Movies Classification

Daniel Gavril
"Alexandru Ioan Cuza" University,
Faculty of Computer Science
General Berthelot, No. 16
daniel.gavril@info.uaic.ro

Adrian Iftene
"Alexandru Ioan Cuza" University,
Faculty of Computer Science
General Berthelot, No. 16
adiftene@info.uaic.ro

ABSTRACT

Every device that is released nowadays has the capability to display videos. From small devices like smart phones to bigger ones likes desktops or smart TVs, movies are available at any time. Therefore, getting important, fast and reliable data about movies at any time and everywhere is very important when it comes to money. Computer users like to get live experiences based on their interests, searches or their personal profile and their needs include movies. When people gather to someone's place or a family unites on holidays, finding a good movie to watch for that moment becomes hard. There are different tastes in movies, persons that like drama more than science-fiction or some that will enjoy movies based on the director of that movie or important actors. Using a good search engine that might understand users desires based on recent searches or interests might come in handy. *Movies Classification* application runs on Windows operating systems and tries to understand user perspective when he tries to find a movie that might suit his tastes. The app updates live the suggested movies based on the customizable profile that every registered user must set.

Author Keywords
Movies Classification; Client-Server Architecture; User Profiling.

ACM Classification Keywords
H5.2. Information interfaces and presentation; H3.3. Information Search and Retrieval.

INTRODUCTION

In the last years, the main search engines use the history of user activities in order to provide more accurate results. Google[1], Yahoo[2], Bing[3] are ones of the greatest companies that during decades have worked on search engines that will help users find related data for a given input. There are some search engines for movies that show recommendations based on their preferences. Even if the main subject is movies, finding correlated data and categorize it for each user, it is a difficult task. Currently, there are around 80.000 TV movies, more than 1.500.000 actors and more than 389.000 directors registered to IMDb database[4] [2]. Jinni[5], Taste Kid[6] or Nanocrowd[7] are some web apps that allows you to get movie recommendations in real time. For example, *Jinni* is a recommendation engine that helps you find films based on your mood, time available, setting, or reviews. *Taste Kid* is another example of finding related data for a movie from other areas: music, images, and books. [6]

In what follows, we present a desktop application that is compatible with Windows operating system which offers for authenticated users suggested films based on their profile and searches from the "Internet movie database". The client application queries with a service hosted online the IMDb database and then downloads and stores related data for movies, directors, and actors in a local database in records related to a user profile.

SYSTEM ARCHITECTURE

The architecture of the application is based on a client-server architecture (see Figure 1). At the *server level* the main components are based on a SQL database and on a WCF [7] service. At the *client level* we used the WPF [8] and related API's used to get data from IMDb and YouTube[8].

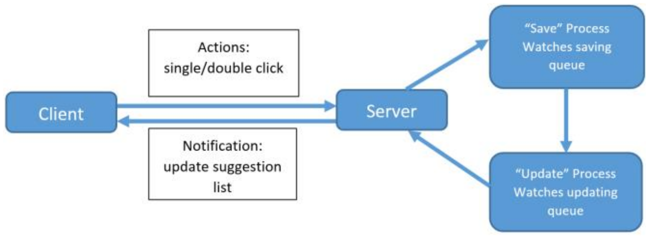


Figure 1. Application Architecture.

Both client and service components use the API's for IMDb and a particular case of the application use the YouTube Data API for trailers. The API's for IMDb are: OmdbApi[9] used to search for movies and MyApiFilms[10] used to find more details (directors and actors with associated movies,

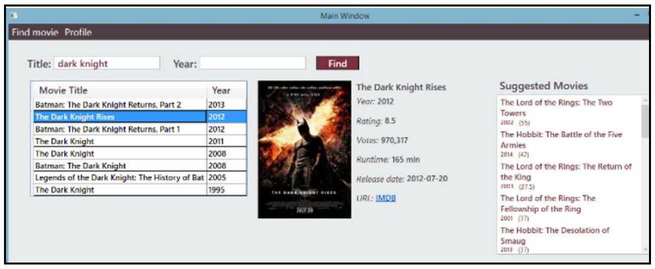
[1] Google: https://www.google.ro/
[2] Yahoo: https://ro.search.yahoo.com/
[3] Bing: http://www.bing.com/?setlang=ro

[4] IMDb: www.imdb.com
[5] Jinni: http://www.jinni.com/
[6] Taste Kid: http://www.tastekid.com/
[7] Nanocrowd: http://www.nanocrowd.com/
[8] YouTube: https://www.youtube.com/
[9] OmdbApi: http://www.omdbapi.com/
[10] MyApiFilms: http://www.myapifilms.com/

genres, plot, votes etc.) for a movie based on a unique identifier (set by IMDb).

The client is structured following the MVVM [3] design pattern recommended by Microsoft community for this type of project. There are three main windows: “login”, “find movies” and “profile”. The “login” window is used to authenticate the user in the application. Therefore, the service can register the actions related to every user in the database. Also, the application has the functionality for new users to register.

The “find movies” window (Figure 2) is divided in 4 parts: the top part has a search section where the user can search for new movies by title and year (optional). The bottom part is divided in three: the left side is used to show the search results, in the middle area is presented the details for the selected movie and the right side is used for suggested movies that the service recommends. There is a secondary window that is opened on double click event on a movie from search list or suggested movies list and give more details about it including a trailer from YouTube.



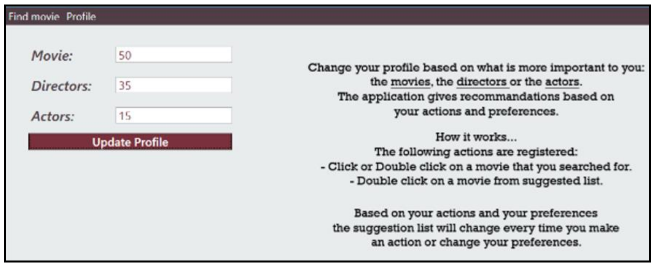
Movie Title	Year
Batman: The Dark Knight Returns, Part 2	2013
The Dark Knight Rises	2012
Batman: The Dark Knight Returns, Part 1	2012
The Dark Knight	2011
The Dark Knight	2008
Batman: The Dark Knight	2008
Legends of the Dark Knight: The History of Bat	2005
The Dark Knight	1995

The Dark Knight Rises
Year: 2012
Rating: 8.5
Votes: 970,317
Runtime: 165 min
Release date: 2012-07-20
URL: IMDB

Suggested Movies
The Lord of the Rings: The Two Towers
2002 (55)
The Hobbit: The Battle of the Five Armies
2014 (42)
The Lord of the Rings: The Return of the King
2003 (37.5)
The Lord of the Rings: The Fellowship of the Ring
2001 (37)
The Hobbit: The Desolation of Smaug
2013 (37)

Figure 2. Find movies window.

The “profile” window (Figure 3) offers the possibility to change preferences about movies, directors or actors. Also, there is a small description about the application functionalities.



Movie: 50
Directors: 35
Actors: 15

Change your profile based on what is more important to you:
the movies, the directors or the actors.
The application gives recommandations based on
your actions and preferences.

How it works...
The following actions are registered:
- Click or Double click on a movie that you searched for.
- Double click on a movie from suggested list.

Based on your actions and your preferences
the suggestion list will change every time you make
an action or change your preferences.

Figure 3. Profile window.

The client interacts with the service in the following cases and each one has its significance: the user clicks on a movie from search list meaning that has some interest for that movie, double click on the same movie means that user starts to get interested in that movie and double click on a suggested movie reflects user high interest on a particular movie. The service may be hosted in IIS[11] web server on a virtual machine or locally and has configured the connection string to the database (locally or hosted in Azure cloud[12]). The service starts two tasks (processes) which checks every five seconds if there is any data to process from their corresponding queue. Every user action is added in the saving queue because the service stores every related data for directors and actors. Therefore, there are some restriction based on the unique id of the movies, directors and actors and saving simultaneously might corrupt or fail the saving process. After a successfully processing of the data, the service sends the results to an updating queue where the corresponding process updates for current user, based on the action significance, the scores for each related entity.

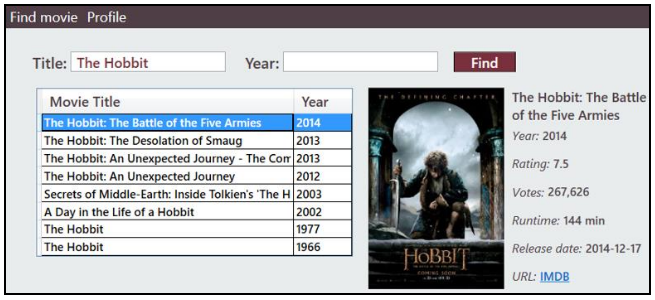
Using a duplex connection between client and service (WCF feature), the service sends to the user a notification that the updating process has finished and he must update the suggested movies list.

CASE STUDIES

Case Study 1

In this section we will present different scenarios related to user actions and how the scoring model works.

Let assume that it is the user’s first login. Therefore, there are no movies related to him. For example, let’s assume he starts to search for “The Hobbit” and no given year. The OmdbApi returns 8 matches that contains the given text.



Movie Title	Year
The Hobbit: The Battle of the Five Armies	2014
The Hobbit: The Desolation of Smaug	2013
The Hobbit: An Unexpected Journey - The Com	2013
The Hobbit: An Unexpected Journey	2012
Secrets of Middle-Earth: Inside Tolkien's 'The H	2003
A Day in the Life of a Hobbit	2002
The Hobbit	1977
The Hobbit	1966

The Hobbit: The Battle of the Five Armies
Year: 2014
Rating: 7.5
Votes: 267,626
Runtime: 144 min
Release date: 2014-12-17
URL: IMDB

Figure 4. Single click selection.

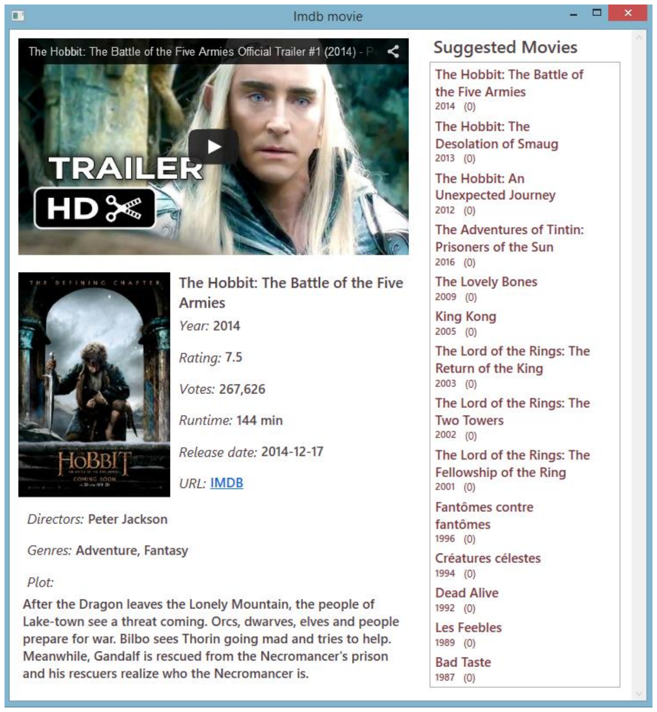
From the list, the user single clicks “The Hobbit: The Battle of five armies” (Figure 4). The action is sent to the server and saves the movie with its details using MyApiFilms. Also, the service saves related movies from the directors and the main actors (with no details). For this pick, there are saved 74 movies in the database and 15 main actors. The service suggests only 1 movie because the single click action is not very relevant to the system. A low score is added for the movie, director and the 15 main actors.

Next, let’s assume that the user double clicks on the same movie (Figure 5). The service will not download any data because it was saved previously so it skips directly to the scoring part. The double click action signifies that the user is starting to get interested, therefore he might like movies

[11] Internet Information Services: https://www.iis.net/

[12] Azure cloud: http://azure.microsoft.com/en-us/

related to the director(s) of this movie. A medium score is added to director of selected movie, to actors of this movie and to every movie of the director. The service suggests this time 14 movies all related to *Peter Jackson*.


Imdb movie

The Hobbit: The Battle of the Five Armies Official Trailer #1 (2014) - P

The Hobbit: The Battle of the Five Armies

Year: 2014

Rating: 7.5

Votes: 267,626

Runtime: 144 min

Release date: 2014-12-17

URL: IMDB

Directors: Peter Jackson

Genres: Adventure, Fantasy

Plot:

After the Dragon leaves the Lonely Mountain, the people of Lake-town see a threat coming. Orcs, dwarves, elves and people prepare for war. Bilbo sees Thorin going mad and tries to help. Meanwhile, Gandalf is rescued from the Necromancer's prison and his rescuers realize who the Necromancer is.

Suggested Movies

The Hobbit: The Battle of the Five Armies
2014 (0)
The Hobbit: The Desolation of Smaug
2013 (0)
The Hobbit: An Unexpected Journey
2012 (0)
The Adventures of Tintin: Prisoners of the Sun
2016 (0)
The Lovely Bones
2009 (0)
King Kong
2005 (0)
The Lord of the Rings: The Return of the King
2003 (0)
The Lord of the Rings: The Two Towers
2002 (0)
The Lord of the Rings: The Fellowship of the Ring
2001 (0)
Fantômes contre fantômes
1996 (0)
Créatures célestes
1994 (0)
Dead Alive
1992 (0)
Les Feebles
1989 (0)
Bad Taste
1987 (0)

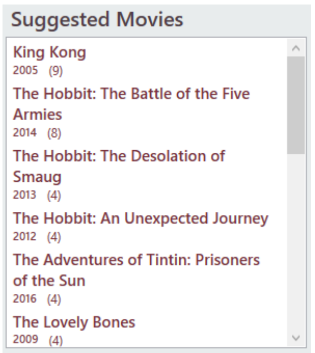
Figure 5. Double click window with no profile set.

Let's assume that the user double clicks on a movie from the suggestion list – "King Kong". This actions suggests that the user might like a movie and shows high interest. Therefore, a high score will be added to director, to the actors of this movie and to the director movies. The service will download any related data - in the database are now 259 saved movies. The suggestion list doesn't change by number because "King Kong" has the same director as "The Hobbit". But now there are 29 main actors associated with current user.

Every movie associated with its directors and actors will determine a score based on the Weighting and Scoring Model [5]. Every user has a profile where he must set "how much" will influence the movie itself, the directors or the actors the ranking of the suggestion list (Figure 2). The total of the percentages must be 100. The following study was structured based on the results from the previous one.

Case Study 2

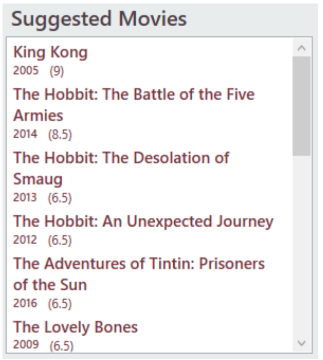
With the current scores for movies, director and actors, if the user sets the percentages of movie to 100 and the rest to 0, the directors and actors will not influence the scoring (Figure 6). Therefore in the top of the list will be the movies that were associated with the most significant action (Top 3: "King Kong", "The Hobbit: The battle of five armies" and 2 other movies related to "The Hobbit" trilogy).


Suggested Movies

King Kong
2005 (9)
The Hobbit: The Battle of the Five Armies
2014 (8)
The Hobbit: The Desolation of Smaug
2013 (4)
The Hobbit: An Unexpected Journey
2012 (4)
The Adventures of Tintin: Prisoners of the Sun
2016 (4)
The Lovely Bones
2009 (4)

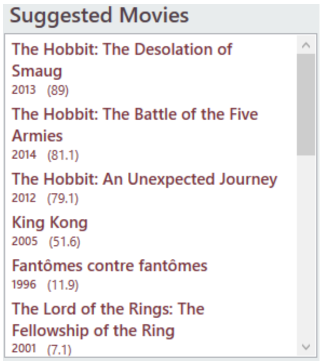
Figure 6. Suggested Movies for Profile Value for *Movie*=100.

Changing the percentages of the movie to 50, the directors to 50 and the actors to 0, the weight of the scores will be split equally between the movie and the director (Figure 7). This set up is useful when there are other directors associated with the user.


Suggested Movies

King Kong
2005 (9)
The Hobbit: The Battle of the Five Armies
2014 (8.5)
The Hobbit: The Desolation of Smaug
2013 (6.5)
The Hobbit: An Unexpected Journey
2012 (6.5)
The Adventures of Tintin: Prisoners of the Sun
2016 (6.5)
The Lovely Bones
2009 (6.5)

Figure 7. Suggested Movies for Profile Values for *Movie*=50 and for *Directors*=50.


Suggested Movies

The Hobbit: The Desolation of Smaug
2013 (89)
The Hobbit: The Battle of the Five Armies
2014 (81.1)
The Hobbit: An Unexpected Journey
2012 (79.1)
King Kong
2005 (51.6)
Fantômes contre fantômes
1996 (11.9)
The Lord of the Rings: The Fellowship of the Ring
2001 (7.1)

Figure 8. Suggested Movies for Profile Values for *Movie*=50 and for *Directors*=20 and for *Actors*=30.

Let's include actors in the scoring model. If the user changes the percentages of the director to 20 and the actors to 30, the suggestion list will show on the top the movies where actors appeared more in the search list (Figure 8). "King Kong" is now on rank 4, top 3 is occupied by the Hobbit trilogy because in all 3 movies appeared the same main actors. Therefore, they "weight" more in the scoring process.

EVALUATION

The process of downloading and saving a movie has the longest execution because a request to MyApiFilms includes a very large amount of data to work with. Moreover, the service must not save twice the same movie, director or actor because of the IMDb id's uniqueness constraint.

Let's assume that the database is empty and a user searches for "Dark Knight" and selects "The Dark Knight Rises" from 2012. The click action is sent to service and starts to work with the given data: the request to MyApiFilms took 10.179 seconds, saving all data to database took 14.259 seconds with a total of 24.474 seconds. In the database are saved: 230 movies, 15 actors, 1 director, with 571 relations between movies and actors and 9 relations between movies and directors. The process of setting the scores between user and movies, director and actors for single click action took 1.168 seconds, but for double click action took 1.562 seconds. Searching again for the same movie the process of saving the movie takes 0.011 seconds, updating the score costs 0.254 seconds for single click and 1.005 for double click.

Testing the application for 10 movies the average score for API to respond is 11.233 seconds, saving to database costs 11.732 seconds, updating the score for single click 1.255 seconds and for double click 1.520 seconds. Moreover, there are saved in the database 233 movies, 1.1 directors, 15 actors, 444 relations between movies and actors and 13.3 relations between movies and directors.

MyApiFilms gives for some movies actors that are not the main cast. This error can't be controlled by the application, therefore the solution might be to contact the developer of the API to look for a solution or correct the request.

Being limited by each API to 2000 request per day, saving the movies to the database save some requests. Also, accessing the data from the database with WCF is faster and easier than downloading and processing it every time.

The WCF service can be hosted in a Web App and can expose the structure of the SOAP messages. Therefore, integrating the service in a different client (Web client) is very easy with .NET technologies.

Deploying the WCF service to Azure cloud is not possible because of the Callback Contract that the service expose. The load balancer can't keep a duplex connection more than 1 minute. Therefore, a different approach for the server-client communication is needed in this case.

Using Entity Framework [1] for mapping the database increases fast and easy development if new features or models will be added. Moreover, Entity has a feature called lazy loading which means that any related data is loaded when the given query requests it.

A custom web scrapper for IMDb will improve the saving process and will offer the chance to download specific details based on user preferences.

CONCLUSION

"Movies Classification" is a desktop application easy to maintain and use, with fast results and live experiences for each user. We decided to create a desktop application, because it can be faster than the web application which depends on the browser's processing power or other elements that requires web control.

For the future we intend to reduce more the duration of the execution for the main methods from the application. Also, we intend to search another solutions to validate information provided now by IMDb, which are not always correct.

ACKNOWLEDGMENTS

The research presented in this paper was funded by the project MUCKE (Multimedia and User Credibility Knowledge Extraction), number 2, CHIST-ERA/ 01.10.2012.

REFERENCES

1. Entity Framework: https://msdn.microsoft.com/en-us/data/ef.aspx
2. Internet movie database stats: http://www.imdb.com/stats
3. Model-View-View Model: https://en.wikipedia.org/wiki/Model_View_ViewModel
4. MyApiFilms: http://www.myapifilms.com/
5. The Weighting and Scoring method: http://www.dfpni.gov.uk/eag-the-weighting-and-scoring-method
6. Top 10 movies recommendation engines: http://www.cnet.com/news/top-10-movie-recommendation-engines/
7. Windows Communication Foundation: https://msdn.microsoft.com/en-us/library/dd456779(v=vs.110).aspx
8. Windows Presentation Foundation: https://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).asp