Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

OMUS: An Optimized Multimedia Service for the Home Environment

**Simon Dooms · Toon De Pessemier ·
Dieter Verslype · Jelle Nelis ·
Jonas De Meulenaere ·
Wendy Van den Broeck · Luc Martens ·
Chris Develder**

Received: date / Accepted: date

Abstract Media content in home environments is often scattered across multiple devices in the home network. As both the available multimedia devices in the home (e.g., smartphones, tablets, laptops, game consoles, etc.) and the available content (video and audio) is increasing, interconnecting desired content with available devices is becoming harder and home users are experiencing difficulties in selecting interesting content for their current context.

In this paper, we start with an analysis of the home environment by means of a user study. Information handling problems are identified and requirements for a home information system formulated. To meet these requirements we propose the OMUS home information system which includes an optimized content aggregation framework, a hybrid group-based contextual recommender system, and an overall web-based user interface making both content and recommendations available for all devices across the home network. For the group recommendations we introduced

S. Dooms - T. De Pessemier - L. Martens
Wica, iMinds-Ghent University
G. Crommenlaan 8 box 201, 9050 Ghent, Belgium
Tel.: +32-09-33-14908
Fax: +32-09-33-14899
E-mail: simon.dooms@intec.ugent.be
E-mail: toon.depessemier@intec.ugent.be
E-mail: luc.martens@intec.ugent.be

D. Verslype - J. Nelis - Chris Develder
IBCN, iMinds-Ghent University
G. Crommenlaan 8 box 201, 9050 Ghent, Belgium
E-mail: dieter.verslype@intec.ugent.be
E-mail: jelle.nelis@intec.ugent.be
E-mail: chris.develder@intec.ugent.be

J. De Meulenaere - W. Van den Broeck
SMIT, iMinds-VUB
Pleinlaan 9, 1050 Brussels, Belgium
E-mail: jdemeule@vub.ac.be
E-mail: wendy.van.den.broeck@vub.ac.be

distinct weights for each user and showed that by varying the weights, the coverage (i.e., items that can be returned by the recommender) considerably increases. Also the addition of genre filter functionality was proven to further boost the coverage.

The OMUS system was evaluated by means of focus groups and by qualitative and quantitative performance assessment of individual parts of the system. The modularity of internal components and limited imposed hardware requirements implies flexibility as to how the OMUS system can be deployed (ranging from e.g., embedded in hardware devices or more software services based).

Keywords Multimedia · information systems · graphical user interfaces · DLNA · recommender systems · algorithms

1 Introduction

In a world ever becoming more digital, information systems are key towards handling and organizing the enormous amounts of data that need to be processed by people every day. This paper focusses on the consumption of media content by multiple people (e.g., group of friends, families, couples) in a home environment. Specifically, it addresses the requirements and implementation of a home information system that assists users with their media consumption process in a personalized way.

Home environments (from a technical viewpoint) nowadays consist of multiple people (referred to as users) living together, where each of them uses a wide variety of heterogeneous devices like smartphones, laptops, tablets, and television sets. Because of the diversity and abundance of available devices, media content is often scattered across multiple locations (and types of devices) in the home network. This problem of having (and using) many different devices on the same network has been partially addressed by the DLNA[1] standard (built on top of UPnP AV[2]) which aims to define a set of protocols allowing networked multimedia devices to discover each other's presence and enable seamless sharing of digital media content.

Another phenomenon within a modern home environment is the massive availability of digital media content that home users are exposed to. Next to television broadcast channels and interactive digital TV services, many other online sources nowadays offer media ready for playback on demand. Due to this information overload, users often experience difficulties in selecting interesting content for the context they find themselves in (e.g., selecting a movie to watch with a group of friends). Recommender systems are systems designed to tackle the information overload problem. For over twenty years [30], researchers have been designing and experimenting with recommendation algorithms that try to bridge the gap between users and content items in an optimal way. Many of these algorithms are designed for very specific contexts or situations (e.g., Netflix movie recommendation [4]) but they often incorporate ideas that are more generally applicable to other domains as well.

In this paper we present the Optimized MUltimedia Service (OMUS) system that aims to offer an overall and integrated information system to overcome the

[1] http://www.dlna.org

[2] http://upnp.org/specs/av

information handling problems in the home environment. By building upon existing technology rather than introducing many new components, the system is able to uphold high compatibility with existing devices and services while offering the necessary improvements towards the in-home situation.

The contributions of this paper include:

1. Analysis of the home environment by means of a user study. Information handling problems are identified and requirements for a home information system formulated.
2. Introduction of an optimized content aggregation framework.
3. Construction of a real-time group-based contextual recommender system suited for the home environment.
4. Demonstration of an overall web-based user interface for a home information system making both content and recommendations available across the home network.

Besides the novelty of these individual components, the true contribution of this work lies in the way they are integrated together to be able to transcend their individual research domains and as a complete system tackle a broader and complexer information problem while still being realistically deployable.

The OMUS system was evaluated by means of focus groups and by qualitative and quantitative performance assessments of individual components of the system. These components include the content aggregation system, the user interface and the recommender system.

The remainder of the paper is organized as follows: Section 2 reviews related literature on the topic of home information systems and recommender systems. Section 3 then discusses the user studies that were conducted and defines a set of user requirements for a home information system. Section 4 introduces the OMUS system in terms of a high-level architecture. The following Sections 5, 6 and 7 cover in greater detail how, respectively the content aggregation system, the user interface and the recommender system, are constructed and evaluated specifically towards the requirements of users in their home environment. The OMUS system is compared to existing information systems in Section 8 and finally, Section 9 presents some concluding notes and overviews the strengths and limitations of the proposed OMUS system.

2 Related Work

The introduction of the UPnP AV (an audio and video extension of UPnP) architecture in 2006 was a leap forward regarding transparency and interoperability of multimedia devices in the home network [20],[10],[17]. A lot of related work has therefore been extending upon the concepts of UPnP, UPnP AV and DLNA [18]. The current content aggregation framework is an extended and optimized version of our preliminary system reported in [27], outlining an initial version of a single home content aggregation framework. The current paper in particular proposes and analyzes optimization (through parallelization) of the performance of the aggregation algorithms. Other works address the issue of quality of service (QoS) and possible adaptation of the video format [7], [12], a complementary topic which we do not further elaborate on here.

Also, effort has been put in the idea of extending universal plug and play mechanisms across different homes [22], [21]. The concept of these interconnected home environments is often supported by a proxy service that is implemented in either a home gateway or home server in both homes. In [19], flexible content searches are enabled over many different home networks by connecting UPnP gateways in unstructured P2P networks.

Following the path set by previous research, it seems that the availability of digital media content inside the home will keep on increasing. Therefore the well-known problem of choice overload that Internet users experience daily is likely to be equally prevalent in content scattered across home environments. This need for personalized services that assist users in sifting through the huge amounts of content inside the home was also noted by Sales *et al.* [31]. They introduced the UPnP-UP extension allowing user authentication and authorization for UPnP devices and applications. Because users could be authenticated, the network was capable of offering them a personalized experience towards their content. To illustrate this, Sales *et al.* integrated a UPnP multimedia system BRisa [14] that recommended songs to users according to their profile. The recommendation subsystem was driven by a hybrid recommendation algorithm combining collaborative filtering and content-based filtering in a cascading way. Although Sales *et al.* paved the way towards personalized services in the home network, an all-round home information system designed specifically towards realistic user requirements like content aggregation and contextual personalization was never introduced.

Three main recommendation algorithms categories stand out in literature: collaborative filtering (CF), content-based filtering (CB) and knowledge-based filtering (KB).

The earliest recommendation technique, and still very popular today, is collaborative filtering (CF) [30]. The main idea behind CF is the assumption that users who were similar in the past, will be similar in the future [16]. In this way similar users are used as guides towards interesting items. The difficulty of CF usually comes down to finding similar users in an efficient way. Comparing every user with every other user in a population is a problem with a quadratic complexity which may potentially be the limiting factor of the scalability of the system.

CB harvests information about content items that a user has rated in the past, to build a user profile. Based on that user profile, interesting items will be matched. The main idea behind CB recommendation is that users will like items that are similar to those that they liked in the past [16]. The similarity between items has to be determined, and again this is a quadratic process. Furthermore, the similarity is based on descriptive item information which may not always be available.

The last main category of recommendation algorithms are the knowledge-based systems. These recommendation systems rely on explicit domain knowledge about items or users that was collected by means of domain experts or market research [9], [25]. From this knowledge, rules like “this category is interesting for male users with age between 20 and 25” can be extracted to generate recommendations.

One of the first group recommendation systems was the MusicFX system [24]. This recommender was designed to provide a gym with workout music tailored specifically to the preferences of all the present customers. Over the years many other group recommenders have been developed in various sectors as tourism, web, music, television, etc. An extensive overview of existing group recommendation systems can be found in [15].

When it comes to the calculation of recommendations for groups, two main strategies can be distinguished: *aggregated models* and *aggregated predictions*. The first strategy is to create a group model that incorporates the user feedback (e.g., ratings) of every individual group member, such that the recommendation system can transparently process the group as an individual user and come up with an appropriate recommendation list. The second strategy calculates individual recommendation lists for each member of the group and afterwards tries to combine them into one list. Both strategies require a way of aggregating data. They only differ in the order in which this aggregation is applied into the recommendation process.

For the aggregation step, multiple approaches are available, as detailed in [23]. Two examples of such aggregating functions are *Average* which simply averages out the values, and *Least Misery* which always takes the minimum of the values. The application of different functions may have different implications: *Least Misery* and (conversely) *Most Pleasure* would for example result in a decrease of accuracy [5] but it has also been stated that the aggregation method itself actually does not have a big influence on the overall quality of the recommendations [1].

Interestingly, Masthoff [23] performed an experiment to find out how humans would decide what to watch based on individual ratings of members in a group. It turned out that they intuitively applied functions as *Average*, *Average Without Misery* (i.e., average with a minimum threshold) and *Least Misery*. Furthermore they tend to care about fairness and individual misery.

3 User Studies and Requirements

The OMUS research project[3], (in which this work is situated), was an interdisciplinary research project involving research on both technological aspects as well as user research. In a first part of the user research, focus was on users current media consumption practices (including media retrieval, storage and consumption on different devices as well as current use of existing recommender systems). Also their annoyances and future expectations were questioned. In a second phase of the research, the developed OMUS system was evaluated by users with different profiles.

The data presented here is derived from a series of 7 focus group interviews totalling 47 respondents. Different target groups were integrated in the research, including students, heavy downloaders, regular interactive digital television viewers and people with an extensive media set-up at home. The findings suggest that despite the rise of new entertainment media, watching television remains an important time-spending activity. This is in line with the Digimeter report[4] which shows that an average Flemish household watches four hours TV per day during the week, which rises to four hours and a half during the weekend. Furthermore, the TV-set is one of the media devices with one of the highest penetration rates in Flanders, being approximately 98%. Television is a very accessible medium,

[3] http://www.iminds.be/en/projects/overview-projects/p/detail/omus-2

[4] http://www.digimeter.be

often used for relaxation and entertainment purposes, and domesticated in daily routines [8]. Also in the focus group interviews respondents indicated that when being at home and wanting to relax, video content consumption is one of the first activities that comes to mind.

The consumption of video content however extends beyond the regular TV-set. Data shows that computers, either portable or stationary, are at least sometimes used by approximately 50% of the Flemish population for watching movies, and 70% sometimes watches video content over the Internet, not restricted to films. Figures on online viewing behavior in the US show that 32% of US adults have watched movies or TV shows online, with the highest rates amongst the youngest segment (18-29) [29]. In addition, some of our respondents stated they also used handheld devices for watching video content such as TV-series or movies. Nevertheless, in terms of watching video content, the TV-set is preferred to other playback devices because of its ease of use, better picture quality and because it is easier to watch content on it together with others. Content is usually watched on the device on which the content is stored. This indicates that users often lack the skills or the willingness to transfer video content from one device to another, mainly because of the hassle and the perceived complexity.

The digitization process has an even more pervasive and deeper impact on content availability. Television has become an interactive medium with a multiplication of broadcasters, access to video on demand services and a more convenient way of time shifting. Furthermore, the Internet has become an even larger pool of content, virtually free of spatio-temporal constraints, but requiring more effort than simply switching on the TV-set. If video content is searched for intentionally, users will rely on sources with which they have developed a relationship. Some still depend on broadcasters whereas others might rely on online sources (e.g., blogs with reviews). But in general, peers with similar preferences are considered to be most trustworthy for advice on video content selection.

In general, our focus group conversations did not reveal a strong need for computer assistance in selecting video content. Sometimes even strong resentments were voiced as our respondents did not want to be dependent on what a machine decided. In order to establish a similar trustful relationship as the aforementioned, a number of requirements can be formulated regarding a home information system. These requirements relate to the way content is approached, the viewing context, and individual differences in the way users want to control and interact with the system.

We found that content attributes such as runtime, genre, ratings, language, actors/actresses, director, etc. are used to find a match between the content and the contextualized preference. Besides the pragmatic value, these attributes also yield symbolic meanings. For instance, genre is a powerful means to describe what kind of content would appeal in a specific context. While there are a number of attributes that are rather commonly used, not everyone ascribes the same weight to every attribute. To some the director is very important whereas others might be more interested in the ratings the item has received from for instance the popular Internet Movie Database (IMDb)[5].

Preferences differ depending on the context. Video content is often watched in the presence of others, implying that a consensus has to be attained. A recurrent

[5] http://www.imdb.com

statement was that settling on an agreement was often one of the peculiar things that makes watching a movie together with friends a pleasant experience. In that respect, shared interest or a common denominator were voiced as the main characteristics. Additional information such as reviews or trailers were also mentioned. On the other hand, it was also voiced that sometimes the opinion of the most knowledgeable is followed. It is clear however that this form of content selection only occurs in rather unusual occasions; for instance a movie night once a month. In general when users watch TV, they have their favorite TV-programs they follow regularly or consult the Electronic Program Guide to see what is scheduled.

Significant differences were recorded among our respondents in terms of willingness to actively engage in searching and selecting video content. Whereas some rather sit back and receive, others indulge themselves in finding out about new content. Willingness and lifestyle are besides skills and knowledge important factors that explain why someone will or will not actively search for new content. To some, this is limited to paper and online TV-guides and/or their personal social network (e.g., friends, neighbors, colleagues), others might also use online sources (e.g., blogs, trailers, sequences).

These social requirements also affect the functional requirements. First, the system needs to be integrated as much as possible with the existing home network configurations. This implies that when content is selected in the system, it should be possible to directly play the content on the preferred device from within the same user interface. Next, the system should be customizable. We noted that movie content is selected based on a number of content attributes that are relevant to the user. As differences exist in what content attributes are relevant, it is advised that only the most commonly used attributes are presented to the user initially, while more information is hidden. Furthermore, not everyone will actively engage with a home information system. Therefore, an explicit user profile should only be an optional feature, which allows the more active users to specify their preferences regarding media content on a fine grained scale. Also, as genre is often used to express one's contextualized content preference, a filtering option would be an asset.

In conclusion we list the most important requirements for a home information system as dictated by the users in our user study.

General requirements

- Support for a wide variety of distinct devices in the home network.
- Easy playback of content on a preferred device.

Recommender system requirements

- Contextual user preferences (e.g., user weights).
- Support for group recommendation.
- Support for different types of user engagement: active and passive users.
- Allow an (optional) explicit user profile.
- Allow users to control their user profiles.

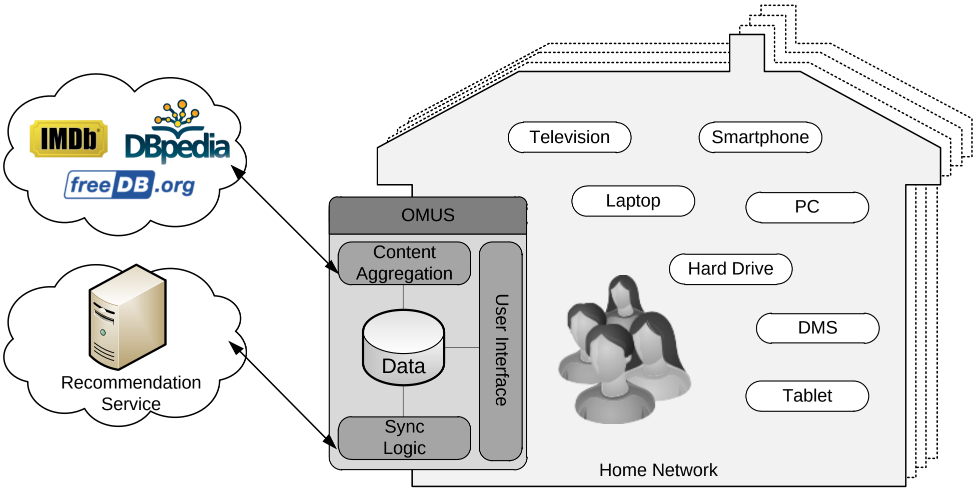


Fig. 1 The high-level view of the OMUS system integrated in a home environment with multiple users and distinct devices.

User interface requirements

- Show only basic item information, show more if requested.
- Allow to filter lists on genre.

4 System Architecture

In this section we discuss the high level architecture (Fig. 1) of the OMUS system that was implemented in view of the aforementioned user requirements. The home network houses different users that each interact with a number of distinct interlinked devices. The OMUS information system, proposed in this paper, is situated at the border of the home network. The system includes components as content aggregation, a recommender system, a user interface and a central data component. The possibility of multiple houses each implemented with the OMUS system and therefore using the same recommendation service, is graphically illustrated in the architecture by the dotted house outlines extending the central home network.

The content aggregation component centrally gathers all the information about the media content that is available to home users (in-home media), complemented with data about other consumable media (e.g., content in a friend's home, online sources). The metadata stored about an item can originate from the device that is offering the item or from online databases like IMDb.

As the OMUS framework aggregates a considerable amount of data about media and users in the home network, a persistence layer to deal with this data was needed. We designed a data model for the items and their metadata, the users and their preferences. Different components in the architecture interact only by means of the data storage component, which results in a loose coupling of the components, thus offering flexibility as to where to deploy each of them. Although this data-central architecture can introduce a performance bottleneck through

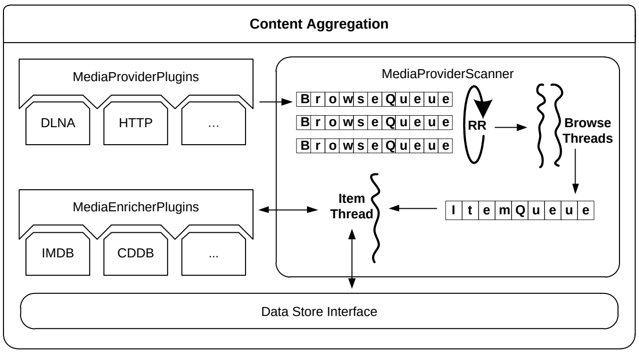


Fig. 2 The internal architecture of the content aggregation component of the OMUS system.

the data component, we believe (and our tests confirmed) that for realistic user scenario's this will not be a problem and outweighs the burden of synchronising data among the different OMUS components.

The sync logic component is responsible for synchronization services between the home network and an external recommendation service. The external recommendation service provides the necessary recommendation functionality without imposing (computing) hardware requirements inside the home network.

The user interface (UI) actively interfaces between users, devices and the OMUS information system. All aggregated content in the home network is integrated into a single content overview list and made available through the UI. Extra information about items (e.g., plot, genre, etc. for movies) is easily accessible, playback functionality is provided and user preferences can be specified. A recommendation list tailored specifically towards any provided context additionally assists users in their content selection process.

Additional information about the inner workings of the content aggregation component, the user interface and the recommender system will be provided in Sections 5, 6 and 7 respectively.

5 Content Aggregation

The content aggregation component tries to minimize the management complexity of multimedia scattered over a range of home network devices. Information about all the media is centrally gathered, so the entire virtual library can be easily browsed and visualized.

5.1 Content Aggregation Architecture

Fig. 2 shows the architecture of the content aggregation (CA). To abstract the types of devices that can provide media (like a DLNA-enabled mobile phone or a pc with some shared folders), we defined the MediaProvider interface. A MediaProvider represents a discoverable device that can be browsed for its content (e.g., through UPnP AV). Eventually the content can be consumed by downloading or streaming using a network protocol (e.g., HTTP or RTSP). The CA architecture foresees flexibility and extensibility concerning the MediaProvider types: several MediaProvider implementations can be plugged in, depending on the use case. For the OMUS system, we implemented the DLNA MediaProvider type, so all DLNA content could be gathered. When the OMUS CA system discovers a DLNA server that comes online, a DLNA MediaProvider component is instantiated, that wraps the UPnP AV device into a MediaProvider and offers it for scanning to the MediaProviderScanner.

The MediaProviderScanner is responsible for discovering the content present on the MediaProviders, detecting and merging duplicates, enriching metadata and storing all this information in the data component. The discovery (i.e., browsing) of content happens transparently through the MediaProvider interface as the specific MediaProvider implementations take care of the underlying discovery. Considering that the objects or folders within the shared media tree can contain a large number of media items, we enabled chunked browsing in the interface, so only a certain maximum number (a chunk) of the content item descriptions is returned in one browse call. E.g., in case a large folder contains 120 items and a chunk size of 50 is configured, the content descriptions will be fetched using 3 browse calls. All outstanding browse calls are stored in a browse queue (starting with a request for the root folder) and are sequentially handled by one of the parallel browse threads. Such a thread removes the first browse request from the browse queue, executes this request and analyses the result: for every subfolder a number of browse requests are added to the browse queue (based on the folder size and chunk size), every item description is added to the item queue for further inspection. Since we require the distributed collection to be scanned as fast as possible (such that turning a media device on or off results in a quick update of the central data store), we tried parallelizing the browse calls by having multiple threads performing browse calls to one or more simultaneously scanned MediaProvider(s). In the latter case, we have one browse queue per MediaProvider, to spread the requests over the MediaProviders as much as possible: with only one browse queue and large folders on one of the scanned MediaProviders, subsequent chunks would all be queued after each other and thus, at some point, requests from different browse threads would be all directed to the same MediaProvider (which does not significantly speed up the overall process, cf. infra).

Subsection 5.2 will look into determining an optimal value for the chunk size and the number of browse threads in terms of overall scanning speed, based on measurements with 3 different DLNA media servers.

Only one item thread processes the discovered media item descriptions to avoid duplicate items in the data storage. When an item is removed from the item queue for further inspection, the data store is queried for items with the same name. If an item is found, this item is merged with the newly found information (new resource location, etc.), otherwise a new item is created in the data store and possible metadata (year, genre, title, etc.) is stored. For the enrichment of item metadata, a new type of plugin is introduced: the MediaEnricher. Such a MediaEnricher tries to find additional metadata about items, based on the already known information and the content itself. We implemented an IMDb MediaEnricher which augments the metadata of an item with IMDb information about actor, director, genre, year, etc. Other possibilities would be face recognition for pictures, address resolution based on the geotags in pictures or videos, etc. While processing a discovered item, all MediaEnricher plugins are presented to enrich this item sequentially.

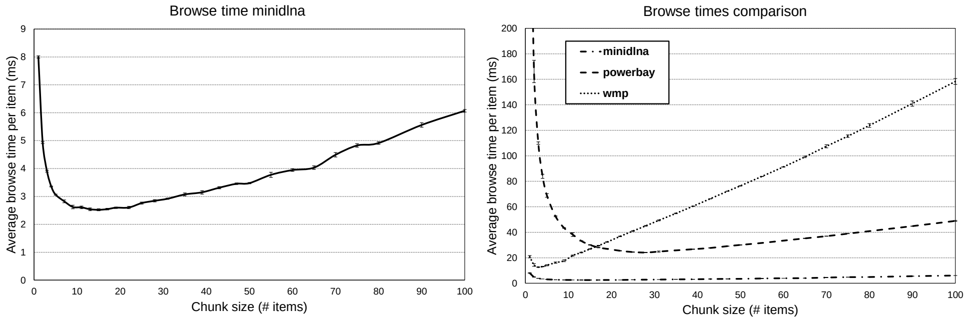


Fig. 3 The browse time averages over 10 experiments for minidlna (left) and a comparison of the three DMSes (right), in function of varying chunk sizes.

Our CA implementation consists of a number of plugins for DYAMAND [26], an adaptive framework for managing networks and connected devices.

5.2 DLNA Performance Testing

Because the scanned folders in DLNA often contain a large amount of items, and folders can be browsed in chunks, we looked into the influence of the chunk size — used when scanning a digital media server (DMS) — on the scanning speed. For these tests we made 10 collections, each comprising 200 items in total. Every item (an MP4 file) was tagged with a random movie title from a list of 1600 movie titles, together with the correct year, genre and poster thumbnail. We used a hardware DMS (Verbatim PowerBay) and 2 software DMSes: minidlna[6] and Windows Media Player (WMP). We sequentially scanned the 10 collections on all 3 media servers for a varying chunk size. Figure 3 shows a detail for the minidlna DMS (left) and compares an aggregated result (averaged over the 10 collections) (right) together with the 95% confidence intervals. We observe a severe performance overhead for PowerBay and small chunk sizes (< 10), which is much more limited for WMP and minidlna. For all approaches, we observe a linear increase of browse time per item, for larger chunk sizes. This penalty for large chunk sizes is especially severe for WMP. We also note the very good performance of the lightweight minidlna, which sends less meta-information, e.g., less resources (minidlna does not transcode the video to other formats and only supports http as transport protocol) and no dummy tags (e.g., "Unknown Author"). A certain video item description in minidlna was only 21% the size in bytes of the WMP description of the same item. PowerBay's descriptions are less bloated as well, but the DLNA software runs on embedded hardware (instead of laptop computers for WMP and minidlna), which explains the lower performance in comparison to minidlna.

Thus, our tests show that every DMS has an optimal chunk size, in which the average browse time per item is smallest. While the exact optimum depends on the particular DMS, a value in the range of 10-30 seems an appropriate choice for all of them.

[6] http://sourceforge.net/projects/minidlna

Table 1 Performance testing with a collection of 200 items, chunk size 15 and a varying number of browse threads. Total browse times are presented in ms.

	1 Thread	**2 Threads**	**3 Threads**	**4 Threads**
minidlna	3477	3088	3081	3042
PowerBay	55 307	53 654	53 396	53 531
WMP	50 529	49 967	49 674	49 631

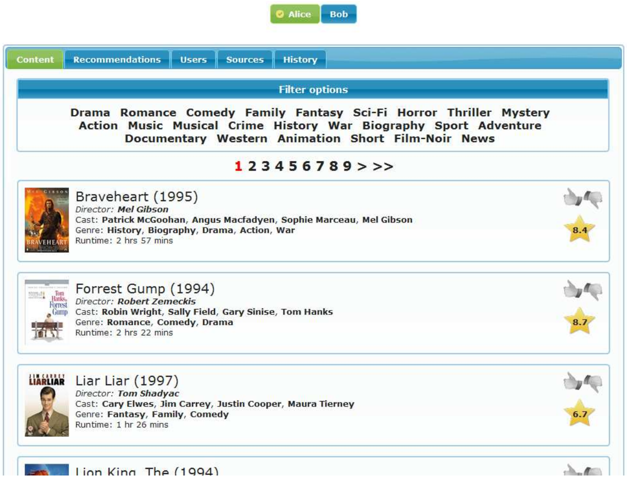
Having determined a (near-)optimal browse chunk size, we still need to investigate whether there is any gain in parallelizing the browse requests. For the second series of tests, we used the previously mentioned DMSes separately, with a collection of 200 items and choose a chunk size of 15 (based on the previous test results). We scanned the content of the devices with a varying number of browse threads (each requesting chunks of the entire media collection). Table 1 shows the average total scanning time (over six tests) of the collection for the different DMSes and differing thread numbers. Only minidlna shows a significant gain of 11% when using two browse threads instead of one. Using multiple browse threads does not result in a speedup for PowerBay and WMP, presumably because the requests are handled sequentially at server side. In conclusion, choosing more browse threads does not speed up the scanning process heavily, at most two browse threads per scanned media server seems fine.

6 Visualizing Content: The User Interface

The user interface (UI) is a crucial part of the OMUS system, as it will intermediate between the content and the users. From the user studies, we learned that users may have very diverse requirements towards how they wish to interact with a home information system. In particular, the way in which they were willing to provide feedback (e.g., provide ratings for media) was very user dependent. This section describes how the user interface was designed to handle these situations while providing the functionality to browse content, control media (i.e., play, pause, and stop media) and to tailor recommendation lists to any given context. Because the UI must be easy to use and intuitively to work with, many of the integrated UI concepts presented here resemble in style and behavior to that of common web applications to which users may already have an affiliation with (e.g., IMDb, YouTube, etc.). The user interface was designed to be web-based (mainly HTML and JavaScript) to make it accessible through any web-enabled device present in the home network.

6.1 Browsing and Interacting

All content available in the home network will ultimately be transparently aggregated into one overview list. For this list to become available, a set of active users must be specified to the system. Active users in terms of the OMUS system are users that want to participate in the same session of media consumption. Showing content only after the user selection allows for possible security policies to


Alice
Bob
Content
Recommendations
Users
Sources
History
Filter options
Drama Romance Comedy Family Fantasy Sci-Fi Horror Thriller Mystery
Action Music Musical Crime History War Biography Sport Adventure
Documentary Western Animation Short Film-Noir News
1 2 3 4 5 6 7 8 9 > >>
8.4
8.7
6.7

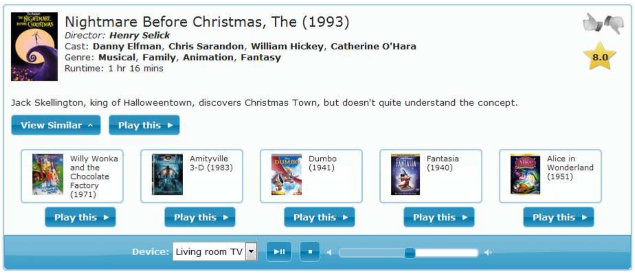
Braveheart (1995)
Director: ***Mel Gibson***
Cast: **Patrick McGoohan, Angus Macfadyen, Sophie Marceau, Mel Gibson**
Genre: **History, Biography, Drama, Action, War**
Runtime: 2 hrs 57 mins

Forrest Gump (1994)
Director: ***Robert Zemeckis***
Cast: **Robin Wright, Sally Field, Gary Sinise, Tom Hanks**
Genre: **Romance, Comedy, Drama**
Runtime: 2 hrs 22 mins

Liar Liar (1997)
Director: ***Tom Shadyac***
Cast: **Cary Elwes, Jim Carrey, Justin Cooper, Maura Tierney**
Genre: **Fantasy, Family, Comedy**
Runtime: 1 hr 26 mins

Lion King, The (1994)

Fig. 4 The basic content overview list. All media content discovered in the network will be enlisted here. At the top, active users can be specified and genre filters can be selected. Basic item information is available together with a thumbs up/down feedback system per item.



Nightmare Before Christmas, The (1993)
Director: ***Henry Selick***
Cast: **Danny Elfman, Chris Sarandon, William Hickey, Catherine O'Hara**
Genre: **Musical, Family, Animation, Fantasy**
Runtime: 1 hr 16 mins

Jack Skellington, king of Halloweentown, discovers Christmas Town, but doesn't quite understand the concept.

Fig. 5 The item-specific content view. Additional information about the media item at hand is provided (e.g., movie plot) together with similar items and some media interaction buttons.

be enforced towards the accessibility of some content (e.g., content not suited for children). Content may be restricted to the smallest section of items that every user in the active user set is allowed to access.

The content overview list should contain all relevant information but at the same time remain simple to use and easy to access. To meet these requirements we propose a two-leveled hierarchical overview. At the first level, only basic information (e.g., title, director, cast, genre and runtime) about content items is shown (Fig. 4). The basic information level offers a quick overview of the available content. When a specific content item is selected from the list, more detailed information is shown (Fig. 5), e.g., the plot in the case of a movie item type or web links to external sources (e.g., IMDb) with extra information such as reviews or trailers.

Aside from item information, the item-specific view can also be used to provide controls and tools to interact with the content. For every available item, similar items can be displayed (Fig. 5). These similar items are calculated (see Subsection 7.4 for details) from the same content pool as shown in the content overview list. Consequently, every item displayed in the similar items list can in turn be interacted with.

The most interesting way of interacting with media content is by actually consuming it (i.e., listening to music or watching video). This is supported by the user interface by means of the media control buttons on the bottom of the item-specific information view (Fig. 5). Clicking the *Play this* button will trigger UPnP AV *SetAVTransportURI* and *Play* messages, to be sent to the currently selected device in the device selection box. These messages in their turn cause the DLNA Digital Media Renderer (DMR) to start buffering and playing the concerning multimedia content item. The device selection box is automatically populated with devices discovered in the home network that announced themselves as being a DLNA DMR. Therefore, the user interface displays every available content item in the network in one interactive list in which content information can be shown, as well as interaction functionality is provided to allow the playback of every media item on every capable device in the home network.

6.2 Providing Feedback

To overcome the problem of overloading users with available items that may be consumed, a recommender system was integrated in the user interface. To enable recommendations to users about which content items they might like, user feedback must be collected first. Collecting user feedback is a way of learning the preferences of users and allows for profile building. User profiles can then be used by the recommender system to come up with interesting matching items. Through the user interface three distinct types of feedback are collected: ratings, item attribute feedback and consumptions.

The first, and most obvious one, is the gathering of explicit user ratings by means of the thumbs up/down widget available on the basic information display of every item (Fig. 4). With this widget, users can straightforwardly express their either positive or negative preference towards any specific content item in the system. The thumbs feedback system is a very intuitive and easy to use feedback system which is usable across a number of different input devices including touch devices (e.g., smartphones and tablets). The thumbs feedback system was therefore chosen over the more commonly used 5-star rating system.

The thumbs up/down feedback system allows users to express preferences on a binary scale. The user study revealed that some users are willing to put more effort into providing feedback than others and would like an increased level of control over their user profiles. To meet these requirements, an additional level of explicit feedback was introduced. Aside from liking or disliking an item in the overview list, users are also able to express their preference on a finer scale more specifically towards an item attribute. When an item attribute in the content overview list is selected, a popup window will allow (thumbs up/down) user feedback towards the relevant attribute. For an item of the movie type, specific user feedback can be provided towards directors, cast and genre by selecting the relevant attributes in the content overview list.

Users are thus able to explicitly express their preferences on either items or specific item attributes. As the user study revealed, some users are actually unwilling to provide any form of explicit feedback. They expect the system to learn without manually specifying likes and dislikes. Therefore implicit feedback is collected from user interactions via the user interface: consumption of a media item itself is regarded as a positive preference towards that item. If a user listens to a song, or watches a movie, the system will infer a positive relationship between that user and the item. Since the user interface serves as the aggregated entry point for media control in the home network, the consumption of media and all its available properties (e.g., duration of consumption, time of consumption, etc.) are easily logged for every user of the system.

To allow users control over their profiles as they are constructed by the system, the user interface also offers a user-specific history view. The history view lists all relevant feedback the system has collected about the user and will be used as input to the recommendation algorithms. In the history view, users can transparently see what information is gathered and unwanted entries can easily be deleted.

In conclusion, there are three ways in which the system tries to collect information about the user. Two types of explicitly providing likes and dislikes in combination with implicitly inferring information from media consumptions provide an adequate feedback framework to support both active and passive users of the system.

6.3 Contextual Recommendation List

When all user feedback is processed and recommendations are calculated (see Section 7 for details on the calculation process), a list with suggested items is available in the user interface. Recommended content items share the same visualization as normal content, providing the same information, similar items list, rating functionality and media control buttons. Since the user should not be overloaded with recommended items (as may be the case for the content items), the system selects the top-N (between 7 and 10, see further [6]) most interesting items for the set of currently active users.

When the active users change, the recommendation list instantly updates its items accordingly in real-time. In that way the system is capable of providing recommendations for single users (i.e., when only one user is indicated as active) as well as for groups of users (i.e., multiple active users). The recommended items list for groups of users aims to be a best estimation of the top-N items that will be liked by (and preferably not already consumed by) all the active users.

The user study indicated that, when people are deciding what movie to watch in group, not every member contributes equally to the final decision. Some users might be indifferent and do not really care what will be watched while others may have a really strong opinion, or are more knowledgeable about the media at hand. To be able to realistically model these situations, user weights were introduced in the system. When a new user account is created, an appropriate weight value (indicated as *importance*) can be set. Three importance weights are available in the user interface (Fig. 6): *low*, *medium* and *high* (indicated respectively by -, ? and +). These weights represent for each user how much its user profile should be taken into account when generating recommendations for groups of users. The availability of these user importance weights allows the system to adapt its recommended items list to very specific user situations (e.g., two parents and a child, or four friends of which one has an expert opinion on movies)

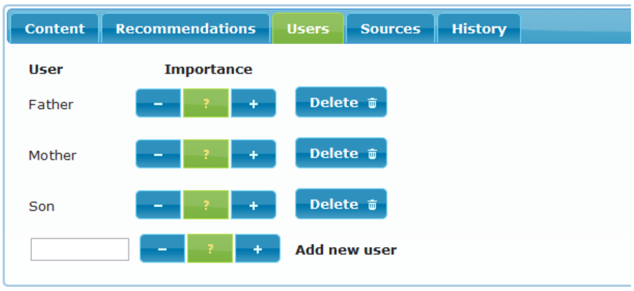


Fig. 6 The *Users* tab of the user interface. Every user in the OMUS system is associated with an importance factor that can be changed according to the desired context.

By manually changing the active users and their importance weights, the recommended items list can be influenced in real-time to make good suggestions for every possible user context. It turns out however (Section 3), that often users already have a specific genre or category in mind when searching for some media to consume. To enable this parameter in our model, genre filters were introduced into the system. The genre filter allows to restrict the recommendation list to items of the indicated genre. This feature enlarges even further the involvement of the user in its recommendation list. Together with the ability to change the active users and set their importance weights, users are able to provide a fine grained context situation to which the recommender system can specifically tailor its suggestions.

By instantly updating the list when a context parameter is changed, we achieve a feeling of real-time interactivity between the user and the system, which may boost user engagement and in the end can lead to higher quality recommendations. This real-time behavior is enabled by precalculating recommendation lists for all possible contexts via a cloud-based recommender system as detailed next.

7 Recommender System

The recommendation list provides users with a selection of the top-N items ([6] hints that a length between 7 and 10 is ideal for offering a good variation while keeping the list manageable for the users) that the system estimated to be most appropriate for the given context of active users, importance weights and possible genre filters. This section describes how the recommender system composes that selection list, by processing the collected user feedback and taking into account the various context situations while providing real-time interactivity with the user.

7.1 Preprocessing Feedback

The feedback collected as detailed in Section 6 will serve as input to the recommendation algorithms of the recommender system. Before the feedback can be used, it must first undergo a series of preprocessing steps. Three distinct types of feedback can be collected via the user interface: explicit binary feedback (i.e., like/dislike) on the content items, explicit binary feedback on specific content item attributes

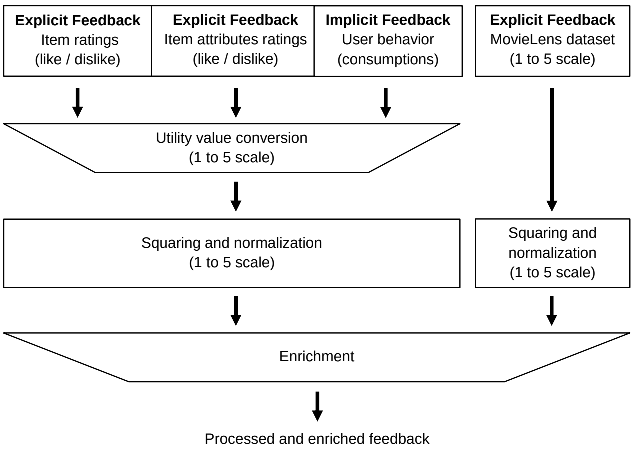


Fig. 7 The different steps of the feedback preprocessing. Feedback is collected in a number of different ways and enriched by means of an external dataset. The preprocessing aligns all feedback and compensates for effects like different scales and the non-linear use of ratings.

(e.g., genre), and implicit feedback in the form of consumption data (e.g., "this movie has been watched by this user"). These distinct types of feedback are linearly transformed and combined into a utility value indicating the overall appreciation level of a user for an item on a scale from 1 (not interesting) to 5 (very interesting).

To alleviate the *cold start* problem that often occurs in recommendation algorithms, the user feedback in the system is further enriched by importing the publicly available MovieLens (100K) dataset[7]. This dataset contains information about 1682 popular movies, including 100 000 ratings (on a 1 to 5 scale) of 943 users. The resulting enriched feedback enables e.g., recommendation algorithms based on collaborative filtering to compare users of the system to existing movie viewers from MovieLens and take this additional community knowledge into account with a potential increase of recommendation accuracy as a result.

The ratings are normalized to compensate for the effect that distinct users may show different rating tendencies. This effect was first noted by O'Connor *et al.* [28] where a test subject remarked that the PolyLens group recommendation system did not take into account that, e.g., "Mark's *5* is Dan's *3* ".

Finally, after the normalization, the ratings are squared to incorporate the effect of non-linear use: research has shown that ratings are not used linearly, and squaring the ratings yields more accurate results [23]. E.g., the difference between a 4-star and a 3-star rating is less significant than the difference between a 5-star and a 4-star rating. Figure 7 graphically overviews the complete feedback preprocessing.

7.2 Group Recommendation Strategy

Since recommendations have to be generated for any set of active users, a group recommendation strategy is needed. Berkovsky *et al.* [5] evaluated a hybrid switch-

[7] http://www.grouplens.org/node/73

ing approach of the group-based *aggregated models* and *aggregated predictions* strategies for their family-based recipe recommender. The accuracy of the strategy was superior to all individual recommendation strategies. They suggest a hybrid switching strategy that applies *aggregated predictions* when the density of user data is low (i.e., few ratings are available for every user) and *aggregated models* otherwise. The exact switching threshold in terms of number of ratings available can be optimized using the mean absolute error (MAE) (predicted ratings vs. real ratings) for a given dataset, as in [5]. The suggested hybrid switching strategy was integrated in the OMUS recommender system.

The strategy requires a way of aggregating data which will be either the user feedback (i.e., ratings) or individual recommendations of multiple users. For this, multiple approaches are available, as discussed in Section 2. In the OMUS system, a variant of the *Average Without Misery* approach was implemented. Research has shown this strategy to be fairly similar to how a group of people intuitively come to a group decision [23]. Also it prevents individual misery which was another attribute of group recommendation most users appreciate.

The *Average Without Misery* strategy averages out individual values but leaves out values below a certain threshold (of misery). This approach is well fit for the OMUS recommender system because it enables the integration of individual user weights (see the importance weights in Section 6). Since averages are calculated over all members of the group, the individual weights of users can straightforwardly be taken into account to finally come to a *Weighted Average Without Misery* function.

The OMUS system allows any combination of active users to be selected and consequently groups are formed. These groups can be interactively defined while using the system and are therefore very ephemeral. At any time users can be added or removed from the group and the system should update its recommendation list accordingly in real-time to provide the user with a feeling of interactivity. To enable this real-time functionality, recommendation lists of all contexts (i.e., all combinations of users and importance weights) are precalculated and stored. This exhaustive recommendation calculation makes rapidly switching between contexts and reloading the relevant recommendation lists possible in real-time, but puts a heavy burden on the calculation infrastructure. The number of associated contexts $f(n, k)$, and therefore also the number of recommendation lists, in function of number of users n and number of importance weights k is:

$$f(n, k) = n + \sum_{i=2}^{n} \binom{n}{i} k^i \tag{1}$$

Fig. 8 puts this exponential formula into a graphical perspective (note that the y-axis is on a logarithmic scale) for an increasing number of users and importance weights. It is clear that in order to provide reasonable calculation times on state-of-the-art hardware, the number of users and/or weights that can be processed by the system will have to be limited. For eight users with three possible weight values (e.g., *low*, *medium*, and *high*), the system would have to precalculate 100K recommendation lists, instead of just 8 (i.e., one list per user) if no groups (and weights) were involved.

Up to 5 users, the number of different contexts (i.e., approximately 1000) seems reasonable for three weights values. Since the system is targeted at home environ-

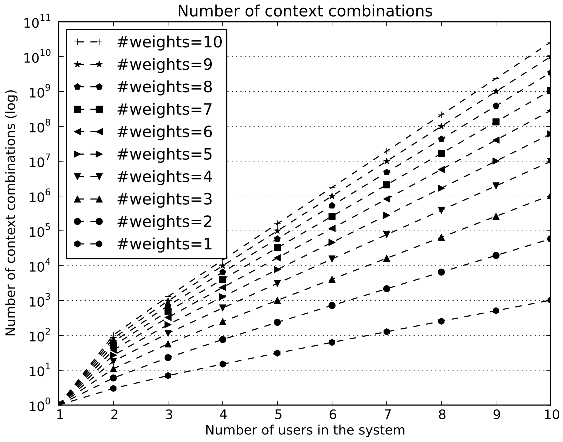


Fig. 8 A visual perspective on the number of different contexts that can be composed in terms of active users and user importance weights. Note that the y-axis has a logarithmic scale.

ments where family members or small groups of friends consume media together, the number of users is not expected to be very high. Furthermore, for large groups (more than 8 members), recommendation accuracy tends to decrease [1] [11], and thus also the potential usefulness of group recommendations.

In conclusion, if real-time context switching behavior is required, a trade-off will have to be made between the supported number of users with their available importance values and the required hardware infrastructure and calculation time.

7.3 Hybrid Recommendation Algorithm

Once user feedback is processed, the group recommendation strategy will require recommendations to be calculated for given users (either as group model, or for single users).

Since both the collaborative filtering (CF) and content-based (CB) recommendation approaches seem interesting for an in-home recommendation system, our OMUS system employed a hybrid combination of CF and CB. Thus, we combined the best of both worlds, while circumventing their respective weaknesses. CF is good at harvesting community knowledge such as ratings from other users, and when provided with sufficient feedback, high quality recommendations can be produced. The CB recommender on the other hand only needs information about the items themselves and is therefore capable of providing suggestions also for users with a very limited amount of ratings. Because of the user profiles it creates, a CB algorithm is perfectly suited to handle the item attribute feedback (e.g., “I don’t like that genre”) that can be collected through the user interface (see Section 6). Although the recommendations of the CB algorithm may lack some serendipity (i.e., surprisingly interesting items) and will not be very diverse, users may find it easier to understand why they are recommended and that increases the trust users have towards the system [13]. If needed, serendipity can always be boosted by integrating an extra module that tries to increase the variability of the recommendations as done by the diversity machine sub-service in [2].

As detailed in the previous subsection, for every context (i.e., set of users with their importance weights) recommendations will have to be calculated. To come to a hybrid recommendation list, both algorithms (CB and CF) calculate the utility value for every item (not previously rated or consumed by a member of the group) in the system. The highest scoring items for the given context from both algorithms are subsequently interleaved into a final hybrid top-N (cf. supra) recommendation list.

In the end, the goal of the recommendation system is to provide users with an intelligent view on their own content such that for every context, the best suited content items are shown. Preferably as much content items as possible are reachable (i.e., show up in some recommendation list) to provide the users with a certain sense of variety and for example not recommend the 5 same popular items over and over again. This amount of reachable (i.e., recommendable) items in the system is often referred to as the *item space coverage* or *catalog coverage* of the recommender system [32] and can be defined as the percentage of all items that can be recommended.

We captured and compared the influence of various design choices on the catalog coverage of the system, in an experiment, calculating recommendations repeatedly for three different context situations: *Users*, *Groups* and *Weights*.

Users defines the situation where every user has a single recommendation list. Users can not be grouped together and therefore also no importance weights are available. With n users, n distinct recommendation lists will be available (i.e., one for every user).

Groups is the situation where any combination of users can be grouped together, but every user has the same importance weight. With n users, the number of distinct recommendation lists is (2).

$$f(n) = \sum_{i=1}^{n} \binom{n}{i} \tag{2}$$

Weights is the context situation where every user can be grouped and 3 individual importance weights can be set. With n users, the number of lists is given by (1) for $k = 3$.

The experiment was repeated for each of these three context situations with the CB algorithm, the CF algorithm and the hybrid CB+CF algorithm and finally also the genre filter was taken into account, since it also influences the number of recommendation lists and possibly the coverage. In total, coverage values for 18 different context situations (9 with genre filter and 9 without) were compared in the experiment. The MovieLens (100K) dataset was used to provide ratings and movie data. As user base, 4 users were uniformly, random sampled from the 943 MovieLens users, and recommendation lists (of length 8) according to the context situation were calculated. For every context situation the catalog coverage was determined by counting the number of distinct recommended items over all recommendation lists for that context (where each list contained N=8 items). The coverage results (Fig. 9) were averaged out over 10 runs (with 95% confidence intervals) and are presented in percentages of the total number of items available in the system. For each context situation, the maximum coverage will be limited

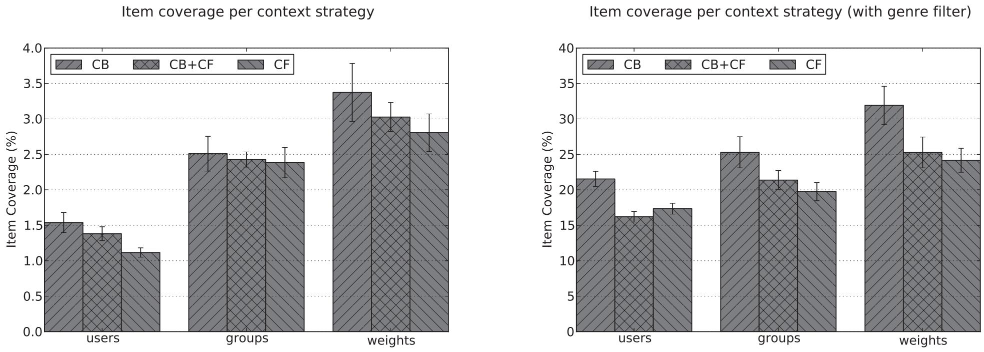


Fig. 9 The catalog coverage (i.e., amount of reachable items through recommendation) for different contexts and algorithms without (left) and with genre filter (right) simulated by the OMUS system for 4 users.

by the number of recommendation lists available in that situation. For the *Users* situation (without genre filter) the OMUS system will calculate 4 recommendation lists (i.e., one for every user) of 8 items. Maximum coverage for this situation is reached if all 32 recommended items are distinct. We had 1586 available items in our experiment and so the maximum coverage would then be 2% ($\frac{32}{1586}$). Similarly the maximum coverage in our experiment for *Groups* and *Weights* (without genre filter) can be determined to be 7.6% and 100% (catalog coverage can not be higher than 100%) respectively.

From Fig. 9 it is clear that there is a difference between coverage values with and without the genre filter option (about 10 times higher with genre filter). The MovieLens dataset comprises 18 different genres, each of which can be filtered on. For every genre an extra recommendation list will be available (comprising only items of that genre) resulting in (18 times) more recommendation lists and thus higher coverage values for context situations with genre filter. Because items can be associated with multiple genres, the different genre recommendation lists are not necessarily distinct and overlap may occur. This overlap and the fact that some genres are so specific that no 8 items could be recommended are the reasons that the enabled genre filter functionality performs only 10 times (instead of 18) better than without genre filter.

For each algorithm, the context situation *Users* has the lowest coverage followed by the *Groups* context and the *Weights* context respectively (always statistically significant on a significance level of 0.05, except for groups vs. weights for the CF algorithm without genre filter). This was to be expected considering they increasingly provide more recommendation lists.

Regarding the distinct algorithms and their coverage values, a trend can be noted. Overall CB comes out with highest coverage values (statistically significant with genre filter). It is known that CB can reach higher coverage than a CF algorithm [3], because the latter tends to recommend items that are somewhat more popular and this often leads to less diverse recommendation lists.

For our system, the hybrid approach (i.e., CB+CF) seems a valid choice where both the community knowledge and item attribute information can be leveraged to recommend items that are both highly interesting and seem familiar enough to gain the trust of the user. An additional benefit that we believe such a hybrid algorithm may offer, is the ability to provide recommendations in more situations than CB and CF separately. Although the dataset used in our experiments was a dense dataset with sufficient information about items and users, this may not always be the case. CF might not find any overlapping users or items because of insufficient user feedback while the CB approach can fail due to a lack of item information or simply no similar items were found. Therefore the hybrid CB+CF algorithm with the high coverage values coming from the weighted context and genre filtering abilities offers the most complete and fail-safe solution for this content discovery system for home environments.

7.4 Item Similarity

As discussed in Section 6, similar items can be shown in the specific item view. These similar items come from the original content items pool and are calculated for every item. This calculation does not require any additional computational resources, as similarities are a byproduct of both the CF and the CB recommendation algorithms. Just as with the recommendations, similar items from both algorithms are straightforwardly interleaved. This allows CB to introduce similar items based on item attributes (i.e., similar in genre, director, or actor) like for example "The Matrix" and "The Matrix Reloaded" while CF can insert less obvious but community inspired similar items.

A downside of restricting the possible similar items to items available for the user, is that the quality of the similar items lists is greatly dependent on the size and properties of the original content pool. With too few items available, a top-N similar items list may not be very sensible or be of high quality. In that case, it may be better to simply not show any similar items with an estimated quality below a certain threshold.

7.5 Synchronization Strategy

The calculation of recommendations is typically a very computationally intensive task, requiring high-performant infrastructure in terms of both memory and processing power. Especially when multiple algorithms are run and recommendation lists for potentially thousands of different contexts must be generated because of the exhaustive processing of all possible groups of users with all of their possible importance factors. For testing purposes, a server with an Intel Core 2 Quad CPU (model Q650 clocked at 3 Ghz) with 8 GB of RAM memory and a 7200 RPM HDD was used to precalculate all recommendation contexts for 4 uniformly, random sampled users of the MovieLens dataset with 3 importance weights. The average recommendation time (out of 10 runs) for this experiment, was 27 minutes with (with an observed minimum of 11 and maximum 39 minutes). Although this is an acceptable computation time for the precalculation of all the contexts, it is unrealistic to assume the necessary hardware infrastructure to support this computing task to be available in an average home network. Therefore, recommendation functionality is offloaded to an external (i.e., outside the home network) recommendation service. This service then implements the recommendation strategies as described in the previous subsections. Because the recommendation functionality resides outside the home network, recommendation quality may be further improved by leveraging across-the-homes data as additional feedback enrichment for the recommendation algorithms.

The external recommendation calculation service needs to synchronize available data with the data storage present in the home network (through the Sync Logic component). Both user data and item data need to be exchanged in the synchronization process to provide the service with the necessary input. When recommendation lists are generated, they can be downloaded back to the home network together with similar items data.

The synchronization process can be triggered by the availability of new data or set to a fixed time interval. In the current setup, the OMUS system (i.e., the external recommendation calculation service component) regenerates the recommendations from scratch when new data becomes available. Future work could integrate more intelligent incremental recommendation algorithms to eliminate this behavior and speed up the recommendation calculation times. A trade-off will need to be made between real-time recommendations (i.e., new data or feedback is instantly uploaded and processed) and communication overhead between the external service and the home network.

8 Comparison With Existing Home Information Systems

In this section we compare our OMUS system with some existing information systems, some of which users have mentioned during the focus group interviews described in Section 3. Table 2 lists the requirements for a home information system as dictated by the users in our user study. For each system the offered functionality is indicated.

Table 2 OMUS functionality comparison with existing home information systems.

Functionality	YouTube	iTunes	XBMC	OMUS
Aggregation of home content	no	✓	✓	✓
Support wide variety of devices	✓	no	✓	✓
Easy playback of content on a device	✓	no	✓	✓
Contextual user preferences	no	no	no	✓
Group recommendation	no	no	no	✓
Support different types of users (active / passive)	✓	✓	no	✓
Optional explicit user profile	✓	no	no	✓
Control over user profile	no	no	no	✓

YouTube[8]**.** With the rise of 'smart TVs' capable of running external applications and streaming content from the web, the video sharing site YouTube has entered the home environment as a medium for online video entertainment. While YouTube is supported on almost every web-enabled device, we find that it lacks some basic features to serve as a central home information system. The most important one being the lack of content aggregation. While YouTube does offer easy

[8] http://www.youtube.com

accessibility and some degree of personalization, it only streams online videos and is incapable of incorporating content from the home network.

iTunes[9]**.** iTunes has long been a poplular choice of users for storing and managing their music libraries. It offers content aggregation and some degree of personalization but no support for group recommendation or contextual input. iTunes also does not offer any explicit control over user profiles and playback is supported for only a limited set of devices.

XBMC[10]. XBMC is an open source software media player, designed specifically for the home environment. While we find that it handles content very well on a wide variety of devices, it does not offer any personalization features.

Comparing the features offered by these systems, we notice that, as was the case for the related work (Section 2), there is, besides OMUS, no single all-round home information system that meets all the requirements. Therefore, users are often forced to choose between managing their home network content and having a rich personalized experience.

9 Conclusion

In this paper we proposed the OMUS system as an information system designed specifically towards the needs of users in a home environment. A user study made it clear that home users struggle with technical issues as content scattering, and media playback on preferred devices. They are also often overwhelmed by the huge amount of media content that is available for consumption in the home network. These problems were met by integrating three main components in the OMUS system: content aggregation, a recommender system and an overall user interface.

The content aggregation component allowed to build a virtual library of available content located on distinct (DLNA-based) devices in the home network. For the enrichment of item metadata, IMDb information, as well as any other online source can be easily integrated because of the plugin-based internal architecture. Through a number of performance analysis experiments, a chunk size in the range of 10-30 was determined best for lowest average (DLNA) browse times (across 3 different DMSes), while parallelization of browse requests to a single DMS device brings a limited speedup (total browse time reduction in the range of a few percent for WMP and PowerBay, order of 11% for minidlna).

We demonstrated how existing recommendation algorithms and strategies can be implemented into a group-based, contextual recommender system fit for a home environment. We proposed a hybrid approach combining collaborative filtering and content-based recommendation. Because of effective precalculation and synchronization, recommendation lists are instantly updated when users change context or group parameters. A downside of the dynamically updating recommendation lists, was the computational burden on the recommendation infrastructure and so the users in the system will have to be limited to a reasonable number (8 or less).

The results of the content aggregation and the recommender system component were integrated in an overall user interface. We demonstrated how the various and sometimes very diverse user requirements were met, while maintaining an

[9] http://www.apple.com/itunes

[10] http://xbmc.org

easy to use interface that encouraged active user involvement with the OMUS system. The system is able to provide recommendations for individual users and groups, where in the latter case we can give each user a distinct weight. We showed that by varying the weights, the coverage (i.e., items that can be returned by the recommender) considerably increases. Adding genre filter functionality further boosted the coverage.

The OMUS system can be implemented in a hardware component in the home network (e.g., a home gateway), integrated into an existing software media center (e.g., XBMC), or serve as a stand alone home information system.

Acknowledgements This work was supported by the iMinds/OMUS project. The described research activities were funded by a PhD grant to Simon Dooms of the Agency for Innovation by Science and Technology (IWT Vlaanderen), and a PhD grant to Toon De Pessemier of the Research Foundation - Flanders (FWO–VL). C. Develder was supported (in part) as post-doctoral fellow of FWO–Vl.

References

1. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proc. 4th ACM Conf. Recommender systems (RecSys 2010), pp. 119–126. ACM, New York, NY, USA (2010)
2. Bellandi, V., Ceravolo, P., Frati, F., Maggesi, J., Waldhart, G., Seeber, I.: Design principles for competence-based recommender systems. In: in Proc. 6th IEEE Conf. Digital Ecosystems Technologies (DEST 2012), pp. 1–6. Campione d'Italia, Italy (2012)
3. Bellogín, A., Cantador, I., Castells, P.: A study of heterogeneity in recommendations for a social music service. In: Proc. 1st Int. Workshop Information Heterogeneity and Fusion in Recommender Systems, HetRec '10, pp. 1–8. ACM, New York, NY, USA (2010)
4. Bennett, J., Lanning, S.: The netflix prize. In: Proc. KDD Cup Workshop 2007, pp. 3–6. ACM, New York (2007)
5. Berkovsky, S., Freyne, J.: Group-based recipe recommendations: analysis of data aggregation strategies. In: Proc. 4th ACM Conf. Recommender systems (RecSys 2010), pp. 111–118. ACM, New York, NY, USA (2010)
6. Bollen, D., Knijnenburg, B.P., Willemsen, M.C., Graus, M.: Understanding choice overload in recommender systems. In: Proc. 4th ACM Conf. Recommender Systems (RecSys 2010), pp. 63–70. ACM, New York, NY, USA (2010)
7. Brewka, L., Skoldstrom, P., Nelis, J., Wessing, H., Develder, C.: Automatic provisioning of end-to-end QoS into the home. IEEE Trans. Consumer Electronics **57**(4), 1670–1678 (2011)
8. Van den Broeck, W.: From analogue to digital: the silent (r)evolution? a qualitative study on the domestication of interactive digital television in flanders. Ph.D. thesis, Vrije Universiteit Brussel, Brussels (2011)
9. Burke, R.: Knowledge-based recommender systems. Encyclopedia of Library and Information Science **69**(32) (2000)
10. Cho, S.Y., Seo, D.Y., Kim, T.Y.: Gateway framework for home appliance's interoperability based on heterogeneous middleware in residential networks. In: Proc. IEEE Int. Conf. Consumer Electronics (ICCE 2002), pp. 98 – 99 (2002)
11. De Pessemier, T., Dooms, S., Martens, L.: Design and evaluation of a group recommender system. In: Proc. 6th ACM Conf. Recommender systems (RecSys 2012), RecSys '12. ACM, New York, NY, USA (2012)
12. Develder, C., Lambert, P., Van Lancker, W., Moens, S., Van de Walle, R., Nelis, J., Verslype, D., Latr, S., Staelens, N., Vercammen, N., Vermeulen, B., Masschelein, B., Van Leeuwen, T., Macq, J.F., Struyve, K., De Turck, F., Dhoedt, B.: Delivering scalable video with QoS to the home. Telecommun. Syst. **49**(1), 129–148 (2012)
13. Dooms, S., De Pessemier, T., Martens, L.: A user-centric evaluation of recommender algorithms for an event recommendation system. In: UCERSTI workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces - 2. Chicago, IL, USA (2011)

14. Guedes, A., Santos, D., do Nascimento, J., Sales, L., Perkusich, A., Almeida, H.: Set your multimedia application free with BRisa framework: An open source UPnP implementation for resource limited devices. In: Proc. Consumer Communications and Networking Conference (CCNC 2008), pp. 1257 –1258 (2008)
15. Jameson, A., Smyth, B.: The adaptive web. chap. Recommendation to groups, pp. 596–627. Springer-Verlag, Berlin, Heidelberg (2007)
16. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems An Introduction. Cambridge University Press (2011)
17. Jeronimo, M., Weast, J.: UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play. Intel Press (2003)
18. Kang, D.O., Kang, K., Choi, S.G., Lee, J.: UPnP AV architectural multimedia system with a home gateway powered by the OSGi platform. In: Proc. IEEE Int. Conf. Consumer Electronics (ICCE 2005), pp. 405 – 406 (2005)
19. Kawamoto, E., Kadowaki, K., Koita, T., Sato, K.: Content sharing among UPnP gateways on unstructured P2P network using dynamic overlay topology optimization. In: Proc. Consumer Communications and Networking Conference (CCNC 2009), pp. 1 –5 (2009)
20. Kim, D.S., Lee, J.M., Kwon, W.H., Yuh, I.K.: Design and implementation of home network systems using UPnP middleware for networked appliances. IEEE Trans. Consumer Electronics **48**(4), 963 – 972 (2002)
21. Kim, J.T., Oh, Y.J., Lee, H.K., Paik, E.H., Park, K.R.: Implementation of the DLNA proxy system for sharing home media contents. In: IEEE Trans. Consumer Electronics (ICCE 2007), pp. 1 –2 (2007)
22. Lee, H.Y., Kim, J.W.: An approach for content sharing among UPnP devices in different home networks. IEEE Trans. Consumer Electronics **53**(4), 1419 –1426 (2007)
23. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction **14**(1), 37–85 (2004)
24. McCarthy, J.F., Anagnost, T.D.: Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In: Proc. 1998 ACM Conf. Computer supported cooperative work, CSCW '98, pp. 363–372. ACM, New York, NY, USA (1998)
25. Mobasher, B.: Recommender systems. In: Kunstliche Intelligenz, Special Issue on Web Mining, vol. 3, pp. 41–43. Bottcher IT Verlag, Bremen, Germany (2007)
26. Nelis, J., Verschueren, T., Verslype, D., Develder, C.: DYAMAND: DYnamic, Adaptive MAnagement of Networks and Devices. In: in Proc. 37th IEEE Conf. Local Computer Netw. (LCN 2012). Clearwater, Florida, USA (2012)
27. Nelis, J., Verslype, D., Develder, C.: Intelligent distributed multimedia collection: Content aggregation and integration. In: Proc. 36th IEEE Conf. Local Computer Networks (LCN 2011), pp. 203–207. Bonn, Germany (2011)
28. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: Proc. 7th Eur. Conf. Computer Supported Cooperative Work, ECSCW'01, pp. 199–218. Kluwer Academic Publishers, Norwell, MA, USA (2001)
29. Purcell, K.: The state of online video. Tech. rep., PewResearchCenter (2010)
30. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proc. ACMF Conf. on Computer supported cooperative work, CSCW '94, pp. 175–186. ACM, New York, NY, USA (1994)
31. Sales, T.B.M.d., Sales, L.M.d., Pereira, M., Almeida, H., Perkusich, A., Gorgônio, K., Sales Jr., M.A.d.: Towards the UPnP-UP: Enabling user profile to support customized services in UPnP networks. In: Proc. 2nd Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '08, pp. 206–211. IEEE Computer Society, Washington, DC, USA (2008)
32. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–297 (2011)