Jonathan Tew
CPSC 6185
Final Project Report

Mood Based Movie Selector

Aim of the Project:

The goal of this intelligent system is to select a movie for the user to watch primarily based on their mood, and secondarily based on their gender. The system will also take into account a few other variables, which will help influence the decision of the Mood Based Movie Selector.

Problem Description:

As mentioned in the proposal, the idea for this project stemmed from a favorite hobby of mine: watching movies with my friends. I thought it would be interesting to have an intelligent system for my friends and I to use when we are trying to decide on what movie to watch. This intelligent system is similar to other movie selection systems, such as The Movie Suggester and Jinni, but is also quite different at the same time. First off, my Mood Based Movie Selector (MBMS) selects movies from a smaller pool of choices than the two previously mentioned examples; this pool of movies is my personal movie collection. Secondly, MBMS is primarily focused on intelligently choosing a movie that corresponds with how the user is feeling at the time (their mood).

The MBMS system consists of several key components. First, the system has been written in Java using NetBeans 8.0. MBMS is an expert system, and as such utilizes traditional bivalent logic. The user is asked to select two mood descriptors (happy, sad, stressed, etc) that represent how they feel at the moment. There are 12 choices of mood descriptors; six in one drop down box and six in another drop down box. This aspect of the system narrows down the selection of movies to two different genres. Below the mood choices, the user is asked to select their gender, Male or Female. Depending on the gender of the user, as well as how many movies currently match the genres translated from the moods, a series of rules are applied to further narrow down the selection of movies. Additionally, users may filter the movie selection based on their preference for run-time (short, medium, or long movies), age of movie (selecting a cut-off year for releases), award winning (has it won awards, not won awards, or either), and Internet Movie Database (IMDb) rating (selecting a minimum IMDb score the movie must have). The final key component of the system is an XML file used to store all the data for my movies. The tag-based syntax of XML lends itself perfectly for organizing the movies and their characteristics. A list of movie objects is created from the XML file, which is then queried based on the parameters provided, and the results of these queries are the output of the system. MBMS returns 1-3 movies for each user, which gives them a small choice of movies to watch based off of their mood.

MBMS has evolved some since the project proposal. The main change comes from the type of intelligent system it is based on. In the proposal, I stated MBMS would utilize a mixture of fuzzy logic and bivalent logic. As I delved into the design stages of the system, it became apparent that the use of fuzzy logic was inappropriate for how I needed the system to perform. The fuzzy logic component was mainly to be used for converting user's moods into genres. This might have worked for a smaller set of moods, which could then be rated on a scale and the output would correspond to different genres. However, I wanted to be able to offer a variety of mood descriptors to users, in which case it made more sense to use traditional bivalent logic to set up rules that dictated how combinations of moods would relate to different combinations of genres. Other than this change, there are just a few minor differences. I cut the list of moods from 15 to 12, eliminating three descriptors that were redundant. Additionally, I removed the parameter for users to choose either black and white or color movies, as my collection of movies did not have a large enough representation of black and white movies for this parameter to be useful at this time. However, I have left the framework in place within the code for this

to be implemented in the future very easily. Finally, the role of the gender parameter ended up playing a larger role in MBMS than originally anticipated. I was able to incorporate the gender parameter into the filtering process based on a study examining gender preferences for different movie genres.

Intelligent System Methodology and Justification:

MBMS is an intelligent system using a rule-based expert system methodology. Rule-based expert systems are constructed of five main components, which are the database, the knowledge base, the inference engine, the explanation facilities, and the user interface. MBMS has been designed and built around these five components. First, the database is where all the facts are stored, which in this case are the movies and their attributes. In the MBMS system, the database is the external XML file, which can grow and change easily with new movies (facts) or updates to existing movies without impacting the other components of the system. Next, the knowledge base is where the domain knowledge is stored. This domain knowledge is used to solve the problems the system has been created for, which in this case is deciding which movies to watch based on a user's mood. The domain knowledge in MBMS is represented by rules constructed in the IF-THEN structure where the IF represents the antecedent (or condition) and the THEN represents the consequent (or the action). The majority of the rules in MBMS's knowledge base consist of the rules used to decide what combinations of moods translate to what combinations of genres. There are also rules dictating how gender affects the decision of the system. Next, the inference engine is where the system's reasoning is performed and solutions are produced. The inference engine combines the rules from the knowledge base and the facts (movies) from the database to accomplish this. In MBMS, the inference engine is where the movie objects are searched using the rules from the knowledge base, as well as the rules for the other parameters the user has control over (release date, IMDb score, awards, and runtime). The next component of the system is the explanation facilities. In MBMS, this is where the reasoning used to arrive at the solution is kept track of, and then provided to the user to explain why those decisions where made. More specifically, MBMS keeps track of how the moods are being translated into genres, as well as how gender is impacting the search through additional genres. When the solutions are presented to the user, the genres used are also presented. Finally, the last component of the rule-based expert system is the user interface. In MBMS, this is the Graphical User Interface (GUI). The GUI allows the user to select the moods that represent how they currently feel, as well as select their gender and any additional parameters they would like to incorporate into the decision. The solutions are displayed in the GUI, along with the reasoning and more information on each movie. The GUI allows the user to conveniently and easily interact with MBMS.

A rule-based expert system was chosen for MBMS for several reasons. First, the majority of the decision making abilities of the system hinged on how user's moods correlated with movie genres. The antecedent/consequent structure of the rule-based expert system made the most sense for this task. This style of rules allowed me to structure how each combination of moods would equate to a specific combination of genres. Additionally, this rule structure would allow the gender parameter to be applied as a “tie-breaker” by having the associated gender rules fire when needed. Another reason for the use of the rule-based expert system was for the purpose of easily explaining to the user why the specific movies presented where chosen. With this type of system, the genres used to reach each solution may be tracked, and then presented to the user. This type of decision making allows the user to understand that genres are the key decision making factor of the system.

Design and Implementation Details:

The design of MBMS evolved through many stages. One of the biggest challenges was working out how the XML file containing all of the movies and their attributes would be queried through Java. I ended up using an open source library called dom4j to work with the XML file. The dom4j library allowed me to parse the XML file and use it as I needed. One of the biggest hurdles I faced was getting

the multiple genres for each movie extracted properly. In the end, what worked best was creating individual movie objects for each movie out of the XML file and storing those objects in a list that I could then apply the different rules to. Each object contains all of the movie attributes. Speaking of the XML file, actually creating it was a very tedious task. As I have mentioned, one of the design goals of MBMS was to use it for my own personal movie collection. I manually entered each movie from my collection, along with each of its attributes into the XML file. I did this so that I could control which attributes I was inputting, as well as being certain everything was formatted correctly and equally. As it stands now, the XML database has all 175 of my movies. It will be easy to add additional movies to the database as they are acquired, as the bulk of the work is already done.

The size of the database created design problems all of its own. The main challenge related to this was defining the rules with my particular collection of movies in mind. As this collection is so small compared to say the IMDb database, it is a challenge to not over filter the results, which could lead to no matching movies. When designing the rules that would be used to decide which genres the certain moods would equate to, I used a website where I have kept track of my movie collection for the past couple of years. The website, blu-ray.com, provides several different statistics on user's personal collections, one of which is a genre distribution break-down. Here is what it looks like for my collection:

Collection statistics

Overview | Owned | Wishlist | Watched | Not watched | Printer friendly | Manage categories | Update collection count | Search collection
Recommendations | Statistics | Cast & crew | Collection matcher | Trade matcher | Import | Export

Movies	133 (5 TV)
SKUs	118
SteelBooks	9
DigiBooks	4
Region coding	Region free: 51, Region A: 112 (locked 57), Region B: 0 (locked 0), Region C: 0 (locked 0), Unverified: 6
Average runtime	138min
Genre distribution	

13,0% Drama
10,5% Thriller
9,4% Action
7,4% Adventure
7,0% Crime
7,0% Sci-Fi
5,1% Mystery
4,2% Comedy
3,4% Foreign
3,2% Period
2,9% Romance
2,7% Epic
2,3% Horror
2,3% Fantasy
2,2% Comic book
2,2% Dark humor
1,6% Surreal
1,4% Psychological thriller
1,3% Biography
1,3% History
1,1% Martial arts

Relative genre distribution ?
Strongly overrepresented: Dark humor
Overrepresented: Comic book, Coming of age, Crime, Epic, Foreign, Melodrama, Mystery, Period, Psychological thriller, Sci-Fi
Underrepresented: Documentary
Strongly underrepresented: Animation, Anime, Erotic, Family, Music, Musical, Short, Sport

Search collection: Action, Adventure, Animation, Anime, Biography, Comedy, Comic book, Coming of age, Crime, Dark humor, Documentary, Drama, Epic, Erotic, Family, Fantasy, Film-Noir, Foreign, Heist, History, Holiday, Horror, Imaginary, Martial arts, Melodrama, Music, Musical, Mystery, Nature, Other, Period, Psychological thriller, Romance, Sci-Fi, Short, Sport, Supernatural, Surreal, Teen, Thriller, War, Western

Using this knowledge, I decided that the first drop down box of six moods would each correspond to one of the top six genres from my collection. These top six genres are Drama, Thriller, Action, Adventure, Crime, and Sci-Fi. As the majority of my collection featured one of these genres (54.3%), I could then pair each of them with a more under-represented genre, such as Romance (2.9%), or Surreal (1.6%). Following this strategy, I was able to create rules that fairly evenly represented my collection of movies. After this, I chose six mood descriptors that I thought best represented the top six genres. The remaining six mood descriptors I chose based on how I thought combinations of moods might

equate to combinations of genres. The final list of mood descriptors and genres were as follows:

Mood 1
Cheerful
Happy
Angry
Good
Sad
Weird
Mood 2
Grumpy
Stressed
Energetic
Peaceful
Mad
Melancholy

Genres	
Action	Foreign
Adventure	History
Biography	Horror
Comedy	Mystery
Crime	Period
Dark Humor	Romance
Documentary	Sci-Fi
Drama	Surreal
Epic	Thriller
Family	War
Fantasy	Western
Film-Noir	

The second set of mood descriptors were much more dynamic in their relation to genres. Testing was done to make sure the different combinations represented the entirety of my collection, and tweaks were made to adjust this. In the end, the rules were as follows:

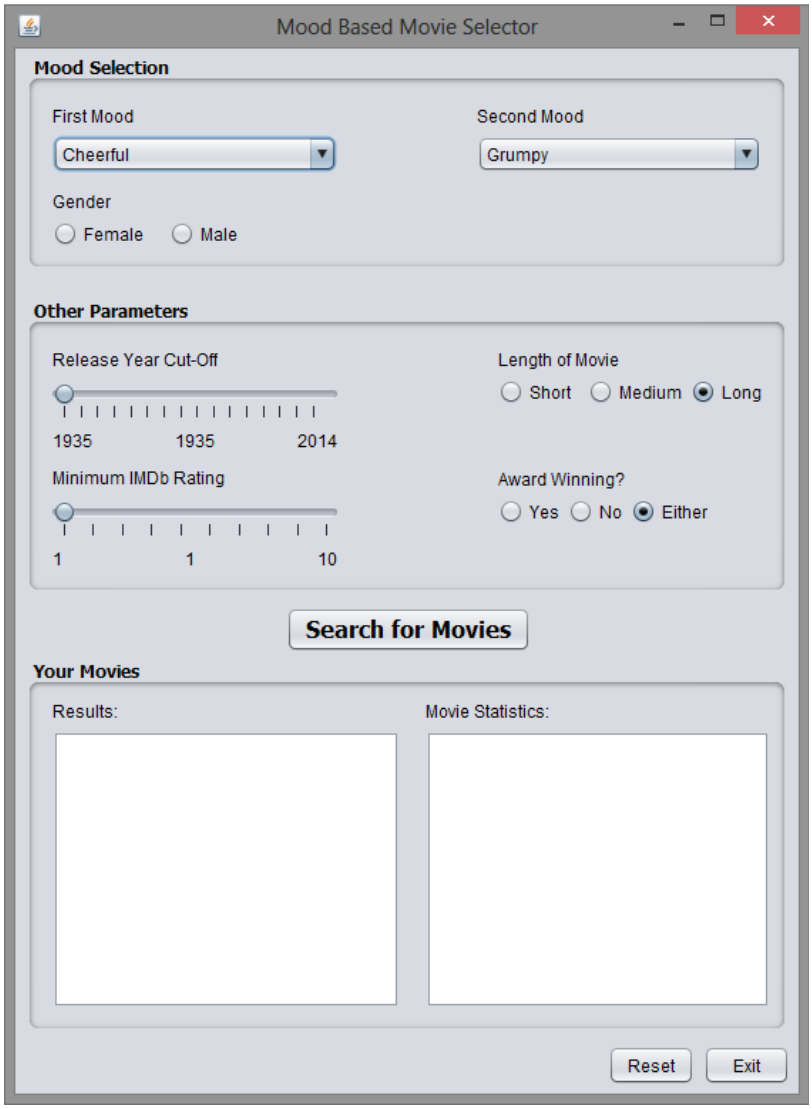
Antecedent		Consequent	
IF	AND	THEN	AND
Mood 1	Mood 2	Genre 1	Genre 2
Cheerful	Grumpy	Comedy	Crime
Cheerful	Stressed	Comedy	Action
Cheerful	Energetic	Comedy	Sci-Fi
Cheerful	Peaceful	Comedy	Adventure
Cheerful	Mad	Comedy	Dark Humor
Cheerful	Melancholy	Comedy	Romance
Happy	Grumpy	Adventure	Mystery
Happy	Stressed	Adventure	Sci-Fi
Happy	Energetic	Adventure	Epic
Happy	Peaceful	Adventure	Action
Happy	Mad	Adventure	Drama
Happy	Melancholy	Adventure	Fantasy
Angry	Grumpy	Action	Drama
Angry	Stressed	Action	Period
Angry	Energetic	Action	Thriller
Angry	Peaceful	Action	Foreign
Angry	Mad	Action	Mystery
Angry	Melancholy	Action	Crime

Antecedent		Consequent	
IF	AND	THEN	AND
Mood 1	Mood 2	Genre 1	Genre 2
Good	Grumpy	Drama	Surreal
Good	Stressed	Drama	Family
Good	Energetic	Drama	Dark Humor
Good	Peaceful	Drama	History
Good	Mad	Drama	Biography
Good	Melancholy	Drama	Film-Noir
Sad	Grumpy	Thriller	Surreal
Sad	Stressed	Thriller	Drama
Sad	Energetic	Thriller	Horror
Sad	Peaceful	Documentary	Epic
Sad	Mad	Action	War
Sad	Melancholy	Thriller	Sci-Fi
Weird	Grumpy	Western	Sci-Fi
Weird	Stressed	Comedy	War
Weird	Energetic	Crime	Period
Weird	Peaceful	Crime	Foreign
Weird	Mad	Crime	Thriller
Weird	Melancholy	Crime	Dark Humor

The rules for the gender parameter are based off a study done by Stuart Fischoff, Joe Antonio, and Diane Lewis. The study, “Favorite Films and Film Genres As a Function of Race, Age, and Gender”, found that the biggest differences in genre preference came in the Action-Adventure genre (males preference), Romance genre (female preference), and Sci-Fi genre (males preference). The study also found that both genders preferred Drama more than any other genre, with females preferring it just slightly more than males. Based on these findings, the gender parameters were implemented as a tie-breaker. If the genres produced from the moods selected (and any other user adjusted parameters) resulted in more than three movies matching these criteria, a gender rule would fire in an attempt to adjust the decision down to three or less movies. If, after this gender rule was applied, there were still

more than three movies returned, another gender rule would fire in an attempt to adjust the decision down to three or less movies again. The first set of gender rules tries applying the Romance genre for females, and the Action genre for males. If these rules have no effect on the results, then a back-up set are applied with Drama for females, and Sci-Fi for males. If the first set of rules were successful, but there are still more than three movies selected, then the second set of rules are applied. This set is the same Drama for females and Sci-Fi for males as used for the back-up rule for the first set of gender rules.

As stated previously, I used the NetBeans 8.0 Integrated Development Environment for implementing MBMS in Java. For the GUI, I used NetBeans design interface for designing and implementing the graphical components. The GUI features the use of combo boxes, radio buttons, sliders, text areas, and buttons. I kept the design simple and clean, breaking up the interface into three distinct areas: “Mood Selection”, “Other Parameters”, and “Your Movies”. The finished design looks as follows:



Instructions for the User:

(The following instructions are for Windows users, I do not have a Mac to test this on and am not sure how it performs in that environment) MBMS was built in NetBeans to work with JDK version 5 or newer. So, first the user needs to make sure their Java is updated to at least version 5. From here, if they open up the “Movie Selector App” folder they will see a “MovieSelectorV1.6.1.exe” file and a

"MovieSelectorV1.6.jar" file. The .exe file was created using a third party software called "Launch4j" and *should* work by just double clicking the file to run. However, I did have an issue testing this file on two of five machines I tried it on. If it does not work, then the .jar file is the backup. If the user has their PATH environment variables configured to run JAR files, then double clicking the "MovieSelectorV1.6.jar" file should open the application right up. If not, the PATH environment variable may need to be configured. Refer to the documentation here under "Setting the PATH Environment Variable" for how to do that.

Once the application is open, the user should choose one mood from each drop-down box that describes how they feel at the moment. Also, they should choose their gender (this is not required, although it is suggested for more accurate results!). From here, the user may click the "Search for Movies" button to receive a list of movies in the "Results" box. The "Movie Statistics" box next to this will display the selected movies information. The "Other Parameters" area is set to the default parameters which do not filter the movies at all. If the user wishes to filter the movies MBMS chooses for them, they may adjust these parameters accordingly. The "Release Year Cut-Off" slider sets the cut-off for year a movie was made. For instance, if 1980 is chosen, no movies before 1980 will be considered. The "Minimum IMDb Rating" slider allows users to set a minimum rating for movies that are returned. For instance, if 8 is chosen, no movies with ratings below 8 will be considered. The "Length of Movie" radio button group allows users to filter the run-times of movies returned. The "Long" button considers movies of all lengths, the "Medium" button considers movies with run time's 135 minutes and shorter, and the "Short" button considers movies with run time's 100 minutes and shorter. Finally, the "Award Winning" radio button group allows users to select whether or not they want movies that have won awards, not won awards, or either way. Here, "award winning" is defined as any movie that has won an Academy Award, a Golden Globe Award, or a BAFTA Award. Note: If no movies can be selected based on the chosen parameters, a message will pop-up advising the user to broaden their search.

Test Results:

Testing MBMS proved to be time-consuming. Each combination of moods had to be tested, along with checking the output to make sure it was working correctly. Several bugs were found, along with several design flaws. These issues were fixed, strengthening the system, and in some cases providing additional features. One example of this was in getting the genres of the movies that were selected. I originally was querying the XML file and was really only able to return the movie titles this way. This forced me to rethink how this was working, which lead to the creation of individual movie objects, each with its own attributes. This allowed me to create the "Movie Statistics" area, a feature I had originally not thought to add.

MBMS selects at a minimum one movie (with default "Other Parameters"), and at most three movies for each user based on their mood and gender (if selected). As instructed in the assignment, I have taken screen dumps of many examples and placed them in a file in my project folder. Please refer to the folder "Screen Dumps" for those images. One of the series of images I would like to point out is a result that selects movies for males who are feeling "Weird" and "Mad" based on four genre matches. This result is an example of a case when both gender rules have been applied to select movies. Also, there are a few more images related to this one that show how applying the "Other Parameters" impact the use of the gender rules.

Looking at the results of MBMS, it is interesting to note that when the male gender is selected, the program has more success applying the first, and sometimes second gender rules to the process. In fact, there is not a case where BOTH sets of gender rules are applied for females. This is interesting as it reflects the actual database of movies, which belong to me – a male. I think this speaks to how accurate and appropriate the gender rules are.

To illustrate the effectiveness of MBMS, I have logged ten sets of test results for ten different

mood combinations, each for Female and Male gender parameters as a brief overview of the typical output. Looking at the spreadsheet, we can see a wide variety of results for the different mood combinations, and also not much overlap between the results for Females and the results for Males. This indicates the rule set is working effectively, giving the desired results.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
Mood 1	Cheerful	Cheerful	Angry	Good	Happy	Happy	Sad	Sad	Weird	Weird
Mood 2	Grumpy	Peaceful	Stressed	Grumpy	Energetic	Melancholy	Mad	Energetic	Mad	Melancholy
Gender	Female	Female	Female	Female	Female	Female	Female	Female	Female	Female
Output 1	Bottle Rocket	The Brothers Bloom	Wyatt Earp	Black Moon	2001: A Space Odyssey	The Princess Bride	Red Dawn	American Psycho	Breathless	Memento
Output 2	Risky Business	The Princess Bride	Samurai II: Duel at Ichijoji Temple	Rosemary's Baby	Game of Thrones	N/A	The Thin Red Line	The Walking Dead	N/A	Seven Psychopaths
Output 3	N/A	N/A	Seven Samurai	American Psycho	Lawrence of Arabia	N/A	Tears of the Sun	N/A	N/A	Pulp Fiction
Genres Used	Comedy, Crime, Romance	Comedy, Adventure, Romance	Action, Period, Drama	Drama, Surreal	Adventure, Epic, Drama	Adventure, Fantasy, Romance	Action, War	Thriller, Horror, Drama	Crime, Thriller, Romance	Crime, Dark Humor, Drama

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
Mood 1	Cheerful	Cheerful	Angry	Good	Happy	Happy	Sad	Sad	Weird	Weird
Mood 2	Grumpy	Peaceful	Stressed	Grumpy	Energetic	Melancholy	Mad	Energetic	Mad	Melancholy
Gender	Male	Male	Male	Male	Male	Male	Male	Male	Male	Male
Output 1	21 Jump Street	Ghostbusters II	X-Men:First Class	12 Monkeys	The Dark Knight	X-Men:First Class	Red Dawn	Alien	The Dark Knight Rises	Repo Man
Output 2	The Other Guys	N/A	N/A	Brazil	Inception	John Carter	The Thin Red Line	Prometheus	Looper	RocknRolla
Output 3	Pineapple Express	N/A	N/A	N/A	Batman Begins	The Avengers	Tears of the Sun	Alien:Resurrection	Inception	N/A
Genres Used	Comedy, Crime, Action	Comedy, Adventure, Action	Action, Period, Sci-Fi	Drama, Surreal, Sci-Fi	Adventure, Epic, Action, Sci-Fi	Adventure, Fantasy, Action, Sci-Fi	Action, War	Thriller, Horror, Action	Crime, Thriller, Action, Sci-Fi	Crime, Dark Humor, Action

Discussion:

Based on my extensive testing of the program, MBMS has proved to provide the intended results, including a wide array of results that accurately reflect my personal movie collection. This was one of the biggest design challenges of this project, being able to implement this system that would not only work, but work well with my collection of movies. Part of this challenge was the size, as mentioned earlier in the paper. My database is not large enough to apply tons of rules to, as it would just end up not being able to find any results. Conversely, I could not just use a handful of rules to solve this problem, as then the results would be too large and generic. It was a difficult balance to achieve between the two extremes. Another challenging aspect of the project was the main set of rules themselves, which correlated the moods to the genres. I could not find much in the way of studies that showed any correlation between these two, so I became my own domain expert and knowledge engineer for the project. The first set of moods relate to a specific genre (for the most part), which was the easier of the two sets of decisions to make: for example it makes sense to me for cheerful to relate to comedy as a person is in a cheerful mood and wants to laugh and have fun, while happy translated to adventure as I thought a happy person might be in the mood for an adventure! Another example is good mood correlated to drama, as I thought the most "standard" mood would go well with the most "standard" genre, as seen from the study of Fischoff et al. that the most popular genre was drama. However, the more challenging aspect came when combining the first mood with the second, and choosing a secondary genre. I approached this by thinking of the combination of moods and what sort of movies might play well to those. I think the results show that these rules worked out nicely.

This was an intensely challenging project for me. I have never designed, implemented, and built a project of this magnitude before. Also, this was my first time using XML for anything, which I think worked out nicely for its intended use in this project. Overall, I would say the Mood Based Movie

Selector project has achieved its aim. I set out to create a program that used a novel approach to selecting movies for users based on their mood and their gender. MBMS fulfills this challenge and has already been a hit with my family and friends!

Conclusion:

I would like to take this section to say thank you for a thoroughly enjoyable class this semester. Of my three classes, this one has definitely been the most work, but I would not have it any other way. I feel like I have learned so much from all the discussions, the projects, the book, and from you that it has certainly been worth the time and effort. See you around campus!

Resources:

[1] Stuart Fischoff, Joe Antonio, Diane Lewis. Favorite Films and Film Genres as a Function of Race, Age, and Gender. Journal of Media Psychology, Volume 3, Number 1, Winter 1998.
[2] Max Sauer, Patrick Keegan. Packaging and Deploying Desktop Java Applications. https://netbeans.org/kb/articles/javase-deploy.html#Exercise_2
[3] Michael Negnevitsky. Artificial Intelligence: A Guide to Intelligent Systems. 2011.
[4] dom4j: http://dom4j.sourceforge.net/
[5] Blu-Ray.com: http://www.blu-ray.com/
[6] The Internet Movie Database (IMDb): http://www.imdb.com/?ref_=nv_home
[7] launch4j: http://launch4j.sourceforge.net/
[8] Jinni: http://www.jinni.com/
[9] The Movie Suggester: http://themoviesuggester.com/
[10] Stereomood: http://www.stereomood.com/more/allmoods