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.

Self-learning system for improving personalisation capabilities

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.:

Disclaimer

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:	UPM, CCMA
DOCUMENT NUMBER:	D4.2
DOCUMENT TITLE:	EasyTV self-learning system for improving personalisation capabilities
WORK-PACKAGE:	WP4
DELIVERABLE TYPE:	Prototype
CONTRACTUAL DATE OF DELIVERY:	31-12-2018
LAST UPDATE:	21-12-2018
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	11/13/2018	Draft	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Table of Contents Architectural overview
v.0.2	11/20/2018	Draft	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI)	Architectural description of the hybrid matchmaker (rule-based matchmaker and statisical matchmaker) section 4
v.0.3	12/03/2018	Draft	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI)	Improvement of personalisation accuracy section 3.1, 3.1.1, 3.1.2
v.0.4	12/10/2018	Draft	Jordi Fabregat (CCMA)	Contribution to Screen Reader and Colour-blind Subtitles section 5.1, 5.2
v.06	12/17/2018	Draft	Silvia Uribe (UPM) Francisco Moreno (UPM)	Collecting data in CS App section 4.1.1.4
v.0.7	12/19/2018	Draft	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI)	Statistical analysis section 3.1.1 Association rules learning section 3.1.2 Indicative use cases section 5
v.0.8	12/19/2018	Final	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI)	Final version ready for internal review

			Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI)	
v.0.9	12/22/2018	Final	Nikolaos Kaklanis (CERTH/ITI) Dimosthenis Elmas (CERTH/ITI) Konstantinos Votis (CERTH/ITI) Georgios Gerovasilis (CERTH/ITI) Salim Ghannoum (CERTH/ITI) Dimitrios Tzovaras (CERTH/ITI)	Final version ready for submission

Definitions, Acronyms and Abbreviations

ACRONYMS / ABBREVIATIONS	DESCRIPTION
AT	Assistive Technology
DIAL	Discovery and Launch
CS	Companion Screen
CF	Collaborative Filtering
KB	Knowledge Base
EM	Expectation Maximization (clustering algorithm)
DBSCAN	Density–based spatial clustering of applications with noise (clustering algorithm)
STMM	Statistical Matchmaker
RBMM	Ruled-based Matchmaker
FOL	First Order Logic
MPD	Media Presentation Description
HbbTV	Hybrid Broadcast Broadband TV
JSON	JavaScript Object Notation
UI	User Interface
UIDL	User Interface Description Language
RMSE	Root Mean Square Error
AR	Association Rule
AOP	Aspcet Oriented programming

Table of Contents

List of Figures

List of Tables

Executive Summary

This document corresponds to the deliverable D4.2 *"EasyTV self-learning system for improving personalisation capabilities"* of the work pacakge WP4. More specifically, it is mostly related to the work conducted in task T4.3 *"Personalised services for people with disabilities, including self-adaptive and tailored services, which can learn from users' actions to improve the accuracy of the personalisation"*. The deliverable addresses the issue of improving the personalisation capabilities of the EasyTV platform by exploiting the historical data produced by all users, focusing on the description of the architecture of the statistical matchmaker and its interaction with the other EasyTV components.

The deliverable is divided into the following chapters:

- **Chapter 1** includes an introduction to the current deliverable goals and its relation to other EasyTV work.
- **Chapter 2** presents related work on UI/Context adaptation.
- **Chapter 3** describes how the system learns from uses' data/interaction towards increasing the accuracy of personalisation.
- **Chapter 4** presents the architectural overview of the hyper-personalisation module with special focus on the architecture of the statistical matchmaker, its inputs, outputs and the main EasyTV components that the matchmaker interacts with.
- **Chapter 5** presents a set of indicative use cases that reflect the functionality and usefulnes of the personalisation approach.
- **Chapter 6** includes the conclusions and future work.

1. INTRODUCTION

One of the main features of the EasyTV platform is the auto-personalisation framework, which supports recommendations and automatic launching of assistive technologies (e.g. screen reader, sign language avatar, etc.). It also supports the automatic adjustments of the corresponding settings (e.g. speech rate, magnification factor, etc.) by applying rule-based and statistical matchmaking based on the user profile, the available settings and current context (e.g. device/OS used, environmental factors, etc.). Towards improving the accuracy of the personalisation, the auto-personalisation framework takes into consideration similar user preferences and learns from the actions of similar users.

The EasyTV hyper-personalisation and adaptation module follows a hybrid matchmaking approach that combines two different filtering techniques, a content-based and a collaborative one. The content-based matchmaking approach infers personalized suggestions based on rules defined by domain experts. On the other hand, the collaborative approach taps into “the wisdom of the crowd” and is based on the assumption that similar users tend to have similar preferences and interaction. This is achieved by exploiting users’ available data (i.e. user profiles and history of interactions) to infer personalized suggestions.

This deliverable presents the architectural overview of the hyper-personalisation module focusing mainly on the collaborative filtering approach and the way it affects the accuracy of the whole personalisation process. The components are presented in a top-down approach addressing design decisions taken at each step. The components are mainly addressed regarding their functionality, input/outputs and their interaction with other components.

The work described in this document serves the objective **OBJ2: Hyper-personalisation of the content experience and interaction**, which is linked to **WP4** and is a part of the implementation of **Service3: Hyper-Personalised access services with profiling of users’ experience and interaction**. It mainly addresses work done for T4.3 *“Personalised services for people with disabilities, including self-adaptive and tailored services, which can learn from users’ actions to improve the accuracy of the personalisation”*. It also depends on and utilizes work done in task T4.1 *“Adaptive menus and graphical interface using user models”* as well as in T4.2 *“Adaptation of level of content description using standardised DASH streaming services”*.

2. RELATED WORK ON UI/CONTENT ADAPTATION

Recommender Systems (RSs) are software tools and techniques providing suggestions for items that best suit to the needs and preferences of a user [1]. The EasyTV matchmaker can be considered as a recommender system, as it provides personalized suggestions for UI and content adaptation.

RSs differentiate based on the sources of information and the information-filtering algorithm they use. Information filtering is the process of removing redundant information using automated or computational methods. Sources of information may be the user’s history of interaction, similar users’ information, demographic data of classes of users etc. The filtering algorithms can be classified into collaborative filtering, content-based filtering, demographic filtering and hybrid filtering [2].

The content-based filtering (CB) approach uses the user’s history of interaction to generate recommendations by matching user preferences with a set of characteristics related to resources or products. It is based on the assumption that past user choices can be used to derive future recommendations. The collaborative filtering (CF) approach, also called social, exploits the information available regarding other users in the recommendation process. It is based on the assumption that similar users, regarding some specific characteristics, tend to match also in their other characteristics. Based on that, a collaborative filtering can infer recommendations for a specific user taking into account the available information of other “similar” users (i.e. having

similar needs, preferences, behavior, etc.).

CFs systems require a sufficient homogenous amount of users' data. They tend to offer poor results when there is not sufficient data, which are characterized by heterogeneity [3]. There are different techniques to handle the sparsity of the data. For example, the Amazon's recommender system handles the sparsity of customer vector of item ratings by applying an "item-to-item" collaborative filtering algorithm, which actually finds similar items and combines these into a list [4]. On the other hand, the CB filtering approach suffers from the cold starting problem, where no historical data are available. To overcome the limitations of both recommendation approaches, a hybrid technique can be used to combine the advantages of multiple filtering techniques.

The EasyTV platform offers a personalisation framework that aims at enabling automated launching of accessibility services and interface/content adaptation according to users' profile and contextual information by taking also into consideration possible functional limitations of the user. Relevant projects in the field of accessibility and UI adaptation include among others the Cloud4All and the IN LIFE project [5] [6], where both designs use a hybrid matchmaking approach. The hybrid design of the Cloud4All matchmaker allows the users to get the most out of the two matchmaking (rule based and statistical) approaches. This matchmaker maps the needs and preferences of a person with respect to customization features and accessibility aids available on the interactive devices that the person is using in a certain context. The IN LIFE project also focuses on accessibility and it follows a hybrid approach too by using a rule-based and a statistical matchmaker. The Scalable Neighborhood Using Clustering algorithm is used by the statistical matchmaker [5] to address sparsity and scalability issues. The usage of an extra recommender using demographic filtering and the k-prototypes clustering algorithm is also used in order to address the cold starting problem that occurs when there are no historical data available. Another relative project is the myUI project, which is based on design patterns for creating accessible UIs [7]. The ExTraS project on the other hand elaborates the usage of a reinforcement-learning algorithm to adapt the contents of a webpage by learning the optimal adaptations from the users' past actions [8]. Other projects on the accessibility domain perform content adaptation for assisting users with special needs. Such a project is REMPAD that recommends content that can be used with reminiscence therapy, for people with mild to moderate dementia [9].

The EasyTV matchmaking approach is also a hybrid approach that exploits both the knowledge coming from domain experts through rules related to content/UI adaptation along with useful results that can be extracted though statistical analysis applied on historical data coming from the same user or other similar users. More details on the EasyTV hybrid matchmaking approach are provided in the following sections by focusing mainly on the improvement of the accuracy of the personalisation process resulted from the analysis performed by the statistical matchmaker.

3. THE EASYTV HYBRID MATCHMAKING APPROACH

The EasyTV platform uses a hybrid matchmaking approach to combine the output of two main matchmaking methods, the content-based (rule-based matchmaking) and the collaborative (statistical matchmaking). Table 1 summarizes the main personalisation suggestions inferred by the hybrid matchmaker. These include suggestions regarding the configuration of built-in accessibility features (enabling or disabling them and proposing proper adjustments) and adaptations on the UI and audio-visual content.

Table 1 Hybrid matchmaker service functionality & sub functionality

SERVICE	FUNCTIONALITY	SUB FUNCTIONALITY
Hybrid Matchmaking (rule-based, statistical)	matchmaking for automatic configuration of built-in accessibility features	accessibility features turning on/off
		accessibility features adjustment

	matchmaking for GUI adaptation	adjustment of built-in settings of TV operating systems and applications
		adjustment of User Interfaces
	matchmaking for personalized DASH streaming services	setting the subtitle that corresponds to the end-user's language
		setting the audio that corresponds to the end-user's language

Not all matchmaking approaches can infer valuable suggestions all the time. There are cases when only one matchmaking approach can offer proper suggestions. For instance, the rule-based matchmaker cannot provide recommendations when there are no defined rules for addressing the very special needs that may be included in a user profile while the statistical matchmaker may not be able to provide recommendations when the amount of historical data is not sufficient. To handle all possible cases, the EasyTV matchmaker follows a hybrid approach that combines both the rule-based and the statistical methods.

Rule-based matchmaking

The EasyTV rule-based matchmaker is based on a set of predefined rules provided by domain experts. However, refinements on these rules as well as generation of new rules can be automatically performed through the statistical matchmaker as described in the following sections. The rule-based matchmaker can also address the cold starting problem, where the statistical matchmaker cannot provide accurate results due to the lack of historical data needed for the statistical analysis.

More details related to the rule-based matchmaker will be included in the deliverable *D4.3 "EasyTV user interface adaptation framework" [M19]*.

Statistical matchmaking

The EasyTV statistical matchmaker provides personalized suggestions for UI/content adaptation based on the needs, preferences and corresponding interactions of similar users and it can also improve the accuracy of the whole personalisation process as described in detail in section 3.1.

Hybrid matchmaking

The EasyTV hybrid matchmaker uses both the rule-based and the statistical matchmaker in a combined manner using dynamically adjusted weights. 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: the weights that correspond to each matchmaking approach.

P_1, P_2: represent corresponding suggestions.

HP: hybrid matchmaker outcome.

The hybrid matchmaker assigns initially a pair of weights, with value 0.5, to each user. Periodically, once a day, it evaluates and adjusts these weights to best reflect the user actual preferences. 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:

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

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.

3.1. Improvement of the accuracy of personalisation

The EasyTV statistical matchmaker is able to further improve the accuracy of the matchmaking results by taking into account the needs, preferences and interactions of similar users. It is straightforward that the results of the statistical matchmaker are strongly dependent on the size of the available data to be analyzed. The improvement of the accuracy of the personalisation is performed through suggestions for modifications/refinements on the:

- **user profile**: New UI/content preferences may be suggested to the user.
- **set of rules supported by the rule-based matchmaker**: The existing rules can be automatically refined and new rules can be automatically generated.

The following sections provide more details regarding the statistical analysis performed by the EasyTV statistical analysis.

3.1.1. User Model refinement

The EasyTV statistical matchmaker uses statistical inference to induce suggestions related to the user profile content. These modifications mainly affect the user preference sections and may result in adding, editing or removing generic or contextual preferences.

Statistical inference aims at finding correlations between data that could result to useful conclusions for the personalized process, which is a computationally intensive task especially when large amount of data is available. Towards boosting the performance of the statistical inference process, two additional steps has been added (clustering and generic user profiles creation) before the core statistical analysis steps:

- **A clustering phase**, where clusters of similar users profiles are identified.
- **A Generic User Profiles generation phase** that included the generation of generic user profiles that correspond to the center of each cluster.
- **A similarity calculation phase**, where the user profile of the current user is compared and matched with a generic user profile generated in the previous phase.
- **Inference**: infers proper suggestions for specific user given a set of similar users' profiles.

More specifically, this results in a chain of steps where the output of one step is the input of the next, as described in the following sections.

3.1.1.1 Clustering phase

The clustering phase corresponds to the grouping of all similar sets of user profiles in groups called clusters. Clustering algorithms differ in the way they understand what constituted a cluster and in the process of finding them. Popular notions of clusters include small distances between clusters

members, dense area of the data space, or intervals on particular statistical distributions.

We are looking to find a clustering algorithm capable of clustering uses' profiles. User profiles are interpreted by the statistical matchmaker as a high dimensional vector. The following overview lists the most prominent clustering high dimensional data sets by just using a similarity measure:

- **K-means**: In *K-means* clustering, given a set of *n* data points in *d-dimensional* space R^d and an integer *k*, the problem is to locate a set of *k* points in R^d, called centers, so as to minimize the mean squared distance from each data point to its nearest center [10]. K-means thus cluster available data into a *k* number of spherical clusters. A limitation of the k-means algorithms is that it cannot find non-convex clusters [11].
- **Mean-shift clustering:** Mean shift is a procedure for locating the maxima, the modes, of a density function given discrete data sampled from that function [12]. Mean-shift clustering does not require the number of clusters to identify; however, it needs the radius "r" of each cluster and ends up identifying spherical set of clusters.
- **DBSCAN:** A density-based clustering algorithm. In density-based clustering, clusters are defined as a higher dense area from the rest of the available data. Given a set of points in some space, it groups together points with many nearby neighbors (closely packed), marking as outliers points that lie alone in low-density regions (with too far away nearest neighbors). DBSCAN is one of the most common clustering algorithms and most cited in scientific literature [13]. It finds clusters of any shape and does not require the number of clusters.

The clustering algorithm clusters users' profiles into a group of similar profiles. That requires a mathematical similarity function capable of comparing users' profiles. As there is no objective similarity function capable of comparing humans and grasping completely how two human preference sets are to be compared. Interpreting users profiles as a high-dimensional vector is good enough to identify potential correlation.

3.1.1.2 Generic users' profiles creation phase

Generic user profiles creation is the process of constructing new user profile that sum up a group of users' profiles. These generic profiles correspond to the center of the users' profiles group. In high dimensional data, such as user profiles, the center of the group corresponds to the center of all its dimensions. For numerical dimensions, the center is the average value of all its dimensions. As a result, for a given set of clusters the output of this step is actually a set of generic user profiles that corresponds to their centers.

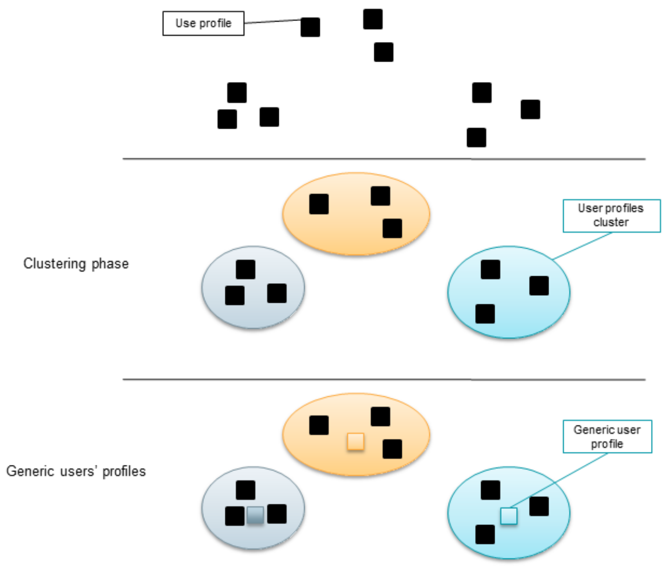

Figure 1 Generic user profiles creation

It is clear that comparing with all available users' profiles poses a serious performance issue. To reduce the runtime performance of the statistical analysis, all clusters of user profiles are summarized by their generic profiles. Figure 1 illustrates the clustering phase and the generic user profile creation phase. Statistical inference is now simplified to finding all similar generic profiles and extracting the corresponding suggestions. This greatly reduces the number of comparisons required for each individual matching case.

3.1.1.3 Similarity calculation

In this phase, the generic user profiles, generated in the previous step, are compared with the user profile of the current user. The process requires a similarity function able to compare user profiles. The statistical matchmaker interprets a user profile as a high-dimensional vector, consisting of key-value pairs, with each pair representing an attribute of the profile. A vector representation allows utilizing many well-known similarity measures. A widely used similarity function is Pearson's distance formula [14].

The distance between two users' profiles, the current user profile and the generic one, are measured using the following formula:

$$dist(up, vp) = N - \sum_{\iota \in key} 1 - \rho_i(up_i, vp_i)$$

up, vp: the user and the generic profile respectively.

N: the total number of generic user profiles.

ρ: Pearson correlation coefficient for measuring the linearity between two variables:

$$\rho_i(up_i, vp_i) = \begin{cases} MAX\left(MIN\left(\frac{1}{max_i}, 1\right), 0\right) & When\ up_i\ type\ is\ enumration \\ MAX\left(MIN\left(\left|\frac{up_i - vp_i}{max_i - min_i}\right|, 1\right), 0\right) & otherwise \end{cases}$$

i: a dimension or key of the user profile.

up_i, vp_i *:* the user value and the generic value of the corresponding dimension.

max_i, min_i *:* the maximum and minimum values of the corresponding dimension.

All generic profiles' distances from the current user profile are calculated. The obtained list of generic profiles are then sorted based on their distance. These are fed to the next step, additive suggestions, to infer personalized suggestions.

3.1.1.4 Inference

The statistical matchmaker follows an additive suggestions method to extract user related suggestions from all similar generic user profiles founded. In additive suggestions, all the similar generic user profiles are taken into account for the inferred user preference suggestions.

On each matching request, the list of similar generic user profiles is found and ordered by their distance to the current use profile. From the list, a new additive preference set is generated, that contains preferences from all generic preferences, weighted by their distance to the current preference set. This allows to better pinpoint the unique distances of the current preference set and nearby clusters and usually generates preferences that are more accurate.

3.1.2. **Rules refinement**

The EasyTV statistical matchmaker is able to properly refine and further extend the rules supported by the rule-based matchmaker by analyzing the needs, preferences and interactions of large sets of users. The EasyTV rule-based matchmaker initially supports a set of rules provided by domain experts. These rules are properly adjusted by the statistical matchmaker as described in the following paragraphs.

3.1.2.1 Semantic rules

The rule based matchmaking approach is based on semantic web technologies. The domain ontology defines the concepts and relationships used to describe and represent an area of concern. Following a first-order logic (FOL) syntax, the domain ontology includes the basic symbols constants, predicates and functions. Constant symbols represent objects, which are the central domain elements. Predicate symbols stand for relations among objects or properties of objects. Function symbols are interpreted as relations with exactly one value for a given input. In other terms, the domain ontology specifies the terms used in declaring statements.

Semantics rules are of the form of an implication between an antecedent (body) and consequent (head), whenever the conditions specified in the antecedent hold, then the consequent's conditions must also hold. For example, in FOL a rule that state that "the colored subtitle accessibility feature of type *colorset1* is used by users who requested *subtitles* and have a color-blindness condition of type *deuteranopia*" is:

$$\forall x \; HasSubtitles(x) \wedge HasColorBlidness(x, deuteranopia) \\ \implies HasAccessibilityFeature(x, colorSubtitles) \wedge SetSubtitleColor(x, ColorSet1)$$

Based on this rule, we can suggest for users that requested subtitles and have declared color-blindness condition of type *deuteranopia* the colored-subtitle accessibility feature with color set of type *colorSet1*.

3.1.2.2 Semantic rules refinement

The statistical matchmaker uses associative rules in the semantic rules refinement process. Associative rules are extracted from the available data with the help of associative learning algorithms, which are rule-based machine learning methods for discovering interesting relations between variables in large databases [15]. Two metrics are used to extract associative rules: support and confidence [16]. Support, is an indication of how frequently an item appear in a dataset, the support of x with respect to database entries T is the number of entries in T that contains x. Confidence, on the other hand, is an indication of how often a rule is found to be true. The formula of support and confidence are:

$$supp(x) = \frac{|\{t \in T; x \in t\}|}{|T|}$$

where:

T: set of transactions.

x: an itemset.

$supp(x)$: the support of x in respect to T

$$conf(x \rightarrow y) = \frac{supp(x \cup y)}{supp(x)}$$

where

x, y: itemsets.

$conf(x \rightarrow y)$: confident of the rule.

These associative rules are then converted to semantic rules. The process is straightforward and done with the help of the operand ⟹ the universal quantifier ∀ and the connection ∧ , and functions. The results of the whole process is actually converting associative rules into semantic rules. For example, the conversion of the following associative rule:

$$\{age = 70, geneder = male\} \rightarrow \{\text{accessibility feature} = Avatar\}$$

To the following rule:

$$\forall x\ Age(x) = 70 \wedge Gender(x) = male \Longrightarrow HasAccessiblityFeature(x, Avatar)$$

Domain experts defines rules that are similar to the rules founded by associative rules. What characterizes these rules is that they know which properties must be associated and what their corresponding values must be. In other words, they represent accumulated knowledge in specific domain.

Associative learning algorithm

Associative learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. The concept of associative learning is to discover and extract the associative set of rules. A formal definition is the following:

Let $I = \{ i_1, i_2, \ldots\ldots, i_k \}$ be a set of binary attributes called *items*.

Let $D = \{ t_1, t_2, \ldots\ldots, t_k \}$ be a set of transactions called the *database*.

Each one of the transactions in D is considered unique and contains a subset of the items in I. A association rule is an implication of the form:

$$X \Longrightarrow Y \ where \ X, Y \subseteq I$$

Multiple associative algorithm exists that extract these associative rules. Some well know algorithms are Apriori [17], Eclat [18], FP-growth [19]. All of them are based on the idea of finding all supported itemsets (frequent itemsets) which are then combined to produce associative rules. Itemset is a set of N items that have proper support. The main idea behind all associative rules algorithms is to recursively find itemsets of *N* items by combining the itemsets of the pervious step of *N-1* items and checking their support. This process stops when there are no more supported itemsets or all itemsets combination has been exhausted.

The statistical matchmaker uses the FP-growth algorithm for extracting associative rules. FP-growth uses special data structure called FP-tree that help identifying and purging unfrequented itemsets early in each step. In the first pass, the algorithm counts occurrence of items in the dataset, and stores them in a special table the 'header table'. In the second pass, it starts building

the FP-tree by inserting sorted list of items called instances. Items in each instance are sorted by their frequency in the dataset. Instance's Items with minimum coverage threshold are purged. FP-tree provides high compression close to the root in case many instances share most frequent items. Figure 2 shows the process of constructing FP-tree from a set of instances.

TID	Items
1	{ age= 70, gender= male, sign language= on}
2	{age= 60, gender= male, sign language= off}
3	{age= 70, gender= male, sign language= on}
4	{age= 44, gender= female, sign language= on}
5	{age= 70, gender= male, sign language= on}
6	{age= 70, gender= female, sign language= on}

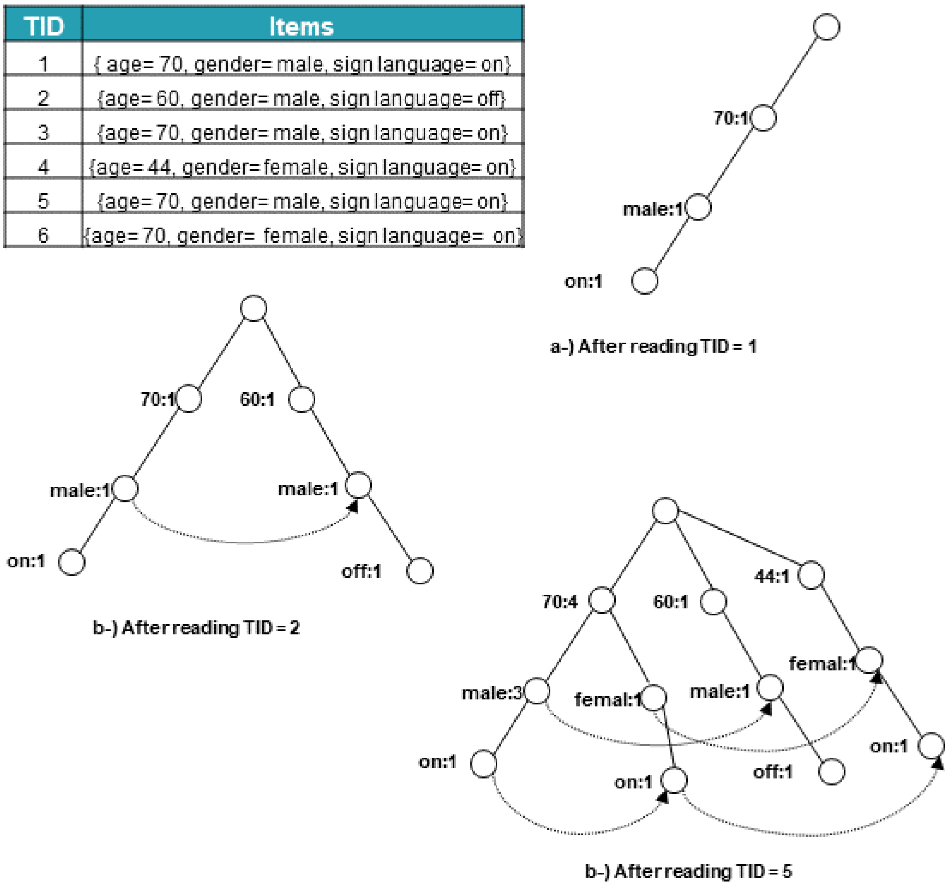

Figure 2 FP-tree construction

Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins [20]. To generate association rules from frequent itemsets two steps are required. First for each frequent itemset X we generate all its non-empty subsets and then for each non-empty subset S of itemset S we generate the following AR $S \rightarrow (X - S)$. Only the AR that has the required confidence is kept.

TID	Items
1	{ age= 70, gender= male, sign language= on}
2	{age= 60, gender= male, sign language= off}
3	{age= 70, gender= male, sign language= on}
4	{age= 44, gender= female, sign language= on}
5	{age= 70, gender= male, sign language= on}
6	{age= 70, gender= female, sign language= on}

Min support = 60%

Frequent Itemset	Support
{ age= 70}	66%
{gender= male}	66%
{sign language= on}	83%
{age= 70, gender= male}	66%
{age= 70, sign language= on}	66%
{gender= male, sign language= on}	66%
{ age= 70, gender= male, sign language= on}	66%

Min confidence = 75%

Rule	Conf
{ age= 70, gender= male => sign language= on}	100%
{ age= 70 => gender= male ,sign language= on}	100%
{ gender= male => age= 70 ,sign language= on}	100%
{ gender= male => sign language= on}	100%
{sign language= on => gender= male}	100%
{age= 70 => sign language= on}	100%
{sign language= on => age= 70 }	100%
{ age= 70 => gender= male}	100%
{gender= male => age= 70}	100%

Figure 3 Association rules extraction

Figure 3 illustrate the process of extracting association rules from a minimum user profiles repository. The user profile in the figure shows only a sub section of the user profile feature for illustration purposes. It must be noted here that not all AR are relevant to the matchmaking process, some of these associate accessibility features with general user profile characteristics, for example the AR {gender= male => sign_language= on}.

4. HYPER SPERSONALISATION FRAMEWORK ARCHITECTURE

This section describes the EasyTV system architecture with special focus on the hyper-personalisation 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.

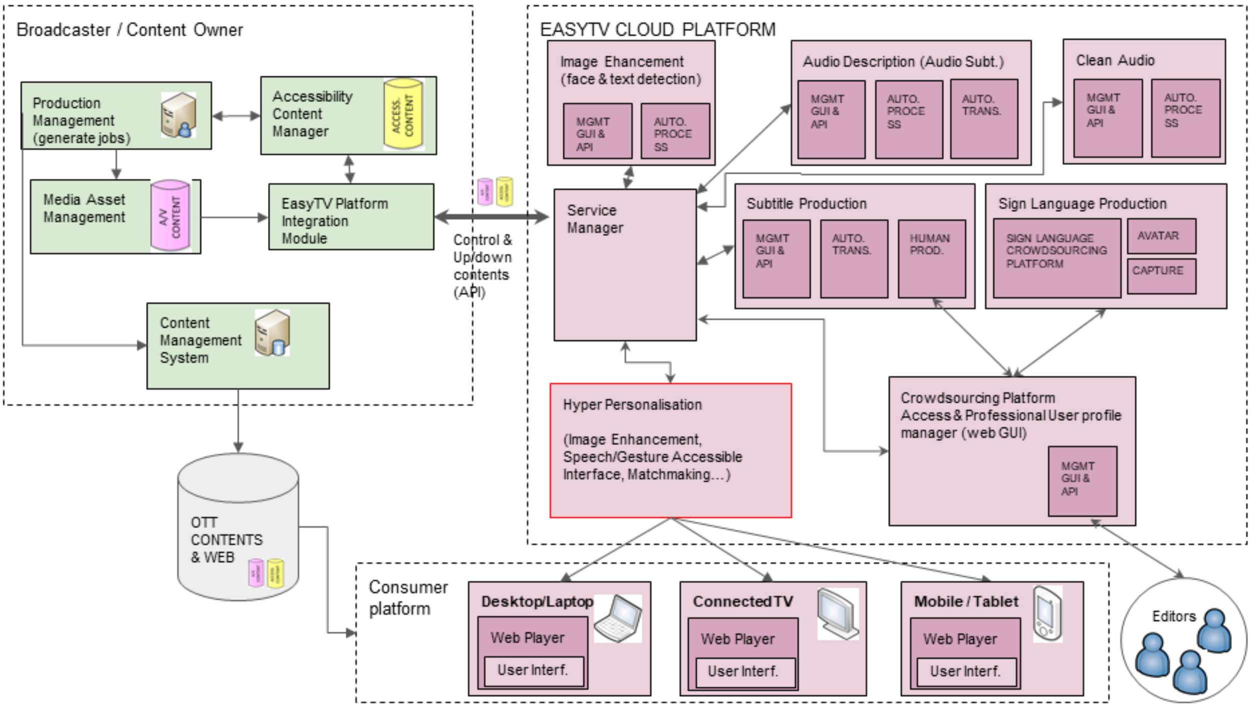

Figure 4 EasyTV system architectural overview [21]

The place of the Hyper-personalisation module inside the EasyTV platform and its interaction with other EasyTV components is shown in Figure 4. The following paragraphs provide more details related to the inner components of the hyper-personalisation module.

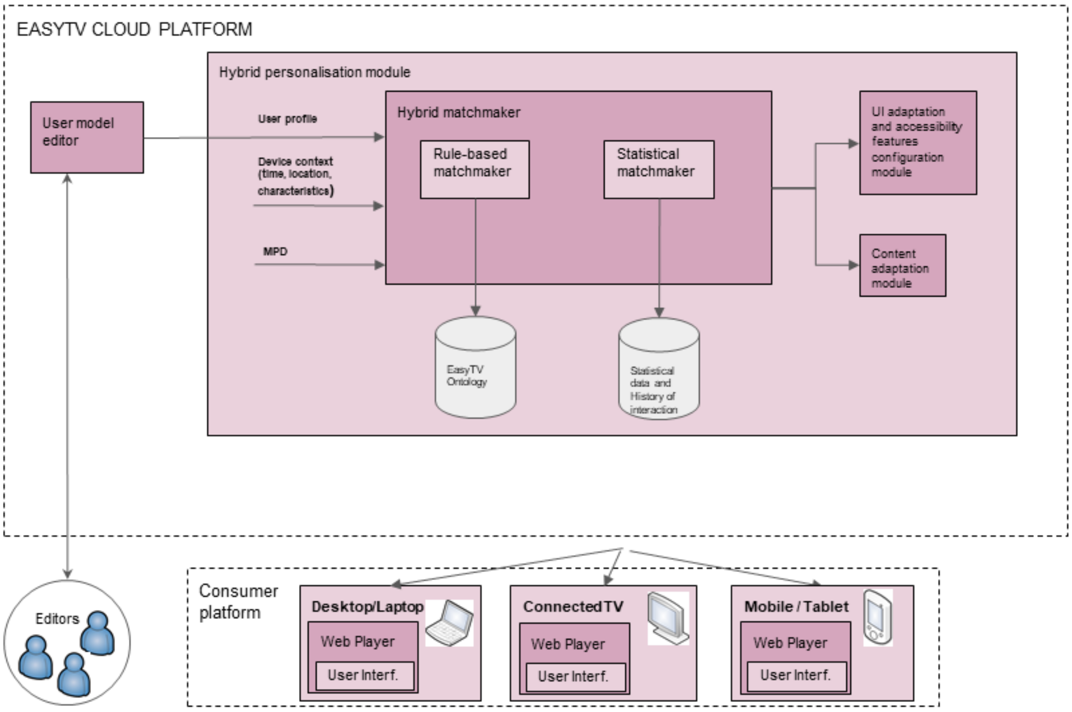

Figure 5 Hybrid hyper-personalisation module overview

The hybrid hyper-personalisation module consists of the following components as shown in Figure 5:

- **User profile**: The EasyTV user profile defines user needs and preferences, including also possible disabilities and functional limitations of the user. The structure of the EasyTV user model is based on previous work conducted in previous projects, such as Cloud4All and VERITAS, as well as on the user models proposed by the VUMS cluster. It is also partially based on popular standards such as ISO/IEC 24752-8:2018. The easy creation and editing of the EasyTV User Models is enabled through intuitive web forms offered by the User model editor developed in T4.1.
- **User model editor**: The EasyTV User Model Editor will be a web-based tool that enables the easy creation and editing of EasyTV end user models through intuitive web forms.
- **Device context**: Device contextual information is essential to the matchmaking process, as user may declare conditional preferences related to a specific device context. That includes information related to the device specifications (operating system, version, and installed applications), location and time. During the matchmaking process, the current device context information is used to find the best matching setting of the user preference.
- **MPD**: Following DASH streaming convention, the played audio-visual content information are located in special files called media presentation description. These files are XML document containing information about media segments, their relationships and information necessary to choose between them, and other metadata that may be needed by clients. For each available audio-visual content, a related MPD file exists.
- **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:
    - **Rule-based matchmaker:** The rule-based matchmaker performs matchmaking on the content and metadata stored in the EasyTV ontologies by applying semantic rules. The implementation of the EasyTV rule-based matchmaker is mainly based on the Cloud4All rule-based matchmaker [22]. The rule-based matching is a context-based in the sense that it infers the proper configuration regarding all three services functionalities, defined in Table 1.
    - **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 (as presented in Section 3) 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. This will eventually lead to a self-learning system that evolves and fine-tunes its personalisation capabilities over time by taking hundreds of thousands of individual user experiences into account (i.e. tapping into the "wisdom of the crowd"). The implementation of the statistical matchmaker is based on related results coming from previous projects such as Cloud4All, PROSPERITY4ALL, IN LIFE, etc.
- **UI adaptation and accessibility features configuration module**: This module takes as input the results of the hybrid matchmaker and performs automatic turn on and configuration of accessibility features (e.g. volume, rate, pitch, colour preferences, etc.) that are built into different TV operating systems, applications and embedded devices that are be supported in the EasyTV project. This mechanism is mainly based on the Cloud4All auto-personalisation framework, which supports recommendation and automatic launching of assistive technologies (e.g. screen readers, magnifiers, etc.) along with automatic adjustment of corresponding settings (e.g. speech rate, magnification factor, etc.) by applying rule-based and statistical matchmaking on user profile,

application capabilities/available settings and current context (e.g. device/OS used, environmental factors, etc.).

- **Content adaptation module**: The EasyTV content adaptation module performs content adaptation based on the results of the hybrid matchmaker, in order to offer streaming content in the best possible form for a specific user (e.g. by selecting the audio and/or subtitles that correspond to user's language and current device context).

4.1. Statistical Matchmaker

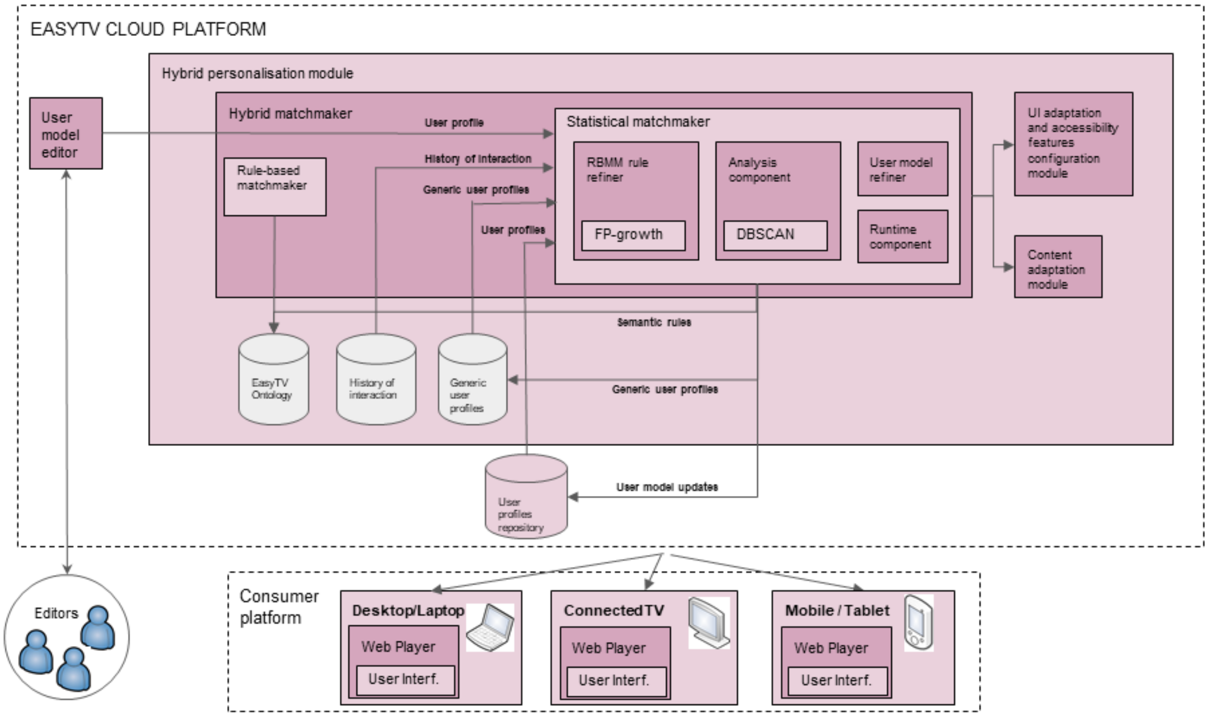

Figure 6 Statistical matchmaker overview

The statistical matchmaker is an inner component of the hybrid matchmaker that implements collaborative filtering (as presented in Section 3). This section describes the components that constitute the statistical matchmaker, its input/output data and its interaction with other EasyTV components as illustrated in Figure 6.

4.1.1. Input of the statistical matchmaker

4.1.1.1 User profile

A user profile represents a user of the EasyTV platform. It includes information regarding the user's demographic, visual and audio functional capabilities and preferences (both generic and conditional). A visual description of the user model schema is shown in Figure 7. In addition to a close up look to the user preference, structure that is shown in Figure 8.

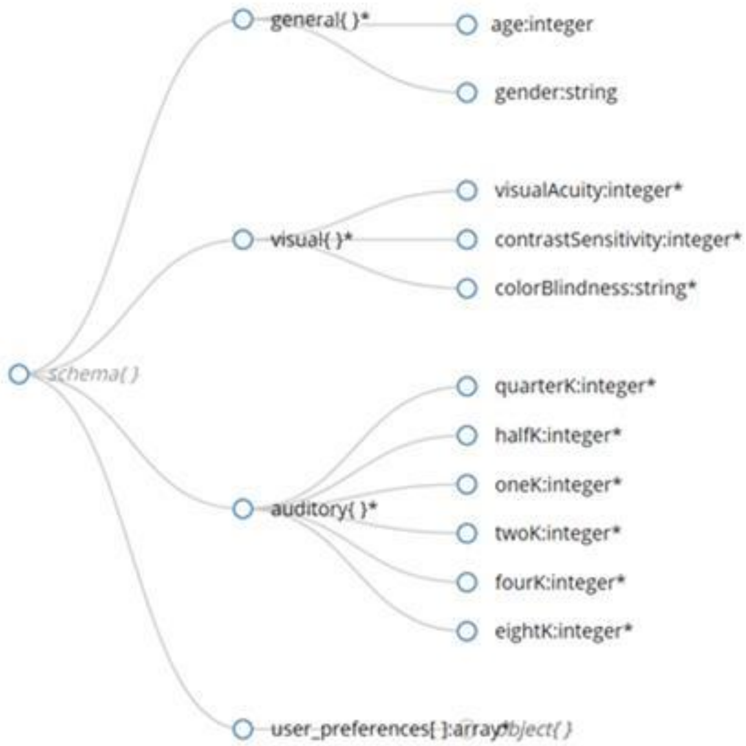

Figure 7 A visual description of user model schema

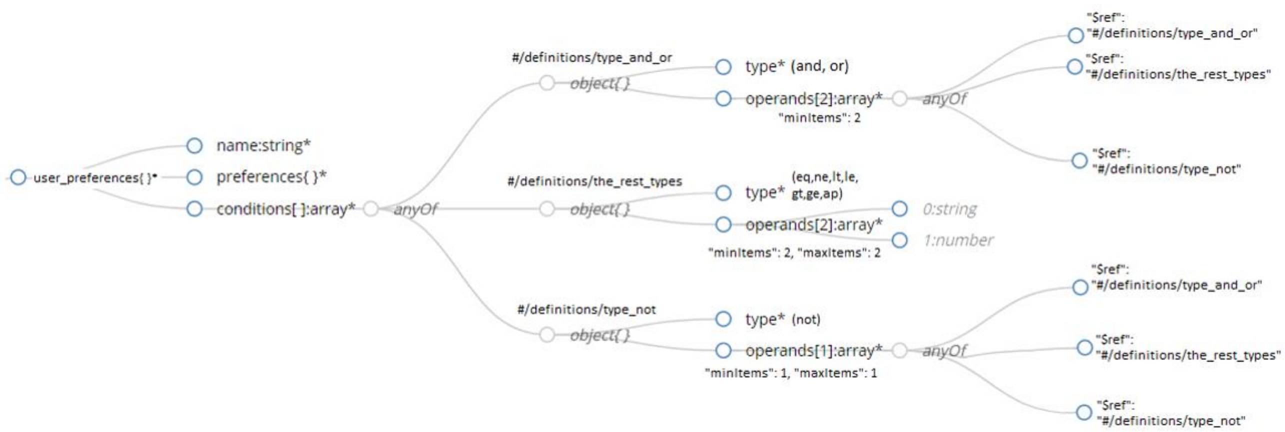

Figure 8 A visual description of the conditional user preferences schema

A user can declare his/her preferences regarding the UI adaptation, the audio-visual content adaptation and the accessibility feature configurations in the user preferences section of the user model. These can be of two types, generic and conditional. Generic preferences include preferences that should be applied generally (not within a specific context), while conditional preferences are associated with a condition and are applied only when the associated condition is fulfilled. When no conditional preference can be applied, the generic preferences are applied.

Supported preference by the current user model are shown in Table 2.

Table 2 User preferences supported by the current structure of the EasyTV User Model

Preference	Action	Description
Audio	Language	Set the language of the played audio/visual content
Subtitle	Set on/off	Enable/disable subtitles
	Language	Set the subtitles language of the played audio-visual content
Sign language	Set on/off	Enable/disable sign language
	Language	Set the sign language of the played audio/visual content
Display	Contrast	Set the value of the displayed contrast colors
	Color temperature	Set the value of the displayed color temperature
Font	Size	Set the font size
High contrast	Mode	Set the high contrast mode
Screen-Reader	Set on/off	Enable/disable screen-reader
	Language	Set the voice language of the screen-reader
	Speed	Set the reading speed of the screen-reader
	Advanced user	Enable/disable more contextualization of the selected content.

Table 3 lists the user model main classes and properties with a short description and their acceptable range of values.

Table 3 EasyTV user model-Main classes/properties [23].

User model element	Short description
Visual functional capabilities	
visual_acuity, contrast_sensitivity	Describes the level of user requirements for bigger font sizes and bigger graphics on a user interface. Smaller values of this variable are translated to larger size for the various visual elements displayed (for example, larger font size). Allowed values for visual acuity are between 1 until 20.
color_blindness	Ability to distinguish colors (without any limitation such as e.g. red green, blue yellow blindness). One of several types : 1. **Normal vision** 2. **Deuteranomaly**: partial insensitivity to green. It is the more common color-blindness deficiency. 3. **Deuteranopia**: total insensitivity to green. 4. **Protanomaly**: partial insensitivity to red.

	5. **Protanopia**: total insensitivity to red. 6. **Tritanomaly**: partial insensitivity to the blue. This vision deficiency is very rare. 7. **Tritanopia**: total insensitivity to the blue. This vision deficiency is very rare.
Auditory functional capabilities	
quarterK, halfK, oneK, twoK, fourK, eightK	Threshold hearing level in dB at 250, 500, 1000, 2000, 4000, 8000 Hz. For the user model, we consider this threshold after the possible compensation from a hearing aid. Allowed values are from the range 0-90 db.
User preference	
user_preferences	Describes the device settings the user wishes, possibly taking effect after some conditions are met.

Table 4 show an example of a complete user profile based on the current version of the EasyTV user model. This user profile represents a 40-year old male with sight difficulties and with quiet normal auditory functional capabilities. His default preference set indicates that he speaks Spanish and knows Spanish sign language. He prefers large font size and yellow colored subtitles, which he sets to Spanish. Moreover, he prefers a screen-reader enabled with beginner mode with reading speed of four and Spanish language. His preference set also includes two conditional preferences, one that declares English subtitle and sign language when there is no Spanish sign language. The second conditional preference enables higher contrast at night, from 17:00 until 23:00 a clock.

Table 4 User model example

```
{
  "general": {
    "age": 40,
    "gender": "male"
  },
  "visual": {
    "visual_acuity": 8,
    "contrast_sensitivity": 24,
    "color_blindness": “Normal vision”
  },
  "auditory": {
    "quarterK": 81,
    "halfK": 35,
    "oneK": 98,
    "twoK": 18,
    "fourK": 57,
    "eightK": 27
  },
  "user_preferences": {
    "default": {
      "preferences": {
```

```
      "contrast": 60,
      "font_size": 20,
      "color_temperature": 0.008,
      "language_subtitles": "spanish",
      "language_sign": "spanish",
      "language_audio": "spanish",
      "backgroundColor": "black",
      "subtitle_position": "bottom",
      "fontColor": "yellow",
      "screen_reader_language": "spanish",
      "screen_reader_advanced_user": false,
      "screen_reader_speed": 4
    }
  },
  "signLanguageAlternatives": {
    "preferences": {
      "language_subtitles": "english",
      "language_sign": "english"
    },
    "conditions": [
      {
        "type": "ne",
        "operands": [
          "language_sign",
          "spanish"
        ]
      }
    ]
  },
  "night_light_mode": {
    "preferences": {
      "contrast": 90
    },
    "conditions": [
      {
        "type": "and",
        "operands": [
          {
            "type": "ge",
            "operands": [
              "time",
              "17:00"
            ]
          },
          {
            "type": "le",
            "operands": [
              "time",
```

```
                "23:00"
              ]
            }
          ]
        }
      ]
    }
  }
}
```


4.1.1.2 User profile repository

A repository of all EasyTV users' profiles. It is a database that allows easy retrieval of users' profiles. As indicated in D5.1 *"Mid-term report on the set up and implementation of the EasyTV multi terminal technical platform*" the database is MongoDB, a NoSQL JSON oriented database.

4.1.1.3 Generic user profiles

A repository of generic user profiles. Generated by the analysis component as a result of the clustering phase (presented in Section 3.1.1.1) and generic user profile phase (presented in Section 3.1.1.2). It is the output of the analysis component and the input of the runtime component.

4.1.1.4 History of interactions

A database of users' modifications actions over their initial user profile. The user's interaction with the platform is taken in relation to a specific content, such as the played audio-visual content, the device specifications, location and time. The platform keeps this information in a separated data based that uses for statistical inference by the analysis component.

One of the main sources of user history of interaction is the EasyTV CS App. The way a user interacts with an app is useful to understand user's behavior. Regarding people with disabilities, it is even more important to know how they interact with the app. For example to know which parts of the app trouble them, in which ones they have difficulties or which settings fit them better for their needs. When using an app, a user produces a lot of information directly or indirectly, such as where is he taping, which content is he watching, which tools is he using or which settings preferences is he using. All this information is fed to the statistical analysis and machine learning algorithms in order to make behavior predictions and personalisation suggestions.

4.1.2. **Output of the statistical matchmaker**

4.1.2.1 Readjustment of the User Model

Use model is refined based on similar users' preferences, interaction history of the current user and the interaction history of similar users. Section 3.1.1 describe analytically the whole process.

4.1.2.2 Readjustment of the rules supported by the EasyTV Rule-based Matchmaker

Readjustment of rule-based matchmaker's rules includes adding, editing or removing rules. Rules to be refined are rules with low confidence. That indicates that these rules may not be valid/accurate. Rules with low confidence is of no use for users and must be refined or removed. Section 3.1.2 describes analytically the process of rules refinement, in addition to extracting new rules from available users' profiles.

4.1.3. **Analysis Component**

The Statistical Matchmaker uses statistical analysis to find and exploit correlations between the users' profiles. The clustering phase corresponds to grouping all similar set of user profiles in

groups called clusters. These are used in the next phase to generate generic user profiles. Our criteria for selecting a clustering algorithm:

- **Undefined number of clusters**: considered the most important selection criteria. The algorithm should not need to know in advance the number of clusters at which it should stop.
- **Unspecified clustering shape**: The algorithm should be able to form a cluster of an arbitrary shape, not only form spherical or ellipsoid shapes.
- **Noise handling**: The algorithm should be able to identify and handle noise (i.e single user profile sets that are very different form all others).

Several algorithms are available to identify an unknown number of clusters from high dimensional preference sets by just using a similarity measure as mentioned is section 3.1.1.1. The algorithm must be able to decide where to stop, because the number of clusters cannot be determined in advance.

The first criteria quickly excludes K-means algorithm from our list, because the number of generated cluster must be feed to the algorithm. The Mean-shift algorithm, on the other hand, does not require the number of clusters to be found, however, it requires the radius “r” of each founded cluster. The later does not satisfy our second criteria and thus the Mean-shift is excluded. The DBSCAN algorithm was chosen for this prototype as it creates a clusters of unspecified shapes and matches all the given set of criteria.

4.1.4. Runtime Component

An inner component of the statistical matchmaker that has an access to the database generated by the analysis component. It processes the matchmaking queries to find similar user profiles and infer proper personalized suggestions to the user. It uses computationally cheap algorithms to optimize runtime performance. Figure 9 sums up the runtime component handling of matching requests.

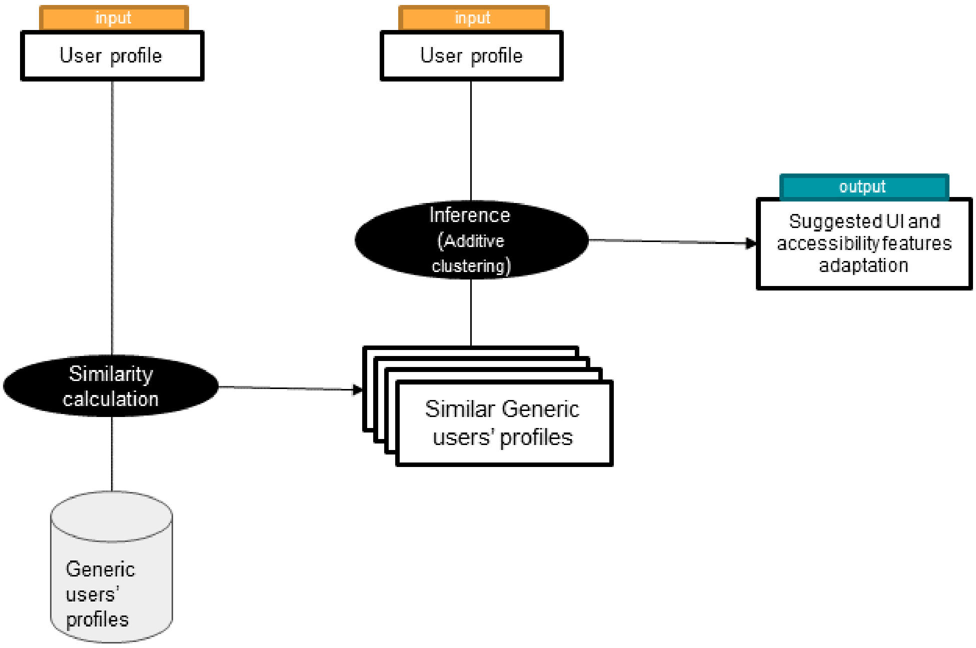

Figure 9 Overview of runtime component handling of matching requests

On each matching request the runtime component, access the generic user profiles database to

find the nearest generic user profiles. This step corresponds to similarity calculation analyzed in section 3.1.1.3. Distance between user profiles, generic and current user, are measured using Pearson distance formula. Additive clustering, section 3.1.1.4, is used to infer suggestions related to the current user. In this step, the current user preference set is taken into account. The generated output is a set of suggestions that refer to the UI and accessibility features adaptation configuration that are tailored to the current user profile.

4.1.5. RBMM rule refiner

RBMM rule refiner requires access to two data sources, the EasyTV ontology and the users' profiles database. It runs on regular basis, for instance once a day, associative learning algorithm over the users' profiles database to extracts associative rules. For this process, it uses the well-known FP-growth algorithm described in section 3.1.2.2. These associative rules are actually supported association relationship between a set of keys-values pairs of user profiles. Supported in the sense that the relationship occurs in more than *N* number of available user profiles.

After extracting these rules, the next step is to read the rule-based matchmaker rules from the EasyTV ontology (Figure 6). Each one of these rules is then compared with all extracted associative rules and the best match is found. A best match is actually an associative rule that has the most common terms with the semantic rule, comparing their heads and body respectively. The platform admin confirmation is required for the new and updated rules to be finally integrated into the EasyTV ontology.

4.1.6. User model refiner

The suggestions accepted by the user are reflected into his/her user profile. After presenting the suggestions to user and asking his/her confirmation all accepted suggestions are incorporated into his/her profile. That will end up refining the user profile by adding, removing or editing his/her preference set, both the default and conditional ones. New condition may end up being added, removed or even edited. Additional source of suggestions that must be reflected into the user profile is the pattern extracted from the user history of interaction.

5. INDICATIVE USE CASES

This section provides some indicative use cases towards a better understanding of the matchmaking process.

5.1. User model refinement for a user with visual impairments

A screen-reader is an accessibility solution that converts digital text into synthesized speech, empowering users to hear content and make possible browsing. The EasyTV platform offers a screen-reader as an accessibility feature with the ability of setting the generated audio speed and audio language of the screen reader (Table 2). The main objective of the service is to give feedback on where the user has the focus inside the application interface, and report on the section and the title of the video where he is browsing. HbbTV or the TV manufacturers itself, do not offer a screen-reader solution for the applications that run on a smartTV set. As consequence, people who are blind or who have low vision to use information technology, or with certain cognitive or learning disabilities does not have the same level of independence and privacy as anyone else.

A user with low vision may not be able to configure or even set the screen reader service. However, the available users' data indicates that users with similar vision capabilities have been using this service with specific speed and language for the generated audio. The hyper-personalisation module, through statistical analysis and based on the available data, infers from similar users (regarding their vision capabilities) the proper screen-reader configuration. The hyper-personalisation proposes to the user enabling the screen-reader service in beginner mode and setting a specific audio speed configuration. After accepting the suggestion, a new screen-reader

preference is added to the user preference set. Table 5 shows the refined user model, the new added suggestions (screen-reader accessibility feature configuration) are indicated with bold.

Table 5 User model refinement example


```
{
  "general": {
    "age": 40,
    "gender": "male"
  },
  "visual": {
    "visual_acuity": 12,
    "contrast_sensitivity": 24,
    "color_blindness": “Normal vision”
  },
  "auditory": {
    "quarterK": 81,
    "halfK": 35,
    "oneK": 98,
    "twoK": 18,
    "fourK": 57,
    "eightK": 27
  },
  "user_preferences": {
    "default": {
      "preferences": {
        "contrast": 75,
        "font_size": 25,
        "color_temperature": 0.009,
        "language_subtitles": "spanish",
        "language_sign": "spanish",
        "language_audio": "spanish",
        "backgroundColor": "black",
        "subtitle_position": "bottom",
        "fontColor": "white",
        "screen_reader_language": "spanish",
        "screen_reader_advanced_user": false,
        "screen_reader_speed": 4
      }
    }
  }
}
```


5.2. Refinement of rules for people with color-blindness

The EasyTV platform offers colored subtitles as another accessibility feature. It aims at helping people with color blindness. Color-blindness is a vision deficiency that causes a decreased ability to see or to differentiate colors to the affected people. It is not an illness; color-blindness is a

condition found in people with limitations in one or more of the three sets of color sensing cones in the eye, and it is usually inherited genetically. In this sense, color-blindness is more likely to happen to males than females, so it is found in about the 8% of the men and only in the 0,5% of the women [24].

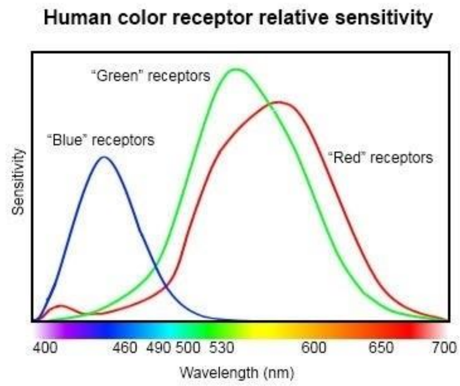

Figure 10 Human color receptor relative sensitivity

In the context of the EasyTV project, two sets of alternative colors (to Spanish subtitles regulation), have been designed to improve the intelligibility of the subtitles when used by color-blind people:

- one color set is adapted to the needs of with deuteranomaly, deuteranopia, protanomaly and protanopia (it has been established in the simulations that both deficiencies result in a similar perception)
- second color set is suitable for tritanomaly and tritanopia .

The color composition of each set is shown in the following figure.

Default Color	Color Set 1	Deuteranomaly simulation	Protanomaly simulation	Color Set 2	Tritanomaly simulation
White (#FFFFFF)	White (#FFFFFF)	EasyTV	EasyTV	White (#FFFFFF)	EasyTV
Yellow (#FFFF00)	Yellow (#FFFF28)	EasyTV	EasyTV	Cyan(#64FFFF)	EasyTV
Aqua (#00FFFF)	Blue (#9696FF)	EasyTV	EasyTV	Yellow (#FFFF00)	EasyTV
Green (#00FF00)	Green (#28FF00)	EasyTV	EasyTV	Red (#FF0000)	EasyTV
Fuchsia (#FF00FF)	Celadon (#5AFFA0)	EasyTV	EasyTV	Grey (#969696)	EasyTV
Red (#FF0000)	Grey (#969696)	EasyTV	EasyTV	Green (#00FF00)	EasyTV

Figure 11 Default and Color-blindness set for subtitling

A set of available user's profile that indicate that the majority of users with Protanopoia color-blindness type condition have enabled colored-subtitles with value set to *colorSet1*, will cause the statistical matchmaker to infer the following association rule during the association learning phase:

$$\{color\ blind\ = Protanopoia\} \rightarrow \{\text{color subtitle} =\ colorSet1\}$$

The rules refinement process ends up adding the following rule to the RBMM set of rules.

$$\forall x \rightarrow HasSubtitles(x) \wedge HasColorBlidness(x, protanopoia)$$
$$\Longrightarrow HasAccessibilityFeature(x, colorSubtitles) \wedge SetSubtitleColor(x, ColorSet1) \wedge HasSubtitles(x)$$

The rule suggests to users with Protanopoia, which have no preference for subtitle, to enable colored subtitles with value set to *colorSet1*. That results in hyper-personalization module making colored-subtitles personalized suggestion to users with Protanopoia color-blindness condition.

6. CONCLUSION AND FUTURE WORK

The EasyTV hyper-personalisation module uses two well-known approaches for matchmaking, a rule-based and a statistical approach. These two approaches are applied independently but finally their output is combined in a hybrid manner. We chose the dynamically adjusted way as it is widely adapted and proposed.

Towards going one step further in improving the accuracy of the whole personalisation process the statistical matchmaker exploits the user-generated data to increase the accuracy of the rule-based approach. In this way, we leave the available data to speak for itself. These rules are well-known to field experts, which know what attributes to be associated and their values. The extracted rules from this process is expected to increase and revaluate rules written by field experts. In addition, the hybrid matchmaking suggestions gradually refine the user model in order to better match with his/her needs and preferences. Following this approach, available users profiles, history of interaction and preferences are not only used to increase the accuracy of the statistical inference suggestions but also used to improve the accuracy of the knowledge based (rule-based) suggestions.

One of the main tasks for future work is to incorporate the work done in T4.1 "*Adaptive menus and graphical interface using user models*", which is related to the enrichment of the user profile properties. The current version of the user profile support only a subset of EasyTV available features and UI adaptation. As UI configurations of TV operating systems and EasyTV accessibility features are being further developed, a wider range of preferences will be taken into account by the EasyTV hybrid matchmaker towards a further refinement of the whole personalisation process.

The next steps towards the improvement of the accuracy of the EasyTV personalization process will be presented in detail in the revised version of this deliverable, which is D4.5: *EasyTV self-learning system for improving personalisation capabilities (revised version)* [M25].

REFERENCES

[1] F. Ricci, L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook,, Springer, 2011, pp. 1-35.

[2] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems 46 (2013) 109–132, 2013.

[3] A. Popescul, L. H. Ungar, D. M. Pennock and S. Lawrence, “Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments,” in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

[4] G. Linden, B. Smith and J. York, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” *IEEE Internet Computing,* pp. 76-80, January-February 2003.

[5] A. Stiegler, C. Loitsch, N. Kaklanis, C. Strobbe, K. Votis, L. Makris, V. Pavlopoulos, G. Stavropoulos, V. Kilintzis and D. Tzovaras, "Matchmaking Algorithms and Their Evaluation," Cloud4All D204.3, 2015.

[6] N. Kaklanis, S. Stavrotheodoros, P. Ioannis, D. Tzovaras and A. Medrano, "Deliverable 2.4 System_architecture_and_tools_revision2," 2017.

[7] Matthias Peissner, Dagmar Häbe, Doris Janssen and Thomas Sellner, "Generating Accessible User Interfaces from Multimodal Design Patterns," in *4th ACM SIGCHI Symposium on Engineering Interactive Computing*, 2012.

[8] Stefano Ferretti, Silvia Mirri, Catia Prandi and Paola Salomoni, "Automatic Web Content Personalization Through Reinforcement Learning," *Journal of Systems and Software,* 2016.

[9] Yang Yang, Niamh Caprani, Adam Bermingham, Julia O’Rourke, Rónán Collins, Cathal Gurrin and Alan F. Smeaton, "Design and Field Evaluation of REMPAD: A Recommender System Supporting Group Reminiscence Therapy," 2013.

[10] T. Kanungo, D. M.Mount, N. S.Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An Efficient k-means clustering algorithm: Analysis and implementation," 2002.

[11] V. Estivill-Castro, "Why so many clustering algorithms – A Position Paper," *ACM SIGKDD Explorations Newsletter,* 20 June 2002.

[12] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE, 1996.

[13] E. Martin, H.-P. Kriegel, S. Jörg and X. Xiaowei, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[14] R. A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis, 4th Edition, 1998.

[15] "DeepAI," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/association-learning.

[16] J. Han and M. Kamber, Data mining concepts and techniques, 3th edition, 2000.

[17] A. Rakesh and S. Ramakrishnan, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, Santiago, Chile, September 1994.

[18] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering,* 2000.

[19] Han, "Mining Frequent Patterns Without Candidate Generation," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD , 2000.

[20] Witten, Frank and Hall, Data mining practical machine learning tools and techniques, 3rd edition, 2005.

[21] "D1.4 Final release of the EasyTV system architecture".

[22] 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,.

[23] N. Kaklanis, P. Biswas, Y. Mohamad, M. F. Gonzalez, M. Peissner, P. Langdon, D. Tzovaras and C. Jung, "Towards standardization of user models for simulation and adaptation purposes".*Towards standardization of user models for simulation and adaptation purposes.*

[24] "Facts About Color Blindness," February 2015. [Online]. Available: https://nei.nih.gov/health/color_blindness/facts_about. [Accessed 29 July 2016].

[25] 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.