Deliverable 4.2 User profile schema and profile capturing

Dorothea Tsatsou, Vasileios Mezaris (CERTH)

Tomáš Kliegr, Jaroslav Kuchař (UEP)

Matei Mancas (UMONS)

Lyndon Nixon (STI GMBH)

Rüdiger Klein, Manuel Kober (FRAUNHOFER)

18.10.2012

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 2012
Actual date of delivery	18th October 2012
Deliverable number	D4.2
Deliverable name	*User profile schema and profile capturing*
File	*LinkedTV_D4.2.*doc
Nature	*Report*
Status & version	Final, v1.0
Number of pages	*88*
WP contributing to the deliverable	*WP 4*
Task responsible	*CERTH*
Other contributors	*UEP* *UMONS* *STI GMBH*
Author(s)	*Dorothea Tsatsou, Vasileios Mezaris* *CERTH* *Tomáš Kliegr, Jaroslav Kuchar* *UEP* *Matei Mancas* *UMONS* *Lyndon Nixon* *STI GMBH*

[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))

	Rüdiger Klein, Manuel Kober *FRAUNHOFER*
Reviewer	*Lyndon Nixon, STI*
EC Project Officer	*Thomas Kuepper*
Keywords	*Ontology, user model, transaction tracking, behavioural tracking, preference learning*
Abstract (for dissemination)	*This deliverable presents methods employed in LinkedTV to create, update and formalise a semantic user model that will be used for concept and content filtering. It focuses on the extraction of lightweight and dense implicit knowledge about user preferences. This process includes the semantic interpretation of information that stem from the user's interaction with the content, together with the estimation of the impact that preferred concepts have for each specific interaction based on the type of transaction and the user's physical reaction to the content. User preferences are then updated based on their age, frequency of appearance and utility, while persistent associations between preferences are learnt. This information evolves to a semantic user model that is made available for predictive inference about relevant concepts and content.*

Table of contents

List of Figures

List of Tables

1 Introduction

This deliverable describes the processes currently being developed within the LinkedTV personalisation task for capturing, manipulating, updating and serializing a user profile with respect to its use for concept and content filtering/recommendation.

The user modeling approach in LinkedTV is oriented towards creating a semantic user profile, based on a formal ontology. The task focuses on implicitly learning user preferences, based on the semantic description of (seed and enriched) content, following the interpretation of raw data in WPs 1 and 2, while factoring in the explicit declaration of semantic preferences and rectification of the learned profile (in collaboration with WP3). The implicit profiling mechanism involves producing a user model from cold start and evolving it over time based on the age, the history of user transactions (what video the user watches, what media fragments s/he looks at, what concepts s/he chooses, what additional content s/he browses), and user reactional behaviour (engagement and attention to viewed/consumed content).

This involves unobtrusively extracting and understanding implicit user feedback and translating it in a machine-understandable format appropriate for making predictive inference about relevant concepts and content. The translation will attempt to bridge disparate information from different vocabularies used to semantically annotate content (WP2) to a more lightweight and meaningful (to the user) knowledge base in the interest of alleviating processing and storage load. Minimizing this load is expected to minimize the need for server-client communication, thus venturing towards better safeguarding user privacy.

There are several issues that need to be dealt with in the context of implicitly capturing and representing a semantic user profile: the ontology that can provide a meaningful, uniform and lightweight reference knowledge base potent to capture domain and user-pertinent semantics; the means to align this uniform knowledge base with the semantic information in the multimedia content; the means to unobtrusively capture the user’s transactional and reactional behaviour; the means to understand user behaviour, i.e. map it to available knowledge and determine its impact; determining the most suitable representation schema of the user model in a manner that renders the synergy between the model and the background knowledge feasible within the context of several inferencing mechanisms used for producing recommendations; and finally address security and privacy issues that arise from the use of personal user information. The user modelling workflow is illustrated in Figure 1.

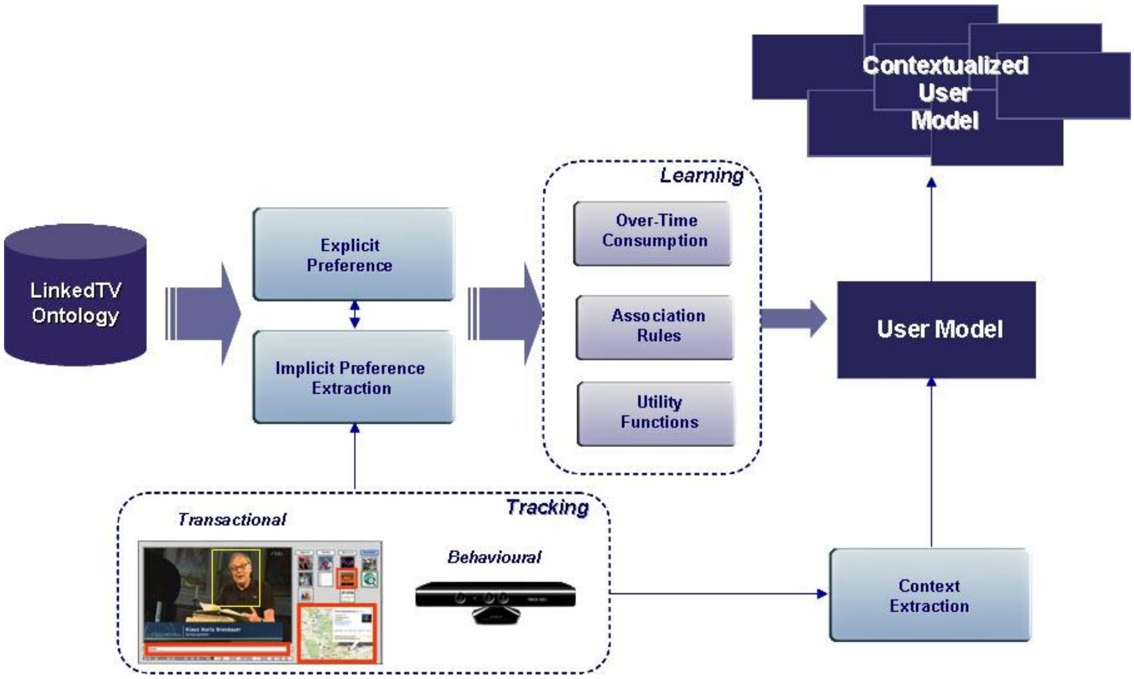

Figure 1: The user modelling workflow

Although contextualising user preferences is an imperative functionality of WP4, this deliverable will focus on the extraction of the long-tem user model, to be adapted based on the contextual situation of the user in latter stages of development. Nevertheless the methodology principles for adapting the long-term user model to contextualised user model instances have already been defined in D4.1.

Since the personalisation task will concentrate on the non-trivial task of extracting user implicit preferences, the substask of implicit preference extraction encompasses several distinct mechanisms, portrayed in Figure 2. Explicit preferences, defined by the user on the platform, are considered aligned to the semantic representation proposed (i.e. from the same ontology), thus following the same update and learning mechanisms as implicit preferences.

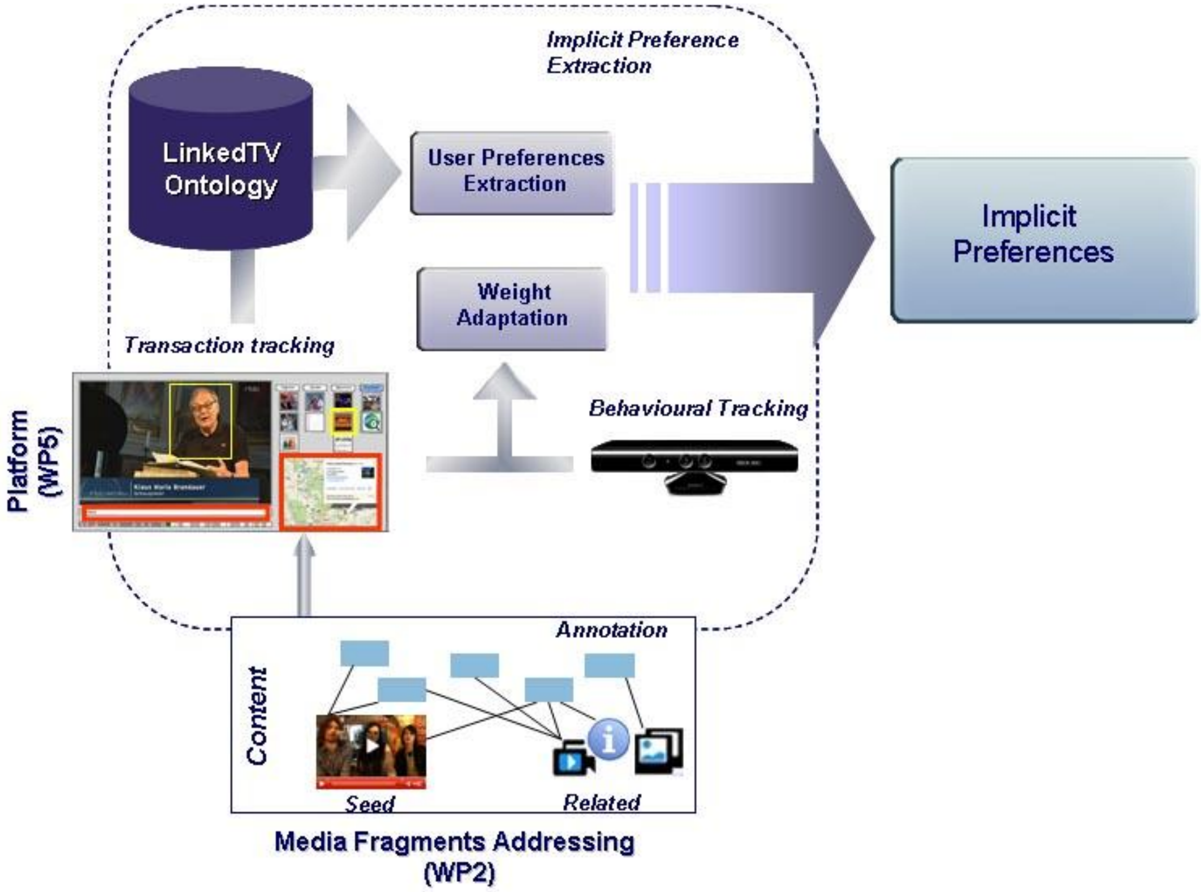

Figure 2: Extracting implicit preferences

1.1 History of the document

Table 1: History of the document

Date	Version	Name	Comment
2012/07/30	V0.01	Dorothea Tsatsou	Suggested ToC
2012/09/10	V0.10	Dorothea Tsatsou	First version, includes invitations for further contributions - requires modifications and sections to be completed
2012/09/19	V0.15	Matei Mancas	Additions to behavioural tracking
2012/09/19	V0.15	Tomas Kiegr, Jaroslav Kuchar	Transaction tracking, association rules, preference extraction expanded
2012/09/25	V0.15	Lyndon Nixon	Added section 3.4
2012/10/03	V0.16	Matei Mancas	Addressing comments of V0.15

Date	Version	Name	Comment
2012/10/04	V0.16	Tomas Kiegr, Jaroslav Ku-char	Addressing comments of V0.15
2012/10/10	V0.20	Dorothea Tsatsou	Version for QA
2012/10/18	V0.30	Dorothea Tsatsou	Final post-QA version
2012/10/18	V1.0	Martha Merz-bach	Final Layout changes

2 Ontology: addressing the world based on user requirements

Advanced personalisation in networked media platforms, such as LinkedTV, is required to efficiently handle and take advantage of the information stemming from the users' digital traces by unobtrusively capturing, understanding and unifying the disparate information originating from the users' transactional behaviour and interaction with peers.

Therefore, it is significant for the efficient elicitation of user preferences to have a holistic and dense (thus somewhat lightweight) vocabulary to classify this information under. We may assume an ontology comprising the shared semantic model of the domain in question, thus the reference knowledge base (KB) for the user model which will contain all relevant domain and/or user-specific concepts and their relationships and providing uniform, compact conceptualizations for ambiguous, synonymous and multilingual knowledge. Such an ontology can be used as the backbone for predictive inferencing of user preferences, as well as for profile-content matching to produce targeted content recommendations.

2.1 Decision and justification of used ontology

A core ontology aiming to adequately describe knowledge relevant for a user in a heterogeneous hypermedia environment is expected to be rather broad. It needs to cover anything from the high level topic conceptualizations and the vastness of the named entities encompassed across various domains, to dedicated entities and relations pertaining a user's context, emotional and physical situation. On the other hand, efficient handling of this immensity of information requires dense and slim conceptualisations in a highly expressive, formal ontology for it to scale well and maintain the accuracy advantage of logical inference algorithms.

This section will cover the reasoning behind the decision of the used ontology and the justification of its requirements and design principles, a work presented in [TSA12].

2.1.1 LOD Vocabularies

The Linked Open Data[2] (LOD) initiative attempts to provide structure to the vast mass of information available online. Most current personalisation approaches for networked media environments have been directed towards employing such open linked vocabularies to efficiently describe and expand the diverse and continuously evolving information in digital media content and in extension reflect and address the variety in user preferences.

[2] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

To this end, WP2 will semantically describe digital content based on a set of predefined LOD vocabularies (cf. D2.2), suitable to represent the information in multimedia content. These semantic descriptions are the primary resource of information that WP4 has in order to understand the users' preferences based on their interaction with the seed video content. In addition, the interaction with additional content related to the media fragments (cf. D2.3) will comprise the second source of seed information about the user's implicit preferences.

A comprehensive overview of LOD vocabularies that are useful to semantically describe and interlink the upper domain of networked media has already been conducted in D4.1. While the list of general-purpose semantic LOD vocabularies of more or less the same expressivity that can encompass knowledge useful for interpreting information in consumed content is non-exhaustive, we sum up some prominent relevant vocabularies, such as schema.org[3] and DBPedia [AUE07], which stands out as the most prominent organization and concentration of knowledge in the current literature on LOD datasets. DBPedia has been currently used in many cases as a reference vocabulary for personalisation, like for instance in the NoTube[4] project's approach for personalised TV content recommendation, which involved building a user profile in which interests were identified using categories from the DBPedia concept space.

Freebase [BOL08] is a public collection of community-contributed interlinked data, or as the community itself describes it "an entity graph of people, places and things". The Freebase ontologies are user-generated and edited, consisting of semi-structured information in the form of folksonomies. It was recently employed by the Google Knowledge Graph [GKG12] to expand Google search results about such entities with related information.

YAGO [SUC07] unifies WordNet [MIL95] with Wikipedia, thus enhancing the semantic relations between entities and individuals of Wikipedia with more descriptive properties. It additionally offers correspondences between the entities and their lexical description (term) while taking into account synonymy and term ambiguity, thus allowing for advanced content classification. However, it is easily understandable that while such a vocabulary adds to the semantics of Wikipedia information, it also adds to the complexity of Wikipedia-based knowledge.

Refraining for enumerating them, we must also mention that the LOD cloud encompasses many interconnected datasets of domain specific knowledge, a collection of which can be found online[5], describing for instance detailed geographical information (Geonames), music relevant semantics (MusicBrainz), etc. These KBs may offer richer information or/and deeper semantics on many aspects important to represent a user's preferences and context, as well as mappings to more general upper knowledge bases and other ontologies in the cloud.

---

[3] http://schema.org

[4] http://notube.tv/

[5] http://www4.wiwiss.fu-berlin.de/lodcloud/ckan/validator/

2.1.1.1 Merits and drawbacks

The main advantage of employing LOD vocabularies to map user preferences lies on their wealth of information. In a vastly heterogeneous and broad environment such as networked media, the LOD datasets offer structure over the magnitude of data. This structure and information abundance is additionally augmented though the interconnectivity between different datasets within the LOD cloud. Moreover, the knowledge encompassed in LOD datasets is not static and does not require manual contribution from experts. Evolving knowledge is constantly updated, mostly through community contributed metadata. This process is further facilitated by the conformity of the knowledge bases to widely accepted and used standards (e.g. Dublin Core[6], SKOS[7], SIOC[8]).

So, are LOD vocabularies enough for intelligent personalisation and contextualisation? Arguably, no. The expressivity of LOD datasets is rather low. They provide structure and semantics to a large amount of entities in a shallow structure [DAM10]. For example DBPedia only deals with concepts, instances and generic relations connecting them. No complex information is available conveying the distinct axioms and specific relations between concepts that adequately describe the semantics prominent to a user or regarding the user's context across domains. Furthermore, the extremely high dimensionality of the reference knowledge raises serious management issues, especially with regard to safeguarding sensitive user information.

Lack of user-related semantics

The majority, if not entirety, of LOD vocabularies and ontologies comprise of shallow ontologies of class-property-instance triples consisting of hierarchical and non-hierarchical relations (at best) or frequently of mere taxonomies or meronymies. Thus, their expressivity and granularity is very low, which is only natural since they describe a vast amount of generic conceptualisations about the world that might concern different applications of different purposes. Consequently, there are no semantic descriptions in such ontologies for people-related facts that discern from general world perceptions. However a personalisation system should be able to comprehensibly understand how people (the users) generally perceive and interact with the world.

For example, let's assume the case where a user's preferences include football and a particular football team, let's say Manchester United - "and" here denoting the last item of a list, not a constructor indicating conjunction between items. It is perceivable that an inferencing engine that relies on LOD knowledge bases to match available content to a user's profile

[6] http://dublincore.org/

[7] http://www.w3.org/2004/02/skos/

[8] http://sioc-project.org/

would not derive any inconsistencies in recommending a video about the football team Chelsea, in which Manchester United does not appear, given a KB that relates the “Chelsea” instance to football. However, it is a shared conceptualisation that if a user is a fan of a particular team and not casually interested in the sport (that being determined by a preference learning mechanism) he would not be interested in recommendations about other teams-opponents, only because chances are that he would frequently view content containing the opponent team in relation to matches or events that his team was involved in. Conversely, an axiom in the reference ontology conveying such a general user-relevant condition (e.g. disjointness between all teams in the same national league) would permit a reasoner to detect the arising conflict.

Privacy and scalability

While the plethora of information in linked open data ontologies and vocabularies is most valuable for efficiently understanding content, they are still hampered by the immense volume of data, as argued in [JAI10a]. The problem for personalised services in particular is twofold: a) Recommendation services are server-bound even in resource-rich devices. The background knowledge required for making inference between the user profiles and available content is too large to be handled outside of a server at any instance. This renders constant client-server communication obligatory, thus giving rise to user privacy compromise problems. b) The volume of data and complexity in the KBs itself and the additional information stemming from mappings across other vocabularies in the Linked Data cloud can prove to be unmanageable for intelligent inferencing services, such as reasoners, to handle.

2.1.2 Upper formal ontologies

General upper or middle formal ontologies (e.g. SUMO[9], DOLCE[10], PROTON[11]) offer rich expressivity and can be used in support of middle or domain ontologies in describing the fundamental liaisons between the various cross-domain data encompassed in digital media. However, such ontologies are too generic (consisting of highly abstract conceptualisations) and voluminous to be used per se for meaningful description of user models or for supporting effective inferencing. They can however serve as the pillar for defining the semantics for a structured, formal ontology and most particularly as means to align different middle or domain-specific ontologies and other vocabularies under a knowledge base of an expressivity appropriate for modelling and handling user preferences.

In the field of broadcasting in particular, the RDF-based BBC Programmes Ontology[12] provides a descriptive vocabulary for TV programmes, describing concepts such as broadcasting events, brands, episodes etc. It is a lightweight ontology recording the broad spectrum of

[9] http://www.ontologyportal.org/
[10] http://www.loa.istc.cnr.it/DOLCE.html
[11] http://proton.semanticweb.org/
[12] http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml

rather abstract broadcasting-relevant aspects. It notably provides semantics for media-related temporal concepts and objects, thus rendering it a considerable basis for an upper level vocabulary in personalised TV environments. On top of that, the ontology is based on the FOAF vocabulary (described further on), thus providing associated mappings that allow for handling information in a user's social media activity.

For the purpose of personalised TV-content recommendations, an expressive OWL[13] ontology was developed within the AVATAR system [BLA05]. This ontology consisted of a full hierarchy based on three levels of granularity of program-related categories and subcategories used to classify TV programs. It also comprised the properties that interrelate them and different important entities within the context (actors, directors, places, scriptwriters, etc.) [BLA08]. Although the ontology appears suitable for re-using as a core for a TV-related personalisation platform based on its design principles and conceptual description, the ontology itself appears to be no longer available for properly analysing its semantics and value of use, and most importantly for reuse as such.

In addition to ontologies and vocabularies expressing knowledge over general world domains, conceptualisations and semantics pertinent to users of personalised platforms have also been proposed (e.g. skills, contextual situations, mood etc). Although such ontologies (cf. overview in D4.1) are in their world-related topics as generic and broad as upper ontologies, they do however include semantics especially useful to describe contextual conceptualisations and sensor extracted related information, i.e. the reactional behaviour of a user.

The most relevant to LinkedTV such user-specific ontology would be GUMO (General user model ontology) [HEC05]. GUMO and its descendants/hybrids record general upper concepts in combination with characteristic attributes of a user. The ontology is very general and broad, consisting of hierarchical/categorical relationships only. However, its user-related subsets efficiently depict top-level, user-relevant concepts such as user state and actions, e.g. personality (agreeable, timid etc), facial expression (happy, scared etc), motion (sitting, standing, walking), knowledge (e.g. computer skills), location (coordinates, spatial location, virtual location), social environment (friends, family etc). Such concepts are especially useful to semantically describe contextual semantics of sensor extracted information, i.e. the reactional behaviour of a user to the TV content. In addition it offers a categorization of interest topics that might serve as the basis of the upper hierarchy for modelling cross-domain interests.

2.1.3 Scenario specific ontologies

In the interest of addressing the specific scenario requirements of LinkedTV (cf. deliverable D6.1), a survey of news-specific and arts/culture-specific ontologies was conducted. However no formal ontology or categorisation was found to address the specific needs of the "An-

[13] http://en.wikipedia.org/wiki/Web_Ontology_Language

tiques Interactive" scenario. The nature of the scenario however, which relies a lot on specific objects (individuals) is expected to be adequately addressed by an upper arts and culture schema from an upper ontology, upon which more specific knowledge stemming from the LOD cloud might be mapped.

In the news domain, several upper but rather shallow ontologies/taxonomies exist. The IPTC[14] news categorization is widely used by news agents to classify news content under distinct categories. It is merely a shallow set of categories and subcategories of news subjects but it offers good coverage of the domain and can be used as a whole or in subsets to complement interest topics, such as the limited ones depicted in GUMO. The categories are available in a taxonomy produced by the WebTLab[15].

2.2 Aligning ontologies and LOD datasets

Although the domain of networked media is rather broad, relevant knowledge can still be limited under a consistent subset of information that mean something to the user. To achieve this task, two requirements arise: a) unifying relevant schemata under a core ontology of expressive formal semantics appropriate for comprehensive inferencing (these schemata encompassing knowledge about user characteristics and about the domains pertinent to digital media at an appropriate granularity) and b) unifying individuals across different vocabularies under a single conceptualisation and mapping them to the uniform schema.

The latter can be achieved through tools focused on aligning instances and concepts within LOD ontologies. The NERD ontology [RIZ11] used in WP2 (cf. D2.3) for instance provides a frame for mapping named entities (NEs) described across several multi-discipline vocabularies (Alchemy, DBpedia Spotlight, Extractiv, OpenCalais, Zemanta) on top of the NER [RIZ11] named entity extraction, classification and disambiguation tool. This ontology can be used for extracting NEs within textual manifestations of digital media (audio transcripts, articles etc) and supports semantic annotation of media content with coherent and interlinked instances belonging to popular LOD schemata, thus substantially enhancing semantic interpretation of diverse user-consumed content.

The former involves the non-exhaustive subject of ontology alignment and is mostly dependent on discovering relations from the lexical manifestations of ontological concepts. Some prominent approaches include AROMA [DAV06] and S-Match [GIU05]. AROMA is an association rule mining-based ontology matcher. S-Match discovers semantic correspondences between concepts by computing the semantic information implicitly or explicitly codified in class labels.

BLOOMS [JAI10b] is a bootstrapping ontology matching tool for aligning information between datasets/ontologies in the LOD cloud at the schema level. The system generates links be-

[14] http://www.iptc.org

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

tween class hierarchies (taxonomies), with subclass and equivalent relations, using Wikipedia (articles and categories) and DBPedia. The advantage of this tool is that it not only maps LOD datasets but can also use and map to upper level ontologies such as SUMO and DOLCE.

Addressing the need identified in [JAI10a] to align the central LOD vocabularies under a uniform upper ontology, the authors of [DAM10] present an approach to facilitate access of LOD data through a single ontology (namely PROTON). Their approach includes using "ontology expressions" to make matching rules for concepts and relations between PROTON and selected LOD datasets and adding new instances to PROTON through inferencing and concludes with an extended version of PROTON, with added classes and properties useful for uniformly accessing the LOD datasets.

This approach instigated a more sophisticated extension of BLOOMS, namely BLOOMS+ [JAI11] that maps LOD datasets to PROTON, which takes into consideration contextual information to support the decision of if and which concepts/relations are going to be aligned.

2.3 The WP4 ontology

This section will describe the design principles of a dedicated reference ontology upon which user preferences and context can be modelled in the digital media domain, based on the requirements identified in the previous sections. We will also present a first view of the LinkedTV user model ontology and an example toy view of its future development.

2.3.1 Design principles

The requirements on deciding over the most suitable semantic knowledge for the users of LinkedTV includes determining the level of granularity, the semantic precision, and the expressivity of the ontology with regard to appropriate inferential services, such as logical reasoning. Another important issue to be elucidated is the content of the ontology. Every concept, relation and rule that may have meaning to a user in the scope of LinkedTV should be represented in the LinkedTV user model ontology (LUMO).

2.3.1.1 Lightweight reference ontology

In order to keep user models and their usage in the information system lightweight and manageable we identify the need to build and maintain an ontological knowledge base that a) can support meaningful representation of world semantics under *a single uniform vocabulary*, b) will encompass the minimum possible concept space among the ample information in the networked media domain with regards to addressing user needs and c) will be able to sustain abstract user-specific conceptualisations such as user status, skill and situation. Evidently, various structured information (different sources, formats) should be integrated, mediated and exchanged within this ontology.

To this end, we will compose a core upper ontology based on existing broadcasting-related and user-related ontologies by re-using and extending existing upper/middle ontologies and taxonomies, such as the BBC programmes ontology, PROTON general entities and semantics, IPTC-WebTLab categories, DBPedia/schema.org upper taxonomy and GUMO subsets. This ontology should bring together context-related (sensor data, user situation) and interest-related semantic information.

In the interest of further conveying more expressive knowledge pertinent to the user, we will further consider building upon the core ontology more granular domain-specific ontologies that will correspond to the LinkedTV scenarios. In order to retrieve all the comprehensive semantic information to be incorporated in these ontologies we can automatically analyse the domains by identifying relevant DBPedia subsets and their interrelations and using them to retrieve the conceptual representations of the domains in existing ontologies.

The selection of the appropriate entities and semantics imported from source ontologies to the LinkedTV reference ontology will focus on:

- Discriminative upper level taxonomy, required for effective low dimensional categorization of entire videos (cf. section 4.1.1) and effective estimation of utility functions (cf. section 5.2).
- Addressing main conceptual features of the LinkedTV scenarios (news, art & culture with focus on antiques).
- Addressing the most relevant sensor information obtained through behavioural tracking.
- Addressing the specific contextual parameters pertaining to the users of the LinkedTV platform.
- Finally and foremost, minimizing the concept space in such a manner that will render it as lightweight as possible, though complete enough to make predictive inference based only on the LinkedTV concept space without extending to additional resources. This might include:
    - Aligning only information that is relevant to the users of LinkedTV in the ontology, e.g. given an imported class from a vocabulary/ontology we might not include all its subclasses as some might be deemed as irrelevant for the user.
    - Similarly, adding information pertinent to a user that the initial vocabulary/ontology might be lacking, e.g. given the previous example, we might also need to add classes that are relevant to the user but was missing from the imported taxonomy, or to produce connections via properties between classes that were not initially connected (e.g. SportsTeam → hasTopic Sports).
    - Detaching the reference ontology to be used for modelling user interests and inferencing from its mappings to external vocabularies.

2.3.1.2 Mappings

As a result of the principles described before, the alignment and mapping detection process will also aim at reducing the reference concept space into a more dense and meaningful vocabulary by integrating semantically equivalent information from different vocabularies under a single conceptualisation with multilingual support. In this section we will illustrate primary thoughts and work on determining the means to create mappings between the information provided by media annotation and the core ontology.

While media annotations can be expressed in a common, language-agnostic vocabulary, the same doesn’t hold for text (subtitles, articles, social networks activity) and audio content. Target users of the LinkedTV platform do not share the same natural language since the project’s envisioned scenarios involve content in several different languages, namely English, German, Dutch and possibly also French, thus obligating content annotation to comply with different vocabularies for the same recognised entity (e.g. de.dbpedia:Fußbal and dbpedia:AssociationFootball semantically denote the same concept but may not be aligned under a single conceptualisation in the annotation of two content items expressed in the two respective languages). In effect, the choice of an ontology for LinkedTV may arguably be based on a common natural language (e.g. English) with multilingual support across LinkedTV scenarios in order to facilitate re-use and exchange of knowledge.

The same assumption might hold for the annotation of different individuals to semantically equivalent concepts from different vocabularies. E.g. one particular building may be classified as a dbpedia:’Building’ in one content item and another building as a schema.org:’Civic Structure’ in another content item. However the relevant information for the user would be that both instances are buildings, i.e. belonging to the lumo:Building class (we can inversely not have a dedicated lumo:Building class but have imported the dbpedia:’Building’ class to our concept space as a unique reference of this conceptualisation).

To this end, the core ontology developed will be modelled under one language and enriched with adequate information to align multilingual and cross-vocabulary information under a unified conceptualisation. In the interest of maintaining the core ontology lightweight we will maintain a single conceptualisation for each class in the user model concept space and a separate mappings archive in order to avoid redundant steps in the inferencing process, such as dbpedia:’Building’ ≡ schema.org:’Civic Structure’ ≡ lumo:Building or de.dbpedia:Fußbal ≡ dbpedia:AssociationFootball ≡ Lumo:Football.

Alternatively, we will consider in future extensions indexing the class’ mappings in the annotation description of the class, under a dedicated annotation property. E.g. given a class in LUMO lumo:Building, if it is found to be semantically equivalent with the class dbpedia:’Building’ then instead of adding the concept dbpedia:’Building’ to the ontology or a mappings archive and an axiom dbpedia:’Building’ ≡ lumo:Building, we can add a predefined descriptive property in the annotation of lumo:Building denoting that axiom dbpedia:’Building’ describes the class:

```
<owl:Class rdf:about="lumo:Building ">
      <isDescribedBy> dbpedia:’Building’</isDescribedBy>
```

```
</owl:Class>
```

The purpose of this considered approach is twofold. It will accommodate producing mappings between concepts for which the concept similarity (assessed by some ontology alignment tool) is not binary, i.e. we can assign a degree of certainty of the matching class describing the seed class, and that might also not be directly semantically equivalent. This degree will improve the quality of the classification as it will be used to modify the degree of confidence by which the given concept participates in the content. It is also expected that by using such indices the classification process can be conducted through simple search and retrieval rather than real-time inferencing in a mappings ontology, thus rendering it arguably faster and lighter.

The second justification point is hypothetical and based on the expected outlook of the semantic interpretation of additional content provided by WP2. Although this is a subject to be investigated later within the course of the project, we might assume that real-time classification of additional content might be incomplete or at times infeasible due to resource limitations and lack of information, but textual descriptions (tokens) might be easily extracted. Thus textual descriptions might also need to be considered alongside available semantic descriptions. By indexing prominent textual descriptions along with semantic descriptions for the (presumed limited) LinkedTV concept space at the schema level is expected to render the classification process more agile and swift.

In essence, the proposed approach aims to aggregate all conceptual and/or lexical information that describe a concept under a single descriptive feature vector. Such feature vectors describing the meaning of a concept have been introduced before in [TSA09], however in that case consisted only of textual information. The value of aggregating interlinked semantic information under these vectors is expected to highly improve the semantic classification of annotated content. In effect, these features will be used to classify the content based on its semantic annotation or textual information under a uniform ontological vocabulary with reduced dimensionality.

Mappings retrieval can be further supported by capitalizing on LinkedTV's envisioned Linked Media Layer. The Linked Media Layer will be interlinking fragments of multimedia content and annotations schemas and will comprise a core tool for expanded multimedia annotation in LinkedTV.

Initial work on creating mappings for LUMO will be conducted manually, since the LOD vocabularies that will be used to annotate content are predefined (cf. D2.2) and will be aided by the NERD ontology and schema.org mappings. In a second stage and after the developed ontology will be versioned into more concise and comprehensive semantics, we will focus on experimenting with the BLOOMS tool in matching classes from the LinkedTV core ontology with classes from different LOD vocabularies used to annotate multimedia content at the schema level.

2.3.2 First view of the WP4 user ontology

A user model contains two main aspects: a description of the user himself (age, profession, social status, etc.) which is more straightforward, and a representation of those things in the world he is interested in. While the representation of user attributes in a reference ontology is straightforward and relies on standard demographic information, the representation of the interests is more complex. Just to maintain a list of topics a user is interested in would not allow us to manage the many interrelationships between topics. A user may be interested in German history and expects that the system knows that the Thirty Years' War was one of the main periods in 17th century German history, or that Bismarck was a major politician in 19th century in Germany. The fields of interest users may have will typically cover a very broad spectrum – from people and organizations to places, events, and, of course, a wide range of content topics like politics, natural science, arts, music, etc.

In most cases these fields of interest will be specified much more precisely: not just music but the beat music of the 60’s, not just sports but the English premium football league (and especially ManU and Arsenal), etc. As a consequence, we have to express both a wide spectrum of interests and sometimes very special preferences. All these topics are related in various ways.

Our user model has to be able to represent these different aspects. That’s the reason why our user model is based on a user-pertinent reference ontology. To this end we are in the progress of constructing a user model ontology for LinkedTV in the networked media domain called LUMO, which represents the main concepts, topics, concrete entities, and semantic relationships between them that the user maintains about the world.

A certain part of the user models contains a kind of standard common sense knowledge that every user (in a certain culture) has about the world: we know that events are connected to places, that frequently people play a role in them, or that companies, institutes, and football clubs are organisations of people. Consequently, the core user model ontology can be the same for all users. The main issue is that this ontology should be able to represent the *mental model* of the user, i.e., contain those terms and their relations the user is using to express his knowledge about the world and his preferences in media consumption.

As a consequence of the situation outlined in the previous subsection we propose our approach to modeling a reference ontology for semantic user models. It is based on the following considerations:

- It represents the mental model of a typical user. We avoid “ontological overhead” like endurants and perdurants and concentrate on the main things a user may be interested in.
- We integrate those parts of existing Web ontologies which are needed to represent the more special interests of a user. Instead of introducing our own music categories we in-

tegrate the music concepts from the music ontology[16]. Instead of building our own topic hierarchy in physics we integrate the physics categories of DBPedia (to that extend needed by the user in order to represent his specific interests in this field).

This approach brings the two main aspects of semantic user models together: to represent the typical structure of a user mental model, and to integrate as much as possible of existing Web ontologies (supporting matches with multimedia annotations).

We introduce the LinkedTV User Model Ontology (UMO) as basis for user models in LinkedTV:

- It has eight top level concepts:
  - people,
  - organizations,
  - locations,
  - time,
  - events,
  - products,
  - topics, and
  - media categories.
- They comprise all main issues of user interests.
- The first version of LUMO will be published in the beginning of the second year of the project under the http://data.linkedtv.eu/lumo/ namespace.
- These concepts can be refined and extended using our own concepts *or* concepts from existing ontologies. For instance, locations can be connected to items in the GeoNames[17] dataspace, events can be refined through schema.org event subconcepts, etc. Concepts from existing ontologies come with their original namespaces clearly indicating their semantic embedding. At the first stage of the mapping process, mappings are aggregated directly to the ontology with an aim to define the needed concept space at first and then separate it from its mappings at a second stage. So, if similar concepts from different ontologies are integrated into the LUMO they are connected to each other through equivalentConcept relations.
- Concrete instances (named entities) can be integrated into UMO in the same way.

[16] http://musicontology.com/
[17] http://www.geonames.org/

- In addition to subclass and type relations we will also maintain domain specific relations. At the moment, we foresee only one kind of such relations by reasons of practicality: relatedTo. All domain specific relations we want to take into consideration in this way are mapped onto such relatedTo links. Which relations this will be has to be decided from the domain ontologies. This will help us to keep the complexity manageable. The main criteria are semantic importance and clarity. In later versions we may extend this approach to a broader spectrum of relations.

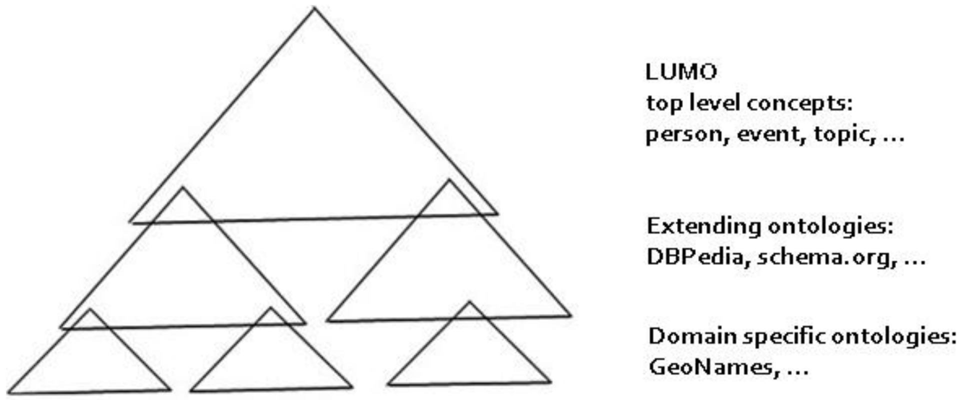

Figure 3: The LinkedTV User Model Ontology abstract schema

As a consequence, the most specific concepts in our user model ontology will typically come from domain specific Web ontologies. They will not necessarily be the most specific concepts there – there may be more specific ones which are also used to classify instances. This fact will be used to specify semantic mappings.

Sometimes in a user model concepts are needed which combine different aspects: French Baroque music combines the main category music with a geographic and a temporal aspect; or “active German politicians” relates politicians to the location Germany and the time period ‘now’. Formal concept modeling techniques (like Description Logics) allow us to describe such conceptual relations in a well defined way.

Instances have to be classified on run time to this user defined subconcept. The best way for such special concepts is if they exist already in their respective domain ontologies (and are used in the multimedia annotations properly).

At the beginning, we will create the LinkedTV UMO and its mappings manually from a set of selected Web ontologies. This UMO will be used as the ontological basis for the user model in the first project phase. This selection should be aligned as much as possible with the multimedia annotations used in WP2. Later, we will extend it towards a more flexible and modular approach. A snapshot of current version of the UMO can be seen in Figure 4.

Figure 4: A snapshot of the LUMO[18]

2.3.2.1 An example of future extensions

Since the task of producing a comprehensive, complete and consistent ontology that models such a broad domain as networked media is non-trivial, the LinkedTV UMO is currently at its infant steps in development, mainly attempting to gather all possible information from the relevant concept space, with a plan to minimize (e.g. separate the needed concept space from its mappings) and better structure it after firstly aggregating all relevant domain information from relevant vocabularies. However, the validity of the use of a dedicated ontology and of the proposed lightweight user profiling methodology can only be efficiently demonstrated through an ontology that strictly maintains the aforementioned design principles (section 2.3.1).

Therefore, for the purpose of illustrating the functionalities and advantages of the profiling methodology, we present a fabricated toy example of envisioned slim and well-structured

---

[18] Since it is not possible to illustrate the IRIs in all the different mappings through a snapshot of the ontology from an ontology editor, we exemplify here the mappings for the LUMO concept ‘Location’:

```
<EquivalentClasses>
	<Class abbreviatedIRI="dbpedia:LOCATION"/>
	<Class abbreviatedIRI="schema:Place"/>
	<Class abbreviatedIRI="lumo:Location"/>
</EquivalentClasses>
```

subsections of the ontology, with domain information mainly oriented towards the news domain, to be followed through the examples of this deliverable, based on a combination of the existing ontology and the aforementioned design principles.

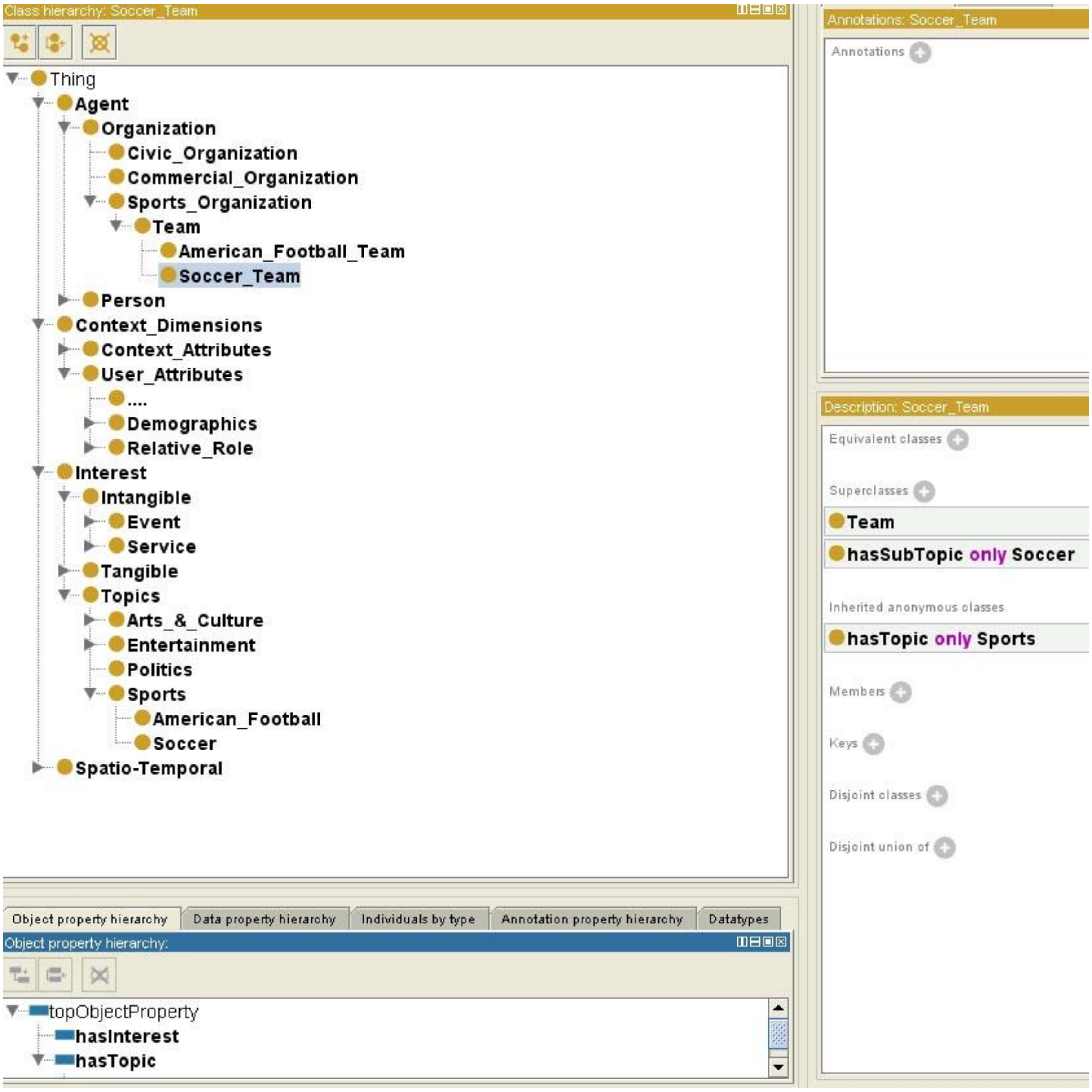

Figure 5: The toy fabricated UMO subset for exemplary purposes

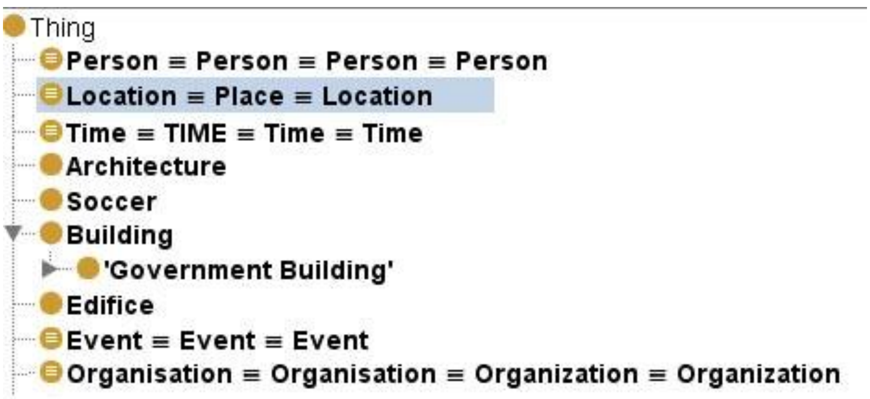

Figure 6: Example of mappings in a separate file[19]

Figure 7: Example of mappings being indexed at the annotations of a class

To simplify the example ontology we do not include dedicated individuals in this model, rather we focus on the schema level. For example purposes we assume that an instance of a LOD dataset will be mapped to some class of the example schema ontology (cf. section 4) based on retrieved mappings at the schema level, thus we may use the original LOD individual to instantiate classes in the formal, structured LUMO.

[19] Refer to footnote 18

3 User interaction listeners

This section describes the mechanisms that will be used to track user interactions within the LinkedTV platform and extract descriptive features (transaction statistics, behavioural analysis, social activity mining) that will determine which actions are performed by the user in relation to consumed content. These features will subsequently help detecting which given content item/fragment/object is most interesting/uninteresting to the user, determine which concepts exactly the user implicitly demonstrates a preference on. They will also indicate the nature of that preference (positive/negative) and the impact of the preference to the profile.

This section is focused on presenting the mechanisms used to bridge the interface front-end to the algorithmic back-end of WP4 in order to track user interactions with the platform. Such interactions might include transactions with the seed and additional content (e.g. playback selections on a media fragment, bookmarking, skipping fragments in seed content, selecting a triggerable object, selecting a recommended related content item etc), reactional behaviour towards displayed content and an introduction to extracting the related social activity of the user.

3.1 Tracking the transactional behaviour

The tracking mechanism is an important functionality of the LinkedTV platform. The mechanism is able to collect information about user behavior – so called implicit feedback. This section is focused on tracking the behavior of the users based on their online transactions with content within the LinkedTV platform.

For the purpose of LinkedTV we will use the General Analytics Interceptor (GAIN)[20] tool in order to collect transactional information about user behavior.

3.1.1 Architecture

Figure 8 illustrates the main parts of the tracking mechanism consisting of the Tracking module, Tracking API and Tracking Analysis.

The tracking module is the first part of the tracking mechanism and consists of a module integrated into the LinkedTV interface/player. This module sends information about the user behaviour as events (user paused, skipped video, view additional content etc.), which occur during the interaction of user with the interface, take place. The physical tracking module is designed in a similar way. Events are sent from the Tracking module to the predefined Tracking API for processing. In the second year of the project, further LinkedTV-specific events, such as user selecting an object on the seed video, will be added.

[20] wa.vse.cz

The tracking API processes events from many interfaces and is designed as a scalable service. The API provides services that can:

1. collect interactions – like play, pause, skip, etc.
2. get aggregated stats about interactions – number of interactions per day, average number of interactions per video etc.
3. get results of analysis for each user.- like interest level per shot, video. etc.

Examples of the services can be found in section 3.2.3.

The tracking Analysis module processes and analyses collected data using different algorithms. The module responsible for analysis also uses annotations of media fragments to better represent the media and to understand the user (More in section 4.1). Results of analysis are provided as user profiles updates. The interest of users can be incremented or decremented based on the event types.

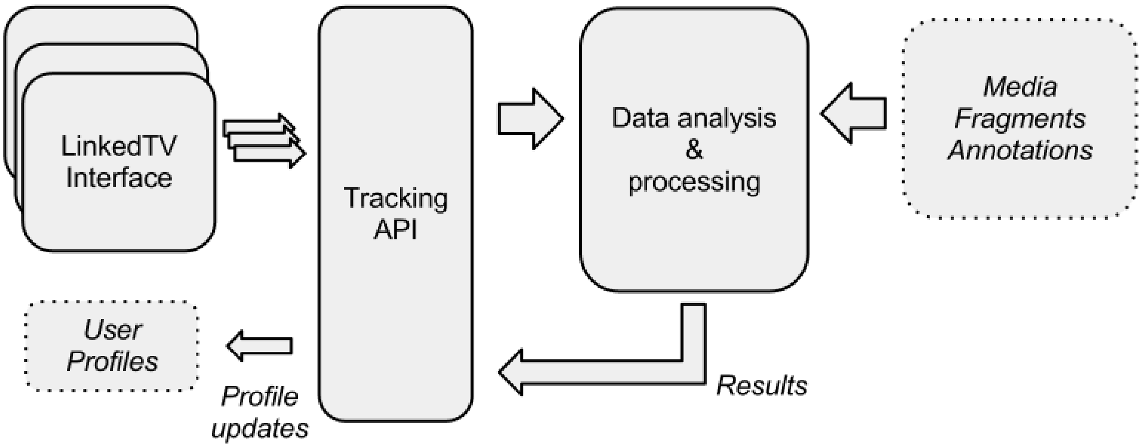

Figure 8: The transactional tracking mechanism architecture

3.1.2 Input and output

The input of the tracking mechanism is derived from the tracking module integrated into the player (click-through behaviour) and from the annotations of the media content. Such click-through behaviour might incorporate interactions on the LinkedTV player (play, pause, skip etc), clicking on a triggerable (semantically interpreted) object in the multimedia content, clicking on a recommended concept or clicking on a recommended content item.

The output results are in a form of aggregated stats about the user behaviour (e.g. average number of interactions pert video) and the levels of user interest (e.g. interest level per video shot). This information will be used as input for personalization and recommendation of media content and can be also used as updates of user profiles.

3.1.3 API

GAIN provides a REST[21] API. The current first version 0.1 provides mainly services for tracking and services for simple aggregated stats about interactions (average number of interactions per video etc.). The next version of the API will provide services to get results of tracking analysis (like interest level of user per video shot etc.) suitable for update of user profiles. Some examples of provided services can be seen in APPENDIX I: Examples of REST services for transactional tracking.

3.2 Tracking the physical behaviour

User behaviour tracking is currently under development. A Kinect [KIN10] sensor which is able to acquire both colour images and a depth map is used. The audio capabilities of the Kinect are not considered at this point. The Kinect is located on the top of the TV with a field of view of about 6 meters long for 4 meters wide which quite well fits with classical TV arrangements in homes.

The use of the Kinect for implicit behaviour tracking is mainly focused on two aspects: inferring user context and the user interest while viewing some content. The first aspect involves determining the user context to adapt the user profile in contextual instances. For example, is the user alone or not, are there children in the room, is there any light in the room or not? By the end of the first year of the project, a Kinect-based system has been developed which counts the number of people in the sensor range. This information can be fused with other data about the viewing hour or the week day.

The second aspect is about complementing information able to provide a degree of interest of the viewer with respect to his/her reaction while consuming a content item (seed video) and its enrichment (media linked within the seed video). Current development is focused towards monitoring user behaviour during watching the seed video. Interest detection at given moments means that the user might be also interested in enrichment of the seed video at that precise moment. Of course, additional information about enrichment manipulation is needed, but this first cue on user interest is useful to detect the kind of enrichment in which the user might be interested in. Knowing which scenes or shots attract user interest helps to define the enrichment linked to those scenes as potentially interesting.

The scenario to be tested will include one user who will be asked to perform a task like answering a questionnaire (primary task) while watching TV (secondary task). The primary task should not be too intense to avoid demobilising the cognitive capabilities of the user. The purpose is to find out when the user stops focusing on his primary task to watch TV. This is observed only when the content on the TV seems very interesting or relevant to the user.

The Kinect observes the scene and is able to extract the following features:

[21] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

- *User face recognition.* The user goes in a specific place and looks to the camera (this place is closer to the Kinect as face recognition needs to analyse face details). The program analyses the face using state of the art algorithms like PCA decomposition [TURK91] and asks the user if he validates the recognized profile. The user can validate, perform the recognition again or register as new profile.
- *Regions of interest.* The user can select the sofa for example by a simple click. This area is automatically segmented and the system knows each time when the user is in this area. The sofa can be seen as an area where the user might be interested in watching TV. The user behaviour tracking is only initiated if the user enters this area.

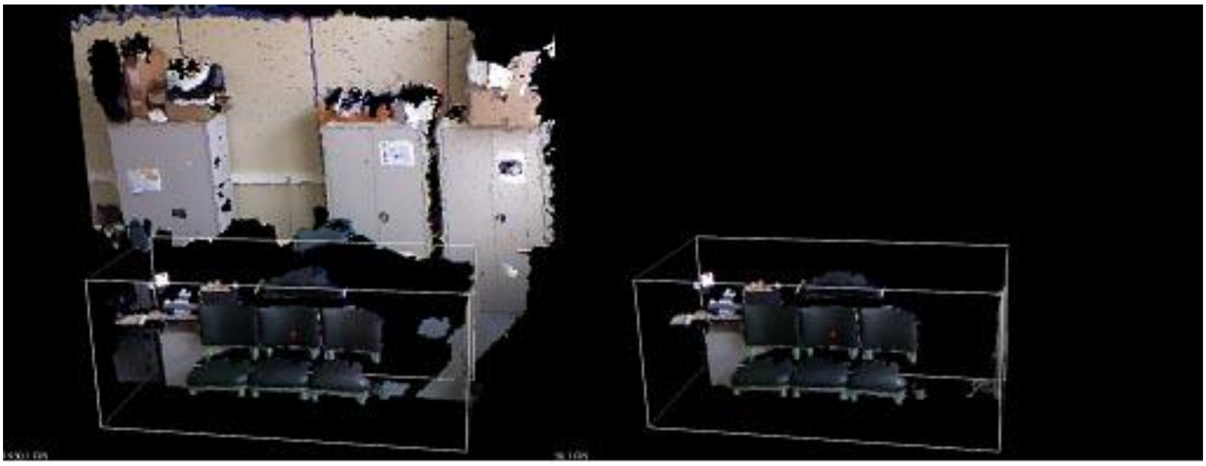

Figure 9: Left image: click on the sofa, Right image; the sofa is segmented.

- *User head localisation*. The user head is localised in 3D coordinates as a bounding cube centered on the skeleton head. In the same time, face detection is performed to detect when the face is looking towards the camera or not. This feature is not too sensitive as face is well detected unless it has a direction very different from the one of the camera, but it is interesting to see if the user is really looking in a very different direction (something happens in the home, the user talks to someone else...). In this case we can infer that user attention is close to 0 during a period where his face is oriented away from the TV set.

Figure 10: The face detector is present in a head area: the user is looking towards the camera axis.

- *User head direction.* After face detection, it is possible to extract the face direction. This direction needs the Kinect sensor to be closer to the face but we can assume that next generation of depth sensors will have a much higher resolution and the same result can be achieved with a Kinect located above the TV. The result may be noisy, so a filtering post-processing is needed to detect the important head direction changes and especially the moments where the head points towards the TV for a period of time which is long enough to determine interest for the on-screen content.
- *User barycenter evolution*. The Kinect sensor can extract the user skeleton. This skeleton barycentre (which generally corresponds to the umbilical region of a person's body) is a quite stable feature compared to hands and feet. Thus, in a first approach, this feature shows the user excitement without explaining which body part is responsible of this. A change in excitement can be used as a sign of interest for the content.
- *User sitting or standing*. In this case it is reasonable to assume that standing means less interest in what the TV display.

Regarding all these features, we must point out that they do not directly depict information on user behaviour if taken globally. In effect, sudden changes relative to the user's short-term physical history and a combination of features will be more descriptive of the level of user engagement to the broadcasted content.

Indeed, if a user has been non-constant to his orientation towards the TV in his recent behaviour but suddenly changes reactions and he begins to look at the TV for a longer period of time, it is reasonable to infer that he might be more interested in the current content even if other aspects of his behaviour (e.g. transactions with the platform) is neutral.

An idea is to provide a classifier with the physical features and their variations to the transactional tracking mechanism previously described, which will provide the degree of interest of the person after learning users' physical features. This supervised approach needs to run several tests and segment the degree of interest the users may have for the TV content manually.

Additional features (for example face orientation, face analysis) could be extracted either with the available Kinect camera from a closer distance or with an enhanced Kinect which will probably be available before the end of the project. But in a first stage we extract features available using only a Kinect on top of the TV or one closer to the user's face.

In addition to those features, proxemic features could also be used between different users, but for this first approach only one user is taken into account.

3.3 Tracking social interactions

Another possibility for generating and evolving a model of user's interests is to use their activity in the Social Web as an indicator. Today's Internet users are sharing their interests using social platforms like Facebook[22] and Twitter[23], and it is a subject of research whether past Social Web behaviour can be a solution to the well known cold-start problem in recommendation systems (that the system does not know anything about a user's interests the first time they use the system). This was explored previously in the NoTube EU project, in work that became known as the Beancounter[24].

The Beancounter is an online tool for generating a user model through semantic analysis of the user's posts/shares on specific social networks. For example, Twitter tweets are public and can be queried by an open API, while Facebook allows third party applications to access details of a user's timeline via its Facebook Connect API, in which the user logs in from the application to Facebook and expressly gives permissions for the application to read his or her posts. Many of the common social networks today support a specification called Activity Streams which provides a structured model for the data being published over time (Facebook posts, Twitter tweets, etc.). Activity Streams[25] extend typical RSS/Atom feed structures (with a title, description, link and some metadata potentially expressed for each posted item) with a *verb* and an *object type* to allow expressions of intent and meaning as well as to give a means to syndicate user activities.

The Beancounter architecture makes use of components called *Tubelets* to interface between the Beancounter and different sources of Social Web data. The key goal of a tubelet is to be able to authenticate the application to the Social Web network's API and to retrieve via an API query a dump of the user's posts to that network over a period of passed time (the extent of the data returned varies from API to API).

[22] www.facebook.com/

[23] https://twitter.com/

[24] http://notube.tv/category/beancounter/

[25] http://activitystrea.ms/

4 Understanding the user

This section focuses on detecting user preferences based on the information tracked from the user's interaction with the platform. This includes the extraction of information from media fragments that the user has interacted with and the semantic interpretation of extracted information, this information stemming from several resources, and consequently mapping them under the uniform LinkedTV user model ontology (LUMO).

4.1 Semantic interpretation of content

The personalization layer requires that the video items in user's history are described by a small fixed number of features that correspond to criteria, which the user (in part subconsciously) applies to assess the degree of interestingness of a particular content item/fragment or single concept.

4.1.1 Description of videos – the low dimensional approach

In this section, the input provided from low dimensional interpretation of video content (WP1) and its interpretation as input to the profiling process will be presented, with an interest of depicting the liaison with the video-extracted information and their role in the profiling process. There are three possibilities to obtain such a description of videos:

a) from content providers,

b) multiclass classification,

c) clustering.

4.1.1.1 From content providers

The ideal source of semantic description of videos is human annotation supplied by the content provider (broadcaster). For example, in some countries TV guides describe movies (as a particular type of video) with number of stars assigned in several categories (romance, comedy, documentary, thriller, etc). Each content item is described by the same set of criteria. Using zero to five assigned "stars" to semantically describe the content is depicted at Table 2. Since we aim to use weights in the [0,1] range, zero assigned stars is represented with 0 value, one star as 0.2, two stars as 0.4, three stars as 0.6, four stars as 0.8 and the highest possible ranking of five stars value 1.0.

Use of semantic description from content providers within LinkedTV is a possibility, the feasibility of which is being investigated.

Table 2: An example description of videos by a broadcaster

VIDEO NAME	SEMANTIC DESCRIPTION		
	Sports	**Architecture**	**…**

prozess	0	0.8	…
stadt	0	0	…
infanticide	0	1.0	…
ratze	1.0	0	…

4.1.1.2 Multiclass classification

Provision of human classification of videos can be prohibitively costly for shorter videos, such as news items. For these content types, it may be feasible to classify only a limited number of videos per target class and use this as a training data. The type of machine-learning problem involved is multiclass document categorization.

Application of this technique requires:

i) Specification of the set of target classes.
ii) Sufficient number of labeled examples (videos) for each target class

4.1.1.3 Clustering

Unsupervised "classification" through clustering will be generated in WP1 to provide input for the personalization layer. The result of this process is a degree of assignment of each video to *all clusters,* thus achieving soft multi-class classification.

Table 3: Clustering result example. This example was fabricated so that the clusters correspond to topics "Sports" and "Architecture" (compare with Table 2)

VIDEO NAME	SEMANTIC DESCRIPTION		
	Cluster 1 (Sports) membership confidence	**Cluster 2 (Architecture) membership confidence**	**…**
prozess	0.03	0.88	…
stadt	0.05	0.93	…
infanticide	0.03	0.96	…
ratze	0.97	0.05	…

4.1.2 Media fragments – the fine grained ontology-aware approach

The semantic information may come from entities (individuals or upper concepts/classes) appearing in the particular video fragment (scene/shot/object). These entities can be described according to a taxonomy/ontology used in WP2 such as schema.org or the NERD ontology. The description of entities coming from WP2 can be of two types – crisp or fuzzy.

- Crisp type: Entities are assigned one type by WP2 NER recognition tools such as NERD or the SemiTags (cf. D2.3) tool. For example, “White House” could be assigned the type “Government Building” according to the schema.org ontology.
- Fuzzy type: Entity is assigned a membership value with multiple (possibly all) ontology concepts. Such classification can be performed by the WP2 BOA (cf. D2.3) algorithm or WordNet similarity measures.

Both entity types are used to describe a shot by a feature vector with the number of dimensions equal to the number of concepts in the input taxonomy. However, these representations differ in what is the value of the feature corresponding to a taxonomy concept:

- Crisp shot description (example in Table 4)
  - Entity name – if the concept is a direct hypernym (type) for the entity
  - Null – if the concept is not a hypernym for the concept and the entity
  - The name of the closest hyponym – if there is a hypernym-hyponym relationship between the concept and the entity, but it is not direct, the value of the feature is the name of the closest hyponym.
- Fuzzy shot description (example in Table 5)
  - The entry contains the similarity (membership value) of the entity with the type.
  - There are multiple possibilities for propagating similarities through the taxonomy. The proposed approach propagates the similarities in an analogy to the Resnik counting [RES95].

Although the entities and their description are generated in WP2, this subsection describes the preprocessing of WP2 output before it can be used to create the user profile in WP4. The subject of this subsection is thus work performed on the border of the two workpackages. Since the preprocessing described here is specific for the personalization purposes, and it is not normally carried out within Named Entity Recognition systems, which are the focus of WP2, this topic is covered in this deliverable.

A fabricated example for both crisp and fuzzy approach to representing semantic information at shot level follows.

4.1.2.1 Crisp shot description

In the crisp representation, an entity is assigned *one* class from the ontology. The semantic subvector of Table 4 contains one instance per ontology class.

In the semantic subvector, underline is used to highlight the feature *activated* directly as a result of an entity in the Entities column being assigned the corresponding type.

The activation is propagated up the taxonomy, while the value of features below the activated features is left unset (NULL).

For example, consider session 2 shot 2 in Table 4: “Thames House” is recognized as an instance of the Government Building class: “Government Building” is set to be instantiated by “Thames House”. The fact that “Thames House“ is also an instance of “Civic Structure” and “Place” is propagated in the following way: the “Civic Structure” feature is set to “Government building” and the “Place” feature is set to “Civic structure”. In contrast, the entity “MI5” is recognized as a type of the top-level “Organization” class – only the value of the organization feature is set to “MI5”, other features in this dimension are not set (or set to NULL).

Note that in this example, the two entities contained in the shot did not interfere, since each was assigned a type from a different semantic dimension, i.e. instance. The semantic subvector here contains three *semantic dimensions* – Place, Organization, and Creative Work. Handling the case when multiple entities with type belonging to the same semantic dimension are present is exemplified in Table 4 for session 1 shot 3.

Table 4: Crisp version of shot based input. One user, two videos, each divided into three shots.

Session	Video	Shot	Length	Semantic subvector based on a LOD vocabulary								Entities
				Place			Organization			Creative Work		
				Place	Civic Structure			Corporation	Sports Team		TV Series	
						Government Building						
1	News1_USFootball	1	28	Civic Structure	Government Building	White House	Sports Team		football team	TV Series	X-Files	White House, X-Files, football team
1	News1_USFootball	2	14				Sports Team		Red skin			Red skin
1	News1_USFootball	3	60	London			U.S. Government					U.S. Government, London, White house
1	News1_USFootball	3	60	Government building	White house	Since two entities within one shot overlapped in their semantic description (both classified as "Place"), an extra line for the second entity (White House) was introduced.						
2	News2_SecurityOlympics	1	20	London			Olympics					Olympics, London
2	News2_SecurityOlympics	2	10	Civic Structure	Government Building	Thames House	MI5					Thames House, MI5
2	News2_SecurityOlympics	3	5	Stadium			Sports Team		Arsenal F.C.			Stadium, Arsenal F.C.

4.1.2.2 Fuzzy shot description

Fuzzy shot description removes two limitations of the crisp shot description

- One entity might be assigned multiple concepts
- One shot can contain multiple entities assigned to concepts from the same semantic dimension[26]

Fuzzy shot description has the following steps:

1. Input entities are assigned concept weights.
2. Weights are propagated through the taxonomy.
3. Weight vectors of all entities within shot are aggregated.
4. The resulting weight vector is normalized.

For example, consider White House and London entities appearing together in one shot. The first step of fuzzy shot description is to assign concept weights for each entity. The example of concepts and weights assigned to entities White House, U.S. Government and London are on Figure 11, Figure 12 and Figure 13. The values of weights are based on the confidence degree returned by the appropriate NER named entity extraction tool such as BOA (cf. D2.3).

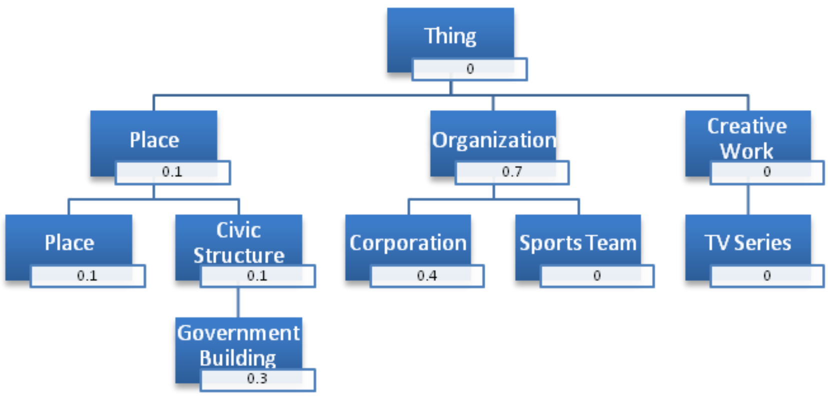

Figure 11: Concepts and weights assigned to entity White House

[26] This is supported by the crisp approach but at the cost of duplicating the corresponding data row.

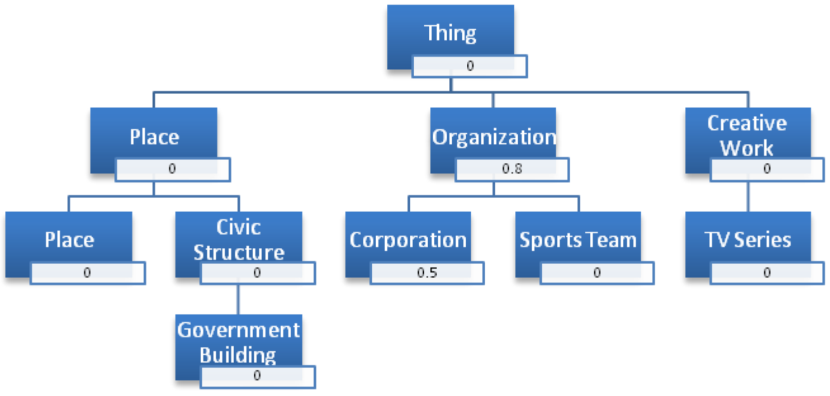

Figure 12: Concepts and weights assigned to entity U.S. Government

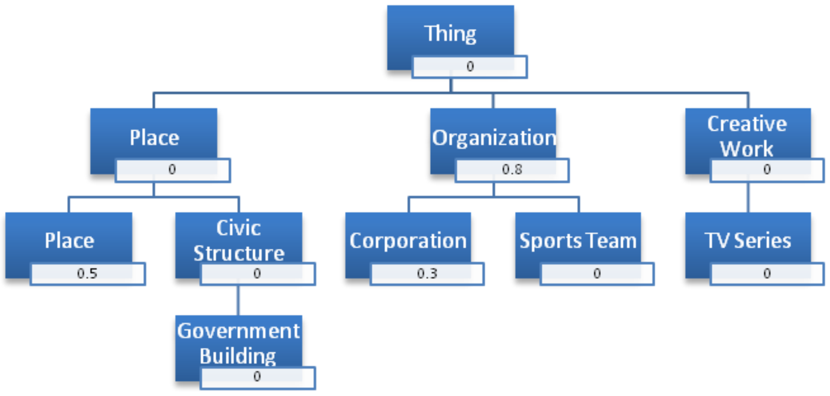

Figure 13: Concepts and weights assigned to entity London

The second step concerns propagation of weights. There is more than one entity within one shot. To get aggregated weights on shot level, we have to aggregate all of them. Firstly the extracted values are propagated up the taxonomy. For each concept in the taxonomy (progressing from leafs to the root): the new concept weight is the sum of the current concept weight and the weight of its children. If there is more than one child with non-zero weight, the sum is divided by the number of contributing children with non-zero weight. The examples are on Figure 14, Figure 15 and Figure 16.

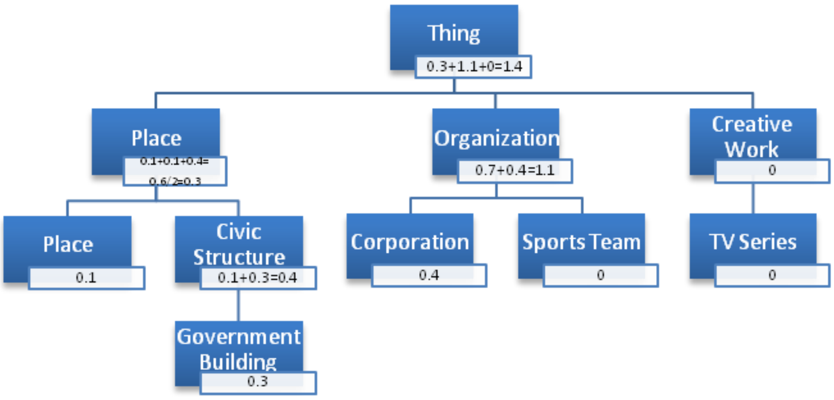

Figure 14: Weight propagation for entity White House

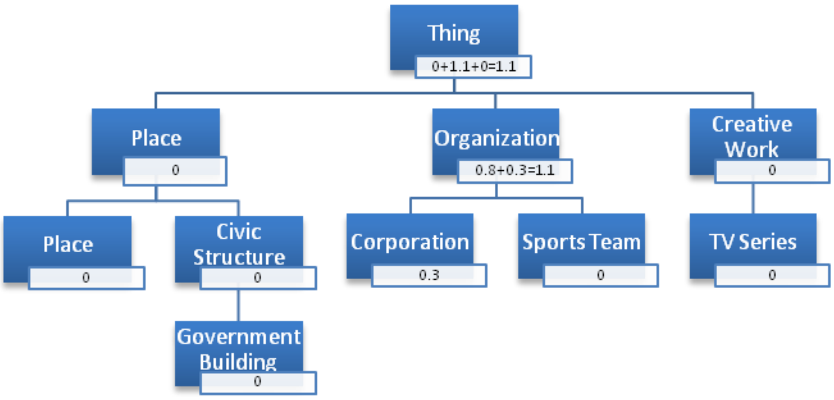

Figure 15: Weight propagation for entity U.S. Government

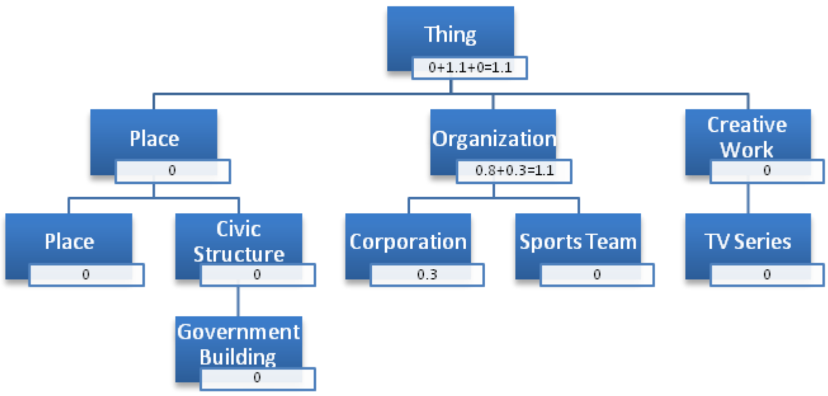

Figure 16: Weight propagation for entity London

The third step pertains to aggregation of weight. As the propagation is finished for each entity within one shot, the aggregation of all weights has to be performed. The proposed approach is to sum the weights of corresponding concepts. An example is on Figure 17.

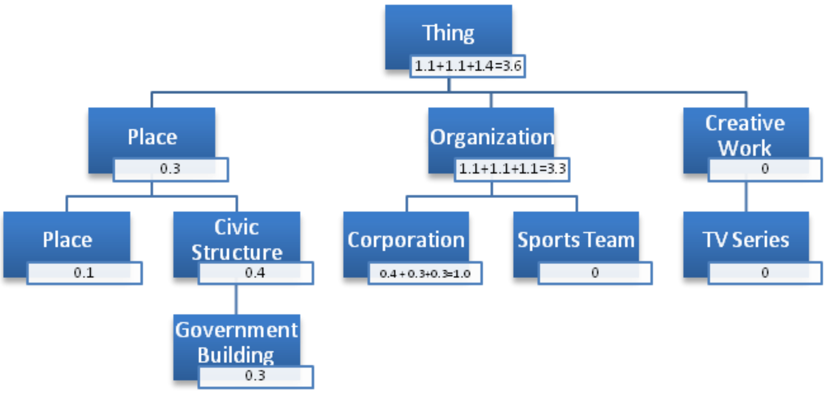

Figure 17: Aggregation of weights per shot

The fourth step pertains to normalization of weights. The aggregated values are normalized by dividing them by the value of the common root in our taxonomy. The example and final values of fuzzy shot description are present on Figure 18.

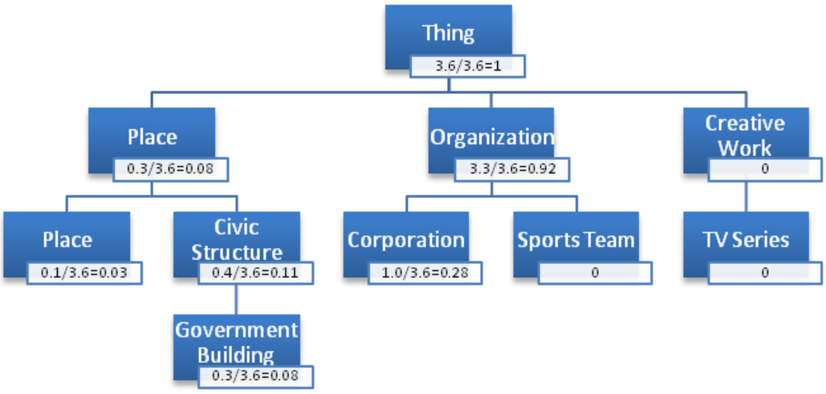

Figure 18: Normalization and final weights

Although weight propagation across related LUMO concepts based on the user model will be conducted in the filtering process (cf. deliverable D4.3), weight propagation across the LOD vocabularies in order to establish whether a particular concept also infers a (weaker) interest in more general LOD concepts is opted. The reason is that LOD-based weight propagation might derive different (more) abstract concepts/preferences that the classification of a specific class to the LUMO (and subsequent propagation in the LUMO-based filterers) might miss.

Table 5: Fuzzy version of shot description. One user, two videos, each divided into three shots. Results shown only for shot 3 video 1.

Session	Video	Shot	Length	Semantic subvector based on a LOD vocabulary								Entities
				Place			Organization			Creative Work		
				Place	Civic Structure			Corporation	Sports Team		TV Se-ries	
						Government Building						
1	News1_USFootball	3	60	0.03	0.11	0.08	0.92	0.28	0	0	0	U.S. Government, London, White house

4.1.3 Understanding information from social interactions

The Beancounter tubelets retrieve a dump of a user's posts in a social network, as described in section 3.3. This dump which is mainly textual (plus links to media) is subjected to entity extraction to derive from the user activity a list of key "named entities" and number of occurrences – for this, the Beancounter is calling directly the Alchemy API[27] to extract the named entities. Alchemy API supports the connection of named entities to concepts from the Linked Data cloud such as from DBPedia, Freebase, UMBEL[28], YAGO and OpenCyc[29]. This process is illustrated below with a tweet that mentions a BBC programme. In this case, the RDF about the programme is retrieved and entity extraction performed, to derive from that tweet that the user has interests in the topics James May, US Air Force, Astronaut, Space Exploration and Apollo Program (all subjects of the tweeted program).

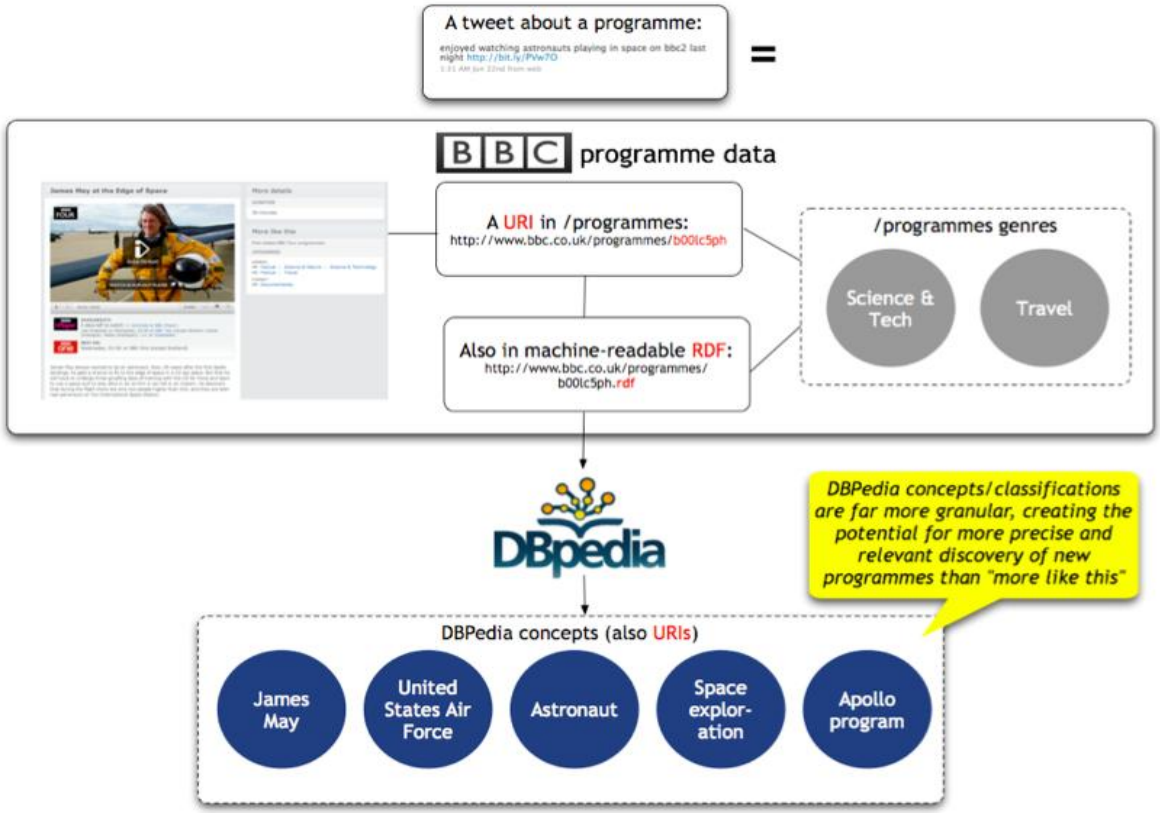

Figure 19: Illustration of Beancounter deriving topics from a tweet.

Hence for the user, Beancounter derives a weighted (by occurrence) list of Linked Data concepts. Also, each source can be given a different weighting which is also applied to each

---

27 http://www.alchemyapi.com/
28 http://www.umbel.org/
29 http://www.alchemyapi.com/api/entity/ldata.html

concept in the list based on from which source it was derived (for example in tests public Twitter posts were weighted lower than Facebook posts shared only with friends, on the assumption private posts would more directly refer to an interest of the user than public tweets).

As a result, an interest graph can be produced for a user made up of topics and weights. The Beancounter uses FOAF (Friend of a Friend) vocabulary[30] as the basis for its user model, with an extension to include weighted interests in the FOAF profile[31]. This extends the existing `foaf:interest` property to include a weighting on the object of the property. Fundamentally however the result of the Beancounter is a set of topics (using DBPedia URIs) each with a weighting value.

Beancounter code is open source, under the Apache 2.0 license[32]. It includes listeners for Facebook and Twitter, as well as a base implementation of topic weighting, as well as storage and serialisation of the resulting user profile. If used in LinkedTV, we would explore, taking advantage of the clear separation of functional components in the code:

- Extension of further listeners to other Social Web sources
- Explore improved named entity extraction from other services such as NERD (cf. LinkedTV deliverable 2.3)
- Analyse the most effective topic weighting algorithm. E.g. if a topic is an instance, its class is also of interest (with a lower weight), and the superclass of that class also (with an even lower weight).
- Analyse the evolution of the user interests weights over time (repeated analysis of Social Web activity) e.g. if a topic is not posted about again its weight could decrease over time.
- Align topic identification in Beancounter to the ontologies chosen in WP2 and WP4 for the media annotation and personalisation. DBPedia, for example, proved in NoTube to be sometimes rather inconsistent in its structure when used for recommendation. It may be that connecting the analysed Social Web data to concepts from other ontologies can be more effective for the personalisation filter.

4.1.4 Mapping LOD entities to LUMO: the uniform and lightweight semantic approach

The concepts identified in the previous processes of section 4 (low level video descriptions, LOD-based annotation of media fragments, LOD-based description of social activities) and are deemed to be of interest or disinterest to the user (based on the techniques described in section 3) will be mapped under the lightweight and uniform LUMO based on available map-

30 http://xmlns.com/foaf/spec

31 http://xmlns.notu.be/wi/

32 https://github.com/dpalmisano/NoTube-Beancounter-2.0

pings (cf. sections 2.3.1.2 and 2.3.2). To better illustrate this process, we will use the clearer toy example of the LinkedTV user model ontology described in 2.3.2.1.

4.1.4.1 Classification via a mappings ontology

Every individual[33] recognised in the semantic content representation will be added or updated in the user's history semantic subvector as an instance of the more specific class in the subsuming LOD hierarchy that maps to LUMO. For instance, let's revisit shot #3 from News1_USFootball from Table 4. The LOD-based concept extraction recognised the individual 'White House', as an instance of schema.org:'Government Building'. The class schema.org:'Government Building' maps to LUMO as a subclass of 'Building' (cf. Figure 6). Therefore the preference retrieved is an instance 'White House' of LUMO class 'Building', which following the fuzzy shot description is present in the shot with a degree of 0.08, this degree also determining the primary impact of the individual to the user before behaviour analysis, detecting a preference to be added or aggregated to the user's consumption history such as: 0.08 · Building(White House).

Similarly, the more general recognised preferences will be mapped to corresponding classes of the LUMO schema. In the previous example, the detected concept schema.org:'Government Building' is subsumed by (detected) schema.org:'Civic Structure' which in terms is subsumed by (detected) schema.org:'Place'. These two classes are deemed as more general preferences since the 'White House' individual does not instantiate them directly but rather as a result of inference. Thus they will be added as abstract concepts which any individual might instantiate. The class schema.org:'Civic Structure' is not directly mapped in the example ontology so it will be disregarded. The class schema.org:'Place' is directly mapped to lumo:'Location' so it will be added as a preference to the consumption history with the extracted degree 0.03 (cf. Table 5), such that: 0.03 · Location(X).

Interpreting the rest of the entities extracted, we follow the same process. We don't expect duplicate entities to emerge since the LOD interpretation process presupposes only one class to be instantiated per individual. In any case, if an entity mapping to LUMO is found more than once in a single content item, we represent the impact of that entity with weight equal to the maximum of the weights among all each distinct instances of that concept.

[33] The term "individual" here is used in the same sense as in the OWL language, i.e. denotes a most specific entity (e.g. a particular person's or organization's name) which belongs to a more general class in an ontology's schema.

Table 6: Classification from a LOD vocabulary to LinkedTV (example) ontology based on mappings ontological structure

<table>
<tr><th rowspan="4">Session</th><th rowspan="4">Video</th><th rowspan="4">Shot</th><th rowspan="4">Length</th><th colspan="4">Semantic subvector based on a LOD vocabulary</th><th>Entities</th></tr>
<tr><th colspan="3">Place</th><th>…</th><td rowspan="3"></td></tr>
<tr><th rowspan="2">Place</th><th colspan="2">Civic Structure</th><th>…</th></tr>
<tr><td></td><th>Government Building</th><th>…</th></tr>
<tr><td>1</td><td>News1_ USFoot-ball</td><td>3</td><td>60</td><td>0.03</td><td>0.11</td><td>0.08</td><td>…</td><td>U.S. Gov-ernment, London, White house</td></tr>
<tr><th rowspan="3">Session</th><th rowspan="3">Video</th><th rowspan="3">Shot</th><th rowspan="3">Length</th><th colspan="4">Semantic subvector based on LUMO</th><th>Entities</th></tr>
<tr><th colspan="3">Place</th><th>…</th><td rowspan="2"></td></tr>
<tr><td></td><th colspan="2">Building</th><th>…</th></tr>
<tr><td>1</td><td>News1_ USFoot-ball</td><td>3</td><td>60</td><td>0.03</td><td></td><td>0.08</td><td>…</td><td>U.S. Gov-ernment, London, White house</td></tr>
<tr><td colspan="4">Preference represen-tation for item News1_USFootball</td><td colspan="3">0.08 · Building(White House)
0.03 · Location(X)</td><td>…</td><td></td></tr>
</table>

4.1.4.2 Classification via indexed mappings

If we were to apply the indexed mapping methodology (cf. section 2.3.1.2) we might be able to benefit from additional information that doesn't map to the class under standard equivalence and subclass relations, and also refactor the participation weight of the classified concept based on the classification confidence degree.

Following the 'White House' example and based on the mapping description of Figure 7, we can see that schema.org:'Government Building' maps to lumo:Building by a degree of 0.8 since the two concepts are not semantically equivalent but bare prominent semantic similarity (i.e. a government building is definitely a building but not all buildings are government buildngs). Therefore the weight of the term Building(White House) is modified as 0.08 · 0.8 = 0.064.

Similarly the parent of lumo:Building → lumo:Edifice might include more information that map to the specific seed LOD ontology that could not be incorporated in an equivalence/subclass relation such as depicted in Figure 20.

Figure 20: Entity 'Edifice' index mappings

So schema.org:'Civic Structure' now maps to class in LUMO (a civic structure is definitely an edifice but not all edifices are civic structures), which modifies the semantic subvector of the shot by adding the class ($0.11 \cdot 0.75 =$) $0.0825 \cdot$ lumo:Edifice. The linkdtv:Location class participation remains unchanged as only a direct equivalent is found in the mappings (semantic equivalence denoted by a degree = 1.0).

Table 7: Classification from a LOD vocabulary to LinkedTV (example) ontology based on indexed mappings retrieval

<table>
<tr><th rowspan="4">Session</th><th rowspan="4">Video</th><th rowspan="4">Shot</th><th rowspan="4">Length</th><th colspan="3">Semantic subvector based on a LOD vocabulary</th><th></th><th>Entities</th></tr>
<tr><th colspan="3">Place</th><th>...</th><th rowspan="3"></th></tr>
<tr><th rowspan="2">Place</th><th colspan="2">Civic Structure</th><th>...</th></tr>
<tr><th></th><th>Government Building</th><th>...</th></tr>
<tr><td>1</td><td>News1_ USFoot-ball</td><td>3</td><td>60</td><td>0.03</td><td>0.11</td><td>0.08</td><td>...</td><td>U.S. Gov-ernment, London, White house</td></tr>
<tr><th rowspan="4">Session</th><th rowspan="4">Video</th><th rowspan="4">Shot</th><th rowspan="4">Length</th><th colspan="3">Semantic subvector based on LUMO</th><th></th><th>Entities</th></tr>
<tr><th colspan="3">Location</th><th>...</th><th rowspan="3"></th></tr>
<tr><th rowspan="2"></th><th colspan="2">Edifice</th><th rowspan="2">...</th></tr>
<tr><th></th><th>Building</th></tr>
<tr><td>1</td><td>News1_ USFoot-ball</td><td>3</td><td>60</td><td>0.03</td><td>0.0825</td><td>0.064</td><td>...</td><td>U.S. Gov-ernment, London, White house</td></tr>
<tr><td colspan="4">Preference represen-tation for item News1_USFootball</td><td colspan="3">0.064 · Building(White House)
0.0825 · Edifice(X)
0.03 · Location(X)</td><td>...</td><td></td></tr>
</table>

4.1.4.3 Non-taxonomical relations propagation

We deem that further propagation of an extracted concept up the LinkedTV taxonomy is not required since the hierarchical propagation of the fuzzy high dimensional LOD representation should be enough to detect more general user preferences while maintaining the semantic user preference subvector per item dimensionally low (correspondingly, the low level description already deals with more general concepts and we expect that the social activity extraction will similarly return a compilation of more specific and more general concepts). In effect, we expect further concepts subsuming the user preferences to be derived in the concept filtering process.

However, in the interest of detecting relevant contextual information (e.g. topic detection), we might consider propagating base preferences along some non-taxonomical relations by activating (instantiating) some predefined object properties. For example, it might be useful to determine the topic/subtopic of the consumed media fragment to detect its general context(s). To this end, the dedicated “hasTopic” and “hasSubtopic” object properties (cf. Figure 5) can be activated in order to determine the connection between extracted entities and interest topics from LUMO.

Thus in the case of the ‘White House’ example, the activation of the ‘hasTopic’ property by a fabricated instance of the given media fragment will infer the general topic ‘Architecture’ for the fragment through appropriate reasoning services such as the reasoner described in D4.3:

Ontology rule (cf. Figure 7):	Building $\rightarrow$ $\forall$hasTopic.Architecture
In predicate logic:	hasTopic(X,Y), Building(X) $\rightarrow$ Architecture(Y)
Assertions:	Building(White House) : 0.064 , hasTopic(White House, mediafragmentID) (*mediafragmentID*: a fabricated instance connecting a given extracted individual to an induced instance of the whole fragment)
Entailment:	hasTopic(White House, mediaFragmentID), Building(White House) : 0.064 $\rightarrow$ Architecture(mediaFragmentID) : 0.064

Since ‘mediaFragmentID’ is a synthetic instance used to extract more general concepts of the particular media fragment, the inferred concept will be added/updated in the user profile as an abstract concept, such as 0.064 · Architecture(X).

In future work, we will also investigate the possibility to take into account the general context of the content item to improve the classification of media fragments to LUMO concepts. I.e. if the user selects a concept on the video, we can consider how the semantic description of the entire shot can influence the classified entities; if the user shows interest in a shot/scene, we can consider the semantic description of the shots/scenes within a wider temporal fragment of the video and so on.

It is easily inferred that if this is the first interaction of the user with content in the LinkedTV platform, these concepts will consist of the first (implicit) user preferences. The impact of the preferences will in parallel be influenced by the nature of the user’s specific actions as he interacts with the item, as will be described in the following section.

4.2 Semantic interpretation of user behaviour

The semantic interpretation of user behaviour depends on the type on the granularity of user feedback. At this analytics stage, we recognize two options:

a) user feedback is produced on video level.

b) user feedback is produced on scene or shot level.

4.2.1 User feedback produced on the video/video fragment level

The user can produce different kinds of implicit feedback during watching media content. There are 3 main categories of feedback:

- Basic player interactions – player controls.
- Additional interactions – extended player controls and additional content interactions.
- Physical interactions – based on tracking physical behavior. Will be described in the separate section.

Table 8: Basic player interactions

Name	*Properties*
Play	userId, mediaFragmentID, time
Pause	userId, mediaFragmentID, time
Skip	userId, mediaFragmentID, time
Fast-forward	userId, mediaFragmentID, startTime, stopTime
Rewind	userId, mediaFragmentID, startTime, stopTime
Jump to	userId, mediaFragmentID, time, jumpToTime

Table 9: Additional interactions

Name	*Properties*
Select object	userId, mediaFragmentID, time
Bookmark	userId, mediaFragmentID, time
View additional content	userId, mediaFragmentID, additionalContentID, time
Percentage Watched	userId, mediaFragmentID, time, percentage

Table 10: Physical interactions

Name	Properties
User leaves/come in the ROI	userId, mediaFragmentID, time
User sit down/stand	userId, mediaFragmentID, startTime, stopTime
User still/excited	userId, mediaFragmentID, startTime, stopTime
User looks to TV/looks in another direction	userId, mediaFragmentID, startTime, stopTime

A combination of these interactions on video level will express the level of user interest. Some of these interactions express positive interest and some of them negative. There are many approaches available, which can be used. We propose as a first approach to manually defined heuristic rules and learning using genetic algorithm that will be analysed in Section 4.3.

4.2.2 User feedback produced on scene or shot level

In this section, we will discuss the possibility to use scene, shots or triggerable objects (concepts) as individual instances of user preference. The advantage of such representation is that there are typically multiple scenes/shots per video, therefore more data are generated per video. However, not all scenes/shots can be used, since the following is required:

- semantically interpretable information on content
- information on user interest

The former was discussed in Sections 4.1.2 and 4.1.3. The latter involves interpretation of the level of user engagement to the content based on his transactional and physical behaviour.

4.2.2.1 Information on user interest

User transactional behavior tracking does not provide sufficient input to assess interestingness of each shot or scene for the user, it can be used to *annotate only a subset of scenes/shots, during which user interaction is recorded.*

Interestingness of virtually all scenes and shots can be conceivably determined from data provided by user physical behavior tracking. For physical behavior tracking we assume all features will be aggregated into one “interest” feature.

4.2.2.2 Examples

Table 11: Crisp version of shot based input. One user, two videos, each divided into three shots. For each shot, entities, which were extracted, are listed along with user interest level determined from user physical behavior tracking.

Session	Video	Shot	Length	Semantic subvector based on a LOD vocabulary								Entities	Interest level
				Place			Organization			Creative Work			
				Place	Civic Structure			Corporation	Sports Team		TV Series		
						Government Building							
1	News1_USFootball	1	28	Civic Structure	Government Building	**white house**	Sports Team		**football team**	TV Series	**Pro Football**	white house, X-Files, football team	low
1	News1_USFootball	2	14				Sports Team		**Red skin**			Red skin	med
1	News1_USFootball	3	60	**City**			**U.S. Government**					U.S. Government, city, White house	low
1	News1_USFootball	3	60	Government building	White house	Since two entities within one shot overlapped in their semantic description (both classified as “Place”), an extra line for the second entity (White house) was introduced.							low
2	News2_SecurityOlympics	1	20	**London**			**Olympics**					Olympics, London	low
2	News2_SecurityOlympics	2	10	Civic Structure	Government Building	**Thames House**	**MI5**					Thames House, MI5	med
2	News2_SecurityOlympics	3	5	Stadium			Sports Team		**Arsenal F.C.**			Stadium, Arsenal F.C.	high

Table 12: Crisp version of shot based input with user interest level determined from user transactional tracking. Only shots for which an event was generated are included.

Session	Video	Shot	Length	Semantic subvector based on a LOD vocabulary								Entities	Bookmark	Additional content view	Skip
				Place			Organization			Creative Work					
				Place	Civic Structure			Corporation	Sports Team		TV Series				
						Government Building									
1	News1_USFootball	2	14				Sports Team		**Red skin**			Red skin	y		
2	News2_SecurityOlympics	2	10	Civic Structure	Government Building	**Thames House**	**MI5**					Thames House, MI5		y	
2	News2_SecurityOlympics	3	5	Stadium			Sports Team		**Arsenal F.C.**			Stadium, Arsenal F.C.			y

Table 13: Fuzzy version of shot based input. One user, two videos, each divided into three shots. Results shown only for shot 3 video 1.

Session	Video	Shot	Length	Semantic subvector based on a LOD vocabulary								Entities	Interest level
				Place			Organization			Creative Work			
				Place	Civic Structure			Corporation	Sports Team		TV Series		
						Government Building							
1	News1_USFootball	3	60	0.03	0.11	0.08	0.92	0.28	0	0	0	U.S. Government, London, White house	0.2

4.3 Weighting: the impact of a preference

The impact of a preference will be depicted by a weight that will be assigned to each user preference. This weight will take into consideration a) the classification confidence of each consumed entity (section 4.1.1) and b) a modification value based on the concrete user behaviour towards that entity (in case of video/scene/shot more particularly several entities – cf. section 4.2). Although the consumed content will be classified into LUMO concepts, the behaviour of the user that will affect the preferences will be observed during user interaction, i.e. in parallel steps with the classification process. Therefore the described algorithms will affect firstly the LOD description extracted from the annotation and consequently the final classification.

4.3.1 The impact of a preference based on user behaviour

As aforementioned, we have manually defined a set of heuristic rules that will define the level of user interest/disinterest based on the transactional and physical actions a user performs. Our idea is to transform a set of interactions on the fragment level into one number that will express the level of interest. This number is in interval [-1, 1], positive values ([0, 1]) represent interest and negative values ([-1, 0)) represent disinterest.

Only rules with the highest priority are activated. The number is normalized after the contribution of all activated rules was summed up. If the sum is greater than 1 the result is 1, if the sum is smaller than -1, the result is -1.

Example 1:

- Actions: Bookmark (priority 1), Pause (Priority 3)
- Only bookmark rule is activated, result is interest = 1

Example 2

- Actions: (Play, Pause) – repeated 14 times, followed by fast forward at 600 seconds to 900 seconds, total duration 1200 seconds.
- All rules with priority 3 are activated.
    - Contribution of (Play ,Pause)x14 is 1.4
    - Contribution of fast forward is -(900 - 600)/1200=-0.25

Result is interest = 1.15 → 1.0

Table 14 lists an overview of the proposed heuristic transactional rules. Stop and start time refer to the physical time and not the playback time, with an interest to detect the duration of the action performed.

Table 14: Rules determining the impact of a transactional action to the user preference

Action	*Rule*	*Priority*
Bookmark	Interest = 1	1
View Additional Content	Interest = Interest + 0.2	2
Skip	Interest = Interest - (time to end - current time)/total time	3
Play	Interest = Interest + 0.1	3
Pause	Interest = Interest	3
Fast-forward	Interest = Interest – (stop time - start time)/total time	3
Rewind	Interest = Interest + (stop time - start time)/total time	3
Jump to	Interest = Interest - (stop time - start time)/total time	3

The behaviour detection module will provide the transactional tracking mechanism with features about user physical reactions. Depending on the features and their temporal evolution, it is more or less easy to deduce the level of user interest about the displayed content, so a weight should be given to the physical actions depending to their ability to describe user internal state. In the same way that the transactional tracking system will provide weights and information about the impact of player actions or clicks, the behavioural tracking mechanism will provide additional information on user involvement using features coming from the physical reaction detection module. Table 15 lists a set of proposed heuristic rules that will influence the impact of user preferences based on their physical reaction to the content. In future versions, the rule will depend on the physical reaction occurrence likelihood: higher occurrences mean that the reaction is usual and the rule weight decreases, while low occurrence show unusual or surprising reactions.

Table 15: Rules determining the impact of a physical reaction to the user preference

Action	*Rule*	*Priority*
User situated in an area of interest	Interest = 1	1
User sited	Interest = Interest + 0.5	3
Sudden change in sitting: user stan-	Interest = Interest – 0.3	5

Action	*Rule*	*Priority*
ding		
User looking towards the screen	Interest = Interest + 0.2	6
Sudden look towards the screen more than 5 seconds	Interest = Interest + 0.6	2
Sudden change in body motion (barycentre)	Interest = Interest + 0.4	4

Table 16: Example: the values in the Interest Level column were computed based on the heuristic impact estimation rules. The interest level is computed from actions listed in "User action subvector" using rules in Table 14. Eg. for video "stadt" the result 0.6=3x0.2 (rule View Additional Content => Interest = Interest + 0.2)

VIDEO NAME	SEMANTIC SUBVECTOR				USER ACTION SUBVECTOR			
	Precise Location	**Sports**	**Archi-tecture**	**…**	**Book-mark**	**Additional content viewed**	**% skip-ped**	**Interest level**
prozess	Berlin	0	0.8	…	1	3	**x**	**1**
stadt	Berlin	0	0		0	3	**x**	**0.6**
infanticide	Postdam	0	1.0	…	0	0	x	**0**
ratze	Berlin	1.0	0	…			1	**-1**

4.3.2 Learning using a genetic algorithm from training data

A second approach is based on learning the user interest level using a genetic algorithm (GA). The advantage of this approach is depicting the result as one value, which represents user interest. This approach is based on the application of symbolic regression on the training data. The training data are obtained during experiments, using questionnaires and values assigned by an expert. An example of training data can be seen in Table 17. Symbolic regression evolves an algebraic function, which will take into account available variables from the User Action Subvector (example in Table 17). Symbolic regression is based on GA. GA has a set of equations (individuals) and combines these equations during the evolution process to find better equations. Example of such an equation is *interest= 0.5 * Bookmark+0.1 * additional content viewed – skip*. The results of this equation for each row in Table 17 represent the interest level (as seen in Table 18). The quality of the evolved equation is defined as the fitness of this equation. Fitness is actually the reciprocal of the total sum of the difference between the ground truth value and the computed value. Fitness is higher if the results of the equation are more similar to the ground truth derived from the training data. The result of Symbolic regression is the best equation from a set of equations observed during the learning process and this equation will finally be used as the model to compute interest level. Example of differences between ground truth and the results of the learnt model is in Table 18. More details can be found in [KLI09, KUC10].

Table 17: The values in the last column were assigned by an expert. The GA uses these data to learn the (example) equation: *interest= 0.5 * Bookmark+0.1 * additional content viewed – skip.*

VIDEO NAME	SEMANTIC SUBVECTOR				USER ACTION SUBVECTOR			
	Precise Location	Sports	Archi-tecture	..	Book-mark	Additional content viewed	% skip-ped	Ground truth conver-sion level
prozess	Berlin	0	0.8	..	1	3	**x**	**0.9**
stadt	Berlin	0	0	.	**0**	**3**	**x**	**0.5**
infanticide	Postdam	0	1.0	..	**0**	**0**	**x**	**0**
ratze	Berlin	1.0	0	..			**1**	**-1**

Table 18: Fitness value computation example. Fitness denotes the quality of the solution. In this case, fitness is in reciprocal proportion to the model error on the training data. Model error is computed as the absolute value of the difference between the estimate and the ground truth.

VIDEO NAME	USER ACTION SUBVECTOR			FITNESS COMPUTATION		
	Bookmark	Additional content viewed	% skipped	Ground truth conversion level	Computed with GA formula	Error
prozess	1	3	x	0.9	0.8	0.1
stadt	0	3	x	0.5	0.3	0.2
infanticide	0	0	x	0	0	0
ratze			1	-1	-1	0
	Total error (negative fitness)					**0.3**

It should be noted that genetic algorithms generally do not guarantee to find the optimal solution. The quality of the model and its true error on unseen data can be verified using a separate validation dataset.

Table 19: Applying learnt model. The interest level is computed with the formula previously learnt (accepted as the best formula): ***interest= 0.5 * Bookmark+0.1 * additional content viewed – skip***

VIDEO NAME	**SEMANTIC SUBVECTOR**				**USER ACTION SUBVECTOR**			
	Precise Location	**Sports**	**Architecture**	**…**	**Bookmark**	**Additional content viewed**	**% skipped**	**Interest level**
Baltikum	Berlin	0.2	0.4	…	1	2	**x**	**0.7**
Wolf	Berlin	0	0.6	..	**0**	**3**	**x**	**0.3**

To foster processing of the resulting values of interest by association rule mining, and other algorithms expecting nominal rather than cardinal input, the Interest attribute can be discretized into a small number of intuitively understandable bins – refer to Table 20.

Table 20: Discretizing interest values

Raw interest value	Discretization bin
<-1;-0.7)	High negative disinterest
<-0.7;-0.4)	Medium negative disinterest
<-0.4;0.15)	Low negative disinterest
<-0.15;0.15>	Neutral
(0.15;0.4>	Low positive interest
(0.4;0.7>	Medium positive interest
(0.7;1.0>	High positive interest

4.3.3 User preferences and initial creation of the user model

The user model will consist of semantic information (concepts and instances from the reference UMO) based on the aggregation of implicit and explicit preferences and their association in the user's consumption history. It is thus understandable that the entities detected in the first transaction of the user with the platform will initialize the user model.

By combining the examples of Table 6 and Table 4, we identify the final set of initial user interests for shot #3 of News1_USFootball and its modification based on user interactions. Priorities between transactional and behavioural actions are treated separately, thus transactional and behavioural tracking information is aggregated.

Table 21: Interest estimation based on heuristic interest rules

Ses	Video	Sh	Len	Semantic subvector based on LUMO	Actions performed (priorities)			Entities
	News1_ USFootball	3	60		**Pause**	**Fast-Forward**	**Looking towards the screen**	
Item semantic representation				0.08 · Building(White House) 0.03 · Location (X) 0.03 · Location (London) 0.92 · Organization(U.S. Government)	3	3 (stop-start = 25min)	6	U.S. Government, London, White house
Preferences				Building(White House)	0.08	0.08-25/60= - 0.336	0.08 + 0.2= 0.28	-0.336 + 0.2 = -0.056
				Location(X)	0.03	- 0.386	0.23	-0.156
				Location (London)	0.03	- 0.0386	0.23	-0.156
				Organization(U.S. Government)	0.92	0.5	1 (normalized)	1 (normalized)

5 Learning user preferences

The task of implicitly learning user preferences involves the aggregation of the information in the user's consumption history over time. The learning mechanisms will have two main foci: a) adding new or updating preferred concepts and their preference weights based on their frequency, age and utility over consecutive consumptions and b) discovering persistent associating relations between concepts.

5.1 Updating weights

Whenever a new concept or instance from LUMO appears in the user's consumption history, it is added to the user's preferences. This preference will carry a weight that expresses the level of interest or disinterest of the user to the concept/instance based on the three aforementioned facets: the participation of the concept/instance in the content item consumed, the transactional and the reactional behaviour of the user towards the content item.

In order to update the preference weights of each concept/instance, two more dimensions of the user interaction will be stored for each entity. A counter depicting the frequency of appearance of the entity to the user consumption history and a timestamp of the last time this entity was accessed by the user. This will allow us to produce a holistic weight that expresses the impact (weight) of the preferences based on their frequency and age, updated over time and use of the platform. Furthermore, a generalised maximum frequency will be stored in order to retain the proportion when a new preference is added to the profile (i.e. a preferred concept might have appeared twice in the user's last 5 transactions but overall in the user history there are other concepts more frequently met, with the most frequent one appearing 276 times – therefore the new concept is not as important).

The age of the preferences is decayed over time based on the recency factor introduced in [STE12] for the damped window model defined as

$$decay = 2^{-\lambda(t - t_{u,c})},$$

where $\lambda \in [0,0.1]$ is the decay rate. The higher the λ, the lower the importance of past preferences compared to more recent ones is [STE12]. Variable t denotes the current time and $t_{u,c}$ denotes the last time the user u has accessed the concept c, i.e. the last timestamp for concept c.

The preference is then updated by the formula:

$$PW = \sum_i w \cdot \frac{f}{\max(f)} \cdot decay \ ,$$

where w is the weight with which the preference appeared in the last transaction, f is its frequency of appearance in the user's history and $max(f)$ is the frequency of the preference that has appeared more frequently than all the others in the user's history.

Table 22: Results of the PW function for current time = 11/09/2012 11:09:00 and λ = 0.01

Preferences	Entry weight*	Frequency	Time	PW
AmericanFootballTeam(Redskin)	0.65	4	2012-4-5 10:45 GMT	**0.155549268**
	0.97			
	0.46			
	0.73			
City(X)	0.89	4	2012-1-27 20:52 GMT	**0.112185478**
	1.0			
	0.86			
	0.51			
Stadium(X)	**0.95**	1	2012-9-9 17:38 GMT	**0.156446315**
SoccerTeam(ArsenalFC)	1.0	6	2012-9-9 17:38 GMT	**0.719653048**
	0.95			
	0.33			
	0.64			
	0.78			
	0.67			

**Note that only the sum is stored in the low-dimensional profile information. The appearance weights from the different instances of consumption of the entities are illustrated here for clarification purposes.*

Not analysing in depth the process of the contextualised user profiles, suffice it to say that, since separate instances of the user profile will be stored for each different contextual situation of the user, each contextual cluster of user preferences will carry different information on frequency and time. For example, in a contextualised instance concerning the context of leisure when the user watches sports on the TV, the entities AmricanFootballTeam(RedSkin), SoccerTeam(ArsenalFC) and Stadium(X) might have a higher f/max(f) ratio than in his long-term profile.

In the case of contextualised user models we might consider modifying the variable t to t_{last}, denoting the last time the user was in this contextual situation and conversely use the relation between t and t_{last} to determine the importance of this contextual situation to the user.

This weighting scheme will update the low level description of user preferences over time and content/concept consumption. The relevant features of the low level description (concept and weight) will be provided to the LFS simple filtering algorithm (cf. D4.3). A pruning threshold based on the preference weights will be statistically determined in order to delete the most obsolete preferences and minimize the profile size (e.g. preference weight below 0.001). It is also considered that a significance threshold will be similarly established in order to spare the filtering algorithms from excessive data overhead, so only the top-N semantically significant concepts might be passed along to the filterer (e.g. preference with a weight over 0.3). We will further examine the need to translate the simple preferences representation to a notation more suitable based on the requirements of the LFS (e.g. FOAF).

Table 23: An example of a low-level description of the user profile. Preferences and weights based on previous example. Additional example concepts inserted manually

Concepts/Instances	Weight
AmericanFootballTeam(Redskin)	0.155549268
City(X)	0.112185478
Stadium(X)	0.156446315
SoccerTeam(ArsenalFC)	0.719653048
Building(White House)	-0.008224342
Location (X)	0.013546674
Location (London)	-0.00034556
Organization(U.S. Government)	0.9853553553

The aggregated set of preferences (interests and disinterests) will also consist of the low-level model description that will be made available for subsequent preference learning.

5.2 The UTA method

Another suggested algorithm for learning user preferences based on their utility is the UTA method, introduced in D4.1.

The input for user profile learning is:

a) low-dimensional representation of videos (cf. section 4.1.1) the user has seen,
b) user interest level for each video (cf. section 4.2)

Table 24: Example input for learning – description of videos seen by given user

Video	Precise Location	Cluster 1/Sports	Cluster 2/ Architec-ture	….	Interest level
Baltikum	Berlin	0.2	0.4		0.7
Wolf	Berlin	0	0.6		0.3

Based on this imput, utility functions are computed for each low-dimensional descriptive concept as seen in the fabricated example of Figure 21

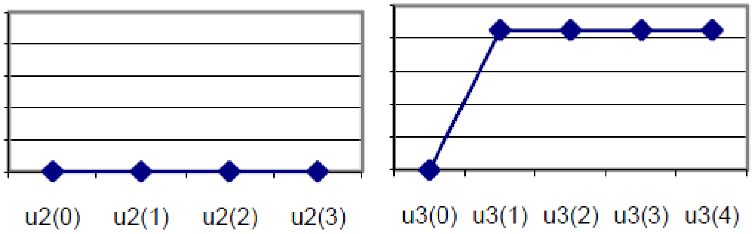

Figure 21: Result of UTA method (fabricated): utility function for sports – left, for architecture – right.

These utility functions are used to rank additional content items based on their low-dimensional description. The UTA method is on the borderline of D4.2 and D4.3: learning the preferences falls within the scope of this deliverable, while their application for content ranking might be more in scope of D4.3. However in the interest of keeping the end-to-end methodology undivided, the description of the use of UTA method for content ranking is described in the present deliverable, but placed into Appendix II so that the consistency of the document is preserved.

5.3 Associating concepts in the user profiles

5.3.1 Association rule mining

Association rule mining provides a different type of output than classification algorithms such as UTA. While UTA always assigns a decision (i.e. assigns a utility value) to a test instance provided that the values of features of the test instance are within bounds seen in the training data, association rules do not provide such complex view; they cover the space only partially.

This can be however seen also as an advantage when the problem space is large and/or it is sparsely covered by training instances. Before giving details directly on association rule mining in LinkedTV, let us review its role in context of the UTA method.

- The UTA method imposes very strong inductive bias, therefore it allows to learn from smaller number of instances. Nevertheless, not to undermine its performance, the input data representation should also be low dimensional. To this end, we suggested in 4.1.1.2 to use multi-class "genre" classification or clustering. The limitation of this approach is that it will not learn subtleties of user interest
    - simultaneous conditions imposed on multiple features (e.g. the user is interested in sport but only if the place is Berlin)
    - fine-grained conditions (the user is interested in football rather than in all sports)

As an alternative to the high dimensional bag of words representation created from the ASR and metadata, a vector created from semantic (taxonomy-based) description of entities extracted from the video can be used, such as the one described in the previous section. Association rules seem as a suitable technique for a user with such a high-dimensional vector. Nevertheless, even association rule mining is based on statistical interest measures, therefore a sufficient number of training instances is necessary. There are several possibilities for increasing the number of training instances:

- As described in section 4.1.2 the number of training instances can be increased by using shots or scenes as training instances rather than entire videos. Only such scene/shot will be used, which contains at least one recognized entity and has physical user behavior feedback available.
- Use physical tracking to obtain more fragments per video with available user interest
- Use videos for all users rather than for a particular user.
    - Even this setting allows personalization with *conditional association rules* – the rule applies to users sharing one or more identical characteristics, such as stereotype

Concerning the input data, association rule mining algorithm are in general:

- unable to natively work with cardinal features.
- very good in dealing with missing data. If a particular feature-value pair (an item in the association rule mining terminology) is missing for a given training instance, the association rule mining algorithm exploits this to reduce the complexity of the mining task.

From this follows, that for the input representation the crisp version is favoured over the fuzzy one (refer to section 4.2.1).

5.3.1.1 Example

Association rule learning applied on all users' data as an input will generate rules, which associate characteristics of the entities displayed in a particular video fragment with user interest level.

We consider GUHA 4ft association rule mining procedure implemented in the LISp-Miner system[34]. The definition of the mining task encompasses particularly the specification of:

- Attributes that can appear in the antecedent, consequent and condition part of the rule.
- Coefficient type for each attribute. The basic coefficient type is “One category” which results in the system considering each single value for the attribute. Advanced coefficients, such as “interval” or subset allow for dynamic binning of values. The use of these coefficients is recommended if there are data sparsity issues.
- Classes of equivalence. This feature of the LISp-Miner system allows to group semantically related attributes. The effect of this setting is that a rule will not contain two attributes from the same class of equivalence.
- Choice of interest measures. The interest measures determine the strength as well as the semantics of the discovered rules. Interest measure is always associated with a minimum value (threshold) for a rule to appear on the output. Examples of well-known interest measures include support, confidence and lift. The LISp-Miner system implements a range of other measures corresponding to well-known statistical tests, such as Chi-Square of Kendall.

Consider the following example setting:

- Attributes for antecedent: (UMO classes: Location, Building, Organization, SportsTeam, Creative Work, Series); Attributes for the consequent: Interest; Attributes for the condition: none
- Coefficient type: one value one category for all attributes
- Classes of equivalence: three classes Location ={Location, Building}, Organization={ Organization, Sports Team}, Creative Work={ Creative Work, Series }
- Choice of interest measures: support 0.05, confidence 0.7

Example rules generated from input which was exemplified in Table 11 are:

1. `Sports Team(Arsenal F.C.) => Interest(high)`

   `Support = 0.142, Confidence = 1.0`

2. `NOT Organization(Sports Team) => Interest(low)`

   `Support = 0.43, Confidence = 0.75`

3. `Location(Building) and Organization(Sports Team) => Interest (low)`

   `Support = 0.142, Confidence = 1.0`

[34] lispminer.vse.cz

**Note that the notation used to exemplify association rules follows the attribute-value pair notation as described in [RAU05].*

Let us investigate the second rule `NOT Organization(Sports Team) => Interest(low)` in detail with respect to its compliance with selected interest measure thresholds.

Support of this rule is the number of objects matching all parts of the rule divided by the total number of objects. In this example, both parts of the rule match 3 rows (objects) in Table 11, the total number of objects is 7, thus we obtain 3/7=0.43

Confidence is computed as the ratio of number of objects meeting the antecedent and consequent of the rule to the number of objects meeting the antecedent of the rule. In the example rule, out of the four objects matching the left side of the rule `NOT Organization(Sports Team)`, three objects match the right side of the rule `Interest(low)` as well. This gives 3/4=0.75

5.3.1.2 Deployment

The application (deployment) of this rule to assess interestingness of a particular additional content item requires the following steps:

1. Entity recognition and classification system is run on the textual representation of the linked content
2. The entities are expanded to the semantic representation
3. The rules are matched with the semantic representation of the available additional content items:
    a. Items matching rules with low interest (disinterest) in the consequent are discarded
    b. Items matching rules with positive interest level in the consequent are kept and assigned priority according to the level of interest (e.g. high → priority=2, medium → priority = 1)
    c. Items not matching any rules are kept, but assigned priority=0
    d. If multiple rules apply to the same content item, only the longer rule applies

If multiple rules of the same length with different consequent apply to the same content item, conflict resolution is executed. A simple conflict resolution requires averaging the priority associated with the consequents.

5.3.1.3 Conditional association rules

The input for this learning process contains in addition to the semantic information relating to the content also description of the user viewing the content, such as the stereotype or location.

This option is similar to the previous one, however the rules can now consist of three parts. In addition to antecedent and consequent separated by => symbol, the consequent can optionally be followed by a condition delimited by "/" symbol.

Supposing that we would run the same task as in 5.3.1.1 on data of multiple users, their interest would not likely be as unanimous. However, there would still likely be commonalities to be found e.g. based on user location. In Table 25 we present a toy example of such a dataset, where we have two users, each from a different location. The two users watched the same content items (described in Table 11), but their interest level differ.

The mining task would not clearly produce as strong rules in terms of confidence as listed in 5.3.1.1. For example, the confidence of the first listed rule

`SportsTeam(Arsenal F.C.) => Interest(high)`

would drop to 0.5, since we have two rows with `SportsTeam(Arsenal F.C.)` in Table 25 but only one has `Interest(high)`.

However, the different characteristics of the users can be factored in by adding the condition part to the association rule mining task setting.

- Attributes for condition: Location, Gender

As a result of this setting, example conditional association rules generated include:

```
1. SportsTeam(Arsenal F.C.) => Interest(high)/UserLocation(Berlin)
        Support = 0.142, Confidence = 1.0
2.  SportsTeam(ArsenalF.C.) => Interest(low)/UserLocation(Vienna)
        Support = 0.142, Confidence = 1.0
```

Example input data for conditional association rule generation are depicted in Table 25.

Table 25: Input data for learning conditional association rules. There are two users viewing the same two news items in a row, the users differ in location and interest levels.

User ID	Session	Video	Shot	Length	User Profile subvector			Semantic sub-vector	Entities	Interest level
					Location	Gender	…	…		
1	1	News 1	1	28	Berlin	Male			white house, X-Files, football team	low
1	1	News 1	2	14	Berlin	Male			Red skin	med
1	1	News 1	3	60	Berlin	Male			U.S. Government, city, White house	low
1	1	News 1	3	60	Berlin	Male				low
1	2	News	1	20	Berlin	Male			Olympics, London	low

		2								
1	2	News 2	2	10	Berlin	Male			Thames House, MI5	med
1	2	News 2	3	5	Berlin	Male			Stadium, Arsenal F.C.	high
2	1	News 1	1	28	Vienna	Male			white house, X-Files, football team	low
2	1	News 1	2	14	Vienna	Male			Red skin	low
2	1	News 1	3	60	Vienna	Male			U.S. Government, city, White house	low
2	1	News 1	3	60	Vienna	Male				low
2	2	News 2	1	20	Vienna	Male			Olympics, London	medium
2	2	News 2	2	10	Vienna	Male			Thames House, MI5	low
2	2	News 2	3	5	Vienna	Male			Stadium, Arsenal F.C.	low

5.3.2 Semantic interpretation of association rules

The semantic interpretation of association rules would comprise translating the rule into a DL axiom with the corresponding constructs, where the body contains the complex relationship and the head would contain an interpretation of the rule's impact by attributing an impact weight to the rule. Since the head of the rule represents the level of interest of the user to the associated concepts and a confidence degree for which each rule holds, the impact weight will consist of the product of the two scores (the interest score in a low-medium-high range will be interpreted as {0.2, 0.5, 0.8}). Rules containing only complements in their body will be negated in order to produce a disinterest rule (the use of disinterest rules will be explained in section 6).

Thus the interpretation (in pseudo-DL notation) of the examples of the previous section will be:

- `Sports Team(Arsenal F.C.) => Interest(high) -- Confidence = 1.0`
    - `Sports Team(Arsenal F.C.) ⊑ InterestRule1 · 0.8`
- `NOT Organization(Sports team) => Interest(low) -- Confidence = 0.75`
    - `Organization(Sports team) => Disintest(High)`
        - `Sports Team ⊑ DisintestRule1 · 0.6`

- `Location(Building) and Organization(Sports Team) => Interest (low) -- Confidence 1.0`
  - Building ⊓ Sports Team ⊑ InterestRule2 · 0.2

Conditional association rules will be treated the same, with the only difference that the rule concept will require of two concepts to be simultaneously true for the rule to apply such that:

- `SportsTeam(Arsenal F.C.) => Interest(high)/UserLocation(Berlin), Confidence = 1.0`
  - SportsTeam(Arsenal F.C.) ⊑ InterestSubRule1 · 0.8
  - InterestSubRule1 · 0.8 ⊓ UserLocation(Berlin) · 1.0 ⊑ InterestRule1 · 0.8
- `SportsTeam(ArsenalF.C.) => Interest(low)/UserLocation(Vienna), Confidence = 1.0`
  - SportsTeam(Arsenal F.C.) ⊑ InterestSubRule2 · 0.2
  - InterestSubRule1 · 0.2 ⊓ UserLocation(Vienna) · 1.0 ⊑ InterestRule1 · 0.2

Such conditional rules when the condition is a contextual information about the user might also be used as preference only in a given contextual instance of the user model. The weighting of conditional rules will be further investigated in the future. It is conceived that they might be better defined by means of applying thresholds above which a statement holds, or require the weighted sum of the conditions that make up the preference rule.

6 Modelling user interests

In the interest of expressing the user model in manner that can fully take advantage of the semantic information extracted by the techniques that were previously presented in this deliverable and at the same time facilitate its synergy with the ontology-aware filtering algorithms presented in D4.3, a lightweight ontological schema will be employed. Consequently, the user profile (and the different contextual instances) will be expressed in logical axioms within the DLP expressivity fragment, as described in D4.1:

(1) $\bigcup_n \exists \mathrm{has}\,Preference.Preference_{i\in n} \rightarrow Interests$,

(2) $\bigcup_n \exists \mathrm{has}\,Preference.Preference_{i\in n} \rightarrow Disinterests$,

such that

(3) $Interests \bigcap Disinterests \subseteq \perp$.

(1) denotes the disjunction of all preferences that are interesting to the user, (2) denotes the disjunction of all preferences that are uninteresting to the user (the user rejects them) and (3) denotes that interests and disinterest are disjoint. The latter indicates that when the profile is found to be satisfied (e.g. by a candidate content item to be recommended), if at the same time a specific concepts is found that indicates a disinterest, then the candidate should be rejected.

DLP is the expressive intersection between Description Logics and Logic Programming. It provides the means to "build rules on top of ontologies" [GRO03] and is much more expressive than a simple taxonomy (RDFS subclass relations) and FOAF interests. The benefits of this expressivity fragment lie on the inclusion of more complex information in the user profile such as rules but still retaining the complexity of the model on a relatively low level, i.e. in a subfragment of the full DL expressivity. The interests/disinterests may consist of instances and primitive concepts from LUMO but also by complex concepts (stemming primarily from association rules). Such complex concepts might include association between concepts via boolean constructors, association between concepts via disjointness but also restriction of a concept to a specific property (again from LUMO) and association between properties via inclusion, transitivity and symmetry.

Preferences with negative weights will be induced to the disinterest (¬Profile) with an absolute weight, thus activating the detection of inconsistencies based on formula (3). An example of how formula (3) affects concept and content filtering is described in the semantic reasoner description of D4.3. As association rules will begin to be introduced, the profile will consist of a disjunction between weighted primitive concepts and weighted complex concepts that. An example of the user model in various iterations is illustrated below.

The proposed modelling schema and expressivity fragment are deemed as the most complete and at the same time most lightweight method to convey existential/universal restrictions and boolean constructors (e.g. derived from association rules), disjointness and negation (e.g. to express disinterests) and fuzzy logic (uncertainty in content annotation, i.e. degrees, preference weights and possibly thresholds derived from association rules).

6.1.1 Example of the creation and update of a user model

Following the preference extraction approaches discussed in sections 3 to 5, an example of the creation and adaptation of a user model, initiating from the first appearance of a new user to the LinkedTV platform up to learning the most complex information about his/her preferences through time.

Step 1: new user

⊤ ⊑ Interests

⊥ ⊑ Disinterests

Step 2: one content item consumption

Preferences from the example in Table 21.

1.0· Organization(U.S. Government) ⊑ Interests

0.056·Building(White House) ⊔ 0.156·Location(X) ⊔ 0.156· Location(London) ⊑ Disinterests

Step 3: few transactions in user's history, associations not yet formed

Preferences from the example in Table 23.

0.155549268· AmericanFootballTeam(Redskin) ⊔ 0.112185478· City(X) ⊔ 0.156446315· Stadium(X) ⊔ 0.719653048· SoccerTeam(ArsenalFC) ⊔ 0.013546674· Location (X) ⊔ 0.9853553553· Organization(U.S. Government) ⊑ Interests

0.008224342· Building(White House) ⊔ 0.00034556· Location (London) ⊑ Disinterests

Step 4: several transactions in user's history, associations rules induced

Preferences combining the previous example and the association rules of section 5.3.1. Crisp concepts are translated as having weight ≥ 1.0.

0.2 · InterestRule2 ⊔ 0.112185478· City(X) ⊔ 0.156446315· Stadium(X) ⊔ 0.013546674· Location (X) ⊔ 0.9853553553· Organization(U.S. Government) ⊔ 0.8 · InterestRule1 ⊔ 0.8 · InterestRule3 ⊔ 0.2 · InterestRule4⊑ Interests

Building ⊓ Sports Team ⊑ InterestRule2 · 0.2

SportsTeam(Arsenal F.C.) ⊑ InterestSubRule1

InterestSubRule1 ⊓ UserLocation(Berlin) ⊑ InterestRule3 · 0.8

SportsTeam(Arsenal F.C.) ⊑ InterestSubRule2

InterestSubRule1 ⊓ UserLocation(Vienna) · 1.0 ⊑ InterestRule4 · 0.2

0.008224342· Building(White House) ⊔ 0.00034556· Location (London) ⊔ 0.6 · DisintestRule1⊑ Disinterests

SportsTeam ⊑ DisintestRule1 · 0.6

7 Storage and communication

7.1 Transactional Tracking

The transactional tracking mechanism uses a password protected NoSQL database for storage of all collected information. The most granular data about user behavior are stored only temporarily. For the purpose of analysis, only aggregated (per user) transactional data are stored for longer time. The mechanism is accessible using the provided REST API. The REST API supports collecting data and exporting results of analysis. The communication using REST API is secured using HTTPS and protected using HMAC (Hash-based Message Authentication Code) – this mechanism allows signing messages based on a shared secret between the client and the service. Only verified clients can communicate with provided REST API.

7.2 Behavioral Tracking

The behavioural tracking module provides an averaged feature evolution along time. Those features are low-level features which do not convey any sensitive information about the users as described in section 3.2. They are stored during short periods of time (some minutes) and then deleted. Only higher level information obtained after low-level processing is stored for a longer period of time into the user profile.

As an example, the body barycenter motion is extracted and this average low-level feature is stored during 10 minutes of viewing. This feature is then used in a classifier which outputs the level of interest that the content has depending on this feature evolution. The statistics of the classifier outputs show that the user is most of the time neutral except during political debates and reality show content. This latter higher-level information only is stored in the user profile.

The user tracking is performed on client side. The Kinect data flow is analyzed locally and only averaged low-level feature are sent to the transactional tracking system which is server-side through a network protocol.

7.3 UTA and Association Rule mining

7.3.1 Conceptual description of data input and output

The data input for training the UTA method as well as learning association rules consists of one data table, where rows correspond to content items. An example input is listed in Table 19. The difference between the UTA method and association rule mining with respect to the input is that for association rule mining, the input data table can contain content items viewed by multiple users (processing such input was discussed in Section 5.2.1.3), while for UTA method there should be one data table per user.

As what concerns applying the set of utility curves, learnt by the UTA method, or the association rules, the input is again one data table with rows corresponding to content items, that can be recommended to the user. Compared to the data table used for training, the column(s) corresponding to the user interest level are missing.

Data mining task description

A) Association Rule mining

The data mining task is described in a variant of the Predictive Modeling Markup Language (PMML). This is an XML-based format, which has the following structure (abridged):

- Data Dictionary: identifies and describes the input data table
- Transformation Dictionary: defines transformation on the input fields. For example, the *age_in_years* data field containing discrete values in the range <0;99> can be transformed to *Age* derived fields containing only three values: *child*, *adult*, *senior*. For example, the *adult* value is defined to span the range of <18;65) of the input data field
- Mining model: describes the setting of the mining algorithm, and if the algorithm was already run, the results. Since the LISp-Miner mining software, which is planned to be used in LinkedTV, offers multiple features not covered by the latest PMML standard, it uses its own mining model called GUHA AR PMML [KLI10].

The output of the association rule learning process is a PMML file containing the learnt rules in the Mining Model part. These rules can then be applied to new content items, with preference level being assigned to matching content items.

A) Preference learning

There is not yet any standardized XML format for preference learning tasks. Our UTA implementation accepts a custom XML-format. This format has the following parts:

- Task Setting: contains technical parameters of the learning algorithm
- Criteria: description of the input data fields; this is an analogy to the dictionaries in PMML
- Alternatives: description of content items. This is the input data table embedded in the XML file
- Stated preferences: the preference information; each alternative is assigned a preference (interest) level

The output of the algorithm is an XML file describing the learnt model. The model can be used to assign preference level to new content items. Since the UTA method is intended to be operated only on a single-user input with low-dimensional semantic representation, one XML file is used to define the parameters of the task and the input data. In contrast, for association rule mining input data are provided separate from the PMML document describing the task.

7.3.2 Communication workflow

Both LISp-Miner and the UTA method will be available through web services.

For UTA method the communication is simple, since both the input and output is constituted by one self-contained XML file. Also, since UTA processes small number of input, it is feasible that the caller waits for the response.

The web service interface for UTA method has not yet been established, therefore it is not possible to provide more detailed description at this stage.

Concerning the LISp-Miner system, the communication follows a more complicated protocol. The reason is that

- the task description and data come separately: task description in PMML, and data are provided as a connection string to MySQL database
- due to higher volume and dimensionality of the data, the processing may take long time. For this reason, on registering a new mining task, the calling application is passed a handle. This handle is then used to poll for task status (finished, running, interrupted), stop the task, and obtain the results (in the form of PMML).

The web API is described at http://connect.lmcloud.vse.cz/

The APIs for both LISp-Miner and UTA are (or will be in case of UTA) accessible via the HTTP protocol.

7.4 The user model

Low dimensional user preferences will be stored in a lightweight text format (.txt or .xml file), consisting only of the names of entities from LUMO (not even URI), the last timestamp of the interaction that the user has made with the concept, a counter representing the frequency of the concept in the user's transaction history, a global constant denoting the maximum frequency that a preference has in the profile and the current preference weight of the concept. It will in the initial stages of the project reside on the server, but the aim is to store the preferences locally, given also the considerations made for pruning the profile so that it doesn't reach an unmanageable size. When on the client, communication with the server will depend on the requests of the more processing-demanding learning and filtering algorithms that request it.

The axiomatic user model and axiomatic contextualized instances will consist of a reduced size of the user preferences full spectrum, again identified by a statistically determined threshold. It will employ a variant of the KRSS2[35] representation language. The main advantage of the KRSS2 vocabulary is that it is significantly more lightweight than other languages (expected >40% size reduction). Furthermore, it is parseable by several reasoners, including the

[35] http://dl.kr.org/krss-spec.ps : the first version (KRSS) syntax. KRSS2 supports additional DL complian semantics as stated in this public communication: https://mailman.stanford.edu/pipermail/p4-feedback/2008-August/001080.html , but no documentation of the additional semantics is currently available

popular commercial Racer[36] reasoner and the f-pocketkrhyper reasoner which will be used in LinkedTV and is introduced in D4.3. It is also parseable and exportable by the Protégé[37] ontology editor, thus alleviating the interoperability between any chosen vocabulary that the ontology will be expressed in, or FOAF-expressed simple interests (e.g. in Beancounter) and the more complex user profile. The lightweight representation again aims to subsequently store (and even enable filtering, provided reference knowledge base reduction – cf. section 8) on the client.

[36] http://www.sts.tu-harburg.de/~r.f.moeller/racer/

[37] http://protege.stanford.edu/

8 Conclusions and future work

This deliverables describes an end-to-end approach for obtaining, updating and serializing a semantic user profile based on implicit information about the user, as these are imprinted from the user's transaction history within the LinkedTV platform. The process involves determining the appropriate background knowledge, semantically interpreting user transactions based on this knowledge base, interpreting the level of interest/disinterest for a given preference based on the user's transactional and reactional behaviour and learning a user model through iterative adaptation of the user profile over time.

In effect, a semantic schema for the user model was defined, with an interest in maintaining the size and complexity of the model as low but at the same time as complete and meaningful as possible. This schema further enables predictive inference via advanced inference engines such as semantic reasoners on top of simpler and faster filtering algorithms such as spreading activation variants and utility functions that take into account just the weighed preferences, while still being able to provide input to the simpler recommenders.

Future work will be oriented towards versioning the LinkedTV UMO ontology and arriving at an ontology that fully complies with the established design principles. Furthermore, we will investigate appropriate probabilistic methods to infer the impact of the preferences in a given transaction of the user with the system based on his behaviour to improve the current heuristic methodology.

The modelling task will also effectively extend on determining the contextual parameters of the user and adapting the long-term user model to different contextual situations. We will establish the appropriate methodology for clustering users and in extent obtain knowledge about user clusters and relate the clusters with some contextual attributes of the user.

Work in the extended future will also turn towards minimizing the volume of background knowledge that will be active at each inference session based on user context, since in a multidiscipline domain such as digital media real-time inferencing is still hampered due to the unnecessarily high complexity that the large concept space introduces, i.e. a large terminological box that has no relation to the domain or the situation of the user at a given session, however light and compact the reference knowledge might be.

Therefore, we will further explore and foster manifold user knowledge acquisition and adaption techniques, such as pulling subsets of (domain) knowledge based on user context to reduce the dimensionality of background knowledge, in order to facilitate intelligent and privacy-preserving concept and digital content recommendation. The approach will aim to understand the extended context during a concrete viewing/browsing condition of the user based on content annotation and other contextual factors (e.g. geo-temporal information) and pull the appropriate sub-ontology that is needed for inferencing at a given situation without having to transmit back to the server specific information about the user.

In addition, we will investigate methods to expand the platform's capabilities by extending reference knowledge through automatically learning group-specific knowledge based on aggregated user information (e.g. through user clustering) and adapting new information to the initial knowledge, e.g. via learning persistent association rules within clusters of similar users.

9 Bibliography

[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.

[BLA05] Y. Blanco, J. J. Pazos, A. Gil, M. Ramos, A. Fernández, R. P. Díaz, M. López, and B. Barragáns. "AVATAR: an approach based on semantic reasoning to recommend personalized TV programs". Proc. Special interest tracks and posters of the 14th international conference on World Wide Web (WWW '05), 2005. ACM, New York, NY, USA, pp. 1078-1079.

[BLA08] Y. Blanco, J. J. Pazos, A. Gil, M. Ramos, A. Fernández, R. P. Díaz, M. López, and J. Garcia. "Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: A case study". Journal of Systems and Software, 81, 12, December 2008, pp. 2371-2385.

[BOL07] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge". Proc. 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08), 2008. ACM, New York, NY, USA, pp. 1247-1250.

[DAM10] M. Damova, A. Kiryakov, K. Simov, S. Petrov. "Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology". Proc. 5th International Workshop on Ontology Matching at ISWC 2010 Shanghai, China, November 2010.

[DAV06] J. David, F. Guillet, and H. Briand. "Matching directories and OWL ontologies with AROMA". Proc. 15th ACM International Conference on Information and Knowledge Management (CIKM '06), New York, NY, USA, ACM, 2006, pp. 830–831.

[GIU05] F. Giunchiglia, P. Shvaiko, M. Yatskevich. "S-Match: an algorithm and an implementation of semantic matching". Semantic Interoperability and Integration. Number 04391 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005.

[GKG12] "Introducing the Knowledge Graph: things, not strings". http://googleblog.blogspot.gr/2012/05/introducing-knowledge-graph-things-not.html. Last accessed on 14th June 2012.

[GRO03] Grosof, B. N., Horrocks, I., Volz, R., & Decker, S. (2003). Description logic programs: combining logic programs with description logic. In WWW '03: Proceedings of the 12th international conference on World Wide Web , (pp. 48-57). New York, NY, USA: ACM.

[HEC05] D. Heckmann, B. Brandherm, M. Schmitz, T. Schwartz, B. M. von Wilamowitz-Moellendorf, B.M.. "Gumo - the general user model ontology". In User Modeling, Edinburgh, Scotland, 2005, pp. 428-432.

[JAI10a] P. Jain, P. Hitzler, P. Z. Yeh, K. Verma, and A. P. Sheth. "Linked Data is Merely More Data". In AAAI Spring Symposium 'Linked Data Meets Artificial Intelligence', 2010.

[JAI10b] P. Jain, P. Hitzler, A. P. Sheth, K. Verma, and P. Z. Yeh. "Ontology alignment for linked open data". Proc. 9th international semantic web conference on The semantic web - Volume Part I (ISWC'10), 2010, Vol. Part I. Springer-Verlag, Berlin, Heidelberg, pp. 402-417.

[JAI11] P. Jain, P. Z. Yeh, K. Verma, R. G. Vasquez, M. Damova, P. Hitzler, and A. P. Sheth. "Contextual ontology alignment of LOD with an upper ontology: a case study with proton". Proc. 8th extended semantic web conference on The semantic web, Volume Part I (ESWC'11), 2011, Vol. Part I. Springer-Verlag, Berlin, Heidelberg, pp. 80-92.

[KIN10] Microsoft. 2010. Microsoft's kinect, http://www.xbox.com/kinect.

[KLI09] Tomáš Kliegr. Representation and dimensionality reduction of semantically enriched clickstreams. *In Proceedings of the 2008 EDBT Ph.D. workshop*, Ph.D. '08, pp.29-38. ACM, New York, NY, USA, 2008.

[KLI10] Tomáš Kliegr, Jan Rauch: An XML Format for Association Rule Models Based on the GUHA Method. RuleML'10: pp. 273-288. Springer-Verlag Berlin. 2010

[KUC10] Jaroslav Kuchař, Ivan Jelínek. Scoring Pageview Based on Learning Weight Function. *In International Journal on Information Technologies and Security*, 2(4):19{28, November 2010.

[MIL95] G. A. Miller. "WordNet: A Lexical Database for English". Communications of the Association for Computing Machinery, 38(11), 1995, pp. 39-41.

[RAU05] Jan Rauch. 2005. Logic of Association Rules. Applied Intelligence 22, 1 (January 2005), 9-28.

[RES98] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*. pages 448-453. 1998.

[RIZ11] G. Rizzo and R. Troncy. "Nerd: evaluating named entity recognition tools in the web of data". In ISWC'11, Workshop on Web Scale Knowledge Extraction (WEKEX'11), October 23-27, 2011, Bonn, Germany.

[STE12] Kostas Stefanidis, Irene Ntoutsi, Kjetil Nørvåg, and Hans-Peter Kriegel A Framework for Time-Aware Recommendations. In: DEXA 2, Vol. 7447, Springer (2012) , p. 329-344.

[SUC07] F. M. Suchanek, G. Kasneci, and G. Weikum. "Yago: a core of semantic knowledge". Proc. 16th international conference on World Wide Web (WWW '07) 2007, ACM, New York, NY, USA, pp. 697-706.

[TSA09] D. Tsatsou, F. Menemenis, I. Kompatsiaris, P. C. Davis. "A semantic framework for personalised ad recommendation based on advanced textual analysis". ACM International conference in recommender systems (RecSys09), New York, USA, 2009, pp. 217-220.

[TSA12] Dorothea Tsatsou, Vasileios Mezaris and Ioannis Kompatsiaris. Semantic personalisation in networked media: determining the background knowledge. In 7th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP 2012), December 3-4, 2012, Luxembourg.

[TURK91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3 (1): 71–86. doi:10.1162/jocn.1991.3.1.71, 1991.

10 APPENDIX I: Examples of REST services for transactional tracking

The following tables depict simple description of an exemplary REST service and provides information about HTTP methods, URIs, data formats etc. All these information will be integrated with the LinkedTV platform.

Table 26: Description of REST service used for tracking interactions

Track interaction	
Description	POST /listener
HTTP Method	POST
Content-Type	application/json
URI	http://wa.vse.cz/listener
cURL	curl -v --data @data.json http://wa.vse.cz/listener

data.json	{
	 "accountId" : "LTV-0-1",
	 "client" : {
	 "type" : "ltv-player",
	 "version" : "0.1"
	 },
	 "object" : {
	 "id" : "2453132",
	 "title" : "News 1",
	 "uri": "/news1?a=20&b=30"
	 },
	 "user" : {
	 "id" : "58"
	 },
	 "interaction" : {
	 "type" : "event"
	 },

	"attributes" : {
	"event" : {
	"category" : "Video",
	"action" : "Pause"
	},
	"variables" : [
	{
	"slot": 0,
	"name": "category",
	"value": "news"
	},
	{
	"slot": 0,
	"name": "topic",
	"value": "sport"
	}
	]
	}
	}

Status codes	201 – Created 400 – Bad request

Table 27: Description of REST service used to get aggregated stats: number of interactions

Number of requests	
Description	GET /api/{accountId}/interactionsCount
HTTP Method	GET
Content-Type	application/json
URI	http://wa.vse.cz/api/{accountId}/interactionsCount
cURL	curl -v http://wa.vse.cz/api/LTV-0-1/interactionsCount
Example of re-	{

sponse	count: 23 }
Status codes	200 – OK 400 – Bad request

Table 28: Description of REST service used to get latest stored interactions

Latest interactions	
Description	GET /api/{accountId}/latestInteractions
HTTP Method	GET
Content-Type	application/json
URI	http://wa.vse.cz/api/{accountId}/latestInteractions
cURL	curl -v http://wa.vse.cz/api/LTV-0-1/latestInteractions

Example of response	[
	{
	"accountId":"LTV-0-1",
	"_id":"4ffbf76c9a8271e63e0004ab",
	"customVars":[
	{"name":"category","value":"news"},
	{"name":"topic","value":"sport"}
	],
	"events":[
	{"category":"Video","action":"Pause"}
	],
	"resource":{"uri":"/news1?a=20&b=30"},
	"date":"2012-07-10T09:35:40.646Z"
	}
	]

Status codes	200 – OK 400 – Bad request

11 APPENDIX II: Ranking content based on the UTA Method

The input for user profile application is depicted on Table 29. However, other profile learning algorithms may be applied on this input. The result of ranked additional content based on the UTA method and this input is depicted in Table 30.

Table 29: Example input for application – description of content items suggested for video "prozess" using the low-dimensional representation of the additional content items with clusters defined for videos (cf. Section 4.1.1.3).

Additional content item	Precise Location	Cluster 1/Sports	Cluster 2/Architecture	…
http://www.rbb-online.de/nachrichten/...	Berlin	0.4	0	
de.Wikipedia.org/prozess_gegen_…...	-	0	0.6	

Table 30: The result of application of the learnt user model is a ranking of the candidate additional content items with respect to the user interests.

Additional content item	Precise Location	Cluster 1/Sports	Cluster 2/Architecture	…	Rank
http://www.rbb-online.de/nachrichten/...	Berlin	0.4	0		**2**
de.Wikipedia.org/prozess_gegen_…...	-	0	0.6		**1**

