



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n: 761999



EasyTV: Easing the access of Europeans with disabilities to converging media and content.

EasyTV user interface adaptation framework

EasyTV Project

H2020. ICT-19-2017 Media and content convergence. – IA Innovation action.

Grant Agreement n°: 761999

Start date of project: 1 Oct. 2017

Duration: 30 months

Document. ref.: D4.3

Disclaimer

This document contains material, which is the copyright of certain EasyTV contractors, and may not be reproduced or copied without permission. All EasyTV consortium partners have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information. The reproduction of this document or of parts of it requires an agreement with the proprietor of that information. The document must be referenced if is used in a publication.

The EasyTV Consortium consists of the following partners:

	Partner Name	Short name	Country
1	Universidad Politécnica de Madrid	UPM	ES
2	Engineering Ingegneria Informatica S.P.A.	ENG	IT
3	Centre for Research and Technology Hellas/Information Technologies Institute	CERTH	GR
4	Mediavoice SRL	MV	IT
5	Universitat Autònoma Barcelona	UAB	ES
6	Corporació Catalana de Mitjans Audiovisuals SA	CCMA	ES
7	ARX.NET SA	ARX	GR
8	Fundación Confederación Nacional Sordos España para la supresión de barreras de comunicación	FCNSE	ES
9	Unione Italiana dei ciechi e degli ipovedenti	UICI	IT

PROGRAMME NAME:	H2020. ICT-19-2017 Media and Content Convergence – IA Innovation Action
PROJECT NUMBER:	761999
PROJECT TITLE:	EASYTV
RESPONSIBLE UNIT:	CERTH
INVOLVED UNITS:	
DOCUMENT NUMBER:	D4.3
DOCUMENT TITLE:	EasyTV user interface adaptation framework
WORK-PACKAGE:	WP4
DELIVERABLE TYPE:	Prototype
CONTRACTUAL DATE OF DELIVERY:	30/09/2019
LAST UPDATE:	30/09/2019
DISTRIBUTION LEVEL:	PU

Distribution level:

PU = *Public*,

RE = *Restricted to a group of the specified Consortium*,

PP = *Restricted to other program participants (including Commission Services)*,

CO = *Confidential, only for members of the LASIE Consortium (including the Commission Services)*

Document History

VERSION	DATE	STATUS	AUTHORS, REVIEWER	DESCRIPTION
v.0.1	9-18-2019	Draft	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Table of Contents Document structure Initial version of chapter (1, 2, 3, 4)
v.0.2	9-20-2019	Draft	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Refinement of Chapter 02, 03
v.0.3	9-21-2019	Draft	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Refinement of Chapter 04, 05
v.0.4	9-26-2019	Draft	UPM	Updates section 2.1 Add section 3.3.3
V0.5	9-26-2019	Peer-review/ revision	UAB	General review of doc
v.0.6	9-30-2019	Final ready for submission	Salim Gannoum (CERTH/ITI) Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Addressing peer-review comments. Refinement of Chapters 01, 02, 03, 04, 05

Definitions, Acronyms and Abbreviations

ACRONYMS / ABBREVIATIONS	DESCRIPTION
GUI	Graphical User Interface
UI	User Interface
RBMM	Rule based matchmaker
STMM	Statistical based matchmaker
HBMM	Hybrid matchmaker
CS	Companion Screen
HbbTV	Hybrid Broadcast Broadband TV
DBSCAN	Density-based spatial clustering of applications with noise (clustering algorithm)
FOL	First Order Logic
KB	Knowledge base
WCAG	Web Content Accessibility Guidelines
TTS	Text To Speech

Table of Contents

Contents

1.	introduction.....	12
2.	EasyTV user model	13
2.1.1.	Generic preferences.....	13
2.1.2.	Application specific preferences.....	13
2.1.3.	Conditional preferences	17
2.1.4.	Contextual information	17
2.1.5.	Conditional preferences example.....	18
2.2.	User model implementation.....	18
3.	User model editor	22
3.1.	Overview.....	22
3.2.	Registration and login.....	22
3.3.	Access to the editor	23
3.4.	Creating a profile	24
3.5.	Editing a profile.....	24
3.6.	Removing a profile	24
3.7.	Initializing a profile.....	24
3.8.	Initialization tests	27
3.8.1.	Color blindness	28
3.8.2.	Contrast sensitivity	30
3.9.	Design approach	30
3.9.1.	Font type	31
3.9.2.	Font size	31
3.9.3.	Font style	32
3.9.4.	Icons	32
3.9.5.	Colours and contrast.....	32
3.9.6.	Layout and Components choices.....	33
4.	Hyper-personalization framework	33
4.1.	Architectural overview	33
4.2.	User model management framework	34
4.2.1.	Login and registration.....	35
4.2.2.	Technologies used	35
4.2.3.	Application specific interface.....	35
4.3.	Ruled based matchmaker	53
4.3.1.	EasyTV personalization ontology.....	53
4.4.	Rules	57
4.4.1.	Suggestion rules	57
4.4.2.	Conditional preferences evaluation rules	61

4.4.3.	Service substitution rules	65
4.5.	Statistical matchmaker	67
4.6.	Hybrid matchmaker	67
4.6.1.	Hybrid matchmaker API	68
5.	Indicative Use Cases.....	72
5.1.	A user with medium vision.....	72
5.2.	A profile with conditional preference	73
5.3.	Device control suggestions	75
5.4.	Content adaptation case	76
6.	Conclusion and future work.....	77
7.	References	78

List of Figures

Figure 1 User model json schema.....	18
Figure 2 User profile example in JSON format.....	19
Figure 3 CS app registration screen.....	23
Figure 4 CS app confirmation screen.....	23
Figure 5 CS app login screen.....	23
Figure 6 CS app main screen.....	23
Figure 7 Editor main screen	24
Figure 8 Editor creating new profile screen	24
Figure 9 Editor removing profile confirmation message.....	24
Figure 10 Editor generic preferences	25
Figure 11 Editor application specific preferences	25
Figure 12 Editor conditional preferences.....	25
Figure 13 Editor new conditional preference screen.....	26
Figure 14 Conditional preference	26
Figure 15 Editor text size preview	26
Figure 16 Editor TTS preview.....	26
Figure 17 Editor subtitle preview	27
Figure 18 Editor audio content preview	27
Figure 19 Editor cursor preview screen.....	27
Figure 20 Editor tests confirmation message	28
Figure 21 Color blindness test.....	28
Figure 22 Contrast sensitivity test	30
Figure 24 EasyTV icons	32
Figure 25 EasyTV system architectural overview [8]	34
Figure 26 Statistical matchmaker overview	67
Figure 27 User profile settings	72
Figure 28 User profile with medium vision.....	72
Figure 29 User profile with medium vision and suggestions	73
Figure 30 UI of magnification level 2	73
Figure 31 Contrast conditional preference 1.....	74
Figure 32 Contrast conditional preference 2.....	74
Figure 33 User profile with low contrast condition	74
Figure 34 User profile with no visual preferences.....	75
Figure 35 User profile with control suggestions.....	75
Figure 36 User profile for content adaptation	76

Figure 37 Flat match with personalized suggestions user profile..... 76

List of Tables

Table 1 Generic user preferences	13
Table 2 TTS application specific preferences	13
Table 3 CS app application specific preferences	14
Table 4 Screen reader application specific preferences.....	16
Table 5 Voice control application specific preferences.....	16
Table 6 Gesture application specific preferences	16
Table 7 Sign Avatar application specific preferences	17
Table 8 Contextual information.....	17
Table 9 Color blindness test answers	29
Table 10 Editor headers sizes	31
Table 11 EasyTV ontology	53
Table 12 EasyTV DataType properties	56
Table 13 Object properties of classes of EasyTV Ontology	57
Table 14 Preferences that affect blind people.....	58
Table 15 Preferences that affects low vision.....	58
Table 16 Preferences that affect deafness	59
Table 17 Preferences that affect hard of hearing	59
Table 18 Suggestion for magnification.....	59
Table 19 Apply magnification on all elements that have size	60
Table 20 Suggest screen reader & voice control	60
Table 21 Suggest sign language	61
Table 22 Equals preferences rule.....	61
Table 23 Not equals preferences rule.....	62
Table 24 Greater than preference rule.....	62
Table 25 Greater than equal preference rule	63
Table 26 Less than preference rule	63
Table 27 Less then equal preference rule	63
Table 28 Logical and rule	64
Table 29 Logical OR rule.....	64
Table 30 Logical Not rule.....	65
Table 31 Rule for recursive appliance	65
Table 32 Rule for declaring service substitutions.....	65
Table 33 Rule for ordering service substitutions	66

Executive Summary

This document corresponds to the deliverable D4.3 “*EasyTV user interface adaptation framework*” that is related to WP4. It describes the work done in T4.1 “*Adaptive menus and graphical interface using user models*”. The task is concerned with implementing an adaptive mechanism that automatically adapts the user interface and content to the user needs. Part of the task is to define a proper representation of the user in the EasyTV platform. A graphical interface tool that allows the user the easy editing of the user profile. Lastly, a personalization mechanism that given the user profile does the proper user interface and content adaptation and suggestions. The document consists of six different chapters:

Chapter 1 A brief introduction to the personalization process and its objectives and components.

Chapter 2 The user model in the EasyTV platform, its data types, format and structure.

Chapter 3 The user model editor design choices and functionalities.

Chapter 4 The hyper-personalization framework inner workings, how it achieves the user interface and content personalization.

Chapter 5 A set of indicative cases.

Chapter 6 Summarizes the conclusions obtained with this deliverable.

1. INTRODUCTION

In the digital age users of digital devices have different requirements for accessing information. These requirements might be based on individual preferences or due to a functional limitation. The EasyTV project focuses on the TV domain, and aims at making TV accessible for all users, including users with audio and visual functional limitation. As results, the goal is achievable by making the UI and audio-visual content accessible for all users.

In the core of EasyTV platform is the hyper-personalization framework that is responsible of the personalization process. Apart from the TV content personalization there is the UI personalization, which is concerned with selecting proper settings for EasyTV UI that better suits the user functional capabilities and preferences. The hyper-personalization framework is a recommendation system that follows a matchmaking approach. More specifically, a hybrid matchmaking approach that combines a rule-based and statistical matchmaking methods. The rule-based matchmaker follows a logical inference approach based on first order logic and implemented using semantic web technologies, while the statistical matchmaking follows a statistical analysis approach that make use of users' available data to achieve inference.

As with any personalization framework, users of EasyTV platforms are modelled by a profile that includes usage relevant information. In the case of EasyTV, the user model includes information regarding the user preferences for UI and audio-visual preferences. Users of the platform have access to their profiles using the user model editor that allows them to create, delete, edit, save and export their profiles.

This document describes an end-to-end overall view of all the components that constitute the EasyTV hyper-personalization framework.

2. EASYTV USER MODEL

The EasyTV user model is a representation of the user in the EasyTV platform. A user model contains the user needs and requirements for accessing UI and audio-visual content. In the user model, these requirements take the form of a set of preferences.

The EasyTV set of preferences are grouped into *general*, *application-specific* and *conditional* preferences. General preferences are settings common to more than one context or service. Application-specific preferences are settings associated with a specific EasyTV service. Lastly, conditional preferences are settings to be applied under specific conditions. We address analytically each one of these in the following sections.

2.1.1. Generic preferences

General preferences are preferences common to various settings and services. These preferences are based on [1], and classified into display, control and content. Preferences under display refer to how the user prefers to have the information displayed or presented. Control preferences describe how a user prefers to control the device. Finally, content preferences describe what enhanced or adapted content the user requires. The following table list the common preferences.

Table 1 Generic user preferences

Preference	Data Type	Value range	Short description	Reference
language	String	["en", "ca", "it", "el", "es"]	Audio language of the played content, Audio language of tts and screen reader, Closed caption language (subtitles, text detection, sound detection), Sign language, Graphical interfaces language	ISO/IEC 24751-2:2008
Audio volume	Numeric	range (0...100)	Audio volume level of the played content, and of tts and screen reader	ISO/IEC 24751-2:2008

2.1.2. Application specific preferences

Application specific preferences are preferences for configuring EasyTV available services. These preferences are related to the common preferences in that services that has multiple similar settings are added to the common preferences. A user profile mainly consists of these preferences. The following table lists the application specific preferences.

Table 2 TTS application specific preferences

Preference	Data Type	Value range	Short description	Reference
Text-to-speech rate	Integer	range (-10...+10)	Speech speed	-

		(default 0)		
Text-to-speech volume	Integer	0 - 100 (default 90)	Speech volume level	-
Text-to-speech language	String	["en", "ca", "it", "es"]	Language	-
Text-to-speech voice	String	-	The voice of the speaker	-
Text-to-speech quality	Integer	range (1...8)	1 (best quality, more storage) to 8 (worst quality, less storage).	-

Table 3 CS app application specific preferences

Preference	Data Type	Value range	Short description	Reference
Touch vibration	Boolean	[True, false]	Touch vibration	-
Text size	Integer	range (1...3)	UI text size	WCAG 2.0 ISO/IEC 24751-2:2008
Cursor size	Integer	range (1...3)	Cursor size	ISO/IEC 24751-2:2008
Cursor color	String	range (#000000...#FFFFFF)	Cursor font color	ISO/IEC 24751-2:2008
Zoom text	Boolean	[True, false]	Zoom text enable/disabled	-
Audio assistance	Boolean	[True, false]	Audio assistance	-

			based on TTS technology	
Language	String	["en", "ca", "it", "el", "es"]	The language of the GUI interface	-
Image magnification scale	Numeric	[1.5 - 3.5]	Image magnification scale	-
Text detection	Boolean	[True, false]	Text detection service	-
Face detection	Boolean	[True, false]	Face detection service	-
Audio description	Boolean	[True, false]	Audio description service	-
Sound detection	Boolean	[True, false]	Sound detection service	-
Character recognition	Boolean	[True, false]	Character recognition service	-
Audio equalizer bass	String	[-15, 15]	Equalizer bass band	-
Audio equalizer mid-range	String	[-15, 15]	Equalizer mid band range	-
Audio equalizer high end	String	[-15, 15]	Equalizer high band range	-
Audio volume	Integer	0 - 100	Audio volume of audio subtitle	-
Audio language	String	["en", "ca", "it", "el", "es"]	The language of the description	-
Audio subtitles	Boolean	[True, false]	Audio subtitles service	-
Subtitle language	String	["en", "ca", "it", "el", "es"]	Subtitle language	-

Subtitle font size	Integer	[1 – 50]	Subtitles font size	-
Subtitle font color	String	range (#000000...#FFFFFF)	Subtitle font color	-
Subtitle background color	String	range (#000000...#FFFFFF)	Subtitle background color	-
Gesture control	Boolean	[True, false]	Gesture control of the application	-

Table 4 Screen reader application specific preferences

Preference	Data Type	Value range	Short description	Reference
Screen reader	Boolean	[True, false]	Enable HbbTV screen reader	-

Table 5 Voice control application specific preferences

Preference	Data Type	Value range	Short description	Reference
Voice control	Boolean	[True, false]	Use voice command to control the application	-

Table 6 Gesture application specific preferences

Preference	Data Type	Value range	Short description	Reference
Gesture control	Boolean	[True, false]	User hand gesture to control of the application	-

Table 7 Sign Avatar application specific preferences

Preference	Data Type	Value range	Short description	Reference
Sign avatar	Boolean	[True, false]	Sign avatar	-
Sign avatar language	String	[“ca”, “it”, “el”, “es”]	Sign avatar language	-

2.1.3. Conditional preferences

Conditional preferences are a set of preferences that are paired with a set of conditions. When the set of conditions are true, the associated preferences are applied. The set of preferences that can be paired with a condition can be any preference from the union set of the generic (section 2.1.1) and application-specific (section 2.1.2) preferences. For defining the conditions, a set of operands is used which are of two types, logical and comparator operands. Logical operands are used to link conditions together and these are {*and*, *or*, *Not*} and the comparing operands are used in condition to set the comparing action to be executed and these are {*greater than*, *greater than equal*, *less than*, *less than equal*} as listed in section 2.2.

2.1.4. Contextual information

Context commonly refers to every piece of information used to characterize the situation of an entity. Such an entity can be a person, a place or any other arbitrary object, which is relevant for the interaction between the user or the application – including the application and the user themselves [2]. In our case, contextual information are information related to the user environment and these are factors that affect the user experience and interaction. The following table list the contextual information used.

Table 8 Contextual information

URL	Data Type	Range	Short description	Reference
Device	String	[“Mobile”, “tablet”, “PC”]	Device type, one of mobile, tablet or PC	
Brightness	Numeric	range (0.1...*)	The ambient light level (illumination) in lx	
Proximity	Integer	range (0...100)	Measures the proximity of an object in cm relative to the view screen of a device	

Location	String	-	The device's location, country code	
Time	String	-	Current time in the form HH:MM:SS	

2.1.5. Conditional preferences example

A user may want to apply some specific colors to font and background in some specific time. The way to realize that is to define a conditional preference that pair these preferences with a condition that check the time. In other words, to define the following: **When** *current_time* is bigger than 8:00 and less than 15:00 **then** set my default preferences of background color = "black" and font color = "yellow". The condition shown in Figure 2 User profile example in JSON format implements this condition.

2.2. User model implementation

We choose to represent EasyTV user model with JSON format. A JSON profile consists of a set of *Key: value* pairs separated by a comma. The profile main section is the **user_preferences** that consists of two sub sections, **default** and **conditional** as shown in Figure 1. The default sections holds the user default preferences (sections 2.1.1 & 2.1.2), which are preferences to be applied always. While the conditional sections holds the set of conditional preferences (sections 0), these are applicable only when the associated condition is fulfilled.

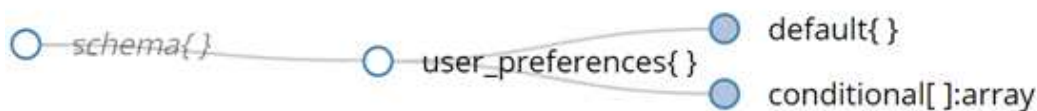


Figure 1 User model json schema

As mentioned JSON format, consist of a set of Key: Value pair. In the case of preferences, The key of each preference value is an indicative URLs that decodes the hierarchy of classes the preference belongs to. For example, font type preference is mapped under the following URL:

"http://registry.easytv.eu/common/display/screen/enhancement/font/type": "serif"

The path section of the URL indicates that the font type preference is actually a common preference that belongs to the display group and is related to screen enhancement. The classification classes used to categorize the generic preferences are extracted from ISO/IEC 24751-2:2008 definition. However, in the case of the application specific we use a set of newly defined classes, such as CS, accessibility etc., that indicates where the preferences belongs to. This classification will prove helpful in profile clustering and its hieratical clustering. Figure 2 shows a JSON formatted user profile.

```

{
  "user_preferences": {
    "default": {
      "preferences": {
        "http://registry.easytv.eu/gazeControl": false,
        "http://registry.easytv.eu/zoomTextField": false,
        "http://registry.easytv.eu/application/cs/eq/bass": 0,
        "http://registry.easytv.eu/application/cs/eq/mids": 0,
        "http://registry.easytv.eu/touchVibration": false,
        "http://registry.easytv.eu/application/cs/eq/highs": 0,
        "http://registry.easytv.eu/common/contrast": 100,
        "http://registry.easytv.eu/application/voiceControl": true,
        "http://registry.easytv.eu/application/cs/GUILanguage": "en",
        "http://registry.easytv.eu/application/cs/audioVolume": 50,
        "http://registry.easytv.eu/application/cs/audioLanguage": "en",
        "http://registry.easytv.eu/application/cs/textDetection": false,
        "http://registry.easytv.eu/application/cs/audioDescription": false,
        "http://registry.easytv.eu/application/cs/csSoundDetection": false,
        "http://registry.easytv.eu/application/cs/characterRecognition": false,
        "http://registry.easytv.eu/common/content/audio/volume": 90,
        "http://registry.easytv.eu/application/cs/audioAssistanceBasedOnTTS": false,
        "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 12,
        "http://registry.easytv.eu/common/display/screen/enhancement/font/type": "Roboto",
        "http://registry.easytv.eu/common/display/screen/enhancement/background": "#18d4dc",
        "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#b23b41"
      }
    },
    "conditional": [
      {
        "name": "conditional preference 1",
        "conditions": [
          {
            "type": "and",
            "operands": [
              {
                "type": "Le",
                "operands": [
                  "http://registry.easytv.eu/context/time",
                  "03:00"
                ]
              },
              {
                "type": "ge",
                "operands": [
                  "http://registry.easytv.eu/context/time",
                  "08:00"
                ]
              }
            ]
          }
        ]
      },
      {
        "name": "conditional preference 2",
        "conditions": [
          {
            "type": "and",
            "operands": [
              {
                "type": "Le",
                "operands": [
                  "http://registry.easytv.eu/context/time",
                  "03:00"
                ]
              },
              {
                "type": "ge",
                "operands": [
                  "http://registry.easytv.eu/context/time",
                  "08:00"
                ]
              }
            ]
          }
        ]
      }
    ],
    "preferences": {
      "http://registry.easytv.eu/common/display/screen/enhancement/background": "#ffffff",
      "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#000000"
    }
  }
}

```

Figure 2 User profile example in JSON format

Following is the JSON schema that describes the conditions description based on [3].

The user model conditional preferences JSON schema

```
{
```

```

"definitions": {
  "type_and_or": {
    "type": "object",
    "properties": {
      "type": {
        "enum": [ "and", "or" ]
      },
      "operands": {
        "type": "array",
        "minItems": 2,
        "items": {
          "anyOf": [
            { "$ref": "#/definitions/type_and_or" },
            { "$ref": "#/definitions/the_rest_types" },
            { "$ref": "#/definitions/type_not" }
          ]
        }
      }
    },
    "required": ["type", "operands"],
    "additionalProperties": false
  },
  "the_rest_types": {
    "type": "object",
    "properties": {
      "type": {
        "enum": [ "eq", "ne", "lt", "le", "gt", "ge", "ap" ]
      },
      "operands": {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "type": "string"
          },
          {}
        ]
      }
    },
    "required": ["type", "operands"],
    "additionalProperties": false
  },
  "type_not": {
    "type": "object",
    "properties": {
      "type": {
        "enum": [ "not" ]
      }
    }
  }
}

```

```

    },
    "operands": {
      "type": "array",
      "minItems": 1,
      "maxItems": 1,
      "items": {
        "anyOf": [
          { "$ref": "#/definitions/type_and_or" },
          { "$ref": "#/definitions/the_rest_types" },
          { "$ref": "#/definitions/type_not" }
        ]
      }
    },
  },
  "required": ["type", "operands"],
  "additionalProperties": false
},
"user_preferences": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "preferences": {
      "type": "object"
    },
    "conditions": {
      "type": "array",
      "items": {
        "anyOf": [
          { "$ref": "#/definitions/type_and_or" },
          { "$ref": "#/definitions/the_rest_types" },
          { "$ref": "#/definitions/type_not" }
        ]
      }
    }
  },
  "required": ["preferences"],
  "additionalProperties": false
},
"type": "object",
"properties": {
  "user_preferences": {
    "type": "object",
    "properties": {
      "default": {
        "$ref": "#/definitions/user_preferences"
      }
    }
  }
}

```

```
    },
    "conditional": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/user_preferences"
      }
    }
  },
  "additionalProperties": false
}
},
"additionalProperties": false
}
```

3. USER MODEL EDITOR

3.1. Overview

The User model editor is a GUI that allows users to manage their profiles, to configure and define their preferences. A user can create, edit, save, delete and export his profile. In addition, some tests has been integrated to help the user setup his/her the profile. Finally, the editor is integrated into the CS app to be easily accessible to users.

3.2. Registration and login

The personalization process requires from the user to be registered in EasyTV platform. The user can register from the registration screen by providing an email and a password Figure 3. After providing these information, the user management platform (section 4.2) sends a confirmation email to the user. The user must confirm his/her email by inserting the received confirmation code in the confirmation screen Figure 4. After these steps the user is able to login by providing his/her email and password Figure 5.

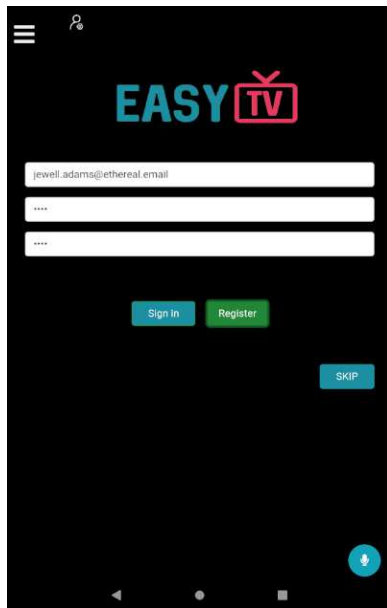


Figure 3 CS app registration screen

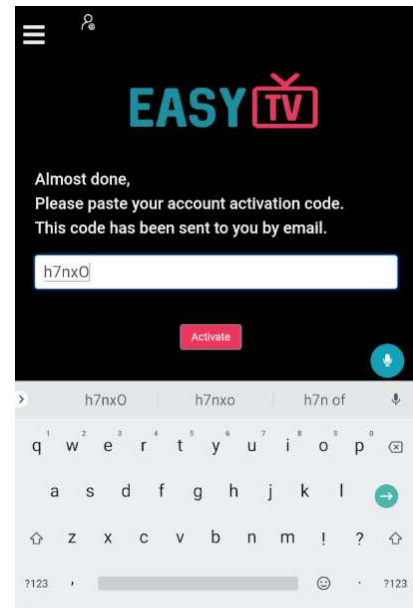


Figure 4 CS app confirmation screen

3.3. Access to the editor

The user can access the editor by selecting the humanoid icon ?? presented in two CS app screens, the login screen and the main screen.

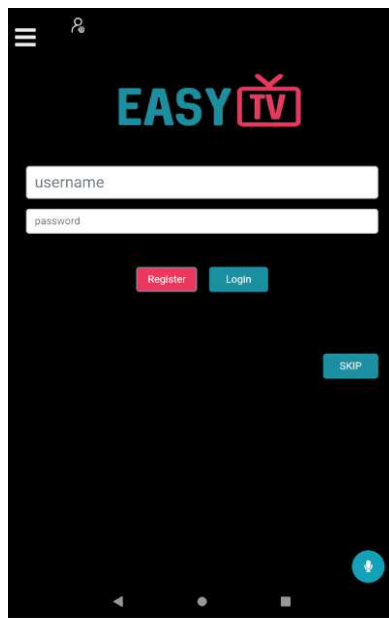


Figure 5 CS app login screen

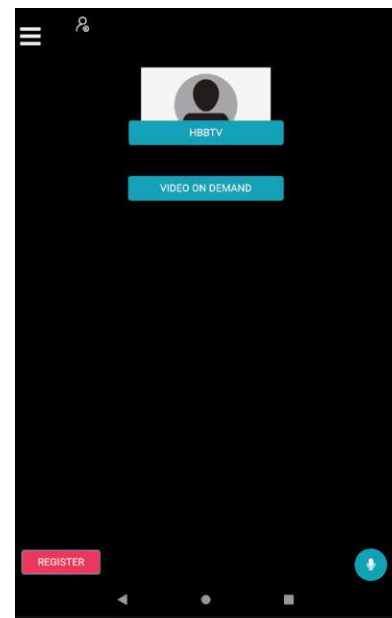


Figure 6 CS app main screen

From the editor main screen the user can create, activate, export and take initialization tests (section 3.8). To create a new profile, the user has to select the plus sign. To export a profile, the user has to select the folder sign. To take the initialization tests, the user has to select the humanoid sign. Lastly, to activate a profile the user has to select, from the profile lists, one of the profiles. The activated profile is used in the personalization process.

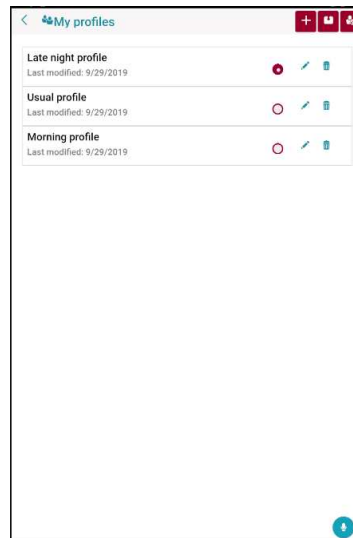


Figure 7 Editor main screen

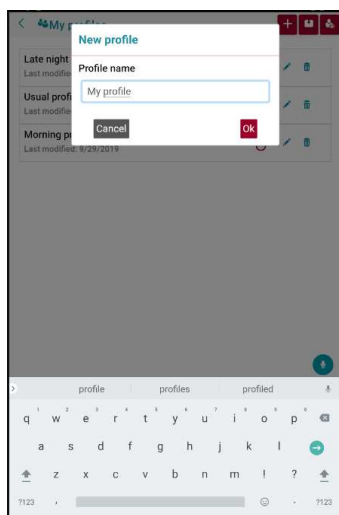


Figure 8 Editor creating new profile screen

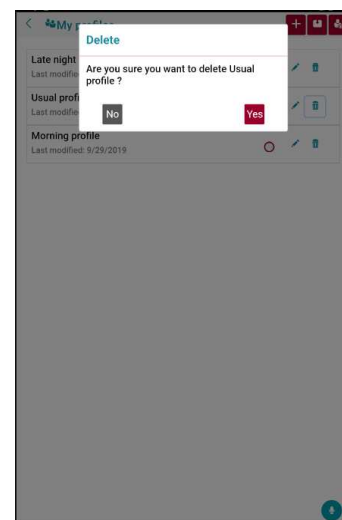


Figure 9 Editor removing profile confirmation message

3.4. Creating a profile

By selecting the plus sign presented on the top right corner of the editor's main screen, Figure 7, a user can create a profile. The user is requested to enter a profile name as shown in Figure 8.

3.5. Editing a profile

A user can activated any of his/her profile by choosing one of the profiles in the list , Figure 7. The activated profile is shown with red dot.

3.6. Removing a profile

From the list of presented profiles , Figure 7, a user can remove any of these by clicking the trash icon presented, a notification message is presented to confirm the action, Figure 9.

3.7. Initializing a profile

For creating or editing a profile, the user is presented with a screen that consists of a three main

tabs. The first tab is the generic preferences tab, Figure 10, which allow the user to initialize his/her common preferences (section 2.1.1). The next in row is the application-specific preferences tab, Figure 7, which allow the user to initialize his/her application-specific preferences (section 2.1.2). In both preferences tabs (common and application specific), preferences are grouped and presented in a set of drop down lists. This grouping of preferences is based on the classification explained in the (section 2.2). The last tab of the editor main screen is the conditional preferences tab, Figure 8. This tab lists the current declared conditions and allows the user to remove or edit them. In addition, by clicking the plus sign (+) the use can add new condition. Figure 9, shows the conditional preferences initialization screen, which consists of two main parts, the preferences part and the conditions part. The conditions part, list all conditions and their logical operations. The logical operations that connects conditions are {*and*, *or*, *Not*} and the comparing operations used to compare the condition value are {*greater than*, *greater than equal*, *less than*, *less than equal*} as listed in section 2.2. To define a condition two main filed are used, the *when* field which specifies what contextual parameter to be checked, and the *is* field which set the value of the condition to compare with. The preferences part, which is the actions part of the condition, can include any preferences that belongs to the union set of generic and application specific preferences. Under the action part of the screen, the user has to select the preference from a drop down list and to select the value to be applied.

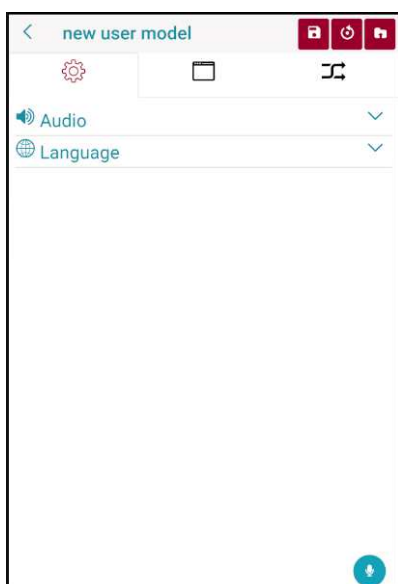


Figure 10 Editor generic preferences

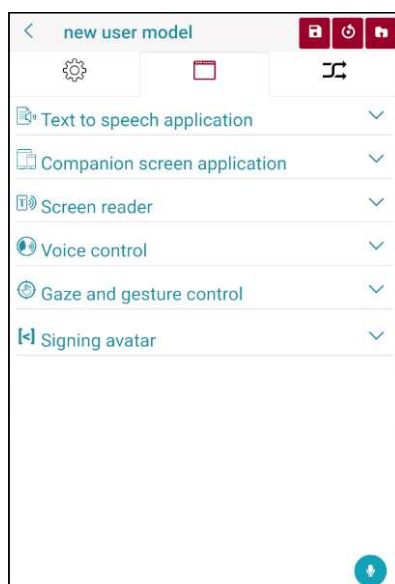


Figure 11 Editor application specific preferences

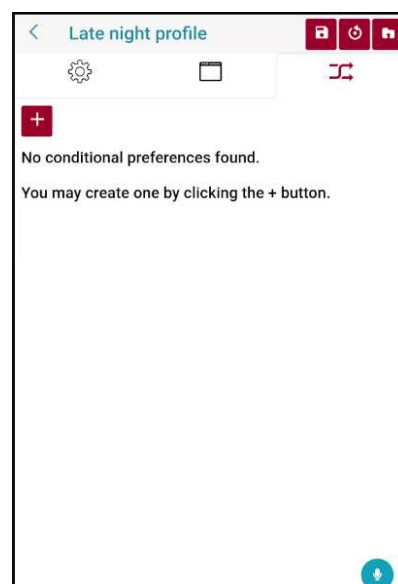


Figure 12 Editor conditional preferences

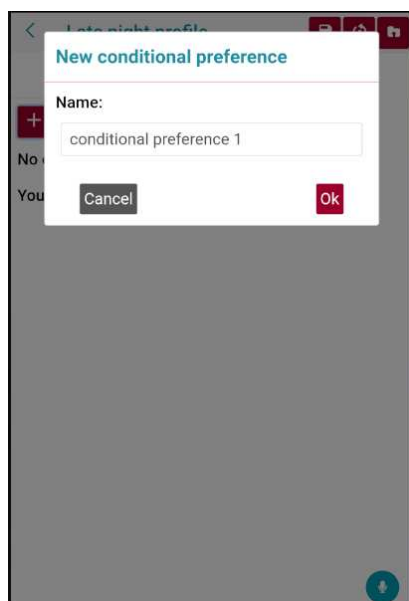


Figure 13 Editor new conditional preference screen

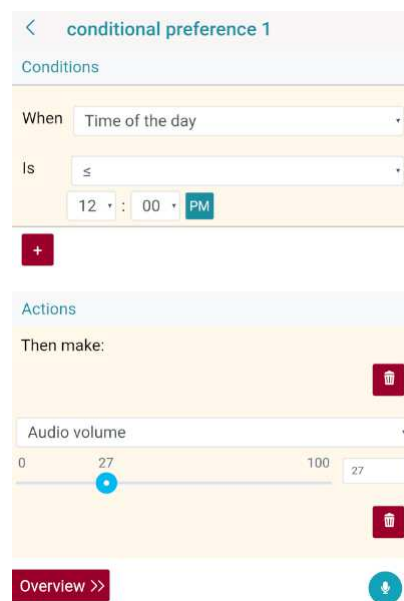


Figure 14 Conditional preference

The editor also has preview for some of the user profile preferences. The preview aims to help the user in distinguishing the side effect of the selected preference value. For instance, the text size preview shows the text size, color and background color as shown in Figure 15. The audio preview plays a sound to indicate the audio level. The subtitles preview shows subtitles size proportion to the screen. The text-to-speech preview interacts with the service and plays the output of a specific text language with the specific parameters. Lastly, the cursor preview shows the cursor size and color.

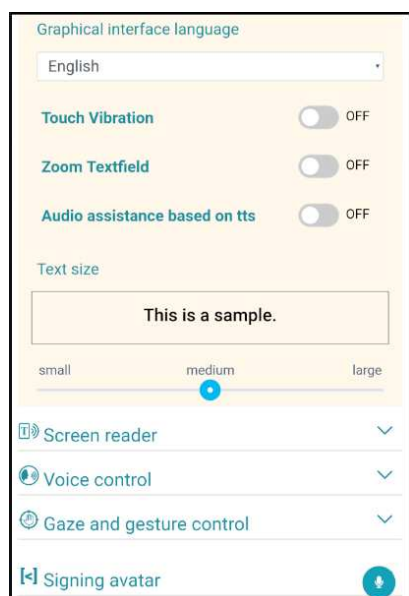


Figure 15 Editor text size preview

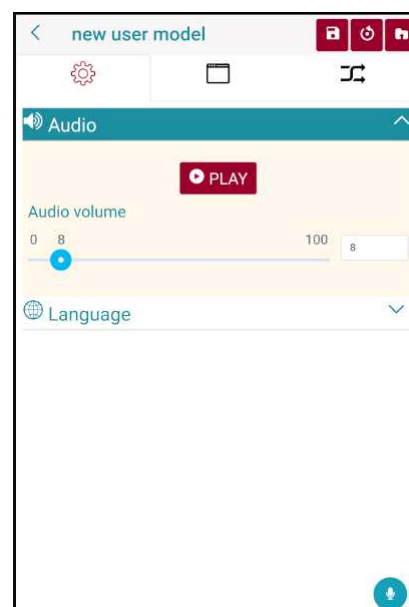


Figure 16 Editor TTS preview

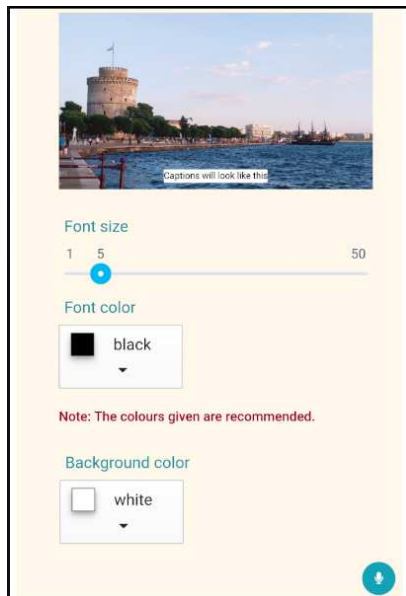


Figure 17 Editor subtitle preview

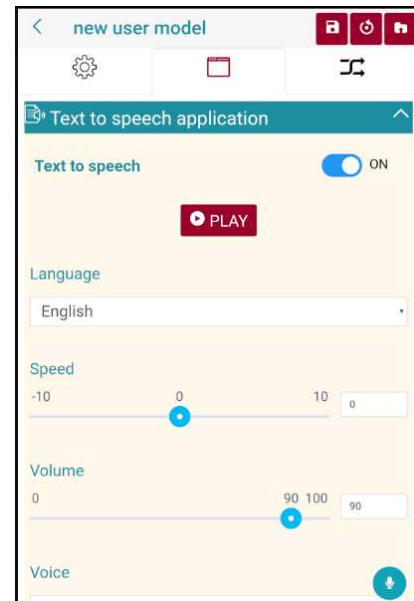


Figure 18 Editor audio content preview

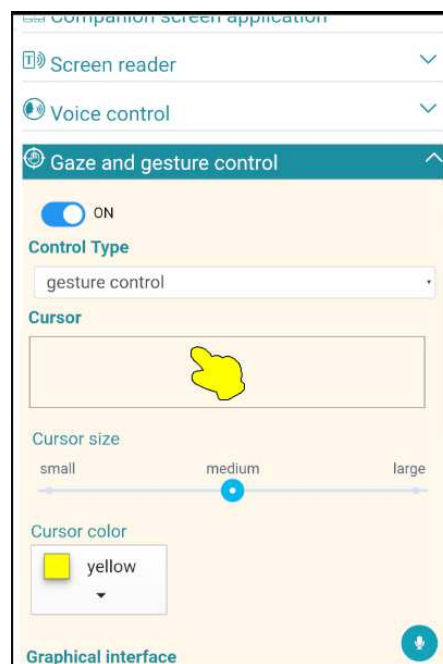


Figure 19 Editor cursor preview screen

3.8. Initialization tests

A second way for a user to setup his/her profile is by taking a set of simple tests. The tests are optional, and the user can opt out at any time easily by selecting the humanoid icon at the editor initial screen shown in Figure 5. On selecting to user the initialization tests, the user is prompt with the confirmation message shown in Figure 20. The tests are not only useful in minimizing the time and effort needed to initialize the profile but are also very useful in case the user does not know what are the proper preferences setup for his/her condition. The tests are a color blindness and contrast sensitivity tests. In addition, there is an auditor tests checks the user requirements for content frequency band.

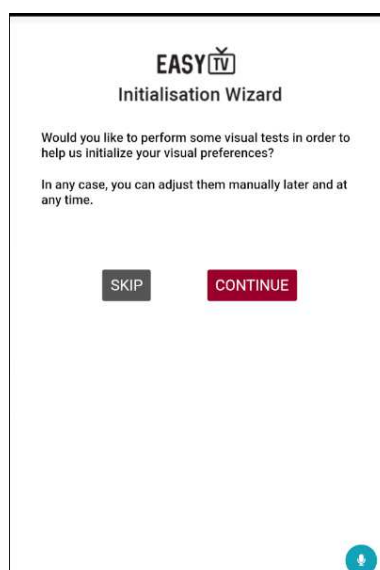


Figure 20 Editor tests confirmation message

3.8.1. Color blindness

The color blindness is a test based on Ishihara color-blindness test plates [4]. The tests consist of 24 plates that present the user with colored number plates and a set of choices. Based on the user set of answers the user color condition is selected. Table 9 lists the expected answers for each case.

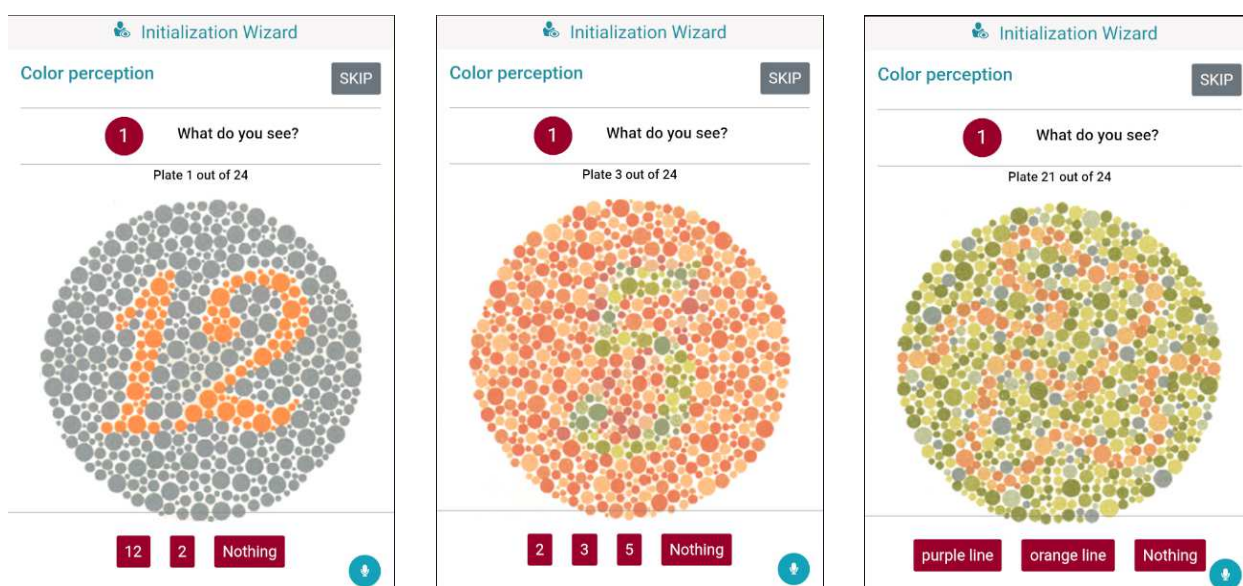


Figure 21 Color blindness test

The available choices for each plate are based on Ishihara book when that is possible. Ishihara book lacks any reference to Tritanopia color blindness condition. To find out what a user of Tritanopia sees we used a colorblind simulator¹ and we checked its validity by comparing Ishihara listed answers with those seen with the help of the simulator.

¹ <https://www.paciellgroup.com/resources/contrastanalyser/>

Table 9 Color blindness test answers

Plate	Normal person	Protanopia	Deuteranopia	Tritanopia	Total colour blindness
1	12	12	12	12	12
2	8	3	3	8	X
3	5	2	2	5	X
4	6	X	X	6	X
5	5	X	X	5	X
6	X	5	5	X	X
7	29	70	70	29	X
8	3	5	5	3	X
9	15	17	17	15	X
10	74	21	21	74	X
11	45	X	X	45	X
12	7	X	X	7	X
13	16	X	X	16	X
14	73	X	X	73	X
15	X	45	45	X	X
16	26	6	2	26	X
17	42	2	4	42	X
18	Both lines	Purple line	Red line	Both lines	X
19	X	Greenish line	Greenish line	X	X

20	Green line	X	X	Bluish line	X
21	Orange line	X	X	Purple line	X
22	Yellow-greenish	Green-blueish	Green-blueish	Blueish-purple	X
23	Orange and red	Yellow-greenish	Yellow-greenish	Purple line	X
24	Orange line	Green line	Green line	Pink line	Grey line

3.8.2. Contrast sensitivity

The contrast sensitivity test is based on [5]. In this test the user is presented a “C” character of progressively lighter color against a white background until the user cannot clearly see the character or the test ends.

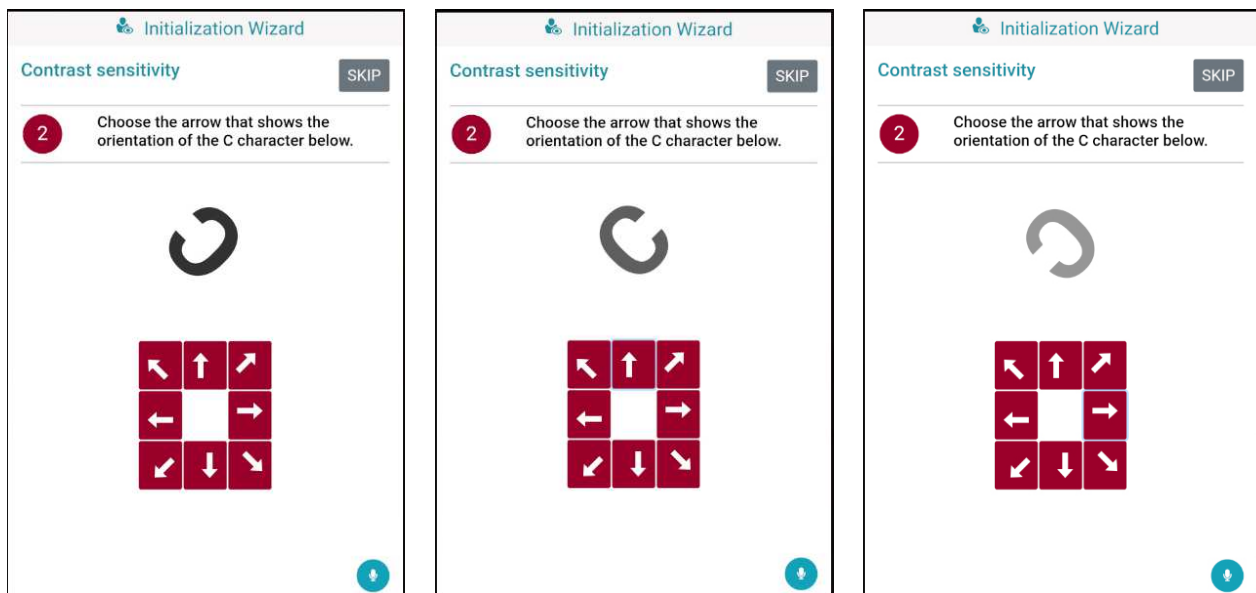


Figure 22 Contrast sensitivity test

Finally, the user model is saved into the database. Any time the user logs in the companion screen application, this user model is loaded from the server (or possibly from the device internal storage) and is used to define a set of appropriate system configurations and as an input to the matchmaker module.

3.9. Design approach

The user model editor graphical interface was designed to enhance and facilitate the access to the editor’s content for as many users as possible - target users, as mentioned in [6], Blind, Low Vision, Deaf, Low Deaf and Elderly, but also users with various related limitations or no limitations at all. Towards that goal along with consistency, we took an effort to:

- Design User Model Editor’s content in such a way that would be compatible, in terms of

components, arrangement, layout and general visualization, with the Companion Screen Application (CS App) GUI configuration settings and services;

- b. Follow WCAG 2.0 guidelines.² Regarding any aspect included in the guidelines, and the complexity of their possibly interconnected implementation, it is not possible to mention to what extent any of them is met, especially without an thorough use and evaluation by end users. Generally, however, there was an effort to make content simple, perceivable, and understandable. Specifically, the focus was on the “Guideline 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background”. Some points of the guidelines were already met up to a point through the CS App GUI configuration settings and services (Touch Vibration, Zoom text field, Text size, Audio Assistance using TTS Technologies). So here we focus on GUI’s layout, colour and font choices, as well as the combination of text and icons.

Given the fact that UI customization settings defined by the design of CS App, (Touch Vibration, Zoom text field, Text size, Audio Assistance using TTS Technologies) also apply to the User model editor, our focus was on using components as much compatible as possible with those settings. Regarding the structure/hierarchy of content and the general layout, we ended up with five main screens:

1. Initialization Wizard test
2. Home page of profile list
 - 2.1. Profile Editor > General TAB
 - 2.2. Profile Editor > App Specific TAB
 - 2.3. Profile Editor > Conditional Preferences TAB

3.9.1. Font type

WCAG 2.0 guidelines do not suggest specific fonts³⁴. There are, however, some general tips we addressed to choose one font for the editor’s default interface⁵.

We chose Roboto, as a font without "tails" very similar to the simplicity of Arial. It is a sans serif font. It also takes up a lesser amount of width space than Arial does, which is more suitable for small screens, in order not to break text in multiple lines when showing the same content in a smaller display. It is the default font both for headings and for body text..

3.9.2. Font size

The root size of the font chosen for the editor interface was 16px (12pt). That is the smallest size used in the interface, to ease the readability.

Table 10 Editor headers sizes

SIZE	Pixel	Points	Rem
Root size	16	12	1
h6	16	12	1

² <https://www.w3.org/TR/WCAG20/>

³ <https://www.w3.org/WAI/GL/low-vision-a11y-tf/wiki/Font>

⁴ <http://w3c.github.io/low-vision-a11y-tf/requirements.html#font>

⁵ <https://www.boia.org/blog/best-fonts-to-use-for-website-accessibility>

h5	20	15	1.25
h4	24	18	1.5
h3	28	21	1.75
h2	32	24	2
h1	40	30	2.5

The default sizes used are h6, h5, h4 for the titles of the categories, the settings titles, the settings sub-titles.

3.9.3. Font style

We also avoided *italic* style of font, as it may reduce readability for some people⁶. Instead, we used different sizes and weight (bold) to indicate essential differences of text.

3.9.4. Icons

We combined icon with text, as much as possible, in essential information such as the names of applications and the general setting categories. Some of the icons used, were defined in [7]

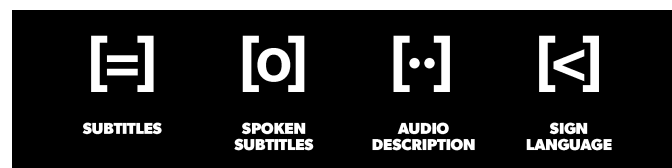


Figure 23 EasyTV icons⁷

Some others were based on icons used in EasyTV, according to “EasyTV UPM CS app_v0.1”. For the rest of the settings, we designed icons to be descriptive and simple. There was an effort to minimize textual information, by using short messages and notifications, with simple words, towards perceivable and understandable text.

3.9.5. Colours and contrast

The editor’s text colours, and background colours have been chosen to have contrast level of 4,8 or higher, thus WCAG 2 AAA Compliant (18pt+) or better. This applied in every heading on background and every text or icon on background of buttons. More specifically, only in one case the contrast level is at 4,8 thus WCAG 2 AAA Compliant (18pt+) - every other combination is of 7 contrast level or higher thus WCAG 2 AAA Compliant. We used this tool⁸ to check the contrast levels.

⁶ <http://w3c.github.io/low-vision-a11y-tf/requirements.html#font>

⁷ <https://www.dr.dk/om-dr/about-dr/smart-icons-design-common-european-standardization>

⁸ https://snook.ca/technical/colour_contrast/colour.html#fg=065A63,bg=FFFFFF

3.9.6. Layout and Components choices

3.9.6.1 Screens and Tabs

There was an effort to minimize the need to back and forth between pages. At the same time, it would be ideal to keep all content in the pages without the need of scrolling down, in order to assist text-to-speech related services, and the general flow and use. But once the application is to run in a variety of ratios and sizes of screens (various types of phones and tablets) it was not possible to totally eliminate the need for scroll down.

To achieve an intermediate state we used “accordion” component for folding-unfolding of content per application or general setting category. When one is unfolded, all others are folded.

This enabled us to keep the main screens (tabs) of settings configuration to only three, with the forth screen of the editor being the home page with the profile list. All the general settings in one tab and all the application specific settings in another, directly selectable, without the need for a plus screen for every category or sub-category in order to control individual settings. In the fifth screen of the Initialization Wizard Tests, every plate-test question is one-to-one to screen view with no need for scroll down. This was the only reasonable choice for the Wizard tests.

3.9.6.2 Previews and overview

We added a “preview” feature, which simulates the result of the user’s choices, encouraging him/her to instinctively adjust some visual and auditory preferences.

3.9.6.3 Error avoidance and general adaptability

The Initialization Tests support an automated customization of settings related to colour and colour contrast properties. The results cannot be considered as absolute or right. The preview features of colour and contrast related preferences, are included exactly to counteract errors or deviations and work along with the tests as a support to user to always check in real time the result of the settings he/she has chosen

4. HYPER-PERSONALIZATION FRAMEWORK

4.1. Architectural overview

This section describes the EasyTV system architecture with special focus on the hyper-personalization framework. The architecture of the EasyTV platform is divided into three blocks:

- **Broadcaster premises / Content Owner** – this block englobes the main workflows of the broadcaster or a content owner related to the management, storage, broadcast and publication of audiovisual contents.
- **EasyTV platform** – within this block several modules are grouped in service components that will be defined in the next sections.
- **Consumer platform** – end-users will consume the contents with accessible services through multi-platform devices like smartphones, desktops or SmartTV, interacting with their devices through improved accessible interfaces that will ease the access and consumption.

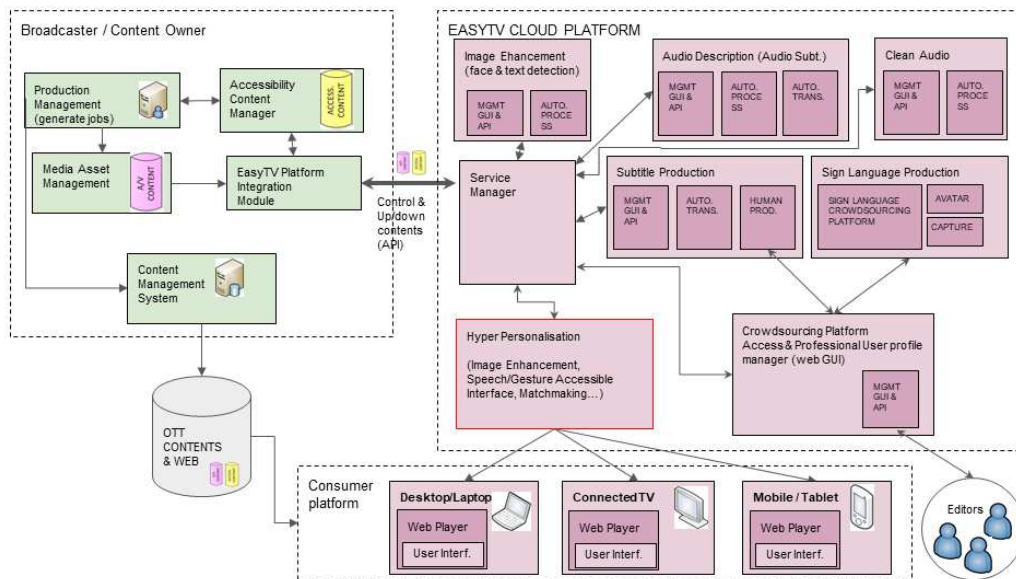


Figure 24 EasyTV system architectural overview [8]

The place of the Hyper-personalization module inside the EasyTV platform and its interaction with other EasyTV components is shown in Figure 24. The hyper-personalization module consists of the following components:

- **Hyper-Personalization:**
 - **User model management framework:** an endpoint for the login/registration process with an authentication mechanism and a database for storing users' profiles.
 - **Hybrid matchmaker:** The EasyTV hybrid matchmaker performs matchmaking between user needs and preferences defined in user profiles, device capabilities, accessibility features specifications and DASH streaming services specifications. The hybrid matchmaker consists of the following two sub-components:
 1. **Rule-based matchmaker:** The rule-based matchmaker performs matchmaking on the content and metadata stored in the EasyTV ontologies by applying semantic rules.
 2. **Statistical matchmaker:** The statistical matchmaker aims at improving the accuracy of the matchmaking results provided by the rule-based matchmaker by supporting self-adaptive and tailored services, which can learn from users' actions. Statistical methods that take into account not only the history of actions of the specific user, but also previous corresponding interactions of other users are applied for this purpose.

4.2. User model management framework

The companion screen application serves as the entry point to the easy TV system. First, the user creates a new account, then logs in, and creates a user model. This is done either by actively providing information relevant to his profile (by taking a series of optional tests as described in the earlier sections of this document) or by letting the matchmaker derive the user model automatically. Every time the user logs in, the matchmaker is called, and the user model is extracted from the database in order for all the user interface adaptations to be applied. In order to facilitate the distribution of the user model files between the different easy TV modules and to provide support for the login and user registration functionalities an application server based on nodeJS and a relational

database management system is used.

4.2.1. Login and registration

In order to use the easy TV system, the user has to create an account. At this point, we opted to enable users to create new easy TV accounts by simply providing and verifying their email. The user's email serves as an identifier much like a username. After the email provided has been verified, (the user has to 'activate' their account by providing the activation code emailed to them) a new user record is added to the database. Our server also provides support for resetting an account in case the user has forgotten the password. This procedure is similar to the registration process and uses an activation code emailed to the user. Every user is associated with a user model and thus, upon successful login their user model is accessed and provided to the matchmaker.

When a user logs into the system successfully, the server signs a json web token using the server's secret and returns it in a 200 code (success) response. Every call the user makes after the login must include this token in the authorization header of the http request. The server passes all requests from an authentication middleware that checks the validity of the bearer token provided by the client before proceeding.

4.2.2. Technologies used

The companion screen's application server is based on the express web framework for Node.js and a relational database (our choice at this point has been the MySQL server). The express web framework is a minimalist yet powerful web application framework that enables the use of JavaScript and nodeJS. It is a popular framework for writing web services. Essentially, it provides all the necessary features we need from an application server and enables us to access all the powerful features of the nodeJS framework and the npm (Node.js package manager). Regarding the authentication mechanism used to access the services exposed by the server's API, we currently use the json web tokens to sign and verify claims between the server and the client. We also use the Sequelize ORM (object-relational mapping. Sequelize⁹ is a promise based Node.js ORM for Postgres, MySQL, MariaDB, SQLite and Microsoft SQL Server.

4.2.3. Application specific interface

4.2.3.1 Register

URL		https://hplogin-api.easytv.eng.it/public/register	
Description		The first step to create a new user account. This service sends an account activation code to the user via email.	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments

⁹ <https://sequelize.org/>

email	String	256 chars	The user's email address
password	String	256 chars	The user's email password
password2	String	256 chars	The user's email password confirmation. 'password' and 'password2' should match.
Output			
Format		JSON	
Name	Type	Value space	Comments
msg	String	256 chars	A brief informative message
err	String	256 chars	A brief description of the error that occurred.
url	String	256 chars	Used for testing at this point and will be omitted in a later stage. This url gives us a link to the ethereal (fake) email account that we use for testing / development.
Service call example			
POST /public/register HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 75 Connection: keep-alive cache-control: no-cache <pre>{ "email": "a@etv.gr", "password": "password", "password2": "password" }</pre>			
Output			
<pre>{ "msg": "Use this link to access the Ethereal (fake, for testing reasons) SMTP mail server", "url":</pre>			

```
"https://ethereal.email/message/XQIMntwpOWUvWKMGXQJBigix6GVxf1jHAAAAEBUkegdNk6L
.pA7MAU7MFWs"
}
```

4.2.3.2 Activate account

URL		https://hplogin-api.easytv.eng.it/public/activateAccount	
Description		Activates a user account. The token query parameter holds the activation code.	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments
email	String	256 chars	The user's email address
Query parameters			
Name	Type	Value space	Comments
token	String	256 chars	This is the account activation token sent to the user's email address.
Output			
Format			json
Name	Type	Value space	Comments
mgs	String	256 chars	A brief informative text
err	String	256 chars	A description of the error.
Service call example			
POST /public/activateAccount?token=\$2a\$10\$wG/bDYWTYdZcJs85NkBFoegszqK0bZ7CRYKR5lhULTxJm/reQTqVS HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json			

Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 24 Connection: keep-alive cache-control: no-cache { "email": "b@etv.gr" }
Output
{ "msg": "Account activated" }

4.2.3.3 Login

URL		https://hplogin-api.easytv.eng.it/public/login	
Description		Login. Returns a bearer token	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments
email	String	256 chars	The user's email address
password	String	256 chars	The user's password in plaintext.
Output			
Format			json
Name	Type	Value space	Comments
mgs	String	256 chars	A brief informative text

err	String	256 chars	A description of the error.
token	String	256 chars	The bearer token
user	JSON	-	user data returned from the database (excluding the fields related to the password, password reset and the account activation)
Service call example			
<pre> POST /public/login HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 49 Connection: keep-alive cache-control: no-cache { "email": "b@etv.gr", "password": "password" } </pre>			
Output			
<pre> { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTM5MTU2MDQzMTEwZXhwIjoxNTYwNjkxMTcyfQ.EiUmjfQ3CMtu7BFJw-7ntdU6FWMycztgex_pEa6mtVQ", "user": { "id": 13, "email": "b@etv.gr", "createdAt": "2019-06-13T12:59:48.000Z", "updatedAt": "2019-06-13T13:00:10.000Z", "wizardResultId": null } } </pre>			

4.2.3.4 Initiate password reset

URL		https://hplogin-api.easytv.eng.it/public/initiatePasswordReset	
Description		Initialtes the password reset procedure and sends an email with the password reset token	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments
email	String	256 chars	The user's email address
Output			
Format			json
Name	Type	Value space	Comments
mgs	String	256 chars	A brief informative text
err	String	256 chars	A description of the error.
url	String	256 chars	Used for testing at this point and will be omitted in a later stage. This url gives us a link to the ethereal (fake) email account that we use for testing / development.
Service call example			
POST /public/initiatePasswordReset HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 24 Connection: keep-alive cache-control: no-cache			

<pre>{ "email": "b@etv.gr" }</pre>
Output
<pre>{ "msg": "Successfully sent password reset link to the user's email.", "url": "https://ethereal.email/message/XQIMntwpOWUvWKMGXQJOkQix6GVxf13IAAAAEik9HzVE6fT C1yFI39K33OU" }</pre>

4.2.3.5 Complete password reset

URL		https://hlogin-api.easytv.eng.it/public/completePasswordReset	
Description		Resets the user's password	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments
newPassword	String	256 chars	The user's new password in plaintext.
Query parameters			
Name	Type	Value space	Comments
token	String	256 chars	The token query parameter holds the password reset code.
Output			
Format			json
Name	Type	Value space	Comments
mgs	String	256 chars	A brief informative text
err	String	256 chars	A description of the error.
res	json	-	user data returned from the database (excluding the fields related to the

			password, password reset and the account activation)
Service call example			
<p>POST</p> <p>/public/completePasswordReset?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTIsIm1hdCI6MTU2MDQzMjY3MCwiZXhwIjoxNTYwNDQzNDcwfQ.xFBk2RnQuZie0KkH6KxAAvdK2Wpn878ee46K_UxXOmM HTTP/1.1</p> <p>Host: 127.0.0.1:2017</p> <p>Content-Type: application/json</p> <p>Accept: */*</p> <p>Cache-Control: no-cache</p> <p>Host: 127.0.0.1:2017</p> <p>accept-encoding: gzip, deflate</p> <p>content-length: 39</p> <p>Connection: keep-alive</p> <p>cache-control: no-cache</p> <pre>{ "newPassword": "someOtherPassword" }</pre>			
Output			
<pre>{ "msg": "Successfully reset the password.", "res": { "id": 12, "email": "a@etv.gr", "createdAt": "2019-06-13T12:28:57.000Z", "updatedAt": "2019-06-13T13:32:13.287Z", "wizardResultId": null } }</pre>			

4.2.3.6 Save user model

URL	https://hlogin-api.easytv.eng.it/private/saveUserModel
Description	Saves a user model to the database.
HTTP Method	POST

Content type		application/json	
Authorization		Bearer token	
Body			
Format		JSON	
Name	Type	Value space	Comments
name	String	256 chars	The user model's name
userModel	Json	-	The json formatted user model data.
Output			
Format			json
Name	Type	Value space	Comments
userModel	Json	256 chars	The user model that has been persisted to the database.
err	String	256 chars	A description of the error.
res	json	-	user data returned from the database (excluding the fields related to the password, password reset and the account activation)
Service call example			
<pre>POST /private/saveUserModel HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NSwiaWF0IjoxNTYwNDIwMzQ2LCJleHAiOiJlE1NjA2Nzk1NDZ9.tK866io086dJ2mEa4mE-vW17wbgiduEfNbJSVudzbyM Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 3074 Connection: keep-alive cache-control: no-cache {</pre>			

<pre>"name": "myUserModel", "userModel": { } }</pre>
Output
<pre>{ "userModel": { "id": 7, "userModel": { }, "userId": 5, "name": "myUserModel", "updatedAt": "2019-06-13T13:41:26.540Z", "createdAt": "2019-06-13T13:41:26.540Z" } }</pre>

4.2.3.7 Get user model

URL		https://hplogin-api.easytv.eng.it/private/getUserModel	
Description		Retrieves a user model from the database.	
HTTP Method		GET	
Content type		application/x-www-form-urlencoded	
Authorization		Bearer token	
Query parameters			
Name	Type	Value space	Comments
name	String	256 chars	The user model's name
Output			
Format			json
Name	Type	Value space	Comments
userModel	Json	256 chars	The user model.
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			

GET /private/getUserModel?name=myUserModel HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/x-www-form-urlencoded Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTMsImhhdCI6MTU2MDQ5MzEwOCwiZXhwljoxNTYwNzUyMzA4fQ.uHz5ChvFZ10FjNvJmh5DIOi2Ct1mtf8QZMaUi5Ggw6w Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate Connection: keep-alive cache-control: no-cache
Output
<pre>{ "userModel": { "id": 8, "userModel": {}, "name": "myUserModel", "createdAt": "2019-06-14T06:19:07.000Z", "updatedAt": "2019-06-14T06:19:07.000Z", "userId": 13 } }</pre>

4.2.3.8 Edit user model

URL	https://hplogin-api.easytv.eng.it/private/editUserModel
Description	Replaces the user model with the one provided in the body argumants.
HTTP Method	POST
Content type	application/json
Authorization	Bearer token
Body	
Format	JSON

Name	Type	Value space	Comments
name	String	256 chars	The name of the user model to be edited/
userModel	Json	-	The new user model to replace the previous one.
Output			
Format			json
Name	Type	Value space	Comments
userModel	Json	256 chars	The user model.
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			
POST /private/editUserModel HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTMsImIhdCI6MTU2MDQ5MzEwOCwiZXhwljoxNTYwNzUyMzA4fQ.uHz5ChvFZ10FjNvJmh5DIOi2Ct1mtf8QZMaUi5Ggw6w Accept: /* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 3069 Connection: keep-alive cache-control: no-cache { "name": "myUserModel", "userModel": {} }			
Output			
{ "userModel": {			

```

    "id": 8,
    "userModel": { },
    "name": "myUserModel",
    "createdAt": "2019-06-14T06:19:07.000Z",
    "updatedAt": "2019-06-14T06:27:55.544Z",
    "userId": 13
  }
}

```

4.2.3.9 Delete user model

URL		https://hplogin-api.easytv.eng.it/private/deleteUserModel	
Description		Deletes a user model	
HTTP Method		DELETE	
Content type		application/json	
Authorization		Bearer token	
Query parameters			
Name	Type	Value space	Comments
name	String	256 chars	The user model's name
Output			
Format			json
Name	Type	Value space	Comments
userModel	Json	-	The user model that has been deleted.
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			
DELETE /private/deleteUserModel?name=myUserModel HTTP/1.1			
Host: 127.0.0.1:2017			
Content-Type: application/json			
Authorization: <div>Bearer</div> eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTMsImIhdCI6MTU2MDQ5MzEwOCwiZXhwljoxNTYwNzUyMzA4fQ.uHz5ChvFZ10FjNvJmh5DIOi2Ct1mtf8QZMaUi5Ggw6w			
Accept: */*			

Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: Connection: keep-alive cache-control: no-cache
Output
<pre>{ "userModel": { "id": 8, "userModel": { }, "name": "myUserModel", "createdAt": "2019-06-14T06:19:07.000Z", "updatedAt": "2019-06-14T06:27:55.544Z", "userId": 13 } }</pre>

4.2.3.10 Get all user models names

URL		https://hplogin-api.easytv.eng.it/private/getAllUserModelNames	
Description		This is to be used only by the user model editor. Retrieves information about all the user model the user possesses.	
HTTP Method		GET	
Content type		application/json	
Authorization		Bearer token	
Output			
Format			json
Name	Type	Value space	Comments
userModel	Json	-	An array holding information about all the user models of the user who owns the bearer token.
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message

Service call example
<p>GET /private/getAllUserModelNames HTTP/1.1</p> <p>Host: 127.0.0.1:2017</p> <p>Content-Type: application/json</p> <p>Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTMsImIhdCI6MTU2MDQ5MzEwOCwiZXhwljoxNTYwNzUyMzA4fQ.uHz5ChvFZ10FjNvJmh5DIOi2Ct1mtf8QZMaUi5Ggw6w</p> <p>Accept: */*</p> <p>Cache-Control: no-cache</p> <p>Host: 127.0.0.1:2017</p> <p>accept-encoding: gzip, deflate</p> <p>Connection: keep-alive</p> <p>cache-control: no-cache</p>
Output
<pre>{ "userModel": [{ "name": "myUserModel", "createdAt": "2019-06-14T06:47:13.000Z", "updatedAt": "2019-06-14T06:47:13.000Z" }] }</pre>

4.2.3.11 Set wizard results

URL		https://hplogin-api.easytv.eng.it/private/setWizardResult	
Description		This is to be used only by the user model editor	
HTTP Method		POST	
Content type		application/json	
Authorization		Bearer token	
Body			
Format		JSON	
Name	Type	Value space	Comments

colorset	Integer	-	The colorset code
volume	Integer	-	The volume
fontsize	Integer	-	The font size
Output			
Format			json
Name	Type	Value space	Comments
res	Json	-	The wizardResult record
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			
POST /private/setWizardResult HTTP/1.1 Host: 127.0.0.1:2017 Content-Type: application/json Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NSwiaWF0IjoxNTYwNDIwNzAwLCJleHAiOiJlE1NjA2Nzk5MDB9.Ums314lozN--OQ2Bax_3J92E1oSza2QbVcx30OvapE Accept: */* Cache-Control: no-cache Host: 127.0.0.1:2017 accept-encoding: gzip, deflate content-length: 50 Connection: keep-alive cache-control: no-cache { "colorset": 1, "volume": 80, "fontsize": 24 }			
Output			
{ "msg": "success", "res": {			

```

    "id": 5,
    "email": "c@a.gr",
    "createdAt": "2019-06-13T09:09:37.000Z",
    "updatedAt": "2019-06-14T06:53:47.440Z",
    "wizardResultId": 1
  }
}

```

4.2.3.12 Get wizard results

URL		https://hplogin-api.easytv.eng.it/private/getWizardResult	
Description		This is to be used only by the user model editor	
HTTP Method		GET	
Content type		-	
Authorization		Bearer token	
Output			
Format			json
Name	Type	Value space	Comments
wizardResult	Json	-	The wizardResult record
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			
GET /private/getWizardResult HTTP/1.1			
Host: 127.0.0.1:2017			
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NSwiaWF0IjoxNTYwNDIwNzAwLCJleHAiOiJlE1NjA2Nzk5MDB9.Ums314lozN--OQ2Bax_3J92E1oSza2QbVcxF30OvapE			
Accept: */*			
Cache-Control: no-cache			
Host: 127.0.0.1:2017			
accept-encoding: gzip, deflate			
Connection: keep-alive			
cache-control: no-cache			

Output
<pre>{ "wizardResult": { "id": 1, "colorset": 1, "fontsize": 24, "volume": 80, "createdAt": "2019-06-14T06:53:47.000Z", "updatedAt": "2019-06-14T06:53:47.000Z" } }</pre>

4.2.3.13 Delete wizard results

URL		https://hplogin-api.easytv.eng.it/private/deleteWizardResult	
Description		This is to be used only by the user model editor	
HTTP Method		DELETE	
Content type		-	
Authorization		Bearer token	
Output			
Format			json
Name	Type	Value space	Comments
err	String	256 chars	A description of the error.
msg	String	256 chars	A brief informative message
Service call example			
DELETE /private/deleteWizardResult HTTP/1.1			
Host: 127.0.0.1:2017			
Authorization:			Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NSwiaWF0IjoxNTYwNDIwNzAwLCJleHAiOiJlE1NjA2Nzk5MDB9.Ums314lozN--OQ2Bax_3J92E1oSza2QbVcxF30OvapE			
Accept: */*			
Cache-Control: no-cache			
Host: 127.0.0.1:2017			
accept-encoding: gzip, deflate			

content-length: Connection: keep-alive cache-control: no-cache
Output
<pre>{ "msg": "success" }</pre>

4.3. Ruled based matchmaker

The rule-based matchmaker suggests UI adaptation based on the user preferences only. From a technical point of view, the rule-based matchmaker uses Apache Jena,¹⁰ a semantic web framework. Semantic web follows a graph data model that represents data in graph like structure. Data under this model is represented as triple statements of the form (*subject predicate object*). The statements' subjects and objects are the graph vertices and the edges or links are the predicates. This set of statements is called the knowledge base (KB); new triple statements are inferred and added to the KB the help of rules. The set of subjects, objects and predicates define the concepts of a domain problem, and these are declared by the domain ontology.

4.3.1. EasyTV personalization ontology

EasyTV personalization ontology defines concepts related to the user, these are declared as a set of classes, their correlation and attributes. The user is represented by an instance of class User and is associated with an instance of class Preference. As with the user model, the preference instance can be associated with default and conditional preferences instances. Other classes represent the user context and the content accessibility services available. The user context is associated with the current time, location, proximity and brightness.

Two content types are the main concerns of EasyTV platform, audio-visual and UI, so the class content has two such subclasses. Each one of these can be associated with other objects that actually describe its structure. For instance, an audio-visual content instance contains video and audio. One of EasyTV services is a face, text detection thus we define that video instances may contain faces and texts. UI content, on the other hand, can be constituted by different set of objects such as text, colour etc.

With this ontology will make statements about the structure of content. The changes that services do on different content and the side effects of different setting on the UI.

Table 11 EasyTV ontology

Class Name	Sub class	Description
User		A user of the EayTV platform

¹⁰ <https://jena.apache.org/>

UserContext		A user context
UserPreference	<i>Super class for user preferences</i>	
	ConditionalPreferences	A conditional preference
	SuggestedPreferences	A suggested preference
Service	<i>A service of EasyTV platform</i>	
	AccessibilityService	A super class for accessibility services
AccessibilityService	<i>Super class of accessibility services</i>	
	ContentAccessibilityService	A super class for all content adaptation services
	UIAccessibilityService	A super class for all UI adaptation services
ContentAccessibilityService	<i>Super class of content adaptation services</i>	
	EnhancementContentAccessibilityService	A content adaptation service that enhance the audiovisual content
	TransformalContentAccessibilityService	A content adaptation service that transforms one content type to another video => audio Audio => video
Content	<i>Super class of all contents</i>	
	AudioVisualContent	An audio-visual content
	UIContent	User interface content
ContentEnhancement	<i>Super class of all content adaptation actions</i>	
	Conversion	A conversion of content from one type to another
	Detection	Automatic detection
	Magnification	Magnification of content
Text	<i>Super class of text types of various content types</i>	
	ImageText	A text included in an audio-visual content
	SpokenText	A text that is read by audio subtitle service
	WrittenText	Any other type of text
Substitution		A class that represents a collection of service substitution.
Language	<i>A super class of all languages</i>	

	CA	Catalan
	ES	Spanish
	GR	Greek
	IT	Italian
Audio		Audio class
Face		A face contained in an audio-visual content and can be detected by a service
Sound		A sound contained in an audio-visual content and can be detected by a service
Video		Video class
Screen		Screen class that represent the whole display with its UI and Audio-visual content
Operand	<i>Conditional preference operand</i>	
	ComparatorOperand	An operand that compares two values
	LogicalOperand	An operand that compare binary inputs
ComparatorOperand	<i>Super class of all comparator operand</i>	
	EQ	Equal
	GT	Greater than
	LE	Less than
	LT	Less than
	NE	Not equal
LogicalOperand	<i>Super class of all logical operand</i>	
	DualOperand	An operand that has two inputs
	MonoOperand	An operand that has one input
DualOperand	<i>Super class of all dual operands</i>	
	AND	And binary operand
	OR	Or binary operand
MonoOperand	<i>Super class of all mono operands</i>	
	NOT	Not binary operand

Table 12 EasyTV DataType properties

Class Name	Datatype properties		
	Name	Description	Type
UserPreference	hasAudioLanguage	Audio language	String
	hasAudioVolume	Audio volume level	Integer
	hasBackground	Background colour	String
	hasFontColour	Font colour	String
	hasFontSize	Font size	Integer
	hasFontType	Font type	Integer
	hasSubtitleLanguage	Subtitle language	String
	hasSubtitleSize	Subtitile size	Integer
	hasSubtitileBackgroundColour	Subtitile background colour	Integer
	hasSubtitileForgroundColour	Subtitile forground colour (text colour)	String
	hasTTSSpeed	TTS speed	Integer
	hasTTSVolume	TTS volume level	Integer
	hasTTSLanguage	TTS language	String
	hasImageMagnification	Image magnification scale	Integer
	hasTextDetection	Text detection service	Boolean
	hasFaceDetection	Face detection service	Boolean
	hasAudioSubtitles	Audio subtitles service	Boolean
	hasAudioDescription	Audio description service	Integer
Content Accessibility service	does	The action preformed by a service	String
ComparatorOperand	hasURI	The preference URI to compare	URI
	hasValue	The value to compare with	String
Operand	isTrue		Boolean

Table 13 Object properties of classes of EasyTV Ontology

Class Name	Object properties		
	Name	Description	Range
User	hasPreference	The user preferences	UserPreferences
	hasSuggestedPreferences	The user suggested preferences	SuggestedPreferences
ConditionalPreference	hasConditions	The user's conditonal preferences	Operand
ContentAcecessibilityService	applies	The actions that apply a content adaptation service	ContentEnhanceme nt
	onContent	The content adaptation service applied on content.	AudioVisualContent
Substitution	by	A substitution by a content adaptation service	ContentAcecessibilit yService
	ofContentAdaptationService	A substitution of content adaptation service	ContentAcecessibilit yService
Owl:Thing	Contains	Anthing can contain anthing	Owl:Thing
ContentEnhance ment	executedBefore	What content enhancement can be exectued before other	ContentEnhanceme nt
LogicalOperand	hasLeftOperand	The left side operand or a logicla operand	Operand
DualOperand	hasRigthOperand	The right side operand or a dual operand	Operand

4.4. Rules

Rules allow us to infer new statements from the existing ones. These rules are grouped into rules for service substitution, rules for suggestions and rules for conditional preferences evaluation. Service substitution rules are rules that infers proper substitution of a service. Suggestions rules are the adaptation rules, these are rules that given the user set of preferences tries to infer proper adaptation. The conditional preferences evaluation rules are those that evaluate the user conditional preferences and applies the requested preferences.

4.4.1. Suggestion rules

Suggestion rules are rules that suggest services and settings based on the user preferences. Suggestion rules are a set of rules that start from the user preferences and end up in suggesting new preferences. The inference chain of these rules follows the logic “user preferences” => “accessibility needs” => “suggestions”. Explained in other words, there are a set of rules that

associated user selected preferences with accessibility needs, these are then fulfilled by preferences suggestions. UI suggestions rules are based on [9].

Visual: blind

People who are blind cannot access visual information in videos, player controls, status indicators, etc. They need the information in an alternative representation of audio or text. People who are blind use a screen reader and/or refreshable Braille display, and media content needs to be operable with these assistive technologies (ATs).

Table 14 Preferences that affect blind people

Needs	Preferences	Device
Screen reader	Enable	Tablet, mobile
Voice control	Enable	Tablet, mobile

Vision: low vision

1. Difficulty discriminating foreground information from background information or discriminating colours.
2. They may have difficulty reading when text is too small, has poor background contrast (too high or too low), or when outlined or other fancy font types or effects are used.
3. They may be using an AT that adjusts all the colours of the screen, such as inverting the colours.

Table 15 Preferences that affects low vision

Needs	Preferences	Device
Magnification	Text size Cursor size Subtitle size	Tablet, mobile, PC
Contrast	Text colour Background colour Subtitle colour Cursor colour	Tablet, mobile, PC
Reading	Text type Text style	Tablet, mobile, PC

Auditory: deaf

People who are deaf generally cannot use audio. Thus, an alternative representation is required, typically through synchronized captions and/or sign translation.

Table 16 Preferences that affect deafness

Needs	Preferences	Device
Sign avatar	Sign avatar	Tablet, mobile, PC

Auditory: hard of hearing

People who are hard of hearing may be able to use some audio material but might not be able to discriminate certain types of sounds, and may miss any information presented as audio if it contains frequencies they can't hear, or it is masked by background noise or distortion. They may miss audio which is too quiet, or of poor quality. Speech may be challenging if it is too fast and cannot be played back more slowly. Information presented using multichannel audio (e.g., stereo) may not be perceived by people who are deaf from one ear. People with cochlear implants may not have issues with audio volume levels, but comprehension may be challenging if the media experience is overwhelming.

Table 17 Preferences that affect hard of hearing

Needs	Preferences	Device
Audio level	Audio volume	Tablet, mobile, PC
Screen reader	Rate Volume	Tablet, mobile
Voice control	Voice control	Tablet, mobile

Rules

The first set of suggestion rules suggest magnification level, and content magnification based on the user text size and the device distance (in case no access to device distance, a default value of 15cm is assumed). Another set of rules suggest screen reader and voice interaction based on the user ability to hear and speak.

The following rules suggest magnification level based on the selected text size. A reference point for the calculation is normal vision text size and distance. A user with normal vision will use text size of 12pt at distance of 15cm, comparing the user selected text size and value with these values we can estimate the user magnification level. Magnification level would affect all UI elements that have a size.

Table 18 Suggestion for magnification

Jena Rule	Textual Description
(?user rdf:type owl:User)	?user is of type User that has preferences for text size ?size
(?user owl:hasPreference ?pref)	

(?pref owl:hasTextSize ?size)	
(?user owl:hasContext ?contx)	?user has context with proximity distance of value ?prox that is less than 10cm away
(?contx owl:hasProximity ?prox)	
le(?prox, 10)	
noValue(?pref owl:MagnificationScale)	The user hasn't set any magnification level
makeTemp(?magnification)	
=>	
(?magnification rdf:type Magnification)	Set the user level of magnification to 2
(?magnification owl:hasLevel 2)	
(?user owl:needAccessibility ?magnification)	

Table 19 Apply magnification on all elements that have size

Jena Rule	Textual Description
(?user rdf:type owl:User)	?user is of type User that has preferences for text size ?size
(?user owl:hasPreference ?prefs)	
(?prefs ?pref ?prefValue)	
(?pref owl:hasSize ?size)	?user has need for magnification accessibility of a level ?level
(?user owl:needAccessibility ?magnification)	
(?magnification rdf:type Magnification)	
(?magnification owl:hasLevel ?mgLevel)	
product(?size, ?mgLevel, ?newSize)	The new size is actually the product of the old value with the new.
=>	
Drop(4)	Change the element size to the new value
(?pref owl:hasSize ?newSize)	

Table 20 Suggest screen reader & voice control

Jena Rule	Textual Description
(?user rdf:type owl:User)	?user is of type User that has preference for volume level of zero.
(?user owl:hasPreference ?pref)	

(?pref owl:hasAudioDescription true)	
(?pref owl:hasAudioSubtitles true)	
noValue(?pref owl:hasScreenReader)	No preference for sign avatar
noValue(?pref owl:hasVoiceControl)	
(?user owl:hasSuggestedPreferences ?sugPref)	
=>	
(?sugPref owl:hasScreenReader true)	Suggest to the user to enable sign language
(?sugPref owl:hasVoiceControl true)	

Table 21 Suggest sign language

Jena Rule	Textual Description
(?user rdf:type owl:User)	?user is of type User that has preference for volume level of zero.
(?user owl:hasPreference ?pref)	
(?pref owl:hasVolumeLevel ?volume)	
equals(?volume, 0)	
noValue(?pref owl:hasSignAvatar)	No preference for sign avatar
(?user owl:hasSuggestedPreferences ?sugPref)	
=>	
(?sugPref owl:hasSignAvatar true)	Suggest to the user to enable sign language

4.4.2. Conditional preferences evaluation rules

Conditional preferences rules are rules that resolve conditions associated with the conditional preferences. These rules use built in Jena functions to resolve comparisons such as, greater than, less than, and equals. A conditional instance is an instance of the classes GE, GT, LT, LE, EQ and NE, with each conditional instance associated to two values . The first is the user preference type to compare with; the second is the value to be compared with. The logic of the rules is to extract the value from the condition the value and the type, then to get the user associated value and to compare them using one of the built in functions. The outcome is then associated with the conditional instance state.

Table 22 Equals preferences rule

Jena Rule	Textual Description
(?cond rdf:type owl:EQ)	?cond is of type EQ
(?cond owl:hasValue ?value)	?cond has value ?value

(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
equals(?nodeValue ?value ?res)	?res is True when ?nodeValue == ?value otherwise is False
(?cond owl:isTrue ?res)	Set ?cond is true value to ?res

Table 23 Not equals preferences rule

Jena Rule	Textual Description
(?cond rdf:type owl:NE)	?cond is of type NE
(?cond owl:hasValue ?value)	?cond has value ?value
(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
notEquals (?nodeValue ?value ?res)	?res is True when ?nodeValue != ?value otherwise is False
(?cond owl:isTrue ?res)	Set ?cond is true value to ?res

Table 24 Greater than preference rule

Jena Rule	Textual Description
(?cond rdf:type owl:GT)	?cond is of type GT
(?cond owl:hasValue ?value)	?cond has value ?value
(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
greaterThanEquals (?nodeValue ?value ?res)	?res is True when ?nodeValue > ?value otherwise is False

(?cond owl:isTrue ?res)	Set ?cond is true value to ?res
-------------------------	---------------------------------

Table 25 Greater than equal preference rule

Jena Rule	Textual Description
(?cond rdf:type owl:GE)	?cond is of type GE
(?cond owl:hasValue ?value)	?cond has value ?value
(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
greaterThanEquals (?nodeValue ?value ?res)	?res is True when ?nodeValue >= ?value otherwise is False
(?cond owl:isTrue ?res)	Set ?cond is true value to ?res

Table 26 Less than preference rule

Jena Rule	Textual Description
(?cond rdf:type owl:LE)	?cond is of type LE
(?cond owl:hasValue ?value)	?cond has value ?value
(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
greaterThanEquals (?nodeValue ?value ?res)	?res is True when ?nodeValue < ?value otherwise is False
(?cond owl:isTrue ?res)	Set ?cond is true value to ?res

Table 27 Less then equal preference rule

Jena Rule	Textual Description
(?cond rdf:type owl:LE)	?cond is of type LE
(?cond owl:hasValue ?value)	?cond has value ?value

(?cond owl:hasType ?type)	?cond has type ?Type
(?user rdf:type owl:User)	?user is of type User
(?user owl:hasPreference ?pref)	?user has preferences ?pref
(?pref ?type ?nodeValue)	?pref has property value ?nodeValue
=>	
greaterThanEquals (?nodeValue ?value ?res)	?res is True when ?nodeValue <= ?value otherwise is False
(?cond owl:isTrue ?res)	Set ?cond is true value to ?res

Table 28 Logical and rule

Jena Rule	Textual Description
(?cond rdf:type owl:AND)	?cond is of type AND
(?cond owl:hasLeftOperand ?leftOp)	?cond has left operand ?leftOp
(?cond owl:hasRightOperand ?rightOp)	?cond has right operand ?rightOp
(?leftOp owl:isTrue ?v1)	The value of isTrue of ?leftOp is ?v1
(?rightOp owl:isTrue ?v2)	The value of isTrue of ?rightOp is ?v2
=>	
and(?v1 ?v2 ?v3)	Set ?v3 to the value of the ?v1 & ?v2
(?cond owl:isTrue ?v3)	Set ?cond isTrue value to ?v3

Table 29 Logical OR rule

Jena Rule	Textual Description
(?cond rdf:type owl:OR)	?cond is of type OR
(?cond owl:hasLeftOperand ?leftOp)	?cond has left operand ?leftOp
(?cond owl:hasRightOperand ?rightOp)	?cond has right operand ?rightOp
(?leftOp owl:isTrue ?v1)	The value of isTrue of ?leftOp is ?v1
(?rightOp owl:isTrue ?v2)	The value of isTrue of ?rightOp is ?v2
=>	
or(?v1 ?v2 ?v3)	Set ?v3 to the value of the ?v1 ?v2
(?cond owl:isTrue ?v3)	Set ?cond isTrue value to ?v3

Table 30 Logical Not rule

Jena Rule	Textual Description
(?cond rdf:type owl:NOT)	?cond is of type NOT
(?cond owl:hasLeftOperand ?leftOp)	?cond has left operand ?leftOp
(?leftOp owl:isTrue ?v1)	The value of isTrue of ?leftOP is ?v1
=>	
not(?v1 ?v2)	Set ?v2 to the value of Not ?v1
(?cond owl:isTrue ?v3)	Set ?cond isTrue value to ?v2

4.4.3. Service substitution rules

Service substitution rules are rules that infer substitutions of a service. A service is considered a candidate for substitution if it does the same action on the same object. The rules actually try to find services that apply the same set of actions on the same set of objects.

Table 31 Rule for recursive appliance

Jena Rule	Textual Description
(?service rdf:type ?type)	A ?service is of a type that is a subclass of <i>ContentAccessibilityService</i> class
(?type rdfs:subClassOf owl:ContentAccessibilityService)	
(?service owl:applies ?something)	The ?service applies ?something on an ?object
(?service owl:onContent ?object)	
(?object owl:contains ?otherObject)	?object contains ?otherObject
noValue(?service owl:onContent ?otherObject)	No other statement exists that declares this service a modifier of ?otherObject
=>	
(?service owl:onContent ?otherObject)	Declare that this service applies the enhancement on the contained object

Table 32 Rule for declaring service substitutions

Jena Rule	Textual Description
(?service1 rdf:type ?type1)	A ?service1 is of a type that is a subclass <i>ContentAccessibilityService</i>
(?type1 rdfs:subClassOf owl:ContentAccessibilityService)	
(?service1 owl:applies ?something)	?service1 applies ?something on an ?object
(?service1 owl:onContent ?object)	

(?service2 rdf:type ?type2)	A ?service2 is of a type that is a subclass <i>ContentAccessibilityService</i>
(?type2 rdfs:subClassOf owl:ContentAccessibilityService)	
(?service2 owl:applies ?something)	That applies the same action on the same object
(?service2 owl:onContent ?object)	
strConcat(?service1, ?service1t)	?service1 and ?service2 are two different services
strConcat(?service2, ?service2t)	
notEquals(?service1t, ?service2t, ?res)	
noValue(?object owl:contains)	?object does not contains other objects
noValue(?service1 owl:substitutePartOf ?service2)	?service1 doesn't subtitled part of ?service2
makeTemp(?sub)	
=>	
(?service1 owl:substitutePartOf ?service2)	Create a substitution that declares that ?service2 substituted ?service1
(?sub rdf:type owl:Substitution)	
(?sub owl:ofContentAdaptationService ?service1)	
(?sub owl:byContentAdaptationService ?service2)	

Table 33 Rule for ordering service substitutions

Jena Rule	Textual Description
(?sub1 rdf:type owl:Substitution)	?sub1 and ?sub2 are two different substitutions of type <i>ofContentAdaptationService</i>
(?sub2 rdf:type owl:Substitution)	
strConcat(?sub1, ?sub1t)	
strConcat(?sub2, ?sub2t)	
notEquals(?sub1t, ?sub2t, ?res)	
(?sub1 owl:ofContentAdaptationService ?service)	
(?sub2 owl:ofContentAdaptationService ?service)	
(?sub1 owl:byContentAdaptationService ?service1)	?sub1 and ?sub2 are substitution of ?service1
(?sub2 owl:byContentAdaptationService ?service1)	
(?service1 owl:applies ?something1)	These services applies two different actions, where ?something1 must be executed before ?something2
(?service2 owl:applies ?something2)	
(?something1 owl:excecutedBefore ?something2)	
=>	

(?sub1 owl:nextSubstitution ?sub2)

Declare that ?sub2 next to ?sub1

4.5. Statistical matchmaker

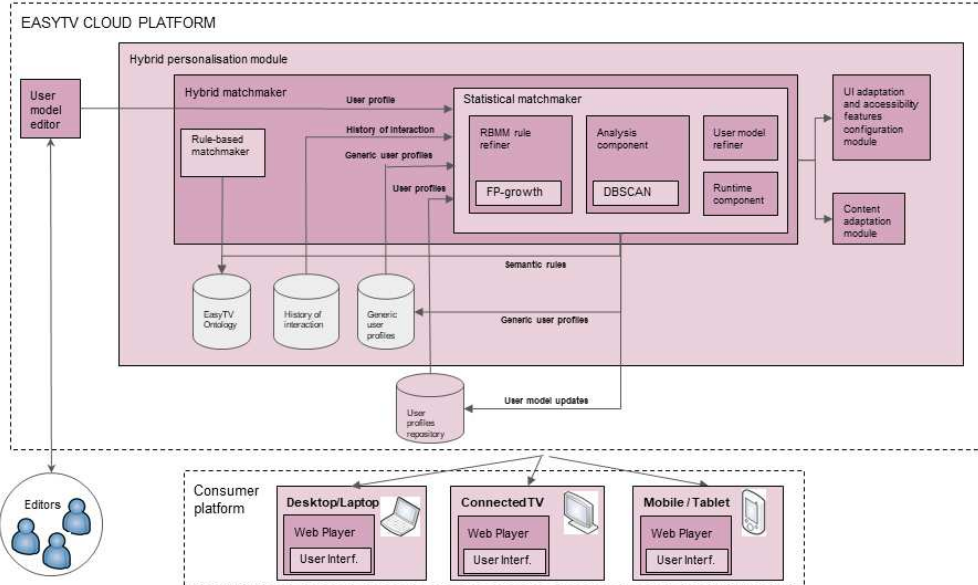


Figure 25 Statistical matchmaker overview

The statistical matchmaker is an inner component of the hybrid matchmaker that implements collaborative filtering (Section 3). A complete description of the components that constitute the statistical matchmaker, its input/output data and its interaction with other EasyTV components as illustrated in deliverable D4.2 [10]. In addition, to an analytic description of the refinement process of personalization. Other deliverables that also address the statistical matchmaker are implementation of the content adaptation D4.1 [11] and its refined version D4.4 [12].

In relation to UI and content personalization no specific handling other than the usual is required during the clustering phase. The STMM converts a user profile into a multidimensional vector, including the user preferences for UI adaptation. As results, during the clustering phase users of similar UI preferences are found and used for UI suggestions. We must note here that because indicative URLs is used to address preferences (section 2.2) clustering of users based only on UI related preferences (dimensions) is also feasible.

4.6. Hybrid matchmaker

The EasyTV hybrid matchmaker uses the rule-based and the statistical matchmaker output in a combined manner. Weights represent the contribution of each approach into the hybrid outcome. The following formula describes the hybrid matchmaking approach:

$$HP = \frac{W_1 \times P_1 + W_2 \times P_2}{W_1 + W_2}$$

Where

W_1, W_2 : are weights that correspond to each matchmaking approach.

P_1, P_2 : represent corresponding suggestions.

HP: hybrid matchmaker outcome.

Initially, the hybrid matchmaker assigns an equal pair of weights to each user. Periodically, once a day, it evaluates a set of weights and select the pair that best reflects the user actual choices. The evaluation process starts by creating different pairs of weights, for a list of hybrid suggestions and user choices it estimates which pairs minimizes the Root Square Error function value:

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (pr_{ui} - tr_{ui})^2}$$

Where

T : represents the list of suggestions (UI and accessibility features adaptation configuration).

pr_{ui} , tr_{ui} : corresponds to the hybrid suggestions and user actual choices respectively.

In this way, the validity of each matchmaking approach (rule-based and statistical) is estimated in relation to the actual user choices. As each pair correspond to a hybrid outcome, the pair that minimizes RMSE value is the one that more precisely reflects the actual user preferences. The pair of weights that minimizes RSME value is assigned to the user.

4.6.1. Hybrid matchmaker API

4.6.1.1 Personalize a user profile

URL		https://hp-api.easytv.eng.it/EasyTV_HBMM_Restful_WS/match	
Description		Personalize the give user profile	
HTTP Method		POST	
Content type		application/json	
Authorization		-	
Body			
Format		JSON	
Name	Type	Value space	Comments
user_profile	JSON	-	The user's profile
user_context	JSON	-	The user context information
Output			
Format		json	
Input			
{ "user_context": { "http://registry.easytv.eu/context/location": "fr", "http://registry.easytv.eu/context/time": "12:00:00"			

```

},
"user_profile": { "user_preferences": {
  "default": { "preferences": {
    "http://registry.easytv.eu/application/cs/accessibility/textDetection": false,
    "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
    "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#b23b41",
    "http://registry.easytv.eu/common/display/screen/enhancement/magnification": 70,
    "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
    "http://registry.easytv.eu/common/display/screen/enhancement/background": "#18d4dc"
  }}
}
}
}

```

Output

```

{
  "user_context": {
    "http://registry.easytv.eu/context/location": "fr",
    "http://registry.easytv.eu/context/time": "12:00:00"
  }
  "user_profile": { "user_preferences": {
    "default": { "preferences": {
      "http://registry.easytv.eu/application/cs/cc/audioSubtitles": false,
      "http://registry.easytv.eu/application/cs/accessibility/textDetection": false,
      "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
      "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#b23b41",
      "http://registry.easytv.eu/common/display/screen/enhancement/magnification": 70,
      "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
      "http://registry.easytv.eu/common/display/screen/enhancement/background": "#18d4dc"
    }}
  }
}
}

```

4.6.1.2 Personalize a content

URL	https://hp-api.easytv.eng.it/EasyTV_HBMM_Restful_WS/content
Description	Personalize the give user profile
HTTP Method	POST
Content type	application/json
Authorization	-

Body			
Format		JSON	
Name	Type	Value space	Comments
user_profile	JSON	-	The user's profile
user_context	JSON	-	The user context information
user_content	JSON	-	The user context information
Output			
Format			json
Input example			
<pre>{ "user_context": { "http://registry.easytv.eu/context/location": "IT", "http://registry.easytv.eu/context/time": "12:00:00" }, "user_content": "http://yt-dash-mse-test.commondatastorage.googleapis.com/media/car-20120827-manifest.mpd", "user_profile": { "user_preferences": { "default": { "preferences": { "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false, "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false, "http://registry.easytv.eu/application/cs/accessibility/textDetection": true, "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 70, "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false, "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2", "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58, "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243", "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA", "http://registry.easytv.eu/application/cs/audio/volume": 39 } } } }</pre>			
Output			
<pre>{ "user_context": { "http://registry.easytv.eu/context/location": "IT", "http://registry.easytv.eu/context/time": "12:00:00" }, "user_content": "http://yt-dash-mse-test.commondatastorage.googleapis.com/media/car-20120827-manifest.mpd",</pre>			

```

"user_profile": {
  "recommended_suggestions": {
    "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
    "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
    "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243"
  },
  "user_preferences": {
    "default": { "preferences": {
      "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
      "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
      "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
      "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 50,
      "http://registry.easytv.eu/application/cs/accessibility/characterRecognition ": false,
      "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
      "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
      "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
      "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
      "http://registry.easytv.eu/application/cs/audio/volume": 39
    }
  }
}
}
}

```

5. INDICATIVE USE CASES

5.1. A user with medium vision

A user with medium level vision creates a profile using the user model editor. The user in the common preference tab of the editor set a preference for audio at 9, language of Italian, enables subtitles and choses quite high text size (28) with black background and yellow foreground color as shown in the following screen shots. The user does not bother setting any preferences on how to control the device and any content accessibility preferences. The last action of the user is to save the profile.

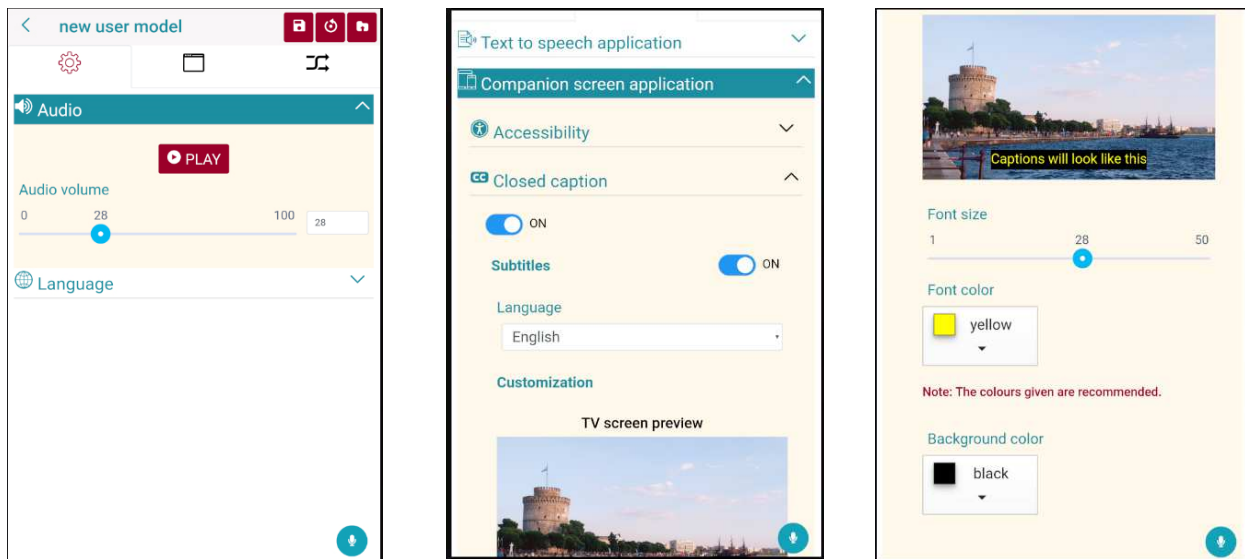


Figure 26 User profile settings

```
{
  "user_context": {
    "http://registry.easytv.eu/context/location": false,
    "http://registry.easytv.eu/context/time": "12:47:33",
    "http://registry.easytv.eu/context/proximity": 15,
    "http://registry.easytv.eu/context/brightness": 15
  },
  "user_preferences": {
    "default": {
      "preferences": {
        "http://registry.easytv.eu/application/cs/audioVolume": 28,
        "http://registry.easytv.eu/common/subtitles": "EN",
        "http://registry.easytv.eu/common/subtitles/font/size": 28,
        "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#000000",
        "http://registry.easytv.eu/common/display/screen/enhancement/background": "#ffffff"
      }
    }
  }
}
```

Figure 27 User profile with medium vision

The personalization process starts when the user accesses the CS app. The CS app send the user model with the contextual information show Figure 27 to the user personalization end-point (section 4.6.1.1). Given the user text size in relation to the distance given, the personalization process assumes that the user may be in need of magnification. A normal vision user would require a smaller text size given the same distance. Thus, a good suggestion at this point would be to suggest a magnification level of 2 for other visual elements that the current profile lacks, such as cursor size

and content magnification as shown in the two profiles below.

```
{
  "user_context": {
    "http://registry.easytv.eu/context/location": false,
    "http://registry.easytv.eu/context/time": "12:47:33",
    "http://registry.easytv.eu/context/proximity": 15,
    "http://registry.easytv.eu/context/brightness": 15
  },
  "user_preferences": {
    "default": {
      "preferences": {
        "http://registry.easytv.eu/application/cs/audioVolume": 28,
        "http://registry.easytv.eu/common/subtitles": "EN",
        "http://registry.easytv.eu/common/subtitles/font/size": 28,
        "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#000000",
        "http://registry.easytv.eu/common/display/screen/enhancement/background": "#ffffff",
        "http://registry.easytv.eu/common/display/screen/enhancement/cursor/size": 2,
        "http://registry.easytv.eu/application/accessibility/imageMagnification/scale": 2,
        "http://registry.easytv.eu/application/accessibility/faceDetection": true
      }
    }
  }
}
```

Figure 28 User profile with medium vision and suggestions

The side effect of the adaptation is show in the following figures.

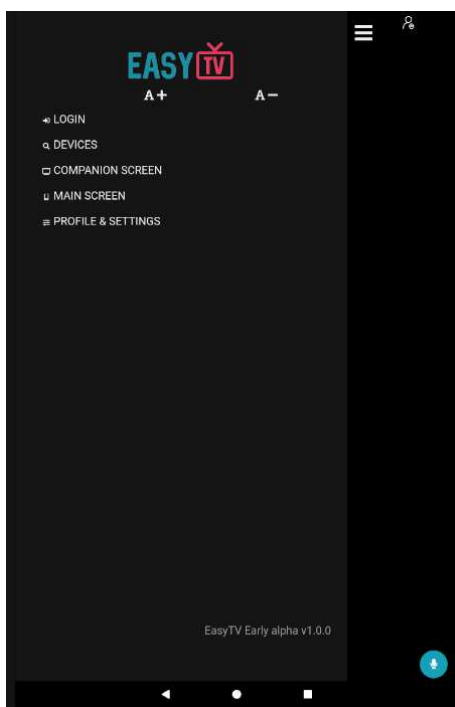


Figure 29 UI of magnification level 2

5.2. A profile with conditional preference

A user create a profile with a specific condition to set a proper contrast level at specific time. More specifically, the user want to say, “Between 21:00 and 23:00 o'clock O would like the background and foreground color to be set to black and yellow”. Using the editor the user would enter the following values at the conditional preferences tab.

Figure 30 Contrast conditional preference 1

Figure 31 Contrast conditional preference 2

```

{
  "user_context": {
    "http://registry.easytv.eu/context/location": "IT",
    "http://registry.easytv.eu/context/time": "21:10:00",
    "http://registry.easytv.eu/context/proximity": "15",
    "http://registry.easytv.eu/context/brightness": "15",
  },
  "user_profile": {
    "user_preferences": {
      "default": {
        "preferences": {
          "http://registry.easytv.eu/common/subtitles": "IT",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#32a852",
          "http://registry.easytv.eu/common/display/screen/enhancement/background": "#000000",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 28,
          "http://registry.easytv.eu/common/content/audio/language": "IT",
          "http://registry.easytv.eu/common/content/audio/volume": 8,
          "http://registry.easytv.eu/common/display/screen/enhancement/cursor/size": 2,
          "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 2,
          "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
        },
      },
      "conditional": [
        {
          "preferences": {
            "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#ecff33",
            "http://registry.easytv.eu/common/display/screen/enhancement/background": "#000000",
          },
          "name": "Low_contrast",
          "conditions": [
            {
              "operands": [
                {
                  "operands": [
                    "http://registry.easytv.eu/context/time",
                    "21:00:00"
                  ],
                  "type": "gt"
                },
                {
                  "operands": [
                    "http://registry.easytv.eu/context/time",
                    "23:00:00"
                  ],
                  "type": "lt"
                }
              ],
              "type": "and"
            }
          ]
        }
      ]
    }
  }
}

```

Figure 32 User profile with low contrast condition

Figure 32 User profile with low contrast condition the generated use profile with the specific condition been located under the “conditional” section. At some point, the user starts interacting with the CS app, which sends the user profiles to the hyper-personalization end-point (section 4.6.1.1).

The hybrid matchmaker in turn calls the rule-based matchmaker. RBMM’s conditional preferences set of rules (section 4.4.2) evaluate the conditional preferences. The contextual information indicates

that the condition is valid; as a result, the preferences paired with the condition is applied to the default set of preferences.

5.3. Device control suggestions

In the case of a user who has enabled all content accessibility services that convert visual content into audible content without enabling the corresponding control accessibility services, the personalization framework will spot such mismatch and suggests to the use to enable also the control accessibility services.

```
{
  "user_context": {
    "http://registry.easytv.eu/context/location": "IT",
    "http://registry.easytv.eu/context/time": "21:10:00",
    "http://registry.easytv.eu/context/proximity": "15",
    "http://registry.easytv.eu/context/brighthness": "10",
  },
  "user_profile" : {
    "user_preferences": {
      "default": {
        "preferences": {
          "http://registry.easytv.eu/common/subtitles": "IT",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#32a852",
          "http://registry.easytv.eu/common/display/screen/enhancement/background": "#000000",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 28,
          "http://registry.easytv.eu/common/content/audio/language": "IT",
          "http://registry.easytv.eu/common/content/audio/volume": 8,
          "http://registry.easytv.eu/common/display/screen/enhancement/cursor/size": 2,
          "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
          "http://registry.easytv.eu/application/cs/accessibility/soundDetection": true,
          "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
          "http://registry.easytv.eu/application/cs/audio/audioDescription": true
        }
      }
    }
  }
}
```

Figure 33 User profile with no visual preferences

```
{
  "user_context": {
    "http://registry.easytv.eu/context/location": "IT",
    "http://registry.easytv.eu/context/time": "21:10:00",
    "http://registry.easytv.eu/context/proximity": "15",
    "http://registry.easytv.eu/context/brighthness": "10",
  },
  "user_profile" : {
    "user_preferences": {
      "default": {
        "preferences": {
          "http://registry.easytv.eu/common/subtitles": "IT",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/color": "#32a852",
          "http://registry.easytv.eu/common/display/screen/enhancement/background": "#000000",
          "http://registry.easytv.eu/common/display/screen/enhancement/font/size": 28,
          "http://registry.easytv.eu/common/content/audio/language": "IT",
          "http://registry.easytv.eu/common/content/audio/volume": 8,
          "http://registry.easytv.eu/common/display/screen/enhancement/cursor/size": 2,
          "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
          "http://registry.easytv.eu/application/cs/accessibility/soundDetection": true,
          "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
          "http://registry.easytv.eu/application/cs/audio/audioDescription": true,
          "http://registry.easytv.eu/application/control/screenReader": true,
          "http://registry.easytv.eu/application/control/voiceControl": true
        }
      }
    }
  }
}
```

Figure 34 User profile with control suggestions

5.4. Content adaptation case

A user has preferences for audio and subtitle language of Catalan, and requested face magnification to be set on. Based on the MPD file of the content, the user preferences can be matched directly. The CSApp sends the user profile and MPD file content and gets in return the output of Table 9.

The outcome shows that the user content preferences has been fulfilled (enable text detection) and that additional accessibility services have been suggested. The user profile has strong indication that the user may need additional accessibility services. The user level of magnification for instance is high in combination with the lacking of face detection service triggers a rule in the RBMM to suggest face detection service. In addition, the STMM clustering method has found that similar users have preferences for colors (foreground and background colors) with higher level of contrast. The output of the personalization is then to suggest enabling the face detection in addition to higher contrast colors for foreground and background colors.

```
{
  "user_preferences": {
    "default": {
      "preferences": {
        "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
        "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
        "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
        "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 70,
        "http://registry.easytv.eu/application/cs/accessibility/characterRecognition": false,
        "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
        "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
        "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
        "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
        "http://registry.easytv.eu/application/cs/audio/volume": 39
      }
    }
  }
}
```

Figure 35 User profile for content adaptation

```
{
  "recommended_suggestions": {
    "http://registry.easytv.eu/application/cs/accessibility/faceDetection": true,
    "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#000000",
    "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ffffff"
  },
  "user_preferences": {
    "default": {
      "preferences": {
        "http://registry.easytv.eu/application/cs/accessibility/soundDetection": false,
        "http://registry.easytv.eu/application/cs/accessibility/faceDetection": false,
        "http://registry.easytv.eu/application/cs/accessibility/textDetection": true,
        "http://registry.easytv.eu/application/cs/accessibility/imageMagnification/scale": 50,
        "http://registry.easytv.eu/application/cs/accessibility/characterRecognition": false,
        "http://registry.easytv.eu/application/cs/cc/subtitles/fontColor": "#39dc2",
        "http://registry.easytv.eu/application/cs/cc/subtitles/fontSize": 58,
        "http://registry.easytv.eu/application/cs/cc/subtitles/backgroundColor": "#ee6243",
        "http://registry.easytv.eu/application/cs/cc/subtitles/language": "CA",
        "http://registry.easytv.eu/application/cs/audio/volume": 39
      }
    }
  }
}
```

Figure 36 Flat match with personalized suggestions user profile

6. CONCLUSION AND FUTURE WORK

The UI and content adaptation process consists of an important part of the personalization process as a whole. The process starts with the end user creating a profile that include all his/her preferences and ends with new enriched profile. The starting point, the user model, is a set of preferences regarding the UI and content. These preferences are classified into generic and application specific and conditional. The user model editor is a GUI tool that allows users to easily create/edit/delete of their profiles. The implementer of the personalization process is the hyper-personalization model, with its three main components the hybrid matchmaker, the statistical matchmaker and the rule-based matchmaker, in addition to the user management framework.

The user management framework stores user profiles and implement an authentication and login process. Users that choose to register have their profile stored in the use management framework. The statistical matchmaker implements a statistical analysis approach on user's stored profiles. The aim of this process is to use similar users' to make suggestions. The rule-based matchmaker follow a rule based approach that is based on an ontology, a set of triple statements and rules. The defined ontology reflects the user preferences and needs in addition to EasyTV available services. The rules are grouped into rules for conditional preferences, rules for suggestions and rules for services substitution. The conditional preferences rules evaluate and handle conditional preferences. The service substitution rules are rules that suggest possible service substitutions. Lastly, suggestion rules are rules for suggesting adaptations. These rules suggest new configuration given the user set of preferences, they identify possible gaps and suggests configuration to cover them.

The UI personalization rules of the RBMM that identify such cases can be quiet, long and messy with many conditions to meet that have discrete values and various ranges. To complement for such tedious task, only a small set of these rules are declared statically, and the majority is to be extracted from the user's data. The personalization framework extracts association rules from user's data to improve the accuracy of the personalization process. These association rules are actually correlations between preferences that occurred in a respectable number of users that would make them good enough for suggestions. These association rules will be used to update existing UI and content adaptation rules or to add or remove others.

7. REFERENCES

- [1] I. 24751-2:2008, "Information technology — Individualized adaptability and accessibility in e-learning, education and training — Part 2: "Access for all" personal needs and preferences for digital delivery," ISO/IEC, 2008.
- [2] Abowd, Dey, Brown, Davies, Smith and Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Gellersen HW. (eds) Handheld and Ubiquitous Computing. HUC 1999. Lecture Notes in Computer Science*, Berlin, Heidelberg, 1999.
- [3] ISO, "ISO/IEC DIS 24752-8," [Online]. Available: <https://www.iso.org/obp/ui#iso:std:iso-iec:24752:-8:dis:ed-1:v1:en>.
- [4] S. Ishihara, The series of plates designed as a test for color blindness, 1972.
- [5] D. M. Bach, "Freiburg Vision Test ('FrACT')," [Online]. Available: <https://michaelbach.de/fract/download.html>.
- [6] "D1.1 "EasyTV User scenario and requirements definition"," p. 20.
- [7] "D7.2 Report on Identification of Standardisation Bodies," [Online]. Available: <https://easytvproject.eu/files/D7.2.pdf>.
- [8] "D1.4 Final release of the EasyTV system architecture".
- [9] "W3C Media Accessibility User Requirments," [Online]. Available: <https://www.w3.org/TR/media-accessibility-reqs/#summary-of-accessible-media-requirements-by-type-of-disability>.
- [10] D4.2 "Self-learning system for improving personalisation capabilities"..
- [11] D4.1 "Content adaptation using DASH streaming services".
- [12] D4.4 "Content adaptation using DASH streaming services".
- [13] C. Loitsch, G. Weber, N. Kaklani, K. Votis and D. Tzovaras, "A knowledge-based approach to user interface adaptation from preferences and for special needs," *User Modeling and User-Adapted Interaction*, vol. 17, no. 3-5, p. 445–491, December 2017,.

- [14] expressjs.com, "Express4.16.4 Fast, unopinionated, minimalist web framework for Node.js," 25 4 2019.
[Online]. Available: <https://expressjs.com>.