Noname manuscript No.
(will be inserted by the editor)

Recommendations in a heterogeneous service environment

Christian Überall · Christopher Köhnen · Veselin Rokocevic · Rudolf Jäger · Erich Hoy · Muttukrishnan Rajarajan

Received: date / Accepted: date

Abstract This paper presents novel algorithms which are able to generate recommendations within a heterogeneous service environment. In this work explicitly set preferences as well as implicitly logged viewing behavior are employed to generate recommendations for Digital Video Broadcast (DVB) content. This paper also discusses the similarity between the

Christian Überall
City University London
Northampton Square, London EC1V 0HB, United Kingdom
E-mail: Christian.Ueberall.1@city.ac.uk
University of Applied Sciences
Wilhelm-Leuschner-Strasse 13, 61169 Friedberg, Germany
E-mail: christian.ueberall@iem.fh-friedberg.de

Christopher Köhnen
City University London
Northampton Square, London EC1V 0HB, United Kingdom
E-mail: Christopher.Koehnen.1@city.ac.uk
University of Applied Sciences
Wilhelm-Leuschner-Strasse 13, 61169 Friedberg, Germany
E-mail: christopher.koehnen@iem.fh-friedberg.de

Veselin Rokocevic
City University London
Northampton Square, London EC1V 0HB, United Kingdom
E-mail: V.Rakocevic@city.ac.uk

Rudolf Jäger
University of Applied Sciences
Wilhelm-Leuschner-Strasse 13, 61169 Friedberg, Germany
E-mail: rudolf.jaeger@iem.fh-friedberg.de

Erich Hoy
University of Applied Sciences
Wilhelm-Leuschner-Strasse 13, 61169 Friedberg, Germany
E-mail: erich.hoy@mnd.fh-friedberg.de

Muttukrishnan Rajarajan
City University London
Northampton Square, London EC1V 0HB, United Kingdom
E-mail: R.Muttukrishnan@city.ac.uk

DVB genres and YouTube categories. In addition it presents results to show the comparison between well known collaborative filtering methods. The outcome of this comparison study is used to identify the most suitable filtering method to use in the proposed environment. Finally the paper presents a novel Personal Program Guide (PPG), which is used as a tool to visualize the generated recommendations within a heterogeneous service environment. This PPG is also capable of showing the linear DVB content and the non-linear YouTube videos in a single view.

Keywords personalized television, recommendations, content-based, collaborative filtering, similarity, media convergence, Personal Program Guide, DVB, YouTube

1 Introduction

In todays connected world large amount of rich content is available to the consumers from several media service providers. They provide users with linear (live), as well as non-linear (on demand) content. In general linear media consists of DVB and the non-linear media content is available through online portals such as YouTube and many others. Today there are several channels offered through DVB and this number is continuously growing. For example DVB-C (DVB-Cable) and DVB-S (DVB-Satellite) offer more than 200 and 2000 channels respectively.

However, new technological advances allow users to access these different content sources using one device. The next generation of set-top-boxes (STB) and television sets are not restricted to DVB tuners. They will also provide an Internet interface to join the World Wide Web (WWW). Furthermore these new devices will also contain applications, which will enable access to video portals, such as YouTube.

On the other hand advancements in the broadcasting technologies and content delivery platforms today overload the users with the large amount of content. This makes it difficult for the consumers to quickly identify content that is of interest to them. Hence a personalization tool is necessary that can automatically filter out the most appropriate content to the user based on his/her past viewing behavior. The generated recommendations should consider that the users have access to several media sources. In order to overcome this problem new algorithms will have to be researched and developed that can generate recommendations within this heterogeneous media environment based on the users past behavior and preferences. These algorithms will have to consider several metadata that are delivered by these media sources.

Furthermore in this study the use of metadata within the media source is exploited. For instance a DVB Transport Stream includes Service Information (specified by a standard of European Telecommunications Standards Institute (ETSI) [14]), which contains metadata and further information. YouTube offers an API, which is able to extract metadata too. The present paper uses the metadata within the media source without accessing other metadata sources. This will guarantee, that the system is able to work without using metadata from other sources.

In addition to this an interface will be developed to present the recommendations to the user on a simple visualizer providing the user of the new system a better browsing experience.

The paper also presents novel techniques to generate recommendations within a media convergent environment. The developed interface will enable users to visualize several media sources within a single view. The paper also suggests a method of connecting linear

DVB content and non-linear YouTube content. In addition to this the paper discusses the types of collaborative filtering methods that are useful to generate recommendations within a media environment by using a small, medium and large group of users.

The paper is organized as follows: Section 2 presents the related work. Section 3 gives a short overview of the media convergent service environment which is used to evaluate the algorithms that are developed in this study. Section 4 introduces the reader to the newly developed Recommendation Engine. Section 5 discusses the creation process of the user profiles in an explicit and implicit manner for linear DVB content. This creation is realized by novel algorithms, which are based on DVB metadata. In addition it contains the results of the evaluation, which connects DVB genres and subgenres with categories of YouTube. Finally a new equation is derived which is used to combine the linear DVB content with the non-linear YouTube content. Section 6 presents the most commonly available collaborative filtering methods, compares them and proposes the most suitable method to generate recommendations within the presented environment. Section 7 discusses the development of the interface that is responsible for visualizing the generated recommendations for linear DVB and non-linear YouTube within a single application.

2 Related Work

Several previous research papers present the creation of recommendations within a media environment. They have used several different types of filtering methods to generate recommendations. The most common ones are based on the content-based filtering method (see Section 5) [6, 10, 15, 16] and the collaborative filtering technique (see Section 6) [8, 16–18]. In addition to these filtering methods a user profiling is also required. User profiles can be created in an implicit [12, 13] and explicit [12] manner. The explicit settings can be realized by e.g. setting stars, which is quite common to many users. Web portals, like YouTube and so forth use these kinds of ratings. The implicit creation of user profiles is generally realized by logging the viewing behavior (see Section 5.1). However a published work [5] proves the strong correlation between spending time on a single view and the importance of this single view. This supports the proposed approach, as it logs the duration a user spends on watching a specific event.

AVATAR [16] uses Content-Based Filtering methods as well as Collaborative Filtering methods. Recommendation engines, which use these two filtering methods are known as ”hybrid”. Beside these methods, AVATAR uses an own ontology (see Figure 1) to build a TV hierarchy. This implemented ontology is described by the means of OWL (Web Ontology Language). This ontology illustrates a TV content hierarchy. Every ”superclass” has one or more ”classes”. A class is the lowest unit in the hierarchy.

AVATAR uses the *Degree of Interest* (DOI) defined by Blanco et al. [1] for the calculation of the level of interest (see Equation (1)). A matching is calculated by using Equation (2), which considers the DOI and the semantic similarity (see Equation (3)).

$$DOI(C_m) = \frac{DOI(C_{m+1})}{1 + \#sib(C_{m+1})} \tag{1}$$

(C_m) is the superclass of C_{m+1} and $\#sib(C_{m+1})$ represents the number of siblings of the class C_{m+1}.

$$match(a, U) = \frac{1}{\#N_U} \sum_{i=1}^{\#N_U} SemSem(a, c_i) \cdot DOI(c_i) \tag{2}$$

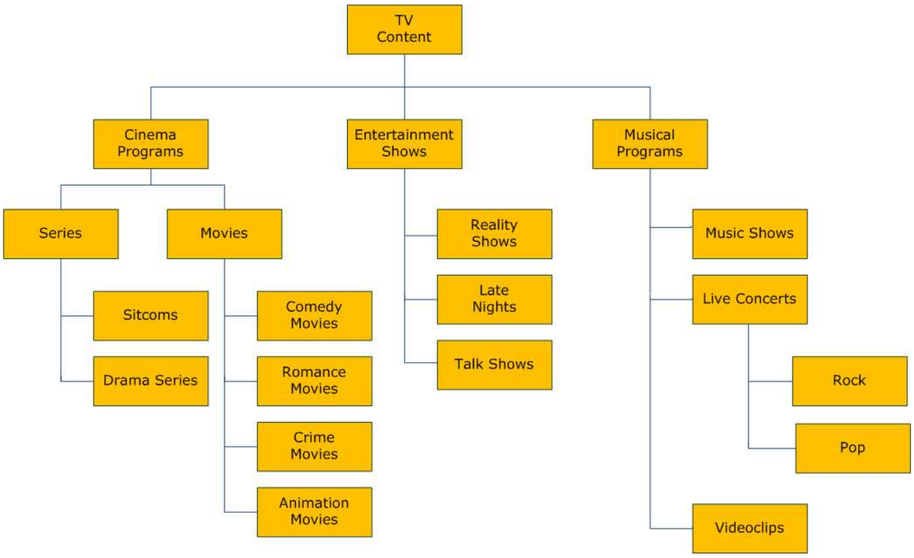

Fig. 1 AVATAR - Ontology [16]

a represents the target content, c_i is the i-th content, which is defined in the ontology profile P_U. $DOI(c_i)$ represents the level of interest of an active user U regarding c_i. $\#N_u$ represents the total number of programs included in P_U.

$SemSem$ is the semantic similarity, which is described by Equation (3). It uses the hierarchical and the inferential similarity, which are combined by means of a factor $\alpha \in [0,1]$.

$$SemSem(a,b) = \alpha \cdot SemSem_{Inf}(a,b) + (1-\alpha) \cdot SemSem_{Hie}(a,b) \qquad (3)$$

However, besides this ontology AVATAR uses *semantic characteristics*, like *hasActor, hasActress, hasTopic, hasTime, hasPlace, etc*. This fact permits the AVATAR system the possibility to infer hidden knowledge in the ontology. Besides these techniques the AVATAR system uses the Pearson-r correlation (see Section 6.3) to calculate the recommendations in a collaborative manner.

Contrary to the AVATAR the presented recommendation engine uses specified genres and subgenres to build the TV hierarchy. This research work exploits the data which are sent within the media stream. Unfortunately the ontology used by the AVATAR system, cannot be built with the Service Information of DVB, which are specified by an ETSI standard. Furthermore attributes, like $hasActor$ and so forth are not sent within the DVB Transport Stream. Therefore the techniques of AVATAR can't be used for achieving the objectives that are proposed in this paper.

Toon De Pessemier et al. [20] uses metadata, like genre, director, keyword, title, actor, coworker, spoken language and caption language to create the recommendations. They use metadata terms t_i, e.g. "soccer", "Antonio Banderas", "violence", etc. Each of these terms belongs to a field $f_i \in \{$*Genre, Actor, Director, Coworker, Keyword, Spoken Language, Title, Caption*$\}$. The author associated each of these terms with the user appreciation u_i, which is in the range [-1,1]. He stores the profile in a form of 3-tuples (t_i, f_i, u_i) in a database.

Furthermore the paper considers the fact that not all metadata information is equally important. For instance a genre may be more significant than a keyword. Therefore the author assigns an important factor W_i for each field f_i.

In this work the 3-tuples are updated in an implicit and explicit manner. The paper uses the time the user spends on watching a video for updating the implicitly logged viewing behavior. The explicitly set preferences get updated by setting ratings. The user appreciation is updated by Equation (4). This equation is used, if there is already a 3-tuples in the profile with the specified term t_i and the field f_i. Otherwise Equation (5) is used.

$$\acute{u} = (1 - \alpha) \cdot u + \alpha \cdot \beta \tag{4}$$

Where $\acute{u}$ represents the new user appreciation of t_i. u stands for the old user appreciation of the term, α is a parameter, which specifies the learning rate and is in a range between 0 and 1. β is in a range [-1,1] and represents the score from the implicit and explicit rating mechanism.

$$\acute{u} = \beta \tag{5}$$

Toon De Pessemier et al. [20] uses these equations for the recommendation algorithm. This algorithm extracts the information from the TV-Anytime metadata of the content item. Then the algorithm checks which terms t_i are available in the user profile. After these steps the algorithm calculates a recommendation score, which is presented in Equation (6).

$$S = \frac{\sum_i u_i \cdot W(f_i)}{\sum_i W(f_i)} \tag{6}$$

u_i represents the user appreciation and $W(f_i)$ represents the important factor of the field f_i.

Figure 2 illustrates the procedure in more detail.

However, the metadata, which are used by Toon De Pessemier et al. [20], are not delivered by the media sources. In order to make the presented recommendation system autarkic, without using proprietary metadata, in this current work the authors use only metadata, which is available for free and delivered by the media source. For instance DVB delivers Service Information and YouTube delivers metadata through an API from Google.

Hopfgartner et al. [3] uses an inverse exponential weighting from Campell et al. [2]. This kind of weighting gives a higher weighting to events which are added recently. Events will be rated with actual interest, so that the users will get recommendations which are fit to their actual likings. However, the inverse exponential weighting does not take an automatic decreasing and other factors, like an episode break into account. For instance a user likes to watch soccer. She/he watches every Saturday soccer until the season will have a break for three month. After three month the season continues. Now the presented equation of Hopfgartner et al. [3] will rate soccer quite low, because the category soccer has not been watched for several months. However, this equation cannot take this behavior into account. In contrast to Hopfgartner et al. the presented equations in Section 5.1.1 use explicit settings and implicitly logged viewing behavior. If a user will not watch an episode over a period of time, the implicitly logged Recommendation Index will be decreased (see Equation 10), but the explicit settings will also be used for the creation of the recommendations (see Equation **??**).

The paper by Badrul Sarwar et al. [8] compares the Cosine Similarity, the Adjusted Cosine Similarity and the Pearson-r correlation. In contrast to the paper by Badrul Sarwar et al. [8], which compares the similarities in an item-based manner, the presented paper compares these algorithms in a user-based manner. In addition the paper by Badrul Sarwar

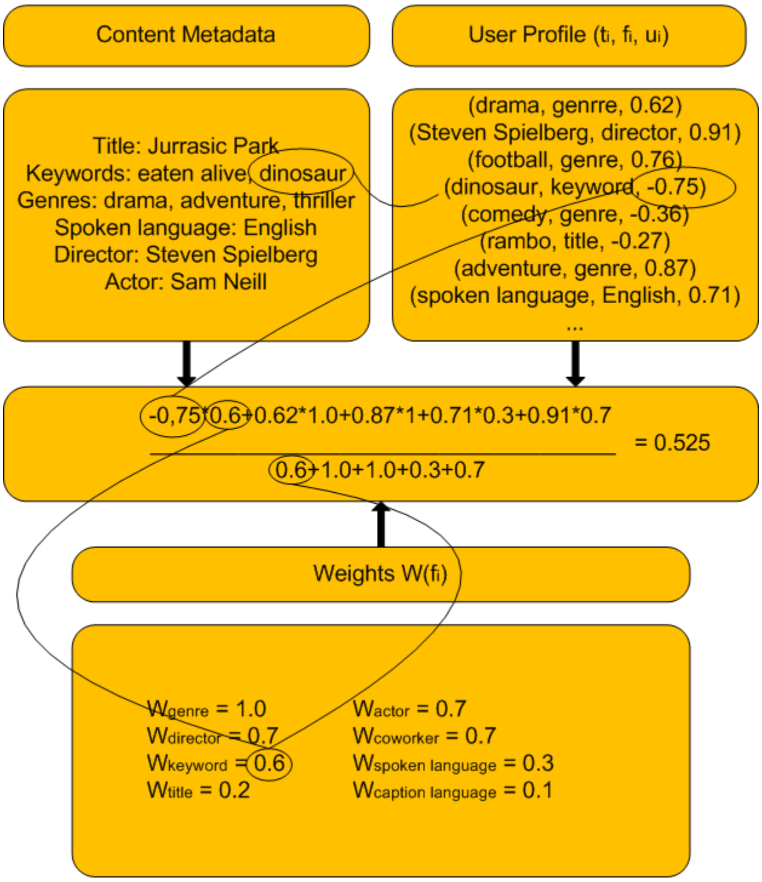

Fig. 2 Recommendation Score - Procedure [20]

et al. does not take DVB genres and ratings of these genres into account. Furthermore the presented paper takes also the *Singular Value Decomposition* into account.

The paper by L. Ardissono et al. [19] presents an interface (see Figure 3), which is able to present recommendations for TV content. It visualizes the start-time and date, a category, the title, the channel and the recommendation. The recommendations are illustrated by smilies. Five smilies represents full interest and zero smilies no interest. The interface also presents half smilies. Therefore eleven different graduations are possible. The paper uses TV content. To the best of the authors knowledge an interface, which is able to visualize recommendations for linear as well as non-linear media content within a single view, has never been discussed before in the literature. The interface presented in this paper is able to visualize recommendations for linear DVB content (TV content) as well as non-linear web content (YouTube videos).

Generally, the current approaches [6,10,12,13,15,16,19,20] take only one media source into account. However, the presented approach here uses a media convergent service environment. In this study linear DVB content as well as non-linear web videos from YouTube are considered. Furthermore specified genres and subgenres to compare the most adequate collaborative filtering methods are used.

Fig. 3 Recommendation Interface [19]

3 Media Convergent Service Environment

Today users have access to several media services like linear DVB content and non-linear video portals. Most of those people use these different media services with different applications or devices. For instance DVB content is consumed using set-top-boxes and video portals like YouTube are consumed using a computer. In an in-house scenario users mostly use more than one set-top-box or computer. The presented environment combines the different media sources in a multiuser interface. User differentiation is quite useful because e.g. if a user does not use his own computer and want to have the full access to her/his individual data, the proposed scheme will be able to manage this. Figure 4 illustrates a home environment where the proposed solution can be exploited. The controller in this scenario is the intelligent router. It manages all the information and saves the entire user profiles (see Section 5), which are created in implicit and explicit manner. Users are able to login and get access to their individual data (like movie storage and so forth) as well as their individual recommendations. This router is also responsible for offering access to Internet services, such as YouTube. The Recommendation Engine (see Section 4) and the respective algorithms are installed on each client. The XML file, which contains the user profiles (see Listing 2 and Listing 3) are stored on the Flash ROM of the router. If the dataset is larger than the capacity of the Flash ROM, the router can use an external hard disk, which can be plugged through a USB link. The updating of the user profiles can easily be realized by PHP (hypertext preprocessor). A major advantage of the intelligent router is the operating system, which allows execution of C code. In addition the router is also responsible for QoS

(Quality of Service) tasks. These QoS tasks take care of the distribution of media content and reserves bandwidth for it. A further advantage is the constant availability of data (like the user profiles). A user could theoretically use the data even she/he is not in the house. E.g. an application for a smartphone could use these data to recommend content, which is delivered by YouTube and so forth.

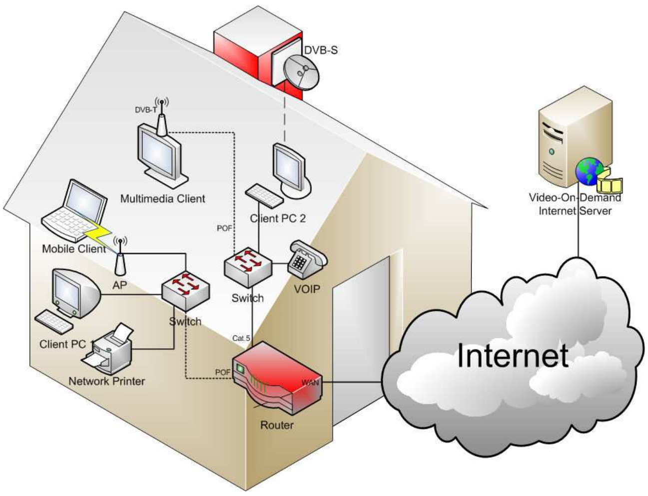

Fig. 4 HomeVision - Media convergent service environment

4 Recommendation Engine

The creation of recommendations within the present environment is realized using a Recommendation Engine. This developed Recommendation Engine is presented in Figure 6. This figure shows which parts are included and realized by the Recommendation Engine. The Recommendation Engine includes the algorithms, which are responsible for creating the user profiles in an implicit manner (see Section 5.1.1). Besides the implicit user profile, the Recommendation Engine governs the explicitly set preferences of individual users (see Section 5.1.1). The DVB parser is responsible to extract metadata from a DVB Transport Stream. These metadata are specified by an ETSI standard for DVB Service Information. The DVB parser extracts:

- title of events
- genres
- subgenres
- start- and end time of events
- start date of events

These metadata are needed to create the implicit user profile. The equations, which are responsible for the creation of the implicit user profile are presented in Section 5.1.1.

The YouTube parser extracts the metadata from YouTube through an API from Google. The Recommendation Generator creates the recommendations for DVB content by considering the implicitly logged viewing behavior and the explicit settings (see Section 5.1.2).

Furthermore the Recommendation Generator generates the recommendations for YouTube content (see Section 5.2). In addition the Recommendation Engine is responsible for storing the implicitly logged viewing behavior in a XML file. It extracts the Service Information, like title of an event, the genre and the subgenre of an event, etc of a DVB Transport Stream. In addition it logs the duration a user watches an event respectively the timespan and also information about what kind of genre or subgenre is watched by the user. This data is logged by a thread, which also stores these data in a XML file (see Listing 3). The following pseudo code shall clarify the logging of a watched genre.

```

string genreNibbleStart, genreNibbleCheck;
DateTime actualTime, checkTime;

// get the EIT (Event Information Table)
getEIT();

// get the nibbles of a genre, which represents the genre, subgenre
genreNibbleStart = getGenreNibbles();

// get the actual date and time
actualTime = getActualDateTime();

// start a thread, which will be called every 10 seconds
startThread(thread);

// the method, which will be called every 10 seconds
thread()
{
        // get the EIT
        getEIT();

        // get the nibbles of genre, which represents the genre, subgenre
        genreNibbleCheck = getGenreNibble();

        // if the "old" genre is not the current genre
        if(genreNibbleStart != genreNibbleCheck)
        {
                // get the actual date and time
                checkTime = getActualDateTime();

                // calculate the timespan between the two times
                calculateTimeSpan(actualTime, checkTime);

                // calculate the Recommendation Index
                calculateRI();

                // add the RI to the XML
                addRItoXML();

                // reset timestamps
                actualTime = checkTime;

                // reset the genre nibbles
                genreNibbleStart = genreNibbleCheck;
                genreNibbleCheck = "";
        }
}
```

Listing 1 ”Pseudo Code - Logging Genre”

Figure 5 shows the structogram of the pseudo code, which is presented in Listing 1.

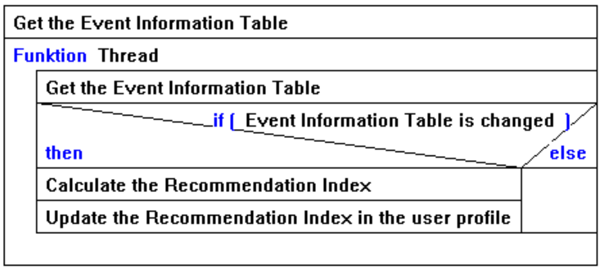

Fig. 5 Logging - Structogram

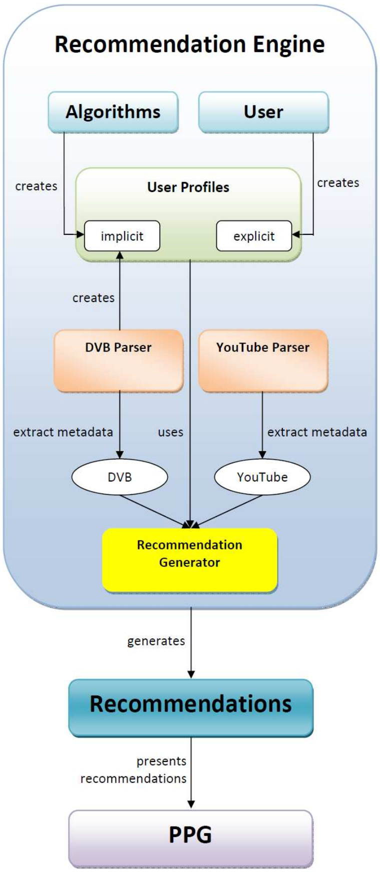

Fig. 6 Recommendation Engine

5 Content-Based Filtering

Filtering methods are required to generate recommendations. The most common filtering methods are Content-Based Filtering [6, 10, 15, 16] and Collaborative Filtering (see Section 6). The Content-Based Filtering method detects similarities of objects by their attributes. An object is described by content and gets assigned by attributes. If a user requests a recommendation, the system searches objects with similar or equal attributes. In this paper recommendations for media content are generated by the Recommendation Engine. Recommendations for Digital Video Broadcast content will be generated by using the implicitly and explicitly created user profiles. The different kinds of user profiles as well as newly developed equations, which use the Content-Based Filtering method are presented in the following sections. Furthermore, a connection between DVB genres and YouTube categories is shown. This connection allows the generation of recommendations for YouTube videos based on a calculated Recommendation Index for DVB content.

5.1 Recommendation Index - DVB

Recommendations are based on explicitly set preferences or implicitly logged viewing behavior. User profiles can be used to save these data. An example of an XML file, which contains explicitly set preferences, as well as implicitly logged viewing behavior, is shown by Listing 2 and Listing 3. The user profile is split into two XML files, because the implicit profile grows in time. This procedure decreases the time for searching and updating the user profile. The developed Recommendation Engine uses DVB Service Information, which are sent within a DVB Transport Stream. These Service Information are specified by an ETSI standard. Each event is described by a title, genre, subgenre, start- and end time, date of broadcasting, and so forth. These metadata are used to build the user profiles in an explicit and implicit manner.

```xml

<user name="chris">
 <favouritemaingenres id="explicit">
  <genre mgenre="0x10" mRI="1"/>
  ...
 </favouritemaingenres>

 <favouritesubgenres id="explicit">
  <genre sgenre="0x11" sRI="1"/>
  ...
 </favouritesubgenres>

 <favouriteevents id="explicit">
  <event name="nano" eRI="1"></event>
  ...
 </favouriteevents>
</user>
```

Listing 2 ”User Profile XML - explicit”

```xml

<user name="chris">
 <favouritemaingenres id="implicit">
  <genre mgenre="0x20" mRI="0,42"/>
  ...
 </favouritemaingenres>

 <favouritesubgenres id="implicit">
  <genre sgenre="0x20" sRI="0,21"/>
  ...
 </favouritesubgenres>

 <favouriteevents id="implicit">
  <event name="tagesschau" eRI="0,43567"></event>
  ...
 </favouriteevents>
```

```
17 </user>
```

Listing 3 "User Profile XML - implicit"

$mgenre$ represents a maingenre, $sgenre$ a subgenre. mRI is the Recommendation Index (see Section 5.1.1) of a maingenre, sRI of a subgenre and eRI the Recommendation Index of an event.

5.1.1 User Profiling

Explicit Profiling The Explicit Profiling is done by the users [9, 12]. Within a newly developed application, users are able to set the individual preferences explicitly. The explicit setting shall help to customize the system in an individual manner. The system shall know, which kinds of events are favored. Generally, DVB events are enriched with Service Information. Each DVB Transport Stream contains these information, which are specified by an ETSI standard [14]. Data, like title of an event, the genre, the subgenre, start- and end time, date of publishing, and so forth are sent within a DVB Transport Stream. ETSI specifies twelve genres:

- movie/drama
- news/current affairs
- show/game show
- sports
- children's/youth programmes
- music/ballet/dance
- arts/culture
- social/political issues/economics
- education
- leisure hobbies
- other
- undefined content

Each genre is split into several subgenres, which classifies the genres in more detail. Table 1 presents all subgenres of the genre 'movie/drama'.

Table 1 subgenres: movie/drama

Genre	subgenre
movie/drama	movie/drama (general)
	detective, thriller
	adventure, western, war
	science-fiction, fantasy, horror
	comedy
	soap, melodrama
	romance
	serious, classical, religious, historical
	adult movie/drama

In order to use the DVB Service Information to generate recommendations, our novel application offers an interface to set favorite genres, subgenres and events. Users are able to rate them by setting stars. 0 stars represent no interest in the selected item and 5 stars represent full interest in the selected item. Figure 7 shows a screenshot of the developed application.

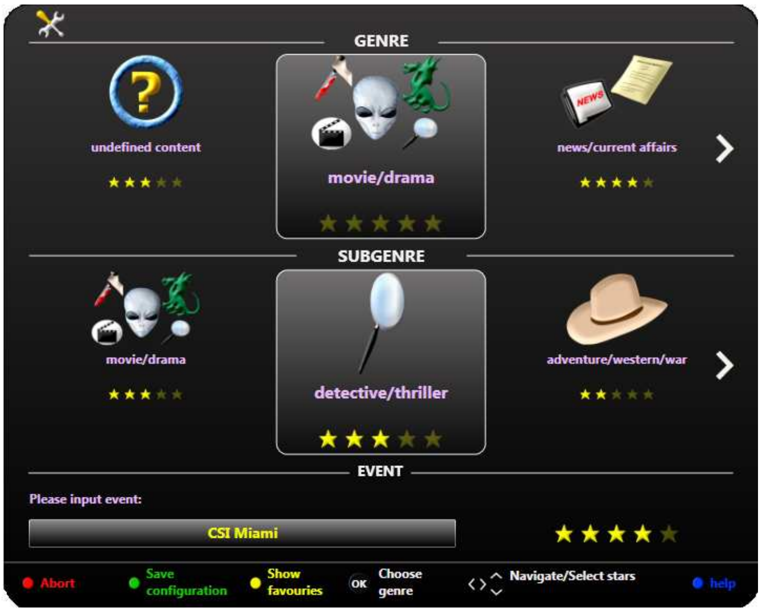

Fig. 7 PPG - Explicit Settings

The Recommendation Index (RI) is in range [0;1] ($RI_{explicit} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$), while 0 is represented by 0 stars, 0.2 by 1 star, 0.4 by 2 stars, 0.6 by 3 stars, 0.8 by 4 stars and 1 by 5 stars. After the user set her/his preferences, the result (RI) is saved into a XML file (see Listing 2), which contains the user profiles of all users.

Implicit Profiling is realized by logging the viewing behavior of users [9, 10, 13]. This implicitly logged viewing behavior enriches the explicitly set preferences. It also takes the real behavior into account. For instance if a user does not set the genre ”education” in an explicit manner, but he/she watches this genre quite often, the implicit profiling will recognize it and add this genre to the user profile. Furthermore published research [5] proves the strong correlation between spending time on a single view and the importance of this single view.

The following developed equations are responsible for logging the viewing behavior. They use the sent DVB Service Information to generate a Recommendation Index (RI), which is in range [0;1], where 0 represents no interest and 1 represents full interest.

Equation (7) is responsible for the calculation of a Recommendation Index for a genre, a subgenre or an event.

$$RI_{implicit} = \sum_{i=1}^{\infty} \frac{t_w(i)}{t_d} \tag{7}$$

$RI_{implicit} = [0; 1]$, $t_w = [0; t_d]$, $t_d \geq 0$

Expression t_d represents the duration the user watched television.
The $RI_{implicit}$ is the value of the Recommendation Index from a genre, subgenre or an event.
The variable $t_w(i)$ represents the duration the user watched the genre, subgenre or an event, where i is a counter.

With this counter the equation is able to calculate the Recommendation Index of the watched genre, subgenre or event over a period of time (t_d). Therefore, the Recommendation Engine is able to sum the times, the user watches a genre, a subgenre or an event. For instance, if the user watches a genre, like movie/drama and she/he switches the channel during a commercial break and switches back after a period of time, the system will sum the duration of watching before the commercial break and the duration of time after it.

With the intention of guaranteeing that the value of the Recommendation Index becomes more and more accurate Equation (8) is used to calculate the average. Each Recommendation Index of a particular genre, subgenre or event is represented by $RI_{implicit}(k)$. The variable n is the counter of the measurements.

$$\overline{RI}_{average} = \frac{1}{n}\sum_{k=1}^{n} RI_{implicit}(k) \tag{8}$$

The $\overline{RI}_{average}$ is the average value of the Recommendation Index of a genre, subgenre or an event.
The expression $RI_{implicit}(k)$ is the value of the recommendation index of one genre, subgenre or event. The variable (k) is the counter of this genre, subgenre or event.
Example: The user watched the genre movie/drama on Monday. This results to a Recommendation Index of *0.6*. On Saturday the user watched the genre movie/drama again and the Recommendation Index for this day is *0.8*. In this case n=2. The average of these RIs is *0.7* (see Equation (9)).

$$\overline{RI}_{average} = \frac{0.6 + 0.8}{2} = 0.7 \tag{9}$$

This procedure guarantees the logging of the real viewing behavior, because the calculation of the Recommendation Index is based on several RIs. E.g. if a user likes to watch a special soap and he/she misses the the first twenty minutes of the first episode, which takes sixty minutes in total, the RI would be 0.6666. But if the user will watch the next episode for sixty minutes the RI will increase to a higher level. This procedure will also work the other way round. E.g. if a user does not like a particular genre, but the television is on while the user takes a phone call, the Recommendation Index would be calculated. But if the user will watch this genre again and will not spend much time on watching, the Recommendation Index will be decreased.

Equation (7) and Equation (8) are basically responsible for creating the Recommendation Indexes for events, genres and subgenres in an implicit manner. These equations have one main problem. If a user watches an event, a genre or a subgenre and e.g. the calculated RI is one and if the user never watches this event again, the RI will always be one.

Equation (10) has been developed to overcome this problem. This equation decreases the RI step by step over time.

$$RI_{adjust} = \overline{RI}_{average} \cdot e^{-(\frac{1}{4})} \tag{10}$$

Figure 8 shows the recommendation index adjustment. The value of the RI decreases every week. After eight weeks the RI is under 0.15, which will be rounded to zero. This guarantees that the RI will be decreased if a user never watches an event, a genre or a subgenre again.

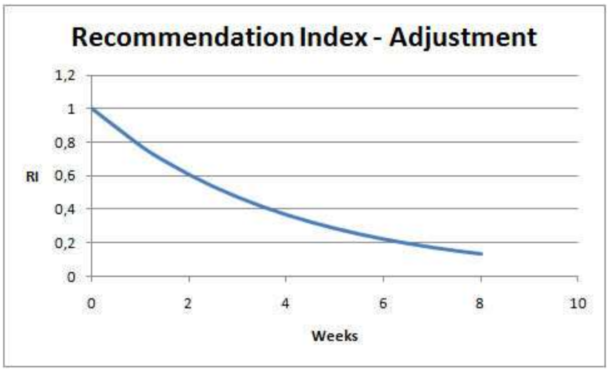

Fig. 8 Recommendation Index - Adjustment

5.1.2 Recommendation Index - Mix

In order to combine events, genres and subgenres as well as the implicitly and explicitly created user profiles, Equation (11) has been developed. This combination is needed, since users don't want to select, if the implicitly logged or the explicitly set preferences shall be shown. This is proven by own accompished study. Within this evaluation users were asked to fill a questionnaire. They had to give a feedback, based on if they would prefer recommendations, which are generated on implicitly logged viewing behavior or explicitly set preferences. However, they had to rank the two possibilities by setting a factor. Results of a realized evaluation have shown, that users prefer the explicit settings more than the implicitly logged viewing behavior. Equation (11) takes these results into account and multiplies the explicit settings with a factor of two.

$$RI_{mix} = \frac{RI_{adjust} + RI_{explicit} \cdot 2}{3} \tag{11}$$

5.1.3 Recommendation Index - Final

Each event is described with a title, genre and a subgenre. The title is the most significant value. Each genre is split into several subgenres which describe the genre in more detail. E.g. the genre movie/drama is split into nine subgenres (see Table 1). With the purpose of taking these factors into consideration a scenario has been developed, which is presented in the following paragraphs. Following boosting constants are the results of an evaluation. The users were asked which of the parameters has the highest priority, which has the second highest priority and which has the lowest priority. Details of the users which participate on the evaluation are shown in Section 7.5. The following equations are derived on the outcome of this study, which shows that the title of the event is the most important parameter and the subgenre has a higher priority than the genre of an event.

If an event from the scheduled information as well as the genre and the subgenre of this event is part of the user profile Equation (12) is used and the RI is defined as:

$$RI = \frac{RI_{event} \cdot 3 + RI_{subgenre} \cdot 2 + RI_{genre}}{6} \tag{12}$$

If an event is in the scheduled information and the subgenre is not part of the user profile, Equation (13) can be used to calculate the RI as:

$$RI = \frac{RI_{event} \cdot 3 + RI_{genre}}{4} \tag{13}$$

If only the title of the event is in the current user profile, Equation (14) can be used to calculate the RI as:

$$RI = RI_{event} \tag{14}$$

If the scheduled information's event cannot be found, but the subgenre's event is part of the user profile, then the Equation (15) can be used to calculate the RI as:

$$RI = \frac{RI_{subgenre} \cdot 2 + RI_{genre}}{3} \tag{15}$$

If only the event's genre of the scheduled information is part of the user profile, Equation (16) calculates the RI for this event as.

$$RI = RI_{genre} \tag{16}$$

These equations guarantee that all factors, like title of an event, the genre and the subgenre, are taken into account to generate a Recommendation Index for events.

The results of the explicitly set preferences as well as the implicitly logged viewing behavior are saved in a XML file (see Listing 1, Listing 2 and Listing 3).

5.1.4 Evaluation of the Recommendation Index - DVB

The system was tested by a group of twelve users. Details on the background of the users are presented in Section 7.5. The outcome of this evaluation proved the usefulness of this approach. The users could check whether the recommendations are consistent with their preferences, almost match or do not match their preferences. $83\frac{1}{3}\%$ of the generated recommendations match with the preferences of the users. $12\frac{2}{3}\%$ of the generated recommendations almost match to user's preferences. 4% of the generated recommendations do not match to the preferences of the users.

These results can be explained by the quality of the delivered metadata. The system uses metadata, which is delivered by DVB Transport Stream. In this case the system must trust these metadata. However, some providers send quite general information with an event. For instance a comic is specified as a movie/drama. In this case the system would recommend this comic to users, which prefer movie/drama.

5.2 Recommendation Index - YouTube

With the aim of combining linear DVB content with non-linear YouTube content, a similarity between these two content sources is required. The genres and subgenres of DVB, which are specified in an ETSI standard and categories of YouTube are quite similar. For instance, DVB contains a genre named ”Comedy” and YouTube offers this category too.

It has carried out an evaluation in which the respondents should combine the DVB genres and subgenres with the categories of YouTube. The questionnaire included all genres and subgenres of DVB and all categories of YouTube. Each respondent had to decide, which genre/subgenre of DVB is quite similar to a category of YouTube. The respondents were allowed to mark more than one similarity. For instance, they were allowed to connect the YouTube category ”Travel & Events” with the DVB subgenre ”foreign countries/expeditions” and the DVB subgenre ”tourism/travel”. Figure 9 presents a snippet of the questionnaire.

No.	*Genres*	Autos & Vehicles	Comedy	Education	Entertainment	Film & Animation	Gaming	Howto & Style	Music	News & Politics	People & Blogs	Pets & Animals	Science & Technology	Sports	Travel & Events
0	**undefined content**														
1	**movie/drama**														
1.1	movie/drama (general)														
1.2	detective/thriller														
1.3	adventure/western/war														
1.4	science fiction/fantasy/horror														
1.5	comedy														
1.6	soap/melodrama														
1.7	romance														
1.8	serious/classical/religious/historical														
1.9	adult movie														
2	**news/current affairs**														
2.1	news/current affairs (general)														
2.2	news/wheather report														
2.3	news magazine														
2.4	documentary														
2.5	discussion/interview/debat														
3	**show/game show**														
3.1	show/game show (general)														
3.2	game show/quiz/contest														
3.3	variety show														
3.4	talk show														

Fig. 9 YouTube - Evaluation

Table 2 presents the results of the evaluation, which have a similarity at a minimum of 70%. The percentages for the similarities result from the number of marks that have made the respondents. The questionnaire was distributed in the University of Applied Sciences Giessen-Friedberg and distributed to employees of this university. Most respondents were thus students, staff and members of university staff. In total 104 participants filled the questionnaire. Out of which 54 of them were male and the remaining females. These results are used to generate recommendation for YouTube videos. These recommendations are based on Equation (11). Due to the fact that the evaluation's results are based on DVB genres, subgenres and categories of YouTube, the new equation has to take these parameters into account.

In order to generate recommendations for YouTube videos based on DVB genres and subgenres, the following equation has been developed.

$$RI_{YouTube} = \frac{RI \cdot similarity}{100} \tag{17}$$

This equation combines the RI of a genre or a subgenre, which is based on explicitly set preferences as well as implicitly logged viewing behavior, with the results of this evaluation. The result is a $RI_{YouTube}$, which ranks YouTube videos in the range [0;1].

6 Collaborative Filtering

In contrast to Content-Based Filtering, Collaborative Filtering [8, 17, 18] does not use information, like attributes of objects, to generate recommendations. This kind of filtering mechanism puts the similarity of users in the center of attention. Basically, the finding of similarities can be realized by item-based or user-based methods. This paper puts the focus on *user-based* similarities. The system explores existing user profiles and searches for similar user profiles on the basis of the user, which requests the recommendations. After the system found similar user profiles, it recommends contents, which are part of similar user profiles.

The searching for similar user profiles can be realized by using several algorithms. The following sections will describe some of the researched algorithms and the usefulness of them by taking DVB genres and subgenres into account. Table 3 lists all genre, which are specified by ETSI and part of the developed Recommendation Engine. Table 4 lists all ratings of five users. This table has been filled by random values. With the aim of finding similarities between the different users, this paper compares the most common techniques like *Cosine Similarity*, the *Bravais-Pearson* or *Pearson-r- correlation*, the *Adjusted Cosine Similarity* and the *Singular Value Decomposition*, which are described in the following sections.

6.1 Cosine Similarity

The Cosine Similarity [8] computes the cosine of the angles between two vectors (see Equation (18), [7]). The expression $\mathbf{i} \cdot \mathbf{j}$ denotes the dot-product of two vectors (in our case users). The results is in the range [0;1], while zero represents no similarity and one full similarity between the angles of two vectors. This technique finds the cosine of angles between two vectors. Table 8 contains the results of the calculation. It shows that User 1 and User 5 are quite similar.

$$sim(i,j) = cos(\mathbf{i},\mathbf{j}) = \frac{\mathbf{i} \cdot \mathbf{j}}{|\mathbf{i}|\,|\mathbf{j}|} \tag{18}$$

Table 2 Similarity YouTube - DVB

YouTube-Category	DVB-Genre	Similarity in %
Sports	sports (general)	96,15
	special event	76,92
	sport magazine	100
	football, soccer	100
	tennis, squash	100
	team sport (excluding football)	100
	athletics	100
	motor sport	96,15
	water sports	100
	winter sports	100
	equestrian	80,77
	martial sports	88,46
Travel & Events	tourism, travel	80.77
Autos&Vehicles	motoring	73,08
Comedy	comedy	84,62
Education	informational, educational, school	80,77
	informational, educational, school	80,77
	education, science, factual (general)	73,08
	further education	92,31
	languages	76,92
Entertainment	entertainment programmes for 10 to 16	73,08
Film, Animation	movie / drama (general)	73,08
	science fiction, fantasy, horror	76,92
Music	music, ballet, dance (general)	80,77
	rock, pop	92,31
	serious music,classical music	92,31
	folk,traditional music	92,31
	jazz	92,31
	musical, opera	80.77
	ballet	76,92
News & Politics	news, current affairs (general)	92,31
	news, weather report	92,31
	news magazine	84,62
	discussion, interview, debate	76,92
Science & Technology	technology, natural sciences	73,08

Table 3 Genre Names

Genre No.	Genre
1	movie/drama
2	news/current affairs
3	show/game show
4	sports
5	children's/youth programmes
6	music/ballet/dance
7	arts/culture
8	social/political issues/economics
9	education
10	leisure hobbies
11	other
12	undefined content

Table 4 Genre Rating Table

Genre No.	User 1	User 2	User 3	User 4	User 5
1	100	20	0	40	100
2	80	60	100	60	80
3	60	80	20	80	60
4	100	100	0	100	40
5	40	60	100	40	20
6	20	80	80	60	0
7	60	0	20	80	100
8	80	0	40	100	80
9	100	20	60	20	60
10	60	40	100	40	40
11	0	100	80	60	20
12	20	40	60	80	0

6.2 Adjusted Cosine Similarity

$$sim(i,j) = \frac{\sum_{\mathrm{g}\in\mathrm{G}}(R_{g,i}-\overline{R}_g)(R_{g,j}-\overline{R}_g)}{\sqrt{\sum_{\mathrm{g}\in\mathrm{G}}(R_{g,i}-\overline{R}_g)^2}\sqrt{\sum_{\mathrm{g}\in\mathrm{G}}(R_{g,j}-\overline{R}_g)^2}} \tag{19}$$

In our case (user-based), the $R_{g,i}$ is the rating of a user *(*i) on an item (genre) g. $\overline{R}_g$ is the average of all ratings from all users for genre g (see Equation (19)).

The main difference between *Cosine Similarity* and *Adjusted Cosine Similarity* is that this technique takes the average of the ratings from users into account [8]. Table 8 shows the results of the computation. In contrast to Cosine Similarity the range of the results is [-1;1]. The results show that the similarity between User1 and User5 is quite high.

6.3 Pearson-r correlation

The Correlation-Based Similarity [8], *Bravais-Pearson* correlation or $Person-r$ correlation [16] computes co-rated items. This method only uses items, which have been ranked by users (see Equation (20), [8]). The results are in the range [-1,1], while -1 represents full negative similarity, 1 full positive similarity and 0 no similarity.

$$sim(i,j) = corr_{i,j} = \frac{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,i} - \overline{R}_i)^2} \sqrt{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,j} - \overline{R}_j)^2}} \tag{20}$$

In our case (user-based computation) the $R_{u,i}$ is the rating of the user *u* on the item *i* (genre) and $\overline{R}_i$ is the average of ratings from all genres (*i*) of the user *u*.

Table 8 lists computation's results. They show that the similarity between User1 and User5 is quite high.

6.4 Singular Value Decomposition

The *Singular Value Decomposition (SVD)*, also known as *Latent Semantic Indexing (LSI)*, dimensionally reduction or projection, is also one of the most used methods to realize *Collaborative Filtering*. This method reduces the number of dimensions of a matrix [18]. In this paper the SVD is used to find clusters of users, which have the same set of interests. Table 4 shows genres, which are rated by users. Equation (21) describes the SVD.

$$A_{mn} = U_{mm} \cdot S_{mn} \cdot V_{nn}^T \tag{21}$$

m is the number of genres and n is the number of users. A_{mn} is a $m \times n$ matrix. The entries of this matrix come from a field (in our case: Table 4). U_{mm} is a $m \times m$ unitary matrix over the field. S_{mn} is a $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal, which are also known as singular values of A_{mn}. These singular values are ordered in descending order. V_{nn}^T is the conjugate transpose of V, which is a $n \times n$ unitary matrix over the field.

A snippet of the results is shown in Table 5, Table 6 and Table 7.

These tables only show the first two columns of the results, which are needed to reduce the number of dimensions of the matrix (see Table 4). Table 5 represents the genres (item-based similarity) and Table 7 represents the users (user-based similarity).

Figure 10 shows the results of the *Singular Value Decomposition*. Furthermore, the Figure 10 shows that the similarity between User1 and User5 as well as the similarity between User2 and User3 is quite high. But in contrast to *Cosine Similarity*, *Adjusted Cosine Similarity* and *Pearson-r correlation*, the most significant similarity is between User2 and User3. By reducing the number of dimensions, SVD also loses data. These data are necessary for finding similar users. Due to this fact the *SVD* method was not taken into account in later comparisons.

Table 5 Singular Value Decomposition - U

-0,2700	-0,4607
-0,3839	0,0468
-0,3067	-0,0386
-0,3547	-0,0841
-0,2597	0,3078
-0,2403	0,4026
-0,2721	-0,3590
-0,3180	-0,2897
-0,2672	-0,1688
-0,2829	0,1530
-0,2546	0,4457
-0,2079	0,2384

Table 6 Singular Value Decomposition - S

438,8554	0	0
0	185,3858	0
0	0	122,3507

Table 7 Singular Value Decomposition - V

-0,4931	-0,4333
-0,3958	0,4834
-0,4217	0,5570
-0,5031	0,0012
-0,4115	-0,5179

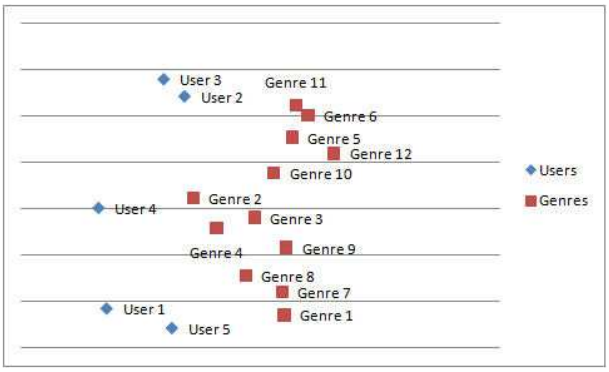

Fig. 10 SVD - Results

6.5 Comparison of similarity methods

With the purpose of finding the best method to generate recommendations, a comparison between the different methods is essential. Table 3 shows all genres specified for DVB content. Table 4 lists the rankings of five users, which has been filled by random values. Each genre is rated by explicit settings and implicitly logged viewing behavior. Figure 11 shows a diagram with the results of the presented techniques, which are based on the results shown in Table 8. The axes of the diagram are adjusted to the values to the different ranges of the three techniques so that they can be easily compared. This is needed because the ranges of

Table 8 Results - Similarities

Connected Users	Cosine Similarity	Adjusted Cosine	Pearson
User1-User2	0,62048368	-0,617506419	-0,3884493
User1-User3	0,59811591	-0,600993663	-0,5015504
User1-User4	0,81297068	-0,184029167	-0,0420579
User1-User5	0,91863818	0,606581208	0,7171372
User2-User3	0,74140928	0,191635261	0,1731795
User2-User4	0,79540843	0,015859536	0,1206452
User2-User5	0,49090909	-0,74056031	-0,6000000
User3-User4	0,68424658	-0,400318015	-0,4687592
User3-User5	0,53314824	-0,54014375	-0,4928945
User4-User5	0,77917561	-0,151894678	0,0402151

the techniques differ. The results of the *Pearson-r correlation* and *Adjusted Cosine Similarity* uses the left y-axis and the results of the *Cosine Similarity* uses the right y-axis. The x-axis represents the comparison between the different users.

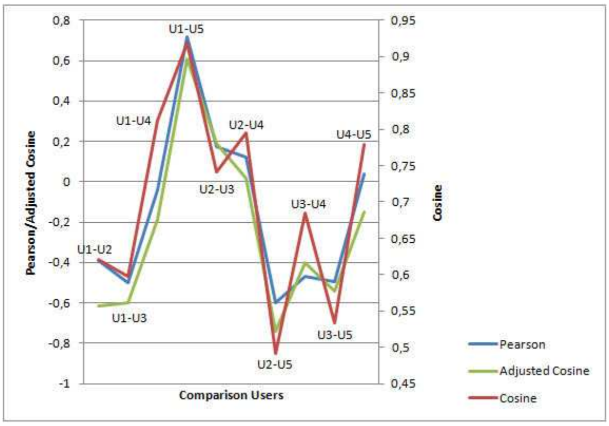

Fig. 11 Comparison Similarities

The results show that each technique is useful to generate recommendations. The most significant values are the same. The comparison between the different techniques shows that each of the presented algorithms are adapted to find similar users. Furthermore the results show that the *Pearson-r correlation* as well as the *Adjusted Cosine Similarity* reduces outlier (see U3-U4 in Figure 11).

6.5.1 Prediction

The comparing of the similarities between these techniques is realized by a computation of predictions. This paper presents one of the most common ones the *Weighted Sum*. Equation (22) [8] shows how the Weighted Sum is computed.

$$P_{a,i} = \overline{r}_a + \frac{\sum_{\mathrm{u} \in \mathrm{U}} (r_{u,i} - \overline{r}_u) \cdot sim_{a,u}}{\sum_{\mathrm{u} \in \mathrm{U}} |sim_{a,u}|} \tag{22}$$

$P_{a,i}$ represents the prediction of the active user a for the item (in our example a genre) i. $\overline{r}_a$ represents the average of the ratings of the active user a. $r_{u,i}$ is the rating of the user u for the item i. $\overline{r}_i$ represents the average of the ratings from user u without the rating of the item i. $sim_{a,u}$ represents the similarity between the active user a and the user u. In order to calculate a prediction a value of an item must be deleted from the rating table. This prediction is needed to calculate the Mean Absolute Error (MAE).

6.5.2 Mean Absolute Error

With the intention of classifying the results of the predictions the *MAE* is used (see Equation (23), [11]). This *MAE* shall prove, which method is the most adequate one to generate recommendations within the presented environment.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{23}$$

p_i is the prediction and q_i is the true value.

The results based on *MAE* are shown in Figure 12.

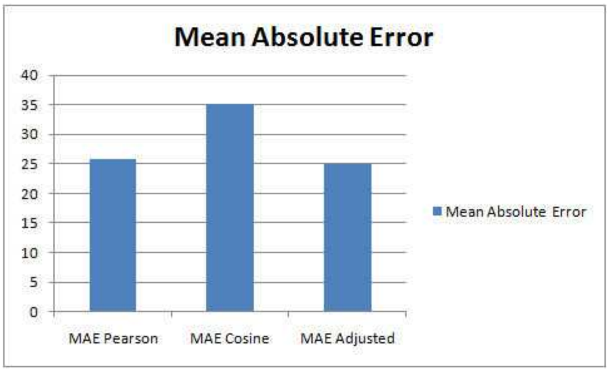

Fig. 12 Mean Absolute Error

6.5.3 Results

Figure 12 shows the results of the calculated *MAE*. The *MAE* of *Cosine Similarity* is significantly higher than the *MAE* of *Adjusted Cosine* as well as the *Pearson-r correlation*. These results prove that the Adjusted Cosine and the Pearson-r correlation are useful to find user-based recommendations by using ratings of DVB genres, which are specified by the ETSI standard for Service Information.

6.5.4 Further Results

Besides this limited test with five users, we also accomplished further tests with twelve users. The values of the user-genre matrix are an output of a questionnaire. Users had to rate each genre by setting a value between 0 and 100 [0, 20, 40, 60, 80, 100]. Figure 13 shows the results of the similarities between the twelve users. The Mean Absolute Error (MAE) of

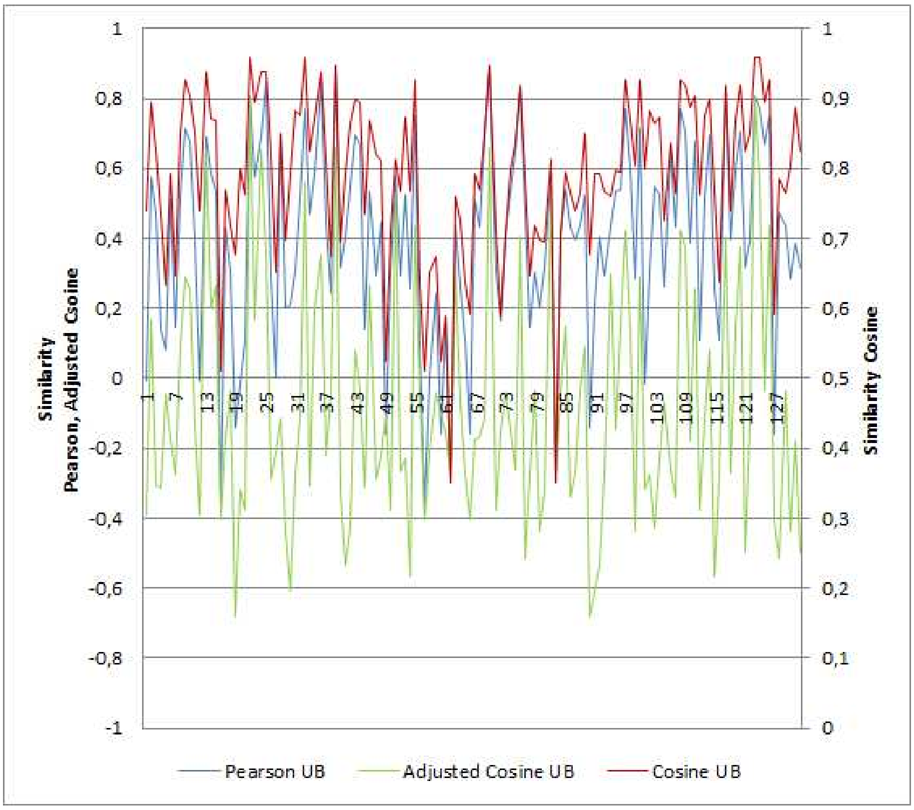

Fig. 13 Comparison Similarities - twelve users

this test is illustrated in Figure 14.

In addition simulations with 5, 10, 15, 20, 30, 50, 100, 150, 200, 500 and 1000 have been accomplished. The simulations present results by using more than twelve users. Furthermore the using of more users shall give a feedback of the behavior of the different algorithms by using twelve genres. These simulations were realized by a software, which fills an array with ratings for the twelve specified genres. Each entry within this array was filled by a random value [0, 20, 40, 60, 80, 100]. Figure 15 shows the MAE of the accomplished simulations. The figure makes a strong connection between the MAE and the users clear. The more the users are part of the system, the lower is the MAE. If the number of user is higher than 100, the MAE is quite static. Besides these results the simulations show, that the fluctuation of the calculated MAE decreases, if the number of the users increases.

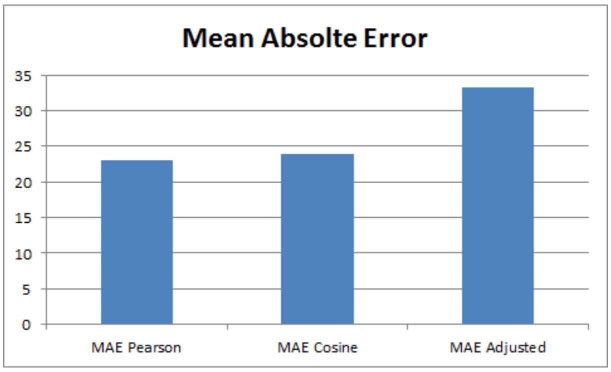

Fig. 14 Mean Absolute Error - twelve users

The results of Section 6.5.2 (see Figure 12), Section 6.5.3 (see Figure 14 and this Section shows the strong correlation between the fluctuation of the MAE and the number of users. The higher the number of users, the lower the fluctuations in the calculation of the MAE. This can be explained by the calculation of the predictions. The predictions use the similarity values of the used algorithm. If only a few users are used to calculate the predictions, only a few similarities can be used. The results also show that the Pearson-r correlation seems to perform well in the tests with five and twelve users. The simulations show that the Adjusted Cosine Similarity performs well by considering a higher number of users.

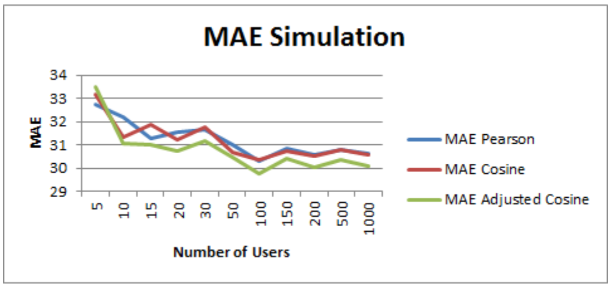

Fig. 15 Mean Absolute Error - Simulation

7 Personal Program Guide - PPG

7.1 Introduction

Since a user has access to several kind of media sources and an immense number of content, the user could be overloaded with information. Furthermore users could have a problem to find contents of interest. Recommendations can help users to find content of interest within the media convergent service environment in less time. In order to present the generated recommendations to the user, an interface is required. This interface has to present the recommendations in a clear and simple manner. Furthermore, it has to take the several media sources into account, so that users are able to see all kinds of recommended content. However, if a user presses a specified button on the remote control, the main menu will be shown (see Figure 16).

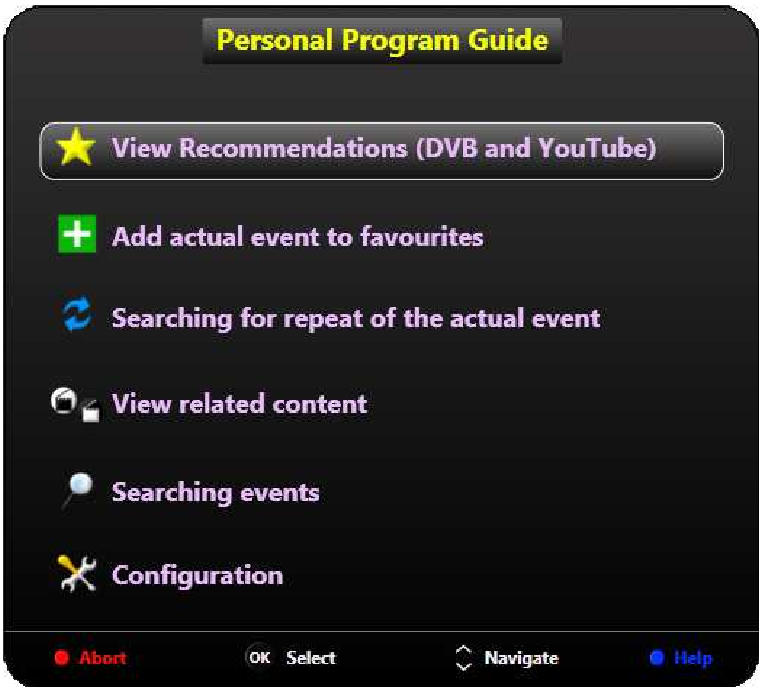

Fig. 16 PPG - Main Menu

The developed PPG provides several features:

- Presentation of recommendations for DVB content
  - whole scheduled information
  - daily recommendations
- Presentation of related DVB content
- Presentation of recommendations for YouTube videos
- Presentation of related YouTube content
- Presentation of collaborative recommendations
- Settings for explicit preferences

The features of the developed PPG will be described in the following sections.

7.2 Recommendations DVB

The developed PPG is able to present recommendations for DVB content. The equations presented in Section 5.1 are responsible for generating the Recommendation Index, which represents individual likings. The PPG lists all recommendations, while the user is able to sort them by title of the events, genres or subgenres (see Figure 17). The recommendations

Fig. 17 Recommendations DVB

will be sorted by start time and start date too.
Furthermore users have two opportunities. They are able to see recommendations for all events within the scheduled information or just recommendations for the current date. This feature has been introduced, since the available scheduled information include events in a timespan of one to three weeks. The result of using the whole scheduled information is a long list of recommendations. But if a user selects 'daily recommendations', the recommendation engine just takes the event into account, which will be broadcast on the current date. Besides these features the PPG presents related DVB content to a currently watched event (see Section 7.2.1).

7.2.1 Related events - DVB

The presented PPG is able to find similar DVB events. This is realized through the available metadata, which is sent within a DVB Transport Stream. The searching for related events is realized with the metadata within the Event Information Table (EIT). This table delivers metadata such as title of an event, the genre and the subgenre of an event. Figure 18 shows the algorithm to implement this feature.

At the beginning the algorithm extracts the metadata of the currently shown event, more precisely the title, the genre and the subgenre. Now the algorithm parses the available scheduled information, which contains several events. It parses the information which will be broadcasted during the next few minutes, hours, days or weeks. This depends on the metadata, which is sent within the EIT. The algorithm goes through and compares them with the metadata of the current event. If the title, the genre and the subgenre of the current event is also available in the scheduled information, the relation between them is 100%, bacause every parameter (title, subgenre and genre) is equal. If the subgenres of the current event is equal to an event of the scheduled information, the relationship between them is 80%. (If

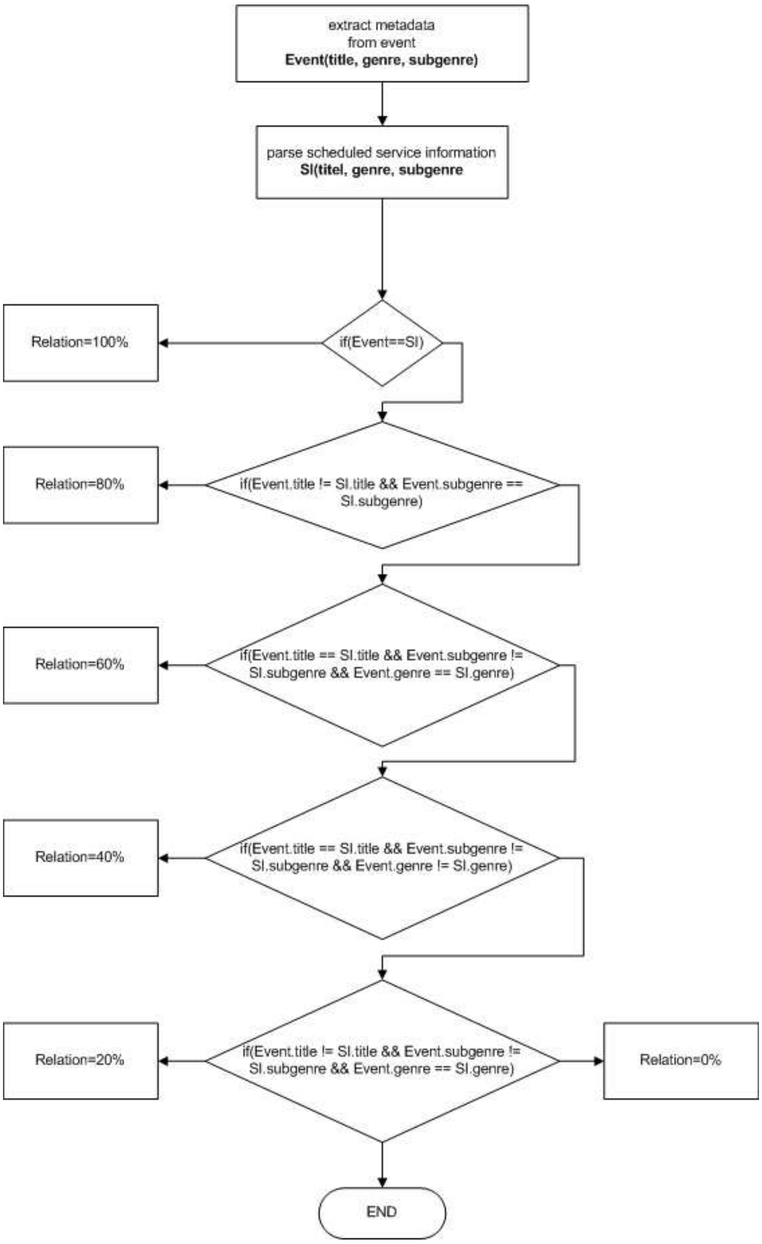

Fig. 18 Process Chart - Related DVB

the subgenre is equal, the genre is equal, too, since a subgenre describes a genre in more detail). If the title and the genre of the current event are equal to a scheduled event, the relationship between them is 60%. If only the title of the current event is equal to an event of the scheduled information, the relationship is 40%. If the genres are equal, the relationship is 20%. Finally if nothing is equal, the relationship is 0%. The percentages are based on the five-stars ranking and the results of the evaluation presented in Section 5.1.3. Since the five-star ranking uses five stars, the algorithm uses this behavior to set the relations.

7.3 Recommendations YouTube

The PPG is also able to present the recommendations for YouTube videos, which are based on the explicitly set preferences, the implicitly logged viewing behavior and the results of the presented evaluation (see Figure 19). Furthermore, is it able to present related YouTube videos (see Section 7.3.1).

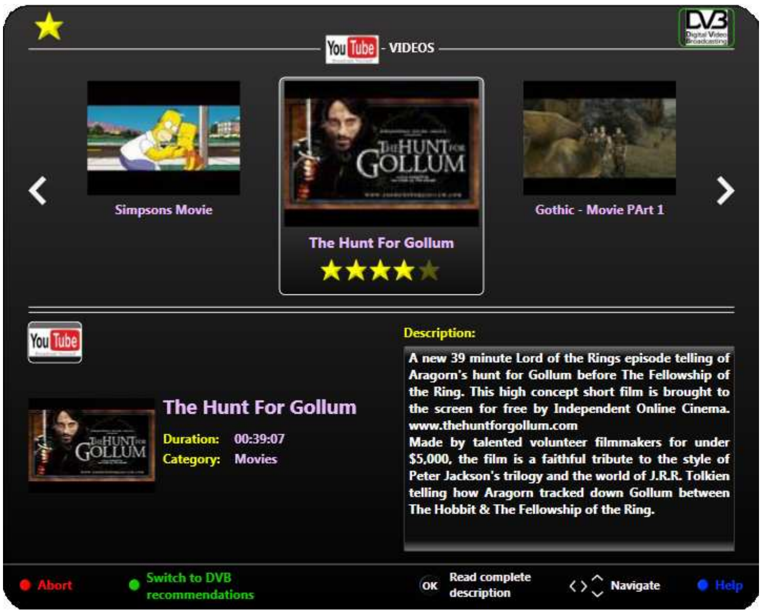

Fig. 19 PPG - YouTube recommendations

The access to YouTube is realized by a Google's API (Application Programming Interface). This API offers the opportunity to get full access to YouTube by sending queries.

7.3.1 Related videos - YouTube

Media sources like YouTube offers the opportunity to find related videos to the ones that are currently being viewed. This helps users to find videos which have similar content. This functionality can be included through an API of Google. The data API can be downloaded from:
`http://code.google.com/intl/de-DE/apis/youtube/overview.html`.

This API is used within the presented PPG to find related YouTube videos to the one that is being viewed. This will offer the opportunity to find interesting contents in an easy and straightforward manner.

7.4 Collaborative Recommendations

The PPG is also responsible for presenting the recommendations, which are based on the collaborative filtering methods (see Section 6. It uses the presented results (see Section 6.5) for presenting the recommendations. The genres of users, which are quite similar with the current user, are displayed. This shall help users to find genres of interest in less time and find content of interest easily.

7.5 Evaluation of the PPG

The first step of the evaluation has been accomplished during the design process. The author's drafted several designs and requested proband for their preferred choice. The result was a mixture of several drafts and was presented using several screenshots above.

In addition, the presented PPG was tested and evaluated by several probands. The questionnaire included several questions about the look and feel of the PPG, expected behavior and so forth. In total 12 probands filled the questionnaire. The following tables (see Table 9, Table 10 and Table 11) show the results of this questionnaire and the usefulness of the presented PPG. The results show that the users are satisfied with the usage of the PPG almost all users would recommend this PPG.

Table 9 PPG - Evaluation - Ages of probands

	12-25	**26-45**	**46-65**
Age	2	8	2
Percent	16,67	66,67	16,67

8 Conclusion

This paper presented new techniques and algorithms to create a Recommendation Index for DVB content and YouTube videos by considering explicitly set preferences and implicitly logged viewing behavior by using content-based filtering methods. The explicit setting is realized by manually set ratings for genres, subgenres and events. A new developed graphical user interface offers the opportunity to set the individual preferences in an easy manner. The logging of the viewing behavior is realized with new algorithms, which have been researched and developed. Furthermore a combination between the explicit settings and the implicitly logged viewing behavior is shown. The generated user profiles, which are saved in XML files, are presented too.

In addition, the paper also shows a comparison between the most known collaborative filtering methods by considering the user-based approach. It shows the most suitable filtering method that is adequate to generate recommendations in a collaborative manner within the presented environment.

Finally a Personal Program Guide (PPG) is presented which can be used to visualize the generated recommendations by considering linear DVB content and non-linear YouTube videos within one application.

Table 10 PPG - Evaluation results - 1/2

Question	Choice	Counter	Percent
PC experience	very good	4	33,33
	good	3	25.00
	average	4	33.33
	not good	0	0.00
	bad	1	8.33
	no answer	0	0.00
Look liking	very good	3	25.00
	good	8	66.67
	average	1	8.33
	not good	0	0.00
	bad	0	0.00
	no answer	0	0.00
Items are clearly	yes	5	41.67
	rather yes	4	33.33
	neutral	2	16.67
	rather no	1	8.33
	no	0	0.00
	don’t know	0	0.00
	no answer	0	0.00
The structure of the windows is understandable	yes	3	25.00
	rather yes	4	41.67
	neutral	3	25.00
	rather no	1	8.33
	no	0	0.00
	don’t know	0	0.00
	no answer	0	0.00
Is it visible at a glance what options are available	yes	5	41.67
	rather yes	5	41.67
	neutral	1	8.33
	rather no	1	8.33
	no	0	0.00
	don’t know	0	0.00
	no answer	0	0.00
The work with the software was fun	yes	6	50.00
	rather yes	3	25.00
	neutral	2	16.67
	rather no	0	0.00
	no	0	0.00
	don’t know	0	0.00
	no answer	1	8.33
The information in the help is helpful	yes	7	58.33
	rather yes	2	16.67
	neutral	1	8.33
	rather no	0	0.00
	no	0	0.00
	don’t know	2	16.67
	no answer	0	0.00
It must be read too much before the software can be used	yes	0	0.00
	rather yes	2	16.67
	neutral	1	8.33
	rather no	2	16.67
	no	7	58.33
	don’t know	0	0.00
	no answer	0	0.00

Table 11 PPG - Evaluation results - 2/2

Question	Choice	Counter	Percent
Software does what was expected	yes	5	41.67
	rather yes	4	33.33
	neutral	0	0.00
	rather no	2	16.67
	no	0	0.00
	don't know	0	0.00
	no answer	1	8.33
The software is very cumbersome to use	yes	0	0.00
	rather yes	0	0.00
	neutral	1	8.33
	rather no	6	50.00
	no	5	41.67
	don't know	0	0.00
	no answer	0	0.00
First use of the software was fraught with problems	yes	0	0.00
	rather yes	1	8.33
	neutral	1	8.33
	rather no	5	41.67
	no	5	41.67
	don't know	0	0.00
	no answer	0	0.00
The software is suitable for beginners	yes	0	0.00
	rather yes	7	58.33
	neutral	2	16.67
	rather no	2	16.67
	no	0	0.00
	don't know	1	8.33
	no answer	0	0.00
Would you recommend this software	yes	4	33.33
	rather yes	7	58.33
	neutral	1	8.33
	rather no	0	0.00
	no	0	0.00
	don't know	0	0.00
	no answer	0	0.00

References

1. Y. Blanco, J. Pazos, A. Gil, M. Ramos, M. López and B. Barragáns, AVATAR: Modeling users by dynamic ontologies in a tv recommender system based on sematic reasoning, Proceedings of the 3rd European Conference on Interavtive TV, 2005
2. Iain Campell and Keith van Rijsbergen, The Ostensive Model of developing information needs, Proceedings of: CoLIS, 1996
3. Frank Hopfgartner and Joemon M. Jose, Semantic user profiling techniques for personlized multimedia recommendation, Multimedia Systems, Volume 16, Pages: 255-274, 2010
4. Frank Hopfgartner, David Hannah, Nicholas Gildea and Joemon M. Jose, Capturing Multiple Interests in News Video Retrieval by Incorporating the Ostensive Model, 2nd International Workshop on Personalized Access, Profile Management, and Context Awareness: Databases, 2008
5. David Elsweiler, Gareth J.F. Jones, Liadh Kelly and Jaime Teevan, Desktop Search, Workshop of the 33rd Annual International, ACM SIGIR Conference on Research and Developmentin Information Retrieval, 2010
6. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen and John T. Riedl, Evaluating Collaborative Filtering Recommender Systems, Third ACM Conference on Digital Libraries in Pittsburgh, 1998
7. Mukund Deshpane and George Karypis, Item-Based Top-N Recommendation Algorithms, ACM Transactions on Information Systems, Volume 22, Pages: 143-177, 2004
8. Badrul Sarwar, George Karypis, Joseph Konstan and John Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, ACM, 2001

9. Christian Überall, Muttukrishnan Rajarajan, Veselin Rakocevic, Rudolf Jäger and Christopher Köhnen, Recommendation Index for DVB content using Service Information, Proceeding of the ICME, IEEE, 2009
10. Shang H. Hsu, Ming-Hui Wen, Hsin-Chieh Lin, Chun-Chia Lee and Chia-Hoang Lee, AIMED - A Personalized TV Recommendation System, Interactive TV: A Shared Experience, Springer Verlag, Pages: 166-174, 2007
11. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John T. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, Volume 22, Pages: 5-53, 2004
12. Wiezhen Dai and Robin Cohen, Dynamic Personalized TV Recommendation System, Proceedings of the UM 2003 Workshop on Personalization in Future TV, 2003
13. Hongguang Zhang, Personalized TV Program Recommendation based on TV-Anytime Metadata, Proceeding of the Ninth International Symposium on, IEEE, 2005
14. European Standard (Telecommunications series, Digital Video Broadcasting (DVB) - Specification for Service Information (SI) in DVB systems, Version: V1.4.1, 2000
15. M. Balabanovic and Y. Shoham, Content-Based Collaborative Recommender, Third ACM Conference on Digital Libraries in Pittsburgh, Volume 40, Pages: 66-72, 1997
16. Yolanda Blanco-Fernandez, Jose J. Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer and Martin Lopez-Nores, A Hybrid Strategy to Personalize the Digital Television by Semantic Inference, Interactive TV: A Shared Experience, 2008
17. Xiaqyuan Su and Taghi M. Koshgoftaar, A Survey of Collaborative Filtering Techniuqes, Hindawi Publishing Corporation, 2009
18. Lothar Papula, Mathematische Formelsammlung, vieweg, 2003
19. L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Chiarotto, A. Difino and B. Negro, Personalized Recommendation of TV Programs, Proceedings of the 8th AI*IA Conference, 2003
20. Toon De Pessemier and Luc Martens, A profile based recommendation system for TV-Anytime annotated content, 8th FirW PhD Symposium, 2008