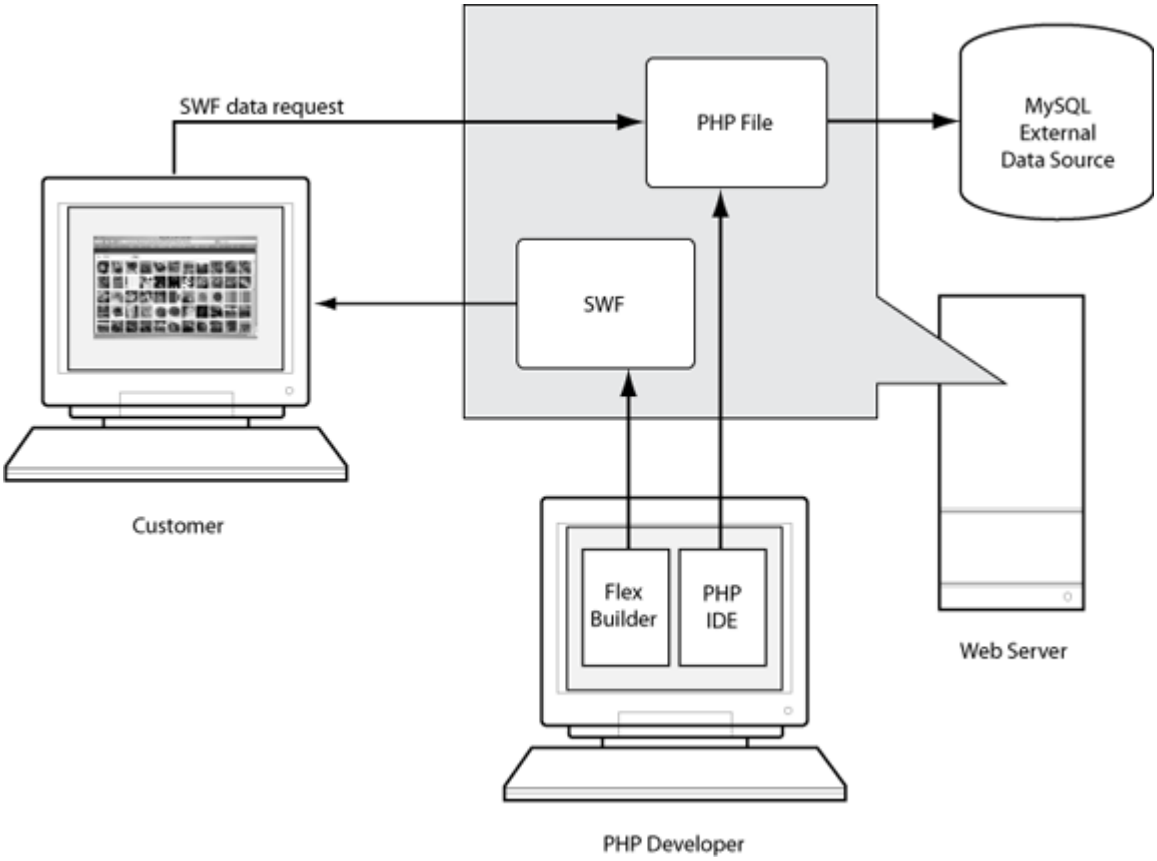


Abbildung 8: Flex und PHP (http://learn.adobe.com/wiki/display/Flex/Flex+and+PHP)

Adobe Flex kann nicht direkt mit einer Datenbank kommunizieren. Um die Anwendung an eine MySQL Datenbank anzubinden, wird PHP mit MySQLi verwendet. Adobe Flex kann über einen HTTPService externe Daten in die Anwendung laden und einen Request an eine PHP Datei versenden. Die PHP Datei erzeugt aus einem erhaltenen MySQL Result eine XML Datei, diese wird dann über die http Antwort in die Anwendung geladen. Adobe Flex kann nun diese Daten, in der Form einer XML Datei, verarbeiten und diese in Diagrammen, Listen oder andere Komponenten darstellen.

Das gleiche Prinzip kann auch mit ASP.NET, ColdFusion oder Java angewandt werden.

Die fünf verschiedene Hauptmenüpunkte (Movies by Group, Movie- Statistic, Charts, Genre-Statistic und Movie Questions) wurden aus Gründen der Lesbarkeit in separaten MXML

Dateien ausgelagert und in der main.mxml aufgerufen. Zusätzlich war das gleichzeitige Arbeiten über CVS möglich.

MXML Datei	Beschreibung
Main.mxml	Kombiniert alle separaten MXML Dateien mit dem Hauptmenü und beinhaltet die Logik für den Fullscreen Modus.
Page1.mxml	Page1.mxml beinhaltet die Visualisierung zu „Movies by Group“ (Top-Film Graph und Top- Film Treemap)
Page2.mxml	Page2.mxml beinhaltet die Visualisierung zu „Movies Statistic“ (Film Suche mit dazugehörigen Statistiken)
Page3.mxml	Page3.mxml beinhaltet die Visualisierung zu „Charts“ (World-Wide Box Charts, IMDb Top Charts, Bottom Movie Charts)
Page4.mxml	Page4.mxml beinhaltet die Visualisierung zu „Genre Statistic“ (Filmjahr Graph, Statistiken)
Page5.mxml	Page5.mxml beinhaltet die Visualisierung zu „Movie Questions“ (7 Fragestellungen inkl. Visualisierung)

Tabelle 6: MXML-Seiten der Appikation

Die Auswahlfelder (XMLComboBox) erhalten ihre Auswahlmöglichkeiten aus externen XML Dateien. Die XML ist nach folgendem Schema aufgebaut:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
    <nodes label="Label 1" data="1" />
    <nodes label="Label 2 data="2" />
</root>
```

Über den httpService von Adobe Flex werden die selektierten Auswahlmöglichkeiten mit der POST oder GET Argumentübertragung an die httpService.php übertragen.

Jede MXML Datei hat eine eigene httpService.php Datei sowie die dazugehörige PHP Datenbankdatei. Die eigentliche SQL Abfrage steht in der PHP Datenbankdatei und wird von der httpService.php aufgerufen. Durch die Übergabe der selektierten oder eingegebenen Felder über POST oder GET kann die gewünschte SQL Abfrage gestartet werden. Die httpService.php wandelt das erhaltene SQL Resultat in eine XML Struktur um und gibt diese auf dem Ausgabestream aus.

Für die Darstellung der XML Dateien gibt es folgende Komponenten im Adobe Flex Framework: AreaSeries, BarSeries, BubbleSeries, CandlestickSeries, ColumnSeries, HLOCSeries, LineSeries, PieSeries und PlotSeries. Für die Darstellung der Treemap in Page1.mxml wurde die Flexlib (Open Source Flex Component Library) von Google Code verwendet (http://code.google.com/p/flexlib/). Der Top Film Graph auf der Startseite wurde mit Birdeye (Information Visualization and Visual Analytics Library) umgesetzt (http://code.google.com/p/birdeye/).

Für die Navigation wurde die ButtonBar Komponente sowie TabNavigator eingesetzt.

Das Farbschema von Social Movie Map wurde mit dem Flex Style Editor (http://examples.adobe.com/flex2/consulting/styleexplorer/Flex2StyleExplorer.html) Version 2.0.1 angepasst.

Caching

Um die Datenbank im laufenden Betrieb zu entlasten gibt es eine Cache Funktion, welche die Ergebnisse einer Abfrage für einen bestimmten Zeitraum zwischenspeichert. Die Cache Dateien werden in dem Ordner „cache“ gespeichert und können innerhalb des bestimmten Zeitraumes von der PHP geladen werden, ohne eine neue Datenbankabfrage zu starten.

Nach Ablauf des Zeitraumes wird die SQL Abfrage gestartet und die alte Cache Datei mit der aktuelleren überschrieben.

Fragestellungen

Alle dick markierten Fragen wurden umgesetzt und als Visualisierung dargestellt. Alle Fragestellungen beruhen auf den Angaben der MySpace Benutzer und können unter Umständen falsch angegeben sein. (Bsp. jährliches Einkommen). Fragen die sich mit dem Freundeskreis befassen, konnten nicht umgesetzt werden da durch die 6 Millionen Profilen nur kleine oder nicht repräsentative Freundeskreise vorhanden sind.

1. **Was sind die Lieblingsfilme einer bestimmten Benutzergruppe (Alter, Herkunft, Geschlecht)**
2. **Von welchen Benutzergruppen wird ein bestimmter Film als Lieblingsfilm angegeben**
3. Welche Film-Genre werden von einer bestimmten Benutzergruppe bevorzugt
4. Bei welcher Benutzergruppe ist ein bestimmtes Genre besonders beliebt.
5. **Wie unterscheiden sich die beliebtesten MySpace-Filme von offiziellen TopMovie-Charts (Top250)**
6. Bei welcher Benutzergruppe ist ein bestimmter Schauspieler besonders beliebt.

7. **Gibt es eine Beziehung zwischen Anzahl der Lieblingsfilme und Anzahl der verlinkten Freunde?**
   Führen viele Freunde auch zu mehr Filmkonsum oder hat der vereinsamte Nerd die größte Lieblingsfilme-Sammlung. Oder ist es eher umgekehrt?
8. Gibt es eine Beziehung zwischen Filmlänge und Lieblingsfilmen? Ab welcher Länge hat ein Film ein erhöhtes Potential ein Lieblingsfilm zu werden.
9. **Gibt es länderspezifische Unterschiede welche Art von Filmen geguckt werden?**
   Welches Genre ist in einem Land vorherrschend.
10. **Wie viel Prozent der User haben Lieblingsfilme, die älter sind als sie selbst.**
11. **Wie hoch sind die Anteile des verschiedenen Genres bei Männern im Gegensatz zu Frauen.**
12. **Wie hoch ist die durchschnittliche User-Bewertung von Filmen, die von Frauen als Lieblingsfilme angegeben werden im Gegensatz zur durchschnittlichen User-Bewertung von Filmen, die von Männern als Lieblingsfilme angegeben werden. Gucken Frauen mehr Schrott als Männer?**
13. Wie hoch ist der Anteil an Lieblingsfilmen in Abhängigkeit von der Zeit, die bereits seit der Veröffentlichung vergangen ist?
14. Wie sähe der anhand von Vorlieben von MySpace-Usern statistisch errechnete Top-Movie aus (Zusammenstellung eines Filmes mit den Top-Schauspielern, Top-Genre, Top-Spielfilmlänge, …)
15. **Welchen Bildungsstand haben die Anhänger eines bestimmten Genres?**
    Sind Horror-Fans gebildeter als Schnulzen-Gucker?
16. **Wie verändern sich die Film-Vorlieben einer Altersgruppe wenn sie Kinder haben?**
17. **Gibt es einen Zusammenhang zwischen Filmbewertung und Anzahl der Lieblingsfilme. (Bspw. Viele eher mittelmäßige vs. wenige, aber gut bewertete Lieblingsfilme)**
18. **Hat die sexuelle Orientierung Einfluss auf die Film-Vorlieben? Haben nicht hetero-sexuelle andere Vorlieben bei den Lieblingsfilmen?**
19. Schauen Jugendliche unter 16 oder 18 Jahren Filme, die mit laut FSK nicht für sie freigegeben sind?
20. Welche Auswirkung hat das Einkommen auf die Art der Lieblingsfilme.
21. Haben Freunde eines bestimmten MySpace-Users den gleichen Filmgeschmack wie diese?
22. **Was ist die durchschnittliche Anzahl an Lieblingsfilmen**
23. **Wie ist die durchschnittliche Anzahl an Lieblingsfilmen pro Altersgruppe? Haben ältere Menschen mehr Lieblingsfilme als junge Menschen?**
24. Welcher Personenkreis hat keine Lieblingsfilme (angegeben)?
25. **Spielt die Religionszugehörigkeit eine Rolle bei der Auswahl der Lieblingsfilme?**

26. Wer sind die Top-Schauspieler bei den MySpace-Usern.
27. Wie hoch ist der prozentuale Anteil alternativer Filmtechniken (bspw. Animationsfilme) bei den Lieblingsfilmen?
28. Wie hoch ist der prozentuale Anteil alter Filmtechniken (bspw. Schwarz-weiß Filme) bei den Lieblingsfilmen?
29. Was sind die Top10 der „Production Companies“ / Producer, die die meisten Lieblingsfilem produziert haben
30. Wie ist das Verhältnis zwischen abgegebenen Votings bei IMDB und der Beliebtheit bei MySpace-Usern? Sind Filme mit vielen Bewertungen auch beliebter?
31. Gucken Menschen mit höherem Einkommen auch höher bewertete Filme als Menschen mit niedrigerem Einkommen?
32. **Gucken Menschen mit höherer Bildung auch höher bewertete Filme als Menschen mit geringerer Bildung?**
33. Haben Singles mehr Lieblingsfilme als Verheiratete?
34. Wie unterscheiden sich die Lieblingsfilme im direkten Vergleich zweier Länder?
35. Wie wirkt sich die Anzahl der gewonnen Movie Awards auf die Beliebtheit eines Films aus
36. Gucken Menschen die in MySpace auch Lieblingsbücher angegeben haben mehr oder weniger Filme als User, die keine Lieblingsbücher haben?
37. Welche Drehorte sind bei den MySpace-Usern am beliebtesten.
38. **Welche Themen (anhand von Keywords) stehen bei den MySpace-Usern hoch im Kurs?**
39. Steht der Nickname von Usern in Zusammenhang mit seinen Lieblingsfilmen (bspw. Name der Hauptrollen, Titel)
40. Wie hoch ist die durchschnittliche Anzahl von Lieblingsfilmen
41. **Inwieweit unterscheiden sich die Lieblingsfilme von verschiedenen ethnischen Gruppen.**
42. Die Einwohner welchen Landes haben die meisten Lieblingsfilme
43. Gucken Menschen mit einer Körpergröße unter 100cm vermehrt Filme aus Genre (Action / Horror / Kriegsfilme …) in denen Normalwüchsige abgeschlachtet werden, oder sind diese Menschen eher besonnener (Komödien / Romanzen / Animationsfilme …)
44. Topfilme / Lieblingsgenre in Abhängigkeit der Größe des Freundeskreises einer Person. (Meinungsbildner)
45. Topfilme / Lieblingsgenre in Abhängigkeit der Anzahl an Kommentaren einer Person. (Meinungsbildner)
46. Haben Personen eines Freundeskreises die gleichen Film-Interessen?

47. Gibt es Auffälligkeiten in der Beziehung zwischen Länge des Nicknames und den Film-Interessen einer Person. Liegen die Film-Schwerpunkte von Personen mit einem 40-stelligem Nickname anders als bei Personen mit kurzem Nickname
48. Sind kürzere Filme generell beliebter als lange Filme oder ist es eher umgekehrt?
49. Wie wirkt sich Überlänge auf die Beliebtheit eines Films aus.

Visualisierung Probleme

Bei dem Versuch eine externe Weltkarte für die Visualisierung der Genreverteilung auf dem Globus einzubinden, gab es Probleme mit Action Script 2. Eine Alternative zu externen Weltkarten ist die Google Maps API für Flash, die auch in Flex eingebunden werden kann. Diese API wurde doch erst gegen Projektende gefunden und fand daher aus zeittechnischen Gründen keinen Platz im Social Movie Map Projekt.

Paket Struktur von Social Movie Map

Paketname	Kurzbeschreibung
db	httpService PHP Dateien
db.db	Social Movie Map spezifische Datenbankklassen
db.inc	Datenbankhilfsklassen
src.assets	Farbschema (Bilder, Style, Fonts)
src.data	Daten für die Auswahlfelder sowie Initialdaten
src.flexlib	Open Source Flex Component Library
src.org	Erweiterungen, benötigten Komponenten
src.reflection	Komponente für die Reflection
src.renderers	Renderer für die Charts
src.styles	CSS Dateien
src	MXML Dateien (main, page1, page2, page3. page4 und page5)

Tabelle 7: Paketstruktur

Die PHP Dateien für die Datenbankabfragen liegen im Ordner „db“, „db.db“ und „db.inc“.

Die benötigten Adobe Flex Dateien und Komponenten liegen im Ordner „src“.

Wireframes / Screen Design

Wireframe

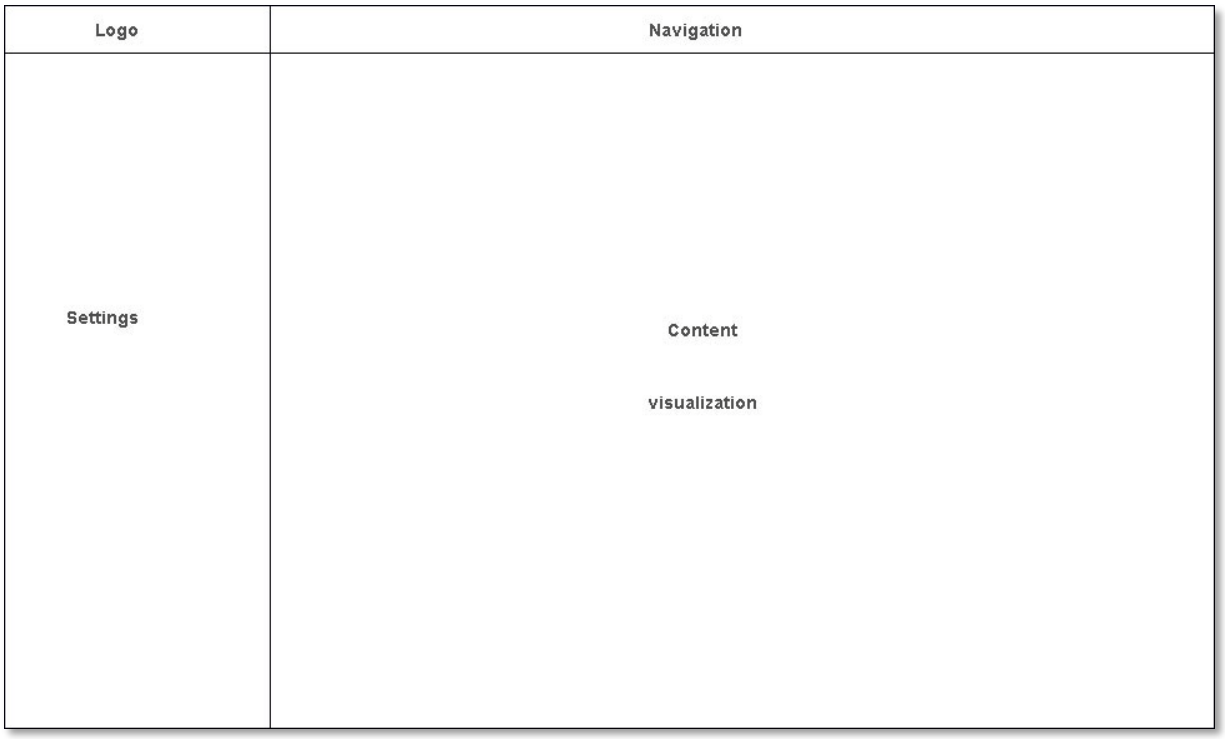


Abbildung 9: Sehr früher konzeptueller Prototyp

Screendesign

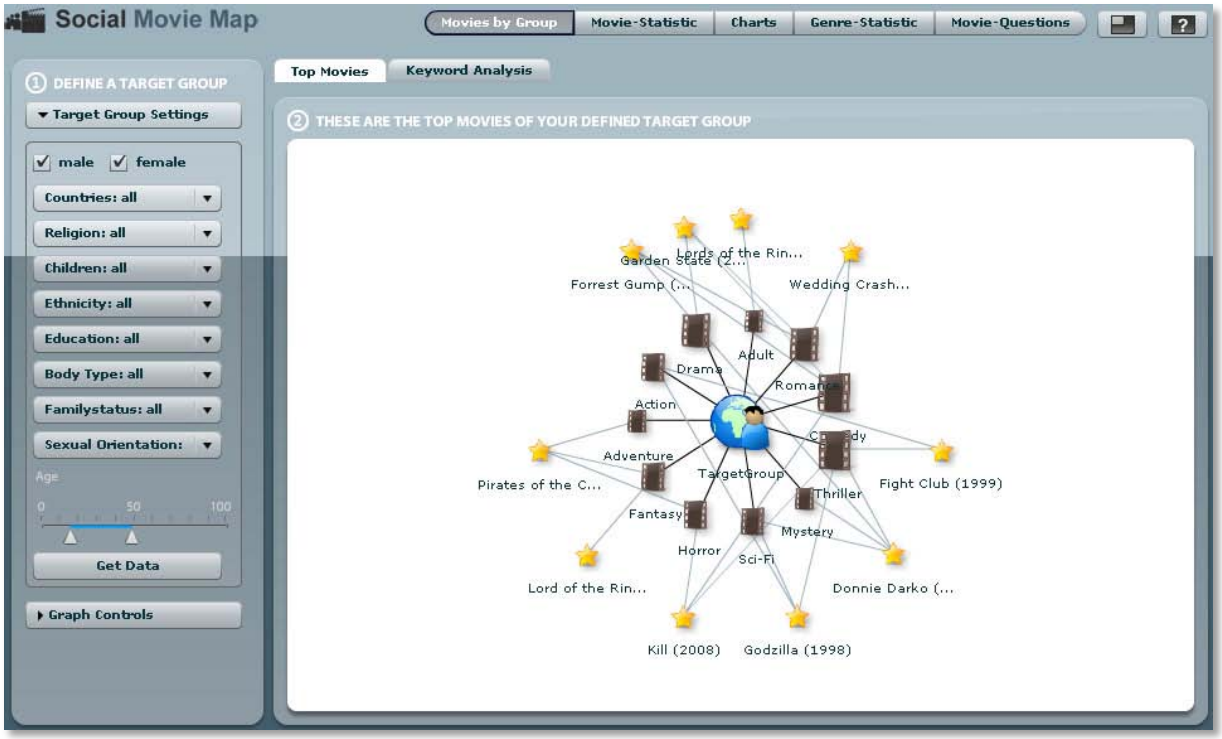


Abbildung 10: Fertige Anwendung

Evaluierung der Ergebnisse

Die Anforderungen des Projektes wurden erfolgreich nach den Anforderungen des Pflichtenheftes erfüllt.

Daten / Fakten

Tabellen	Zeilen / Einträge	Datenlänge
MySpace Lieblingsfilme	3169052	161,5 MB
MySpace Profile	6037948	1788,1 MB
Filme nach Abgleich	813363	34 MB
Profile Keyword Analyse	5000000	213,2 MB

Tabelle 8: MySpace Tabelle

Tabelle	Zeilen / Einträge	Datenlänge
IMDB Filme	771299	63,3 MB

Tabelle 9: IMDb Tabelle

Fazit

Was wurde erreicht?

Erreicht wurde eine funktionale und erweiterbare Web Anwendung für die Analyse von MySpace Profilen und deren Lieblingsfilmen nach den Wünschen des Auftraggebers.

Erweiterungsmöglichkeiten

Das Projekt kann durch weitere gewünschte Statistiken ergänzt werden. Hierbei ist zu achten, dass die gewünschten Informationen im Datenbanksystem zur Verfügung stehen.

Die Google Map API für Flash kann integriert werden, um z.b. die Genreverteilung auf der Google Map Weltkarte darzustellen. Natürlich sind auch weitere Visualisierungen auf der Grundlage von Google Maps denkbar.

Der MySpace Crawler kann mit zusätzlichen Funktionen, für das Auslesen eines Benutzer Profils, ausgestattet werden.

Literaturverzeichnis

- **Information Visualization and Visual Analytics Library**
  http://code.google.com/p/birdeye/wiki/RaVis (26.01.2009)

- **Classifier4J is a Java library designed to do text classification**
  http://classifier4j.sourceforge.net/ (26.01.2009)

- **Google Map API für Flash**
  http://code.google.com/intl/de-DE/apis/maps/documentation/flash/ (26.01.2009)

- **Jericho HTML Parser – Java**
  http://jerichohtml.sourceforge.net/doc/index.html (26.01.2009)

- **MySpace Social Network#**
  http://www.myspace.com/ (26.01.2009)

- **International Movie DB**
  http://www.imdb.com/ (26.01.2009)

- **Adobe Flex Getting Started**
  http://learn.adobe.com/wiki/display/Flex/Getting+Started (26.01.2009)

- **Adobe Flex Style Editor**
  http://examples.adobe.com/flex2/consulting/styleexplorer/Flex2StyleExplorer.html (26.01.2009)

- **Adobe Flex – Komponenten Explorer**
  http://examples.adobe.com/flex2/inproduct/sdk/explorer/explorer.html (26.01.2009)