Deliverable 4.4 User profile and contextual adaptation

Dorothea Tsatsou, Pantelis Ieronimakis, Dimitrios Panagiotou, Vasileios Mezaris (CERTH)

Tomáš Kliegr, Jaroslav Kuchař (UEP)

Matei Mancas, Juliens Leroy (UMONS)

Rüdiger Klein, Manuel Kober, Jingquan Xie (FRAUNHOFER)

15.10.2013

Work Package 4: Contextualisation and Personalisation

LinkedTV

Television Linked To The Web

Integrated Project (IP)

FP7-ICT-2011-7. Information and Communication Technologies

Grant Agreement Number 287911

Dissemination level[1]	PU
Contractual date of delivery	30th September 2013
Actual date of delivery	15th October 2013
Deliverable number	D4.4
Deliverable name	*User profile and contextual adaptation*
File	*LinkedTV_D4.4.doc*
Nature	*Report*
Status & version	v1.0
Number of pages	*79*
WP contributing to the deliverable	*WP 4*
Task responsible	*CERTH*
Other contributors	*UEP* *UMONS* *FRAUNHOFER IAIS*
Author(s)	*Dorothea Tsatsou, Pantelis Ieronimakis, Dimitrios Panagiotou, Vasileios Mezaris* *CERTH* *Tomáš Kliegr, Jaroslav Kuchar* *UEP* *Matei Mancas, Juliens Leroy* *UMONS* *Rüdiger Klein, Manuel Kober, Jingquan Xie* *FRAUNHOFER*
Reviewer	*Jan Thomsen, CONDAT*

[1] • PU = Public
• PP = Restricted to other programme participants (including the Commission Services)
• RE = Restricted to a group specified by the consortium (including the Commission Services)
• CO = Confidential, only for members of the consortium (including the Commission Services)

EC Project Officer	*Thomas Kuepper*
Keywords	*Ontology, semantic user model, attention tracking, context tracking, preference learning, association rules, user modelling, context*
Abstract (for dissemination)	*This deliverable presents the methods employed in LinkedTV to create, update and formalise a semantic user model. In addition, the first approach on extraction of context and contextual features and its adaptation onto the semantic user profiles is presented.*

Table of contents

List of Figures

List of Tables

1 Introduction

LinkedTV's personalisation and contextualisation services involve capturing, learning and creating semantic user profiles based on the user's online behaviour with networked media and on user-defined preferences expressed through dedicated LinkedTV utilities. These profiles will serve as the basis for providing personalised and contextualised filtering of seed and enrichment content on the LinkedTV platform, as well as of concepts recognized in the content annotation and/or within the LinkedTV concept space.

This deliverable is oriented towards presenting the aspect of building a user profile (implicit or explicit) and adapting it the user context, whereas the sequent deliverable D4.5's deal with how we present personalized and contextualized concepts content to the user based on these profiles.

The concept and current state of implementation for the user profiling and contextualisation of two complementary approaches is presented in this document: (a) implicit profiling, the extraction of lightweight and dense implicit knowledge about user preferences and (b) explicit profiling, the explicit expression of interest by the users. The former process includes the extraction of information that stem from the user's interaction with the content, with the implication of unobtrusive transaction capturing, semantic interpretation and homogenization of preferences, preference learning over time and consumption and mining association rules among preferences. The latter process involves a dedicated user interface that allows the manual creation and maintenance of a user profile. This information evolves to a semantic user model that is made available for predictive inference about relevant concepts and content.

Currently both approaches are in a separate research state, which will be merged in the next stages of the workflow implementation within year 3 of LinkedTV. The collaborating workflow of the user profile capturing, updating, learning and construction is illustrated in Figure 1.

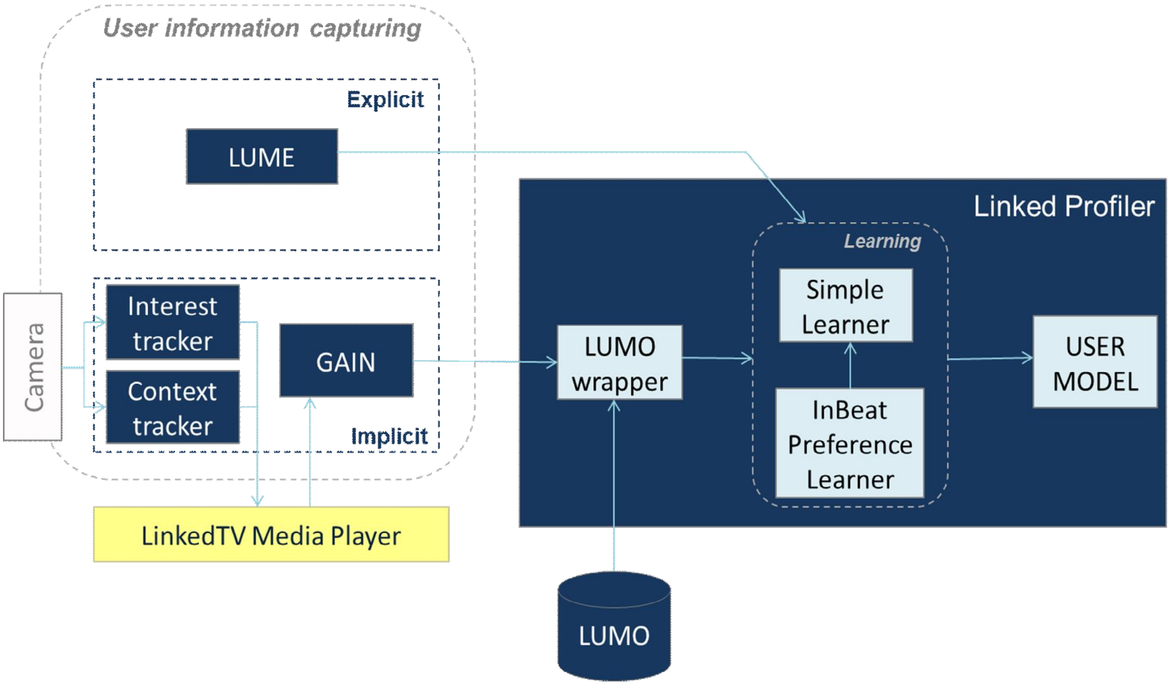


Figure 1: The user modelling tools workflow

In the following sections, Chapter 2 deals with the reference knowledge base upon which the creation of semantic user profiles will be based, Chapter 3 provides an overview of how implicit user behaviour is captured and user preferences are derived from it, Chapter 4 describes the profile learning and construction services implemented, while handling the communication of the tools within the personalisation workflow, Chapter 5 outlines the methodology that will enable contextual adaptation of user profiles and Chapter 6 illustrates the technology available for creating and managing explicit user interests. Finally, Chapter 7 conducts an overview of this document and presents the following steps of the personalisation and contextualisation task.

1.1 History of the document

Table 1: History of the document

Date	Version	Name	Comment
2013/07/29	V0.1	Dorothea Tsatsou	Empty document with ToC for contribution filler
2013/08/04	V0.2	Dorothea Tsatsou	Incorporation of first UEP, UMONS and CERTH input
2013/08/15	V0.3	Dorothea Tsatsou	Incorporation of the first FhG input, minor modifications (figure 1)
2013/09/06	V0.4	Matei Mancas Tomas Kliegr	Revised contextualization sections and related diagrams

Date	Version	Name	Comment
2013/10/02	V0.5	Dorothea Tsatsou	QA-ready version
2013/10/04	V0.6	Jan Thomsen	QA performed
2013/10/10	V0.7	Dorothea Tsatsou	Post-QA revision Incorporated UEP, UMONS, FhG post-QA input
2013/10/11	V1.0	Dorothea Tsatsou	Final D4.4 document

1.2 List of related deliverables

D4.4 has tight connections to several LinkedTV deliverables, most prominently with:

- D4.2, where the approach and methodologies of the techniques implemented and described in this deliverable was defined.
- D4.5, a sequent of this deliverable, where it is reported how content and concept filtering is implemented based on the contextualised user models described in this deliverable.
- D2.4, where the connection between the reference knowledge base employed by tools in this deliverable (and in D4.5) and the content annotation is described.

Complementary, since the personalisation and contextualisation task is the last end (in terms of research tools) of the LinkedTV pipeline, there are connections with several other deliverables:

- D6.1, where the scenario descriptions, based on which use case profiles are generated, are portrayed.
- D3.6, where the LinkedTV media player, from which implicit user input is received, is described.
- D5.4, where the integration of the personalization and contextualization tools with the LinkedTV platform is described.
- D7.4, where demonstrators and APIs of the services described in this document (and in D4.5) are reported.

2 Reference Knowledge

Within LinkedTV, multimedia content is segmented into spatio-temporal fragments, accompanied by metadata stemming from several analysis services within the LinkedTV pipeline. Among this metadata, semantic annotation of media fragments, describing 'what this fragment is about' may be retrieved. The content analysis tools provide this description into a plurality of Linked Open Data (LOD) vocabularies, in the interest of efficient, complete and standardized disambiguation and linking of media fragments.

However, in the interest of more efficient personalisation services, the heterogeneity of content annotation imposed by this variety of descriptive vocabularies, the lack of user-pertinent semantics in vocabularies of a more general purpose, the overflow of information unrelated to a user-centric perspective and the imperfection in the structure and semantics of often user-contributed vocabularies have led to the need of expressing user preferences in a homogeneous, user-centric reference knowledge base.

This knowledge base, namely the LinkedTV User Model Ontology (LUMO), aims to cover all information pertinent to a user in the networked media domain in a uniform, lightweight and formal ontological vocabulary.

While implementation is at the state of exploring the convergence between implicit user feedback and its semantic interpretation and on the other hand explicit preferences expressed manually by users, it was deemed necessary to address the reference knowledge base in two parallel yet complimentary instances that serve each task's requirements. Therefore LUMO v1 was developed in order to address the most prominent requirements of implicit profiling: coverage of the domain concept space and its correspondence to the concept space used in the annotations of the content and coping with the absence of user-pertinent semantics in content annotation (such as topic detection). In addition, LUMO-rel was developed in order to cover in depth more expressive information that the user can offer explicitly, such as preferences where preferred concepts can be restricted in specific non-taxonomic relations via a plurality of object properties.

LUMO v1 and LUMO-rel are tightly related and the optimal merge of the two knowledge bases (and in extent personalisation approaches) will be a major subject of research for year 3. In this chapter, the implementation of LUMO v1 is presented as the basis of the reference knowledge of the personalisation and contextualisation task, and although LUMO-rel comprises an integral part of the reference knowledge, it was deemed more suitable, for reference purposes, to be coupled with the description of the approach that is based on it, namely the explicit user profiling task, and is thus presented in Chapter 6 (Section 6.1).

2.1 LUMO v1

A first version of LUMO ontology has been developed based on the design principles set within LinkedTV and reported previously in deliverable D4.2, to semantically represent user-pertinent information, in order to enable semantic personalization and contextualization (i.e.

filtering) of concepts and content for the networked media domain. In a nutshell, its design principles consists of having a comprehensive coverage of the domain in a uniform, finite and expressive vocabulary that is oriented towards modelling the user's perspective of the domain, while at the same time it remains as lightweight as possible so that it can enable the semantic expression and storage of user preferences and also the filtering process performable on a limited resources client. This choice aims to alleviate both scalability issues but, most prominently, user privacy safeguarding by limiting the need to communicate sensitive user information outside the user client.

LUMO is designed as an OWL[2] ontology, but its expressivity is limited to the DLP fragment [GRO03] (which is backwards compatible to the most recent OWL 2 RL[3] fragment), in order to maintain the minimal complexity as per the LinkedTV requirements and address scalability and user privacy issues. LUMO is currently accompanied by a separate mappings ontology, modelling mappings of LUMO to several existing ontologies and vocabularies.

LUMO v1 is published under the http://data.linkedtv.eu/ontologies/lumo namespace. LUMO's accompanying mappings ontology (cf. D2.4, Chapter 3) serves as both the means to interpret content annotation unto the LUMO concept space and as the means to render LUMO shareable and re-usable by the Semantic Web.

LUMO addresses four major user-pertinent aspects: Context Dimensions, Domain-dependent dimensions, Agents and Spatio-Temporal concepts. The two latter aspects may regard both contextual and domain-dependent aspects of the user and as so are modelled independently on the top of the hierarchy.

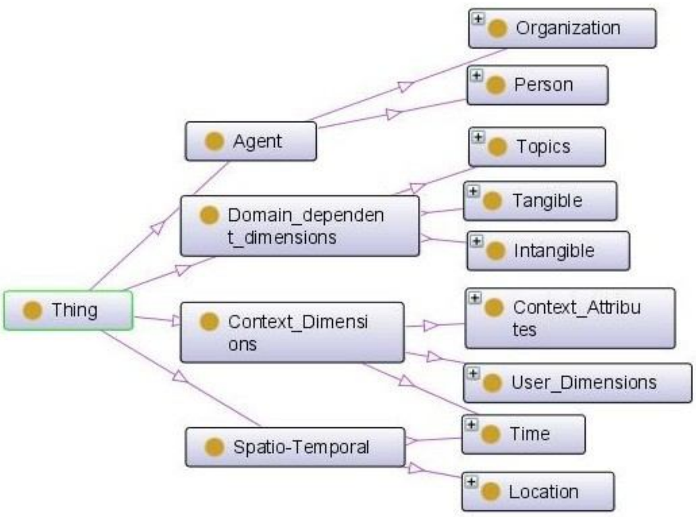


Figure 2: Graphical representation of the top of the LUMO v1 hierarchy

[2] http://www.w3.org/TR/owl-guide/

[3] http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

2.1.1 Connection to existing vocabularies

LUMO engineering aims to model the most relevant entities and semantics from open vocabularies and adapt them to the networked media domain and the needs of the LinkedTV users. This includes adopting, adding new and appropriately discarding domain-extraneous information from relevant established vocabularies, as well as redefining semantics, with respect to leveraging LOD inconsistencies and enhancing coherency and completeness of modeled semantics.

Therefore, the current version of LUMO is accompanied by a separate mappings ontology that maps LUMO to existing vocabularies. Mappings were generated automatically via the LogMap [JIM11] tool and evaluated and revised manually.

Mappings are detached from the ontology in order to enable user modeling and inferencing (recommendation) to take place in the minimum representative concept space, with regard to scalability and user privacy safeguarding issues. Mappings thus serve a) for interpretation of content annotation and b) as the means to facilitate re-use of the ontology by the Semantic Web.

The major existing vocabularies that have inspired and guided the engineering of LUMO are:

- **GUMO**[4] has guided the design of the top level context and domain-dependent dimensions, while several GUMO contextual dimensions where adopted and adapted to LinkedTV's platform requirements (including camera-based sensor related concepts). Similarly, some LUMO "Topics" have been modeled to a great extent according to the corresponding GUMO "Topics".
- The **IPTC**[5] news codes were the main influence towards modeling the LUMO "Topics" hierarchy. Most upper topic categories were adopted per se and subcategories and concepts related to them where revised and adapted to the topics hierarchy. For instance, several subcategories were created (w.r.t. to also **Wikipedia categories**), while concepts in IPTC that were semantically direct descendants (with the "is-a" relationship) of the "Tangible" and "Intangible" LUMO categories were moved to that subhierarchy and related to "Topics" with the "hasTopic" and "hasSubtopic" object properties.
- **Schema.org**[6] influenced the modeling of the "Agent", "Location", "Intangible" and "Tangible" categories.
- These categories where populated also in correspondence to the **DBPedia schema**[7] and the **NERD**[8] ontology; in a smaller extent they were influenced by several other open vocabularies.

---

[4] http://www.ubisworld.org/ubisworld/documents/gumo/2.0/gumo.owl

[5] http://webtlab.it.uc3m.es/results/NEWS/subjectcodes.owl

[6] http://schema.org/docs/schemaorg.owl

Modelling these aspects heavily relied on representing information stemming from content annotation provided by the TV Metadata and TV Enrichment services of WP2 (cf. D2.4 and D2.5).

2.1.2 Non-taxonomical axioms: topic detection

Apart from a mere taxonomy, LUMO v1 incorporates a number of universal quantification axioms that aim to retrieve additional information about the user preferences based on given non-taxonomical relations between the concepts. In the current version of LUMO, such relations (object properties) and in extent axioms that produce non-taxonomic connections between concepts via these properties have been modeled so as to enable automatic detection of topics in both the content's annotation and in the user profile.

Two such relations, namely "hasTopic" (parent relation) and "hasSubtopic" (child relation) are currently used to connect entities such as objects, events, agents, locations etc which are detected by LinkedTV's content analysis and annotation tools (in essence what is modeled under the "Tangible" and "Intangible" categories) to their topics (i.e. concepts under the "Topics" category).

Through these relations and their corresponding axioms the profiling components are enabled to detect topics of interest/disinterest for the user based on his transaction with the content and the topics pertaining to a given content item which consists the recommendation candidate for the filtering process (cf. also D2.4).

Topics are deemed important to better understand and formulate user preferences in the networked media domain while automatic topic detection via analogously modeled knowledge was deemed mandatory in the case of LinkedTV, where topics can be derived from the entities retrieved in the content analysis process.

The connecting axioms are of the form *entity ⊑ ∀has(Sub)Topic.topic*, where *entity* is subsumed by the Agent, Tangible and Intangible concepts/categories of LUMO and *topic* is subsumed by the Topics concept/category of LUMO. An illustrative example of such an axiom in the LUMO ontology can be seen in Figure 3, where "Political_Agent" and all its subclasses (by inheritance) are linked to the topic "Politics" through the universal quantification axiom "Political_Agent ⊑ ∀hasTopic.Politics".

[7] http://wiki.dbpedia.org/Ontology

[8] http://nerd.eurecom.fr/ontology/

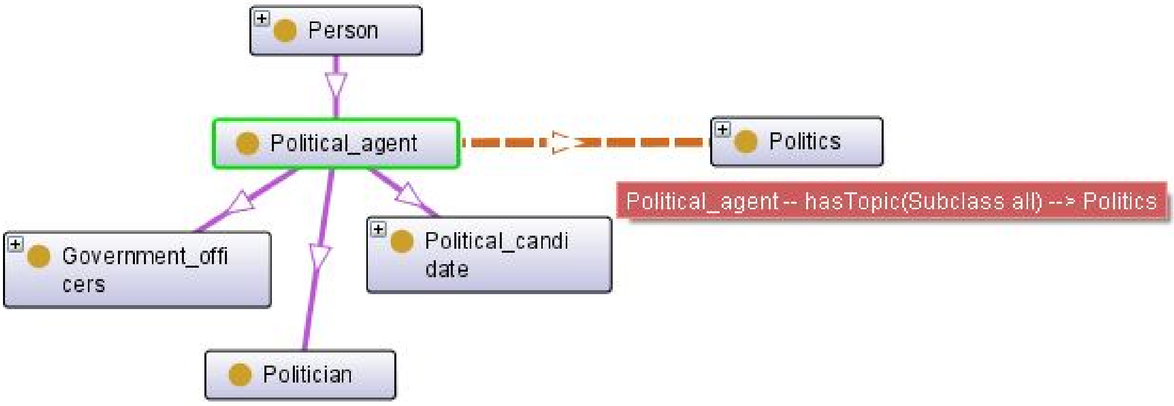


Figure 3: An illustration of an axiom pertaining a universal quantification of an entity subsumed by „Political_Agent" by the „hasTopic" property.

An example showcasing the use of these axioms would be the detection of the Topic "Politics" for the existence of entity "Barack_Obama", recognized as type:"Politician" in a given content item.

We have from the annotation of content item that:

$\langle Barack\_Obama{:}Politician \rangle \geq 0.75$

Given a relation between the instance "Barack_Obama" and the default individual "content" (cf. also D2.4) such as:

$\langle Barack\_Obama, content{:}hasTopic \rangle$

And given the modeled LUMO knowledge:

$Politician \sqsubseteq Political\_Agent$

$Political\_Agent \sqsubseteq \forall hasTopic.Politics$

The reasoning services can deduce that the media item's *content* at hand involves the topic *Politics* by 0.75.

3 User input

This chapter involves the process of capturing implicit user information via the user's transactions with the player. This information, combined with the explicit user input provided by the LUME interface (cf. Chapter 6.2) is the input for the profile learning tools.

Subsection 3.1 describes the implementation of the GAIN service which receives and manages information derived from user transactions within the LinkedTV player, while Subsection 3.2 describes the reception and management of user behaviour while transacting the LinkedTV platform via camera-based attention tracking.

3.1 GAIN

GAIN (General ANalytics INterceptor) is a web-service framework for capturing and processing user interactions originating in web-connected applications.

The input of GAIN is a variable number of *interest clues* and a *semantic description* of content. Interest clues are user interactions (clicking a button) or come from observation of user behaviour (attention level). Additionally, GAIN can work with contextual interest clues (ref. to Subsection 5.1.2). Context is handled as a special type of interaction, which spans over a period of time. An example type of context is the number of children in the room.

Content is described in terms of shots, which are a subtype of media fragments and the smallest amount of content to which the interest clue can be related to. Shot can contain multiple entities linked to domain ontology by Linked Open Data (LOD) identifiers.

GAIN outputs a number of aggregations of the received data. A prominent type of GAIN output is a data table export, which can be consumed by generic machine-learning algorithms.

This table contains instances. One instance corresponds to one shot. The instance feature vector composes of the scalar value of interest and two subvectors: the context subvector (ongoing work) and the description of content of the shot, which is a fixed-length vector of weighted classes from the domain ontology, created by an aggregation of entities detected in the shot, The scalar value of *interest* is computed from all interest clues raised during the shot.

To foster integration with other components of the personalisation and contextualisation process, GAIN is encapsulated into a convenient REST web service interface. In the remainder of this section, the internals of GAIN are described.

Interaction Tracking & Application Modules capture interactions from the user interface (LinkedTV player). The application module embeds the GAIN core logic. GAIN is implemented on top of Node.js (http://nodejs.org/), a lightweight scalable platform with event-driven, non-blocking I/O model. After processing, the data are passed to the storage module.

Storage module is implemented as a NoSQL document-oriented database MongoDB (http://www.mongodb.org/). The key advantage of this solution is a schema-less design, which fosters the attachment of semantic description of shots.

Aggregation module performs final aggregations for GAIN clients. This module also performs the machine learning instance generation.

Content Platform Interface module obtains content description from the platform through the designated query points (SPARQL queries or REST service).

Profile Learning Connector REST API provides aggregated output to the profile learning components.

Figure 4 illustrates in detail the workflow and services of GAIN in conjunction with the intra-LinkedTV workflow, from the input derived from the LOU[9] media player and the LinkedTV platform to the output of GAIN to the Linked Profiler[10].

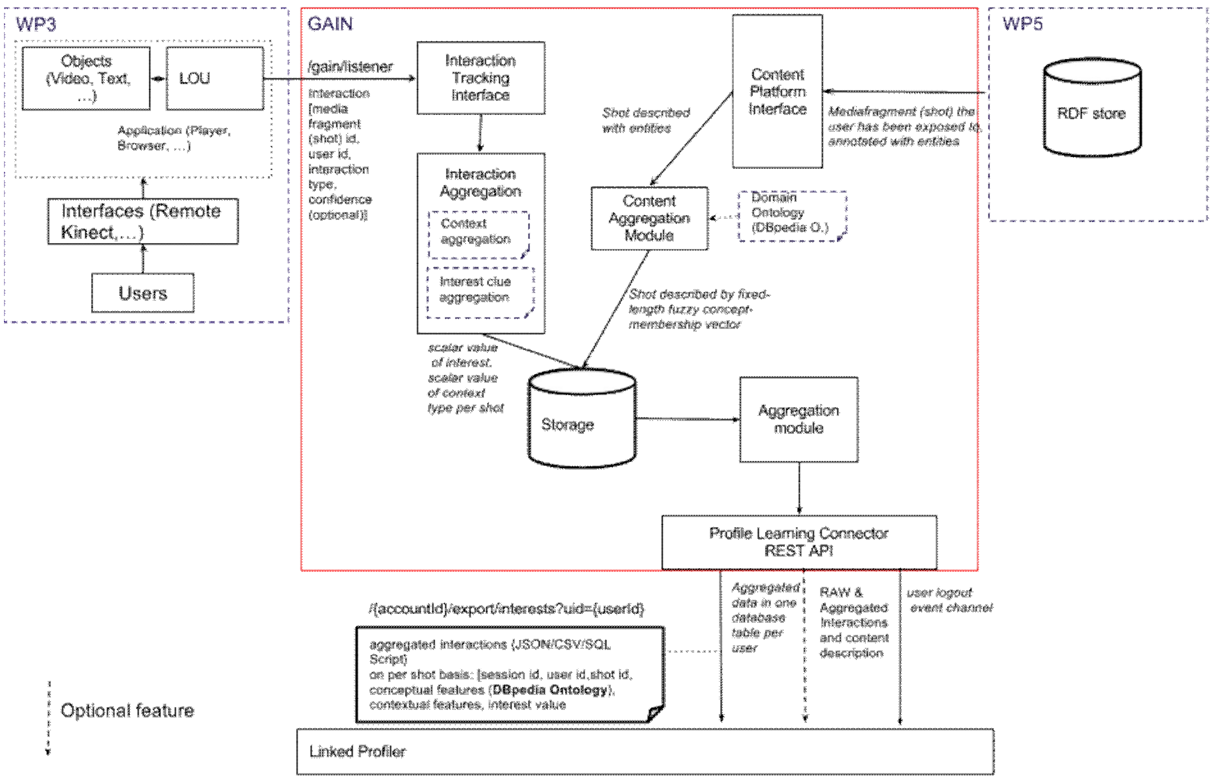


Figure 4: GAIN internal architecture, interfaces with other LinkedTV components.

3.1.1 GAIN Output

GAIN outputs a data table in a format that complies with the fuzzy version output option described in D4.2 Section 4.1.2.2. The events recognized and passed to the learning tools comprise the consumption/selection of concepts from the platform's seed media and the

[9] Cf. D3.6 for a description of LOU and its functionalities.

[10] Cf. Chapter 4.

concepts from browsing of enrichment content. The output data generated by GAIN looks as follows:

Table 2: GAIN output example

User_id	d_r_ North_Korea	…	d_o_SoccerPlayer	…	c_userlooking	…	interest
1	0		0.9		1		0.3
2	1		1		0		1

In Table 2, the prefix d_r_ indicates that the feature corresponds to a DBpedia resource from the English DBpedia. For DBpedia in other languages, the prefix is d_r_lang_.

For example, for d_r_North_Korea, we obtain http://dbpedia.org/resource/North_Korea.

The prefix c_ indicates that the given feature is the name of the context. For example, c_userlooking corresponds to the user is looking context.[11]

Features with prefix d_o_ correspond to classes in the DBpedia Ontology. For example, the d_o_Soccer_Player feature corresponds to http://dbpedia.org/ontology/SoccerPlayer class.

DBPedia concepts are retrieved and expanded by the content annotation and consist of the entities recognized by the content analysis tools to be describing the content. For a more thorough description of how DBpedia concepts are handled in GAIN and how their interest weights are derived, cf. D4.2 (Chapter 4.1.1, 4.1.2, 4.2, 4.3).

As seen in Figure 4, the output of GAIN serves as input to the Linked Profiler, which uses it to learn the user profile. Linked Profiler is in detail described in Chapter 4.

3.2 Attention tracker

The attention tracker is a module dedicated to analyse the head behaviour of a user in front of a screen. This module detects if the viewer is looking at the screen or not and it can provide a rough idea about the TV screen areas which are gazed by the viewer. This module uses a RGBD sensor which provides both RGB video images and a depth map (D) of the same scene. This sensor (like the Microsoft Kinect [KIN10] for example), provides X, Y, Z coordinates for all the pixels (voxels) of the scene.

The goal is to achieve head tracking in real time and estimate the six degrees of freedom (6DOF) of the detected head (spatial coordinates, pitch, yaw and roll). The advantage of a 3D system is that it uses only geometric information on the point cloud and is independent of the illumination issues which can dramatically change in front of a device like a TV. The proposed system can even operate in the dark or in rapidly varying light conditions, which is not

[11] Planned feature

possible with face tracking systems working on RGB images. In addition, the use of 3D data provide more stable results than 2D data which can be misled by projections of the 3D world on 2D images.

Figure 5 shows the global pipeline of the head pose estimation sub-system. First, the 3D point cloud is extracted from a RGBD sensor using the PCL library [PCL11].

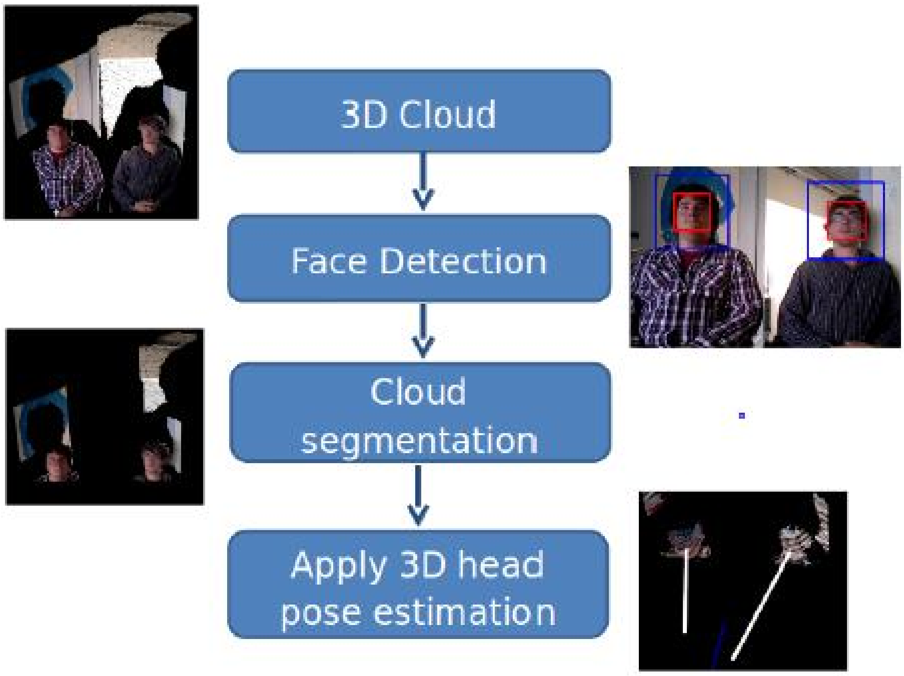


Figure 5: Algorithm pipeline: 3D cloud extraction from the RGBD sensor, face localization and detection, 3D cloud segmentation and pose estimation.

In a second step faces of people are detected and localized (the blue larger boxes in Figure 5). Those boxes are computed from the head of the skeleton extracted from the depth maps by using the OpenNI library [ONI10]. The skeleton head provides the 3D coordinates of the area where a face might be located. The smaller red boxes are 2D face detection and recognition. The face recognition implementation fuses the results of 3 complementary state-of-the-art face recognition algorithms (Local Binary Patterns Histograms – LBPH [APH04], FisherFace [BEL97] and EigenFace [TUR91]) by using a majority voting approach. The system is able to recognize people faces and assign them a known ID (the same as the one of their LinkedTV profile). **If the viewer's face is not recognized, than his information is either not sent to the user profile adaptation because the user is not registered in the system, or it is sent as anonymous user,** thought the automatic generation of a unique anonymous ID by the player**.** This choice will be made during year 3 depending on the way that users will be managed in the final approach. If the person is recognized, or an anonymous user ID has been generated, than the ID of his profile is sent along with the message containing information about his attention changes.

From an ethical point of view, the face recognition system is equivalent with a classical registration form where the user has to log on. The difference is that faces snapshots are recorded locally on the client. Those images are not sent to any server and are only used locally to recognize the user. Only the user ID is sent to the server along with the behaviour message.

Once the 3D position of the head is extracted, the 3D cloud is segmented to optimize the last 3D head pose estimation step. The segmentation eliminates a lot of the points of the 3D clouds where the chances to find a face are very low and therefore boosts the computational efficiency of the method. An optimization between the number of points eliminated is to be done as less points will boost the computation time while not enough points can provide errors in the 3D head pose estimation.

The 3D head pose estimation used here follows the development in [LER13] which is improved by 4 in terms of computation time due to the 3D point cloud segmentation. The 3D pose estimation algorithm is based on the approach in [FAN11][FAN13] and implemented in the PCL library [PCL11]. This solution relies on the use of a random forest [BRE01] extended by a regression step. This allows us to detect faces and their orientations on the depth map. The method consists of a training stage during which we build the random forest and an on-line detection stage where the patches extracted from the current frame are classified using the trained forest.

The training process is done only once and it is not user-dependent. One initial training is enough to handle multiple users without any additional configuration or re-training. This is convenient in a setup where a wide variety of people can watch TV. The training stage is based on the BIWI dataset [FAN11] containing over 15000 images of 20 people (6 females and 14 males). This dataset covers a large set of head poses (+/-75 degrees yaw and +/-60 degrees pitch) and generalizes the detection step. During the test step, a leaf of the trees composing the forest stores the ratio of face patches that arrived to it during training as well as two multi-variate Gaussian distributions voting for the location and orientation of the head. This step of the algorithm provides the head position and a rough head orientation on any new individual without the need of re-training. We then apply a final processing step which consists in registering a generic face cloud over the region corresponding to the estimated position of the head. This last step greatly stabilizes the final head position result.

The attention tracker was integrated in the main workflow as shown in Figure 6 From the 3D sensor two paths using the 3D point cloud on one side and the RGB image on the other side are used to extract the 3D gaze approximation and to recognize the face of the viewer. The modules called “User Profile” and “TV Behaviour Analysis” fuse the information of interest (only the changes in interest are selected) and the information from the face recognition algorithm by adding the corresponding ID to the 3D head pose change. The “Network Manager” module adds additional information coming from the context tracker (which provides the number of people) and from the gesture module (which provides play/pause/jump information from the recognized viewer gestures). All those messages are packed and sent through the WebSocket protocol [WEB10] using a C++ websocket library to a server which is handled by the GAIN interface. This server receives the information and fattens it with the video ID and time which is playing on a test player and forwarded to the GAIN module. For the moment, the communication is done with a test server handled by GAIN, but in the final version, the communication will be done with the LOU[12] media player server which will fatten the messages with the video ID and time and forward them to GAIN.

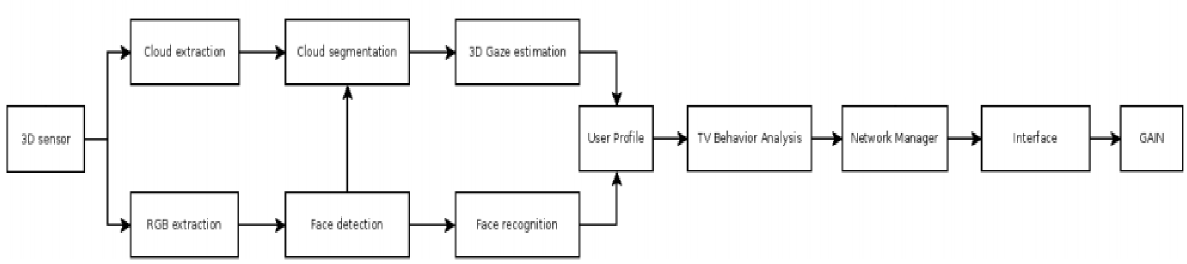


Figure 6: Attention tracking workflow

The attention tracking workflow is illustrated in Figure 6: 3D head pose and face recognition result go to the user profile and TV behaviour analysis which proceed to information low-level processing and fusion and forward it to the network manger module. The network manager takes all the messages (from the interest module, context tracking module and gesture module) and sends them to the player using the websockets protocol. The player enriches the messages with the video ID and time and forwards to the GAIN module.

3.2.1 Experiment

To validate our approach of 3D head pose estimation and the first integration of the processing workflow, we realized a preliminary experiment on 10 users. The objective was to see our system can provide in a realistic way the viewer interest in part of a media he watches and if different behaviors can be observed.

To constrain and simulate the interest of a user, s/he was asked to answer to a set of questions about a subject related to part of the videos which are displayed to him. In the same time, he was asked to play a simple game on a tablet: the purpose is to correctly answer to the questions AND to have a good score on the tablet-based game while bot actions needed attention. The game was used to simulate the interactions on a second screen and to provoke attention switches between the main screen and the second screen. Figure 7 shows the physical setup used for the tests. A main screen (laptop) where the videos are displayed (media test player), a second screen with a game (tablet) and a third (tablet) for answering the questionnaire. The RGBD sensor which computes the 3D head pose estimation is located on top of the main screen.

[12] Cf. D3.6 for a description of LOU and its functionalities.

Figure 7: Setup of the experiment

The main screen is the one of a laptop, while the second screen and the questionnaire are on two tablets. Figure 8 shows the main modules working: on top-left the 3D head pose estimation from the 3D point cloud, on bottom-left, the face recognition module, on the right the UEP test player displaying the content which is here a set of three videos. The second video is related to the questions of the questionnaire (about sports). The two others (politics) were not related to the questionnaire topic.

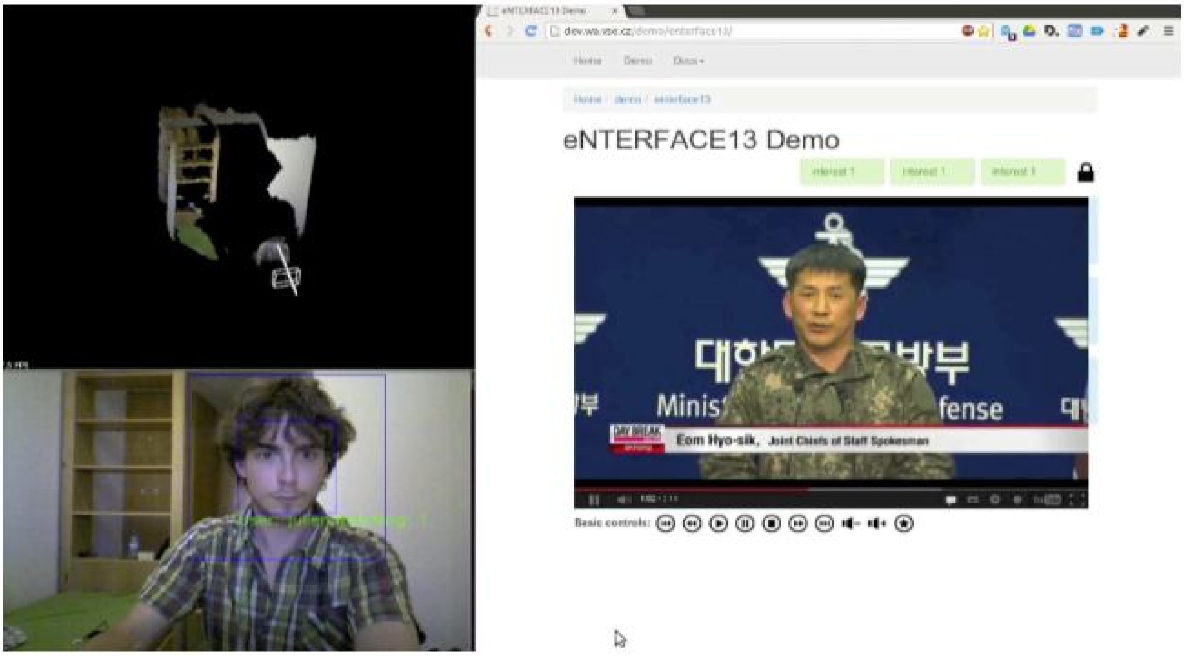


Figure 8: Top-left: head direction computation, Bottom-left: head detection and face recognition, Right: media test player displaying content. The left attention tracking modules are linked to the player by using the websocket technology.

The test scenario was the following: three videos were displayed to the viewer with the second video which is related to the questionnaire. The expected behavior was the following:

- While the videos not related to the questionnaire play (1st and 3rd), the user will mainly focus on the game on the second screen (and therefore will not look to the main screen)
- While the video related to the questionnaire plays, the viewer looks to the main screen and therefore stops playing the game on the second screen. He can also stop or jump in this video to go back to a topic related to the questions of their questionnaire

The results of the main behavior are presented in Figure 9 also call the “interest beat”. The timeline comes from the data received by the GAIN module. It represents the synchronized viewer behaviour over the displayed content (first video from 00:00:00 to ~00:02:30, second video until ~00:05:20 and the third video until the end ~00:07:00). On the Y axis, from top to down we can see: clicks on links, pause, play, interest-0 (not looking to the main screen), interest-1 (looking to the main screen) and interest-no_user (user not found anymore).

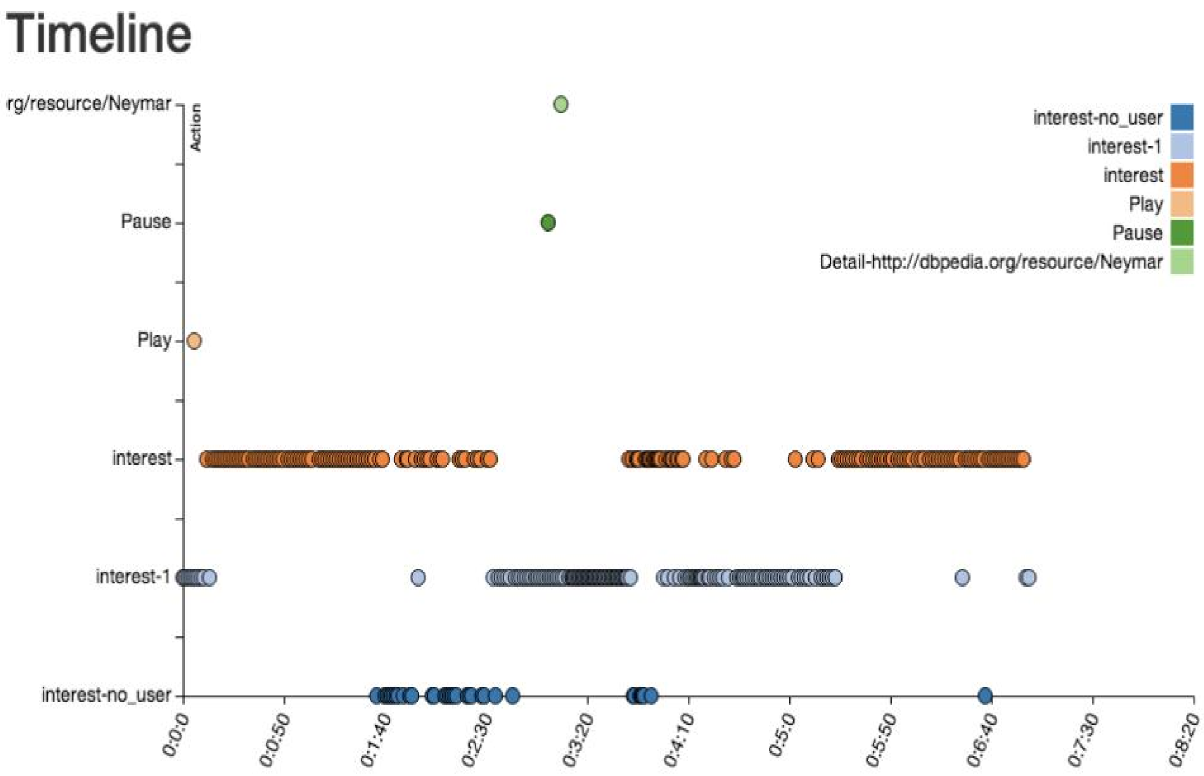


Figure 9: Interest beat illustration

The interest beat is a graph presenting the input of the GAIN module and consists in synchronized events happening all over the content for the main (recognized) viewer. From top to down we can see the different events (click on links, pause, play, interest off main screen, interest on main screen, interest in a specific concept and no information about viewer interest). The latest category means that the face of the user is not visible anymore and the head pose is not available anymore (out of the range of +/-75 degrees yaw and +/-60 degrees pitch.

The example of a viewer in Figure 9 summarized the typical viewer behaviour:

- First he clicks on play ...
- In the same time he watches the content during some seconds (Interest-1)
- When he realizes that the first video is not related to his questionnaire he does not look anymore to the main screen and begins to play the game on the second screen (tablet)
- He is only attracted to the main screen once (probably it is only a noise as it just occurs once). The system loses his head (he is really focused on the tablet game)
- When the second video (which contains the answers to his questionnaire) begins, he focuses on the main screen
- At some point he uses pause and play again to have time to answer the questions of his questionnaire. He also clicks on a link provided by the viewer as one of the questions requires the user to get extra information using this link.
- Once the second video is finished, he mainly focuses again on his tablet game and not on the main screen.

The ten users provided results which sometimes differ in terms of behaviour: sometimes the viewer forgot to answer the questions, so he had to go back in the video to answer the questions at the end. But for the 10 viewers the people 3D head pose was correctly detected mainly on the main screen during the second video display, while the head direction was out of the main screen or impossible to determine during the two other videos where the viewer was playing the game on his tablet.

4 Linked Profiler

Linked Profiler is responsible for receiving input by captured or explicitly declared user interests, for incorporating and handling cooperation among all preference capturing and learning tools within the LinkedTV personalisation and contextualisation task and that ultimately produces a fuzzy semantic user profile that can be received as input by the LinkedTV filtering tools.

Three subcomponents are intertwined within the Linked Profiler and cooperate to produce the final semantic user profile:

1. LUMO wrapper, which interprets the semantic description of captured user preferences under the LUMO vocabulary.
2. Simple Learner, which is responsible for learning a user profile over time and content consumption.
3. The association rules produced via the InBeat preference learner engine.

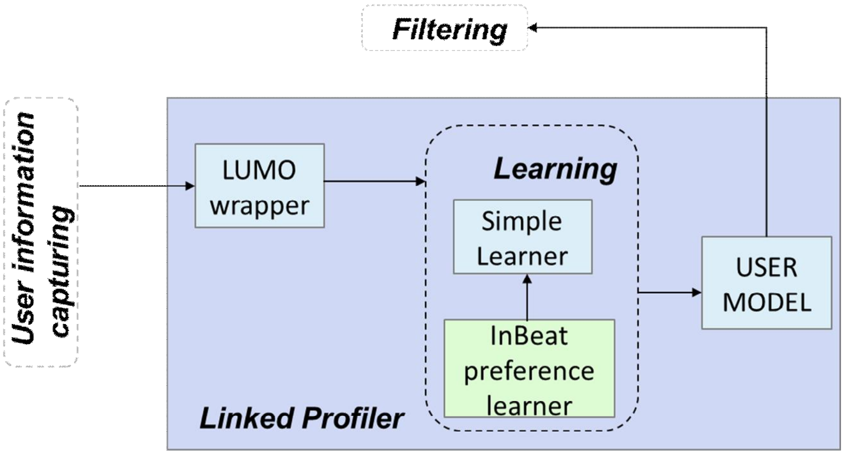


Figure 10: Graphical representation of the Linked Profiler sub-workflow

The Linked Profiler is in particular responsible for:

- the communication among and unification of all preference reception, transformation and learning components,
- the production of the final semantic user profile.

In Figure 10 the internal Linked Profiler workflow is illustrated.

4.1 LUMO wrapper for preference capturing

The LUMO wrapper is used in the context of WP4 to transform input received from WP2 via GAIN from the WP2-used vocabularies to LUMO v1 and thus to maintain the principles of minimizing and unifying the problem space under a single, common vocabulary for all tools involved in the profile building process. Through this transformation, uniformly expressed preferences are passed onto the profile learning modules, thus also minimizing the concept space and maintaining the profiles more cohesive and lightweight.

4.2 Preference learning

4.2.1 Simple learner

Simple learner is a baseline profile learning component that adds new preferences to the user profile, weighted based on the interest shown by the user in a given content consumption session, based on the input received by the GAIN component. Given this interest, it is able to manage and communicate both captured interests and disinterests. It also updates existing preferences based on their frequency and age of the preference in the user's transaction history. It is integrated directly into the Linked Profiler.

The learning stage consists of updating the interest weights of existing preferences according to their frequency of appearance in the user's transaction history and the age of the preference based on a time decay factor (cf. D4.2).

It is also responsible for incorporating association rules learned by InBeat’s EasyMiner subcomponent. The latter is achieved by introducing (apart from primitive concepts-preferences) synthetic pseudo-concepts which serve as pointers to the existing rules learned by EasyMiner.

In an intermediate step, the simple learner incorporates the transaction history of the user, updated upon each transaction of the user with content, along with information about the time when the last interaction that the user had with a certain preference took place and the frequency of interactions that s/he had with that preference. As a result it exports an intermediate output which includes entries per preference of the form:

```
"timestamp":<last timestamp that the concept was accessed>,
"concept":<preference concept>,
"aggregated weight":<sum of interest across all user sessions for this con-
cept>,
"positive":<TRUE if interest/FALSE if disinterest>,
"frequency":<times that concept was accessed>
```

Note that "*preference concept*" here refers to either a primitive preference (one entity, concept or instance, from LUMO v1), or complex preferences (a rule between LUMO concepts, the contecendent of EasyMiner output, see next chapter). In the case of rules which provide the weight by which the rule holds, the entry *“aggregated weight”* includes only the confidence degree by which the rule holds and the *“frequency”* entry is always equal to 1. Only the timestamp of the last time this rule was updated is used to apply a time decay factor to the extracted rules.

4.2.2 EasyMiner

EasyMiner (easyminer.eu) (originally named I:ZI Miner) is a web-based association rule mining software based on the LISp-Miner system (lispminer.vse.cz). EasyMiner is both an interactive web application, which allows interactive pattern mining, and a web-service stack, which provides access to LISp-Miner via its LM-Connect component [SKRA12].

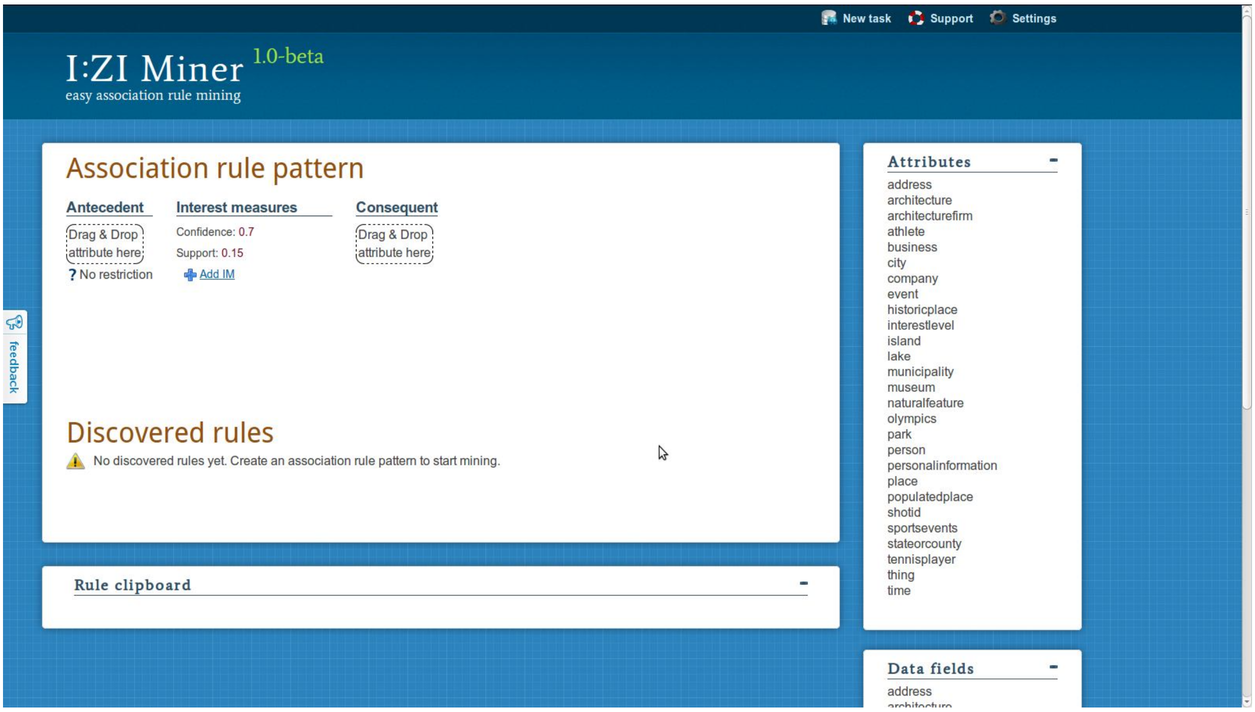


Figure 11: Screenshot of EasyMiner Web interface

Within LinkedTV, EasyMiner/LISp-Miner are used to learn preferences, with the input data being provided by GAIN. The learning workflow is handled by the *InBeat Preference Learner*.

4.2.2.1 InBeat Preference Learner

The InBeat Preference Learner is a standalone component which wraps EasyMiner and LISp-Miner, as the underlying rule learning stack, and provides services specific for preference rule learning in the LinkedTV context. These services include:

- Mining task generation, as described in deliverable D4.2.
- External connection to the GAIN interface. For a given user, GAIN outputs one table, where rows correspond to shots recorded for the user. The Table has $n+m+c+1$ columns, where n is the number of classes in the domain ontology, m number of unique objects (entities), c the number of contextual features, and the last column is the estimated value of interest. Within the context of LinkedTV however this functionality is wrapped under the Linked Profiler API and runs GAIN input through the transformation of LUMO wrapper.
- LinkedTV interface and Rule Storage.

The *InBeat Preference Learner* provides input for Simple Learner. The Rule Store is accessed by the *InBeat Recommender*, which is described in D4.5.

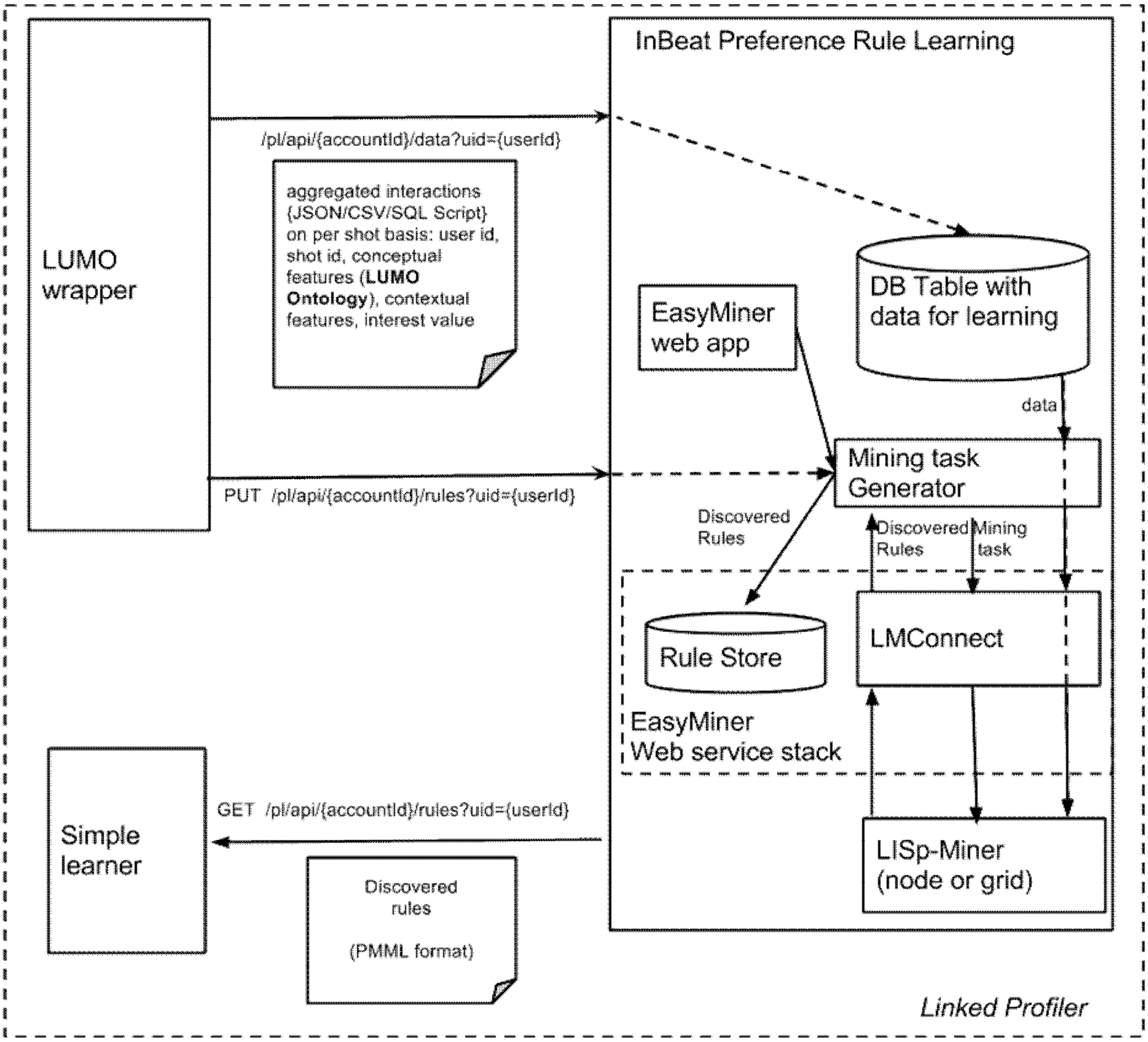


Figure 12: Communication between EasyMiner, GAIN and other LinkedTV components

4.2.2.2 EasyMiner output

The default rules output by the EasyMiner Web service stack follow the following pattern

```
[concept({0,1})]* AND [resource({0,1})]* AND [context({0,1})]* => interest(value)
```

If context is handled in the condition part of the rule, rather than in the rule antecedent, the output rules follow this pattern:

```
[concept({0,1})]* AND [resource({0,1})]* => interest(value) / [context({0,1})]*
```

In both cases, the value interest is discretized. Again, to decrease the size of the learning space, the number of categories was decreased as compared to D4.2 (Table 3).

Table 3: Interest value preprocessing in EasyMiner

Interest value	Output value
[-1;0)	Negative
0	Neutral
[0;1)	Positive

What concerns the interest measures used to filter the hypothesis, confidence and support are used (for details refer e.g. to D4.2).

An example output rule is:

```
band(yes) => interest(Positive)
```

The discovered rules are presented in an extension of the PMML format (Predictive Markup Modeling Language),[13], which contains values of the interest measure used as constraints, and additionally values of supplemental measures.

The output rules are passed by the InBeat Preference Learner to the Simple Learner.

4.3 User profiles generation

The results of the aforementioned learning process are fed to the profile generator which takes into account the primitive preference information from the Simple Learner and enriches it with more expressive and complex information by incorporating the association rules formed between user interests via EasyMiner to produce an axiomatic, ontological user profile within the LUMO concept space.

The semantic LUMO-based content annotation profile would therefore be of the form:

$\exists hasInterest.(w_A \cdot A \sqcup w_B \cdot B \sqcup w_C \cdot C \sqcup w_D \cdot \langle a{:}D \rangle \sqcup \ldots) \sqsubseteq UIC$

where A,B,C,D are concepts that denote user interests, w_A, w_B, w_C, w_D correspond to their preference weight for the specific user, $\langle a{:}D \rangle$ denotes a specific preference for an instance *a* that instantiates a LUMO concept D , *hasInterest* is a default LUMO object property connecting the prefences with the *UIC* concept and *UIC* is a manufactured concept which corresponds to the user ID. These concepts can either be LUMO classes, individuals instantiating a LUMO class, or complex concepts, i.e. logical expressions, for example:

$\langle b{:}E \rangle \sqcap F \sqsubseteq B$

$\exists R.E \sqsubseteq C$

$\exists R.(E \sqcap F) \sqsubseteq D$

---

[13] dmg.org

where *R* is a LUMO property, *E,F* and *G* are LUMO classes, *b* is a specific instance instantiating LUMO class *E.*

Similarly, in the case where disinterests[14] exist in the user profile another axiom exists for disinterests, implying a manufactured concept *UDC* (after User Disinterests Concept), that bares the same axiom engineering principles as *UIC*, and for which it holds that it is disjoint with *UIC*, such that:

$UIC \sqcap UDC \sqsubseteq \perp$

4.3.1 Profile formalization, storage and management

As previously described in D4.2, the user profile produced is expressed in a variant of the KRSS[15] ontological formalization. This stems from the need to keep the user profile as expressive but at the same time as lightweight as possible so that it can be storable and manageable in limited resource devices, i.e. the user client. In this way, LinkedTV's personalization services aim to further secure user privacy preservation by minimizing the communication of meaningful sensitive data to an external server, on top of encryption and/or hashing methods used to communicate transaction data wherever in the workflow a communication with the server is mandatory. Therefore the profile formalization choice can be broken down to two points of reasoning:

1. The expressivity of the user profile is much richer than a list of weighted concepts and as such can only be expressed in an ontological formalization - rather than a simpler web standard such as JSON[16].
2. KRSS is significantly more lightweight than other known ontological formalizations/standards (OWL[17], RDF[18]) and as such accommodates storage and management of ontological profiles much more efficiently.

Suffice to say, to explain the latter argument, that the LUMO v1 ontology converted from its OWL/RDF serialization (the preferred public formalization) to the KRSS format (to render it also transportable and usable from the within client in the recommendation phase[19]) bares a reduction of ~89% in size. It is evident that this reduction rate bares a significant beneficiary impact to the storage and management capacity demanded for user profiles when relayed to

[14] By disinterests we denote negative preferences, i.e. preferences that the user has displayed based on his/her transactions a particular *dislike* to, rather than mere indifference to.

[15] http://dl.kr.org/dl97/krss.ps

[16] http://www.json.org/

[17] http://www.w3.org/TR/owl-features/

[18] http://www.w3.org/RDF/

[19] The f-PocketKRHyper reasoner, employed as one of the recommenders in the filtering process and described in D4.5, is capable algorithmically to run in limited resource devices and therefore can make use of the lightweight data to perform filtering within the user client.

a profile continuously populated and thus expanding with more and more information over time.

Table 4 indicates the KRSS serialization of the constructors used, while in addition some conformities used in the construction of user profiles for LinkedTV purposes are exposed.

Table 4: KRSS serialization for the constructors used in the engineering of semantic user profiles

DL constructor	KRSS serialization	Profile construction conformity
$A \sqsubseteq B$	`(IMPLIES A B)`	-
$A \sqcup B$	`(OR A B)`	-
$A \sqcap B$	`(AND A B)`	-
$\exists R.A$	`(SOME R A)`	-
$\neg A$	`(NOT A)`	-
$A \sqcap B \sqsubseteq \bot$	`(DISJOINT A B)`	-
$w \cdot A$	`(WEIGHT A w)`	`w ∈ {0,1}` *Default weight when not specified otherwise:* `w = 1.0`

$\langle a{:}A \rangle \bowtie d$	`(INSTANCE a A ⋈ d)`	`d ∈ {0,1}`
		Instance handling for profile construction:*
		`(IMPLIES inst_a A)`
		`(INSTANCE a inst_a < -1)`

$\langle a,b{:}R \rangle$	`(RELATED a b R)`	-

*** Instance handling in the user profile**

For reasons of compliance with the reasoning logic (first-order logic[20]) principles which is employed by the semantic reasoner used for filtering, we comply with concept instantiation as the means to assert whether a concept is satisfied (i.e. predicate holds true) by a given

[20] http://en.wikipedia.org/wiki/First-order_logic

truth value. To this end, we treat instances in the content annotation (cf. D2.4) as truth values for the reasoning/filtering process and rather treat instances in the profile as concepts extending the domain knowledge base by subsumption via added user-specific predicates in the user knowledge base/profile (i.e. the manufactured axiom: `IMPLIES inst_a A`)). The connection of an instance in the annotation and an instance in the user knowledge base/profile is achieved by the instantiation of the manufactured concept `inst_a` by its instance `a` in the user profile but outside the degree range which can quantify an instance to hold as true (i.e. since `d ∊ {0,1}`, when `d < -1` then `a` is not a truth value for `inst_a`;rather, when `a` is found in a content item with a confidence value `∊ {0,1}` then `a` holds true for `inst_a`).

4.3.2 User profiles based on scenarios

An example of a user profile in the KRSS formalization is the following, involving the persona Ralph and the general description of his interests (cf. D6.1):

```
(IMPLIES (SOME hasInterest (OR profsubconcept1 inst_silbermond profsubconcept2
crafting weather)) ralph)
(IMPLIES inst_silbermond band)
(IMPLIES (AND sports_club inst_bradenburg) profsubconcept1)
(IMPLIES inst_bradenburg user_location)
(IMPLIES (AND olympics athlete) profsubconcept2)
(IMPLIES (SOME hasInterest politics) ralph_dis)
(DISJOINT ralph ralph_dis)
(WEIGHT inst_silbermond 0.82)
(WEIGHT inst_bradenburg 0.85)
(WEIGHT profsubconcept1 0.76)
(WEIGHT profsubconcept2 0.51)
(WEIGHT crafting 0.68)
(WEIGHT weather 0.54)
(WEIGHT politics 0.63)
(INSTANCE silbermond inst_silbermond < -1)
(INSTANCE bradenburg inst_bradenburg < -1)
(RELATED content bradenburg has_interest)
(RELATED content silbermond has_interest)
```

Additional axioms added to enable topic detection (optional, engineered if topic detection is opted):

```
(RELATED bradenburg content hasTopic)

(RELATED silbermond content hasTopic)
```

4.4 Profiling tools REST API

Linked Profiler and its subcomponents are supported by a REST API which gives access to services for retrieving and uploading user information on the profiling server. The API is hosted on a local server of the partner integrating/developing the involved technology.

The calls available are:

GET all profiles available
(For internal use only) Retrieves a list of all user ids available on the server.
Description: GET /api/profiles
HTTP method: GET
Content-Type: application/json
Example of response:


```
{
   "[{"ralph"}]",
   "[{"ralph_context"}]",
   "[{"nina"}]",
   "[{"peter"}]",
   "[{"rita"}]",
   "[{"bert_anne"}]",
   "[{"michael"}]"
}
```


GET a KRSS user profile
Receives a given user profile from the server in the KRSS format. This is the final export of Linked Profiler, incorporating a learned model through both Simple Learner and EasyMiner.
Description: GET /api/KRSS_profile?uid={uid}
HTTP method: GET
Content-Type: application/json
Example of response:


```
{
"Profile":
"(IMPLIES (SOME hasInterest (OR profsubconcept1 inst_silbermond profsubconcept2
crafting weather)) ralph),
(IMPLIES inst_silbermond band),
(IMPLIES (AND sports_club inst_bradenburg) profsubconcept1),
(IMPLIES inst_bradenburg user_location),
(IMPLIES (AND olympics athlete) profsubconcept2),
(IMPLIES (SOME hasInterest politics) ralph_dis),
(DISJOINT ralph ralph_dis),
```

```
(WEIGHT inst_silbermond 0.82),
(WEIGHT inst_bradenburg 0.85),
(WEIGHT profsubconcept1 0.76),
(WEIGHT profsubconcept2 0.51),
(WEIGHT crafting 0.68),
(WEIGHT weather 0.54),
(WEIGHT politics 0.63),
(INSTANCE silbermond inst_silbermond < -1),
(INSTANCE bradenburg inst_bradenburg < -1),
(RELATED content bradenburg has_interest),
(RELATED content silbermond has_interest),"
}
```

GET a plain user profile
Receives a given user profile from the server in its intermediate pre-ontological form (i.e. from Simple Learner)
Description: GET /api/plain_profile?uid={uid}
HTTP method: GET
Content-Type: application/json
Example of response:


```
{
"[{"timestamp":"1.22943E+12","concept":"silbermond:band","aggregated
weight":"22.8682","frequency":"38","positive":"TRUE"}]",
"[{"timestamp":"1.22943E+12","concept":"crafting","aggregated
weight":"14.2667","frequency":"26","positive":"TRUE"}]",
"[{"timestamp":"1.22943E+12","concept":"sports_club, braden-
burg:user_location","aggregated
weight":"0.85","frequency":"1","positive":"TRUE"}]",
"[{"timestamp":"1.22942E+12","concept":"politics","aggregated
weight":"10","frequency":"17","positive":"FALSE"
}
```


PUT a profile on the server
Uploads a LUMO-based content annotation profile in the KRSS formalisation on the local server where the content filtering service resides. Currently supports uploading a file but will be converted into a URI- based call.
Description: PUT /api/upload_profile
HTTP method: PUT
Content-Type: text/plain
Requisite: a .txt file containing a profile in the KRSS format

GET user transactions in LUMO
Receives a given transaction from GAIN and converts the DBPedia-based GAIN output to LUMO-based preferences. Returns the LUMO-based concepts that comprise of interests/disinterests for each transaction. A LUMO wrapper output reception functionality.
Description: GET /api/transformTransaction?uid={uid}
HTTP method: GET
Content-Type: application/json
Example of response:

{
 	
"["sessionId":"1380042132275","objectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-9923b23ef8f5#t=0.099,5.4590000000000005","parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-9923b23ef8f5","interest":"0.01","l_o_organization":"0.46","l_o_periodical_literature":"0.46","l_o_company":"0.23","l_o_agent":"0.69","l_o_newspaper":"0.23"]",
 	
"["sessionId":"1380059132173","objectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-9923b23ef8f5#t=118.069,123.229","parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-9923b23ef8f5","interest":"0.67","l_o_ politician ":"0.83"]",
 	
}

GET user transaction entities in LUMO
(For internal use only) Retrieves the DBPedia-based named entities in a given object within a transaction from GAIN and assigns the corresponding LUMO-based types (preferences) to those entities. Used in conjunction with the previous call to retrieve instantiated concepts as user interests where applicable. A LUMO wrapper output reception functionality.
Description: GET /api/transformTransaction?cid={objectid}
HTTP method: GET
Content-Type: application/json
Example of response:
{ "[{"objectID":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-9923b23ef8f5#t=40.28,45.760000000000005","attributes":{"start":40.28,"end":45.760000000000005},"accountId":"LINKEDTV-TEST","_id":"5241c57a6b4dd3cb72000007","__v":0.0,"entities":{"LUMOtype":"Person","label":"James D. Thurman","entityType":"named en-

```
tity","confidence":0.0,"relevance":0.0},"rating":0.0,"group":0.0,"type":0.0
,"parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5"}]",

"[{"objectID":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5#t=0.099,5.4590000000000005","attributes":{"start":0.099,"end":
5.4590000000000005},"accountId":" LINKEDTV-
TEST","_id":"5241c57a6b4dd3cb72000013","__v":0.0,"entities":{"LUMOtype":"Ne
wspaper","label":"North","entityType":"named en-
tity","confidence":0.85,"relevance":0.0},"rating":0.0,"group":0.0,"type":0.
0,"parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5"}]",

"[{"objectID":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5#t=100.2,106.119","attributes":{"start":100.2,"end":106.119},"a
ccountId":" LINKEDTV-
TEST","_id":"5241c57a6b4dd3cb72000015","__v":0.0,"entities":{"LUMOtype":"Se
ttlement","label":"Pyongyang","entityType":"named en-
tity","confidence":0.85,"relevance":0.0},"rating":0.0,"group":0.0,"type":0.
0,"parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5"}]",

"[{"objectID":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5#t=118.069,123.229","attributes":{"start":118.069,"end":123.229
},"accountId":" LINKEDTV-
TEST","_id":"5241c57a6b4dd3cb72000018","__v":0.0,"entities":{"LUMOtype":"Po
litician","label":"Park Geun-hye","entityType":"named en-
tity","confidence":0.994,"relevance":0.0},"rating":0.0,"group":0.0,"type":0
.0,"parentObjectId":"http://data.linkedtv.eu/media/6026703a-c02c-41bc-9ac3-
9923b23ef8f5"}]",

}
```

5 Contextual adaptation

The contextual adaptation is a key feature which is needed for an efficient user profile setup. If it is clear that a specific user has different interests compared to other people, this profile is also more or less stable. Depending on the user personality, centres of interest and context, the same person can be interested in (very) different concepts, depending on the time, the mood, the immediately previous experience, the other people being around, etc.

This contextual information is of a high importance for user profile dynamical adaptation. A user will behave differently if he is alone or with other users that he knows or not, with kids or adults. He will also wait for different types of content if he is tired after a work day or during vacation at home or in a hotel room. Depending also on his short term previous experience, one concept of interest might have an increased importance as usual. The global societal context (politics, hot news, etc…) can also have an important impact of biasing his user profile. In this section we will describe the kind of contextual features which can be extracted via different means, how the learning of the context is done and how the profiles context is handled and updated,

5.1 Contextual features

This section presents the first implementation of contextual features detected by the LinkedTV context tracking services and their management within the contextualisation workflow.

5.1.1 Contextual features detected by attention tracking

The use of an RGBD sensor providing both RGB images and the depth map let us extract several features which can be used for the context.

There are two categories of features which can be used to provide clues on viewer's context. Some of them are directly related to his context. Among them we can cite: viewer number, age, if they are known or not, etc.

In Table 5 we list the direct contextual features which are planned to be extracted by the direct viewing of the user using a RGBD sensor.

Table 5: The direct contextual features.

Event label	LUMO concept	Description	Message
Viewers_NB	If Viewers_NB = 1 then lumo#Alone. Else if Viewers_NB > 1 then lumo#With_company	Total number of detected viewers. An event is sent when the number of detected people changed for more than some seconds	{"interaction":{"type":"event"}, "attributes":{"action":"Viewers_NB", "value":"number"}}

Event label	LUMO concept	Description	Message
Recognized_ Viewers_NB	If Recognized_Viewers_NB = 1 and Viewers_NB = 1 then lumo:Alone. Else if Recog-nized_Viewers_NB = 1 and Viewers_NB > 1 then lumo:With_acquaintances. Else of if Recog-nized_Viewers_NB > 1 and Viewers_NB > 1 then lumo:With_friends ⊔ lumo:With_family.	Total number of users which are known by the system. If Recog-nized_Viewers_NB=0, no information will be sent as the viewer has no profile to update. If Recognized_Viewers_NB = 1, this means that only the main user is known. If Recognized_Viewers_NB > 1, this means that the main user is watching TV with people that he knows.	{"interac-tion":{"type":"event"}, "at-tributes":{"action":" Recog-nized_Viewers_NB ", "value":"number"}}
Viewer_adults	¬ lumo:With_children	This event indicates the number of adults present during a given context	{"interac-tion":{"type":"event"}, "at-tributes":{"action":" Viewer_adults ", "value":"number"}}
Viewer_kids	lumo:With_children	This event indicates the number of kids present during a given context	{"interac-tion":{"type":"event"}, "at-tributes":{"action":" Viewer_kids ", "value":"number"}}
Viewer_ off_couch	lumo:Area_of_Interest if Viewer_off_couch = 0 ¬ lumo:Area_of_Interest if Viewer_off_couch = 1	Main viewer not in the couch area = 1, Main viewer in couch area = 0	{"interac-tion":{"type":"event"}, "at-tributes":{"action":" Viewer_off_couch ", "value":"boolean"}}
Viewer_standing	lumo:Standing	Main viewer stands in front of TV = 1, the default is the viewer sits=0	{"interac-tion":{"type":"event"}, "at-tributes":{"action":" Viewer_standing ", "value":"bolean"}}
Viewer_looking	lumo:Looking_at_screen	This feature tells if the viewer is looking at the TV or not. If the person is look-ing to the TV a set of coor-dinates is provided. The username of the recognized viewer (face recognition) is also provided in the mes-sage	{"interac-tion":{"type":"event"}, "at-trib-utes":{"action":Viewer_looking", "value":(bool)(val) ,"gaze":{"intersection":"+"[" pixX","pixY"]""}}, "user": {"id":""username"}}

These features are mapped to the events sent to GAIN through the player server and they are presented in four columns. In the first one, the label of the event is written, while the second column displays the correspondence to a LUMO concept which will be derived through transformation of the raw features to the LUMO concept space via LUMO wrapper. In the third column, a short description of the event allows a better understanding of the feature and the last column shows the message as sent (or planned to be sent) through the network and the player server to the GAIN component.

Other features are indirectly linked to a viewer context (viewer clicks, volume up, etc.) Taken separately, those features do not provide context information, but mainly viewer interest information. Also, these features do not comply with the definition of context, which is an event that spans over a period of time. These instantaneous events are described in Chapter 5.1.2.

5.1.2 Interest clues passed from Kinect to GAIN

In addition to the contextual features described in the previous section, Kinect provides also several features, corresponding to instantaneous events, that can be used as interest clues. These are summarized in Table 6.

In effect, interest clues serve as the means to determine the degree of interest[21] that the user displays for the concept/s within the content item (media fragment, specific annotation of a media fragment, or enrichment content) that they interact with.

Table 6: The interest clue features.

Label	Description	Message
Play/Pause/Stop	Message coming from the gesture control module to start/pause/stop the player	{"interaction":{"type":"event"},"attributes":{"action":"Play"}}" {"interaction":{"type":"event"},"attributes":{"action":"Pause"}} {"interaction":{"type":"event"}, "attributes":{"action":"Stop"}}
Volume	Gesture message to modify the player volume	{"interaction":{"type":"event"},"attributes":{"action":"Volume","value" : -)int)(val)}}

[21] The computation of the degree of interest based on each action is described in D4.2, Chapter 4.3.

Label	Description	Message
Next/Previous	Messages from gesture control module allowing the user to jump between different multimedia fragments, videos, etc.	{"interaction":{"type":"event"},"attributes":{"action":"Next"}} {"interaction":{"type":"event"}, "attributes":{"action":"Previous"}}
Bookmark/ UnBookmark	Message from the gesture module allowing to bookmark/unbookmark a media fragment	{"interaction":{"type":"event"}, "attributes":{"action":"Bookmark"}} {"interaction":{"type":"event"}, "attributes":{"action":"UnBookmark"}}
Click/Select	Message from the gesture module allowing to click on/select a link (enrichment content or specific annotation in a seed media fragment)	{"interaction":{"type":"event"}, "attributes":{"action":" Click", "value":"(int)(val)" }}
Lock/Unlock	Message send to unlock/lock the player in a similar way that we find on smartphones but here with a define and gesture	{"interaction":{"type":"event"},"attributes":{"action":"Lock"}} {"interaction":{"type":"event"}, "attributes":{"action":"Unlock"}}
Mute/Unmute	Gesture-based message send to mute/unmute the player	{"interaction":{"type":"event"},"attributes":{"action":"Mute"}}" {"interaction":{"type":"event"},"attributes":{"action":"Unmute"}}
Viewer_looking	See Table 5	See Table 5

Those features are mapped to the events sent to GAIN through the player server and they are presented in three columns. In the first one, the label of the event is written, a short description of the event allows a better understanding of the feature and the third column shows the message as sent (or planned to be sent) through the network and the player server to the GAIN component.

All the features/events from tables Table 5 and Table 6 are not yet implemented, and depending on the project evolution and on the needs in terms of interface especially for the gesture control module, the number of events might evolve. The evolution will depend on user tests, on the possible availability of novel technologies (like the Microsoft Kinect 2 sensor) and on the efficiency and stability of the features.

It should be noted that GAIN can use some of the contextual features detected by the Attention Tracker (see Subsection 5.1.1) also as interest clues. This applies especially to the Viewer_looking feature described in Table 5.

5.1.3 Contextual features detected by the player

Contextual features can be raised also by the player. Examples of these events include time of day (morning, afternoon, evening, night) and location (e.g. at home/second home/travelling).

The contextual features raised by Kinect are relayed through the player. In the previous two sections, the messages that communicate the events captures from the tracking mechanisms to the media player (LOU) were analyzed. Table 7 demonstrates the communication of the contextual clues from LOU to GAIN.

As of time of writing, one additional contextual feature raised directly by the player has been defined – whether the TV is muted or not. In a later stage the time/date of user actions if taken into account on a short to medium-term history will be considered as contextual features.

Table 7: Contextual clues raised by LOU in GAIN

Label	Description	Message

Viewer_Looking	Message originating at the Kinect passed through player
	
	value: “true” if main user is looking or “false” if the main user is not looking
	confidence: always 1
	client.type: the primary source of the event
	location: time in video (seconds)
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "context",
		"userId":
		"http://example.com/users/user1",
		"objectId":
		"http://example.com/objects/object1",
		"attributes": {
		”action": ”looking”,
		“value”: ”true”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

Recognized_Viewers_NB	value: integer (sent as string - quoted)
	Number of currently recognized viewers.
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "context",
		"userId":
		"http://example.com/users/user1",
		"objectId":
		"http://example.com/objects/object1",
		"attributes": {
		”action": ”viewers”,
		“value”: ”1”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”

Label	Description	Message
		} } }

Viewer_adults	value: integer (sent as string - quoted)
	Number of currently recognized adults.
		{
		"accountId": "LINKEDTV-TEST",
		"type": "context",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		”action": ”adults”,
		“value”: ”1”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

Viewer_kids	value: integer (sent as string - quoted)
	Number of currently recognized children.
		{
		"accountId": "LINKEDTV-TEST",
		"type": "context",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		”action": ”children”,
		“value”: ”1”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

Viewer_off_couch	value: true if the main viewer is out of the couch area, false if the main viewer is in the couch area.	{
		"accountId": "LINKEDTV-TEST",
		"type": "context",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		”action": ”off_couch”,

Label	Description	Message

		“value”: ”true”,
		"location": "32",
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

Viewer_standing	value: true if the main viewer is standing, false if the main viewer is sitting.	{
		"accountId": ”LINKEDTV-TEST",
		"type": "context",
		"userId":
		"http://example.com/users/user1",
		"objectId":
		"http://example.com/objects/object1",
		"attributes": {
		”action": ”standing”,
		“value”: ”true”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

TV_muted	value: true of false
	True if the TV set is muted, otherwise false.
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "context",
		"userId":
		"http://example.com/users/user1",
		"objectId":
		"http://example.com/objects/object1",
		"attributes": {
		”action": ”mute”,
		“value”: ”true”,
		"location": "32",
		“confidence”: ”1”,
		“client” : {
		“type”: “lou/kinect”,
		“version”: “0.0.1”
		}
		}
		}

5.1.4 Interest clues detected by the player

There are two primary sources of interest clues: Kinect (either explicit gestures or clues originating from observation of user behaviour) and explicit user actions done with the remote control (i.e. player actions). Many of the events can be raised by both Kinect and the remote (e.g. the Play actions). To GAIN, the primary origin of the event is transparent, although GAIN is given the information about the primary origin of the event. The utilization of this information might be the subject of a future extension. The “volume manipulation” events are complex to interpret as interest clues and therefore will be left for future work.

Table 8: Interest clue events raised by LOU in GAIN

Label	Description	Message

Play/Pause/Stop	action: “Pause”, “Stop”, “Play”
	value: time in video
	client.type: “lou/remote” or “lou/kinect” dependig on the primary source of the event
	location: time in video (seconds)
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "event",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		"category": "Video",
		"action": "Pause",
		"location": "32",
		“client” : {
		“type”: “lou/remote”,
		“version”: “0.0.1”
		}
		}
		}

Next/Previous	action: “Next”, “Previous”,
	value: time in video
	client.type: “lou/remote” or “lou/kinect” dependig on the primary source of the event
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "event",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		"category": "Video",
		"action": "Next",
		"location": "32",
		“client” : {
		“type”: “lou/remote”,
		“version”: “0.0.1”
		}

Label	Description	Message
		} }

Bookmark/
UnBookmark
	action: “Bookmark”, “Unbookmark”,
	value: time in video
	client.type: “lou/remote” or “lou/kinect” dependig on the primary source of the event
	
	Bookmark of the media fragment.
		{
		"accountId": ”LINKEDTV-TEST",
		"type": "event",
		"userId": "http://example.com/users/user1",
		"objectId": "http://example.com/objects/object1",
		"attributes": {
		"category": "Video",
		"action": "Bookmark",
		"location": "32",
		“client” : {
		“type”: “lou/remote”,
		“version”: “0.0.1”
		}
		}
		}

Click/Select	Status of action label and value pending implementation. client.type: “lou/remote” or “lou/kinect” dependig on the primary source of the event Click on/select a link (enrichment content or specific annotation in a seed media fragment).	The design of the reception of the click/select action’s message from the player for the gesture module is currently under implementation. Therefore overall status of message pending implementation.
Mute/Unmute	This is handled as contextual event	This is handled as contextual event – see Table 7
Viewer_looking	This is handled as contextual event	This is handled as contextual event – see Table 7

5.2 Context learning

5.2.1 Handling context in GAIN

For handling context, GAIN adds a new temporal interaction type “context”. When the context starts to be valid, the data provider (player) raises a *context start* event in GAIN by setting the value for the contextual feature. When the context is no longer valid, the data provider raises the *context end* event in GAIN by setting the contextual feature to a different value.

All the interactions that happen between the timestamps of the “start” and “end” interactions are assumed to happen in the given context. If the *context end* interaction is not sent (i.e. the contextual feature is not set to a different value), the context is terminated by GAIN at the end of the session (on the user log out event). The contextual features are passed onto the Linked Profiler as displayed in Table 2.

GAIN aggregates interactions into one vector per given timespan. The foreseen timespan granularity is one shot. For the given contextual feature, the value that spans for the largest part of the shot is used, or no context is set in case “no context” (the contextual feature is not set e.g. because Kinect was not switched on) spans for the largest part of the shot.

Example: Consider event stream from player to GAIN for a 26 seconds log shot. Nine seconds into the shot, additional person, Peter, enters, but stays only 11 seconds. Nina, the logged-in user, is looking at the screen for 10 seconds, when she is interrupted by Peter. One second after Peter leaves at second 15, Nina resumes looking at the screen. Altogether, Nina is looking at the screen for 20 seconds out of 26.

…	…
0:00:50	Context start/Viewer_looking=true Context start/Recognized_Viewers_NB=1
…	…
0:01:00	--- SHOT 8 STARTS --
0:01:09	Context start/Recognized_Viewers_NB=2
0:01:10	Context end/Viewer_looking=false
0:01:15	Context end/ Recognized_Viewers_NB=1
0:01:16	Context start/Viewer_looking=true
0:01:26	--- SHOT 8 finishes ---

The result of the context aggregation from the above-described example is as follows:

Since the logged-in (primary, recognized) user was looking at the screen for 20/26=0.77% of time, the value for *Viewer_looking* context is set to true. In contrast, the additional person, Peter, is present only for 6/26=23% of the shot duration, therefore the *Recognized_Viewers_NB* context is set to 1.

UserID	ShotID	c_looking	c_viewers
U1	S8	true	1

As an extension to the described mechanism for context handling, *permanent context* can be considered. Permanent context would be set by the player to a fixed value per user. The validity of this context would be permanent (i.e. the session end event does not terminate the context). The context can be changed only by the player setting it to a different value.

5.2.2 Handling context during learning in EasyMiner

In EasyMiner input, the contextual features are reflected in two ways:

- for each context type, a contextual feature is created in the data (as described in 5.2.1),
- selected contextual features are used as one source of evidence in computation of user interest in the given shot. Contextual features can be dealt with in the same way as "physical" actions (events) as exemplified in D4.2 in Table 14.

Within EasyMiner, the contextual features can be either used in the antecedent of the rule in the same way as all other features, or in the "condition" part of the rule (see Section 5.3.1.3 in D4.2). The latter assumes that sufficient data has accumulated (the condition creates a submatrix containing only the rows matching the context).

5.2.3 Handling context in Linked Profiler

Simple Learner also receives the information stemming from GAIN and the contextual rules from EasyMiner. In the next steps of implementation, the following methodology is going to ensure contextual adaptation of the long-term user profiles.

For each transaction received by GAIN, the contextual attributes are subjected to the LUMO wrapper transformation from the raw contextual features to concepts denoting contextual attributes within LUMO.

In Simple Learner, the non-rule concepts in the intermediate profile are updated with the contextual features in a separate entry, namely “context”, a vector that holds all contextual aspects of the user where this concept was consumed:

```
"timestamp":"1.22943E+12",
"concept":"silbermond:band",
"context":"with_acquaintances, alone",
"aggregated weight":"22.8682",
"frequency":"38",
"positive":"TRUE"
```

In the case of rule representation where the context features are a part of the antecendent as a conditional statement, the context entry will hold the LUMO-based contextual feature :

(case #1)

```
"timestamp":"1.22943E+12",
"concept":"sports_club, bradenburg:user_location",
"context":"alone",
"aggregated weight":"0.85",
"frequency":"1",
"positive":"TRUE"
```

In the case of rule representation where the context features are part of the contecendent, the rule is passed directly onto the “concept” entry of the intermediate profile:

(case #2)

```
"timestamp":"1.22943E+12",
"concept":"sports_club, bradenburg:user_location, alone",
"context":"",
"aggregated weight":"0.85",
"frequency":"1",
"positive":"TRUE"
```

The final user profile will include the contextual features as a complex concept (axiom) through a conjunctive constructor in the semantic user profile, such that:

(case #1)

```
(IMPLIES (SOME hasInterest (OR (AND profsubconcept1 alone) (AND inst_silbermond
with_acquaintances alone) …)) ralph)
(IMPLIES inst_silbermond band)
(IMPLIES (AND sports_club inst_bradenburg) profsubconcept1)
(IMPLIES inst_bradenburg user_location)
(WEIGHT inst_silbermond 0.82)
(WEIGHT inst_bradenburg 0.85)
(WEIGHT profsubconcept1 0.76)
(INSTANCE silbermond inst_silbermond < -1)
(INSTANCE bradenburg inst_bradenburg < -1)
(RELATED content bradenburg has_interest)
(RELATED content silbermond has_interest)
```

Or

(case #2)

```
(IMPLIES (SOME hasInterest (OR profsubconcept1 (AND inst_silbermond
with_acquaintances alone) …)) ralph)
(IMPLIES inst_silbermond band)
(IMPLIES (AND sports_club inst_bradenburg alone) profsubconcept1)
(IMPLIES inst_bradenburg user_location)
(WEIGHT inst_silbermond 0.82)
(WEIGHT inst_bradenburg 0.85)
(WEIGHT profsubconcept1 0.76)
(INSTANCE silbermond inst_silbermond < -1)
(INSTANCE bradenburg inst_bradenburg < -1)
(RELATED content bradenburg has_interest)
(RELATED content silbermond has_interest)
```

6 Explicit User Interest Models

User models and their usage in content filtering for personalization and contextualization have many facets: how to build them explicitly or from user observations, how to maintain them and adapt them to changing user interests, etc. In this chapter a complementary approach to modelling explicit user interests is described. It is currently based on user models which are explicitly specified by users. This allows us to formulate expressive and precise user interests[22] by extending the profiling methodology described in the previous chapters by allowing the expression of a plurality of non-taxonomical relations among preferred concepts, defined explicitly within the user profile. In D4.5 we outline how it is used for personalization and contextualization through semantic content filtering. It becomes clear that expressive user interest models provide attractive opportunities to describe the interests a user has in multimedia content with more precision and, consequently, improved recommendation quality compared to the simpler models outlined in the previous chapters. On the other side, it is far from trivial to build and manage such more expressive user interests with current modelling capabilities (user interfaces, preference learning, etc.), and to find media fragment annotations with sufficient semantic precision.

6.1 LUMO-rel

The LUMO-rel ontology introduced in this chapter was created as a trade-off between 'too simple' and 'too complicated'. 'Too simple' means not sufficiently detailed, not expressive enough, not comprehensive enough[23] to support users to find effectively what they are interested in. If we just have top-level concepts like person, location, event, and maybe a few sub-concepts we are severely restricted in describing video content as well as user interests. Users will not be sufficiently supported finding what they are interested in. If we cannot relate entities to each other through semantic relations and not express user interests in such terms we are too restricted, too. We need a fine-grained concept hierarchy *and* relations in our ontology.

The LUMO-rel ontology was built as an extension of the LUMO-V1 ontology presented in Section 2.1[24] with the aim to investigate and expand on object properties that can provide more non-taxonomic relations between concepts. It is a curated, broad-scope, light-weight ontology. It is curated in order to avoid the semantic shortcomings of current LOD ontologies. It is broad-scope in order to describe the broad spectrum of multi-media content as precisely

[22] In Year-3 we plan to investigate how such expressive models can be built from user observations.

[23] This may depend on the concrete usage. If a user wants to formulate her interests just related to a concrete (local or thematic) TV channel much less expressive means may be sufficient compared to a broad spectrum media portal. The German broadcaster RBB just brings news from the Berlin Brandenburg region – so local restrictions of user interests can be left implicit resulting in significantly simpler expressions for them.

[24] In Year-3 of the project we will work on a merged version of our ontologies in order to combine the advantages of both versions.

as possible. It is light-weight because formal ontologies are currently not practical on the annotation, enrichment, and user interest side.

LUMO-rel provides a comprehensive taxonomy of concepts and a set of semantic relations between them (see Figure 13). The top-level concepts cover the main kinds of things and topics people may be interested in:

- person,
- organisation,
- living being,
- object,
- event,
- location,
- time, and
- topics.

(see also the LUMO ontology we presented in D4.2 and related ontologies like the BBC content ontology [BBC] or the NERD top level ontology [NERD]).

Topics are *abstract things* like politics, science, history, arts, etc. Currently, we do not provide any means to deal with assumptions, opinions, suggestions, etc. because they are much harder to represent precisely, because the media fragment annotations do not deliver them, and reasoning on them is significantly more complicated.

The top-level concepts are refined into more precise sub-concepts (see Figure 13). This refinement collects those sub-concepts[25] where people are able to express their more specific interest in.

[25] In a first implementation, this collection was performed empirically and will be subject to refinement after dedicated user trials.

Figure 13: Second level of the LUMO-rel taxonomy

We did not introduce common high-level concepts though some concepts show some commonalities. Persons are, of course, living beings; somehow they are also objects; and persons and organisations share some properties of "agentness" in events or actions. On the other side, why should users be interested in farm animals or African giraffes also be interested in politicians or business men – though they all are living beings? High level ontology concepts like "agent" or "intangible" cannot help people to express their interests in something. Interests are more concrete. Consequently, we focused the LUMO-rel ontology on this level of abstraction.

In a similar pragmatic way we decided which *relations* to include into LUMO-rel (see Figure 13). It would be very attractive to have more semantic precision through more differentiated relations. The main limitations here come from the annotation side, where automatic semantic relations retrieval is a demanding, open research subject. While research on the matter progresses we expect to be able to integrate dedicated solutions into our system. Meanwhile, we can resort to user-contributed annotations of the content where interests are expressed with better precision through designated object properties

From formal knowledge representation theory we know that concepts can be combined to build more fine-grained concepts through logical relationships and "concept forming operators" like existential or universal quantification on relational expressions [StSt04]. LUMO-rel follows a pragmatic trade-off: we use semantic relations to form more precise concepts but avoid the full expressive power of description logics, which is not compatible with the light-weight annotation approach followed by LinkedTV. We do not just have politicians but French socialist politicians - without introducing this concept explicitly (which would blow up the concept space considerably!). We simply combine the concept *politician* with a semantic relation to *France* (an instance of the concept *NationalState*) and to the topic *socialism* as a subtopic of *political direction*. In this way a user can express her interests with more precision through *unnamed concepts*. At the same time we may find out (from DBPedia or similar LOD sources) that a certain person is a politician, is French, and related to socialism. Without this definition it may be hard to find out directly from LOD that this person is a *French socialist politician*.

Instead of discriminating people as politicians, athletes, composers, etc. we could have used here the same relations approach, too. We would not introduce a concept *politician* but express this just through the concept *person* and the *dealsWith* relation to the topic *politics*. On the other side, our experience shows that users will frequently describe people primarily by the role they play in society, i.e., by their profession or position. As a consequence, we opted to use profession as the main discrimination criterion for the sub-concept space of person. Additionally, we introduced attributes like *gender* or *age* to discriminate persons in user interests. Using age as a filter for media fragment annotations implies to have corresponding reasoning capabilities.

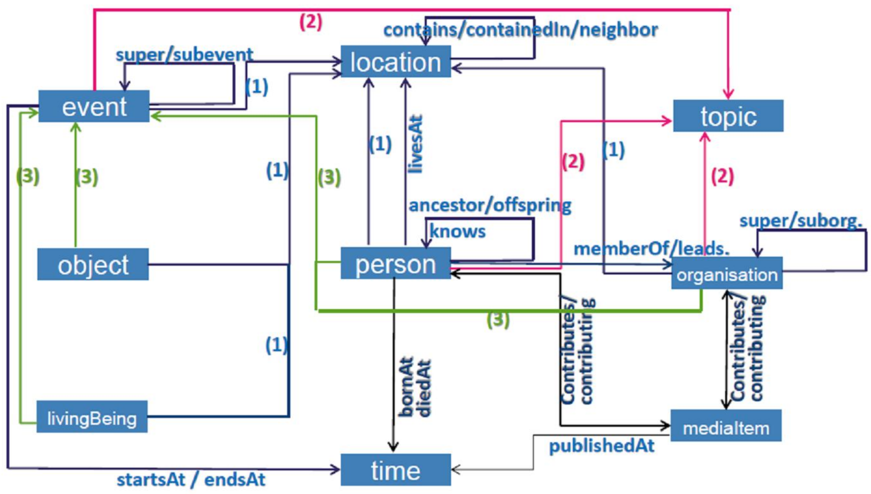


Figure 14: Schematic overview of the LUMO-rel ontology top level. (1) is the locatedAt relation, (2) the dealsWith/treatedBy relations, and (3) the involves/involvedIn relation pair.

This combination of explicit concepts and relations allows us to form a large *concept space* of unnamed concepts [StSt04]. This enables users to express their interest in a large number of things quite precisely.

6.1.1 LUMO-Pedia

The LUMO-rel ontology is the semantic corner stone of our approach – but needs an appropriate knowledge base as compliment containing the correspondingly shaped *concrete* knowledge. We nickname[26] this knowledge base LUMO-Pedia as a LUMO-rel compatible version of Wikipedia- or DBPedia-like knowledge. It is a set of curated named entities related to the LUMO ontology. Somehow it can be considered as an extension of Wikidata [Wikidata] with a more elaborated taxonomy/ontology.

Practically, the current version of LUMO-Pedia is built from DBPedia, Yago2 [HOF13], and some other LOD sets. An instance needed to express user interests or to annotate a video is transformed from its DBPedia or Yago2 counterpart into a LUMO-Pedia instance. We assign each LUMO-Pedia instance to its most specific concepts in the LUMO ontology using heuristic mapping rules.

In a similar way the relations are treated. DBPedia, Yago2, Wikidata, and other LOD sources contain a lot of relational information – but in a way which needs consolidation. We use heuristic mappings from LOD to LUMO-rel relations.

As a summary, LUMO-Pedia is a knowledge base consisting of named entities (instances) assigned to LUMO-rel concepts and connected to each other through LUMO-rel relations. This knowledge base is used to annotate media fragments, to enrich MF annotations semantically, and to formulate user interests.

6.1.2 User Interests

Users can express their interests in MF content directly through annotations in terms of LUMO-rel and LUMO-Pedia. Every user interest is an expression combining LUMO-rel concepts and topics, LUMO-rel relations, and LUMO-Pedia instances.

By reasons of practicality we currently apply some restrictions to the shape of user interests: a user interest is represented as a concept or a topic from LUMO-rel, or a named entity from LUMO-Pedia - the “anchor” expression. This anchor may be restricted through LUMO-rel relations to other concepts, topics, or named entities. Users may say that they are interested in political meetings involving Barack Obama and located in Berlin where political meeting is a LUMO-rel concept, involves and locatedIn are LUMO-rel relations, and Barack Obama and Berlin are LUMO-Pedia named entities.

Alternatively, users may express their interests in political meetings involving US presidents in Berlin, or political events located in Germany. US president is a LUMO-rel concept sum-

[26] To build such a knowledge base is, of course, a bigger effort. We just want to outline its principles and how to use it in the context of our research.

marizing all office holders, political event is a LUMO-rel concept more general than political meeting, and Germany is a LUMO-Pedia instance of the LUMO-rel concept NationalState.

LUMO-rel concepts, topics, relations, and LUMO-Pedia entities can be combined to express quite complex user interests. Logically, each user interest is a conjunction of expressions where the “anchor” expression is a concept, topic, or named entity, and the other expressions are existentially quantified relational expressions restricting this “anchor” through connections to related concepts, topics, or named entities. A more formal description of user interests expressed in terms of LUMO-rel and LUMO-Pedia is given in D4.5 in conjunction with the filtering procedure.

The expressivity for user interests has been tailored in such a way that interests can be expressed as easy as possible while at the same time allowing for still great expressivity. We provide a graphical editor enabling users to express such user interests interactively. If more appropriate means will be available to support users in this respect (for instance natural language translations into logical expressions) the expressivity can easily be extended.

6.1.3 User Interest Models

In contrast to concrete user queries which will frequently be very precise long-term user interests need more general expressions. If a user knows that Barack Obama was at a state visit in Berlin yesterday he can search for corresponding videos just with these “tags”. A long-term user interest will typically not be so focused but contain more general expressions like “political events in Germany” or “international sport events”.LUMO-rel allows us to express such broader interests.

We can form user interest models (UIM) as basis for personalised media consumption. User interest models are just collections of user interests. They can be formulated explicitly by a user and collected in his user model. They can also be adapted by the users following their changing interests using preference learning (cf. Chapter 4.2).

6.1.3.1 Weights and ranking

In order to allow users to express their interests with varying degrees we assign weights to user interests. Currently, we are using real numbers in the interval [0,1] in order to assign weights to user interests. Additionally, we can explicitly express “dislike” – with the meaning that every MF matching with a “dislike” user interest is filtered out – regardless how well it matches with other UI.

The weights are used to rank all those MF which fulfil one of the user interests in a user model: each MF fulfilling a user interest is weighted with the highest weight of a matching user interest. In this way the set of all filtered MF can be ranked according to the weighted user interests.

Currently, MF annotations are not weighted. Assigning weights to them can help to represent MF content more precisely because things shown in a shot or scene have different importance in the context of the video (in an interview the interviewed person is normally more important than the journalist or the microphone).

6.1.3.2 Contextualization and active user interest models

User interests may depend on the context in which the user just is. A user watching TV at home at the weekend with his family may have different interests than being alone at home late at night or on holidays. We enable a user to structure his user interest model into different contexts. These contexts can form a hierarchy each containing specific user interests which are also inherited to all of its sub-contexts.

For instance, a user may decide to structure his UIM in contexts (with grey color) as follows:

myUIM

- single at home
  - politics
    - politicalEvents located in Germany – 0.8
    - politicalEvents dealWith EuroFinanceCrisis – 0.85
    - …
  - sports
    - Football involving GermanBundesliga – 1.0
    - EuropeanChampionship – 0.8
    - WorldChampionship – 0.8
    - OlympicGames – 0.9
- with family at home
  - sports
    - Figure skating – 0.9
    - Swimming – 0.5
  - entertain
    - popMusic – 0.8
    - shows – 0.8
- …

Some more examples can be found in the next chapter on concrete user interest models.

The structural hierarchy of the contexts in the UIM is completely at the user's disposal. If the user has different interests being at home alone or with his family, or if separate contexts are defined for politics, sports, entertain, etc. – is simply the user's decision. A user may also decide to put all interests into one context.

The user can *activate* one of his contexts – making it the active UIM. All user interests in this activated context including those inherited from the higher contexts in the context hierarchy are used to filter and rank videos. A user may also select just one of his interests as the current active UIM to be used for filtering. Later we may support[27] users in selecting the "right" active UIM by some context rules taking user data, location, time, other people joining him, etc. into account. This can be achieved when the explicit user modelling services will be fully integrated with the context detection and management tools described previously in Chapter 5.

6.1.4 User model descriptions

According to our use case scenarios (D6.2) we created three user models[28] in our LinkedTV User Model Editor LUME available for testing issues: Nina, Lukas and Peter. To access the user models you can login at http://data.linkedtv.eu:8081/LUME3-war/user.xhtml via:

- o Username: lukas, Password: demo
- o Username: nina, Password: demo
- o Username: peter, Password: demo

These user models are manually created based on the first set of 69 video annotations from WP2. This means that the interests in each model are fitted to the video content. So we have comprehensive user models for generating appropriate recommendations about this small set of videos. In the following we outline the interest profiles of our three users. Additionally you will see the extracted preferences which are modeled via LUME.

The first step is to recognize the relevant information from the interest description and to decide how to model this issue in LUME. Based on the LUMO-rel ontology we can model the preferences either as single concept, as instance, or as a combination of concepts and instances via predefined relations. The expressivity provided by the LUMO-rel ontology allows us to express very generic interests as well as very focused ones (see examples below).

The second step is to define what the main interests of a user are which apply to all contexts a user can be in, and what special interests in specific contexts are. This is important because each interest is assigned to one context. There is a root context that inherits its interests to underlying sub contexts. Therefore, when we decide that a specific interest appears in the root context you will also find it in all sub contexts and inherited interests will be incorporated during the semantic filtering process.

[27] Currently it is not clear how much support users want to have at this point. Some users may prefer to have full control on their media consumption – selecting manually the active context, others may like to be guided by a context management tool. This will be determined based on dedicated user trials which will be conducted when the profile presentation will be integrated fully with the LinkedTV player and platform in year 3.

[28] The user models in the meantime have been further remodeled and detailed in D6.1 and D6.2 and will be accordingly adapted to the user model descriptions in the (near) future.

In a third step we have assigned weightings to each preference in the user model to express its user relevance (see also D4.3 where the user models have been introduced initially according to our use case scenarios). For this, we have made assumptions between a high interest (1.0) and an absolutely dislike attitude.

Nina

Nina is an urban mum living in Berlin **(Interest: Berlin)** since birth. Because of her child she is highly focused on healthy food **(Interest: Health *restrictedTo* Food)**. On her bill of fare you can find a lot of healthful food products, mainly fruit **(Interest: Fruit)** and vegetables **(Interest: Vegetable)** and only sparse meat **(Disinterest: meat)**. She likes to **cook (Interest: cooking)** by herself und cultivates her own food in the garden. Sometimes she goes out for having a dinner in one of the upper class restaurants **(Interest: restaurant)** around Berlin. Therefore, she is willing to spend some more money to get high quality food instead of well-known fast-food restaurants. You can see her engagement for healthy eating by her shopping behavior. She mostly prefers biological-certified food and uses offers from local farmers.

Her child visits a school in Potsdam **(Interest: Potsdam)**. Nina works at the **Berlin airport (Interest: Airport *locatedAt* Berlin)** since 2006 and she is responsible for the statistical processing of flight data generated by the daily traffic. Her studies of mathematics **(Interest: Mathematics)** and her affinity in natural science **(Interest: NaturalScience)** support her work. In her leisure time **(Context: leisure time)** she often watches documentations about physical experiments **(Interest: documentation about Physics)** and reads articles about new air and space transportation technologies.

Nina has very special sports favorites. Although she is not very sporty she likes to visit sports events in Berlin when her favorite football team “Hertha BSC” is involved. **(Interest: HerthaBSC, sportEvent *locatedAt* Berlin *dealsWith* HerthaBSC)** Besides, she is a big boxing fan **(Interest: Boxing, sportEvent *locatedAt* Germany *dealsWith* boxing)** and already has participated in live events. But mostly she wants to watch the live broadcast on TV. It does not bother her to get up at night when a boxing event takes place somewhere in the world. When she is watching the daily news she only wants to see the sports part. She is not very interested in local politics **(Disinterest: Politics)** or something else. And sometimes she dreams about skipping all other news reports in the news TV until the sports parts appears.

In her leisure time Nina reads a lot or visits exhibitions in museums **(Interest: Exhibition *locatedAt* Museum)**. Her favorite author is Theodor Fontane **(Interest: Theodor Fontane)**. She has a lot of books written by him and when she is on holiday **(Context info: on holiday)** she always carries one of his novels **(Interest: novel *author* Theodor Fontane)** in her handbag. Whether she is on the beach, in a hotel or wherever, she finds enough time to read. It is very important for her to relax after a hard working day. She also likes painting **(Interest: painting)** and likes to visit museums where paintings are exhibited **(Interest: Exhibition *dealsWith* painting)**. In this way she is looking for new impressions for her own paintings.

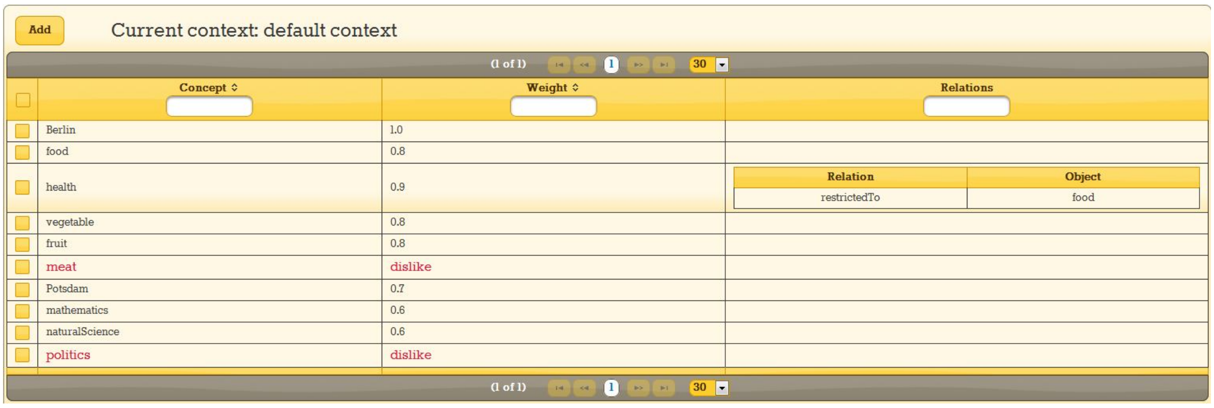
Nina is always very confused about her daughter. She is afraid of a typical problem in her living area that deals with alcoholism **(Interest: alcoholism)**. A lot of children in her daughters' age hang around in city parks and drink a lot of beer and Nina fears that her daughter could get into this scene. So she is always well-informed about all aspects of alcoholism and it is very important to her to educate her daughter about the dangers of alcoholism. To stay up-to-date she regularly watches news or reports about drug problems in our society and addiction prevention measures **(Interest: drug_addiction)** to protect her daughter.

Modeling Nina

In Nina's user model you can find seven selectable contexts that we have extracted from her interest description.

Figure 15: LUME Screenshot about Nina's context directory

The default context, or root of all contexts, contains all concepts which are in any case fundamentally important for her information needs.



Add Current context: default context

(1 of 1) 1 30

Concept	Weight	Relations
Berlin	1.0	
food	0.8	
health	0.9	Relation: restrictedTo; Object: food
vegetable	0.8	
fruit	0.8	
meat	dislike	
Potsdam	0.7	
mathematics	0.6	
naturalScience	0.6	
politics	dislike	

(1 of 1) 1 30

Figure 16: LUME Screenshot about Nina's preferences in her 'default context'

In this default context we have added all necessary interests that become relevant whenever Nina uses the recommendation system. This could be the first active context after logging into the LinkedTV player. The semantic filtering process could consider this preference list for the initial video recommendation. As soon as Nina activates an underlying sub context, like "Sports", you can see the inherited preference list from the default context (green lines) and the additional interests which are only relevant in this specific sub context. By testing the Linked Semantic Filtering we have recognized that certain concepts in the default context may distort the recommendation list in a crucial way. Some concepts have dominated this list in that way, that the top of each recommendation list included videos about these concepts.

To avoid this, we have outsourced the concepts “alcoholism” and “drug_addiction” into a “special interest” context.

Figure 17: Screenshot about Nina's preferences in her "special interest" context

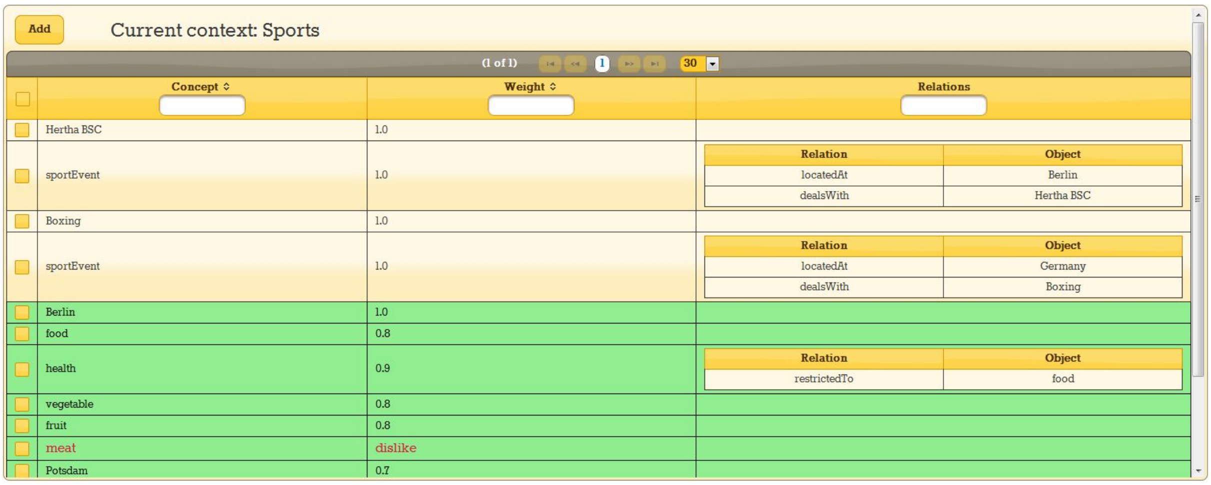


Figure 18: LUME Screenshot about Nina's preferences in her 'Sports' context

Depending on the changing contexts the recommended videos will be adapted through the filtering process. Here, the recommendation list will show more videos containing sport annotations.

Lukas

Every morning, before going to work, Lukas goes for a walk with his dog **(Interest: dog)** in the park. Often he is annoyed that his pet does not obey properly. One day, he remembers that there is a special program on TV about training of dogs. He does not know the show’s name but it is broadcasted in the later afternoon program **(Context info: leisure time)**. So Lukas decides to spend time in that education **(Interest: Education *restrictedTo* Dog)** issue in the future.

Lukas is born in Flensburg **(Interest: Flensburg)** and 10 years ago he moved to Berlin **(Interest: Berlin)** because of a new job offer in a museum of north-american culture **(Interest: Culture *restrictedTo* NorthAmerica)**. Since he was a young boy, Lukas shows a great fascination in this topic **(Context info: Default context).** For his work it is important to know a lot about the colonization in the past centuries. And also the religious impact **(Interest: Religion)** of European settlers **(Interest: Religion *restrictedTo* Christianity)** on the Native

American people. In the next month there will be a new exhibition about the Second World War. Lukas' job is to prepare some video presentations about Jewish emigrants **(Interest: Religion *restrictedTo* Judaism)** during the war and how they were treated. For this, he has to watch a lot of archive material. Additionally, Lukas is looking for recent documentations in social media. He browses through Youtube videos and is surprised about the huge amount of relevant content. It is impossible to watch all these videos. He notices that there are two different types of relevant videos. Long videos (more than an hour) about the north American culture at all and shorter videos with specific detailed topics. This circumstance provides a good idea to Lukas. When he is at work he wants to watch the longer videos **(Context info: at work)** because he has enough time to concentrate on it. And when he is at home, sometimes interrupted by his dog, he watches the shorter videos. **(Context info: busy at home)**

Lukas is highly interested in Politics **(Interest: Politics)** and Economy **(Interest: Economy)**. Whenever there are political events in Berlin he tries to participate. Especially when he expects to get new information about the handling of issues in the European Union **(Interest: EuropeanUnion)** and the economic situation in Berlin **(Interest: Economy *restrictedTo* Berlin)** and Germany **(Interest: Economy *restrictedTo* Germany)**, Lukas gets all available information. He is a little bit confused about Berlins' economic development **(Interest: Politician *dealsWith* Finance and *belongsTo* Berlin)**. Interviews of local politicians **(Interest: Politicians *belongsTo* Berlin)** in "rbb aktuell" news show are a very important information source for Lukas when he doesn't have enough time to read newspapers. So at least one time per day he watches "rbb aktuell" **(Context info: RBB Aktuell)**. Sometimes he watches it in the morning **(Context info: in the morning, at home)** and sometimes in the evening during dinner time **(Context info: in the evening, at home, during dinner)**.

Regularly, Lukas and some friends go to their favorite driving range in one of Berlin's golf sites **(Interest: Golf)**. Between the drives they often talk about actual political topics. Whenever it is possible he takes action with his friends. One of them is a hobby-sculptor with great aptitude. Lukas likes this art handcraft. In his flat you can find some sculptures **(Interest: Sculpture)** which were produced by his friend. Every time when he receives guests, he is asked about his impressive art objects. And Lukas is very proud when he can tell some stories about their origin. But you will not find only sculptures in his flat. He is a big fan of the german visual artist "Gerhard Richter" **(Interest: GerhardRichter)** and owns a personally signed photorealistic painted work **(Interest: Painting *createdBy* GerhardRichter).** There is a film about this artist called "Gerhard Richter's Window" and another one called "Gerhard Richter's painting". Gerhard Richter is a very media-shy artist and Lukas is highly interested in each video content where he appears or someone talks about him.

When Lukas is currently not located on the golf course during his leisure time, you can find him three times a week in an indoor swimming pool **(Interest: shortDistanceSwimming)**. Mostly he swims with a colleague with whom he battles a small swimming challenge to increase his fitness level. In his youth he was member in a swimming team fighting for competitions in short distance swimming. And to this day he watches any competition in that swimming discipline. It does not matter whether it is a local swimming event **(Interest: sportEvent *dealsWith* shortDistanceSwimming)** or Olympic games **(Interest: OlympicGames)**.

At the weekend **(Context info: at the weekend)** Lukas often goes to music concerts located in Germany **(Interest: Concert *locatedAt* Germany *dealsWith* Blues, Concert *locatedAt* Germany *dealsWith* Jazz)**. He has seen a lot of internationally known music bands but has no specific favorite musician. He is generally interested in each kind of Jazz **(Interest: Jazz)** and Blues **(Interest: Blues)** music. But he mustn't inevitably go out for experience music events. He is also satisfied in watching a music-live-stream while sitting on the couch at Friday or Saturday evening **(Context info: Friday evening, Saturday evening)**. Sometimes he spends a whole Sunday afternoon **(Context info: Sunday afternoon)** with watching concert-records.

Modeling Lukas

In Lukas' user model you can find seven selectable contexts from which he can choose as needed.

Figure 19: LUME Screenshot about Lukas context directory

The default context shows all concepts that are relevant at any time. The weightings are randomly assigned. In the context overview you may miss some context information that is highlighted within the user model description but not added to the LUME user model of Lukas. We have consolidated the context information that have a big overlap. For example, "Friday evening", "Saturday evening" and "Sunday afternoon" are summarized as "at the weekend". The context "RBB Aktuell" is a little bit special, because here we want to model a specific context that in which Lukas only wants to see news videos created by this broadcaster. At the moment the context name is only a description and has no impact on the semantic filtering process, but in future work we want to take into account that a user may watch a specific broadcaster.

Peter

Peter works as an electronic engineer **(Interest: electronic engineering)** in a large automotive group in Germany. Besides the development of new technical components he is responsible for the analysis of innovations of business competitions. So, he often watches TV shows where new cars **(Interest: Car)** are introduced. To stay on the current state of the art, he regularly organizes work time at home **(Context info: work time, at home)** to check automotive magazines and advertisement spots **(Interest: News *about* Car)** of the competitors. This is very important for his career, because back at work Peter can suggest creating new products that may be assembled in the next generation of cars. Even in design issues, he can often give worth full hints. His fascination for cars is extremely pronounced, so he often visits car exhibitions **(Interest: Exhibition *dealsWith* car)** with his camera. His handling of photographing **(Interest: photography)** equipment is very professional. Sometimes his photos are published in magazines. This Hobby supports him greatly in his daily work.

Since years, Peter plays in a local football team in Berlin **(Interest: football *restrictedTo* Berlin)**. They are not very successful but it is a good compensation with respect to his work-life-balance. Additionally to that, he has bought a bicycle **(Interest: Cycling)** some month ago. At least one time per week he drives around with his colleagues.

Peter's favorite national football teams are "FC Bayern Munich" and "FC Energie Cottbus". If a football match of one of these teams will be broadcasted on TV he wants to see it **(Interest: sportEvent *involves* FC Bayern Munich, sportEvent *involves* FC Energie Cottbus)**. But sometimes he doesn't have enough time and in this case he programs his video recorder to record a match for later watching in the evening. At the weekend he organizes football events with his friends. He invites them to his home for a specific time and the only thing that matters in this moment is football. They all watch interviews, pre-reports **(Interest: sportEvent *dealsWith* football)** and similar shows. Peter's favorite interview-guy is "Franz Beckenbauer" **(Interest: Franz Beckenbauer)**.

On his vacation **(Context info: on vacation)** Peter often goes fishing **(Interest: Fishing)**. He enjoys the repose while residing in the nature. Here he can relax very well. After a long outdoors stay he likes to relax his mind with a good movie. Interestingly the best way to do this is via watching an action movie **(Interest: actionMovie)** or crime movie **(Interest: crimeMovie)**. The best bet is to watch a movie with his favorite actors "Brad Pitt" or "Angelina Jolie" **(Interest: actionMovie *about* Angelina Jolie, actionMovie *about* Brad Pitt).** But this kind of movies he looks not only at the holidays. If an appropriate movie is announced on TV he would like to see the movie as well except when there are new football contributions available. Football is his highest priority.

Modeling Peter

In Peter's user model you can also see seven different contexts.

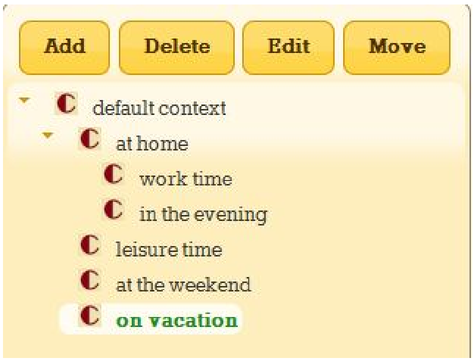
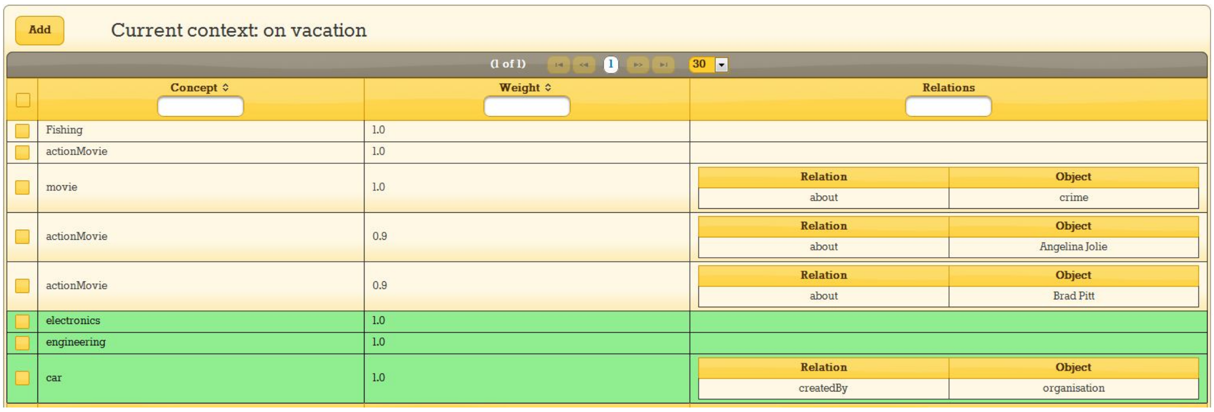


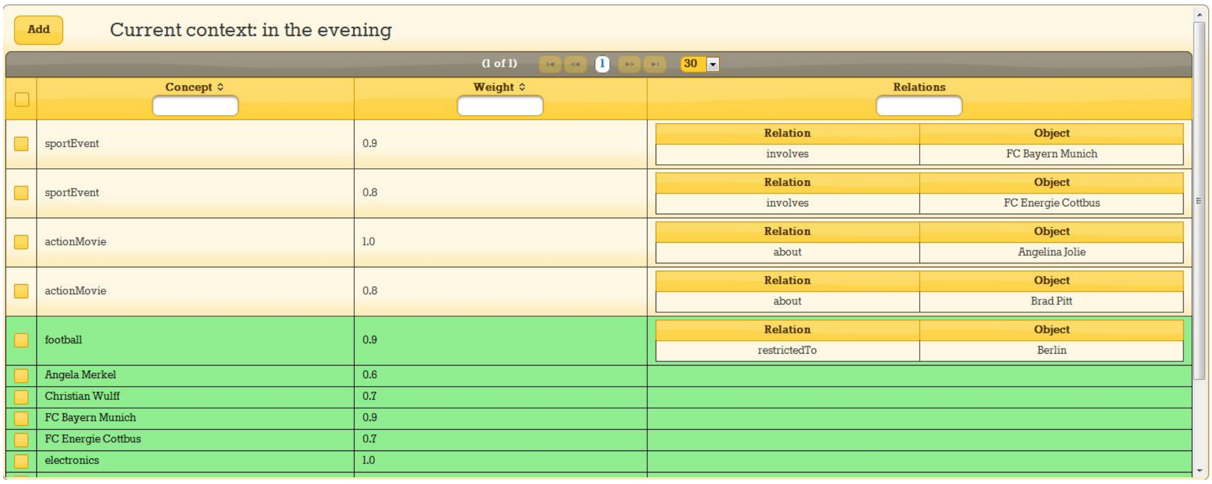
Figure 20: LUME Screenshot about Peter's context directory

In the contexts "in the evening" and "on vacation" you can see double entries. This is caused by the inheritance rules of the entire context directory. We cannot model concepts in an upper context when the concepts are only relevant for a specific context and not for the upper context. So we modeled "actionMovie" in both contexts.



Concept	Weight	Relation	Object
Fishing	1.0		
actionMovie	1.0		
movie	1.0	about	crime
actionMovie	0.9	about	Angelina Jolie
actionMovie	0.9	about	Brad Pitt
electronics	1.0		
engineering	1.0		
car	1.0	createdBy	organisation

Figure 21: LUME Screenshot about Peters preferences in his "on vacation" context



Concept	Weight	Relation	Object
sportEvent	0.9	involves	FC Bayern Munich
sportEvent	0.8	involves	FC Energie Cottbus
actionMovie	1.0	about	Angelina Jolie
actionMovie	0.8	about	Brad Pitt
football	0.9	restrictedTo	Berlin
Angela Merkel	0.6		
Christian Wulff	0.7		
FC Bayern Munich	0.9		
FC Energie Cottbus	0.7		
electronics	1.0		

Figure 22: LUME Screenshot about Peters preferences in his "in the evening" context

6.2 LUME

LUME (LinkedTV User Model Editor) aims to provide an intuitive user interface for the end users of LinkedTV to manage their user models. It is implemented as a web application and provides RESTful web services for further integration with other LinkedTV components, in particular with the LinkedTV video player from WP3.

6.2.1 User model

A user model contains a set of user interests. A user interest consists of an IRI[29] and a weight. The value of Weigth is a real number between 0 and 1. It can be -1 as well denoting that user X dislikes the given IRI.

6.2.2 Constraints of user interest

Each user interest can be associated with a set of constraints like 'politician **comes from Germany**'.

Each constraint contains a relation and a value (see for example Figure 22).

Adding constraints to an interest is optional. It however greatly increases the expressive power of the modelling approach. The price for that is the higher complexity of the semantic graph matching algorithm which is adopted in the LinkedTV semantic Filtering (LSF) component (cf. deliverable D4.5).

6.2.3 Contextualized user model

Users can explicitly define the contexts[30] they want to use to structure their interests (see Chapter 6.1). Each context consists of a set of user interests. As depicted in Figure 23, the blue context contains three user interests {I1,I2,I3}, the orange context contains two user interests {I2,I3}, and the green context contains two user interests {I3,I4}. On the other side, one user interest can exist in different contexts. In Figure 23, user interest I2 is contained in both orange and blue contexts while I4 is only in the green context.

With LUME users can manage (create, modify and delete) their contexts as well as the user interests defined in those contexts in an intuitive way.

[29] We use IRI (Internationalized Resource Identifier) instead of URI (Uniform resource identifier) in this document.

[30] In Year-3 we will investigate how the explicit contextualization approach described in this chapter can be brought together with the observation oriented approach of Chapter 5.

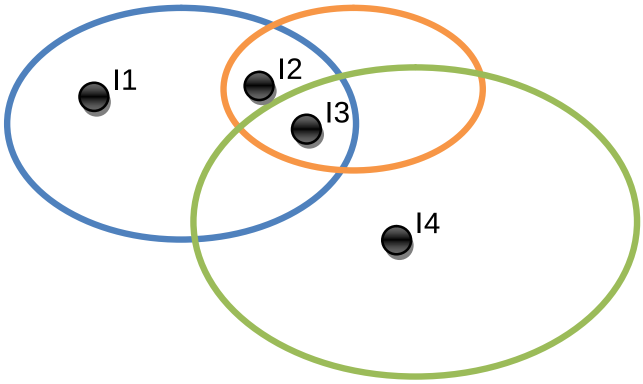


Figure 23: Contextualised user models

6.2.3.1 Context inheritance

In LUME contexts are intrinsically inheritable. If a context CTX2 is inherited from CTX1, then all user interests defined in CTX1 are available in CTX2. For example, in Figure 24, an inheritance with three layers is depicted: red context inherits orange context, green context inherits red context. Hence, the user interests with inheritance in three contexts are:

- Orange context contains only I1
- Red context contains I2 and I1 which is inherited from the orange context
- Green context contains all three interests: I3 and I1 which is inherited from the orange context and I2 which is inherited from the red context

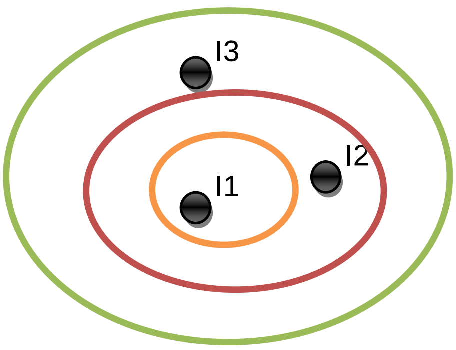


Figure 24: Illustration of context inheritance

6.2.3.2 Model override in inherited contexts

In LUME, inherited user interests can be overridden by defining the same interest in the sub-context. This provides additional modeling flexibility for complex scenarios.

6.2.4 Architecture of LUME

LUME is implemented as a client-server web application. It is developed with HTML5, mobile devices and scalability in mind. The application runs on most major browsers on top of various desktop and mobile platforms.

A database backend is integrated for efficient and scalable data persistence and retrieval (see Figure 25). The main application logic is developed as a Java enterprise application consisting of several EJB modules and JSF web modules. RESTful web services are provided based on JAX-RS with Jersey implementations.

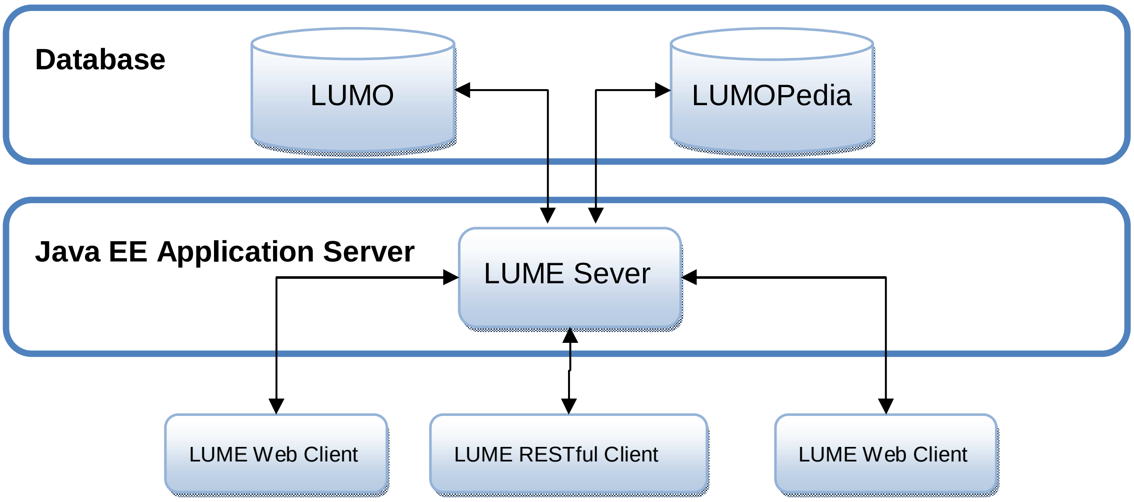


Figure 25: The architecture of LUME application

6.2.5 Registration and login authentication

LUME will be integrated into the LinkedTV video player later. The registration and login authentication will be done there. At the moment, since the integration work is still not finished, we provide a login interface (see Figure 26) to use LUME. A registration interface is not provided, but registered users in the older LUME versions can still login with their user name and password. For new users, a request can be sent via Email to the Fraunhofer team to get the access information.

Figure 26: LUME login interface

If the login is failed, detailed error information is displayed below the login form (see Figure 27). Based on the feedbacks, users can correct the login information accordingly and try again.

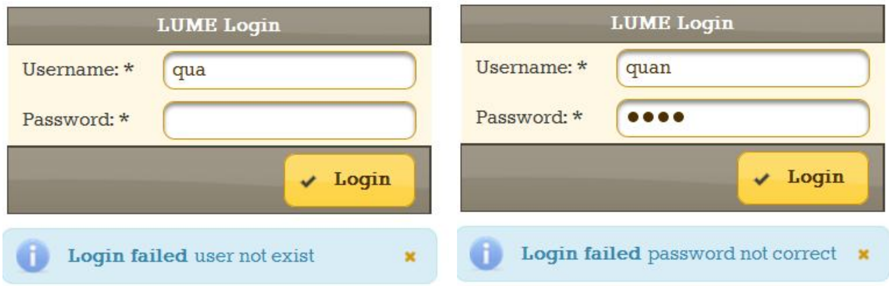


Figure 27: Detailed error information for invalid logins

6.2.6 Context management

After the user is successfully logged in, the main view of the user model management is displayed as depicted in Figure 28.

On the top right of the view, the current user name is displayed indicating who is currently logged in (see the red rectangle in Figure 28). Next to the current user name, a logout button (see the green rectangle in Figure 28) is provided to invalidate the current session with the LUME server.

The left side of the main view contains the context hierarchy tree with four buttons “Add”, “Delete”, “Edit” and “Move” (see the blue rectangle in Figure 28). The context hierarchy tree visualised the context inheritance mechanism introduced in Section 6.2.3.1. The four buttons provides functionalities to manage contexts:

- Add – add a sub context under the currently selected context
- Delete – delete the currently selected context
- Edit – change the name of the currently selected context. All contexts of a user share the same namespace, hence no duplications are allowed. The user interface will check the uniqueness before it is committed.
- Move – change the father context of the selected context

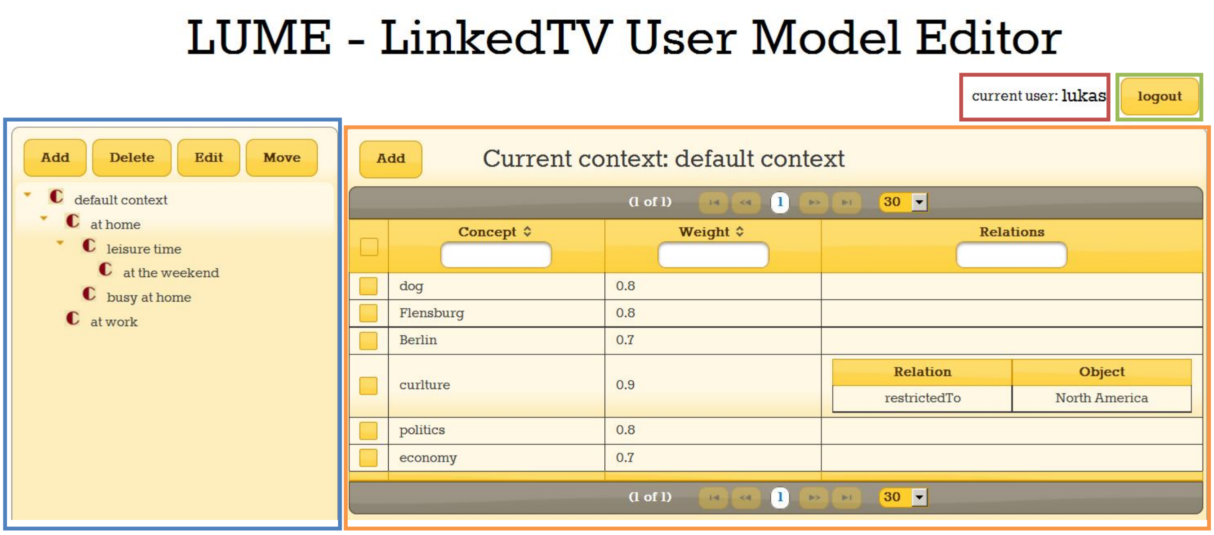


Figure 28: The main view of the user model management

After these buttons are clicked, different dialogs will appear if further information is needed (see Figure 29).

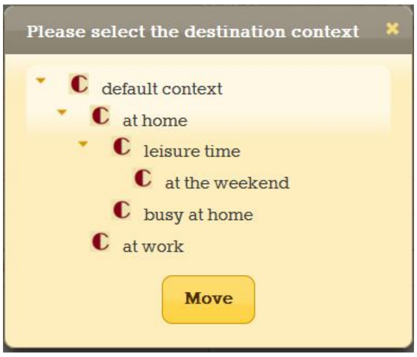


Figure 29: Further information is needed for managing contexts

6.2.7 User interest management

On the right side of the main view is the list of user interests (see the orange rectangle in Figure 28). This part contains a full-featured table with all user interests under the selected context.

6.2.7.1 Paging, filtering, sorting user interests

For contexts containing large number of user interests, it is very useful and comfortable to provide paging, filtering and sorting functionalities over the whole interest spaces.

- Paging – the paging toolbar is positioned on top of the interest table (see the red rectangle in Figure 30). Users can navigate between pages or change the number of interests displayed in a page.
- Filtering – the filtering functionality is provided for each column in the interest table (see the green rectangle in Figure 30).The filtering pattern based on case insensitive sub-strings and no regular expression is supported. For example, the filter term ‘iO’ will match both ‘religion’ and ‘union’. For the relations column, both relation names and object names will be used: either the relation or the object name is matched it will be included.
- Sorting – to sort the table based on a column, just click the up and down arrows next to the column name. It will switch between ascending and descending sort.

If a new context is selected, all these information will be reset.

Figure 30: Paging, filtering, sorting user interests

6.2.7.2 Highlight the user interests of specific types

In the current version two specific types of user interests are highlighted. The first one is the user interest inherited from a more general context. The inherited user interests are highlighted with green background (see the two user interest ‘European Union’ and ‘economy’ in Figure 30).

The second one is the ‘dislike’ interest. The font size is different from the normal ones and the color is red. For example, in Figure 31, the interest ‘celebrity’ is both ‘inherited’ (with green background) and ‘dislike’ (with slightly larger and red font).

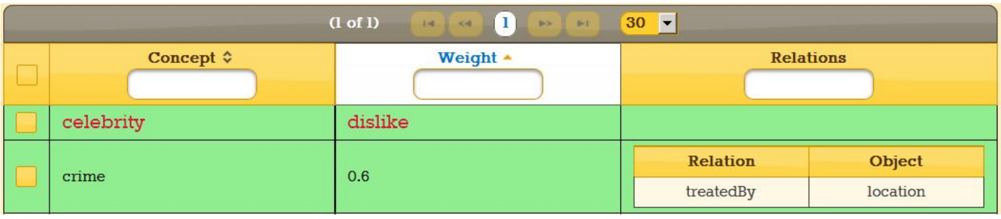


Figure 31: Highlight the user interests of specific types

6.2.7.3 Context sensitive toolbar

Multiple selections are supported. Based on the selected user interests, the toolbar of available commands for managing user interests differs (see the blue rectangle in Figure 32). There are totally five buttons available and they are:

- Add – add a user interest. This is always available and is independent of the selections (more about how to add an user interest, please see Section 6.2.8).
- Delete – delete a set of user interests from the active context. This is only available when the selected user interests do not contain inherited ones.
- Edit – edit a set of user interests from the active context. This is only available when the selected user interests do not contain inherited ones.
- Copy – copy a set of user interests from the active context and paste them to the target context. This is only available when at least one user interest is selected, no matter whether it is inherited or not.

- Move – move a set of user interests from the active context to the target context. This is only available when all of the selections are not inherited.

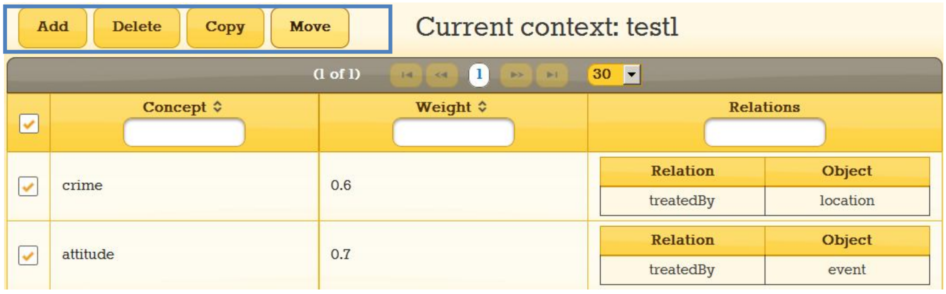


Figure 32: Context sensitive toolbar

6.2.8 Add an user interest

To add a new user interest, click the “Add” button in the toolbar as depicted in Figure 32. The add interest view is displayed as in Figure 33.

Figure 33: Search and add interest view

On the top of this view, the current active context and the current user plus the logout button are displayed (see the blue rectangle in Figure 33). Below that, a “Cancel and go back” button is provided if the user want to cancel the current operation and go back to the main view.

To add an interest, the user has three options:

- Search in LUMO-rel and LUMOPedia
- Browse the LUMO-rel ontology
- Browse the LUMO-V1 ontology

Searching in the LUMO-rel and LUMOPedia is facilitated with auto-completion. As the user types some texts, LUME will try to suggest some possible candidate entities in LUMO-rel and LUMOPedia (see Figure 34).

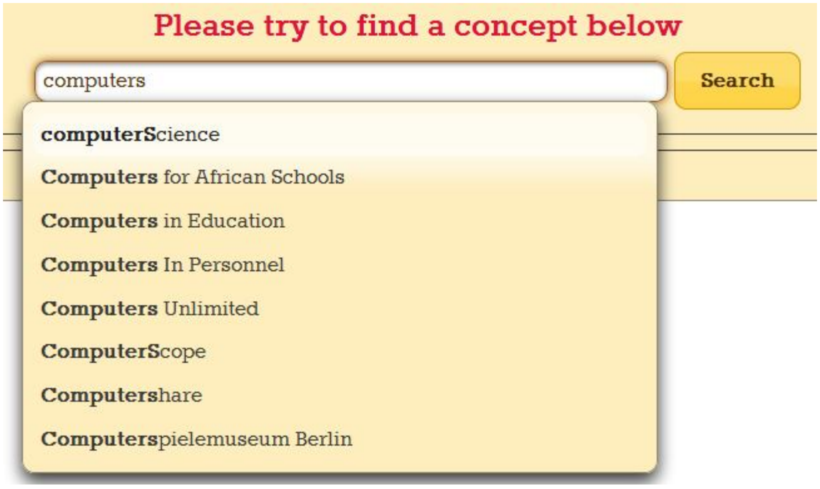


Figure 34: Auto-complete for searching LUMOPedia

After a specific human-readable name is selected from the auto-completion list, LUME will search in the knowledge base and list all semantic entities which are "matched" with this human-readable text. Since a human-readable text can be given to more than one semantic entities like "program" can be "computer program" or "TV programs", the result list can contain more than one result, though normally it is one-to-one relationship.

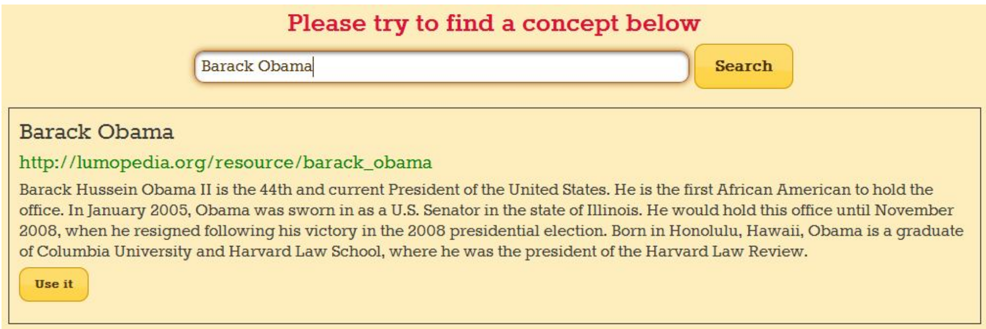


Figure 35: Search result based on human-readable texts

Now the user can press 'Use it' and give some weight of this interest in the user interfaces (Figure 36). If the chosen entity is an instance in LUMOPedia, then no constraints are allowed to be added. On the other side, if it is from LUMO-rel, the "Add Relation" button will appear and the user can continue to add constraints if he / she likes.

Figure 36: Wegiht the user interest

It is also possible to browse the ontology to select the concept. Both LUMO-rel and LUMO-V1 are integrated in LUME, the user can use any of them to add concepts (see Figure 37).

Figure 37: Browsing the LUMO-Rel and LUMO-V1 ontologies

6.2.9 Add constraints

After the user has chosen his/her main interest the user can click the “Add Relation” button to add constraints (see Figure 38).

Figure 38: Add constraints for a given interest

The available relations for a given interest are defined in the LUMO-rel ontology. Auto-completion is available here and the user can select one relation from the dropdown list.

Now the user can click “Add” button in the object column to choose the value of this relation. The same search view (see Figure 35) will be displayed again and the user can search one entity or browsing the ontologies to select one.

More constraints can be added and the logic relation between these constraints is conjunctions, i.e. all of these constraints must be valid.

6.2.10 RESTful web services

In order to support the inter-operability between software components, and in particular with the LinkedTV platform and the other personalisation and contextualisation services, LUME provides RESTful web services to access the contexts and user models.

Currently for security reasons, services for the modification (create, edit, and delete) of any resources are not provided. We are still testing our modification services, so that other partners can for example modify the user model based on their user observations and preference learning.

In order to use the web services, the user must apply a key from Fraunhofer at first. After you get the key, set this key in the HTTP header as the value of "`X-Authentication`". Since RESTful is stateless, i.e. not session-based, you must attach this key-value pair in each HTTP request header.

The result of the web services can be formatted in XML or JSON. The default one is XML. To change the format produced by the web services, please specify another key-value pair in the HTTP header. To request the result as JSON, use the following key-value pair in HTTP header ‘`Accept : application/json`’; To request the result as XML, use the following key-value pair in HTTP header ‘`Accept : application/xml`’ or just remove this key-value pair in the header since XML is the default format.

The following tables give more details about the RESTful web services:

GET all users
Retrieve all users.
Description: GET /service/user
HTTP method: GET
Content-Type: XML, JSON
Authentication: You need a key from Fraunhofer
Example of response:
<users><user><model/><userName>quan</userName></user></users> `{"user":{"model":"","userName":"quan"}}`

GET a user
Retrieve a specific user.
Description: GET / service/user/{username}
HTTP method: GET
Content-Type: XML, JSON
Authentication: You need a key from Fraunhofer

Example of response:
{"contextTree":{"root_context":{"child_contexts…model":"","userName":"quan"} <user><contextTree><root_context><child_contexts><context> …<model/><userName>quan</userName></user>

GET a user model
Retrieve the user model for a specific user under a context.
Description: GET / service/model/{username}/{contextName}
HTTP method: GET
Content-Type: XML, JSON
Authentication: You need a key from Fraunhofer
Example of response:
<userModelEn- tries><userModelEntry><concept><iri><id>http://www.linkedtv.eu/lumo#person</id></i ri...<label>Applied_sciences</label><relations/></concept><user_model_entry_weight >0.9</user_model_entry_weight></userModelEntry></userModelEntries> {"userModelEn- try":[{"concept":{"iri":{"id":"http://www.linkedtv.eu/lumo#person"},"label":"perso n","relations":{"relation":{"label":"authorOf","concept":{"iri":{"id":"http://www. linkedtv.eu/lumo#object"},"label":"object","relations":null}}}},"user_model_entry_ weight":"0.9"},{"concept":{"iri":{"id":"http://data.linkedtv.eu/ontologies/lumo#Ap plied_sciences"},"label":"Applied_sciences","relations":null},"user_model_entry_we ight":"0.9"}]}

7 Conclusions & Future Work

This deliverable has presented a concrete implementation in implicitly constructing contextualised user profiles based on the user's interaction with the LinkedTV platform and concrete tools that enable a user to explicitly declare and update his/her preferences on the LinkedTV platform.

For both these processes, the specific available subservices that constitute the end-to-end workflow were described: from the reference knowledge which serves as the semantic vocabulary and concept space of the user-centric world in the networked media domain, to the reception and management of user input, whether implicitly captured or explicitly expressed, to the learning and construction of user models and finally to their adaptation to the user context.

Future work will orient towards aggregating the two approaches under a common workflow, starting from merging the most suitable features in the two reference knowledge bases used, namely LUMO v1 and LUMO-rel, under one common, lightweight and expressive knowledge base. Following that, optimization of the cooperation of the tools within the profile construction workflow will be pursued, with a focus on the implementation and communication of the context extraction and modeling components in conjuction with the long-term user profiles. This process involves also the optimization of communicating strategies among WP4 tools and a complete integration into the LinkedTV platform and player.

In addition, the presentation of implicit user profiles in conjunction with the explicit user profiles directly on the player for user awareness and evaluation will be conducted, while the validation of constructed user models will be sought after, based on a synthetic evaluation as a first step and subsequently based on dedicated user trials. In effect, user models will be evaluated both as standalone data models based on their acceptance by the users whose preferences they represent and in connection with the quality of the content and concept filtering services they accommodate.

The tools and services illustrated in this document comprise the direct input for the concept and content filtering tools which consummate the personalization and contextualization task of LinkedTV. Deliverable D4.5 follows this deliverable and describes how produced user models are employed to filter seed and enrichment content available on the LinkedTV platform, based on its semantic annotation, as well as concepts within the LinkedTV user-pertinent concept space.

8 Bibliography

[APH04] T. Ahonen, A. Hadid, and M. Pietikainen, "*Face Recognition with Local Binary Patterns".* In Proc. of the Europeen Conference of Computer Vision (ECCV), 469–481, 2004

[AUE07] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. "DBPedia: a nucleus for a web of open data". Proc. 6th international The semantic web (ISWC'07/ASWC'07), 2007, Springer-Verlag, Berlin, Heidelberg, pp. 722-735.

[BBC] Corenews Ontology: http://www.bbc.co.uk/ontologies/news/ (last visit 7th August, 2013).

[BEL97] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE PAMI), 19(7):711--720, 1997.

[BRE01] L. Breiman. Random forests. Machine Learning, 45(1):5{32, 2001

[FAN11] G. Fanelli, T. Weise, J. Gall, and L. V. Gool. Real time head pose estimation from consumer depth cameras. In Proceedings of the 33rd international conference on Pattern recognition, DAGM'11, pages 101{110, Berlin, Heidelberg, 2011. Springer-Verlag.

[FAN13] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool. Random forests for real time 3d face analysis. International Journal of Computer Vision, 101:437-458, 2013.

[HOF13] J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence, Volume 194, January 2013, Pages 28–61 http://dx.doi.org/10.1016/j.artint.2012.06.001

[GRO03] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. 2003. Description logic programs: combining logic programs with description logic. In Proceedings of the 12th international conference on World Wide Web (WWW '03).

[JIM11] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau: LogMap: Logic-Based and Scalable Ontology Matching. International Semantic Web Conference (1) 2011: 273-288

[KIN10] Microsoft. Kinect sensor. http://www.xbox.com/kinect

[LER13] J. Leroy, F. Rocca, M. Mancas, and B. Gosselin. "Second screen interaction: an approach to infer tv watcher's interest using 3d head pose estimation". In Proceedings of the 22nd international conference on World Wide Web companion, WWW '13 Companion, pages 465-468, Switzerland, 2013

[NERD] Giuseppe Rizzo, Raphael Troncy: NERD - A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools, in: 13th Conf. of the European Chapter of the Association for Computational Linguistics (EACL'12), Demo Session, Avignon, France, April 23-27, 2012.

[ONI10] OpenNI. Open natural interfaces. http://www.openni.org/.

[PCL11] PCL. Point cloud library. http://pointclouds.org/.

[SKRA12] Radek Škrabal, Milan Šimůnek, Stanislav Vojíř, Andrej Hazucha,Tomáš Marek, David Chudán, Tomáš Kliegr. Association Rule Mining Following the Web Search Paradigm. In Proc. of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2012), Bristol, UK, 24-28 September 2012. Springer- Verlag

[StSt04] Steffen Staab, Rudi Studer (Eds.): Handbook on Ontologies. Springer 2004 International Handbooks on Information Systems ISBN 3-540-40834-7.

[TUR91] M. Turk and A. Pentland. "Face recognition using eigenfaces". Proc. Of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 586–591, 1991

[WEB10] Websocket community website: http://www.websocket.org/