NoTube
Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet

FP7 – 231761

D7b.1 Personalized Program Guide Specifications and Design with Personal Advertising

Coordinator: Roland Schijvenaars (SIT)
With contributions from: Annelies Kaptein, Erik van Dijk, Geert van Kerckhove (SIT), Eylem Kehribar (Engin), Jaesuk Ahn, Eunjeong Choi, Seongsoo Park (KT), Anne-Lore Mevel, Raoul Monnier (TGV)
Quality Assessor: Gerhard Stoll (IRT)
Quality Controller: Lyndon J B Nixon (STI)
EXECUTIVE SUMMARY

This document is the first deliverable for work package 7b in the NoTube project. It describes in detail the Use Case 7b which is focused on the TV domain. One of the projects key results is this showcase for a personalized TV guide service with personal advertising. Furthermore, the focus in WP7b is on multi device, multilingual and multimodal support. This document is mainly a report on the specification and design of a personalized TV program guide which are based on use case scenarios. The development of a showcase for a personalized program guide is described and evaluated.
## Deliverable 7b.1

### Document Information

<table>
<thead>
<tr>
<th>IST Project Number</th>
<th>Acronym</th>
<th>Full Title</th>
<th>Project URL</th>
<th>Document URL</th>
<th>EU Project Officer</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP7 – 231761</td>
<td>NoTube</td>
<td>Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet</td>
<td><a href="http://www.notube.tv/">http://www.notube.tv/</a></td>
<td></td>
<td>Leonhard Maqua</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deliverable Number</th>
<th>Title</th>
<th>Work Package Number</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>7b.1</td>
<td></td>
<td>7</td>
<td>TV-related Use Cases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date of Delivery</th>
<th>Contractual</th>
<th>Actual</th>
<th>Status</th>
<th>Nature</th>
<th>Dissemination Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M13</td>
<td>10-02-10</td>
<td>version 1.0</td>
<td>prototype ☐ report ☐ dissemination ☐</td>
<td>public ☐ consortium ☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authors (Partner)</th>
<th>Resp. Author</th>
<th>E-mail</th>
<th>Partner</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annelies Kaptein, Erik van Dijk, Geert van Kerckhove(SIT)</td>
<td>Roland Schijvenaars</td>
<td><a href="mailto:roland.schijvenaars@stoneroos.nl">roland.schijvenaars@stoneroos.nl</a></td>
<td>SIT</td>
<td>+31 (0) 35 628 47 22</td>
</tr>
</tbody>
</table>

### Abstract (for dissemination)

Stoneroos, iFanzy, personalized electronic program guide, personal advertisements

### Keywords

Stoneroos, iFanzy, personalized electronic program guide, personal advertisements

### Version Log

<table>
<thead>
<tr>
<th>Issue Date</th>
<th>Rev No.</th>
<th>Author Name</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 23, 2010</td>
<td>1</td>
<td></td>
<td>The name of the revision</td>
</tr>
</tbody>
</table>

### Important information for Quality Assessors/Quality Controllers (PLEASE DELETE IN THE EC-READY VERSION)

- Send a PDF version AND a Latex/MS-Word of the deliverable to the Quality Assessor/Quality Controller to ensure that no links are broken, pictures are correctly converted, etc.,
- ALL pictures in the deliverable must be B&W readable,
- After addressing the Quality Assessor’s comments, report back to him/her using the review form.
# Project Consortium Information

<table>
<thead>
<tr>
<th>Participant’s name</th>
<th>Partner</th>
<th>Contact</th>
</tr>
</thead>
</table>
| Vrije Universiteit Amsterdam | [VU logo] | Guus Schreiber  
Phone: +31 20 598 7739/7718  
Email: schreiber@cs.vu.nl |
| British Broadcasting Corporation | [BBC logo] | Libby Miller  
Phone: +44 787 65 65 561  
Email: Libby.Miller@bbc.co.uk |
| Asemantics SRL Uninominale | [ASEMANTICS logo] | Alberto Reggiori  
Phone: +39 0639 7510 78  
Email: alberto@asemantics.com |
| Engin Medya Hizmetleri A.S. | [Engin logo] | Ron van der Heiden  
Phone: +31 6 2003 2006  
Email: ron@engin.tv |
| Institut fuer Rundfunktechnik GmbH | [IRT logo] | Christoph Dosch  
Phone: +49 89 32399 349  
Email: dosch@irt.de |
| Ontotext AD | [Ontotext logo] | Atanas Kiryakov  
Phone: +35 928 091 565  
Email: naso@sirma.bg |
| Open University | [Open University logo] | John Domingue  
Phone: +44 1908 655 014  
Email: j.b.domingue@open.ac.uk |
| RAI Radiotelevisione Italiana SPA | [RAI logo] | Alberto Morello  
Phone: +39 011 810 31 07  
Email: a.morello@rai.it |
| Semantic Technology Institute International | [STI logo] | Lyndon Nixon  
Phone: +43 1 23 64 002  
Email: lyndon.nixon@sti2.org |
| Stoneroos B.V. | [Stoneroos logo] | Annelies Kaptein  
Phone: +31 35 628 47 22  
Email: annelies.kaptein@stoneroos |
| Thomson Grass Valley France SA | [Thomson logo] | Raoul Monnier  
Phone: +33 2 99 27 30 57  
Email: raoul.monnier@thomson.net |
| TXT Polymedia SPA | [TXT logo] | Sergio Gusmeroli  
Phone: +39 02 2577 1310  
Email: sergio.gusmeroli@txtgroup.com |
| KT Corporation | [KT logo] | Myoung-Wan Koo  
Phone: +82 2 526 6347  
Email: mskim@kt.co.kr |
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION
  1.1 Document structure .................................................. 10
  1.2 Electronic program guides ......................................... 10
  1.3 iFanzy - personalized EPG ......................................... 11
  1.4 Objectives .............................................................. 13

2 USE CASE: THE SCENARIOS
  2.1 Scenario 1: Combining PEPG, Internet an ads ..................... 14
  2.2 Scenario 2: EPG in social context .................................. 14
  2.3 Scenario 3: Adaptive ad in video .................................. 15
  2.4 Overview .............................................................. 15
  2.4.1 Goals ............................................................... 16

3 REQUIREMENTS
  3.1 Functional requirements .............................................. 18
    3.1.1 Multi device support ........................................... 18
    3.1.2 Multilingual support and localization ......................... 20
    3.1.3 Multimodal support ............................................. 23
    3.1.4 EPG .............................................................. 24
    3.1.5 User profiling and TV program recommendations ............ 24
    3.1.6 Personalized advertising ....................................... 28
  3.2 Technical requirements ............................................... 35
    3.2.1 Service provider side .......................................... 35
    3.2.2 Home ambient side ............................................. 35
    3.2.3 Security and privacy issues ................................... 36
  3.3 Development plan ..................................................... 36

4 FIRST PROTOTYPE
  4.1 Architecture .......................................................... 38
  4.2 Story ................................................................. 38
  4.3 Swimlanes ............................................................. 39
  4.4 Implementation ....................................................... 42

5 EVALUATION
  5.1 Recommendations and similar-programs testing ................... 44

6 CURRENT STATUS AND FUTURE WORK

REFERENCES
## List of Figures

1.1 Screenshot of the iFanzy Web application, providing an EPG for TV programs .............................................. 11  
1.2 Global overview of the client-server architecture of iFanzy ................................................................. 12  

3.1 Web-based GUI - PEPG ........................................ 18  
3.2 Web-based GUI - Profile ........................................ 19  
3.3 TV-centric GUI(I) - PEPG ....................................... 19  
3.4 TV-centric GUI(I) - Log in ....................................... 20  
3.5 TV-centric GUI(I) - User profile coupling .................................................. 21  
3.6 TV-centric GUI (II) - PEPG ..................................... 21  
3.7 TV-centric GUI (II) - PEPG with TV program details .................................................. 21  
3.8 iPhone GUI - Recommendations ........................................ 22  
3.9 iPhone GUI - PEPG ........................................ 22  
3.10 iFanzy Web application in Turkish and with Turkish EPG data .................................................. 23  
3.11 an iPhone application providing multimodal support, developed by KT .................................................. 24  
3.12 Heat map for ad placement on Web pages .................................................. 30  
3.13 Ad insertion flow ........................................ 31  
3.14 Ad insertion job ........................................ 32  
3.15 Ad insertion workflow ........................................ 32  
3.16 Ad insertion get list ........................................ 33  
3.17 Ad insertion job create ........................................ 34  
3.18 Ad insertion job modify ........................................ 34  

4.1 NoTube WP7b architecture ........................................ 38  
4.2 Scene 1 ........................................ 39  
4.3 Scene 2 ........................................ 39  
4.4 Scene 3 ........................................ 40  
4.5 Swimlane - Scene 1 ........................................ 40  
4.6 Swimlane - Scene 2 ........................................ 41  
4.7 Swimlane - Scene 3 ........................................ 41  
4.8 Semantic Broker-iFanzy Adapter ........................................ 42  
4.9 Component and deployment diagram ........................................ 43  

5.1 Test for determine accuracy of recommendation algorithm .................................................. 45  
5.2 Test for determine accuracy of recommendation and similar-programs algorithm .................................................. 45
LIST OF TABLES

2  List of abbreviations .................................................. 9
3.1 EPG specification .................................................. 25
3.2 User profile requirements ........................................... 27
3.3 User interaction requirements ...................................... 27
3.4 Contextual requirements ............................................. 28
3.5 Ad specification .................................................... 29
3.6 Ad insertion - Workflow get list .................................. 33
3.7 Ad insertion - Job create ........................................... 33
3.8 Ad insertion - Job modify ........................................... 33
3.9 Development plan ................................................... 37
<table>
<thead>
<tr>
<th>Abbr</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad</td>
<td>advertisement</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>DTV</td>
<td>Digital TV</td>
</tr>
<tr>
<td>EPG</td>
<td>Electronic Program Guide</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical UI</td>
</tr>
<tr>
<td>HD</td>
<td>High-Definitions</td>
</tr>
<tr>
<td>IMDb</td>
<td>Internet Movie Database</td>
</tr>
<tr>
<td>IRS</td>
<td>Internet Reasoning Service</td>
</tr>
<tr>
<td>PEPG</td>
<td>Personalised EPG</td>
</tr>
<tr>
<td>PVR</td>
<td>Personal Video Recorder</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>STB</td>
<td>Set-top box</td>
</tr>
<tr>
<td>TV</td>
<td>Television</td>
</tr>
<tr>
<td>UGC</td>
<td>User Generated Content</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>VOD</td>
<td>Video On Demand</td>
</tr>
<tr>
<td>WP</td>
<td>Work package</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
</tbody>
</table>
1. Introduction

In this chapter, the structure of the document is described and an introduction in electronic program guides and the iFanzy\(^1\), which is a personalized EPG, is given. At the end of this chapter, an overview of the objectives is given.

1.1 Document structure

This document starts with an introduction in personalized electronic program guides and an overview of the objectives for the work package responsible for this document. In chapter 2 the use case, which forms the basis behind the work to be done, and corresponding goals are described. Next, in chapter 3, requirements and a development plan, which are derived from the use case, are stated. Chapter 4 describes the development of the first prototype used as demonstrator for the personalized program guide showcase. The prototype is evaluated in chapter 5 and finally the current status and future work is described in chapter 6.

1.2 Electronic program guides

Like the growth of the amount of information available on the WWW, the amount of television (TV) content available to people has risen during the last years. Hundreds of TV programs are broadcasted at various channels at any time of day. The digital revolution is partly responsible for the expanding amount of TV content. Until the 2000s broadcasted TV programs were generally recorded and transmitted as an analogue signal, but in recent years public and commercial broadcasters have been progressively introducing digital television (DTV) and even high-definition television (HDTV) broadcasting technologies. An advantage of DTV above the traditional analogue television is that channels do take up less bandwidth. This means that digital broadcasters can provide more channels in the same space, better quality services like HDTV or additional services such as electronic program guides (EPGs). An EPG is an overview of scheduled broadcast television or radio programs and is used to see what programs are currently broadcasted or will be broadcasted in the (near) future. The EPG is the electronic equivalent of the printed television program guide. Typically an EPG is an application running on the TV or on a set-top box (STB), which is a device that is able to receive and decode DTV signals and turn them into a format which is viewable via a TV. EPGs often are accessed via a remote control and offer interactive functionalities such as setting a recording (on a PVR) or a reminder for a TV program. Figure 1.1\(^2\) shows a screenshot of an EPG. In general an EPG presents two types of an overview about broadcasted TV programs. One type of overview shows the TV programs first ordered by time (vertically) and next by channel (horizontally). The other type of overview shows programs first ordered by channel (vertically) and next by time (horizontally). As mentioned before, the number of channels and thus the number of TV programs available to the users of an EPG can be quite large. To find TV programs that suit the individual interest of a user can be an exhaustive task. Therefore, users of EPGs end up watching a limited number of favourite channels while

\(^1\)Octrooinummer NL 2002231, Patent pending; PCT no. 2009/050251
\(^2\)http://www.ifanzy.nl
ignoring the other channels [7]. Probably these users miss interesting TV programs that are broadcasted on one of those other channels. In other words, EPGs today typically lack on personalization. More and more personalized EPGs are developed to suggest individual (or groups of) users which TV programs should be interesting to watch.

Figure 1.1: Screenshot of the iFanzy Web application, providing an EPG for TV programs

1.3 iFanzy - personalized EPG

iFanzy is a personalized EPG developed by Stoneroos Interactive Television in collaboration with Eindhoven University of Technology (TU/e). Its main functionalities are to provide recommendations over and intelligent search through a large set of broadcasted TV programs. While using iFanzy, users are able to sit back and relax watching TV programs of their interest, instead of zapping through all TV channels to find something interesting to watch. This is because iFanzy suggest TV programs to its users that are potentially interesting for them to watch. In order to come up with suggestions, the recommendations iFanzy provides are based on a content-based filtering approach. To cope with the cold start problem, demographic filtering techniques are applied. iFanzy consists of a client-server system with multiple clients operating at various devices as shown in Figure 1.2.

At the start of the NoTube project an iFanzy Web application client was available in dutch (beta version online, see http:\www.ifanzy.com). In the NoTube project the focus will be on multilingual and multi device support for the iFanzy clients and the use of services developed in the NoTube project. With the availability of such clients, a user is able to use iFanzy almost everywhere, at any time of day and in different environments. The clients offer different functionalities tailored to the platform and device it is developed for. However, in the background, at the server side, they are closely related to each other. Every action performed at one platform has a direct influence on the other. For example, a rating given to a TV program at the Web
application will influence the generated recommendations at the STB immediately. User behaviour differs at each of these platforms. While using a Web browser, a user is more active and willing to give explicit feedback than while watching TV at home for instance.

The iFanzy Web application is one of the iFanzy clients in development. It is used for browsing and searching TV content. The screenshot in Figure 1.1 shows a part of the graphical user interface (GUI) of the iFanzy Web application, the ‘HOME’ page of the Web application. At the bottom left, a small personalized EPG with only three TV channels is shown. The details of one of the TV programs in this EPG are shown together with a rating bar.

For a more elaborate overview of the TV guide, the interface also contains a large listing which can be found under the tab ‘TV-GIDS’ showing all the TV channels in the system. These channels can be ordered by the preference of the user. Next to the personalized EPG, on the right, the home page shows a search form to look for specific TV content accompanied by a tag cloud showing the most popular terms searched for in the past.

Besides the opportunity for users to experience TV via various devices, also the possibilities for multilingual support will be explored within the NoTube project.

The marks and colours in the TV guide are based on the recommendations provided by the iFanzy system and are represented by marks and colours. The higher the mark, the more the TV program is supposed to fit the interests of the user. Also the yellow/orange-like colours are to express to what extend TV programs match the interests of the user according to the iFanzy system. The more orange the colour is, the more the TV program suits the user’s interests. A user is able to express his interest in a TV program by using the rating bar at the right of the title of a program in the program details information box (blue). Ratings given to a certain TV program by a user are stored in the user profile. This information is used to generate recommendations for (other) TV programs. The better iFanzy knows the interests of a user, the better it can serve recommendations to him. In addition a good notion by using metadata of the items of interests, i.e. TV programs, can help iFanzy to
generate better recommendations.

1.4 Objectives

The NoTube project has decided from the very beginning to develop and implement typical use cases, which embody the objectives of the project. The work is structured and divided in several work packages. Work package 7 had to develop and implement three uses cases: a, b and c. This chapter is describes the target set for one of these use cases, namely use case b.

The main objectives of the NoTube Project are according to the NoTube Annex I [1]:

“The core objectives of this project are targeted at putting the TV viewer back in the driver’s seat by developing semantics-based and context aware tools and services for (1) personalized content selection, (2) packaging of content with additional information, and (3) social interaction in/consumer communities”.

To narrow down to objectives of the work package, NoTube Annex I [1]:

“The overall goal of WP7 is to produce the use cases which showcase the NoTube technology. This covers:

- Analyzing and defining the telecom application domain scenarios, providing input points where NoTube will develop its business processes;
- Implementing and deploying the use case scenarios over real time devices and systems, integrating them with the NoTube infrastructure;
- Testing and validating the semantic technologies developed in WP1-5 and integrated into the NoTube architecture (WP6) in a real life TV scenario, assessing the project results over real applications, obtaining feedback which serves as input on the next iteration of the project development.”

This document is the first deliverable for work package 7b. According to the NoTube Annex, this document is a [1]:

“Report on the specification and design of Personal Program Guide with personalized advertisement delivery.”

One of the NoTube project key results is the realisation of a showcase for a personalized TV guide service for which work package 7b is responsible. WP7b focuses on different control interfaces to offer a new program guide experience to the consumers; End-to-End personalized Advertising: creation, manipulation, delivery and consumption of personalized advertising messages, also allowing user-generated content and consumer participation to the content value chain taking context into account.
2. Use case: The scenarios

As written above, the NoTube project is very much use case driven. The main objective of the use case described in this deliverable is to illustrate the design and development of a personalized EPG recommending TV programs to the viewer and proposing him/her additional content and services, including advertising material. The use case focuses on different control interfaces, i.e. user interfaces, to offer a new television program guide experience to the consumers. This use case includes personalized advertising, allows for user-generated content and consumer participation to the content value chain taking context into account. Topics are: EPG and Ad metadata, user identification/distribution of user profiles, recommendations, user events. Given the objective of this use case, three specific scenario’s were stated.

2.1 Scenario 1: Combining PEPG, Internet and ads

Right after work, Sena drives to Schiphol and takes the airplane to Istanbul to join Tarik for the weekend. While she is waiting at the airport, Sena takes her laptop and connects to the local Wifi. She goes to the iFanzy.nl website to update her profile and set a few recordings for programs on television that weekend. She looks what movies are recommended to her that weekend and decides to record a movie that is strongly recommended and has a high IMDb rating. Before she can watch the video previews, she is shown a pre-roll ad for a book or CD that might interest her. iFanzy knows that she regularly travels on Friday and presents products that are relevant in that context and time. It might be a newly released CD in the Netherlands of a Turkish singer-songwriter from Germany. She can buy the CD through the iStore to listen to it during her flight. Whilst browsing through the EPG she recognizes the comedy program Little Britain because her brother Nadir had sent her a compilation of funny fragments that he made. She thinks Nadir made a nice compilation and decides to add it to the iFanzy website as a reference for others.

2.2 Scenario 2: EPG in social context

Saturday afternoon Sena meets up with her German friend Claudia that works in Istanbul as an expat to go shopping. After shopping they go to Claudias place to have a drink. After a while Claudia gets an iPhone reminder for a new English soap series that is broadcasted on Turkish television. She wants to show it to Sena so they turn on the television and log in with their combined profile to check if Sena would like it. Since Claudia is German and Sena Dutch the PEPG and ad for a home delivery service, are in English. iFanzy knows the two women are logged in and dinner time is not far away so it could be convenient that they can have a TV evening without having to go outdoors. But Sena already has dinner plans so she decides not to order any food but start watching TV right away. Sena really liked the soap, so after watching it she picks up her iPhone and ads the program to her favourite programs, and she sets her iFanzy to record the future broadcasts automatically.

Sunday morning Sena visits her Korean friend Jung in the center of Istanbul. Jung turns on the TV and logs on to the Korean iFanzy EPG that lists the Korean programs
that fit Jungs profile. Since Jung previously added “Jewel In The Palace”, a Korean historical drama on the Royal Court cuisine and traditional medicine, to her favourites, the listing also contains other Korean historical dramas; like “Jumong” and “Yi San”. Jung is so fond of “Jewel In The Palace” that she wants to watch her recording again, together with Sena. Before they can see the movie, iFanzy rolls a commercial for traditional Korean cooking ingredients. This commercial is recommended by the system because it knows the user is watching a drama about Korean traditional food. After watching “Jewel In The Palace” Sena accesses the iFanzy website (English version) on her iPhone, and adds a related link to another resource to the TV program she just watched with her friend. Sena decides to share the TV program she just watched with her friends. Her friends can add the TV program to their favourites, watch it right away if it is available in the VOD database or set a reminder for when it is broadcasted by their provider.

2.3 Scenario 3: Adaptive ad in video

Sena arrives in Istanbul around 7 PM and goes home by public transport to take a shower and gets ready to have dinner with Tarik in the city. After dinner they decide to have a relaxed evening and they both go home to watch a movie together. They start the iFanzy PEPG on their television and log in with their combined profile. Their available recordings do not suit them at the moment so they launch the broadcasters VOD portal. In the VOD portal they watch movies that are either in Turkish or have Turkish subtitles available because of Tariks background, and that are recommended for them as a couple. In the meantime they are presented with an ad promoting a banks low mortgage rate in the VOD portal. The ad system knows that they are watching together regularly as a couple and that they have both the right age and profession, which means enough income, to be able to buy a house with a mortgage. They decide to watch an American romantic thriller with Turkish subtitles. Before the movie starts they are invited in a pop-up to watch a few commercials in return for VOD credits. They choose to play the movie with commercials to get the discount. While they are watching the movie, a commercial that corresponds to their profile is inserted as an overlay to the TV program. An ad in the video stream for gold jewelery and one for a city trip for two as the setting is even more romantic during the movie.

2.4 Overview

Within NoTube the focus of WP7b is on a connected multi device environment for TV content consumption. Globally these devices can be categorized into the following physical components:

- Desktop;
- Mobile device;
- Set-top box and television.

The complementary functionalities normally available for the devices mentioned above, can be grouped into the following categories:
• EPG and Ad metadata, for showing an EPG and ads;
• Recommendations, for personalization (e.g. PEPG, personalized ads);
• User identification and distribution of user profiles over various devices;
• User (inter)actions;
• Context, the condition in which something exists.

The use, look and feel of application on these devices differ per device. The different devices are often used in a different context. Therefore different GUIs will be developed for each of them. GUIs to develop:

• Web based user interface;
• Mobile user interface;
• TV-centric user interface.

2.4.1 Goals

The following goals are abstracted out of the scenarios described in Chapter 2:

• Ad placement in a (P)EPG;
• Ad placement before a video;
• Ad placement in a video;
• Support for multiple languages (for interfaces and metadata);
• Metadata enrichment (for EPG/TV programs and ads);
• Similar program’s;
• Select favourite TV programs/make personal TV program planner;
• Automatic record function for TV programs;
• Reminder function for TV programs;
• Multi device support;
• Distribute profile over various different devices;
• Friend notification functionality (e-mail or notification to user via an application like twitter);
• User generated content (UGC) related to a TV program (e.g. user adds a tag or a link to another resource related to the TV program).
Service goals

The goals abstracted from the scenarios can be linked to services that are or will be developed in the NoTube project. Next a more detailed overview of goals per service is given:

User profiling service (WP3)

- Create a new profile for a user. Date of birth, gender, level of education, preferred language(s) for TV programs (spoken/subtitled) and clients, preferred services/channels, etc.;

- Update a profile of a user (see 'create a new profile for a user' for more details);

- Users can login to load their (combined) profile;

Recommendation service (WP3)

- Get metadata of TV programs with level of recommendation for each TV program on the basis of the profiles, history/log of (inter)activity and current context of the user (see ‘get metadata of TV programs’ for more details);

- Get a link to an advertisement on the basis of the user profile, context (e.g. time, location, platform) and activity (e.g. traveling) of the current users and the displayed content;

- Get similar TV programs given a particular TV program (based on the content/metadata of the program).

Activity handling service (an action occurs in a certain context) (WP3)

- Set a recording for a TV program by a user. A user can set an option to record future broadcasts of a certain TV program;

- Get list of recordings (recorded/set recordings) of a user;

- Set a reminder for a TV program for a user;

- Get list of reminders of a user;

- A user can watch a TV program;

- Add a TV program to the user’s favourite TV programs;

- Get a list of TV programs in the favourite TV programs of a user;

- A user can rate a TV program.

Enrichment service (WP4)

- Get/enrich metadata of TV programs (e.g. title, starttime, endtime, service, genre, description, trailer, language (spoken/subtitled), type, resource, IMDb-rating);

- Get/enrich metadata of services/channels (displayname, longname, description, country of origin, default language);

- Get/enrich user profiles (e.g. user interests, user activities).
3. Requirements

Out of the use case and the abstracted goals, requirements are specified. The requirements are split up into functional and technical requirements. The functional requirements hold requirements for building up a PEPG, designs of graphical user interfaces for the various devices this use case focus on, and personalized advertising. Requirements for specifications to be developed in the project and descriptions of technologies to be used are given. The technical requirements are defined for the service provider side and the home ambient side. For more information see “D6.1 NoTube System Specifications and Architectural Design” [2]. Also security and privacy has to be taken into account in the project.

3.1 Functional requirements

3.1.1 Multi device support

Three different GUI’s will be developed to support convenient TV experience by end users. The designs for the GUI’s are shown and described in this section. Furthermore research is and needs to be done about how user can login at, or make connections between, each separate device.

A user profile should be accessible and/or distributed to clients on different devices. More about the user profile can be found in section 3.1.5

Web-based graphical user interface

In Figure 3.1, a screenshot of a Web-based GUI is shown. In this screenshot an example of a PEPG is shown. At the left side a number of channels are visible. At the right of a channel, the TV programs that are broadcasted by this channel are displayed ordered by time. Above the PEPG a navigation bar is given to browse through time. The various colours in the PEPG represent the level of interest (recommendation score) for the particular TV program for the user which is signed in.

![Figure 3.1: Web-based GUI - PEPG](image)
Figure 3.2 shows a screenshot of the Web-based GUI of the existing iFanzy web application. This is the section where users are able to sign in, register or couple a Web account to a former build up set-top box account.

![Figure 3.2: Web-based GUI - Profile](image)

**TV-based graphical user interface**

Figure 3.3 shows a PEPG. At the bottom of the figure, the guide shows three channels with broadcasted TV programs at the right side. Above the guide, three recommended TV programs are shown.

![Figure 3.3: TV-centric GUI(I) - PEPG](image)

Figure 3.4 shows another design of a set-top box application for personalized television. These design shows a log in screen. Whenever a user logs in, his personal
profile will be loaded and a PEPG will be shown.

Figure 3.4: TV-centric GUI(I) - Log in

Figure 3.5 shows an early design about how users can register for a user profile on a set-top box. It turns out to be that users are not likely to enter alphabetic characters on a remote control because it costs a lot of effort and time. At the moment of writing, research is done about distributing and coupling user profile from application running at various devices. The focus here is on user friendliness however, security and privacy issues must be kept in mind.

Figure 3.6 shows a design for a PEPG at a set-top box. On the left side the PEPG is shown. The TV programs are ordered by time. At the right of each TV program a number is given. This number represents the recommendation score.

Figure 3.7 shows the same set-top box PEPG. However more details are given for a particular TV program. This action should be initiated by the user.

Mobile-based graphical user interface

Figure 3.8 and Figure 3.9 show a mobile-based graphical user interface for the Apple iPhone. These designs show recommendations for TV programs ordered by time. At the right of each TV program the recommendation score is shown.

3.1.2 Multilingual support and localization

To be able to support multiple languages, the textfields presented in the user interface need to be translated into the supported languages. Also the TV metadata should be
Figure 3.5: TV-centric GUI(I) - User profile coupling

Figure 3.6: TV-centric GUI (II) - PEPG

Figure 3.7: TV-centric GUI (II) - PEPG with TV program details
Figure 3.8: iPhone GUI - Recommendations

Figure 3.9: iPhone GUI - PEPG
displayed in the correct language. The broadcasted TV content, and thus the set of TV programs in an EPG, is often related to specific countries or regions. Broadcasters usually do not broadcast worldwide, they broadcast in one country or sometimes in a few countries. Therefore also the EPG data feeds are rather country specific. Examples of data feeds from various countries are supplied by partners in WP7. An overview is given in Appendix A. Figure 3.10 shows as an example the iFanzy Web application with a Turkish user interface and with Turkish EPG data.

![Figure 3.10: iFanzy Web application in Turkish and with Turkish EPG data](image)

### 3.1.3 Multimodal support

Multimodal interaction provides a user with multiple modes of interfacing with a system. For example, combining a visual modality (e.g. display and keyboard) with voice modality (e.g. speech for input). An example of an interface providing touch and voice modality into one interface is shown in Figure 3.11. While touching the 'Hold & Talk' button on the display of the iPhone, the user can give commands by voice. The application turns speech into text and the user can then confirm or cancel the command.

Some examples of input modalities are:

- Keyboard;
- Mouse;
- Pen;
- Remote control;
- Speech;
- Touch;
- Gaze;
- Body movements (e.g. head movements, manual gestures).
The advantage of multiple input modalities is the increased usability. On a mobile device with a small visual interface and keypad or on a TV controlled by a remote control, a word may be quite difficult and time consuming to type but very easy to say.

### 3.1.4 EPG

To be able to build up an EPG, a specification is defined for TV programs in the EPG. The specification is a ‘contract’ between the clients, that show the EPG, and the services providers, that deliver the EPG data. This specification can be divided into two subcategories:

- **TV program**: this category includes elements to describe the TV program;
- **EPG Entry**: this category includes elements to describe the entry of a program in the guide.

Table 3.1 gives an overview of requirements for the EPG specification according to WP7b.

The TV program specification can also be used to describe VOD content. In collaboration with WP4 descriptive TV program information can/will be expanded and/or supplemented. Furthermore this specification is used as input for WP2.

### 3.1.5 User profiling and TV program recommendations

To be able to personalize EPG’s, to give users recommendations for TV programs to watch, two important aspects come into play:

1. The knowledge of the system about the **TV programs**;
2. The knowledge of the system about the **users**.
Table 3.1: EPG specification

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TV program</strong></td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>The identifier of the TV program</td>
</tr>
<tr>
<td>Title</td>
<td>The title of the TV program</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the TV program</td>
</tr>
<tr>
<td>Genre</td>
<td>One or multiple genres of the TV program</td>
</tr>
<tr>
<td>Credit</td>
<td>One or more persons/companies related to the TV program (e.g. actors, producers)</td>
</tr>
<tr>
<td>Resource</td>
<td>One or more resources related to the TV program (e.g. trailers, links to related websites)</td>
</tr>
<tr>
<td><strong>EPG entry</strong></td>
<td></td>
</tr>
<tr>
<td>Start time</td>
<td>The start time of the TV program</td>
</tr>
<tr>
<td>End time</td>
<td>The end time of the TV program</td>
</tr>
<tr>
<td>Channel</td>
<td>The channel at which the TV program is broadcasted</td>
</tr>
</tbody>
</table>

(1) is described in the previous section and WP4 is involved in enriching TV program metadata. More information about enrichment can be found in “D4.2 Semantic Annotation Framework for TV content” [6]. WP3 is involved in (2) and also in the recommendation part. More information about user profiling can be found in “D3.1 User and Context model specification” [4]. A user profile can contain different types of user information:

- Personal information;
- Demographic information;
- User preferences;
- User activities.

Table 3.2 gives an overview of requirements for the user profile according and Table 3.3 gives an overview of requirements for the user (inter)actions according to WP7b.

**Personal information**

Personal information includes information such as name, e-mail address, and image/photo of a person. This information is used to identify and communicate with the user. Furthermore, the use of this information can give the person the feeling that he is interacting with a personalized application. For example, the application can send an e-mail that starts with the name of the user to give that user the feeling that the application gives the user some personal attention. Personal information includes:

- First name;
- Surname;
- E-mail address;
Image/photo;
Username;
Password;
Date of birth;
Place of birth.

Demographic information

Demographic information is information about personal characteristics of a user, such as age and gender. These characteristics can be used to classify users into groups, which share common interests. This classification can be used in the recommender system to apply demographic filtering. Also for usability, demographic information can be used. The age characteristic for instance can be used to determine the font size in an application. Usually older people prefer larger font sizes. Another example that is related to the age and gender characteristic is the use of colours and themes within the application. Young children are very sensitive for colours and themes that correspond to their gender. Demographic information includes:

- Gender;
- Age;
- Level of education;
- Marital status;
- Ethnicity;
- Health;
- Economic activity;
- Income.

Preferences

Preferences are the user’s preferred settings for the client application(s). You can think of a user’s preferred:

- Language;
- Theme;
- Starting page.
### Table 3.2: User profile requirements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Personal information</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>The name of the user (e.g. the firstname, the lastname)</td>
</tr>
<tr>
<td>User identification</td>
<td>Information to identify/verify the user (e.g. a username/password combination)</td>
</tr>
<tr>
<td>E-mail address</td>
<td>E-mail address of the user</td>
</tr>
<tr>
<td><strong>Demographic information</strong></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>The gender of the user (male/female)</td>
</tr>
<tr>
<td>Age</td>
<td>The age of the user</td>
</tr>
<tr>
<td>Level of education</td>
<td>The level of education of the user</td>
</tr>
</tbody>
</table>

### Table 3.3: User interaction requirements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User identification</td>
<td>Sign up/in to gain access to the PEPG and sign off to exit the PEPG</td>
</tr>
<tr>
<td>Viewing</td>
<td>View the details related to TV programs</td>
</tr>
<tr>
<td>Watching</td>
<td>Watch TV programs</td>
</tr>
<tr>
<td>Recording</td>
<td>Set and remove recordings for TV programs</td>
</tr>
<tr>
<td>Reminding/Notification</td>
<td>Set and remove reminders for TV programs</td>
</tr>
<tr>
<td>Rating</td>
<td>Set ratings for TV programs. A rating given by a user for a TV program</td>
</tr>
<tr>
<td></td>
<td>represents the level of interest of that user in that program</td>
</tr>
<tr>
<td>Selection/Favourites</td>
<td>Set and remove TV programs to personal selected TV programs (most favourite TV programs)</td>
</tr>
</tbody>
</table>
**Context**

Can hold various kinds of information. However, not all contextual aspects are relevant in the TV domain. Furthermore, although some contextual aspects do influence the watching behaviour of a user, like the mood of the user, they are not always feasible to monitor. The following contextual aspects are particularly important for describing a user’s situation while interacting with interactive television applications and are feasible to retrieve with the available technologies and resources. Table 3.4 gives an overview of requirements for context according to WP7b. More information about context can be found in “D3.1 User and Context model specification” [4].

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>The time at which the action did occur. (e.g. date, time, weekends)</td>
</tr>
<tr>
<td>Location</td>
<td>The type of device at which the action did occur. The (geographical) location the user was at that moment.</td>
</tr>
</tbody>
</table>

3.1.6 Personalized advertising

To be able to personalize advertisements (ads), also here two important aspects come into play:

1. The knowledge of the system about the ads;
2. The knowledge of the system about the users.

(2) is described in the previous section. There are some aspect which could be useful to personalize ads, see Table 3.5. WP2 is involved in specifying a metadata schema for describing and classifying ads. More information can be found in “D2.1 Requirements analysis” [5]. To personalize ads, the recommendation service developed in WP3 could be used. Table 3.5 gives an overview of requirements for ad specification according to WP7b.

**Ad placement**

The best placement of an ad depends on a lot of variables. For example, while taking the three devices (Desktop, mobile device and set-top box) into account:

- Size of the screen to place the ad;
- Size of the ad;
- Type of media of the ad;
- Design of the ad;
- Type of content on the screen;
- Design of content on the screen;
Table 3.5: Ad specification

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Descriptive information</strong></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Title of the ad</td>
</tr>
<tr>
<td>Description</td>
<td>Description of content in the ad</td>
</tr>
<tr>
<td>Duration</td>
<td>The duration of the ad</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the ad (e.g. TV commercial, Website banner)</td>
</tr>
<tr>
<td>Campaign</td>
<td>The campaign the ad is associated to</td>
</tr>
<tr>
<td>Product/Service</td>
<td>An identification, name and description of the product or the service the ad is developed for. This can also contain the global brand</td>
</tr>
<tr>
<td>Category</td>
<td>The category the product or service can be classified in</td>
</tr>
<tr>
<td>Text</td>
<td>The text used in the ad (e.g. dialogue used in the ad)</td>
</tr>
<tr>
<td>Keywords</td>
<td>One or more keywords representing the nature of the ad</td>
</tr>
<tr>
<td>Intended audience</td>
<td>One or more categories the intended audience for the spot can be classified in (e.g. age group, geographical location group)</td>
</tr>
<tr>
<td><strong>Exploitation information</strong></td>
<td></td>
</tr>
<tr>
<td>Validity Period</td>
<td>The period of permitted exploitation of the ad</td>
</tr>
<tr>
<td>Planned publication</td>
<td>The date/time when the ad is planned to be published</td>
</tr>
<tr>
<td>Target delivery platform</td>
<td>The delivery media on which the ad is planned to published (e.g. TV, Internet, mobile phone)</td>
</tr>
<tr>
<td><strong>Technical information</strong></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Information about the file (e.g. name, format)</td>
</tr>
<tr>
<td>Audio</td>
<td>Information about the audio used in the ad (e.g. codec, format, language)</td>
</tr>
<tr>
<td>Video</td>
<td>Information about the video used in the ad (e.g. codec, format, aspect ratio)</td>
</tr>
<tr>
<td>Text</td>
<td>Information about the text used in the ad (e.g. language, encoding)</td>
</tr>
<tr>
<td><strong>Credits information</strong></td>
<td></td>
</tr>
<tr>
<td>Person name</td>
<td>Name of a person or company</td>
</tr>
<tr>
<td>Role</td>
<td>The role of the person/company (e.g. actor, producer)</td>
</tr>
<tr>
<td>Character name</td>
<td>The name of the character related to the person/company (e.g. a character played by an actor in the ad)</td>
</tr>
</tbody>
</table>
• Place of focus of the user while watching the screen;
• Activity of the user while watching the screen;
• Duration of the user watching the screen;

Certain locations on a Web page tend to be more successful for placing ads than others. See Figure 3.12\(^1\). For mobile pages most of the time ads are placed on top of the screen, or at the bottom of the screen.

![Figure 3.12: Heat map for ad placement on Web pages](image)

As examples for ad placement, the screenshots of Figure 3.1 and Figure 3.2 show ads in the iFanzy Web application at the right side of the screen. The designs of set-top box GUI(I) show ads at the top right of the screen in Figure 3.4 and Figure 3.3. Figure 3.9 shows an ad in the design of the iPhone GUI. The ads in the designs of set-top box GUI (II) are at the top left and bottom right of the screen, see Figure 3.6 and Figure 3.7.

**Ad insertion in video technology based on Regions of Interest (RoI) and Sequences of Interest (SoI)**

The technology context is digital video ad-insertion. The goal of this technology is to automatically insert advertising into a video at the best position and at the best time. Indeed, the time and the position of the ad inlay have a considerable impact on the perception and the interest of this Ad the viewer can have. The problem is currently resolved manually but the NoTube project is aiming at an automatic solution. The idea is to couple two algorithms to detect the best position and at the best time:

• The first one will find, in the video, the best sequence which corresponds to the best time to insert the advertising (according to motion activity, sound loudness, etc ) in order to detect the quietest sequence or scenes with minor changes.
• The second one will find, in this sequence, the best region to insert the advertising (according to the analysis of the regions of non-interest).

30 of 61
Configuration of ad insertion with Web services

To configure the ad insertion on the platform, the followings steps have to be done:

- Definition of the ad insertion workflow which is based on the different basic workflows like MPEG2FileInput
- Instantiation of this workflow with a job creation

These two operations can be done with Web services. For an easier management of the platform, the ad insertion workflow is already created on the platform and the user needs only to create a job on this workflow.

A workflow is a model that describes a treatment like ad insertion and consists of a sequence of connected steps. A Job is an instantiation of a workflow. That corresponds to a user request. (For example, insert the car ad in the movie “Star Wars”. The workflowIdRef indicates the referenced workflow.

Figure 3.15 shows a workflow for ad insertion.

The XML description of ad insertion workflow is based on various basic workflows which can be found in Appendix B.

Ad insertion Web services

The following Web services can be called after the workflow configuration of the platform.

Workflow get list
This Web service operation is used to know the different configured workflows and get back the workflow id and workflow name. Refer to “WorkflowGetList.xsd” file (see Appendix B.1) for more details.

Job create
This Web service operation is used to create a job corresponding to a workflow (for example, to create the job ‘insertion of ad’ and start this job) Refer to “JobCreate.xsd” file (see Appendix B.1) for more details.

Job modify
This Web service operation is used to modify an existing job. Refer to “JobModify.xsd” file (see Appendix B.1) for more details.

1http://google.com/support/adsense/bin/answer.py?answer=17954
Figure 3.14: Ad insertion job

Figure 3.15: Ad insertion workflow
### Table 3.6: Ad insertion - Workflow get list

<table>
<thead>
<tr>
<th>request parameters</th>
<th>response possibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of items</td>
<td>For each workflow:</td>
</tr>
<tr>
<td></td>
<td>- Workflow name</td>
</tr>
<tr>
<td></td>
<td>- Workflow id</td>
</tr>
<tr>
<td></td>
<td>- Workflow brief description</td>
</tr>
<tr>
<td></td>
<td>- Webservice ‘instability’ flag</td>
</tr>
<tr>
<td></td>
<td>- Creator (User, System release)</td>
</tr>
<tr>
<td></td>
<td>- Type</td>
</tr>
<tr>
<td>Bad rights</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram of Workflow Get List](image)

Figure 3.16: Ad insertion get list

### Table 3.7: Ad insertion - Job create

<table>
<thead>
<tr>
<th>request parameters</th>
<th>response possibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job name</td>
<td>Job Id</td>
</tr>
<tr>
<td>WorkflowIdRef</td>
<td>Bad rights</td>
</tr>
<tr>
<td>Priority</td>
<td>Coherency problem with its report</td>
</tr>
<tr>
<td>Job parameters</td>
<td>Workflow not found</td>
</tr>
</tbody>
</table>

### Table 3.8: Ad insertion - Job modify

<table>
<thead>
<tr>
<th>request parameters</th>
<th>response possibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job name</td>
<td>OK</td>
</tr>
<tr>
<td>WorkflowIdRef</td>
<td>Bad rights</td>
</tr>
<tr>
<td>Job parameters</td>
<td>Coherency problem with its report</td>
</tr>
<tr>
<td></td>
<td>Job not found</td>
</tr>
</tbody>
</table>
Figure 3.17: Ad insertion job create

Figure 3.18: Ad insertion job modify
3.2 Technical requirements

3.2.1 Service provider side

In the NoTube project, various sources and services will be used to bring TV experience to a higher level. Alignment is desired to relate objects from various sources with each other. Objects specified in these sources are mapped to objects defined in the specifications which will be developed in this project. More information about alignment in the project can be found in “D1.3 Vocabulary alignment service of NoTube vocabularies” [7]. Mappings to the following specifications are required:

- EPG/TV content metadata specification;
- Ad metadata specification;
- User profile specification;
- User (inter)action specification;
- Context specification.

To be able to do build up a PEPG. Stores, delivery systems and services are required. These can be categorized into:

- EPG/TV metadata;
- EPG/TV material;
- Ad metadata;
- Ad material;
- Ad placement;
- User profiles;
- Recommendations.

The recommendation engine should be able to generate recommendations for TV/Ad material based on a user profile, user (inter)actions of this user and EPG/Ad metadata. Furthermore, the recommendation engine should also be able to take context into account. The EPG and Ad metadata could be enriched by the enrichment service to allow more accurate recommendations or present more and/or better EPG metadata to the user via the clients.

3.2.2 Home ambient side

The home ambient side, which represents the clients, needs to be connected and able to communicate with the service provider side, which represent the stores, delivery systems and services. There are some technical requirements for the clients:

- User data should be handled in a secure way and privacy issues should be taken into account;
• The Web client should be served by a Web server and accessible via common Web browsers;
• The mobile client should be supported and accessible by a mobile device;
• The STB application should be running on a set-top box and be accessible via a TV by using a remote control;

3.2.3 Security and privacy issues

Security and privacy issues are an important aspect of the project. Personal data is handled on both sides, on the service provider side as well as on the home ambient side. A user should be able to connect to various devices which also should be handled with caution. There is a trade-off between user friendliness and security. Both aspects should be met. A user usually does not want to load his personal profile on his set-top box by for example typing in lengthy username/password combination. Another area that should be handled with care is when user are able to upload content to a system. This use case is dealing with UGC which brings up security issues.

3.3 Development plan

Table 3.9 gives an overview about what is being develop within this use case.
### Table 3.9: Development plan

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad metadata</td>
<td>M23</td>
</tr>
<tr>
<td>EPG metadata</td>
<td>M13</td>
</tr>
<tr>
<td>User profile</td>
<td>M13</td>
</tr>
<tr>
<td>Context</td>
<td>M13</td>
</tr>
</tbody>
</table>

### Analysis and Design

<table>
<thead>
<tr>
<th></th>
<th>Web</th>
<th>Mobile</th>
<th>STB</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPG</td>
<td>M13</td>
<td>M13</td>
<td>M13</td>
</tr>
<tr>
<td>Ad placement</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Recommendations/Ratings</td>
<td>M13</td>
<td>M13</td>
<td>M13</td>
</tr>
<tr>
<td>Reminders</td>
<td>M23</td>
<td>M13</td>
<td>M23</td>
</tr>
<tr>
<td>Recordings</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>User profile coupling</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Selection/Favourites</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Similar programs</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Sharing with friends</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Multilingual support</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>UGC</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Multimodal support</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
</tbody>
</table>

### Implementation

<table>
<thead>
<tr>
<th></th>
<th>Web</th>
<th>Mobile</th>
<th>STB</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPG</td>
<td>M13</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Ad placement</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Recommendations/Ratings</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Reminders</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Recordings</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>User profile coupling</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Selection/Favourites</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Similar programs</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Sharing with friends</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Multilingual support</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>UGC</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>(Multimodal support)</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
</tbody>
</table>

### Testing

<table>
<thead>
<tr>
<th></th>
<th>Web</th>
<th>Mobile</th>
<th>STB</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPG</td>
<td>M13</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Ad placement</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Recommendations/Ratings</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Reminders</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Recordings</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>User profile coupling</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Selection/Favourites</td>
<td>M23</td>
<td>M23</td>
<td>M23</td>
</tr>
<tr>
<td>Similar programs</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Sharing with friends</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Multilingual support</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>UGC</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>(Multimodal support)</td>
<td>M33</td>
<td>M33</td>
<td>M33</td>
</tr>
<tr>
<td>Accuracy Recommendations</td>
<td>M36</td>
<td>M36</td>
<td>M36</td>
</tr>
<tr>
<td>Accuracy Similar programs</td>
<td>M36</td>
<td>M36</td>
<td>M36</td>
</tr>
</tbody>
</table>
4. First prototype

To demonstrate part of the work done, a prototype is developed. The goal is to demonstrate an EPG in a nice GUI while making use of services developed in the project. First an architectural overview is given. Then the scenario, abstracted from the three initial scenario’s in the use case is described. This story is translated into swimlane activity diagrams. Finally a description of the implementation is given.

4.1 Architecture

The global architecture used as reference in the project can be found in [2]. The clients, that represent the home ambient side, are developed in WP7b. Also the adapter and the ROI detection service are developed in this WP. The other services and components are developed by other workpackages.

![Figure 4.1: NoTube WP7b architecture](image)

4.2 Story

For the first prototype, the following scenario story is used for the demonstrator

Sena, who lives in Holland, wants to know if there is something interesting on TV this week. She visits iFanzy.nl to have an overview of what is broadcasted on TV. Sena logs in and sees what TV programs are recommended to her. She browses the EPG, sees one of her favourite programs and decides to rate it and add it to her personal planner. Furthermore she sees a movie and watches the trailer of this movie. Besides that, she also wants to check whether this movie has a high IMDb rating or not. After browsing the iFanzy website, reading some information about TV programs, rating some programs, put some programs in her planner and so on...Sena forget time...
and suddenly has to hurry to catch her flight to join her boyfriend Tarik in Turkey. (See Figure 4.2)

![Figure 4.2: Scene 1](image)

Fortunately, Sena did catch her flight and landed in Istanbul safely. While waiting for her luggage at the airport, she is bored. She gets her iPhone and start up the iFanzy iPhone application. She browses what’s on this evening and she sets a reminder for a movie recommended to her. There is her luggage already. (See Figure 4.3)

![Figure 4.3: Scene 2](image)

Later that day, Sena and Tarik have a nice dinner together. After dinner, while they have a drink, Sena gets a notification that the movie she set a reminder before starts in about 15 minutes. Sena asks Tarik if he would like to watch this movie with her. Tarik is not really interested in this movie. However he likes the idea of watching a movie together that evening. He grabs his laptop, logs in on iFanzy.tv and searches for a movie they both like. They watch a movie together happily ever after. (See Figure 4.4)

### 4.3 Swimlanes

The three parts of the scenario are described by the following three swimlanes activity diagrams: Scene 1 in Figure 4.5, Scene 2 in Figure 4.6 and Scene 3 in Figure 4.7. The second and third column in these figures represent the home ambient side and the last four columns represent
Figure 4.4: Scene 3

Figure 4.5: Swimlane - Scene 1
Figure 4.6: Swimlane - Scene 2

Figure 4.7: Swimlane - Scene 3
4.4 Implementation

The iFanzy clients communicate with the iFanzy server via the SOAP protocol for exchanging information. The Semantic TV Broker, which is used in the NoTube project and the achieve-goal' functions are exposed through a REST API. A client process can invoke a goal by making an HTTP GET request to the IRS server. The iFanzy server as well as the Semantic Broker have their own interfaces that vary from each other. The interfaces of the iFanzy server and the Semantic Broker are thus not compatible. Therefore a translation is needed from the iFanzy client requests into requests to the Semantic Broker and in return, the responses from the Semantic Brokers need to be translated into responses the iFanzy clients can accept and understand. More information about the Semantic Broker’s get-EPG-metadata-goal can be found in “D5.1a Semantic TV Broker Requirements” [3]. In Appendix C the WSDL, which describes the iFanzy Web service, is given.

The translation part is done by using the adapter design pattern:

- Target: the interface that the iFanzy clients use;
- Client: an iFanzy client;
- Adaptee: the interface of the Semantic Broker;
- Adapter: adapts the interface of Adaptee to the Target interface.

There are two types of Adapter patterns:

- Object adapter pattern (implemented with object composition);
- Class adapter pattern (implemented with multiple inheritance)

The JAVA programming language, which is used in the implementation of the adapter, does not support multiple inheritance. In the implementation, the object adapter patterns is used as shown in Figure 4.8. A component and deployment diagram is given in Figure 4.9

![Figure 4.8: Semantic Broker-iFanzy Adapter](image_url)
Figure 4.9: Component and deployment diagram
5. Evaluation

According to the development plan: Requirements for Ad metadata specification, EPG metadata specification, User profile specification and context are given in section 3.1.4 section 3.1.5 and section 3.1.5 as input for other work packages.

Analysis is done and designs for graphical user interfaces are given to build up a PEPG for various devices in section 3.1.1. Some research is done in Ad placement and design, see section 3.1.6.

In the first prototype communication between the iFanzy Web application and the Semantic Broker is implemented and an EPG filled with data from services developed in the project is possible. The Web applications also show that implementations are done regarding multilinguality. A user can set reminders for TV programs and get notified by using the iFanzy iPhone application. A user can also do ratings and get recommendations for TV programs in the developed iFanzy Web and iPhone application, although at the moment in the prototype only the EPG data is delivered by services developed in the project.

5.1 Recommendations and similar-programs testing

To determine the accuracy of the iFanzy recommendation engine and the iFanzy similar-programs engine, two tests were applied (See Figure 5.1 and Figure 5.2). The first test was conducted by 50 end users who used the iFanzy Web application for over 3 weeks (training set) and gave ratings to 50 TV programs at the end of the test (the test set). The second test was conducted by 1200 respondents. These respondents were given 30 TV programs with for each program three options: like, don’t like and unknown. If the like or don’t like option was chosen, then three 'similar programs' as given by the similarity engine were given and the user had to say for each program whether it was indeed a similar TV program or not. Besides that, he had the opportunity to add similar TV programs by himself.

The collected data in these tests can be used to test the recommendation and similar-program engines developed in the NoTube project. The outcomes of the iFanzy and NoTube tests can be compared with each other to have some knowledge about performance of both recommender systems.
Figure 5.1: Test for determine accuracy of recommendation algorithm

Figure 5.2: Test for determine accuracy of recommendation and similar-programs algorithm
6. Current status and future work

Besides the development done for the first prototype, analysis is done in various areas. Next steps will be to refine the findings and take these to a next level by implementing functionalities for the second prototype. Effort should be put into further implementation of the graphical user interfaces for the various devices and into implementation and testing of functionalities that are available via services delivered by other work packages in the project. Specifically more effort will be spent to implement functionalities based on the recommendation, user profiling and user activity services into the GUI's. Another main objective will be to make the described services available on various end devices, including the synchronisation of these devices with each other in a user friendly way.
REFERENCES


A. EPG data examples

A.1 English EPG data example

Listing A.1: English EPG data example

```xml
<sxml version="1.0" encoding="UTF-8">  
<diavtech_epg_version "3.6">  
<schedule channel_id="239" channel_name="Test Channel" utc_offset="0">  
<event event_id="618246239200961030110" event_time="2009-06-13 11:10:00" end_time="2009-06-13 11:10:00" genre="Comedy" title="True Story" description_language="en" Synopsis="The second half of the Sandinista Revolution in Nicaragua and the conflict that ensued with the United States."
</event>  
</schedule>  
</diavtech_epg>
```

A.2 Turkish EPG data example

Listing A.2: Turkish EPG data example

```xml
<sxml version="1.0" encoding="UTF-8">  
<diavtech_epg_version "3.6">  
<schedule channel_id="239" channel_name="Test Channel" utc_offset="0">  
<event event_id="618246239200961030110" event_time="2009-06-13 11:10:00" end_time="2009-06-13 11:10:00" genre="Comedy" title="True Story" description_language="en" Synopsis="The second half of the Sandinista Revolution in Nicaragua and the conflict that ensued with the United States."
</event>  
</schedule>  
</diavtech_epg>
```

A.3 Korean EPG data example

Listing A.3: Korean EPG data example

```xml
<sxml version="1.0" encoding="UTF-8">  
<diavtech_epg_version "3.6">  
<schedule channel_id="239" channel_name="Test Channel" utc_offset="0">  
<event event_id="618246239200961030110" event_time="2009-06-13 11:10:00" end_time="2009-06-13 11:10:00" genre="Comedy" title="True Story" description_language="en" Synopsis="The second half of the Sandinista Revolution in Nicaragua and the conflict that ensued with the United States."
</event>  
</schedule>  
</diavtech_epg>
```
B. Ad insertion

B.1 Ad insertion basic workflows

The XML description of ad insertion workflow is based on the following basic workflows:

Listing B.1: Description of Mpeg2 File Input Workflow

```xml
<WORKFLOW name="Mpeg2FileInput">
  <PARAM name="FileMode" type="String"/>
  <PARAM name="FileName" type="String"/>
  <PARAM name="FilePath" type="String"/>
  <PARAM name="IPAddress" type="String"/>
  <PARAM name="Login" type="String"/>
  <PARAM name="Password" type="String"/>
  <OUTPUTS>
    <OUTPUTDEF type="CompressedVideo" name="Video"/>
    <OUTPUTDEF type="CompressedAudio" name="Audio"/>
  </OUTPUTS>
</WORKFLOW>
```

Listing B.2: Description of Video Decoder Workflow

```xml
<WORKFLOW name="Video_Decoder">
  <INPUTS>
    <INPUTDEF type="CompressedVideo" name="DecoderInput"/>
  </INPUTS>
  <OUTPUTS>
    <OUTPUTDEF type="UncompressedVideo" name="DecoderOutput"/>
  </OUTPUTS>
</WORKFLOW>
```

Listing B.3: Description of Audio Decoder Workflow

```xml
<WORKFLOW name="Audio_Decoder">
  <INPUTS>
    <INPUTDEF type="CompressedAudio" name="DecoderInput"/>
  </INPUTS>
  <OUTPUTS>
    <OUTPUTDEF type="UncompressedAudio" name="DecoderOutput"/>
  </OUTPUTS>
</WORKFLOW>
```

Listing B.4: Description of basic Ad insertion Workflow

```xml
<WORKFLOW name="BasicAdInsertion">
  <INPUTS>
    <INPUTDEF type="UncompressedVideo" name="VideoAdInsertionInput"/>
    <INPUTDEF type="UncompressedAudio" name="AudioAdInsertionInput"/>
  </INPUTS>
  <OUTPUTS>
    <OUTPUTDEF type="UncompressedVideo" name="VideoAdInsertionOutput"/>
    <OUTPUTDEF type="UncompressedAudio" name="AudioAdInsertionOutput"/>
  </OUTPUTS>
</WORKFLOW>
```

Listing B.5: Description of MPEG2 video encoder Workflow

```xml
<WORKFLOW name="MPEG2Encoder">
  <PARAM name="Bitrate" type="UInt32" default="384000" unit="bps"/>
  <PARAM name="MinGop" type="UInt32" default="1"/>
  <PARAM name="MaxGop" type="UInt32" default="50"/>
  <INPUTS>
    <INPUTDEF type="UncompressedVideo" name="EncoderInput"/>
  </INPUTS>
  <OUTPUTS>
    <OUTPUTDEF type="CompressedVideo" name="EncoderOutput"/>
  </OUTPUTS>
</WORKFLOW>
```

Listing B.6: Description of MPEG1 audio encoder Workflow

```xml
<WORKFLOW name="MPEG1Encoder">
  <PARAM name="Bitrate" type="UInt32" default="480000" unit="bps"/>
  <PARAM name="Language" type="String" default="eng"/>
  <PARAM name="Mode" type="String" default="Implicit"/>
  <RESTRICTION>
    <ENUMERATION value="Implicit"/>
    <ENUMERATION value="Explicit"/>
  </RESTRICTION>
</WORKFLOW>
```
Listing B.7: Description of Ad insertion workflow

<Inputs>
  <InputDef type="UncompressedAudio" name="AudioEncoderInput" />
</Inputs>

<Outputs>
  <OutputDef type="CompressedAudio" name="AudioEncoderOutput" />
</Outputs>

<Workflow name="AdInsertion" type="offline" priority="0..50" id="AdInsertion">
  <Tasks>
    <Workflow id="100" workflowIdRef="Mpeg2FileInput">
      <Param name="FileName" visibility="public" label="Input Video File" />
      <Param name="FilePath" visibility="public" label="Input Video File Path" />
      <Param name="IPAddress" visibility="protected" value="ftp.notube.com" />
      <Param name="Login" visibility="protected" value="NoTube" />
      <Param name="Password" visibility="protected" value="NoTube" />
    </Workflow>
    <Workflow id="101" workflowIdRef="Mpeg2FileInput">
      <Param name="FileName" visibility="public" label="Input ad File" />
      <Param name="FilePath" visibility="public" label="Input ad File Path" />
      <Param name="IPAddress" visibility="protected" value="ftp.notube.com" />
      <Param name="Login" visibility="protected" value="NoTube" />
      <Param name="Password" visibility="protected" value="NoTube" />
    </Workflow>
    <Workflow id="102" workflowIdRef="VideoDecoder">
      <InputRef name="EncoderInput" id="19" />
      <OutputRef name="DecoderOutput" id="10" />
    </Workflow>
    <Workflow id="103" workflowIdRef="AudioDecoder">
      <InputRef name="EncoderInput" id="19" />
      <OutputRef name="DecoderOutput" id="6" />
    </Workflow>
    <Workflow id="104" workflowIdRef="VideoDecoder">
      <InputRef name="EncoderInput" id="19" />
      <OutputRef name="DecoderOutput" id="7" />
    </Workflow>
    <Workflow id="105" workflowIdRef="AudioDecoder">
      <InputRef name="EncoderInput" id="19" />
      <OutputRef name="DecoderOutput" id="5" />
    </Workflow>
    <Workflow id="106" workflowIdRef="BasicAdInsertion">
      <InputRef name="EncoderInput" id="19" />
      <InputRef name="VideoAdInsertionInput" id="13" />
      <InputRef name="AdVideoAdInsertionInput" id="14" />
      <InputRef name="AudioAdInsertionInput" id="15" />
      <OutputRef name="VideoAdInsertionOutput" id="17" />
      <OutputRef name="AudioAdInsertionOutput" id="18" />
    </Workflow>
    <Workflow id="107" workflowIdRef="MPEG1Encoder">
      <Param name="Bitrate" visibility="public" label="Video bitrate" />
      <Restriction RANGE min="100000" max="200000" />
    </Workflow>
    <Workflow id="108" workflowIdRef="MPEG1Encoder">
      <Param name="Bitrate" visibility="public" label="Audio bitrate default" min="100000" />
    </Workflow>
    <Workflow id="109" workflowIdRef="MPEG2FileOutput">
      <Param name="FileName" visibility="public" label="Output file name" />
      <Param name="FilePath" visibility="protected" value="ftp.notube.com" />
      <Param name="Login" visibility="protected" value="NoTube" />
      <Param name="Password" visibility="protected" value="NoTube" />
      <Param name="Program tracking visibility="protected" value="1" />
      <Param name="VideoPID" visibility="private" value="0x100" />
      <Param name="AudioPID" visibility="private" value="0x101" />
    </Workflow>
  </Tasks>
</Workflow>

<Inputs>
  <InputDef name="AudioEncoderInput" />
</Inputs>

<Outputs>
  <OutputDef name="AudioEncoderOutput" />
</Outputs>

<Workflows>
  <Workflow workflowIdRef="MPEG2FileOutput"/>
  <Workflow workflowIdRef="VideoDecoder"/>
  <Workflow workflowIdRef="AudioDecoder"/>
  <Workflow workflowIdRef="BasicAdInsertion"/>
  <Workflow workflowIdRef="MPEG1Encoder"/>
  <Workflow workflowIdRef="MPEG1Encoder"/>
  <Workflow workflowIdRef="MPEG2FileOutput"/>
</Workflows>

<Tasks>
  <Link/>
</Tasks>

<Link/>
</Link>
</Workflow>

SRC idRef="1" />
<IST idRef="5" />
</Link>

51 of 61
For the job configuration, the user receives the public description of the workflow.

The public description of ad insertion is the following:

Listing B.8: Instantiation of the workflow

```xml
<WFLOW name="AdInsertion" type="offline" priority="0..50" Id="AdInsertion">
  <PARAM name="InputVideoFile" type="String" />
  <PARAM name="InputVideoPath" type="String" />
  <PARAM name="InputAdFile" type="String" />
  <PARAM name="InputAdPathFile" type="String" />
  <PARAM name="VideoBitrate" type="UInt32"/>
  <RESTRICTION RANGE min="100000" max="200000"/>
  <PARAM name="AudioBitrate" type="UInt32" default="100000" />
  <PARAM name="Output FileName" type="String" />
  <PARAM name="OutputFilePath" type="String" />
</WFLOW>
```

Listing B.9: Example Job instantiation of the workflow

```xml
<JOB name="CarAdInsertion" workflowIdRef="AdInsertion" priority="1">
  <PARAMDESC name="InputVideoFile" value="movie.ts" />
  <PARAMDESC name="InputVideoPath" value="FTP/" />
  <PARAMDESC name="InputAdFile" value="car.ts" />
  <PARAMDESC name="InputAdPathFile" value="FTP/" />
  <PARAMDESC name="VideoBitrate" value="150000" />
  <PARAMDESC name="AudioBitrate" value="32000" />
  <PARAMDESC name="OutputFileName" value="FinalMovie.ts" />
  <PARAMDESC name="OutputFilePath" value="FTP/" />
</JOB>
```

Listing B.10: WorkflowGetList.xsd

```xml
  xmlns:mg="MalteseGlobal" xmlns:wg="WorkflowGlobal" xmlns:wd="WorkflowDesc" targetNamespace="WorkflowGetList">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This document gives the description of the "WorkflowGetList" Web service operation.</xsd:documentation>
  </xsd:annotation>
  <xsd:import namespace="MalteseGlobal" schemaLocation="MalteseGlobal.xsd"/>
  <xsd:import namespace="WorkflowGlobal" schemaLocation="WorkflowGlobal.xsd"/>
  <xsd:import namespace="WorkflowDesc" schemaLocation="WorkflowDesc.xsd"/>
</xsd:schema>
```
<xsd:complexType name="TypeWorkflowInfo">
  <xsd:sequence>
    <xsd:element ref="wGetList:WorkflowInfo" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TypeWorkflowGetListRsp">
  <xsd:key ref="wGetList:KeyWorkflowName"/>
  <xsd:sequence>
    <xsd:element ref="wGetList:WorkflowInfo" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TypeRspWorkflowOk">
  <xsd:complexContent>
    <xsd:extension base="mg:TypeRsp">
      <xsd:attribute name="Name" type="xsd:string" use="optional" default=""/>
      <xsd:attribute name="Type" type="wg:TypeWorkflowType" use="optional" default="Not specialized"/>
      <xsd:attribute name="Creator" type="wg:TypeWorkflowCreator" use="optional" default="User"/>
      <xsd:attribute name="Description" type="xsd:string" use="optional"/>
      <xsd:attribute name="WorkflowId" type="wg:TypeWorkflowId" use="required"/>
      <xsd:attribute name="WorkflowRights" type="wg:TypeWorkflowRights" use="optional"/>
      <xsd:attribute name="ImplFlag" type="xsd:boolean" use="required"/>
      <xsd:attribute name="InstanciableFlag" type="xsd:boolean" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="TypeWorkflowGetListReq">
  <xsd:complexContent>
    <xsd:extension base="mg:TypeReq">
      <xsd:attribute name="Name" type="xsd:string" use="optional" default=""/>
      <xsd:attribute name="WorkflowId" type="wg:TypeWorkflowId" use="required"/>
      <xsd:attribute name="WorkflowRights" type="wg:TypeWorkflowRights" use="optional"/>
      <xsd:attribute name="Type" type="wg:TypeWorkflowType" use="optional" default="Not specialized"/>
      <xsd:attribute name="Creator" type="wg:TypeWorkflowCreator" use="optional" default="User"/>
      <xsd:attribute name="CreatorId" type="wg:TypeWorkflowCreatorId" use="optional" default=""/>
      <xsd:attribute name="Description" type="xsd:string" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
Listing B.11: JobCreate.xsd

Listing B.12: JobModify.xsd
  
  <xsd:documentation xml:lang="en">This document gives the description of the "JobModify" Web service operation.</xsd:documentation>
  
  <xsd:import namespace="MalteseGlobal" schemaLocation="MalteseGlobal.xsd"/>
  <xsd:import namespace="JobGlobal" schemaLocation="JobGlobal.xsd"/>
  <xsd:import namespace="WorkflowGlobal" schemaLocation="WorkflowGlobal.xsd"/>
  <xsd:import namespace="WorkflowDesc" schemaLocation="WorkflowDesc.xsd"/>

  <!-- JobModify request -->
  <xsd:complexType name="TypeJobModifyReq">
    <xsd:complexContent>
      <xsd:extension base="mg:TypeReq">
        <xsd:sequence>
          <xsd:element name="JOB"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexType>
  </xsd:complexType>

  <!-- JobModify response -->
  <xsd:complexType name="TypeJobModifyRsp">
    <xsd:complexContent>
      <xsd:extension base="mg:TypeRsp">
        <xsd:sequence minOccurs="0">
          <xsd:element name="RspJobNotFound" />
          <xsd:element name="RspConsistencyReport" />
          <xsd:element name="RspJobOk" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexType>
  </xsd:complexType>
</xsd:schema>
C. iFanzyService WSDL

Listing C.1: iFanzyService WSDL

```xml
  <types>
      <element name="CreateProfile">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="Provider" type="tns:ProviderData"/>
            <element maxOccurs="unbounded" minOccurs="0" name="AdditionalInformation" type="tns:NameValuePair"/>
          </sequence>
        </complexType>
      </element>
      <element name="CreateProfileResponse">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="ProfileID" type="xsd:string"/>
            <element maxOccurs="unbounded" minOccurs="0" name="Errors" type="tns:ErrorMessage"/>
          </sequence>
        </complexType>
      </element>
      <element name="NameValuePair">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="Name" type="xsd:string"/>
            <element maxOccurs="1" minOccurs="1" name="Value" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="ProviderData">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="ID" type="xsd:string"/>
            <annotation>
              <documentation/>
            </annotation>
          </sequence>
        </complexType>
      </element>
      <element name="ErrorMessage">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="Type" type="xsd:string"/>
            <element maxOccurs="1" minOccurs="1" name="Message" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="DeleteProfile">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="Provider" type="tns:ProviderData"/>
            <element maxOccurs="1" minOccurs="1" name="ProfileID" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="DeleteProfileResponse">
        <complexType>
          <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="Errors" type="tns:ErrorMessage"/>
          </sequence>
        </complexType>
      </element>
      <element name="ResetProfile">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="Provider" type="tns:ProviderData"/>
            <element maxOccurs="1" minOccurs="1" name="ProfileID" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="ResetProfileResponse">
        <complexType>
          <sequence>
          </sequence>
        </complexType>
      </element>
    </schema>
  </types>
</definitions>
```
<xsd:element maxOccurs="1" minOccurs="0" name="Type" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="ServiceID" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="StartTime" type="xsd:dateTime"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="EndTime" type="xsd:dateTime"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="Source" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="ProductionYear" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="ProductionCountry" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="Rating" type="xsd:double"/>
</xsd:element>

<xsd:element maxOccurs="unbounded" minOccurs="0" name="Descriptions" type="tns:Descriptions"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="Genres" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="Resources" type="tns:Resource"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="Cast" type="tns:Cast"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="FilterResult" type="tns:FilterResult"/>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Filter">
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1" name="Type" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SearchByContentID">
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1" name="Provider" type="tns:ProviderData"/>
</xsd:element>

<xsd:element maxOccurs="unbounded" minOccurs="1" name="ContentID" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="RecommendationSettings" type="tns:RecommendationSettings"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="1" name="ResultSettings" type="tns:ResultSettings"/>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType>
</xsd:element>

<xsd:complexType name="SearchByContentIDResponse">
<xsd:sequence>
<xsd:element maxOccurs="unbounded" minOccurs="0" name="Content" type="tns:ContentElement"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="TotalContentCount" type="xsd:integer"/>
</xsd:element>

<xsd:element maxOccurs="unbounded" minOccurs="0" name="Errors" type="tns:ErrorMessage"/>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType>
</xsd:element>

<xsd:complexType name="SearchBySimilarity">
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1" name="Provider" type="tns:ProviderData"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="1" name="BaseContentID" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="1" name="Period" type="tns:Period"/>
</xsd:element>

<xsd:element maxOccurs="unbounded" minOccurs="1" name="Sources" type="xsd:string"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="0" name="RecommendationSettings" type="tns:RecommendationSettings"/>
</xsd:element>

<xsd:element maxOccurs="1" minOccurs="1" name="ResultSettings" type="tns:ResultSettings"/>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<wsdl:portType name="IFanzyServiceSoapBinding">
  <wsdl:operation name="CreateProfile">
    <wsdl:input message="tns:CreateProfileRequest"/>
    <wsdl:output message="tns:CreateProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteProfile">
    <wsdl:input message="tns:DeleteProfileRequest"/>
    <wsdl:output message="tns:DeleteProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ResetProfile">
    <wsdl:input message="tns:ResetProfileRequest"/>
    <wsdl:output message="tns:ResetProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SearchByPeriod">
    <wsdl:input message="tns:SearchByPeriodRequest"/>
    <wsdl:output message="tns:SearchByPeriodResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SearchByContentID">
    <wsdl:input message="tns:SearchByContentIDRequest"/>
    <wsdl:output message="tns:SearchByContentIDResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SearchBySimilarity">
    <wsdl:input message="tns:SearchBySimilarityRequest"/>
    <wsdl:output message="tns:SearchBySimilarityResponse"/>
  </wsdl:operation>
  <wsdl:operation name="RateContent">
    <wsdl:input message="tns:RateContentRequest"/>
    <wsdl:output message="tns:RateContentResponse"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:service name="IFanzyService">
  <wsdl:port binding="tns:IFanzyServiceSoapBinding" name="IFanzyService">
    <soap:address location="http://sigs06.8180/IFanzy/2009/07/IFanzy"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>