



Semester: 4

Project Period: Autumn 2012

Finished: 11/01/2013

Copies: 3

Pages: 115

Personalized Interactive EPG based on HbbTV

Abstract:

EPG was envisioned as a vital service in digital TV. However the EPG in its current form has not been a success. People continue zapping to select and change programs. The prototype developed during this master thesis uses web-services and recommender techniques to increase the personal benefit of watching television. The main focus is on a personalized interactive electronic program guide with an appealing user interface. To achieve this, state-of-the-art for recommender and profiling systems are studied and user-centric design principles are used to formulate the requirements.

Project group no.: 4.4

Members:

Mohammad Homayoon Fayez

Student no: 20101104

Padam Chhantyal

Student no: 20110511

.....

.....

AAU Supervisor: Reza Tadayoni

Table of Contents

1.	Introduction	7
1.1	Background and motivation.....	7
1.2	Problem Formulation	8
1.3	Methodology.....	9
1.3.1	Project Planning	9
1.3.2	Desktop research.....	10
1.3.3	Software process model	10
1.3.4	User-centric design	11
1.4	Structure of the report	12
2.	HBBTV.....	13
2.1	MHEG.....	13
2.1.1	Application Signalling.....	13
2.1.2	Interactivity.....	14
2.2	MHP.....	14
2.3	HbbTV	15
2.3.1	Hybrid terminal.....	16
2.3.2	CEA-2014.....	18
2.3.3	DAE.....	20
2.3.4	DVB (TS 102 809).....	20
2.3.5	Signalling interactive application and services	21
2.3.6	Applications lifecycle	22
2.3.7	AIT.....	23
2.3.8	Application transport	23
2.3.9	Security	24
3.	Personalization	25
3.1	User Profile	27
3.1.1	User identification.....	28
3.1.2	User information collection	29
3.2	Content and metadata	31
3.2.1	MPEG-7	31
3.2.2	TV-Anytime	35
3.3	Recommender Systems	42
3.3.1	Content Based Filtering.....	44
3.3.2	Collaborative Filtering	46
3.3.3	User-based CF algorithms	47
3.3.4	Item-based CF algorithms.....	50
3.4	Summary of the personalization chapter	52
3.4.1	Collecting user information.....	52
3.4.2	Content and metadata	53
3.4.3	Recommendation Techniques	54
4.	Prototype development.....	55

4.1	User interface	56
4.1.1	Low fidelity prototypes.....	56
4.1.2	Existing EPGs.....	56
4.1.3	Appealing EPG	57
4.1.4	From low fidelity to High fidelity	59
4.1.5	Final version.....	60
4.2	System requirements	61
4.2.1	IRT broadcast server	61
4.2.2	ASI output cards	63
4.2.3	Function requirements	63
4.2.4	Non-functional Requirements.....	64
4.3	Use case diagram	65
4.4	Database Design.....	66
4.5	Class Diagram.....	67
4.5.1	Class responsibility	67
4.6	Sequence diagram	68
4.7	Implementation	69
5.	Testing	72
5.1	Black box tests.....	72
5.2	Usability test	73
5.2.1	Feedback.....	74
6.	Conclusion	76
7.	Future work	78
8.	Appendices	79
8.1	Requirements tests	79
8.2	Use case specification	84
8.3	Class diagram	96
8.4	Class responsibilities	97
8.5	Milestone Plan	99
8.6	Action Item List.....	100
8.7	Usability test feedback	100
8.8	Black box tests.....	101
8.8.1	Mysettings page tests	101
8.8.2	Tests carried out in the TV emulator.....	102
8.8.3	Tests carried from browser	106
8.9	IRT Broadcast server configuration.....	110
8.10	AIT configuration.....	111
9.	Bibliography	112

Figure list

Figure 3: Structure of a hybrid terminal [16].	16
Figure 1: The Specification overview	18
Figure 2: Application metadata.	21
Figure 4: Key elements required for personalization	27
Figure 5: MPEG -7 main elements [29]	32
Figure 6: Role of mpeg-7 in facilitating interoperable services and applications [31]	33
Figure 7: Broadcast model without rights management protection [34]	36
Figure 8: Metadata and content flow	38
Figure 9: CRID Location resolution [37]	41
Figure 10 : Metadata that references a programme CRID [35]	41
Figure 11: prototype overview	55
Figure 12: Example of existing EPGs	57
Figure 13: Ideas for initial paper prototype	57
Figure 14: First EPG paper prototype	58
Figure 15: EPG section	58
Figure 16: MyPage (paper prototype)	59
Figure 17: MyPage	59
Figure 18: Login page	61
Figure 19: EPG section with channel DR 1 selected	61
Figure 20: MyPage	61
Figure 21: EPG section with a program from channel TV2 selected	61
Figure 22: Use case - HbbTV service	65
Figure 23: Use case - EPG	65
Figure 24: Use case – mypage	65
Figure 25: Use case – mysettings	65
Figure 26: database, ER diagram	66
Figure 27: sequence diagram, smart EPG	69
Figure 28: frontend, mysettings	70
Figure 29: frontend, mypage	71
Figure 30: class diagram	96
Figure 31: mysettings login page	101
Figure 32: create wish list	102
Figure 33: add/remove channels	102
Figure 34: show red button	102
Figure 35: show services	103
Figure 36: select users	103
Figure 37: login user	103
Figure 38: smart EPG	103
Figure 39: mypage film loading	104
Figure 40: mypage series selected program	104
Figure 41: mypage underholding loading	105

Figure 42: mypage wished programs.....	105
Figure 43: Sony Bravia HbbTV 1.1.1.....	106
Figure 44: IRT broadcast server configuraion.....	110
Figure 45: AIT configuration	111
Figure 46: AIT table	111

Acknowledgement

We would like to send our special thanks to our supervisor Reza Tadayoni associate professor at Aalborg University Copenhagen for the guidance and support during the entire project.

We would like to thank Ove Færgemand project director from YouSee A/S for providing research materials, broadcast server and access to the broadcasted program metadata. Our thanks go as well to Per Fjællegaard who participated in the meetings and gave us good suggestions.

Thanks to Allan Hammershøj COO at mediathand and Jannick Sørensen Assistant Professor at Aalborg University for sharing their ideas with us.

We would also like to thank our friends and families for their time to test our prototype.

Preface

This Master thesis is realized by the collaboration of the department of ICTE at Aalborg University Copenhagen and YouSee A/S. The project started on 3rd September 2012 and ended on 11th January 2013. The source code together with this report in PDF format is delivered in a CD to Aalborg University's administration.

Abbreviations	
A/V	Audio/Video
AIT	Application Information Table
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ASI	Asynchronous Serial Interface
CE	Consumer electronics
CEA	Consumer Electronics Association
CE-HTML	Consumer electronics - Hypertext Markup Language
CSS	Cascading Style Sheets
CBF	Content Based Filtering
CF	Collaborative filtering
CRID	Content Referencing Identifier
DAE	Declarative Application Environment
DOM	Document Object Model
DSM-CC	Digital Storage Media – Command and Control
DVB	Digital Video Broadcasting
DDL	Description Definition Language
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
FR	Functional Requirement
GUI	Graphical User Interface
HBBTV	Hybrid Broadcast Broadband Television
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IRT	Institut für Rundfunktechnik
JSON	JavaScript Object Notation
LSA	Latent Semantic Analysis

MPEG	Moving Picture Expert Group
MHP	Media Home Platform
MHEG	Multimedia Hypermedia Information Coding Expert Group
NFR	Non Functional Requirements
OIPF	Open IPTV Forum
OITF	Open IPTV Terminal Function
PAT	Program Association Table
PMT	Program Map Table
PDR	Personal Digital Recorder
RS	Recommender System
SDT	Service Description Table
SI	Service Information
SSL	Secure Socket Layer
STB	Set-Top Box
TDT	Time and Date Table
TFIDF	Term Frequency Inverse Document
TS	Transport Stream
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1. Introduction

The Internet has evolved from being an Internet of computers to the Internet of things, where almost all the electronic devices are interconnected. This has expanded the domain of the web-services to other devices like smart phones, tablets, connected TVs etc.

Connected TV on one hand has provided the possibility of running broadcast independent services like Instant messaging, video on demand etc. On the other hand it has provided the opportunity of convergence between broadcast and broadband. This opportunity has been utilized by HbbTV which is a new concept of combining broadcast and broadband to harmonize the delivery of entertainment to the consumers through connected televisions and set-top boxes [1]. Historically television is not a personalized medium but it can now be realized through HbbTV. The convergence of broadcast and broadband in HbbTV and the interactivity and personalization features of the web-services provide the possibility of delivering personalized contents on television sets.

Personalization is one of the success criteria of the popularity of web-services which has been assisted by recommendation techniques. Personalized contents on TV will enhance the user experience. In this project an effort has been done to formulate requirements for the implementation of an interactive personalized EPG. To prove the concept a prototype is developed. The specifications for the user interface of the prototype are developed by following the user-centric design principles. For the personalization purposes, state of the art recommendation techniques, user-profiling, MPEG-7 and TV-Anytime standards are discussed.

1.1 Background and motivation

Digital broadband televisions provide the opportunity of connectivity to the broadband and internet services. This leads to an increase in the number of choices. To find the relevant stuff from huge archives is difficult and time consuming. Thus there is a need for some mechanism to search in the large content archives on behalf of the user.

Web-services on the Internet are already successfully assisting users by delivering personalized content based on the users' preferences. Companies like Amazon and

Netflix rely on personalized services to attract more customers by helping the users to find the items of their choice in huge archives. Traditional EPG cannot assist the user to surf on many pages of program guide with hundreds of channels and broadcasted programs. Thus there is a need for a personalized EPG that beside basic functionality such as navigation can build the user profile and recommend TV programs that best matches user preferences.

Research has been done on personalized EPGs and there exist implementations like Personalized EPG based on TV-Anytime Metadata [2], Architecture of a system for personalized EPG [3] and iFanzzy personalized EPG [4]. These and most of the other approaches basically rely on set-top-boxes which add one extra device on the already existing pile of electronic devices at home. Another problem with the existing personalized EPGs is that their user interfaces are very complex for television screens.

EPG was envisioned as a vital service in digital TV. However, the static nature of the traditional EPGs, the non-attractive and complex interface of the existing personalized EPGs seems to be the main reason that zapping still prevails.

Our contribution to this area is a personalized interactive EPG, interfaced with a web-service. This personalized interactive EPG will have a very simple and appealing user interface. This means only HbbTV enabled TV sets are needed to use the service.

1.2 Problem Formulation

Digital TV provides interactive services that were not possible with analog TV [5] but the personalization of watching television has not taken advantage of the mechanisms known from web-services. HbbTV provides the opportunity of interaction with a web-service, where TV sets, which are low interactive devices, can benefit from the high interactivity of web-services which can lead to personalization.

On the basis of above problem description we will study state-of-the-art for recommendation and profile systems for live TV content and from this formulate requirements for the implementation of a personalized EPG. The prototype implementation focuses on integration of program metadata and user profile to deliver personalized content on an appealing user-interface.

Study of the problem domain also directs our concerns to the following issues.

- Traditionally Zapping and non-interactive EPG have been the paradigms for exploiting TV-channels and their programs.
- Most often the broadcast stream, user-data and metadata (EPG, TV programs) are distributed. E.g. at content aggregators like YouSee and at broadcasters websites. This further arise the following issues.
 - Content discovery
 - Content presentation

Keeping in mind these issues our work will attempt to answer the following questions.

- How can the selection of optimal TV-programs, for a certain person, be supported by web-services implemented via HbbTV?
- How can a web-interface overlaying the traditional EPG (Electronic Program Guide) benefits the user?
- How can the broadcast streams, user-logins and metadata (EPG, TV and Film archive) be combined to offer a more personalized service to the user?

1.3 Methodology

In this thesis the prototyping method will be used because prototyping is cheap and fast and also because it could be used to collect user feedbacks to formulate the requirements for the personalized EPG. In the initial stage paper prototype will be used and eventually it will evolve into a software prototype through a series of iterations. To be able to develop a prototype that best meets the users' needs, user-centric design approach will be used. Furthermore software process models that best suits prototyping and that best supports user-centric-design approach will be used. The reasons for choosing these approaches are discussed in section 1.3.3 and 1.3.4.

1.3.1 Project Planning

To achieve the best possible results the project work has been divided in four categories namely research, analysis, implementation and testing. Projects are always limited by a time constraint, thus planning is very crucial for a successful project (Well begun is half done-*Aristotle*). In order to best utilize the available time span we use both Gant chart and Milestone plan. As we know that the milestone plans focuses on the major goals or objectives to be achieved, where the Gant chart defines

the activities to be carried out in order to reach the milestones [6]. Therefore in our plan, the milestones are further divided into small tasks/activities. The benefit of small tasks is that we will detect the time miscalculation at each activity and we will be able to take remedial actions quickly. Furthermore an Action Item List is used to show tasks and the responsible person of the task.



The detailed Milestone plan and action item list can be found in the [appendix. 8.5](#) and [8.6](#) respectively.

1.3.2 Desktop research

The main focus of the desktop research is the state-of-the-art for recommendation and profile systems. We use the following sources of information.

- Literature reading
- Journals from HbbTV consortium
- Secondary data like reliable journals on the web
- Materials from the lectures

1.3.3 Software process model

Software process models explain different approaches to software development [7]. After investigating different software process models it is concluded that both waterfall model and Evolutionary development can be used in our project.

Waterfall model breaks the process of system development into a number of sequential stages which are represented by separate phases such as requirements specification, design, implementation and testing [7]. Waterfall is not a simple linear model but iterations are involved during each stage. The stages overlap and information from one stage could be used in another e.g. problems with requirements can be identified during design and design problems can be found during coding. These iterations and reworks could go for a long time. Because of the time limitations for the project the iterations can be stopped at some point. Waterfall model also assumes pre knowledge of requirements.

In this project Waterfall model can be used only for the backend because the backend engine will be implemented after all the requirements are known. On the other hand

all the requirements for the GUI are not known in advance. Though we will realize requirements during the research phase but customer requirements can change due to the results of the usability tests at different development stages of the user interface.

The model that best suits the GUI is the Evolutionary Development, which is based on development of an initial prototype which is then developed to an adequate system through a series of versions. This is achieved by exposing the initial prototype to the users and the feedback is used to refine the product. Using the feedback from the customer and users, the next version is developed. This model with an exploratory approach, where one works with the users to explore their requirements, seems promising. Firstly because all the requirements for a personalized EPG are not known in advance, secondly because it goes hand in hand with the user-centric approach which will be used to know user requirements.

1.3.4 User-centric design

In user centric design the focus is on the users' needs and limitations for the wannabe product. By using this technique during the user interface design a designer can foresee how users are likely to use the product and also to test the designers own assumptions.

The connectivity of television to broadband introduces new challenges to the design of user interface. For example navigation in a TV application using a remote control is not as easy as using a mouse for a PC application. A designer must use innovative methods to make the navigation and usage of the application easy. To detect the problems associated with navigation and completing a task using interactive EPG, the users will be involved in the design of the software from the very beginning. For this project an initial paper prototype will be developed and cooperative evaluation technique, which is one of the methods of the user centric design, will be used to detect the possible usability problems.

Monk et al. in "improving your human-computer interface" [8] describes the cooperative evaluation as a technique that improves a user interface specification by detecting possible usability problems in an early stage of a prototype. Through cooperative evaluation data about usability problems of a software product can be obtained. This feedback can be used in an iterative design-evolution-redesign cycle to improve the product. Cooperative evaluation could also lead to changes in initial requirements.

1.4 Structure of the report

Apart from the introduction that contains the background, motivation, problem formulation and methodology sections, the report is composed of 7 chapters. A short overview of the chapters is given in the following.

CHAPTER I - HBBTV

This chapter explains Hybrid Broadcast Broadband TV according to HbbTV specifications. The technical requirements for the implementation of HbbTV services and the terminal components of HbbTV are also explained in this chapter. Moreover the technologies and standards necessary for the realization of HbbTV services are described as well.

CHAPTER II - Personalization

This chapter describes how to deliver personalized contents. The key elements of a personalized service i.e. user profiles, content and metadata and recommendation techniques are discussed. This chapter also formulates the requirements for the implementation of a personalized EPG.

CHAPTER III - Prototype Development

The steps taken to develop a prototype based on the requirements realized in the personalization chapter are explained here. The steps describe in this chapter are requirements, design and implementation of the system.

CHAPTER IV - Testing

This chapter includes the results of the usability and black box tests.

CHAPTER V - Conclusion

This chapter concludes the project.

CHAPTER VI - Future work

In this chapter the future perspectives for the further possible improvements of the prototype are mentioned.

2. HBBTV

The European Broadcasting Union (EBU) [9] considers a Hybrid Broadcast Broadband TV as a device where it is possible to “... *combine the functionalities of both broadcast and broadband tuners...*” to offer a seamless experience to the consumer.

Interacting with TV is not a new idea; it was there many decades ago. The most famous and wide spread one is the teletext, developed in UK in 1974 [3]. The teletext added a value to the broadcast content and allowed the viewer to get different information, made available by the broadcaster, using the TV set whenever he/ she wanted it. With the evolution of the communication technology the user need for interactivity grows alongside with the demand for access to information and different content from any device at any time. To follow up with user needs, standards such as MHEG-5, MHP/GEM and HbbTV, have been developed over the years.

In this project we will focus on HbbTV because it is an open standard, built from the components of mature standards and browser-based interactive TV applications etc. Before going into the depths of HbbTV its predecessors MHEG-5, MHP/GEM will be discussed.

2.1 MHEG

MHEG stands for the Multimedia Hypermedia Information Coding Expert Group and is an ISO standard published in 1997. The purpose of the standard was to define a method of storage, exchange and display of multimedia presentations [10]. MHEG was not dedicated to TV, but a more general platform that can be used in CD-ROM-based encyclopedia and interactive books for example.

Even if the standard consists of eight parts – MHEG-1 to MHEG-8, only the fifth part MHEG-5 – is widely used (MHEG-6, MHEG-7 and MHEG-8 are considered as extensions to MHEG-5) in the interactive TV domain.

MHEG-5 Application signalling and Interactivity

2.1.1 Application Signalling

MHEG-5 applications and related data are loaded from the data carousel of the broadcast stream. To be able to run these applications, the receiver must know about

their existence as well as how to manage them, this mechanism is referred to as Application Signalling. MHEG-5 leaves application signalling to specific profiles to specify it. The ETSI profile uses the PMT of the MPEG-2 TS and a Service Gateway object, acquired from the file system delivered by an object carousel, to implement it [11].

2.1.2 Interactivity

MHEG-5 Interaction Channel (MHEG-IC) enables connected TV to interact to broadcast services via an IP connection [12]. MHEG-IC is also adopted in the ETSI standard. MHEG-IC supports the standard protocols normally used to deliver the web content as TCP-IP and HTTP; and makes use of a special “web browser” to present HTML applications.

2.2 MHP

MHP, or the Multimedia Home Platform, is the collective name for a compatible set of middleware specifications [13] developed by the MHP group, a sub group of DVB, between 1997 and 2000.

The MHP standard defines these profiles:

- Enhanced Broadcasting profile (Profile 1): The simplest version of an MHP environment supports this profile. It is aimed for STBs without a return channel in a low-cost area. In this case, applications may only be downloaded from the broadcast stream – the MPEG-2 TS. [14]
- Interactive Broadcasting profile (Profile 2): Same as Profile 1 but includes support for a standardized return channel that allows an interaction between the audience and the broadcast.
- Internet Access Profile (Profile 3): In this profile, Profile 2 is extended with support for Internet applications making it possible to combine Internet-based services, like email and web content, with the broadcast world.
- IPTV profile (Profile 4): This profile integrates support for DVB-IPTV into MHP.
- MHP Interactivity and Application signalling

MHP supports many communication protocols; some are TV specific as DVB-SI, DSM-CC and MPEG-2-TS, while others are network protocols such as IP, TCP, UDP

and HTTP. The main way to get broadcast-related applications remains the DSM-CC object carousel, as the return channel is not mandatory for all MHP profiles. In absence of a return channel only light interactive applications and services can be carried in the broadcast stream e.g. teletext and EPG. Depending on the supported MHP profile, services may range from simple interactivity (as voting in a live program) via the return channel (Profile 2) to online games (Profile 3) and catch-up TV and VOD (Profile 4).

GEM (Globally Executable MHP) is a subset of MHP standard specification. It enables deploying interactive applications over broadcast and broadband networks and for Blu-ray discs as well. It is based on Java and offers a platform independent application execution environment [15].

MHP interactivity

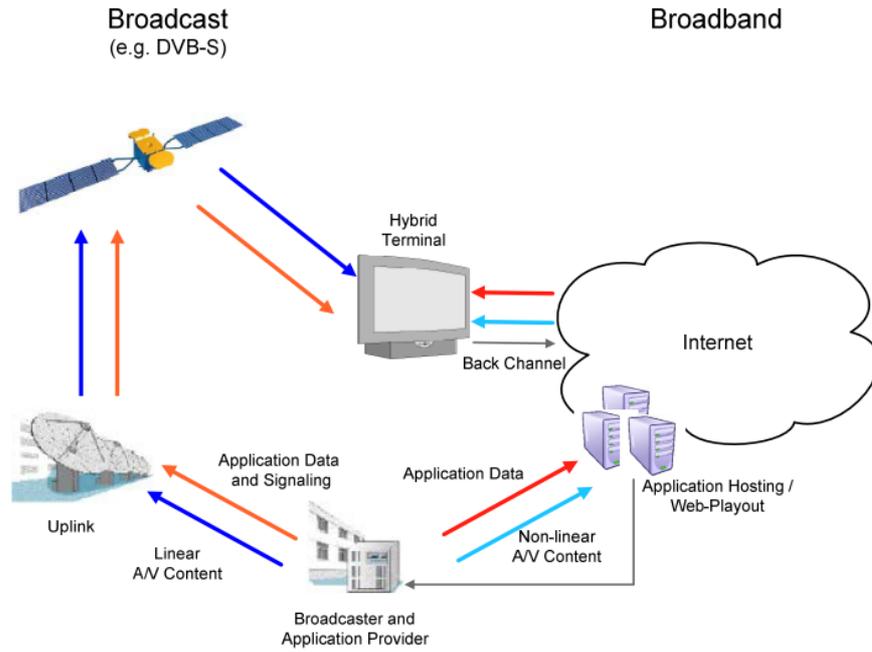
2.3 HbbTV

HbbTV (Hybrid Broadcast and Broadband TV) is specifically designed for the Hybrid TV by the HbbTV consortium and published by ETSI as ETSI TS 102 796 V1.1.1 in June 2010 [16]. In HbbTV the applications are categorized in two groups:

- Broadcast-independent applications like instant messaging and video on demand services, where the application is not dependent on broadcast service to function.
- Broadcast-related applications are broadcast dependent services where the services rely on broadcast content to run. These applications are in control of the broadcasters.

The hybrid TV can be connected to two networks in parallel i.e. to the Internet via a broadband interface and to the DVB network e.g. DVB-T, DVB-S or DVB-C. The broadband connection provides bi directional communication with the application provider. Through this connection the terminal can receive non-linear audio visual content. Through broadcast DVB network the hybrid TV can receive standard linear A/V content, application data and application signalling information.

The following figure is an example of a hybrid terminal connected both to DVB-S and broadcast networks.



Overview of a hybrid TV connected to DVB-S and Internet [16]

2.3.1 Hybrid terminal

A hybrid terminal is composed of the following functional components:

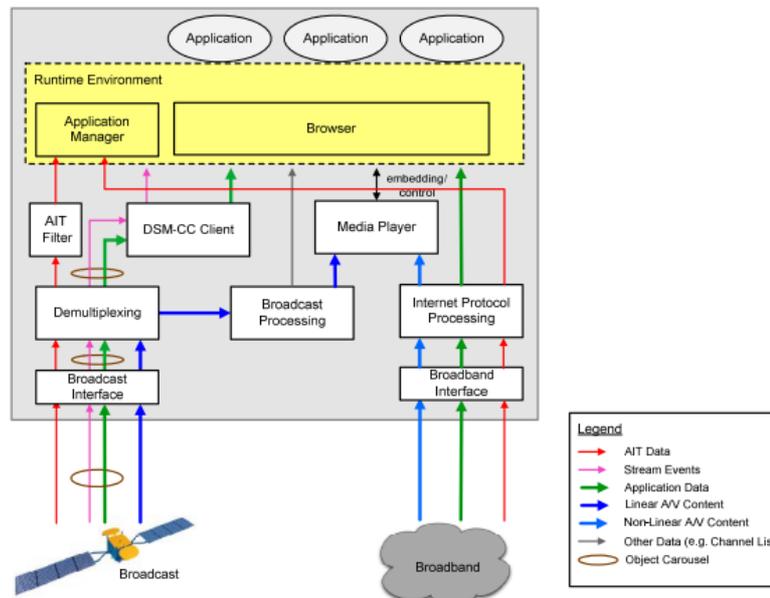


Figure 1: Structure of a hybrid terminal [16].

- Runtime Environment: Interactive application is presented and executed in the Runtime Environment inside the Browser.
- Application Manager evaluates the AIT to control the lifecycle for an interactive application. It is also responsible for controlling visibility of an application, access to private data of an application, starting and stopping of applications
- DSM-CC Client recovers the data from the object carousel and provides them to the Runtime Environment.
- Broadcast Processing Processes Linear A/V content in the same way as non-hybrid DVB terminal. Some information and functions such as channel list information and tuning functions are accessed by the runtime environment. These are included in the Other Data in the figure.
- Media Player is used for scaling and embedding linear and Non-linear A/V content in the user interface.
- Broadband Interface: The terminal connects to the Internet via this interface. This is the return channel of the Hybrid TV, which is used to request application data from the servers of an application provider. This connection is also used for Non-linear content like for content on demand applications.
- Internet Protocol Processing has the functionality to handle the data coming from the Internet and provide the application data to the Runtime Environment. This component forwards the Non-linear A/V content to the Media Player, which in turn can be controlled by the Runtime Environment, and therefore it can be imbedded into the user interface provided by an application.

The HbbTV technical specifications are developed by the joint effort of the following bodies.

- Joint Technical Committee (JTC)
- Broadcast of the European Broadcasting Union (EBU)
- Comité Européen de Normalisation ELECTrotechnique (CENELEC)
- European Telecommunications Standards Institute (ETSI).

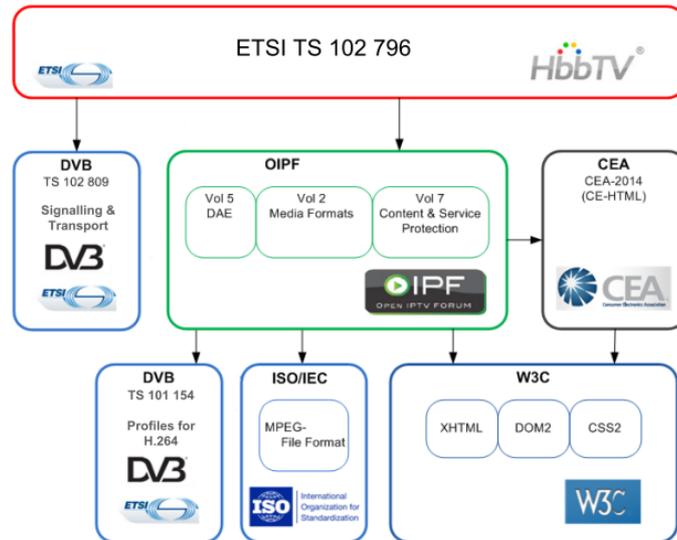


Figure 2: The Specification overview

HbbTV is not a totally a new standard i.e. it is not started from scratch but it adapts the necessary parts of already available standards. Following are some significant reference documents used in HbbTV.

- CEA-2014 - Web-based Protocol and Framework for Remote User Interface and the Internet (Web4CE), also known as CE-HTML.
- Open IPTV Application Environment of the Open IPTV Forum
- TS 102 809 (DVB) Signalling and carriage of interactive applications and services in Hybrid Broadcast Broadband environments

2.3.2 CEA-2014

Web-based Protocol and Framework for Remote User Interface on UPnP and the Internet (Web4CE) is published by Consumer Electronics Associations' R7 Home Network Committee. Currently there is no single standard that simultaneously insures home network device interoperability and a good user experience with the Internet services. CEA-2014 addresses these needs and is a framework for transmitting user interfaces over both the Internet and a home entertainment network. [17]

Web4CE (CEA-2014) has been accepted as the baseline remote user interface technology within the Digital Living Network Alliance (DLNA). Web4CE is the technology which can cross different domains like PC, mobile and TV in order to achieve: [18]

- Remote rendering of UI on other devices in the network

- For discovering client and servers in the network
- To exchange capability information between devices

The standard allows the UPnP Home network devices to present its interface as a web page, which can be displayed on any other device connected to the home network. The standard provides a browser based communication method for devices on UPnP network, which will allow users to control their CE devices using Ethernet and a special version of HTML called CE-HTML.

CE-HTML is one of the features of the Web4CE specially designed for television sets to allow user interaction take place using a remote control, keypad or a touch screen. Furthermore CE-HTML is optimized for rendering HTML/JavaScript web pages on CE devices, specifically on TV screens and scaling the contents for optimal use of the display area.

The collections of technologies used in CE-HTML are: [16]

- Definition of the application language (XHTML, CSS and JavaScript including AJAX).
- Definition of embedding non-linear A/V Content in an application.
- Definition of DOM event-handling (e.g. key events).
- Specification of still image formats.

Following is an example of a typical CE-HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"ce-html-1.0-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CE-HTML</title>
</head>
<body onload="document.getElementById('video_id').play(1);">
CE-HTML a/v object:<br/>
<object type="video/mp4" id="video_id" data="myvideo.mp4" width="640"
height="480"></object>
</body>
</html>
```

CE-HTML cannot convey interfaces to the “DVB world”, shortcomings like this are provided by the browser specifications of Open IPTV Forum, specifically the DAE part.

2.3.3 DAE

Open IPTV is a Global organization containing representatives from all key stakeholders in the IPTV area like network providers, CE/Home/Mobile device manufactures, Network Infrastructure providers, Content & Service providers and Technology providers.

The Open IPTV Forum has developed an end-to-end solution that allows consumer devices to access enriched and personalized IPTV services. The OITF block of this solution resides inside the residential network. OITF has the functionality to access IPTV services from both the managed and the unmanaged network. One of the functionalities is the Declarative Application Environment (DAE), which is a browser-based declarative language, based on CEA-2014 (CEA-2014-A). The DAE is for the presentation of user interfaces and also includes scripting support for the interaction with server-side applications and access to the APIs of the other OITF functions. It provides functionalities such as personalization, seamless experience across Devices and Red button functionality. Red button on the remote control is one of the means of accessing broadcast-related applications. [19]

CEA-2014 is profiled through the OIPF DAE specification and the Open IPTV Forum Release 1 Volume 5 - Declarative Application Environment.

2.3.4 DVB (TS 102 809)

TS 102 809 is a Technical specification of a framework for signalling and carriage of interactive applications and services in Hybrid Broadcast Broadband environments. This specification covers. [20]

- Starting and stopping of applications, which are bound to broadcast, service, bound to content on demand item or bound to a network operator.
- The Distribution of interactive applications files or services through broadcast or broadband networks
- Synchronisation of interactive applications or services to video or audio contents distributed through broadcast or broadband networks

- Referencing video, audio or subtitle content from interactive applications or services distributed through broadcast or broadband networks
- Signalling interactive application and services

Important components provided by TS 102 809 for Hybrid TV are the application signalling and application transport via broadcast or broadband using HTTP. The signalling is done via AIT of the relevant DVB service as defined in MHP standard. The AIT carries the signalling of all applications, which are supposed to run in the context of the broadcasted program. If an application signalled in the AIT is marked AUTOSTART then the application will be automatically launched when the corresponding broadcast service is selected. More about this will be described in “Broadcast signalling”.

2.3.5 Signalling interactive application and services

This component of the TS 102 809 is necessary for HbbTV because it helps the receiver to identify the applications associated with a service and find the location from which to retrieve them, the source of the broadcast data required by the broadcast-related application and how the signalling enables the broadcast to manage the lifecycle of applications. This is achieved by the following properties of applications.

Application metadata

Applications may have the following metadata associated with them.

Item	Description
Type	Identifies the platform needed to run or present the application.
Identifier	Identifies the application.
Control code	Defines the lifecycle state of the application.
Profile	Defines the minimum profile of terminal needed for this application
Visibility	Identifies whether the application is visible to the user or to other applications via an application listing API (where such an API is supported).
Priority	Defines the priority of the application relative to other signaled applications.
Icons	Identifies the location of icons for this application.
Graphics constraints	Identifies any constraints on this application with respect to changes in graphics configuration or presented video.
Storage information:	Defines whether an application should be stored, and which application files should be stored.

Figure 3: Application metadata.

Application type

Application type is associated with each application, which is used by the terminal to discard unsupported applications.

In MPEG-2 encoding the application types are 15 bit numbers. The receivers use the type identifier to filter out signalling for unsupported application types and discard them. Application types are, for historical reason, registered with MHP_APPLICATION_TYPE_ID.

Application Identification

The item identifier in Figure 2 above is for the application identification. Each application is associated with an identifier. The application identifier consists of two parts, the organization_id and the application_id.

- **organization_id:** This field is a globally unique value that identifies the responsible organization for the application. This field is also used for authentication when X.509 certification is used.
 - **application_id:** This field is allocated by the organization, which is registered with the organization_id field and is used to uniquely identify the application.
- [20]

2.3.6 Applications lifecycle

Starting and stopping application may be initiated directly by the end user by using dedicated buttons on the remote control or from a menu provided by the application or in response to signalling in a broadcast service.

Broadcast signalling

The broadcast signalling is a mechanism, which enables the broadcasters to control the lifecycle of a standard application. This is achieved through the application control code, which is used by the broadcasters to signal to the receiver what to do with the application with regards to its lifecycle. A typical scenario could be when the user selects a broadcast service:

- The terminal shall determine if the service has signalled any application related to it.

- If the application is broadcast-related and is signalled with a control code of AUTOSTART and is not still running from a previously presented service shall be started.
- If it is already running, the service shall continue to run uninterrupted and a second instance of the application shall not be started.
- If it is signalled with a control code of PRESENT and it is already running then it should continue to run but shall not be started if it is not already running
- If an application is added to the service with a control code of AUTOSTART the application shall be automatically started when the terminal detects its addition.
- Applications added to the service with any other control code shall not be automatically started.
- If a service related applications control code changes from another value to AUTOSTART, it shall automatically start.
- Applications whose control code changes to KILL or DESTROY shall be stopped
- An application with DISSABLED control code shall not be started and attempt to start it shall fail.

2.3.7 AIT

The application control code is signalled through the `application_control_code` field in the AIT (Application Information Table). The AIT provides full information about the broadcast related applications e.g. the `application_type`, `application_identifier`, `application_control_code` and `AIT_version_number` appendix 8.10b. The `version_number` is incremented by 1 whenever a change in the information carried within the `sub_table` occurs.

2.3.8 Application transport

In HBB environment the applications can be transported via Object carousel using broadcast network and via HTTP using broadband network.

2.3.8.1 Via Object carousel

In ETSI TS 102 809 V1.1.1 [21] transport of applications via DSM-CC object carousel is defined in section 7.1 and is based on the following specifications.

- ISO/IEC 13818-1 - MPEG 2 systems.

- ISO/IEC 13818-6 - DSM-CC.
- EN 301 192 - DVB specification for data broadcasting.
- TR 101 202 - Implementation Guidelines for Data broadcasting.

For the HBB devices, which are not connected to the Internet ETSI TS 102 809, specifies the carriage of HbbTV applications via the DVB broadcast channel. The DSM_CC object carousel is used to transmit applications, similar to MHP, for such devices. Stream events in a DSM-CC system consist of small data packets, which can be transmitted synchronously with the program signal.

The amount of data that can be distributed via broadcast channel is very limited as compared to broadband but could be sufficient for lightweight information application. These applications could be for example teletext services.

2.3.8.2 Via HTTP

HBB terminals allow download and execution of applications which could be a collection of documents for a self-contained or interactive service. The documents of an application are HTML, JavaScript, CSS, XML and multimedia files.

“When applications are downloaded using the HTTP protocol, the HTTP 1.1 protocol shall be supported as defined in RFC 2616.” [21]

2.3.9 Security

Section 11 of the HbbTV specification defines two levels, trusted and not-trusted, for applications. Broadcast related applications are trusted by default while broadcast-independent applications are not trusted. However the specification further specifies that the default behaviour may be modified as follows:

- Terminals may include a mechanism such that the end user is able to configure the broadcast-related application from a particular channel as non-trusted or the broadcast independent application as trusted.
- Applications from non-regulated channels, such as satellite channels, should not automatically be trusted. For example adult applications from Hybrid broadband broadcast satellite channels except following explicit end-user approval should not be trusted.
- Thawte and VeriSign TLS and SSL Root Certificates are supported.
- In HTTP over TLS, a client certificate is needed when it is required that Hybrid broadband broadcast application is delivered to the trusted terminal.

3. Personalization

Personalization tends to be defined according to how it is implemented for different services. The exact definition for personalization is hard to find. Kramer, Noranha and Vergo in the journal '*a user-centered design approach to a personalization*' write: "*Personalization is a toolbox of technologies and application features used in the design of an end-user experience. Features classified as "personalization" are wide-ranging, from simple display of the end-user's name on a Web page, to complex catalog navigation and product customization based on deep models of users' needs and behaviors.* [22]" Personalization is when something changes in the configuration or behavior of the application by the user information. Generally, the application gives a large set of possible choices to the user by learning a user's preferences or interest on a particular domain that has the highest value to them. Personalization can also be user-driven by providing explicit data in order to support the personalization process. This keeps the user in control of what might be important to them.

There are different scenarios of personalization that can be found in the web, such as [23]:

Link Personalization: Link personalization is used to recommend the items based on user buying history or user's ratings and feedback. The application selects the links that are more relevant to the specific user by changing the original navigation space of sites. It is widely used in www.amazon.com to link the site with shopping groups, new releases, recommendations etc.

Content Personalization: It involves in presenting pages with different information to different users based on user's implicit or explicit profile. Content personalization is divided into node structure customization and node content customization. In structure personalization, the application shows the sections and details in which the user is interested. The user can explicitly provide his preferences, e.g. in www.mycnn.com, the users can select a set of module from a large set including weather, news, music, etc. In node content personalization, different users receive different values for the same node attribute. E.g. in online shopping, price of item attribute may be personalized i.e. different users may receive different discounts based on their buying history.

Context personalization: The same information can be reached in different context. The same item may appear in different sets and different users may have different access restrictions depending on the roles.

Authorized Personalization: Different users may have different roles. Depending on roles, different users may have different access authorization for the same item. E.g. in exam registration application, the students may register a specific course before the deadline. But, only the administrator may register the course after the deadline.

Categories of personalization

Based on technology, market sectors personalization applications can be categorized into four groups [24].

Targeted marketing/advertising: Today, the consumers have a wide choice available for shopping the same product. But if the personalization technique could help the consumers to narrow down the choices, it would definitely increase the shopping experience and attract the consumers to the products that are mostly likely to meet their interests.

Customer relationships: The organization may have millions of customers. It may not be possible to treat each customer uniquely to keep better customer relationships. But through the automation of personalization process, the organization can increase customer loyalty. That also helps to reduce the costs in real business world and keep the customers to the organization.

Service integration: The users may have different profiles on different services. These profiles can be logically linked to a single profile identity. This single profile identity can provide authentication and single sign-on. This helps to integrate different services and to utilize data for personalization.

Knowledge management: Managing the users is a challenging task in the business environment. Through the personalization process, it is easier to group the users based on their interests and behaviors. This may be, to find out the common interests and preferences of the users that share in a particular task.

To deliver a personalized content, three key elements are necessary to link and process: *user profile*, *content* and *recommender system*. This is shown in the following diagram.

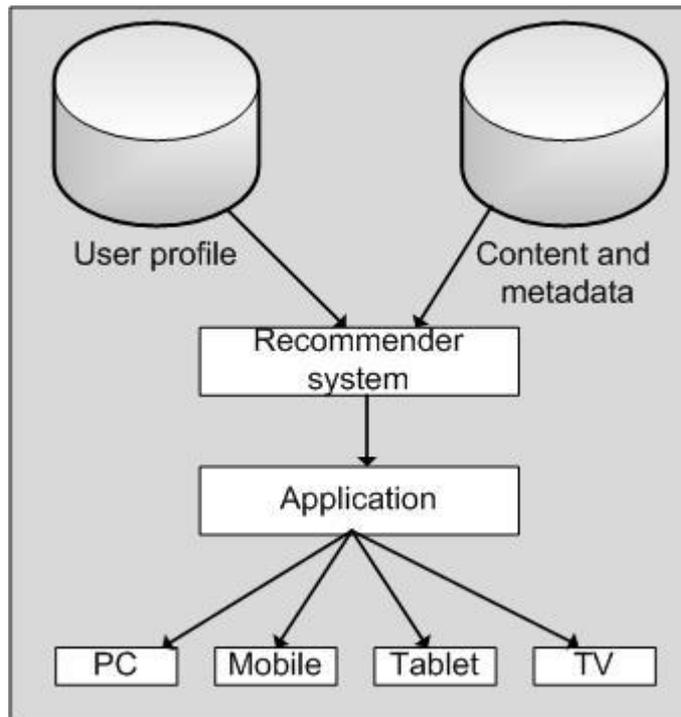


Figure 4: Key elements required for personalization

As shown in figure above, the user profile holds the information about a user's preferences and interests. The content includes metadata, which could be information about the content such as actors, genres, copyright, ownership, description etc. The recommender system determines business rules that define how user profile and content should be combined and the application describes how to present a personalized content on a specific device. These three key elements will be discussed below.

3.1 User Profile

User profile describes the user's data. The user data is said to constitute the user model and the user model profiles the user i.e. encodes user's preferences and needs [25]. Users may have different preferences and constrains such as demographic information, e.g., location, age, name, education level, etc. The users' preferences and constrains can be collected *explicitly*, through direct user intervention, e.g. by rating a special item that he likes and *implicitly*, through an agent that monitors users' activity, e.g. by monitoring browsing history. The collected user profile can be *static*, which does not change over time or *dynamic*, which changes over short term and long term interests.

3.1.1 User identification

To collect user data, unique user identification should be assigned. However, the accurate user identification may not be required for the system that constructs a profile representing a group of users. There are 5 basic approaches to user identification: login, software agent, proxy server, cookies and session id [26].

Login

Login gives accurate user identification and the user can use the same profile on different machines. But it requires user to go through registration process and login and logout each time they visit the site.

Software agent

It is a piece of small application installed on user's computer. It collects information about the user and sends to server through some protocol. It is reliable but it requires user-participation to install the application on his machine.

Proxy server

It acts as an intermediary between server and client to evaluate client's request to the server. It provides accurate user information, but the user requires registering his machine to the proxy server.

Cookies

It is a piece of data sent from a website and stored in a user's browser to remember the state of user's previous activity. It provides no burden on the user because the user does not need any extra action. But it may be less accurate in some cases. This is due to the fact that the user id is set to an individual device rather than individual user. If the user changes the device, the new user id is set to the cookie, providing inaccurate user profile for the same user. Another user may also use the same device, which provides the same user profile for the different user. The user himself, clearing and turning off the cookies on the computer may also kill the cookies.

Session id

New Session id is created each time user visits the site and lasts for the same session. It requires no burden on user, but it provides no permanent user profile but adaptation is possible during the session.

Depending on the problem domain, the best approach is to use multiple techniques for the user identification instead of using only one [26]. Cookies are widely used because it gives no burden on the user. For the accuracy and consistency, login-based system may be provided if the user chooses to login.

3.1.2 User information collection

When the user is identified, the next step is to collect the user's information about preferences and constrains. As we have mentioned in section 3.1, basically there are two ways to collect user information: explicit and implicit. We will discuss these two techniques briefly.

Explicit user information collection

The user participates to provide his personal information. The application may provide a form where user inputs his demographic information like, age, gender, marital status, hobbies etc. or check boxes or thumps up/down or likes/dislikes to express his preferences. It can also be 5 stars rating scale to review certain item. The problem with this method is that the user should have willingness to participate in the process that places an additional burden on the user. No user profile can be built if the users do not voluntarily provide his personal information. Sometimes the user may not choose to involve because of privacy issue or, the user may provide inaccurate data to his demographic information. Since the profile becomes static and the user's interest may change over time, the profile may represent inaccurate information. However, in some cases, the users enjoy sharing, providing feedback such as in a movie-rating site Netflix [27] and a social networking site Facebook [28].

Implicit user information collection

The user does not need any additional involvement during the process to construct a profile. The application may automatically monitor user's activity on the application to construct a profile. Some of the popular techniques used to collect implicit feedback are overviewed in the table following [26].

Collection technique	Information collected	Pros	Cons
Browser cache	Browsing history	No need to install extra application.	User may see old cached information.
Proxy servers	Browsing activity	Speeds up the browsing	Users must use proxy server.
Browser agents	Browsing activity	Agent collects all web activity.	Need to install new application.
Desktop Agents	All user activity	All user files and activity available	Users need to install new application.
Web logs	Browsing activity	Information about multiple users collected.	May be little information since only from server side.
Search logs	Search activity	Collection and use of information all at same site.	Cookies must be turned on and/or login to site.

Table 1: Implicit user information collection techniques

Browser cache, proxy servers, browser agents and desktop agents are client side approaches to collect implicit information about the user. These techniques capture the information about the users what they browse or perform other activities on the application. The activities can be a browsing history that contains the URLs visited by the user with dates and times. The problem with these approaches is that it places some burden on the user such as, to install a new application. However, it collects a large amount of user data.

The web logs and search logs are server-side approaches to collect implicit information about the user. These techniques collect the user information silently through session IDs, logins or cookies. The web logs collect the browsing history of a particular user to a particular website. The search logs collect the information about the search queries submitted by a particular user with dates and times. The advantage of these approaches is it does not place any burden on the user. However, it collects a little information about the user. When a user clicks some links or views an item, it assumes that the user has some interest in the item.

3.2 Content and metadata

Content and metadata is another key element to combine with user profile in order to deliver personalized content. Basically, the content includes metadata that is information about the content such as name, title, description, ownership, formatting date, copyright, genre etc. MPEG7 [29] and TV anytime [30] are well-accepted standards for description of audio video contents. They both have capabilities to specify structured metadata descriptions of both the AV content and the user preferences regarding AV content. In this section, we will look briefly into these two standards.

3.2.1 MPEG-7

MPEG-7 is an ISO/IEC standard developed by Moving Picture Experts Group. It is officially named as 'Multimedia Content Description Interface'. It became standard in 2001 and it was formally called 'multimedia content description'. MPEG -7 provides a set of standardized tools to describe multimedia content by adding metadata (data about data). The amount of audio-visual content is growing day by day. It became more and more difficult to find and manage an enormous amount of content. So, MPEG -7 was developed in order to make the content easier to identify and manage from both professionals and users perspective. The main elements of MPEG -7 are [29]:

Description Tools

The descriptor (D) defines the syntax and the semantics of the metadata element and Description Schemes (DS) specifies the structure and semantics of the relationship between the components that may be both descriptors and description schemas.

Description Definition Language (DDL)

It defines the syntax of the MPEG-7 description tools. It is based on XML. It allows creating new description schemas and descriptors. It also allows the extension and modification of existing description schemas.

System tools

It supports binary coded representation for efficient storage and transmission, transmission mechanisms, multiplexing of descriptions, synchronization of

descriptions with content, management and protection of intellectual property in MPEG -7 descriptions.

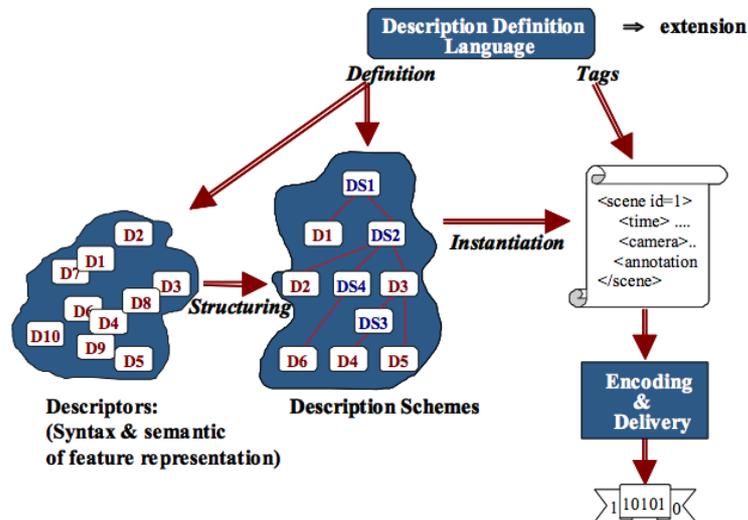


Figure 5: MPEG -7 main elements [29]

The figure above shows the relationship among main elements of MPEG -7. The DDL allows the definition of the MPEG -7 description tools, both descriptors and description schemes. The DDL also allows the extension of any application of DSs. Binary format of descriptions is obtained by means of the BiM(Binary format for MPEG-7) which is defined in the system part of MPEG-7 specification.

MPEG-7 provides a broad range of applications that makes the web searchable for multimedia content. This would also apply for a large content archives which would easily be accessible. All multimedia applications will benefit from MPEG -7. Some of the applications examples are listed below:

- Architecture, real estate, and interior design (e.g., searching for ideas).
- Broadcast media selection (e.g., radio channel, TV channel).
- Cultural services (history museums, art galleries, etc.).
- Digital libraries (e.g., image catalogue, musical dictionary, bio-medical imaging catalogues, film, video and radio archives).
- E-Commerce (e.g., personalized advertising, on-line catalogues, directories of e-shops).
- Education (e.g., repositories of multimedia courses, multimedia search for support material).

- Home Entertainment (e.g., systems for the management of personal multimedia collections, including manipulation of content, e.g. home video editing, searching a game, karaoke).
- Multimedia directory services (e.g. yellow pages, Tourist information, Geographical information systems).
- Multimedia editing (e.g., personalized electronic news service, media authoring).
- Shopping (e.g., searching for clothes that you like).
- Social (e.g. dating services).

MPEG-7 provides interoperability among systems and applications those are mentioned above. The use of mpeg-7 descriptions results a flexible and scalable framework for designing services that can be accessed from different terminals like Mobiles, set-top boxes and computers as shown in figure below.

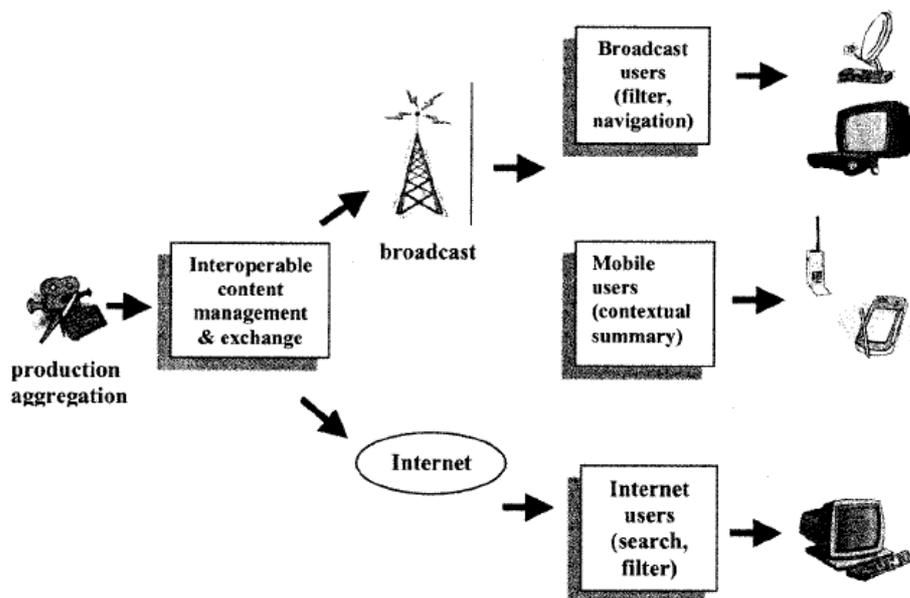


Figure 6: Role of mpeg-7 in facilitating interoperable services and applications [31]

MPEG-7 describes new ways to produce, offer, filter, search and manage digitized multimedia information. It allows finding the multimedia content easily that the user needs. Currently, MPEG-7 standard (ISO/IEC 15938) consists of 13 different Parts [32]. Each part covers a certain aspect of the whole specification as presented in table below:

Part No.	Standard number	Title
1.	ISO/IEC 15938-1	Systems
2.	ISO/IEC 15938-2	Description Definition Language
3.	ISO/IEC 15938-3	Visual
4.	ISO/IEC 15938-4	Audio
5.	ISO/IEC 15938-5	Multimedia Description Schemes
6.	ISO/IEC 15938-6	Reference Software
7.	ISO/IEC 15938-7	Conformance
8.	ISO/IEC 15938-8	Extraction and use of MPEG-7 Descriptions
9.	ISO/IEC 15938-9	Profiles
10.	ISO/IEC 15938-10	Schema definition
11.	ISO/IEC 15938-11	Profile Schemas
12.	ISO/IEC 15938-12	Query Format
13.	ISO/IEC 15938-13	Compact Descriptors for Visual search

Table 2: MPEG-7 parts

MPEG-7, part 5: Multimedia Description Scheme includes two tools that have been specially implemented for *user interaction tools*, namely the User Preferences and the User History [33].

User preferences

It describes preferences for different types of content and modes of browsing, including context dependency in terms of time and place [29]. It also describes the weight of the relative importance of different preferences, the privacy characteristics of the preferences and whether the preferences are subject to update.

User history

It logs the history of activity carried out by a user of a multimedia system. The history can be exchanged between consumers, their agent, content providers and devices. This helps to determine the user preferences regarding AV content.

MPEG-7 descriptions schemas are mainly used in audiovisual digital environment or web applications to describe AV content and user preferences. MPEG-7 DDL is also used by TV-Anytime for its metadata representation. TV-Anytime specifications are used for broadcasting environment and interactive TV services. TV-Anytime will be discussed in the following section.

3.2.2 TV-Anytime

The increase in the number of channels and programs due to digitalization of television has increased the difficulty level of browsing the program guide. As described in background and motivation section there is a need for technologies which can recommend TV programs that best match user preferences. In order to access content more efficiently structured and standardized description of multimedia content is necessary. TV-Anytime forum is an association of organizations that provides an XML based framework for the structure and standardized description of multimedia content.

TV-Anytime forum was formed in California USA on September 1999. It has developed open specifications that allow Consumer Electronics Manufacturers, Content Creators, Telcos, Broadcasters and Service Providers to exploit personal media storage.

TV-Anytime system architecture

In a broadcast situation TV-Anytime system can contain three major elements [34]:

- Service provider that delivers TV-Anytime service
- Transport provider that carries the delivered service
- A piece of equipment at consumer's residence that stores and plays the content

A simple broadcast model without digital rights protection is illustrated in the following figure.

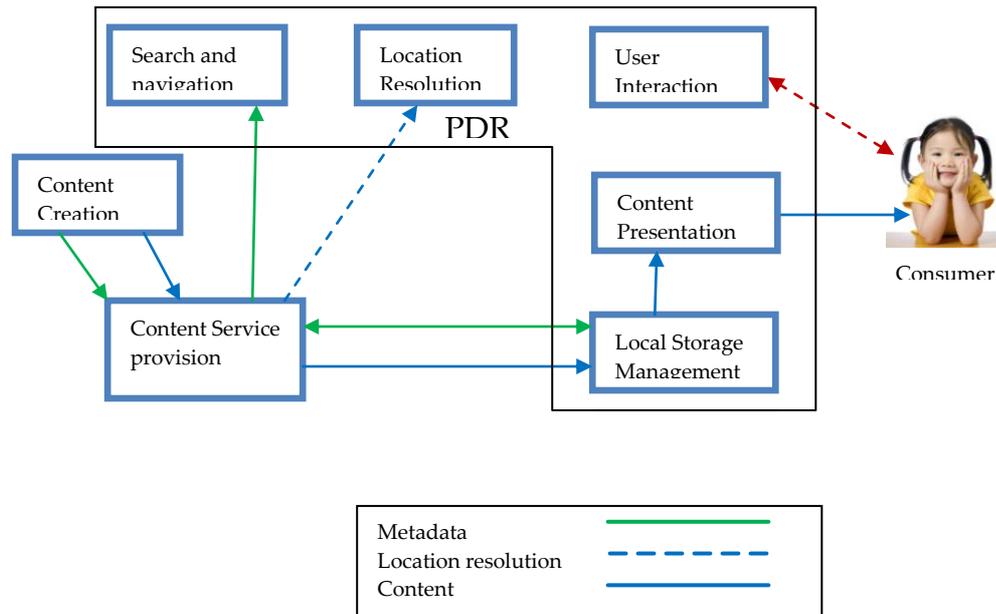


Figure 7: Broadcast model without rights management protection [34]

Each box in the above figure represents a function of TV-Anytime system. Different service providers can implement them in different ways. The features supported by this broadcast model are listed in TS 102 822-1 [34]. As it can be seen in the figure above three functions, content creation, content service provision and access are external to PDR (Personal Digital Recorder). Usage history data and user preference data can be collected by the service provider, which is shown as a bi-directional green link between the PDR and Content service provision. Content could be created by a movie studio or entertainment company. The metadata addition, repackaging and broadcast can be done by a broadcaster. The PDR, which is a device at consumer's premises that allows him to store and view the content, can contain the remaining functions.

This system will allow the user to search, select, locate and acquire content that he likes. The search and selection e.g. by an EPG depend on metadata that advertises the availability of content. The metadata can be added by one or more parties such as broadcaster, the content creator or a third party. The third party could be for example the rights management and protection.

Following is the list of enabled feature for broadcast model described in technical specification [34] and set by TS 102 822-4 and TS 102 822-3-1.

- Use of EPG to find and capture broadcast content
- Search and selection of on-demand content with associated pricing information

- Capture and playback of audio, video and data (AVD)
- Cross linking of A/V content to related content
- Support of consumer preferences
- Content can be updated/replaced by newer in-coming versions
- Support for a variety of broadcast content types
- Support for all broadcast delivery mechanisms
- Multi-user preference support
- Model 2 - Consumer Response Model
- Updated listing/capture data can be delivered to "broadcast" analogue personal recorders (via return path or other mechanism)
- Updated listings/capture data can be delivered to "broadcast" PDRs
- Verification of usage of content on PDR
- Ability to collect usage data

3.2.2.1 *TV-Anytime Functionality*

TV-Anytime functionality can briefly be described as follows [35]:

- Enables applications to exploit personal media storage in consumer electronics Platforms
- Network independent for content delivery
- Interoperability
- Personalization
- Automatic recording

3.2.2.2 *TV-Anytime metadata model*

TV-anytime uses metadata, which is the descriptive data about contents of a program, such as title and genre. Metadata can also include information about user preferences, such as favorite actors or TV-show and history. As described in TV-Anytime specification, they have adopted XML as the representation format for metadata. The main reasons for using XML is its extensibility, wide usage and the separation of data from the program.

TV-Anytime uses MPEG-7 Description Definition Language (DDL) to define the structure and type of metadata as well as the XML encoding. [35]

TV-Anytime metadata includes

- Program information
- Segmentation information

- Related material information
- Purchase information
- User profiles
- Usage history

Within TV-Anytime environment the consumer or agent will use descriptors or hyperlinks to access a particular piece of content.

TV-anytime metadata assists the consumer to find, navigate and select and manage the desired content. As TV-anytime is network independent, therefore the content source could be internal or external for example at broadcaster, interactive TV, internet or local storage.

The process of creation and delivery of metadata to the consumer may involve many organizations. A standardized metadata is necessary because it ensures a high level of interoperability. It allows authoring once for many destinations and editing of metadata that originate from different sources. Therefore TV-Anytime has adopted MPEG-7 XML based Description Definition Language as its representation format for metadata.

3.2.2.3 TV-Anytime Metadata Process Model

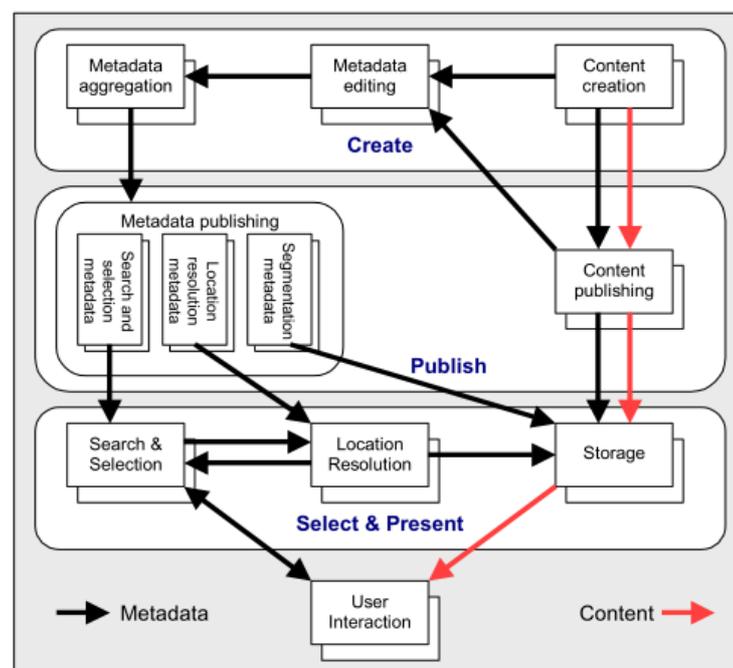


Figure 8:Metadata and content flow

The processes in the metadata process model are defined by the TV-Anytime Metadata Specification, S-3 v1.3, Part A as follows. [35]

Content creation

The production of content or a program is represented by content creation. This role can be fulfilled for example by a movie studio or an entertainment company. Information metadata and the content of a program may also be captured during the production process, but the metadata may not be in the form that can directly be used at this stage.

Content publishing

After the content creation process the data is now available for the publisher e.g. a broadcaster. At this stage a broadcaster will do the repackaging, addition of metadata and broadcasting. Metadata editing might still be required at this stage. Instantiation of programs e.g. where the program can be found, is done at this level. In case of a broadcast this could be EPG or program schedule.

Metadata editing

This process edits the raw information available from the content creation and publishing process to make it suitable for representing to the end consumer. The output of this process is edited metadata.

Metadata aggregation

Because of the existence of different types of TV-Anytime systems it might be necessary to aggregate the metadata from different content creators and publishers. This may result in original data being changed.

Metadata publishing

Content selection and location resolution processes depend on metadata to locate and select the programs sought by the user-agent. The content selection process is concerned with the metadata that describes programs but may also involve use of the program resolution metadata. The location resolution service needs information about the location of programs.

Content selection

Content selection can be done by the consumer itself or by a software agent. For the software agent to function correctly it is necessary to provide user preferences and user profile to the content selection process. The preferences could be implicit data e.g. from user's history or explicit data specified by the user.

Location resolution

Location resolution is the discovery of a program i.e. where and when a program can be found.

3.2.2.4 Content referencing in TV-anytime

TV-Anytime uses a CRID (Content Referencing Identifier) to refer to content. CRID is assigned by an authority which can resolve the CRID to a location. A CRID points to a piece of content or other CRIDs. The purpose of CRID is to allow the user/user-agent to acquire a specific instance of a specific item of content. [34]

On tv-anytime.org the TV-Anytime forum has announced that they have received confirmation that TV-Anytime CRID will become an RFC. In the introduction of RFC 4078 at <http://tools.ietf.org/html/rfc4078> CRID is defined as:

```
The 'CRID:' Uniform Resource Locator is designed to be the bridge  
Between program-related descriptive metadata and corresponding  
program location data that may be published over a different  
distribution network or at a different time. [36]
```

CRID:// <authority> / <data>

Where authority is the DNS name and data is the URL that points to the data. CRID could be the value of a Group Id or a Program Id. If it is a Group Id then it may point to one or more Program Information. Program information could further point to other CRIDs. Eventually CRID must point to one or more locations. The following figure is an example of a Star Trek program where the CRID refers to series of Star Trek which refers to episodes and finally resolves to a location.

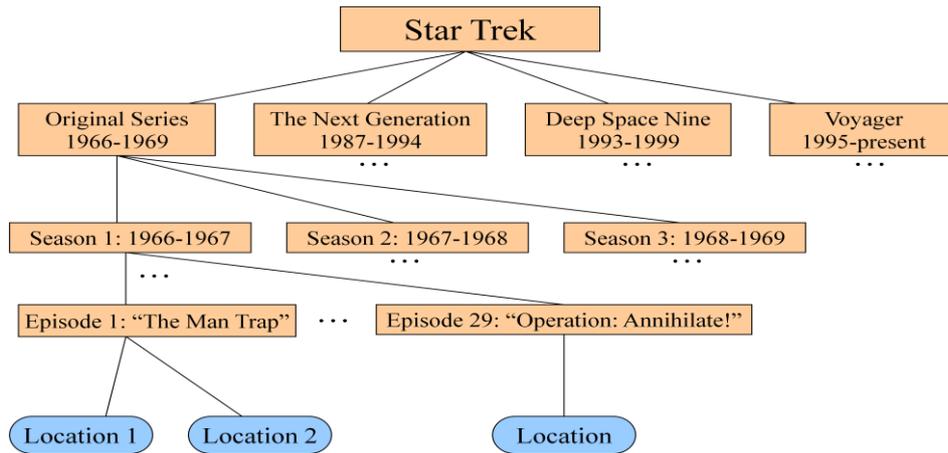


Figure 9: CRID Location resolution [37]

The CRID also acts as the link that connects different content-related metadata descriptions [35]. This is illustrated with the following diagram.

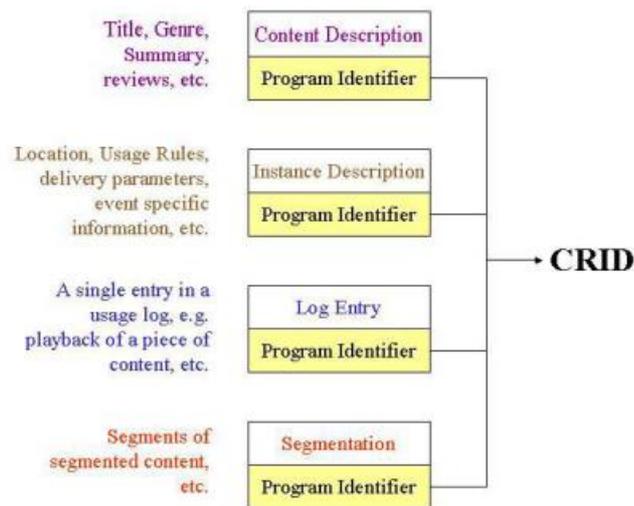


Figure 10 : Metadata that references a programme CRID [35]

TV-Anytime content referencing makes it possible to, for example, even plan to record a program whose date, time and the channel it will be broadcasted from is not known yet. It is possible due to its ability to refer to content independent of its location. In this case the PDR will be provided with a reference to this program. When more information, for example to link the reference to the program and time and date of the program is available, it will be provided to the PDR in due time [34].

3.3 Recommender Systems

A century ago it was easy to choose among different items because there was not a huge diversity and choices were limited. A book seller in a town knew almost all of his customers by name and what kind of books they like. He could easily recommend one of the new books to them. At the only restaurant of the town the waitress asks Mr. Jones if he wants "the regular" because she knows that Mr. Jones always orders the regular and Mr. Jones knows what the regular meal is. Henry Ford once said, "Any customer can have a car painted any color that he wants as long as it is black", so limited was the color choice [38].

Now the choices are in millions, iTunes has more than 10 million tracks, Amazon has millions of titles of books to choose from and at Netflix and Viaplay you have to choose from thousands of movies to watch one. Choice is good but to choose among millions of items even is difficult for an expert. To find the relevant stuff from huge archives is difficult and time consuming. We definitely need a recommender, but in 21st century with millions of choices, a friend, a librarian or the guy at blockbuster cannot help us. This is why we need an intelligent computer assisted recommender system.

Recommender systems are software techniques that make predictions about users and recommend items to the users. The recommendations are based on user preferences and information filtering techniques. Amazon, Netflix, Apples iTunes and other companies with huge archives use recommender systems to help users choose from their archives. The recommender systems assist the users and the service providers in creating a personalized view of the service. In "methods and techniques of adaptive hypermedia" the personalized view is defined as one of the six types of adoptive hypermedia systems [39]. Because of the sheer amount of information it is difficult for many users to access easily a subset of information. A recommender system, which is the central unit of personalized view, helps the users to access information easily [39].

Recommender systems have different roles; on the one hand they help users by recommending items based on user's preferences and/or browsing history while on the other hand they help companies to increase their turnover by selling more stuff [40].

From service providers perspective a Recommender system is a valuable tool if it:

Attracts more users: This depends on the accuracy of the recommender system and the trust of the user in the system. The trust of the user is hugely

dependent on the accuracy of the RS; the more accurate the RS, the more the user's trust grows in the system. Privacy is another important factor which can directly affect the users trust in the system. Some users may be more sensitive about the data collected by the RS e.g. users buying habits, demographic data etc. which are collected and stored by the RS for future recommendations.

Increases the turnover by selling more items: If the RS is used for a commercial service then this might be regarded as the most important function of the RS.

Help the service provider to sell more diverse items: Because of the diversity and sheer number of items it is hard to go through all the items, read the description and choose one. Therefore, for example from an online movie store, normally the users buy or rent popular movies. But, the service provider is definitely interested in renting all the movies and not just the popular ones.

From users perspective a Recommender system is a useful tool if the following points are met.

Fulfill the end user's tasks: According to [41], there are eleven popular user tasks for recommender systems to fulfill. These tasks are derived from the research literature and from the deployed recommender systems [41]. In this thesis only the tasks relevant to a personalized EPG will be discussed.

Find good items: Besides recommending some good items as a ranked list to the users, it is important to predict how much the user would like the item being recommended. This prediction could be a scale, for example from one-to-five stars [40] [41]. Many commercial service providers, for instance Netflix, Amazon etc., has this prediction scale as one-to-five stars in their recommendations.

Find all good items: For Television programs/movies this could be applicable only if the number of relevant items is relatively small or when it is important to show all the good items. These items could be shown as a ranked list, where the items matching closely to the user's preferences, likes and history will start at the top of the list in a descending order.

Annotation in context: If for some reasons, even in a personalized part, it's important to list all the items e.g. an EPG, then the items matching the user preferences should be emphasized. Annotation in context is not a ranked list

as in “find all good items”, therefore the items matching the users preferences, likes and history should be emphasized.

Recommend a sequence: In the context of EPG, recommending a sequence will be a TV series, which might be more interesting to the user than recommending a single episode.

Just browsing: The RS should also have recommendations for the new unregistered users. This can be achieved by interpreting their browsing and usage of hyperlinks as interest. Items could be recommended based on this “interest”. The ability of RS to help the just browsing user can convince them to register and become loyal customers.

Content based and collaborative filtering are some of the approaches used for information filtering in recommender systems. In the following content based and collaborative filtering techniques and the algorithms used in these approaches will be discussed.

3.3.1 Content Based Filtering

In this approach the recommendation is based on the match between user profile/preferences and the content/description of the items. The system recommends items that are similar to the ones that the user liked in the past. For example the RS can recommend an action film to the user because the user has positively rated a film belonging to this genre. Content Based Filtering could be used both for structured and unstructured data. Structured data could be for example in a database where each item is represented by a set of attributes and values associated to those attributes. In this case it is relatively easy to find the match as compared to unstructured data.

The unstructured data could be an article in a web page or an unrestricted description of an item in a database. In CBF the words in user’s preferences/wish-list can be treated as attributes with a Boolean value of 0 or 1 depending on its existence in the description of the item and an integer value representing the number of times the word appears in the description. To achieve this, a numerical statistic method called *tfidf* (term frequency inverse document frequency) is used which reflects the importance of a word in the description/document.

3.3.1.1 TFIDF

Term frequency inverse document frequency is a weighting factor in data mining where the *tfidf* value is directly proportional to the number of times a word appears in the document. Traditionally the word document is used because the motivation was to retrieve documents [42], but in recommender systems basically the description of the items are used. Term frequency is the number of times a term occurs in a document/description. However some terms like conjunctions and prepositions are very common which will weight documents incorrectly. These terms are ignored. Some other terms which are common in all documents of the same subject for example the term car will most likely be common in all the automobile documents but these words will get less weight and thus less relevance because in *tfidf* a term gets high weight if it appears frequently in the same document and not in corpus i.e. all documents.

tfidf is the product of term frequency and inverse of document frequency i.e. $tf * idf$. The term frequency $tf(t,d)$ simply means the number/frequency of occurrence of the term t in document d .

Mathematically it is expressed as:

$$tf(t, d) = \frac{f(t, d)}{\max \{f(w, d) : w \in d\}}$$

Where

t is the term that appears in document d

w is the number of words in document d

The inverse document frequency measures the commonness or rareness of a term in all documents (corpus). To achieve this: corpus N i.e. the total number of documents is divided by the number of documents which contain the term t [43].

$$idf(t, N) = \log \frac{N}{df_t}$$

Where:

N is the corpus i.e. all the documents

t is the term that appears in N

df_t is the number of documents that contains the term t

To calculate a weight for each term in each document the *tf-idf* is calculated as:

$$tfidf_{t,d} = tf_{t,d} \times idf_t$$

Latent Semantic Analysis is one of the CBF approaches that use tf-idf.

3.3.1.2 LSA

Latent Semantic Analysis (LSA) is used to analyze documents to find the concepts behind those documents. The need for LSA arises from the fact that meaning of the words change in different contexts. The word bank could mean bank of the river or financial institution when mentioned together with money [44]. The synonyms, multiple meaning of the word and the context of the word can create ambiguities. When a search for documents is based on word comparison then the result will not very often be what the user had in mind. What we actually want to compare is the meanings or concept of the words and not the word itself. So only the meaning of the word is not important, it is the context which makes the concept clear and removes the ambiguity. For example if the word java and computer are in a document then the document is very likely about a programming language. However if the word java is mentioned together with Sumatra and Bali then the document is probably about Indonesia.

One of the properties of LSA is to filter the noise from the word-concept relationship and to find the smallest set of concepts that spawn over the documents. This noise is introduced to the relationship when the authors use the different words for the same concept.

3.3.2 Collaborative Filtering

In collaborative Filtering the recommendation is made to the active user based on his previous likings and as well based on the items that other users, with the similar taste, liked in the past. In CF the user-item space is represented by a ratings matrix where the users are represented by a set of m users $U = \{u_1, u_2, u_3, \dots, u_m\}$ and the items are represented by a set of n items $I = \{i_1, i_2, i_3, \dots, i_n\}$. Any user from the set m has a list of items from the set I about which the user u_i has expressed his opinion i.e. u_i has a list of l_{ui} items. If we call this matrix E , then each entry $e_{i,j}$ represents the i th user's opinion for the j th item. Opinions are the explicit ratings, a numerical value usually from 1 to 5, given by the user to the items. The opinion of the users could also be gathered implicitly by analyzing users browsing history and purchase records. It

should be noted that the recommended items should not be on the list of purchased items by the active user u_a i.e. $l_r \cap l_{ua} = \emptyset$ [45].

In the rating matrix E each rating is a numerical value which is predefined by a scale e.g. 1 to 5. The rating could also be a 0, meaning that the user has not yet rated the item. It also applies to the newly added item. The matrix will become sparse if many items are not rated where the probability of finding a set of users with similar tastes is difficult.

Collaborative Filtering are divided by the researchers in two main types namely user based CF and item based CF [45]. Different algorithms used in each type will be discussed here.

3.3.3 User-based CF algorithms

In User-based CF algorithms statistical techniques are used to find neighbors. Neighbors are set of users who have agreed with the active user in the past i.e. they have rated different items similarly or they have purchased similar items. Algorithms used in this category are described in this section.

3.3.3.1 Similarity computation

There are used several algorithms to find users with similar taste, in other words to find users who enjoy the same type of e.g. movies that the active user does. Some of those algorithms ranging from the simplest to somewhat complex ones will be discussed here.

Let us assume that we have two movies rated by three users and we want to recommend one of these movies to active user u_a who rated the *Gone with the wind* with 3 and the *Godfather* 1. Following table shows the ratings of the users for each movie.

Users	Gone with the wind	Godfather
Allan	4	5
Louise	5	5
Hans	2	4

Table 3: User ratings

3.3.3.2 Manhattan distance

Manhattan distance is the easiest method to measure the distance to find the person who is most similar or close to the active user u_a .

$$d = \sum_{i=1}^n |x_i - y_i|$$

In a two dimensional space users (Allan, Louise and Hans) could be represented by (x, y) points. To calculate distance between the active user u_a and the users in table 1 we will represent Allan with $(x_1 - y_1)$ and user u_a with $(x_2 - y_2)$. Applying Manhattan distance to them we get:

$$|d| = |x_2 - x_1| + |y_2 - y_1|$$

Using this formula the Manhattan distance for user u_a and Allan is 5. Calculating Manhattan distance between remaining two users and user u_a we get 4 for Louise and 2 for Hans. From this calculation we can see that Hans is the closest match. Looking at Hans ratings in table 1, we can see that he gave Godfather 4 stars therefore Godfather will be recommended to user u_a .

3.3.3.3 Euclidean Distance

The Euclidean Distance is another formula to measure distance between two points, p and q . The Euclidean Distance is given by Pythagorean formula:

$$a^2 + b^2 = c^2$$

Using the above formula the distance in Euclidean n -space between point p and q can be calculated by:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

The closeness/similarity between the active user u_a and another user in a 2d space can be calculated as:

$$d(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

In the above example there were only two items to be rated but in real life applications users rate many more items. If we want to find closest neighbors i.e. similar users for user u_a and compare u_a with, say Allan who has rated only 2 items and Louise how has ratted five items then it might give false results. One of the solutions is to get average of each user's distance i.e. the result should be divided by the number of items rated. Besides this another problem is the scale system for rating items e.g. from 1 to 5. The users' ratings habits are very diverse; some tend to give 5 stars to all the items they like and 1 to the items they don't like, others seldom rate 5 or 1 they tend to give 2, 3 or 4 stars. It becomes difficult to interpret these ratings. Does 3 means the user does not like it or it's close to like it, should we interpret one user's 4 as another user's 5? One way to deal with this problem is to drop the scale system and adopt only thumbs up and thumbs down as Pandora does. Another solution is to use Pearson Correlation Coefficient.

3.3.3.4 Pearson Correlation Coefficient

Correlation is the measure of relationship between two variables. The Pearson Correlation Coefficient measures the degree of correlation between two variables. The degree of correlation is 1 in case of perfect agreement/positive and -1 in case of perfect disagreement/negative. The possible values for Pearson Correlation are between -1 and 1 inclusive. Perfect positive, perfect negative and 0 which is no correlation are rare [46] [47].

There are different forms of the formula for Persons Correlation Coefficient but it's better to implement the following variant described in [38]. It's faster because it only needs one pass.

$$r = \frac{\sum_{i=0}^n x_i y_i - \frac{\sum_{i=0}^n x_i \sum_{i=0}^n y_i}{n}}{\sqrt{\sum_{i=0}^n x_i^2 - \frac{(\sum_{i=0}^n x_i)^2}{n}} \sqrt{\sum_{i=0}^n y_i^2 - \frac{(\sum_{i=0}^n y_i)^2}{n}}}$$

3.3.4 Item-based CF algorithms

In item-based CF algorithms probabilistic approach is used to compute the prediction of a users rating for a new item. As opposed to user-based filtering, where the most similar users were found, in Item-based filtering the most similar items are found. In this technique first a model of the user ratings is created and then the prediction is made. One of the advantages of this technique is that the model can be developed in advance. The model is created by finding the most similar items and combines that with the users' ratings. Suppose an Item-based algorithm finds that *The Untouchable* is similar to *Scarface*, then if a user rates *The Untouchable* high, the system will recommend *Scarface* to the user. Among other algorithms correlation-based similarity, which is described in previous section, is also used in Item-based content filtering. Three other algorithms, cosign similarity, Adjusted cosign similarity and Slope one which are used in Item-based CF are discussed in the following.

3.3.4.1 Cosign-based Similarity

In cosign similarity items are treated as vectors in m dimensional user space. The similarity between two items is measured by cosign of the angle between these two vectors/items. Cosign similarity is used in text mining and collaborative filtering as well. Given an $m \times n$ rating matrix the similarity between items "a" and "b" can be denoted as:

$$\text{similarity}(a, b) = \cos(\vec{a} \cdot \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_{i=0}^n a_i b_i}{\sqrt{\sum_{i=0}^n (a_i)^2} \sqrt{\sum_{i=0}^n (b_i)^2}}$$

The similarity ranges between -1 and 1 inclusive. Where -1 is perfect dissimilarity and 1 is perfect similarity. Perfect similarity and dissimilarity is rare. The values are normally between -1 and 1.

One important drawback of the cosign-based similarity is that the grade inflation is not taken into account. The grade inflation occurs, as described before, when some users rate very high and others tend to rate very low. This problem is solved in adjusted cosign similarity algorithm.

3.3.4.2 Adjusted cosign similarity

In adjusted cosign similarity the corresponding user average is subtracted from each co-rated pair. The similarity between items i and j , in the seminal article "Item-based collaborative filtering recommendation algorithms" [45], is given by the following formula.

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

Where: \bar{R}_u is the average of u -th user's rating.

3.3.4.3 Slope one

Slope one is also widely used in item-based collaborative filtering. The popularity of slope one is in its simplicity and ease of implementation. Slope one predicts how much a user will rate an item by using the average difference of two items from the ratings of users who rated both items. There are different slope one algorithms but the weighted slope one is more accurate.

3.3.4.4 Weighted slope one algorithm

The algorithm can be implemented in two steps. First the average deviation of an item i with respect to item j can be computed. Then the computed deviation could be used in second step to make the prediction.

Step I

$$dev_{j,i} = \sum_{u \in S_{j,i}(X)} \frac{u_j - u_i}{card(S_{j,i}(X))}$$

Where:

$card(S)$ is the number of elements in S

X is the set of all ratings

$card(S_{j,i}(X))$ is the number of people who have rated both j and i [38]

Step II

$$pwSl_{(u)j} = \frac{\sum_{i \in S(u) - \{j\}} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u) - \{j\}} c_{j,i}}$$

Where: $c_{j,i} = card(S_{j,i}(X))$

3.4 Summary of the personalization chapter

The discussion in the Personalization chapter can be summarized as follows. This summary answers the question raised in problem domain for the need of formulating the requirements for a personalized interactive EPG for Hybrid Broadcast Broadband Television.

3.4.1 Collecting user information

In user information collection the drawbacks and advantages of both explicit and implicit methods were discussed in detail. In the context of HbbTV their pros and cons can be summarized as follows.

3.4.1.1 *Explicit*

Pros

- A companion device e.g. smart phone, tablet or pc can be used
- Extra software is not needed to be installed by the user
- A web service could be used for getting user preferences
- User is in control

Cons

- User may not provide his preferences
- User may not rate a program
- User may provide false information/ratings
- User may not update his preferences if his interest changes

3.4.1.2 *Implicit*

Pros

- Does not put burden on users
- Can be collected seamlessly
- User intervention is not required

Cons

- User actions can be wrongly interpreted; clicking on a program can be interpreted as liked but the user may zap from that channel. We will be under the false assumption that the user is watching the program he clicked on.

- There is a need for extra code or another application on the client/server to collect the data about user actions.
- Because of the current limitations of HbbTV it's not possible to install a broadcast dependent agent on TV. TV manufacturer's app. stores are not publically accessible to upload programs to it.

Looking at the drawbacks of the explicit method, the implicit method seems to be a better choice. But in the context of a personalized EPG, the users are very willing to provide their preferences when they were asked during the usability test. They want to get a cleaner, colorful and filtered list of channels and programs that best suits their taste. The problem of providing false information is often on social medias where one wants to go with the flow. But as mentioned in explicit method the user may not update his preferences if his interest changes. In this project the explicit method of gathering user preferences is used but hybrid technique i.e. both explicit and implicit methods can be used together.

3.4.2 Content and metadata

Content and metadata is discussed in personalization chapter, it is summarized in the following.

3.4.2.1 TV-Anytime

- TV-Anytime is widely used
- It is XML based; XML is extendable
- It is structured and is standardized
- It is vendor independent
- It is transport protocol agnostic
- It has adopted MPEG-7 as its representation format

Based on the above advantages and features discussed in TV-Anytime section its usage along with a recommender system will provide good results and will enhance the user experience. Though in this project due to the lack of TV-Anytime compliant metadata TV-Anytime is not utilized but due to its advantages it is recommendable.

3.4.3 Recommendation Techniques

In this project content based filtering is used where the user's explicit preferences are matched against the content that describes the programs. Depending on the context and the information available CF or CBF or both could be used.

For content based filtering most recommender systems use keyword matching model with TFI-DF weighting [45]. One of the examples is the latent semantic analysis which is described above in section 3.3.1.2.

Item-based CF approach is preferable over User-based CF approach because the user-based approach suffers from scarcity and scalability. Another advantage of item-based approach is that the model of user ratings can be calculated in advance.

For similarity measures both Pearson correlation and cosign similarity is widely used. The test results might be biased by the particular settings but the Recommender systems handbook [45] points out that the best results were achieved by using cosign similarity. However, Sarwar et al. In the paper "*Item-Based Collaborative Filtering Recommendation Algorithms*" [45] states that "*computing similarity using basic cosign similarity in item-based has one important drawback, the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair.*"

4. Prototype development

During this master thesis an interactive EPG prototype was developed that delivers personalized content on HbbTV platform. The prototype is described by the following diagram.

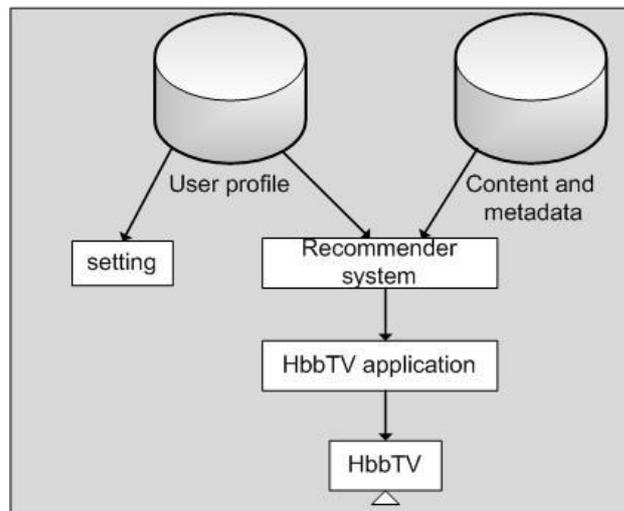


Figure 11: prototype overview

As shown in figure above, 'user profile' is the collection of user's preferences. For testing purpose, we have created some dummy users. The user can provide his preferences explicitly through the setting page. The setting page can be accessed from a computer, a tablet or any smart phone. 'Content and metadata' includes the metadata of TV programs. For this project the contents including metadata is provided by YouSee through a REST API. The 'recommender system' defines how user profile and metadata should be used to deliver personalized content. The 'HbbTV application' determines how to present the recommended content on the HbbTV. To present content, we will involve users in the design of the user interface to get user requirements. In the following, the process of users' involvement in the prototype development is discussed.

4.1 User interface

As described previously in the methodology section, Evolutionary development is chosen for the development of the user interface which takes into account the users feedback. In evolutionary development the design starts with an initial prototype which goes through repeated design-evolution-redesign to refine the prototype [48]. In the introduction of [48] the author states *“For users to evaluate the design of an interactive product effectively, designers must prototype their ideas”*. Furthermore in section Prototyping and Construction [48] states that the users can't tell you what they want, but they can tell you what they don't want when they get the chance to use a prototype.

4.1.1 Low fidelity prototypes

In the early stages of software development low fidelity prototypes such as paper prototyping are useful because they are cheap and simple to produce. Besides being cheap and simple, modification of such prototypes is quick and easy. According to [48] low fidelity prototypes should mainly be used for getting ideas for modification of the software. The author suggests the use of paper prototype in order to know how the user might perform a set of tasks and if our design will help the user perform those tasks. The importance of prototyping and user involvement is also emphasized in other Interaction Design literature such as in *“improving your human-computer interface”* [8]. That is why we started with a low fidelity prototype in our project and ended it with a high fidelity prototype for testing out technical issues. The steps taken from a paper prototype to a fully functional prototype are described here.

4.1.2 Existing EPGs

To get some inspiration we started searching for already implemented commercial EPGs. Most of the existing EPGs we came across are not appealing to the eye. Following are two examples of such EPGs. Both of them are dull and the list of programs are almost scattered on the page. It needs a number of clicks to come back and select another channel if a user in the program list navigates to the right.

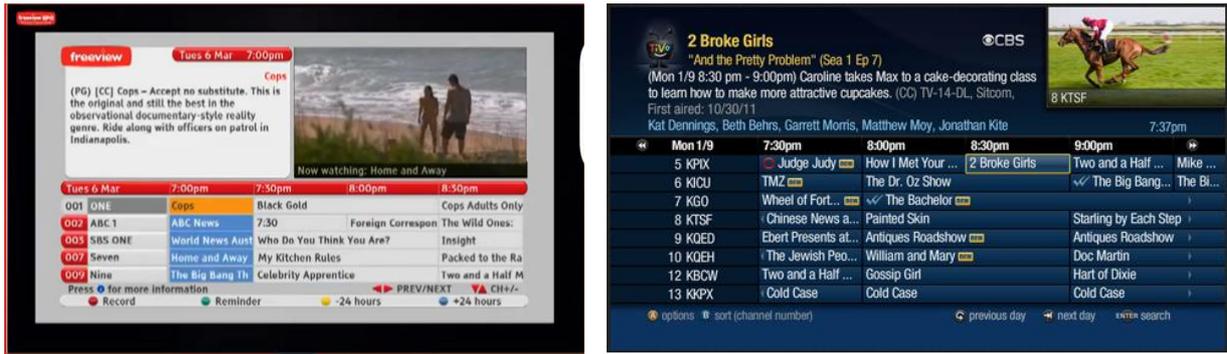


Figure 12: Example of existing EPGs

4.1.3 Appealing EPG

We wanted to come up with a simple yet appealing user interface for an interactive EPG. Experimenting by sketching different paper prototypes were carried out and finally a paper prototype suitable to be exposed to the users was designed. Following sketches illustrate some of our ideas.

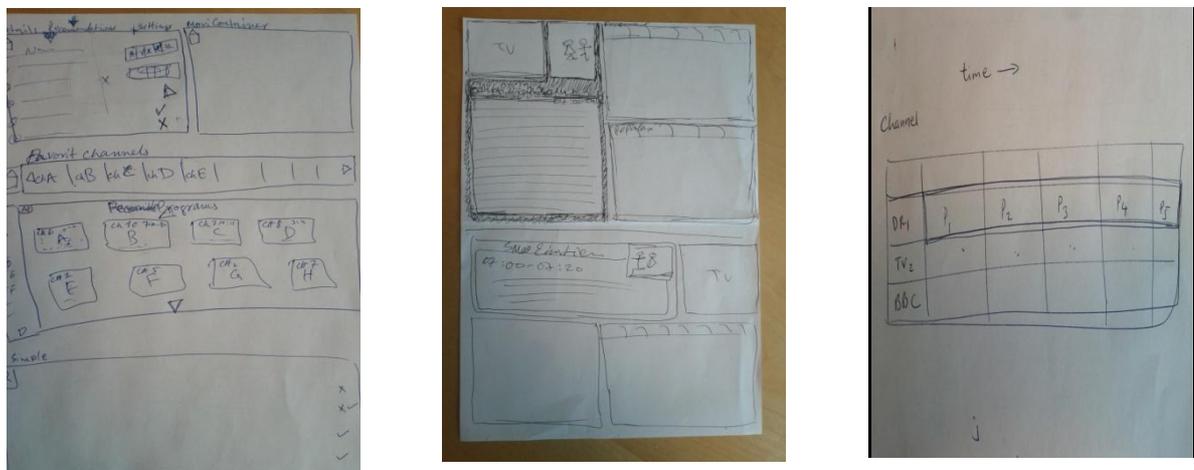


Figure 13: Ideas for initial paper prototype

4.1.3.1 First paper prototype

Based on the above sketches the following paper prototype was designed, printed on paper and exposed to the users. The paper prototype proved to be a good starting point for exploring users' ideas and alternative designs. Users' opinions were collected by exposing this paper prototype to them and asking them questions like what they expect the system will present if they for example select a channel. The

process of testing the prototype and the feedback from users is described in section 5.2 usability test.



Figure 14: First EPG paper prototype

4.1.3.2 Second prototype

Using the design-evolution-redesign principle the interface was redesigned based on the users' comments and opinions. Furthermore a paper prototype of the personalized section MyPage was created. The feedback of the users from the usability tests of MyPage can be found in Testing chapter.



Figure 15: EPG section

MyPage paper prototype was commented by the users and experts as crowded and distracting. Another important drawback of MyPage was the lack of consistency between the First page i.e. EPG and My Page.



Figure 16: MyPage (paper prototype)

4.1.4 From low fidelity to High fidelity

At this point reasonable user feedback and requirements were gathered. It was time to move from low fidelity prototyping to high fidelity prototyping i.e. from paper prototype to real implementation. Based on users' feedbacks, comments and ideas from experts the following prototype was implemented. In this prototype there is a clear separation between popular Today and the recommended content. To keep the consistency with the EPG page the upper left box is used for the description of the programs and the upper right box is used for the currently running program.



Figure 17: MyPage

Usability tests of the personalized MyPage revealed the following problems.

- The usage of green button to toggle/choose between the recommended container and Popular Today was confusing.
- Because of the small size of the programs listed in recommended container the text was not readable from distance.
- Experts from Aalborg University suggested one tabular container
- Experts from YouSee suggested to make it more simple and easy to navigate
- Poor response time, some times more than 3 seconds to navigate to next program
- There was no loading feedback from the system when a large list of programs was loaded

4.1.5 Final version

Based on the data collected during the usability tests, users' feedback from paper prototype and opinions of the ID and Interface experts MyPage was redesigned. Because of the time limitation and positive feedback of the users we decided it to be our final version.

When a program is selected the description is shown in the upper left corner. According to the Interaction Design principles the users start scanning the screen from the upper left corner but the program images are very eye-catching therefore, in order to be noticed, the description is put in the upper left box. In this final prototype another noticeable change from the first prototype is that the channel name is shown in the description and under the program list as well. This decision is made based on user feedbacks. Users often used to forget to which channel the programs belonged.

Following are major improvements in the final version.

- Consistency between EPG and MyPage is retained by
 - Merging the two containers together
 - Introducing a tabular menu
- Clock added
- Response time improved to the response-time limits as suggested by Jakob Nielsen [49]
- Loading feedback is added by showing a circling animation

Following are the screenshots of the final version. In this version it could be clearly noticed that the consistency is preserved.



Figure 18: Login page



Figure 19: EPG section with channel DR 1 selected



Figure 20: MyPage



Figure 21: EPG section with a program from channel TV2 selected

4.2 System requirements

To implement the HbbTV application as shown in figure 11, the system requires

- IRT broadcast server
- HbbTV
- ASI output card (Dektec DTU-215)

4.2.1 IRT broadcast server

IRT broadcast server is developed by Institut für Rundfunktechnik GmbH (IRT) [50], the research center for the German broadcasters. The broadcast server permits to insert the necessary information in the broadcast signal for HbbTV applications. The server supports MPEG transport streams of the DVB standard family including IPTV. The compression methods, MPEG-2 or MPEG-4/AVC can be used to distribute

both SDTV and HDTV. The server can be configured through a GUI, for example if you want to change AIT table, PSI or input services etc.

The output transport stream is provided through one of the following output devices

- Hard disk (integrated recording mode)
- ASI output cards as following:
 - Computer Modules Master FD II (output and recording of input streams supported)
 - Dektec DTA-series and DTU-series (output supported), including direct control for
 - Dektec DTA-107 and DTA-110 QPSK and QAM-modulator boards (setting modulator parameters from the server user interface)
- UDP and RTP-streaming to an adjustable IP-Address: Port

The server configuration can be found in appendix 8.9.

4.2.1.1 Key features

The broadcast server has the following key features: [50]

- Real-time multiplexing of multiple DVB transport streams containing SD/HD-TV
- Remapping and filtering of PIDs and services
- Recreation of PSI, user-configurable reassembly of components and services, generation of services,
- Creation of NIT/SDT/TDT
- Real-time DSMCC object carousel generator for HbbTV (or MHP)
- Simple configuration of module assignment and caching modes
- Support of multi-PID object carousels
- Live-update of whole carousels or modules
- AIT-Editor
- Multiple output options: DVB-ASI, UDP/IP, RTP/IP streams

4.2.1.2 System requirements

- It can run on Windows XP, windows 7, Server 2003, Server 2008
- Dual core processor, 2 GHz (x86-based)
- Minimum 2 GB RAM
- ASI or RF card. Supported are a selection of Dektec and Deltacast cards

4.2.2 ASI output cards

The broadcast server supports Dektec DTA-series and DTU-series. For this project, we used DTU-215.

4.2.2.1 Dektec DTU-215

It is USB-2 based multi-standard modulator that supports QAM-, OFDM- and VSB based modulation standards. It doesn't need any external power adapter because it is powered from the USB-2 bus [51].



Dektec DTU-215

4.2.3 Function requirements

Functional requirement describes specific functions, tasks or behaviors the system must support when the user or another system interacts with it. These requirements are based on previously known requirements and some requirements, which are the result of the users' feedback during prototype testing in section 5.2 usability tests. The functional requirements are listed in the following table.

Functional Requirements (FR)	
FR1	The system shall show red button on HbbTV related program for 5 seconds.
FR2	The system shall show HbbTV services: <i>mypage</i> & <i>EPG</i> , when red button is pressed.
FR3	The system shall allow users to exit.
EPG (TV-Part)	
FR4	The system shall allow users to select channels.
FR5	The system shall allow users to zap to a channel by selecting current program.
Mypage (TV-Part)	

FR6	The system shall allow user to login.
FR7	The system shall display programs based on user's wishlist.
FR8	The system shall allow user to select popular channels.
FR9	The system shall allow user to select categories.
Mysettings (PC-Part)	
FR10	The system shall allow users to login.
FR11	The system shall allow users to create a wishlist.
FR12	The system shall allow users to add channels to wished-channels.
FR13	The system shall allow users to remove channels from wished-channels.
FR14	The system shall allow users to add categories to wished-categories.
FR15	The system shall allow users to remove categories from wished-categories.

Table 4: Functional requirements

4.2.4 Non-functional Requirements

Non-functional requirement describes constrains on various attributes of the functional requirements. The non-functional requirements are listed in the table below:

Non-functional Requirements (NFR)	
NFR1	The application shall be implemented using PHP, JavaScript, CSS, jQuery, SQL and HTML.
NFR2	The system shall use Apache web server.
NFR3	The system shall use IRT broadcast server.
NFR4	The system shall respond within response-time limits [52].
NFR5	The system shall notify the user if there is no Internet.
NFR6	The application shall be extensible.
NFR7	The GUI of HbbTV application shall be appealing.

Table 5: Non-functional requirements

4.3 Use case diagram

On the basis of functional requirements, use case diagrams are designed that capture the system's behavioral requirements. The user is a TV viewer who interacts with the personalized Interactive EPG prototype. Use case diagrams have been divided into four groups as shown below.

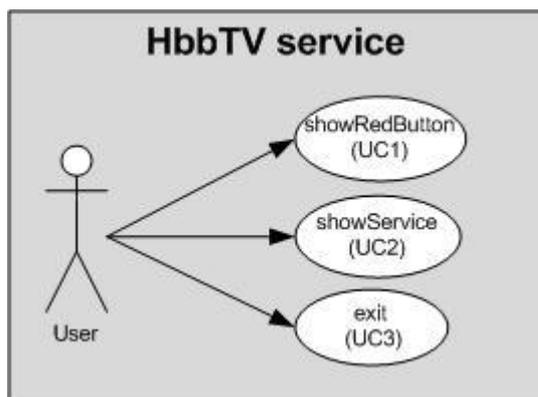


Figure 22: Use case - HbbTV service

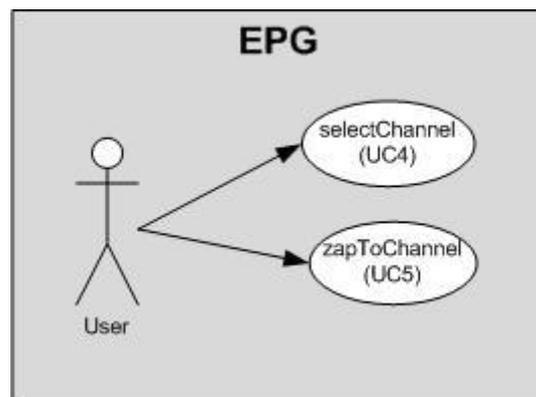


Figure 23: Use case - EPG

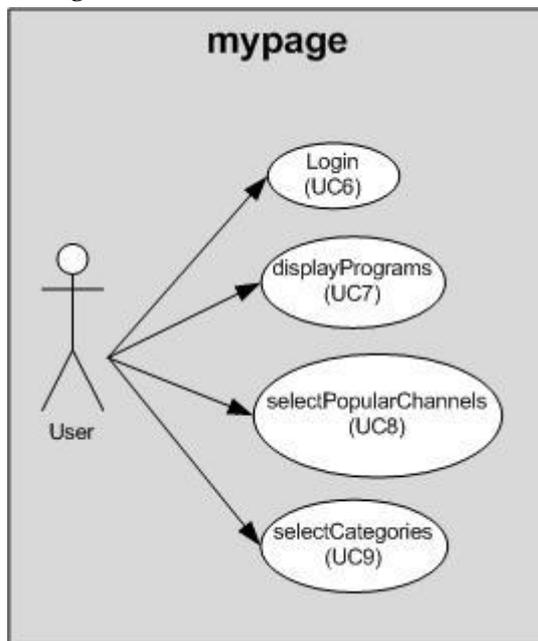


Figure 24: Use case – mypage

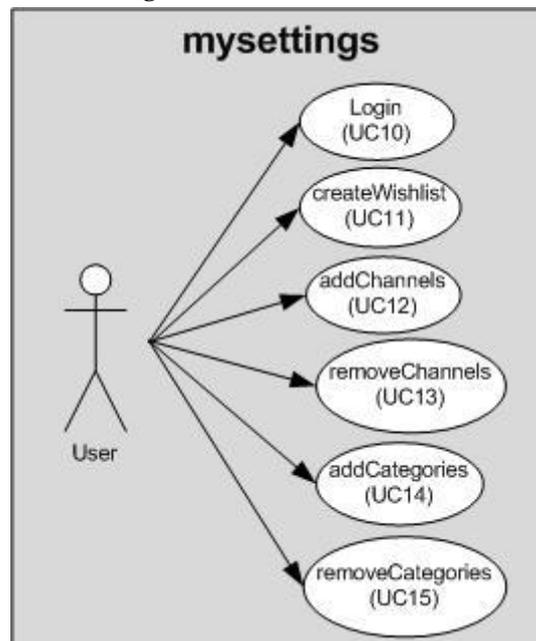


Figure 25: Use case – mysettings

Each use case is described in the appendix 8.2 use case specification section.

4.4 Database Design

We have designed a database to store the user's activities. Basically, the user provides his explicit preferences through the *setting page*. The user has 3 possibilities as mentioned in requirement specifications to express his preferences.

- Create a wishlist, (wishlist are words or phrases entered by the user)
- Add/remove channels
- Add/remove categories

Based on these 3 explicit preferences, a simple database was designed to store user's preferences as shown in figure below. Each user will be identified by a unique identification login id.

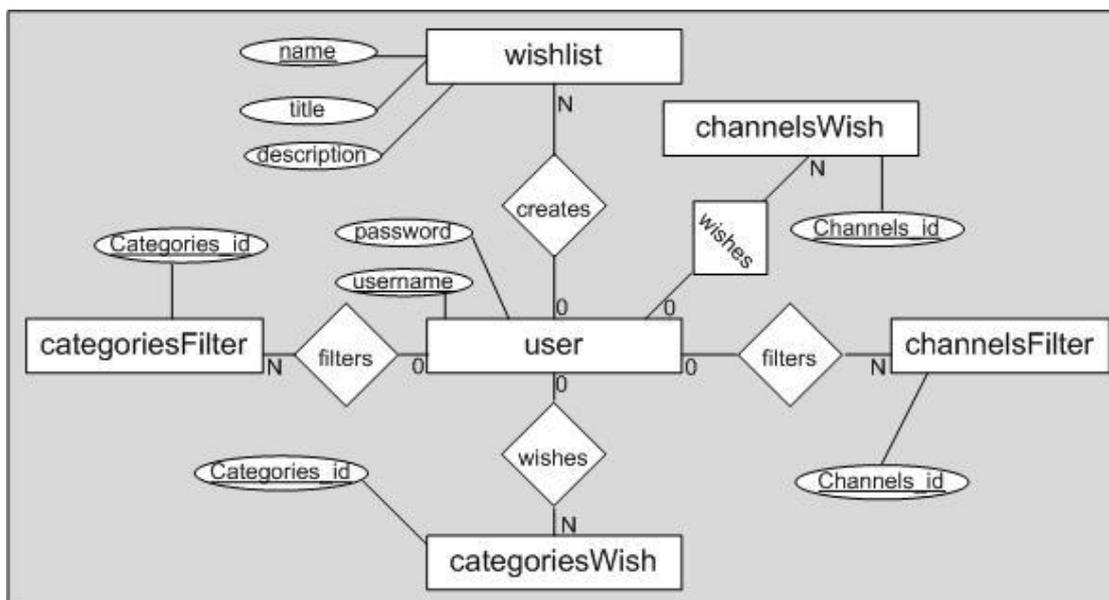
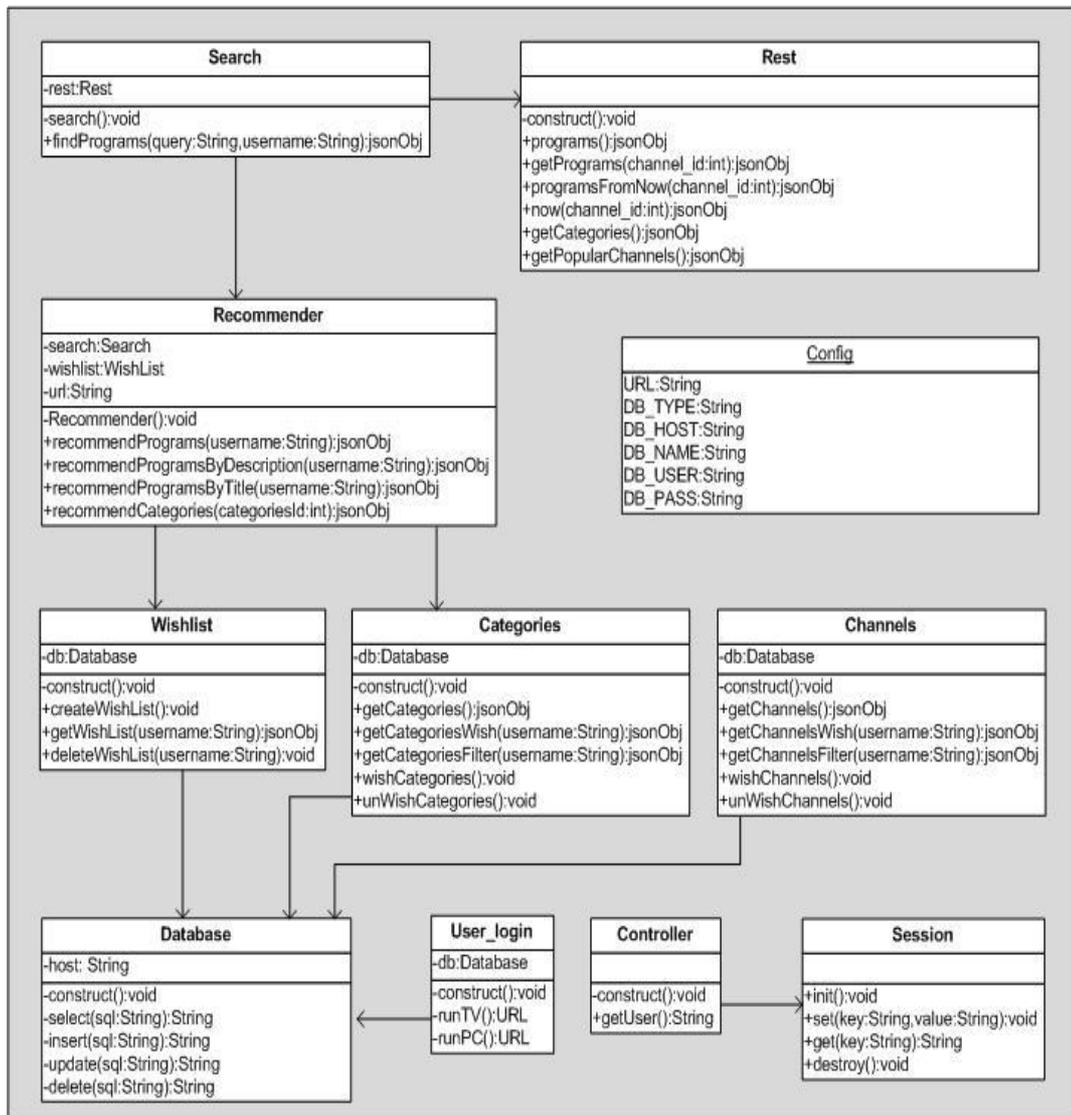


Figure 26: database, ER diagram

By default, the application sets all the channels and categories to the users' wished channels and wished categories. Later the user can filter channels and categories and, can also create a wishlist.

4.5 Class Diagram

The following class diagram gives an overview of a backend part of the system by showing classes and relationships among them. Each class is described in the next section: class responsibilities 4.5.1.



Class diagram

4.5.1 Class responsibility

The following tables describe the responsibilities of some of the major classes contained in the class diagram of backend part in section appendix 8.3. The rest of the tables can be found in the appendix 8.4 under the Class responsibilities section.

Rest
<p><i>Responsible area:</i></p> <p>It handles the REST api provided by YouSee. It has important functions like: to get program detail, to get popular channels, to get categories etc.</p>

Table 6: class responsibilities, Rest

Search
<p><i>Responsible area:</i></p> <p>It uses <i>Rest</i> class to perform its task. It has mainly two functions. Firstly, it provides the programs that match the categories. Secondly, it provides the programs that match a wish word in the programs.</p>

Table 7: class responsibilities, Search

Recommender
<p><i>Responsible area:</i></p> <p>It uses <i>Search, Wishlist and Categories</i> classes to perform its task. It has mainly two functions. Firstly, it recommends the programs that match the wished categories of a particular user. Secondly, it recommends the programs that match a wish word of a particular user in the programs.</p>

Table 8: class responsibilities, Recommender

4.6 Sequence diagram

Sequence diagram is an interaction diagram that describes how operations are carried out, what messages are sent and when. The following diagram describes the operations when a user presses a red button to access to smart EPG HbbTV service. The steps are:

1. HBBTV receives transport stream through broadcast server that contains AIT data, which has an indexing to application.
2. HBBTV downloads application with AUTOSTART
3. Red button is visible. In this case, the application Smart EPG is downloaded.
4. The user presses red button on the remote control.

5. The application requests services to the web server.
6. The HBBTV gets services.

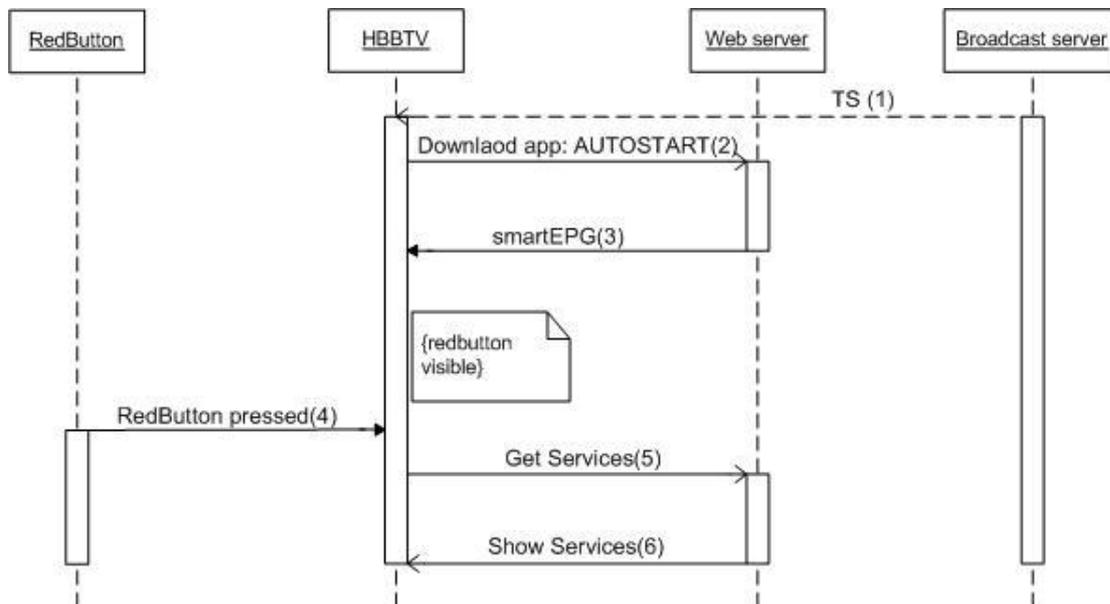


Figure 27: sequence diagram, smart EPG

After steps 6, the user can interact with the services. For example, the user can check the EPG or the user can select one of the users to go to mypage as mentioned in the section use case specifications.

4.7 Implementation

Implementation was carried out both on back-end and front-end simultaneously. This provided the opportunity to test the user-interface while it was evolving. Inspired by the initial prototyping method of the evolutionary model and the user-centric approach we exposed the product at different levels of development to the selected group of users to get feedback for the next prototype. Each new prototype is a refined version of the previous prototype where the feedback from the users is used.

For the implementation of back-end PHP was used as the programming language. For the user interface HTML, CSS, JavaScript and jQuery were used. The source code is delivered to the university administration.

Though the tests has shown good result most of the time i.e. the response time is within the response-time limits described by Jakob Nielsen [52]. But due to the memory and processing limitations of current connected televisions it is better to move client side code to the server as much as possible.

Bandwidth is not a real issue nowadays for downloading images the bottleneck is the limited memory and processing power of the current connected televisions. We tried local cache of the Sony BRAVIA KDL 46Hx755 TV [Appendix8.8 Figure 43] with user-agent Opera 9.8 for our EPG and it was full after a few hours of browsing in channels and program lists.

In this prototype lists of channels and programs are generated using JavaScript. Each list item contains a link to the related image which is accessed from the server. Cache proxies could be used as is done normally for web services. The program information is provided by YouSee which is accessed via a REST interface as JSON objects.

For the frontend, we have designed two block diagrams that show an overview of frontend operations i.e. *Mysettings* & *Mypage*.

Mysettings

This activity is carried out in the PC.

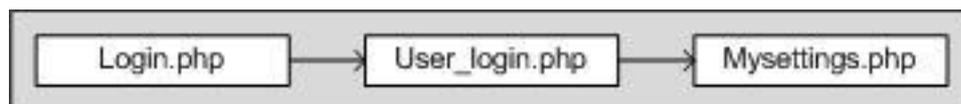


Figure 28: frontend, mysettings

As shown as in figure above, the user logs in thru '*login*' page using his credentials and the '*User_login*' validates the user. When the user is logged in, he can edit his explicit preferences at *mysettings* page. For example, he can add a wish word or wish categories etc.

My page

The user interface on the TV is generated by using the following files. These files are located on the web server. This activity is carried out in the TV.

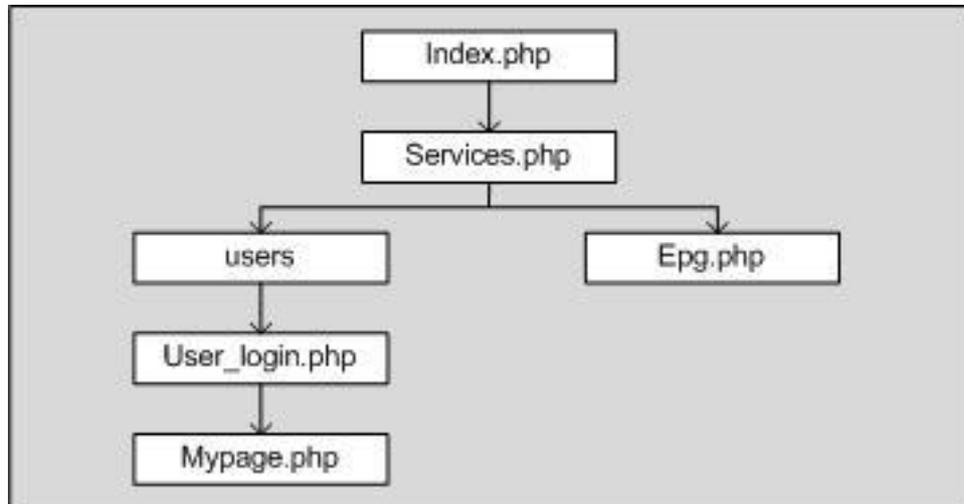


Figure 29: frontend, mypage

As show in figure above, *index.php* uses hbbtv1.1 library that is embedded in the HbbTV terminal. It is also responsible for showing/hiding red button, creating/destroying application and handling remote control key codes. When the user presses red button on the remote control, the user gets the *services*. The user can now select *users* or *epg*. 'User_login' validates the user to go to *mypage*.

5. Testing

For this project black box and usability tests were performed. Black box testing is a method of software testing that tests the functionality of an application or piece of code without looking into the internal structure of the code. The purpose of this test is not how the code works but the focus is on the expected result of the application. In user-centered interaction design usability testing is used which is a technique that evaluates a product by testing it on the users. Usability testing is performed so that any significant issues can be identified and addressed.

5.1 Black box tests

These tests were performed in order to make sure that our system functions according to our expectations and fulfill the system requirements. Results of the requirement tests can be found in appendix 8.1

FR	Result	References
FR1	Passed	Table 10: Requirement tests, FR1
FR2	Passed	Table 11 Requirement tests, FR2
FR3	Passed	Table 12: Requirement tests, FR3
FR4	Passed	Table 13: Requirement tests, FR4
FR5	Passed	Table 14: Requirement tests, FR5
FR6	Passed	Table 15: Requirement tests, FR6
FR7	Passed	Table 16: Requirement tests, FR7
FR8	Passed	Table 17: Requirement tests, FR8
FR9	Passed	Table 18: Requirement tests, FR9
FR10	Passed	Table 19: Requirement tests, FR10
FR11	Passed	Table 20: Requirement tests, FR11
FR12	Passed	Table 21: Requirement tests, FR12

FR13	Passed	Table 22: Requirement tests, FR13
FR14	Passed	Table 23: Requirement tests, FR14
FR15	Passed	Table 24: Requirement tests, FR15

Table 9: black box tests

5.2 Usability test

TV is a medium which is used by population of different ages, taste, occupation and education. To test the prototype by inviting different groups of people at different times was very time consuming. Because of the short time span of the project it was desirable to carry the tests in an easy and less time consuming manner. The first low-fidelity as described in prototype chapter was a paper prototype which could conveniently be carried around and exposed to the users anywhere and at anytime. We used the think aloud technique when possible. In think aloud users are encouraged to speak up their intention before or while they act e.g. "I am selecting channel DR1 and I am expecting the system will present ..." Users' intentions, navigation, selection, clicking and the outcome of their actions was observed and feedback was collected. The feedback was used to develop the next refined version of the prototype.

The prototype was tested by the users with the following properties.

Age: 13-55

Gender: both Male and Female

Literary level: secondary school to University

Occupation: unskilled and skilled; lecturers, professors, IT engineers, design, students

To get the honest comments and suggestion of the users it was explained to them that the purpose is to test the interface and not the skills of the users. Furthermore, that we are interested to find the weakness of the system and where or when the system misleads the user. Briefing the user is important because the studies have shown that most often the users provide wrong answers. They don't want to be considered ignorant and not familiar with the technology.

The test was carried out in the following manner:

First paper prototype was carried by each team member and was exposed to family members, friends and other people. We tried to select users representing different age, education and technology awareness.

The software prototype from second to last version which went through a design-evaluation-redesign cycle was mainly tested at the university by students the technology aware people, interaction design experts and experts from YouSee. The results could be biased but their experience in the field and expertise in interaction design pointed us in the right direction. The user feedbacks can be found in [Appendix 8.7](#)

5.2.1 Feedback

In the following the feedback from the users in the form of comments and suggestions for the first second and final version of the prototype is summarized.

5.2.1.1 *First paper prototype*

- The simplicity of the interface was liked by most of the users.
- Elderly users wanted a larger font size
- The description with blue background and white fonts was not readable from distance. Especially by elderly users.
- Arrow or some kind of indicator needed to show if there are more elements in the program list

5.2.1.2 *Second prototype*

Second prototype was divided in two parts EPG and MyPage. EPG was the real software implementation based on the user feedbacks acquired from the paper prototype test. MyPage was another paper prototype for the personalized section of the system. The feedback from the usability test of the second prototype is as follows.

- After selecting a channel user was supposed to press down arrow to navigate in the program menu. It was annoying for most of the users.
- It was desirable by most of the users that the program menu should be auto selected
- Users used to forget to which channel the displayed programs belong
- System did not tell if there were more programs in the list

- The selected program needs to be zoomed
- The tab of the selected channel should remain open until another channel is chosen
- Program description does not catch the user attention
- Personalized section was commented as
 - Crowded; too many items
 - A clear separation between the recommended contents and popular channels was suggested
 - Inconsistent; very different from the first page (ID experts and IT professionals)

5.2.1.3 Final prototype

Because of the limited time of the project it was decided to stop the tests and use the feedback to implement the final version of the prototype. The final version as described in the prototyping chapter included a personalized section named MyPge with a program list based on user's wish list and a tabular list of categories based on users preferences.

Based on the feedback from the second prototype testing following improvements were made.

- Tabs remain open until another channel is selected
- Selected channel's name is also displayed under program list to show which channel the programs belong to
- Arrows are displayed to indicate that the program list contains more elements
- The users can now jump to the first or last program in the program list by pressing the yellow or blue button
- My page now has only one row of programs as EPG page
- Background color of the description is changed to light gray with black font
- Consistency between EPG and MyPage is retained

6. Conclusion

EPG was envisioned as a vital service in digital TV. However the EPG in its current form has not been a success. People continue zapping to select and change programs. The personalized interactive EPG prototype developed during this master thesis is an effort to change this behavior. The prototype shows that Hybrid Broadcast Broadband Televisions can benefit from the advantages of web-services. One of the popular features of the web-services is the personalization aspect. To benefit from this feature the GUI of our EPG- prototype implemented according to HbbTV specifications is interfaced with the web-services.

This prototype shows that the combination of program metadata, user profiles and recommender techniques can result in a more appealing and personalized EPG that will enhance the user experience. This was observed during the usability tests of the prototype which were carried as part of the user-centric design approach that is used for the development of this prototype.

In order to personalize the EPG, the system uses a wide range of information about the users and broadcasted programs. This information can be modeled based on the selection of recommendation techniques. Recommender techniques can mainly be divided in two classes namely Collaborative Filtering and Content based filtering. For the personalized interactive EPG we have chosen the Content Based Filtering recommendation technique using the keyword matching model. As the focus of this thesis is mainly on an appealing personalized interactive EPG utilizing web-services therefore instead of the complex recommender techniques a simpler keyword matching algorithm is used.

Though, in personalized interactive EPG prototype we have used CBF method but for real implementation hybrid technique could be used i.e. both CBF and CF methods. A hybrid technique is necessary because there could be situations where there is not much content associated with the items for CBF. In such a case Item-based CF will provide the recommendations and in situations where the model of user ratings in CF is not developed yet the CBF will provide the recommendations.

Recommender systems rely on item/program information and user profile. For collecting user information we have chosen explicit method because the explicit method is more accurate, and users are in control of their choices. The limitations, like less memory and processing power of current HBBTV-sets makes the implicit method less desirable. But if some users do not provide their preferences, the system

cannot build the user profile therefore in real implementation it is better to use implicit method along with explicit.

The contents extended with metadata are easier to find, select and acquire by software agents. For the live EPG in a broadcast environment, TV-Anytime can provide programs information like genre, title, actors, program description, time and date in the form of metadata. This structured information could be exploited by the recommender system to deliver personalized content.

Compared to the time and resources used, the performance of our prototype is satisfactory in response time and usability. But, sometimes the response time exceeds the acceptable limits. This happens when large lists of programs with rich multimedia content are generated on client side. JavaScript that runs in the TV browser for generating these lists, the memory and processing power of TV are the bottlenecks. Code issues can be solved by moving the program-list generation to the server if possible but the terminal issues will remain until more powerful and advanced TV sets are available in the market.

7. Future work

Most of the design described in the product development chapter is implemented during this project but some parts like creating user profile implicitly remains and some additional work can be done to improve the system. Following is a list of existing features that can be improved and some new features which can be added to the system.

Work for the future:

- Make it possible to rate programs using the remote control
- Use the Implicit method together with explicit to create accurate user preferences.
- Improve response time
- Show wish list on TV
- Show soft keyboard on TV
- Develop a mobile app for setting page
- Integrate with users profile on social networks
- Recommend items in descending order based on its rating weight
- Use thumbs up and thumbs down instead of wished and unwished

8. Appendices

8.1 Requirements tests

Requirement test	
ID	FR1 (show red button)
Preconditions	I) Broadcast server is running II) Internet is connected III) Web server is running
Test description	The user zaps to HbbTV enabled channel e.g. Hello Test
Expected result	Red button visible for 5 seconds
Actual result	Red button visible for 5 seconds (Figure 34: show red button)
Result	Passed

Table 10: Requirement tests, FR1

Requirement test	
ID	FR2 (show service)
Preconditions	I) FR1 exists.
Test description	The user presses red button on the remote control.
Expected result	Mypage and EPG
Actual result	Mypage and EPG (Figure 35: show services)
Result	Passed

Table 11: Requirement tests, FR2

Requirement test	
ID	FR3 (exit service)
Preconditions	I) FR2 exists.
Test description	The user presses red button on the remote control.
Expected result	Broadcast content and red button visible
Actual result	Broadcast content and red button visible (Figure 34: show red button).
Result	Passed

Table 12: Requirement tests, FR3

Requirement test	
ID	FR4 (select channels)
Preconditions	I) FR2 exists. II) The user selects EPG
Test description	The user selects channel.
Expected result	Displaying programs of selected channel.
Actual result	Displaying programs of selected channel. (Figure 38: smart EPG)
Result	Passed

Table 13: Requirement tests, FR4

Requirement test	
ID	FR5 (zap channel)
Preconditions	I) FR4 exists.
Test description	The user selects a program currently running on a specific channel.
Expected result	Zaps channel

Actual result	Zaps channel (Figure 38: smart EPG)
Result	Passed

Table 14: Requirement tests, FR5

Requirement test	
ID	FR6 (user login)
Preconditions	I) FR1 & FR2 exist.
Test description	The user selects a user.
Expected result	Display user's page
Actual result	User's page (Figure 37: login user)
Result	Passed

Table 15: Requirement tests, FR6

Requirement test	
ID	FR7 (show wish list programs)
Preconditions	I) FR1 & FR2 exist.
Test description	The user selects a wishlist.
Expected result	Display programs based on user's preferences
Actual result	Programs related to wish list (Figure 42: mypage wished programs)
Result	Passed

Table 16: Requirement tests, FR7

Requirement test	
ID	FR8 (select popular channels)
Preconditions	I) FR1 & FR2 exist.
Test description	The user selects popular channels.
Expected result	Display popular channels.

Actual result	Display popular channels (Figure 41: mypage underholding loading)
Result	Passed

Table 17: Requirement tests, FR8

Requirement test	
ID	FR9 (select categories)
Preconditions	I) FR1 & FR2 exist.
Test description	The user selects categories.
Expected result	Display programs related to selected category.
Actual result	Display programs related to selected category (Figure 39: mypage film loading)
Result	Passed

Table 18: Requirement tests, FR9

Requirement test	
ID	FR10 (user login to mysetting page)
Preconditions	I) Database & Apache server running II) User exists in the database.
Test description	The user logs in login page.
Expected result	User is logged in.
Actual result	User is logged in (Figure 31: mysettings login page)
Result	Passed

Table 19: Requirement tests, FR10

Requirement test	
ID	FR11 (create wishlist)
Preconditions	I) The user is logged in

Test description	The user creates wish list.
Expected result	Display created wish list.
Actual result	Display created wish list (Figure 32: create wish list)
Result	Passed

Table 20: Requirement tests, FR11

Requirement test	
ID	FR12 (add channels)
Preconditions	I) The user is logged in
Test description	The user adds channels to his wished channels.
Expected result	Display added channels to his wished channels.
Actual result	Display added channels to his wished channels (Figure 33: add/remove channels)
Result	Passed

Table 21: Requirement tests, FR12

Requirement test	
ID	FR13 (remove channels)
Preconditions	I) The user is logged in
Test description	The user removes channels from his wished channels.
Expected result	Display removed channels to his unwished channels.
Actual result	Display removed channels to his unwished channels. (Figure 33: add/remove channels)
Result	Passed

Table 22: Requirement tests, FR13

Requirement test	
ID	FR14 (add categories)
Preconditions	I) The user is logged in
Test description	The user adds categories to his wished categories.
Expected result	Display added categories to his wished categories.
Actual result	Display added categories to his wished categories. (Figure 33: add/remove channels)
Result	Passed

Table 23: Requirement tests, FR14

Requirement test	
ID	FR15 (remove categories)
Preconditions	I) The user is logged in
Test description	The user removes categories from his wished categories.
Expected result	Display removed categories to his unwished categories.
Actual result	Display removed categories to his unwished categories (Figure 33: add/remove channels)
Result	Passed

8.2 Use case specification

In the following tables the behavior of the system is described according to the use cases realized on the basis of functional requirements of the system. It is the description of section 4.3-use case diagram.

Use case: showRedButton
ID: UC1
Brief description: The system displays the red button on broadcast content for 5 seconds.
Primary actors: User
Secondary actors: None
Preconditions: I) Broadcast server is running. II) Webserver is running. III) Internet connection exists.
Main flow: I) It displays red button on the broadcast content. II) The user presses 'red button' from remote control.
Post-conditions: A presentation of HbbTV services: <i>mypage</i> and <i>EPG</i> .
Alternative flow: I) Users press the red button before it shows up for the first time. II) No Internet
Post-Conditions: I) Nothing happens on pressing red button. II) It displays message, ' <i>internet download fail</i> '.

Table 24: UC1 specification

Use case: showService
ID: UC2
Brief description: The system displays the services: <i>mypage</i> and <i>EPG</i> .
Primary actors: User

Secondary actors: None
Preconditions: UC1 exists.
Main flow: A representation of the HbbTV services: <i>mypage</i> & <i>EPG</i> .
Post-conditions: A presentation of HbbTV services: <i>mypage</i> and <i>EPG</i> .
Alternative flow: I) No Internet
Post-Conditions: I) It displays message, ' <i>internet download fail</i> '.

Table 25: UC2 specification

Use case: exit
ID: UC3
Brief description: The system allows users to exit from HbbTV service.
Primary actors: User
Secondary actors: None
Preconditions: UC1 & UC2 exist.
Main flow: The user presses red button to exit the application.
Post-conditions: A presentation of broadcast content.
Alternative flow:

I) No Internet
Post-Conditions: I) It displays message, ' <i>internet download fail</i> '.

Table 26: UC3 specification

Use case: selectChannels
ID: UC4
Brief description: The system allows the user to select channels from interactive EPG.
Primary actors: User
Secondary actors: None
Preconditions: UC1 & UC2 exist.
Main flow: I) The user selects EPG. II) The user selects a channel.
Post-conditions: I) The broadcast content minimizes. II) The system displays programs on a specific channel from currently running to 6 a.m.
Alternative flow: I) No Internet II) No program
Post-Conditions: I) It displays message, ' <i>internet download fail</i> '. II) It displays message, ' <i>currently no program available</i> '.

Table 27: UC4 specification

Use case: zapToChannel
ID: UC5
Brief description: The system allows the user to zap to specific channel.
Primary actors: User
Secondary actors: None
Preconditions: UC1, UC2 & UC4 exist.
Main flow: I) The user selects a channel. II) The user selects a program. III) The user presses 'OK' on a program.
Post-conditions: I) The system displays programs on a specific channel from currently running to 6 a.m. II) The system displays a program description of the selected program. III) The system zaps to the specific channel.
Alternative flow: I) No Internet II) No program
Post-Conditions: I) It displays message, ' <i>internet download fail</i> '. II) It displays message, ' <i>currently no program available</i> '.

Table 28: UC5 specification

Use case: login
ID: UC6
Brief description: The system allows the user to login to mypage.
Primary actors: User
Secondary actors: None
Preconditions: UC1 & UC2 exist.
Main flow: I) The user selects a user.
Post-conditions: I) Broadcast content is minimized. II) Representation of wishlist, popular channels and categories.
Alternative flow: I) No Internet
Post-Conditions: I) It displays message, ' <i>internet download fail</i> '.

Table 29: UC6 specification

Use case: displayPrograms
ID: UC7
Brief description: The system displays programs based on the user's wish list.
Primary actors: User
Secondary actors: None
Preconditions:

<p>I) UC1 & UC2 exist.</p> <p>II) The user has already created wishlist thru setting page.</p> <p>III) Relevant programs based on user's wishlist exist.</p> <p>IV) The user is logged in.</p>
<p>Main flow:</p> <p>I) The user selects wishlist.</p>
<p>Post-conditions:</p> <p>I) The system displays programs related to user's wishlist.</p>
<p>Alternative flow:</p> <p>I) No Internet</p> <p>II) No wished programs.</p>
<p>Post-Conditions:</p> <p>I) It displays message, '<i>internet download fail</i>'.</p> <p>II) '<i>No program available</i>'</p>

Table 30: UC7 specification

Use case: selectPopularChannels
ID: UC8
<p>Brief description:</p> <p>The system displays popular channels.</p>
Primary actors: User
Secondary actors: None
<p>Preconditions:</p> <p>I) UC1 & UC2 exist.</p> <p>II) The user is logged in.</p>
<p>Main flow:</p> <p>I) The user selects popular channels.</p>

<p>Post-conditions:</p> <p>I) Broadcast content is minimized.</p> <p>II) The system displays channels by popularity order.</p>
<p>Alternative flow:</p> <p>I) No Internet</p>
<p>Post-Conditions:</p> <p>I) It displays message, '<i>internet download fail</i>'.</p>

Table 31: UC8 specification

Use case: selectCategories
ID: UC9
<p>Brief description:</p> <p>The system displays wished categories of a particular user.</p>
Primary actors: User
Secondary actors: None
<p>Preconditions:</p> <p>I) UC1 & UC2 exist.</p> <p>II) The user is logged in.</p>
<p>Main flow:</p> <p>I) The user selects categories.</p>
<p>Post-conditions:</p> <p>I) Broadcast content is minimized.</p> <p>II) The system displays programs related categories.</p>
<p>Alternative flow:</p> <p>I) No Internet</p>
<p>Post-Conditions:</p> <p>I) It displays message, '<i>internet download fail</i>'.</p>

Table 32: UC9 specification

Use case: login
ID: UC10
Brief description: The system allows users to login to mysetting page.
Primary actors: User
Secondary actors: None
Preconditions: I) Apache server running II) Database exists. III) User exists.
Main flow: I) The user logs in to mysetting page thru the browser.
Post-conditions: I) The system displays wished channels, unwished channels, wished categories, unwished categories and wishlist if exists.
Alternative flow: I) Not logged in.
Post-Conditions: I) 'check username'

Table 33: UC10 specification

Use case: createWishlist
ID: UC11
Brief description: The system allows users to create wishlist.

Primary actors: User
Secondary actors: None
Preconditions: I) Apache server running II) User is already logged in.
Main flow: I) The user creates a wish list. E.g. football.
Post-conditions: I) The system displays user's wishlists. The user can also delete his wishlists.
Alternative flow: I) The wishlist already exists.
Post-Conditions: I) <i>'The same wishlist already exists.'</i>

Table 34: UC11 specification

Use case: addChannels
ID: UC12
Brief description: The system allows users to add channels to his wished channels.
Primary actors: User
Secondary actors: None
Preconditions: I) UC10 exists
Main flow: The user adds channels to his wished channels.
Post-conditions: I) The system adds channels to his wished channels and removes from his unwished

channels.
Alternative flow: I) No channels added to wished channels.
Post-Conditions: I) 'Check connection'.

Table 35: UC12 specification

Use case: removeChannels
ID: UC13
Brief description: The system allows users user to remove channels.
Primary actors: User
Secondary actors: None
Preconditions: I) UC10 exists. II) Wished channels added.
Main flow: I) The user removes channels from his wished channels.
Post-conditions: I) The application removes channels from user's wished channels and adds to his unwished channels.
Alternative flow: I) No channels removed.
Post-Conditions: I) 'Check connection'.

Table 36: UC13 specification

Use case: addCategories
ID: UC14
Brief description: The system allows users to add categories to his wished categories.
Primary actors: User
Secondary actors: None
Preconditions: I) UC10 exists
Main flow: The user adds categories to his wished categories.
Post-conditions: I) The system adds categories to the user's wished categories and removes from unwished categories.
Alternative flow: I) No categories added to wished categories.
Post-Conditions: I) 'Check connection'.

Table 37: UC14 specification

Use case: removeCategories
ID: UC15
Brief description: The system allows users to remove categories from his wished categories.
Primary actors: User
Secondary actors: None
Preconditions:

I) UC10 exists
Main flow: The user removes categories from his wished categories.
Post-conditions: I) The system adds categories to the user's unwished categories and removes from wished categories.
Alternative flow: I) No categories added to wished categories.
Post-Conditions: I) 'Check connection'.

Table 38: UC15 specification

8.3 Class diagram

This is a class diagram, which will be described in the section 8.4.

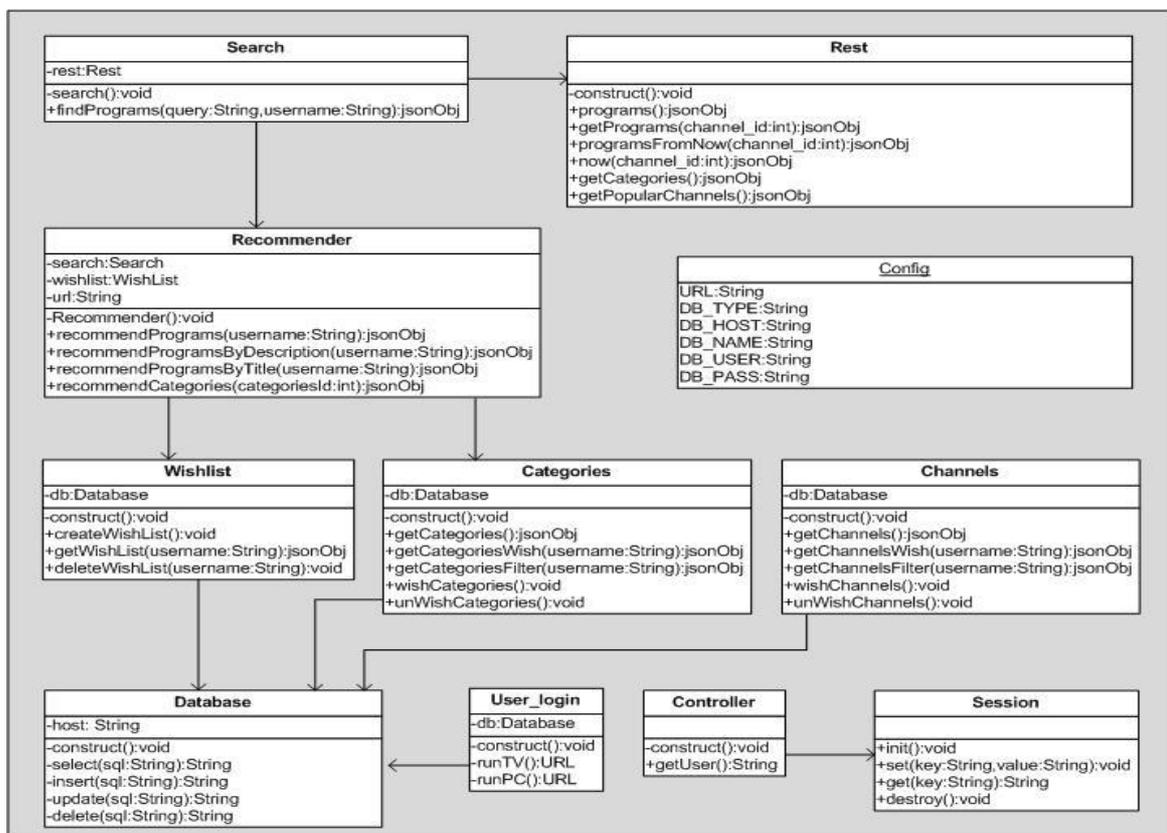


Figure 30: class diagram

8.4 Class responsibilities

This is the description of classes based on class diagram in section 8.3.

Wishlist
<i>Responsible area:</i> It uses <i>Database</i> class to perform its task. It has three functions: create, get and delete wishlist of a particular user.

Table 39: class responsibilities, Wishlist

Categories
<i>Responsible area:</i> It uses <i>Database</i> class to perform its task. Its functions are: wish and unwish categories and get wished and unwished categories of a particular user.

Table 40: class responsibilities, Categories

Channels
<i>Responsible area:</i> It uses <i>Database</i> class to perform its task. Its functions are: wish and unwish channels and get wished and unwished channels of a particular user.

Table 41: class responsibilities, Channels

Database
<i>Responsible area:</i> It is responsible for connecting to database and allows invoking different methods needed for SQL such as, select, delete, update etc.

Table 42: class responsibilities, Database

User_login
<i>Responsible area:</i> It uses <i>database</i> class to perform its task. It validates user and, it decides whether to run in TV or in PC when user is logged in.

Table 43: class responsibilities, User_login

Controller
<i>Responsible area:</i> It uses <i>Session</i> class to perform its task. It provides who is logged in.

Table 44: class responsibilities, Controller

Session
<i>Responsible area:</i> It is responsible for session. E.g. session start, destroy session etc.

Table 45: class responsibilities, Session

Config
<i>Responsible area:</i> It is responsible for configuration. It allows editing hostname, username, password, database name etc.

Table 46: class responsibilities, Config

8.6 Action Item List

Item no.	Tasks	Responsible
1	Personalization	Padam
2	Recommender systems	Homayoon
3	MPEG-7	Padam
4	TV-Anytime	Homayoon
5	Database Design	Padam
6	User Interface Design	Homayoon
7	The rest of the Report e.g. Problem formulation, Methodology, Implementation, testing and conclusion	Group

Table 48: Action item list

8.7 Usability test feedback

Experts

- Red button is in the forgotten corner i.e right bottom corner of the TV screen.
- Use users instead of login for the name of the login button because it is not normal to login in the TV.
- Use yellow and green button of the remote control to select user menu and EPG menu instead of using arrow keys to navigate to it.
- Instead of a dropdown list of users show it directly on TV screen as buttons or clickable/selectable areas.
- Add loading feedback animation if the page takes longer time to load
- People expect google speed in response time
- First Item and Last Item may not be understood by the users (In this prototype they are used for the first program in the list and last program in the list of programs)
- It should be possible to see ones wish list in the TV

Technology aware users

- Use thin borders
- The corners of the boxes are very round make it less round
- The color combination is good
- Font size is ok

Aalborg university students

- Tabs should remain open until another channel is selected
- When hovering over tabs the background color should change
- MyPage,s design needs to be changed to make it consistent with the EPG page
- Some student wanted the description to be shown under the program when it is selected.
- Some users suggested images instead of buttons with text

The system was considered useful and easy to use once it was used for a while by all the above groups.

They said that they would like to use it if they acquired HBBTV.

8.8 Black box tests

Black box tests were performed using browser and TV emulator. Appendix 8.8.1, 8.8.2 and 8.8.3 contains the results of these tests.

8.8.1 Mysettings page tests

login page

The login was tested by providing the username and password as shown in the figure below.

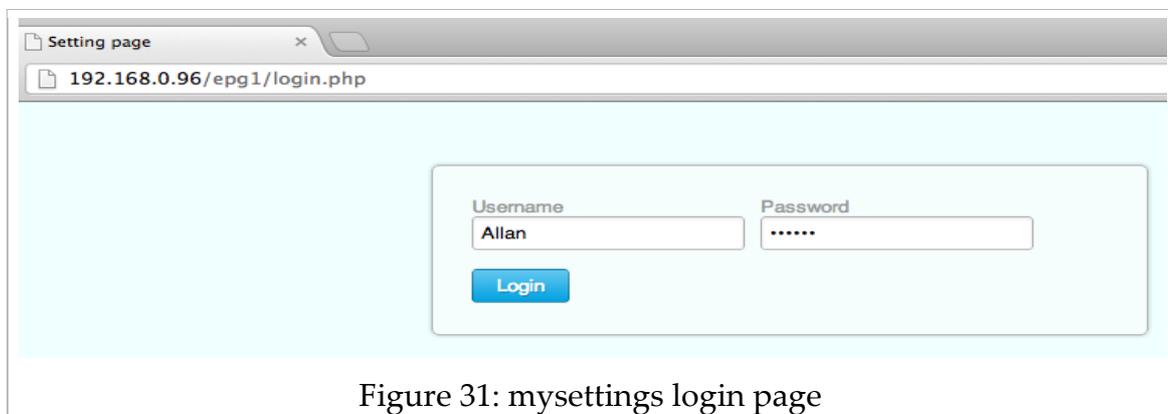


Figure 31: mysettings login page

Create wish list

The following is the result of the wishlist test.

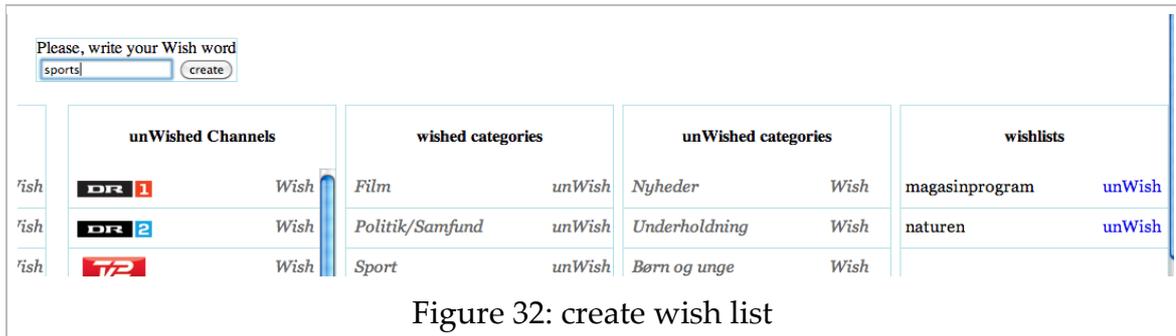


Figure 32: create wish list

Add/remove channels/categories

The user clicks at “wish” to add channels or categories and at “unwish” to remove channels or categories.

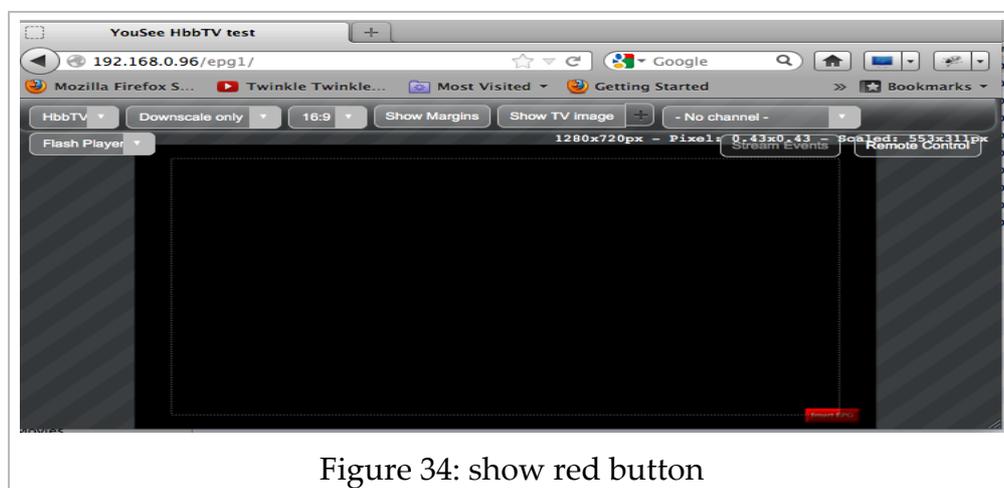


Figure 33: add/remove channels

8.8.2 Tests carried out in the TV emulator

Show red button

The test is performed in the Firefox browser. The application shows the ‘Red button’ for 5 seconds for the first time.



Show service

The user presses red button on the remote control to get the services: users & EPG. The user selects users from the list to login. The following is the result of the test.

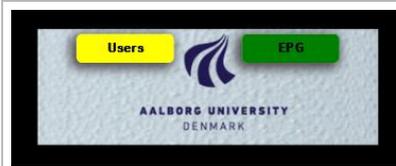


Figure 35: show services



Figure 36: select users



Figure 37: login user

EPG

When the user selects 'EPG', the application displays the EPG page



Figure 38: smart EPG

My page film loading

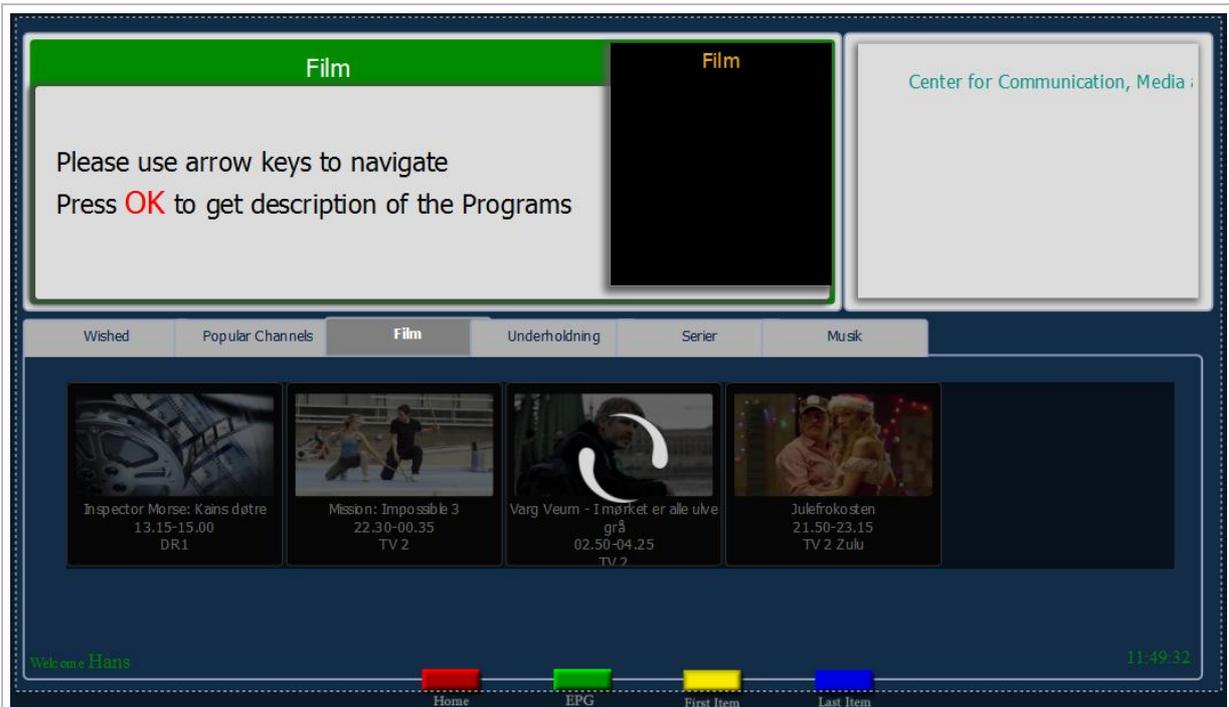


Figure 39: mypage film loading

My page series selected program

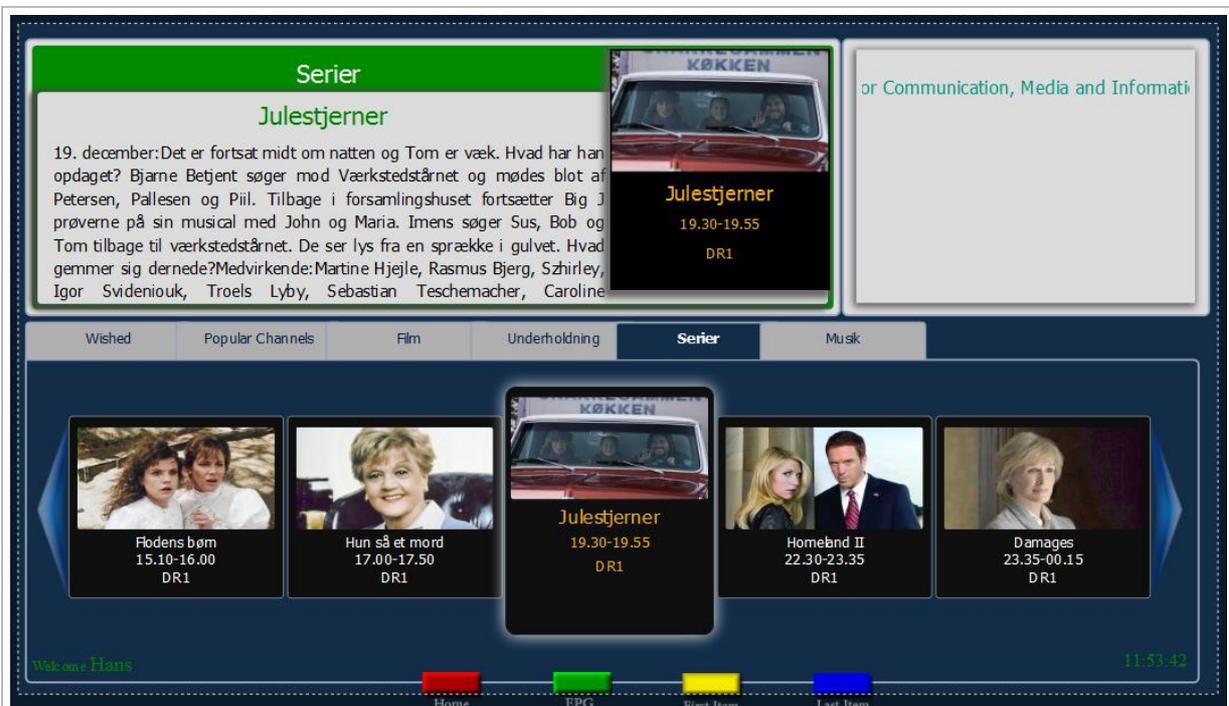


Figure 40: mypage series selected program

Mypage underholding loading

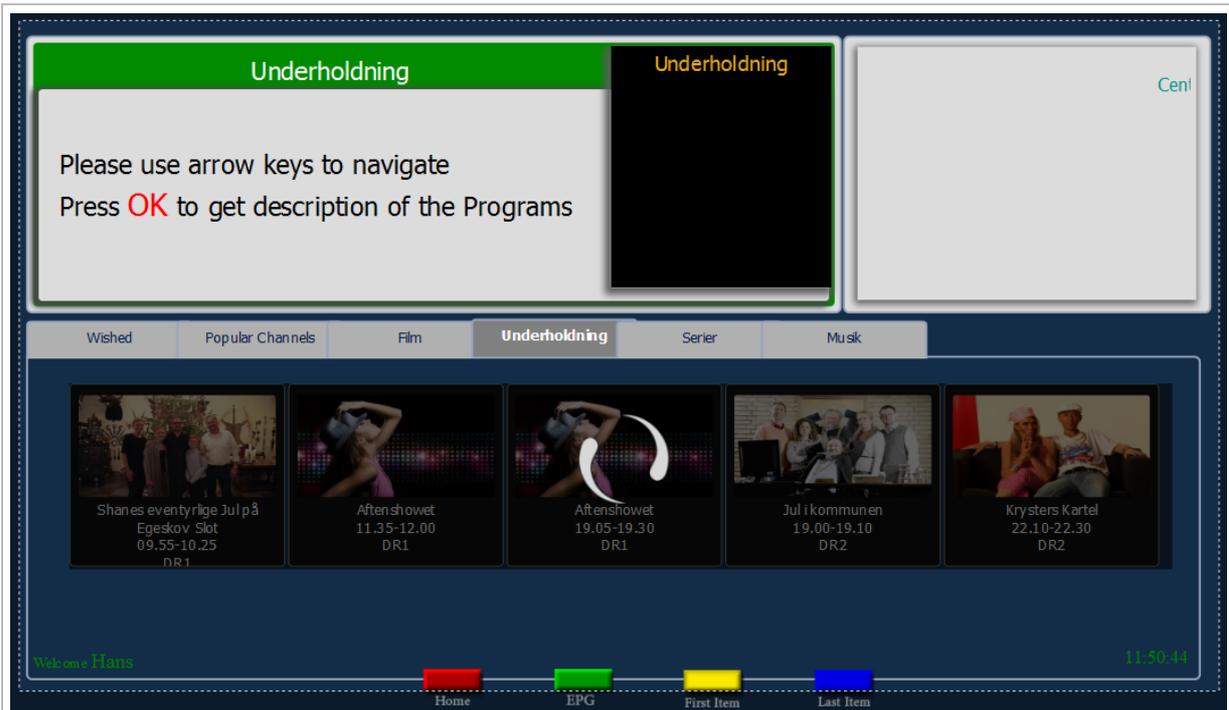


Figure 41: mypage underholding loading

Mypage wished programs

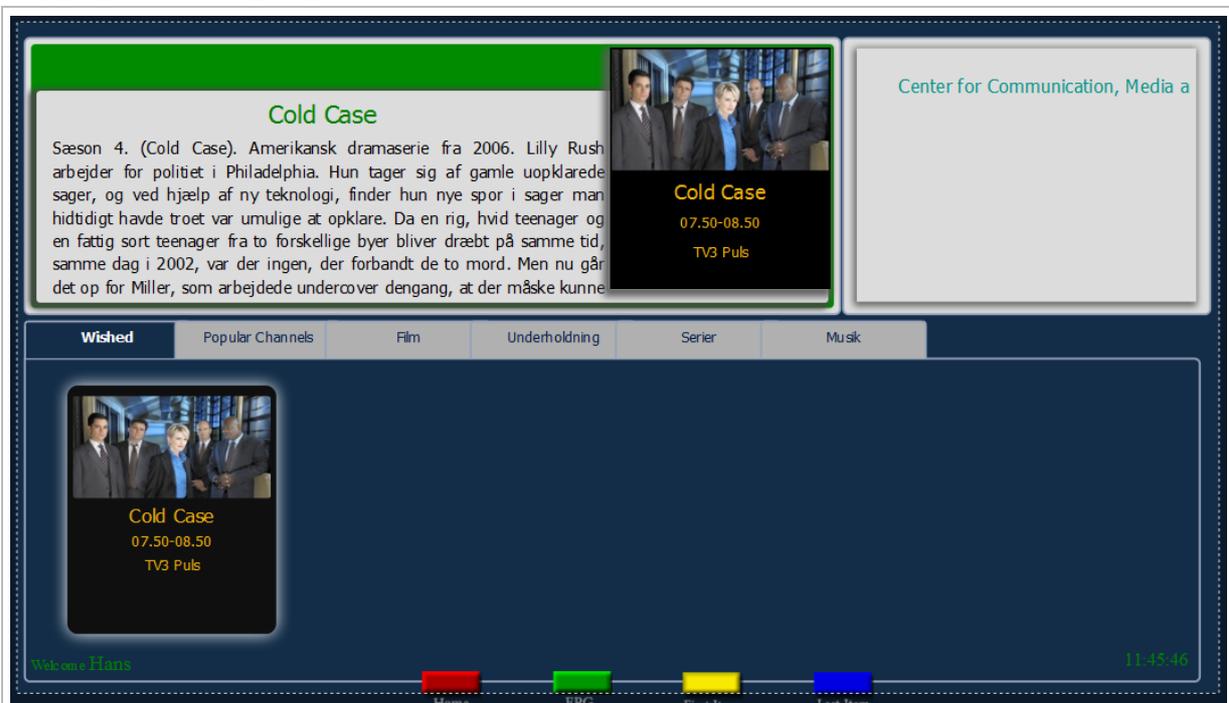


Figure 42: mypage wished programs

Sony Bravia with HbbTV 1.1.1

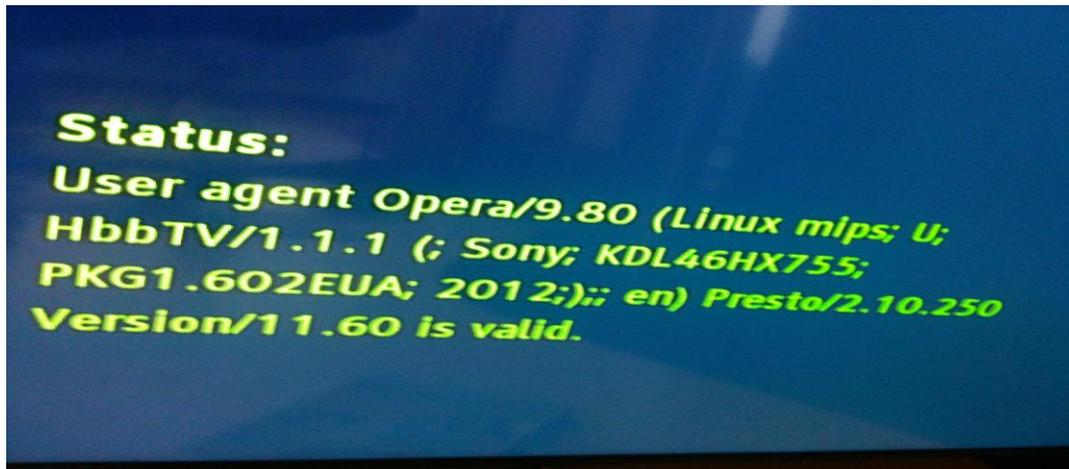


Figure 43: Sony Bravia HbbTV 1.1.1

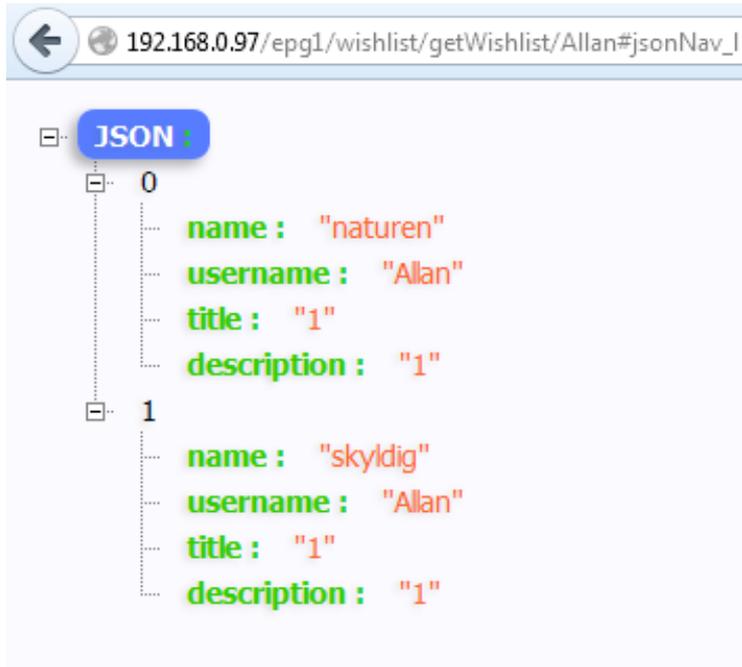
8.8.3 Tests carried from browser

Result of the test of recommendation for the user Allan based on Allan's preferences

A screenshot of a web browser displaying a JSON response from a recommendation service. The browser's address bar shows the URL "192.168.0.97/epg1/recommender/recommendPrograms/Allan#jsonNav_l". The JSON data is expanded to show the following fields:

```
JSON :
  0
  id : "-975867644"
  dvb_id : 19534
  channel : 23
  channel_info
  name : "NatGeo Wild HD"
  logo_image_prefix : "http://cloud.yousee.tv/static/img/logos/"
  logos
  orgtitle : "Caught In The Act: Strange And Savage"
  cast : ""
  directors : ""
  series_info : ""
  series_id : 107651
  category : 10
  subcategory : 65
  begin : 1357844400
  end : 1357848000
  tvdate : 1357772400
  formatted_date
  title : "Taget på fersk gerning"
  description : "Sammenstød i <span class='txtred'>naturen</span> går ikke altid efter planen. "
  archive : false
```

Test result for getWishlist for user Allan



```
JSON :  
0  
  name : "naturen"  
  username : "Allan"  
  title : "1"  
  description : "1"  
1  
  name : "skyldig"  
  username : "Allan"  
  title : "1"  
  description : "1"
```

Test result for getWishlist for user Louise



```
JSON :  
0  
  name : "Scott Calvin"  
  username : "Louise"  
  title : "0"  
  description : "0"
```

Test result of getCategories for Allan



```
192.168.0.96/epg1/categories/getCategorieswish/Allan#jsonNav_I
JSON :
0
  id : "1"
  name : "Film"
1
  id : "5"
  name : "Politik/Samfund"
2
  id : "9"
  name : "Sport"
3
  id : "10"
  name : "Fakta/Videnskab/Natur"
4
  id : "11"
  name : "Fritid/Hobby"
```

Test result for getChannels for Kasper



```
192.168.0.96/epg1/channels/getChannelswish/Kasper#jsonNav_I
JSON :
0
  id : "1"
  name : "DR1"
  smallogo : "http://cloud.yousee.tv/static/img/logos/small_dr1.png"
1
  id : "2"
  name : "DR2"
  smallogo : "http://cloud.yousee.tv/static/img/logos/small_dr2.png"
2
  id : "3"
  name : "TV 2"
  smallogo : "http://cloud.yousee.tv/static/img/logos/small_tv2.png"
3
  id : "38"
  name : "TV 2 News"
  smallogo : "http://cloud.yousee.tv/static/img/logos/small_tv2news.pr
4
```

Test for programFromNow

```
192.168.0.97/epgl/rest/programsfromnow/1#jsonNav_1
JSON
23
  id : "100248563"
  dvb_id : 54912
  channel : 1
  channel_info
    name : "DR1"
    logo_image_prefix : "http://cloud.yousee.tv/static/img/logos/"
    logos
      small : "small_dr1.png"
      large : "large_dr1.png"
      extralarge : "IK_DR1.png"
      super : "large3_dr1.png"
      mega : "mega_dr1.png"
    orgtitle : ""
    cast : ""
    directors : ""
    series_info : ""
    series_id : 130029
    category : 4
    subcategory : 36
    begin : 1357745100
    end : 1357745700
    tvdate : 1357686000
  title : "Skæg med bogstaver"
  description : "Ta´ med Hr. Skæg ud at lege, danse og synge i en verden af ord, magi og masser af skæg."
  archive : false
  startover : true
  imageprefix : "http://cache.ysimag.es/pics/100248564/"
  title : "Aftenshowet"
  description : "DR1s aktualitetsprogram direkte fra Islands Brygge i København."
  archive : true
  startover : true
  title : "TV AVISEN"
  description : ""
  archive : true
  startover : true
  imageprefix : "http://cache.ysimag.es/pics/100248575/"
  images_sixteenbynine
    large : "950x535.jpg"
    medium : "475x268.jpg"
    small : "208x117.jpg"
```

8.9 IRT Broadcast server configuration

Following are the configuration screen dumps of IRT broadcast server.

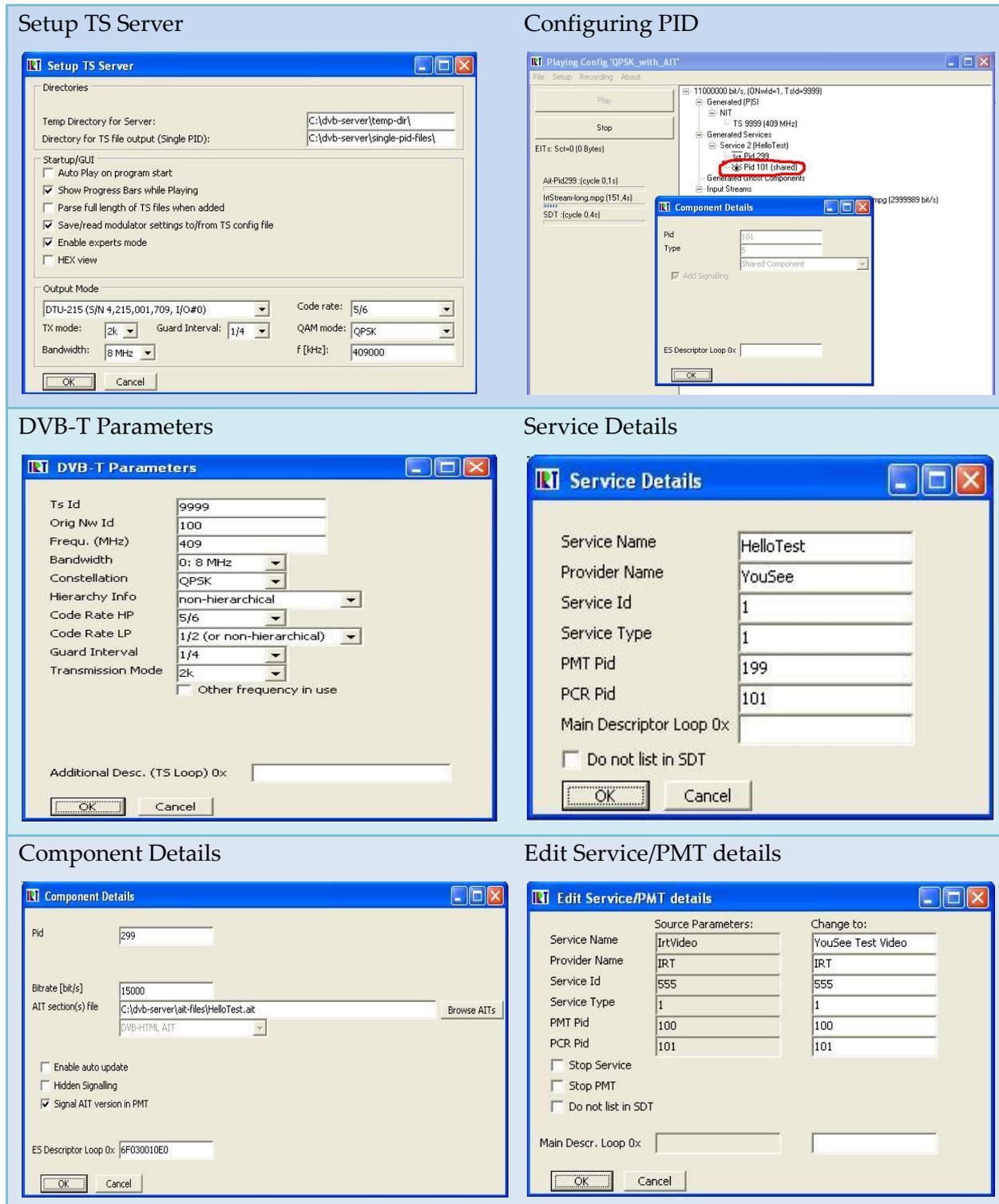


Figure 44: IRT broadcast server configuraion

8.10 AIT configuration

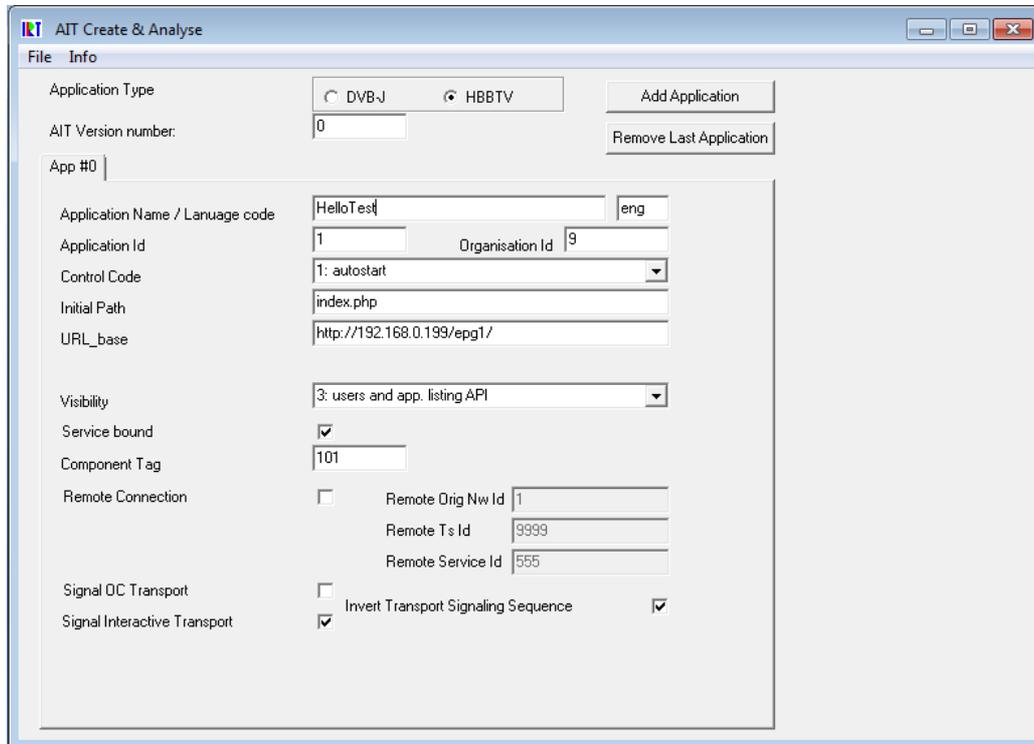


Figure 45: AIT configuration

AIT table

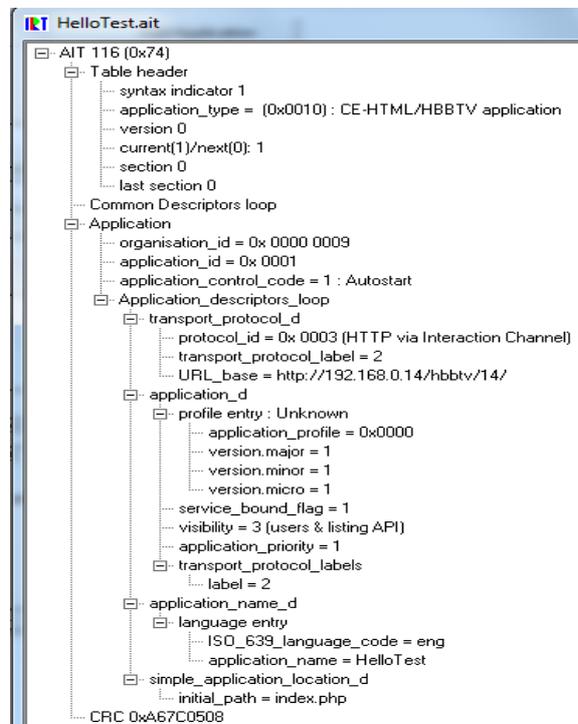


Figure 46: AIT table

9. Bibliography

- [1] IRT GmbH. (2012) Hbbtv consortium. [Online]. <http://www.hbbtv.org/>
- [2] Hongguang Zhang and Shibao Zheng, "Personalized TV program recommendation based on TV-anytime metadata," in *IEEE International Symposium on Consumer Electronics*, 2005. [Online]. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01502378&tag=1>
- [3] F.Portis, PP. Torasso, L. Ardisson. Dipartimento di Informatica. [Online]. <http://www.di.unito.it/~liliana/UM01/ardissono-et al.pdf>
- [4] Geert Van Kerckhove, Annelies Kaptein Pieter Bellekens. EuroITV 2009. [Online]. http://www.euroitv2009.org/docs/adjunct_proceedings/demos/BELLEKENS,%20PIETER%20et%20al.%20-%20demo.pdf
- [5] dtv. DTV.gov. [Online]. <http://www.dtv.gov/whatisdtv.html>
- [6] Fred Sherwin. Lindsay Sherwin. [Online]. http://www.lindsay-sherwin.co.uk/improvement_guide/html_tools/milestone_plans_gantt_charts.htm
- [7] Ian Sommerville, *Software Engineering*, 8th ed.: Addison-wesley, ch. 4, p. 131.
- [8] Wright P., Haber J. & Davenport L., Monk A., "Cooperative Evaluation," in *Improving your human-computer interface : A practical technique*. UK: Prentice Hall International, 1993.
- [9] EBU. (2011, September) EBU. [Online]. https://tech.ebu.ch/docs/events/ibc11-ebutechnical/presentations/ibc11_10things_hybrid-tv.pdf
- [10] Stephen Done. (1996, June) Imperial College London. [Online]. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/srd2/article2.html
- [11] etsi. (2004, November) ETSI. [Online]. http://www.etsi.org/deliver/etsi_es/202100_202199/202184/01.01.01_60/es_202184v010101p.pdf
- [12] impala. impala. [Online]. <http://www.impala.org/sites/default/files/MHEG%20Interaction%20Channel.pdf>
- [13] dvb project office. (2011, May) dvb. [Online]. http://www.dvb.org/technology/fact_sheets/DVB-MHP_Factsheet.pdf
- [14] Tilmann Rabl, Harald Kosch, Günther Hölbling, "Overview of Open Standards for Interactive TV (iTV)," in *Multimedia Semantics – The Role of Metadata.*, 2008, vol. 101, pp. 45-64. [Online]. http://link.springer.com/chapter/10.1007%2F978-3-540-77473-0_3?LI=true
- [15] DVB project office. (2012, august) Globally executable middleware. [Online].

- http://dvb.org/technology/fact_sheets/DVB-GEM_Factsheet.pdf
- [16] ETSI TS 102 796. (2010, June) ETSI. [Online].
http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.01.01_60/ts_102796_v010101p.pdf
- [17] Meghan Henning, Jeff Joseph. (2006, October) CEA. [Online].
http://www.ce.org/CorporateSite/media/Standards-Media/Standards%20Listings/CEA-2014_Press_Release_Detail.pdf
- [18] Paul Shrubsole, Walter Dees. (2007) International World Wide Web Conference Steering Committee. [Online].
<http://www.conference.org/www2007/posters/poster1017.pdf>
- [19] Open iptv forum. (2012, September) oipf. [Online].
<http://www.oipf.tv/specifications>
- [20] etsi. (2010, January) ETSI. [Online].
http://www.etsi.org/deliver/etsi_ts/102800_102899/102809/01.01.01_60/ts_102809_v010101p.pdf
- [21] etsi. (2010, January) ETSI. [Online].
http://www.etsi.org/deliver/etsi_ts/102800_102899/102809/01.01.01_60/ts_102809_v010101p.pdf
- [22] Joseph an Noronha, Sunil and Vergo, John Kramer, "A user-centered design approach to personalization," *Commun. ACM*, vol. 43, p. 5, aug 2000.
- [23] Daniel Schwabe, Robson Mattos Guimaraes Gustavo Rossi. (2001, May 2-5) Designing Personalized Web Applications. [Online].
<http://www10.org/cdrom/papers/395/>
- [24] Stephen Searby, "Personalisation - an overview of its use and potential," vol. 21, no. 1, p. 19, January 2003.
- [25] Lior Rokach & Bracha Shapira Francesco Ricci, *Chapter 1 Introduction to Recommender Systems Handbook*.: Springer Science+Business Media, 2011.
- [26] Mirco Speretta, Aravind Chandramouli and Alessandro Micarelli Susan Gauch. (2007) Warwick. [Online].
<http://www.dcs.warwick.ac.uk/~acristea/courses/CS411/2008/Book%20-%20The%20Adaptive%20Web/UserProfilesforPersonalizedInfoAccess.pdf>
- [27] Netflix Website. [Online]. <http://netflix.com>
- [28] Facebook. [Online]. <http://facebook.com>
- [29] ISO/IEC JTC1/SC29/WG11 N6828. (2004) The Moving Picture Experts Group. [Online].
http://mpeg.chiariglione.org/sites/default/files/files/standards/docs/w6828_mp7_Overview_v10.zip
- [30] TV-anytime forum. (2012) [Online]. <http://www.tv-anytime.org/>
- [31] Thomas Sikora and Atul Puri Shih-Fu Chang. (2001) [Online].

- http://www.img.lx.it.pt/~fp/cav/Additional_material/MPEG7_overview_1.pdf
- [32] ISO/IEC 15938. the moving picture experts group website. [Online].
<http://mpeg.chiariglione.org/standards/mpeg-7>
- [33] (2012) the moving picture expert group website. [Online].
<http://mpeg.chiariglione.org/standards/mpeg-7/multimedia-description-schemes>
- [34] ebu, etsi. (2007, November) ETSI. [Online].
http://www.etsi.org/deliver/etsi_ts/102800_102899/10282202/01.04.01_60/ts_10282202v010401p.pdf
- [35] TV-Anytime forum. (2002, December) TV-Anytime Forum. [Online]. www.tv-anytime.org
- [36] Network Working Group. EITF TOOLS. [Online].
<http://tools.ietf.org/html/rfc4078>
- [37] Ronald Tol. (2004, January) TV-Anytime Forum. [Online]. www.tv-anytime.org/ftp/San-Jose_IDE_Sept04/Phase_1_Overview.ppt
- [38] Ron Zacharski. (2012) guidetodatamining. [Online].
<http://guidetodatamining.com/guide/ch1/DataMining-ch1.pdf>
- [39] P. Brusilovsky, "Methods and techniques of adaptive hypermedia," *User modeling and user-adapted interaction*, vol. 6, no. 2, pp. 87--129, 1996.
- [40] Lior Rokach, Bracha Shapira, Francesco Ricci, "Recommender Systems Function," in *Recommender Systems Handbook*, Paul B. Kantor, Ed.: Springer, 2011, ch. 1.
- [41] Herlocker et al., "User Tasks for Recommender Systems," *Evaluating collaborative filtering recommender systems*, vol. 22, no. 1, p. 53, January 2004.
- [42] Michael J. Pazzani and Daniel Billsus, "Content-based recommendation systems," in *THE ADAPTIVE WEB: METHODS AND STRATEGIES OF WEB PERSONALIZATION. VOLUME 4321 OF LECTURE NOTES IN COMPUTER SCIENCE.*: Springer-Verlag, 2007, pp. 325--341.
- [43] Prabhakar Raghavan, and Hinrich Schütze, Christopher D. Manning. (2008) The Stanford Natural Language Processing Group. [Online].
<http://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html>
- [44] Puffinware. (2010, January) Puffinware iMetaSearch. [Online].
<http://www.puffinwarellc.com/index.php/news-and-articles/articles.html>
- [45] Badrul and Karypis, George and Konstan, Joseph and Riedl, John Sarwar. (2001) The Ohio state university, Department of statistics. [Online].
http://www.stat.osu.edu/~dmsl/Sarwar_2001.pdf
- [46] (2006, July) University of the West of England. [Online].
<http://hsc.uwe.ac.uk/dataanalysis/quantinfasspear.asp>

- [47] Statistics How To. [Online]. <http://www.statisticshowto.com/articles/what-is-the-pearson-correlation-coefficient/>
- [48] Rogers Y. and Sharp H., Preece J., "Design, Prototyping and Construction," in *Interaction Design: Beyond human-computer interaction.*: John Willy & Sons Inc., 2011, ch. 11, pp. 389-432.
- [49] Jakob Nielsen. (2010, june) Jakob Nielsen's Website. [Online]. <http://www.useit.com/alertbox/response-times.html>
- [50] Institut für Rundfunktechnik. (2012, May) Institut für Rundfunktechnik. [Online]. <http://www.irt.de/webarchiv/showdoc.php?z=NDcxOSMxMDA0MjEzMTAjcGRm>
- [51] Dektec. (2011, September) Dektec. [Online]. www.dektec.com/Products/USB2/DTU-215/Downloads/DTU-215%20Leaflet.pdf
- [52] ISSN 1548-5552 Jakob Nielsen. (2010) useit.com: Jakob Nielsen's Website. [Online]. <http://www.useit.com/alertbox/response-times.html>
- [53] J. de Rosnay. (1997, Feb 17) Principia Cybernetica Web. [Online]. <http://pespmc1.vub.ac.be/ANALYSYST.html>